TECNOLÓGICO DE MONTERREY®

Campus Ciudad de México

Escuela de Diseño, Ingeniería y Arquitectura

Maestría en Ciencias de la Computación

"Uso de patrones para el desarrollo de aplicaciones web seguras en un entorno financiero"

Autor:
Lic. Nayeli Violeta Villalpando Fernández

Director de la tesis:
Dr. José de Jesús Vázquez Gómez

Mayo 2012

Resumen

La globalización de la información ha llevado a las instituciones financieras del mundo, a desarrollar aplicaciones para su uso en internet, las cuales se han convertido en parte fundamental de sus procesos de negocio. Sin embargo, éstas se encuentran amenazadas por los cada vez más frecuentes ataques maliciosos que ocurren en internet; cuyo objetivo no solo es afectar el servicio ofrecido, sino también, el robo de la información confidencial de los clientes y de la misma empresa con el fin de venderla al mejor postor. Lo cual finalmente ocasiona pérdidas millonarias en imagen y disponibilidad, debido al mal uso de la información en operaciones fraudulentas, por mencionar algunos.

Con el fin de mantener la protección de la información manejada por estas instituciones, los organismos gubernamentales han establecido regulaciones para promover la implementación de mecanismos de seguridad en este tipo de aplicaciones. Y donde además, desde el punto de vista técnico, se ha observado que la integración de consideraciones de seguridad de la información en todo el ciclo de vida de desarrollo de software, es esencial para la creación de aplicaciones robustas y confiables las cuales mitiguen los riesgos que las amenazas existentes suponen.

Dentro de este contexto, el desarrollo de la presente tesis se enfocó a disminuir el número de vulnerabilidades de este tipo de aplicaciones, a través del uso de una metodología de desarrollo de software que incorpora actividades de seguridad y controles, mediante el empleo de patrones de seguridad, en todas las fases de desarrollo de la aplicación. Y la cual además, proponga una forma de cumplir con la normatividad general del sistema financiero mexicano en cuanto a seguridad de la información.

Contenido

Lista de Tablas

Lista de Figuras

CAPÍTULO 1

1 Introducción

1.1 Antecedentes

El rápido crecimiento y penetración de internet en los últimos años, ha originado un incremento en el número de usuarios que utilizan este servicio, lo cual ha contribuido, al aumento del volumen de transacciones emitidas entre las diferentes partes del mundo, tal como se muestra en la "Estadística mundial de internet y de la población" publicada por la Unión Internacional de Telecomunicaciones [1]. (Véase Tabla 1.1)

Regiones	Población (2011 Est.)	Usuarios, Dic. 31, 2000	Usuarios, Mrz. 31, 2011	% Población (Penetración)	Crecimiento (2000-2011)	% Uso Mundial
África	1,037,524,058	4,514,400	**118,609,620**	11.4 %	**2,527.4 %**	5.7 %
Asia	3,879,740,877	114,304,000	**922,329,554**	23.8 %	**706.9 %**	44.0 %
Europa	816,426,346	105,096,093	**476,213,935**	58.3 %	**353.1 %**	22.7 %
Oriente Medio	216,258,843	3,284,800	**68,553,666**	31.7 %	**1,987.0 %**	3.3 %
Norte América	347,394,870	108,096,800	**272,066,000**	78.3 %	**151.7 %**	13.0 %
Latinoamérica / Caribe	597,283,165	18,068,919	**215,939,400**	36.2 %	**1,037.4 %**	10.3 %
➢ **México**	113,724,226	2,712,400	**34,900,000**	30.7 %	**1,186.7 %**	-----
Oceanía / Australia	35,426,995	7,620,480	**21,293,830**	60.1 %	**179.4 %**	1.0 %
TOTAL MUNDIAL	**6,930,055,154**	**360,985,492**	**2,095,006,005**	**30.2 %**	**480.4 %**	**100.0 %**

Tabla 1.1 - Estadística mundial de internet y de la población.

Debido a esto, podemos aseverar que grandes volúmenes de información de todo tipo se transfieren cada segundo por este medio; desde cuestiones de carácter informativo hasta la transmisión de información crítica y confidencial.

Hoy en día, este envío y recepción de información es ejecutado comúnmente por aplicaciones web. Su relevancia en el mundo actual, no solo se debe a que estos sistemas transmiten, muestran información o son parte de una estrategia competitiva, sino porque son capaces de ejecutar complejos procesos sobre los datos de las compañías. Es así, como este conocimiento las sitúa dentro de la "lógica de negocio de las empresas", permitiéndoles manejar información crítica y confidencial de los usuarios así como de la misma compañía.

De esta manera, el desarrollo de las aplicaciones web se ha extendido en los diversos sectores de la industria, difundiéndose también dentro del sector financiero. Como ejemplos tenemos a: las compras en línea, la banca por internet, las transferencias electrónicas interbancarias y la publicación de indicadores económico-financieros por parte del gobierno; donde estos últimos, sirven de base a las empresas para la toma de decisiones, afectando así al progreso de sus negocios. En general, este tipo de transacciones tienen en común la transferencia de información crítica, es por ello que la participación de las aplicaciones financieras en el mundo es cada vez más importante para el progreso del comercio, las empresas y la economía a nivel global.

Sin embargo, junto con este gran auge, se debe prestar una atención especial a la seguridad de la información manejada por estas aplicaciones. Esto se debe, a que los delitos informáticos así como las actividades maliciosas a través de internet, han proliferado a la par junto con el avance de esta tecnología. Lo cual además se acentúa, debido al incremento en el número de incidentes dirigidos no solo hacia las computadoras de los usuarios finales sino también, hacia los sistemas de las compañías.

Con referencia a este punto, podemos analizar las últimas estadísticas publicadas por Deloitte, en las cuales durante el 2011, tan solo el 25% de las empresas entrevistadas no sufrió un ataque a la seguridad informática de sus negocios [2] (Véase Figura 1.1).

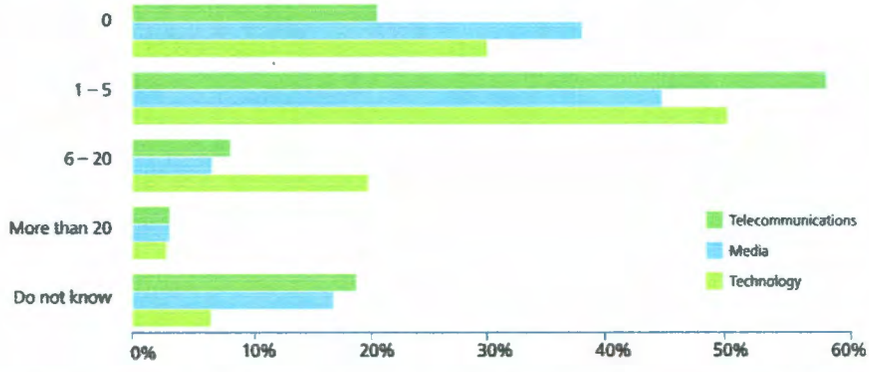


Figura 1.1 – Frecuencia en la que se experimentó un ataque a la seguridad de la información.

Una especial mención debe realizarse a los crímenes recientemente cometidos, los cuales se enfocan a la sustracción de información confidencial, hurtada normalmente a través del *phishing* (captación ilícita de datos personales) o por *spyware* (programas maliciosos ejecutándose directamente en las *PC's* del usuario final). Sin embargo, existen otros tipos de agresiones en donde los delincuentes se abren paso directamente hacia las aplicaciones web de las instituciones o bien hacia los servidores corporativos, sus redes y bases de datos para robar

este tipo de información a grandes volúmenes. Esto lo podemos consultar en el cuadro siguiente proporcionado por Kroll (Véase Tabla 1.2) dentro de su "Informe Global sobre Fraude" [3], el cual muestra el tipo de información a la que se dirigieron los ataques sufridos por la institución.

Tipo de información substraída	Porcentaje
Información de identificación personal - clientes	16.7%
Información de identificación personal - empleados	11.9%
Información de salud personal	2.8%
Datos de propiedad exclusiva, incluida la propiedad intelectual.	20.6%
Otro	6.5%
No sabe	8.3%
No hemos sufrido este tipo de fraude	47.4

Tabla 1.2 –Tipo de información a la que se enfoca los ataques.

Las consecuencias de estos delitos pueden ser mínimas como la denegación de servicios, o bien más impactantes como la substracción de información confidencial de los clientes o de la misma compañía, lo cual puede llegar a ocasionar grandes pérdidas financieras. Ponemon en su último reporte sobre el "Costo del Ciber-Crimen" [4], nos informa que el costo promedio anualizado ocasionado por los crímenes informáticos fue de 5.9 millones para el 2011, lo cual representa un incremento del 56% en comparación al costo promedio anual del año pasado. Más específicamente, los costos ocasionados por los diferentes tipos de actividades maliciosas pueden revisarse en la imagen siguiente (Véase Figura 1.2).

Figura 1.2 – Costo promedio anual del cibercrimen por frecuencia de ataque.

Ahora bien, con respecto al sector financiero, actualmente los criminales computacionales buscan substraer información la cual puedan convertir rápidamente en ganancias; y cómo el tipo de información manejada por las instituciones financieras ofrecen este tipo de características, éstas se han vuelto el principal objetivo de sus delitos. Esto se puede observar en las diferentes

estadísticas en seguridad, como el "Reporte de Amenazas y Riesgos del 2011" de X-Force [5], donde nos muestra que el 80% de las empresas expuestas a ataques *phishing* se encontraban dentro de este sector (Véase Figura 1.3).

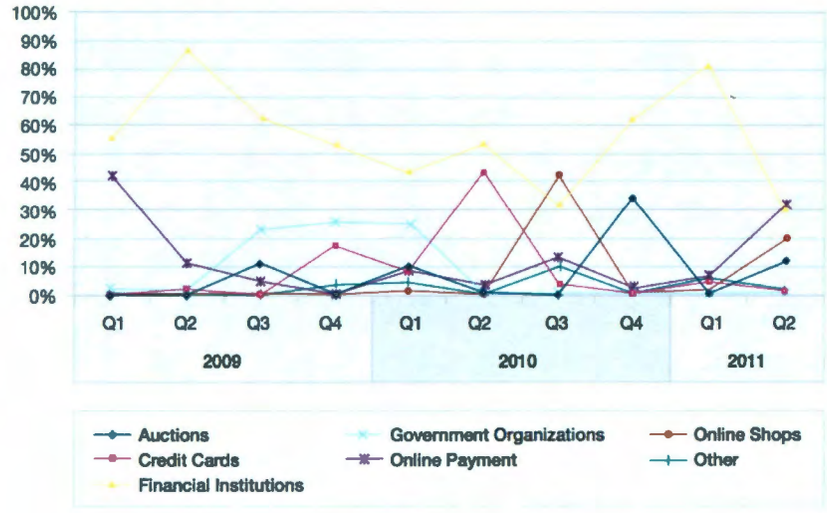


Figura 1.3 - Industrias afectadas por phishing.

Con referencia al costo promedio ocasionado por las actividades del ciber-crimen en el sector financiero, Phonemon [4], comenta que estos se elevaron a 14.70 millones de dólares en el 2011, a diferencia de los 12.37 millones alcanzados el año pasado.

Estas cifras como vemos, hacen ver a la seguridad como una necesidad en nuestros días, motivando a las empresas, a introducir controles de seguridad en sus aplicaciones con el fin de prevenir este tipo de ataques y las pérdidas que ocasionan.

1.2 Definición del problema

De acuerdo con los últimos reportes en seguridad como el "Reporte de Amenazas" de Sophos [6], más de 30,000 sitios web son infectados cada día, siendo el 80% de ellos, sitios web legítimos. De esta manera, las aplicaciones web se están convirtiendo en el centro principal de las actividades maliciosas, debido a que sus vulnerabilidades pueden ser explotadas para ganar acceso no autorizado a las computadoras sobre las cuales están corriendo dichas aplicaciones. De esta manera, los criminales pueden robar información confidencial o bien utilizar estos sitios para cometer otros delitos, como es el caso de la técnica *Blackhole*.

La intensidad de estas actividades maliciosas puede observarse claramente, a través del "Informe sobre Amenazas a la Seguridad en Internet" de Symantec [7], el cual nos muestra para

el año 2010 un crecimiento del 93% en los ataques basados en web con respecto al año anterior. (Véase Figura 1.4)

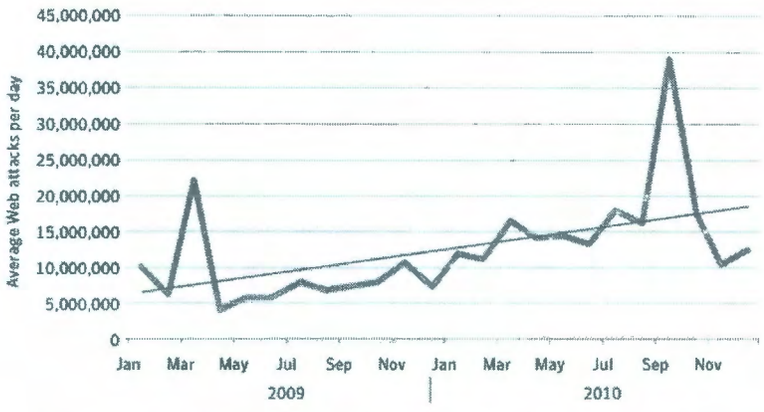


Figura 1.4 – Promedio diario de ataques basados en web.

Este tema, fue tratado también por Stephen Trilling, vicepresidente de Symantec Security Technology and Response [8], quién mencionó: "En el pasado, bastaba con aconsejar a los usuarios que evitaran los callejones oscuros de Internet. Hoy en día, los delincuentes se centran en vulnerar sitios web legítimos los cuales utilizan para lanzar sus ataques hacia los usuarios finales".

Esta situación además se intensifica, a raíz de las últimas estadísticas sobre seguridad, como aquellas publicadas por X-Force [5], las cuales muestran un incremento constante durante los últimos años, en el número de vulnerabilidades de seguridad detectadas en el software de los servidores web (Véase Figura 1.5). Donde el 37% de estas vulnerabilidades para el último año, pertenecen a las aplicaciones web.

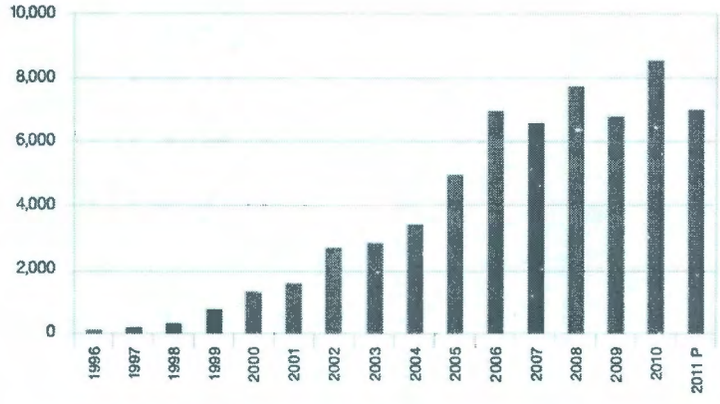


Figura 1.5 – Vulnerabilidades en la seguridad del software.

¿Y por qué las aplicaciones presentan este tipo de vulnerabilidades?

En las últimas décadas, la industria de desarrollo de software ha encaminado sus esfuerzos hacia la creación de sistemas informáticos que modelen de forma adecuada la "lógica del

negocio" de las empresas. Frente a esto, la ingeniería de software ha evolucionado para ofrecer metodologías, guías, arquitecturas, marcos de trabajo y patrones de diseño, entre otros, para la creación de sistemas robustos y eficientes los cuales den soporte a los procesos de las compañías.

Sin embargo, en muchas ocasiones estas aplicaciones, aunque han modelado eficientemente los procesos de negocio de las empresas, han dejado a un lado la incorporación de aspectos de seguridad dentro de las mismas. A tal suerte que, conforme diversos análisis del Centro de Coordinación de Emergencias Informáticas de la Universidad Carnegie Mellon (CERT/CC, por sus siglas en ingles), la mayoría de las vulnerabilidades de seguridad (más del 90 por ciento) detectadas en las aplicaciones, tienen su origen en defectos conocidos de software, introducidos durante el diseño y codificación de los sistemas; las cuales pudieron haberse evitado, de haberse seguido buenas prácticas en seguridad durante el desarrollo del mismo. [9]

Una de las principales razones por las cuales muchas veces no se integran controles de seguridad al inicio de la construcción de las aplicaciones, es debido a que normalmente los ciclos de vida utilizados, no incluyen actividades de seguridad dentro de ellos. Con respecto a esto, tenemos el análisis realizado por el Instituto en Seguridad Computacional [10], en donde el promedio global en el 2010, sobre si la empresa usa un proceso formal de desarrollo seguro de software, fue tan solo del 13.2%. (Véase Figura 1.6)

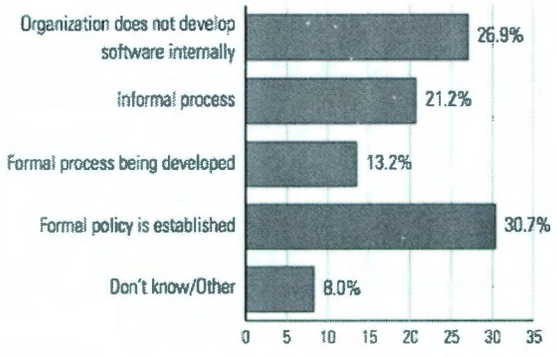


Figura 1.6 – Uso de un proceso de desarrollo seguro.

Otro de los aspectos que influyen en la falta de mecanismos de seguridad dentro de las aplicaciones, se debe a una filosofía de desconexión entre los profesionales de seguridad y los desarrolladores de sistemas. En donde los profesionales de seguridad sólo se centran en la implementación de controles (externos) de seguridad como *firewalls*, encripción y antivirus; los cuales no siempre previenen errores como la secuencia de comandos en sitios cruzados (XSS), la inyección de SQL o la inclusión de archivos. Mientras que los desarrolladores por su parte, se concentran únicamente en cómo construir un sistema "funcional" [9], [11], [12].

Esto además se ve reflejado dentro de las estadísticas publicadas por Ponemon sobre el "Estado de la Seguridad en las Aplicaciones Web" [13], donde el 43% de las compañías entrevistadas mencionaron que la seguridad no se considera una prioridad corporativa, así como los desarrolladores no presentan interés sobre la misma (Véase Tabla 1.3).

Razones por las que no se arreglan rápidamente las vulnerabilidades en el código	Porcentaje
La codificación segura requiere de recursos que no se tienen.	70%
Los desarrolladores no son responsables por la seguridad.	56%
Los desarrolladores están muy ocupados para responder a incidentes de seguridad	55%
No es una prioridad corporativa y a los desarrolladores no les importa.	43%
El código fuente es externo a los desarrolladores.	28%
No se tiene el código fuente.	16%

Tabla 1.3 –Principales razones por las que no se arreglan rápidamente las vulnerabilidades.

Finalmente otra de las causas, es la creencia generalizada por parte de las organizaciones, sobre que el desarrollo de software seguro y de alta calidad es demasiado costoso. Sin embargo, conforme a uno de los casos de estudio analizado por CERT/CC [14], se mostró que el costo por arreglar los problemas en los requerimientos durante las etapas avanzadas del proyecto Windows Vista de Microsoft, llegaban a los 2.5 millones de dólares, mientras los costos por arreglar estos problemas en las fases tempranas del ciclo de vida, fue tan solo de 500,000 dólares. Además, se redujo el número de vulnerabilidades en un 45%.

Es por estos motivos, que cada vez se hace más indispensable el uso de una metodología de desarrollo de software la cual obligue al desarrollador y en general a los administradores de proyectos, a llevar a cabo actividades de seguridad durante todo el ciclo de vida del sistema y no solamente durante las etapas finales del mismo. Complementando la misma, con guías y mecanismos los cuales permitan integrar más fácilmente controles de seguridad en las aplicaciones durante las etapas de diseño y codificación del sistema.

Con referencia a este último punto, recientemente ha surgido un nuevo tipo de tecnología denominada "patrones de seguridad" [1]. [15], los cuales tratan precisamente de solventar la brecha entre desarrolladores y profesionales de seguridad. Su objetivo es encapsular los conocimientos acumulados sobre seguridad en la forma de soluciones estructuradas que provean guías para el diseño y evaluación de sistemas seguros.

Luego entonces, el problema a tratar por esta investigación será el de disminuir el número de vulnerabilidades en las aplicaciones del sector financiero, las cuales involucren procesos o

[1] Para mayor referencia consultar el capítulo 4. "Patrones de seguridad"

transacciones web por internet con información confidencial. Para ello, se propondrá una metodología de desarrollo en donde se incorporen actividades y patrones de seguridad en todo el ciclo de vida del software. Puesto que, si los ataques maliciosos logran acceder a la información crítica manejada por éstas aplicaciones, pueden ocasionar faltas legales y grandes pérdidas económicas a las instituciones financieras.

1.3 Objetivos

Considerando la problemática actual, en cuanto al incremento del número de vulnerabilidades en las aplicaciones web, y enfocándonos a aquellas que manejan información confidencial dentro del sector financiero, la presente propuesta de tesis pretende incrementar la seguridad de las aplicaciones durante el desarrollo de las mismas. Para ello, haremos uso de diversos mecanismos de seguridad en la forma de patrones de seguridad.

De esta manera, el objetivo específico de la presente tesis será:

"El desarrollo de una metodología la cual incorpore, por medio del uso de patrones de seguridad, actividades y controles de seguridad al ciclo de vida de desarrollo de los sistemas pertenecientes al sector financiero. Los cuales, realicen transacciones con información confidencial a través de internet; cumpliendo con la normatividad general del sistema financiero mexicano".

Con este fin en mente, los objetivos particulares a desarrollar serán los siguientes:

- Identificar las actividades, los requerimientos y los controles generales de seguridad a establecer dentro del desarrollo de las aplicaciones financieras. Siendo además especificados por la normatividad del sistema financiero mexicano, así como por los estándares internacionales relevantes en este tema.
- Determinar aquellos patrones de seguridad que permitan implementar controles de seguridad para las aplicaciones financieras.
- El desarrollo de una metodología la cual incorpore el uso de estos patrones de seguridad durante el ciclo de vida de desarrollo de software.

1.3.1 Alcances

En concordancia con los objetivos establecidos, la metodología a generar se enfocará hacia el desarrollo de aplicaciones dentro del sistema financiero mexicano, las cuales manejen transacciones con información confidencial a través de internet. Sin embargo, dado la gama de aplicaciones diferentes a desarrollar, y el poco tiempo existente para un análisis exhaustivo de las mismas; en la presente investigación se decidió acotar la misma hacia las aplicaciones pertenecientes al grupo de intermediarios financieros en México, dado su importancia y vulnerabilidad. [2] Además, la metodología, al enfocarse en el desarrollo de aplicaciones seguras, aspectos como el establecimiento de un "Gobierno" y una "Infraestructura con seguridad", serán tomados como previamente definidos e implementados dentro de la institución.

1.4 Estructura del documento

El presente documento de tesis se encuentra divido en ocho capítulos, a continuación se presentan los aspectos más relevantes de cada uno de ellos:

Capítulo 2. Estado del arte. Expone los resultados de la investigación bibliográfica sobre la legislación nacional e internacional que rige la seguridad informática en las aplicaciones financieras, permitiendo identificar los requerimientos de seguridad a satisfacer. Posteriormente, analiza las actividades en cuanto a seguridad que proponen las metodologías comerciales para el desarrollo de sistemas, las cuales comunmente son usadas por las instituciones financieras.

Capítulo 3. Seguridad en las aplicaciones financieras. Revisa los conceptos básicos de seguridad para los sistemas financieros. Así como los ataques más comunes contra este tipo de aplicaciones, sus vulnerabilidades y sus riesgos. Para finalmente determinar los controles mínimos de seguridad a implementar.

Capítulo 4. Patrones de seguridad. Describe a grandes rasgos el elemento fundamental de nuestra investigación, en donde los patrones de seguridad se establecen como una solución efectiva para la incorporación de mecanismos de seguridad durante el ciclo de vida de los sistemas. Demostrando que su aplicación nos ayudará a satisfacer la mayor parte de los controles de seguridad identificados para las aplicaciones financieras.

---

[2] Para conocer más sobre los Intermediarios Financieros y la normatividad aplicable, favor de referirse a la sección 2.1 "Normas para el desarrollo de aplicaciones financieras".

Capítulo 5. Metodología para el desarrollo de aplicaciones web financieras seguras. Expone la propuesta de la solución, la cual consiste en el desarrollo de una metodología con actividades de seguridad, que incorpora el uso de patrones de seguridad; permitiendo así el desarrollo de aplicaciones web financieras con un mínimo de vulnerabilidades. Adicionalmente, describe las actividades a seguir para su aplicación y evaluación.

Capítulo 6. Caso de estudio. Describe a grandes rasgos el desarrollo de una aplicación web financiera; realizada primero con un ciclo de vida común de desarrollo de software, y luego a través de la aplicación de la metodología propuesta.

Capítulo 7. Análisis e interpretación de resultados. Presenta la evaluación de los resultados que se obtuvieron al aplicar la metodología propuesta al caso de estudio. Los resultados son analizados en función de las variables definidas en la sección anterior.

Capítulo 8. Conclusiones de la Investigación. Finalmente se presenta las conclusiones del estudio, sus contribuciones, las limitaciones surgidas dentro de su desarrollo y las líneas de investigación identificadas. Donde estas últimas, podrían convertirse en trabajos futuros, las cuales aporten y den continuidad al presente trabajo.

Para una mayor comprensión de la estructura del presente documento, a continuación se muestra el mapa conceptual del mismo.

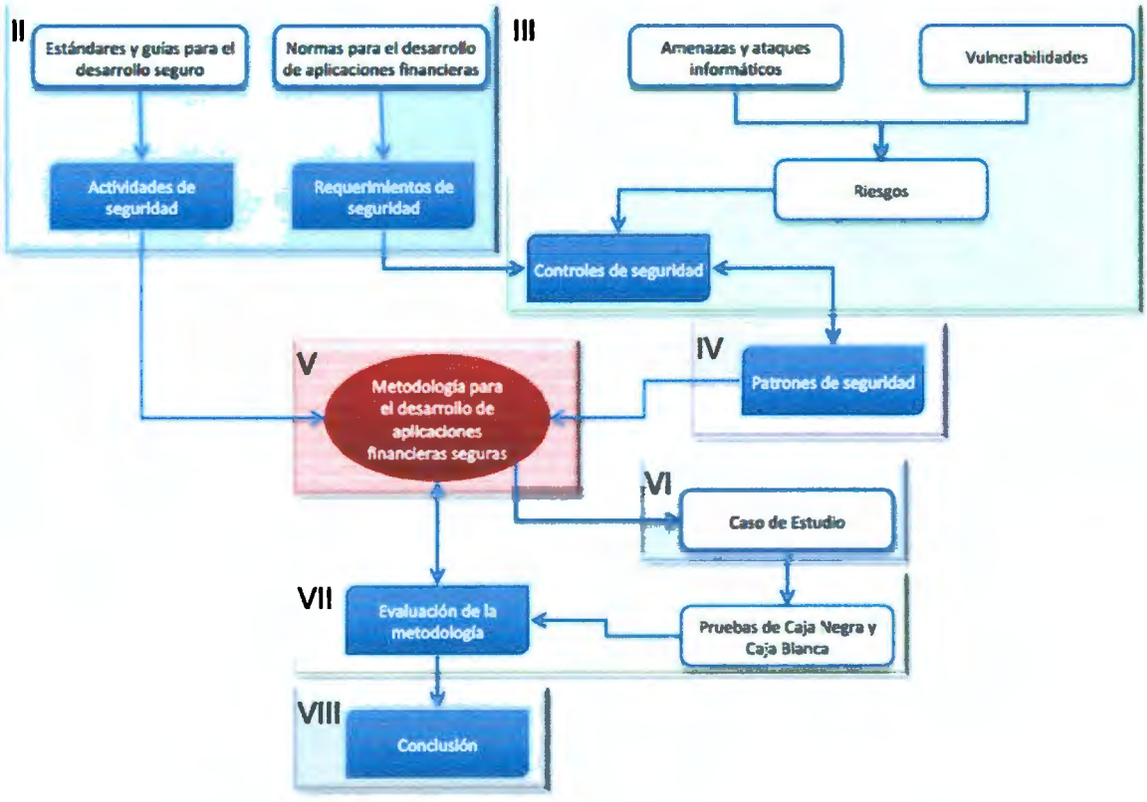


Figura 1.7 – Mapa conceptual de la investigación.

1.5 Resultados esperados

Como mencionamos en el objetivo, por medio del presente trabajo de tesis se desarrollará una metodología de desarrollo de sistemas, la cual contemple aquellos patrones de seguridad que permitan incorporar controles de seguridad a las aplicaciones financieras desde el inicio de su desarrollo, mejorando así la seguridad de las mismas.

Conforme a esto, los beneficios mínimos esperados, debido a la incorporación de "buenas prácticas" de seguridad en el desarrollo del software, son:

- Disminuir el número de vulnerabilidades de las aplicaciones financieras.
- Aumentar el nivel de seguridad de las aplicaciones financieras.
- Proteger la información confidencial manejada por las aplicaciones financieras.
- Optimizar el desarrollo de aplicaciones financieras seguras, las cuales cumplan con los estándares del sector financiero mexicano.

CAPÍTULO 2

2 Estado del Arte

Como se comentaba en la "Introducción" del presente escrito, los ataques dirigidos hacia los sistemas informáticos de las empresas, y en especial a aquellos pertenecientes al sector financiero, se están volviendo cada vez más frecuentes y agresivos. En donde éstos, no solo impactan en la infraestructura, sino también en la información crítica y lógica del negocio.

Es por esto que, actualmente es necesaria y urgente una cultura de seguridad de la información dentro de la organización; la cual se permee a cada una de las etapas del ciclo de vida de desarrollo de sistemas usados por la empresa, con el fin de reducir el número de vulnerabilidades presentadas. Para ello, las instituciones financieras con el fin de desarrollar y revisar sus controles, políticas, procedimientos y/o procesos sobre la seguridad de la información, se pueden auxiliar de una variedad de recursos a los cuales recurrir. [16]

El primero y más importante de ellos, son las **leyes federales o regulaciones** nacionales para el sector financiero, éstos incorporan una serie de artículos encargados de reglamentar a nivel general la seguridad en estas instituciones. Donde generalmente, este tipo de regulaciones terminan por concretarse en la implementación de políticas institucionales y controles para mantener la seguridad, incluyendo en su alcance a los sistemas de información de la empresa.

El segundo de ellos, son los **estándares en materia de seguridad de la información** desarrollados por diversas organizaciones nacionales e internacionales. Si bien estos estándares normalmente no se enfocan hacia las instituciones financieras, muchas de ellas los utilizan debido a que ofrecen un conjunto de mejores prácticas para el establecimiento de controles de seguridad, los cuales, les permiten cumplir total o parcialmente con las disposiciones establecidas por las leyes en materia de seguridad de la información.

De esta manera, en el presente capítulo se comenzará por revisar el marco legal existente en cuanto a seguridad de la información para el sector financiero mexicano, complementando el mismo con algunas normas internacionales relevantes. Posteriormente, nos enfocaremos a aquellos estándares y metodologías que incorporan actividades de seguridad dentro del ciclo de vida de desarrollo de sistemas, y son utilizados por las instituciones financieras como un marco de trabajo el cual les permite cumplir con las disposiciones vigentes en materia de seguridad.

2.1 Normas para el desarrollo de aplicaciones financieras

En concordancia con los objetivos descritos en el capítulo anterior, la presente tesis se enfoca al desarrollo de aplicaciones web, encaminadas a manejar transacciones seguras por internet dentro del sistema financiero mexicano. Conforme a este criterio, a continuación se presenta una breve descripción del mismo, lo cual nos permitirá enmarcar y justificar el alcance de las normas utilizadas para el desarrollo del presente trabajo.

Comenzaremos por definir al Sistema Financiero Mexicano, el cuál: [17]

"Procura la asignación eficiente de recursos entre ahorradores y demandantes de crédito. Se encuentra constituido por un conjunto de instituciones que captan, administran y canalizan el ahorro de las personas hacia la inversión. Se conforma por: grupos financieros, la banca comercial, las administradoras de fondos para el retiro (Afores), las aseguradoras, las sociedades financieras de objeto limitado (Sofoles), la banca de desarrollo, las casas de bolsa, las sociedades de inversión, las arrendadoras financieras, las afianzadoras, los almacenes generales de depósito, las uniones de crédito, las casas de cambio y las empresas de factoraje entre otras."

Este conjunto de instituciones, pueden dividirse para su estudio en tres grandes grupos:

- **Mercados Financieros.** Son espacios físicos o virtuales, así como el conjunto de reglas los cuales permiten a los inversionistas, emisores, intermediarios y personas recibir o dar financiamiento, comprar y vender divisas y acciones.
- **Intermediarios Financieros.** Son instituciones que actúan como mediadores entre aquellos quienes desean recibir recursos y quienes desean invertirlos, logrando con esto transformar plazos, montos, riesgos y reducir costos.
- **Sistemas de Pagos.** Están constituidos por un conjunto de instrumentos, procedimientos y normas las cuales permiten transferir recursos financieros entre sus participantes.

Como podremos observar, las instituciones financieras se concentran principalmente dentro del grupo de los intermediarios financieros. Siendo éste la principal agrupación, dado que maneja el 85 por ciento de los activos del sistema financiero en nuestro país [18]. Un sistema financiero sano no solo requiere de intermediarios eficaces y solventes, y de mercados eficientes y completos, sino también de un marco legal que establezca claramente los derechos y obligaciones de las partes involucradas. En este sentido, las instituciones financieras en México se encuentran reguladas por la Secretaría de Hacienda y Crédito Público (SHCP) y son supervisadas por el Banco de México (BANXICO).

Existen además, otros organismos nacionales encargados de promover regulaciones hacia las instituciones financieras como [19]: la Comisión Nacional Bancaria y de Valores (CNBV), la

Comisión Nacional de Seguros y Fianzas (CNSF), la Comisión Nacional del Sistema de Ahorro para el Retiro (CONSAR), la Comisión Nacional para la Protección y Defensa de los Usuarios de los Servicios Financieros (CONDUSEF), y el Instituto para la Protección del Ahorro Bancario (IPAB). No obstante, el alcance de estas leyes es parcial dependiendo del giro de la institución.

Con los anteriores criterios en mente, la presente investigación se concentró en revisar las normas de carácter general en cuanto a seguridad, las cuales aplican a las instituciones dentro del grupo de intermediarios financieros [20], y las cuales son dictadas por los organismos gubernamentales más importantes en este rubro (la SHCP y BANXICO). Adicionando además, las normas de carácter financiero emitidas por la CNBV; dado que ésta institución es la responsable de supervisar y regular en el ámbito de su competencia a las entidades integrantes del sistema financiero mexicano [21] . Finalmente, analizaremos la legislación internacional de mayor relevancia para este tema. Concluyendo, con un análisis sobre los requerimientos mínimos de seguridad, que deben de cumplir las instituciones financieras en sus aplicaciones.[3]

2.1.1 Legislación nacional

Dentro de la legislación general que rige al sistema financiero mexicano, el presente estudio concluyó, la no existencia hasta el momento de una normatividad específica encargada de dictaminar los requerimientos mínimos de seguridad que las instituciones financieras debieran de implementar en sus sistemas informáticos. Sin embargo, dentro de la legislación existente para las mismas, podemos encontrar algunos artículos que, directa o indirectamente proporcionan criterios a seguir para el establecimiento de dicha seguridad.

2.1.1.1 Ley para regular las agrupaciones financieras

La normativa básica para las instituciones financieras, abarca tanto a la Ley para regular las agrupaciones financieras, como a la Ley de instituciones de crédito. En cuanto a la primera, ésta fue promulgada por el H. Congreso de la Unión en 1990, y ha sido modificada en diversas ocasiones por la Secretaría de Hacienda y Crédito Público; siendo su principal objetivo [22]:

"Regular las bases de organización y funcionamiento de los grupos financieros; establecer los términos bajo los cuales habrán de operar, así como la protección de los intereses de quienes celebren operaciones con los integrantes de dichos grupos."

[3] A continuación, se expondrá un breve resumen del contenido de los artículos de estas leyes, los cuales se consideraron más importantes debido a su relación con la seguridad en las aplicaciones financieras. Para mayor referencia, se sugiere consultar directamente las leyes a tratar.

Analizando, la presente ley se enfoca en establecer la normativa a seguir para la constitución y funcionamiento de los grupos financieros; indicando cómo éstos serán integrados y vigilados por una sociedad controladora, supervisada adicionalmente por la Comisión Nacional Bancaria y de Valores. Además de esta situación, la ley no contiene artículos relevantes en los que se establezcan o mencionen requerimientos o controles de seguridad para los sistemas informáticos de las Instituciones.

2.1.1.2 Ley de instituciones de crédito

En cuanto a la Ley de instituciones de crédito, ésta también ha sido decretada por el H. Congreso de la Unión y modificada por la Secretaría de Hacienda y Crédito Público. Su finalidad es [23]:

"Regular el servicio de banca y crédito, la organización y funcionamiento de las instituciones de crédito, las actividades y operaciones que las mismas podrán realizar, su sano y equilibrado desarrollo, la protección de los intereses del público y los términos en que el Estado ejercerá la rectoría financiera del Sistema Bancario Mexicano".

Revisando los artículos contenidos en la misma, los requerimientos de seguridad identificados en este estudio fueron los siguientes:

Artículo	Requerimiento de seguridad
10-IV. La solicitud de autorización para organizarse y operar como una institución de banca múltiple, deberá tener un plan general de funcionamiento que comprenda las medidas de seguridad para preservar la integridad de la información.	Contar con políticas de seguridad Garantizar la integridad de la información
45-Q. Las instituciones de banca múltiple, deberán adoptar las medidas de control interno y contar con sistemas informáticos y de contabilidad, que aseguren su independencia operativa con respecto a cualquiera de los demás integrantes del grupo empresarial al que pertenezcan.	Independencia operativa con otros sistemas Garantizar disponibilidad de los servicios
46 Bis. Las instituciones de banca múltiple deberán contar con la infraestructura y los controles internos necesarios para realizar sus operaciones, tales como sistemas operativos, contables y de seguridad, y sus manuales respectivos.	Documentación operativa de procesos y herramientas Contar con una infraestructura de TI adecuada a las funciones
46 Bis 1. Al pactar con terceros, establecer los lineamientos técnicos y operativos para salvaguardar la confidencialidad de la información de los usuarios.	Garantizar la confidencialidad de la información por parte de terceros
52. Las instituciones de crédito podrán pactar la celebración de sus operaciones y la prestación de servicios con el público mediante el uso de equipos, medios electrónicos, ópticos o de cualquier otra tecnología, sistemas automatizados de procesamiento de datos y redes de telecomunicaciones, ya sean privados o públicos; estableciendo para ello los contratos respectivos.	Manejo de contratos de servicios con el usuario Contar con una infraestructura de TI adecuada a las funciones

96. Establecer medidas básicas de seguridad que incluyan la instalación y funcionamiento de los dispositivos, mecanismos y equipo indispensable, con objeto de contar con la debida protección en las oficinas bancarias.	Infraestructura de seguridad
100. Las instituciones de crédito podrán microfilmar o grabar en discos ópticos, o en cualquier otro medio, todos aquellos libros, registros y documentos en general, relacionados con los actos de la propia institución. Encontrándose obligadas a la conservación de los mismos.	Manejo de respaldos de la información Conservar la información Constancia electrónica auditable de las operaciones
110 Bis 11.- Las notificaciones por medios electrónicos, con acuse de recibo, podrán realizarse cuando el interesado lo haya aceptado o solicitado expresamente por escrito a las autoridades financieras a través de los sistemas automatizados y mecanismos de seguridad que las mismas establezcan.	Garantizar el no repudio
Artículo 112 Bis. Aplicación de sanciones a quien en forma indebida: Obtenga, comercialice o use la información sobre clientes, cuentas u operaciones de las instituciones de crédito. Así como altere, copie o reproduzca el medio de identificación electrónica con estos datos. Y sustraiga, copie o reproduzca información confidencial o reservada.	Detección y notificación de delitos o uso indebido de la información
112 Quáter. Se sancionará a quién acceda, altere o modifique a los equipos o medios electrónicos, ópticos o de cualquier otra tecnología del sistema bancario mexicano, para obtener recursos económicos, información confidencial o reservada.	Detección y notificación de delitos o uso indebido de la información Controles para actualizar y acceder a la información
115. Las instituciones de crédito y sociedades financieras estarán obligadas a establecer medidas y procedimientos para prevenir y detectar actos, omisiones u operaciones que pudieran favorecer, prestar ayuda, auxilio o cooperación de cualquier especie para la comisión de los delitos. Además, deberá resguardar y garantizar la seguridad de la información y documentación relativas a la identificación de sus clientes y usuarios.	Detección y notificación de delitos o uso indebido de la información Detección de faltantes Manejo de respaldos de la información Garantizar la confidencialidad de la información
Artículo 133. La CNBV podrá efectuar visitas a las instituciones de crédito, que tendrán por objeto revisar, verificar, comprobar y evaluar las operaciones, organización, funcionamiento, los procesos, los sistemas de control interno, de administración de riesgos y de información.	Realización de auditorías Administración de riesgos
Artículo 134 Bis 3. Las instituciones de banca múltiple deberán clasificar la información relativa a las operaciones, en los sistemas automatizados de procesamiento y también la conservación de datos.	Clasificar información Conservar la información

Tabla 2.1 –Requerimientos de seguridad en la Ley de instituciones de crédito

2.1.1.3 Ley para la transparencia y ordenamiento de los servicios financieros

Adicional a la normativa básica para las Instituciones financieras, la presente investigación consideró por su carácter general a la Ley para la transparencia y ordenamiento de los servicios financieros, la cual tiene por objeto [24]:

"Regular las Comisiones y Cuotas de Intercambio así como otros aspectos relacionados con los servicios financieros y el otorgamiento de créditos de cualquier naturaleza que realicen las Entidades, con el fin de garantizar la transparencia, la eficiencia del sistema de pagos y proteger los intereses del público."

Bajo una perspectiva de tecnologías de información, los artículos que se consideraron relevantes para el presente estudio, fueron los siguientes:

Artículo	Requerimiento de seguridad
11. Los Contratos de Adhesión que utilicen las Entidades Financieras para documentar operaciones masivas deberán estar escrito en idioma español y deberá contener la firma o huella digital del Cliente o su consentimiento expreso por los medios electrónicos que al efecto se hayan pactado.	Manejo de contratos de servicios con el usuario Autenticación de los usuarios Garantizar el no repudio
22. b) Cuando las disposiciones, actos administrativos y notificaciones del Banco de México se envíen a las instituciones de crédito, entidades o intermediarios financieros, a través de medios electrónicos, que permitan adjuntar el mensaje de datos y firmarlo electrónicamente, las firmas respectivas deberán corresponder a los funcionarios competentes, y haber sido generadas con base en los datos de creación de firma electrónica conforme a los procedimientos y sistemas de la Infraestructura Extendida de Seguridad que administra el propio Banco de México.	Garantizar la confidencialidad de la información transmitida Administración de certificados Autenticación del personal facultado para realizar operaciones
42. La Comisión Nacional para la Protección y Defensa de los Usuarios de Servicios Financieros sancionará con multa a las Instituciones que no cuenten en su página electrónica en "Internet", con la información actualizada relativa a los montos, conceptos y periodicidad de las Comisiones.	Garantizar disponibilidad de la información Garantizar tiempo de respuesta en la divulgación de información

Tabla 2.2 –Requerimientos de seguridad en la Ley para la transparencia de servicios financieros

2.1.1.4 Ley de protección y defensa al usuario de servicios financieros

Esta ley promulgada por el H. Congreso de la Unión y modificada por la Secretaría de Hacienda y Crédito Público, el 30 de agosto de 2011, tiene como finalidad [25]:

"La protección y defensa de los derechos e intereses del público usuario de los servicios financieros, que prestan las instituciones públicas, privadas y del sector social debidamente autorizadas, así como regular la organización, procedimientos y funcionamiento de la entidad pública encargada de dichas funciones."

Dado su carácter de resguardo hacia la información privada de los usuarios de los servicios financieros, el estudio de la presente ley identificó los siguientes requerimientos de seguridad a contemplar dentro de las instituciones financieras:

Artículo	Requerimiento de seguridad
8o. La Comisión Nacional establecerá y mantendrá actualizado, un Registro de Usuarios que no deseen que su información sea utilizada para fines mercadotécnicos o publicitarios.	Contar con políticas de privacidad
94. La Comisión Nacional estará facultada para imponer sanciones a la Institución Financiera que envíe directamente o por interpósita persona cualesquiera publicidad relativa a los productos y servicios que ofrezcan las mismas Instituciones Financieras a aquellos Usuarios que expresamente hayan solicitado que no se les envíe dicha publicidad.	Contar con políticas de privacidad
68. La Comisión Nacional podrá en todo momento, requerir a la institución financiera la entrega de cualquier información, documentación o medios electromagnéticos que requiera con motivo de una reclamación.	Constancia electrónica auditable de las operaciones

Tabla 2.3 –Requerimientos de seguridad en la Ley de protección al usuario de servicios financieros

2.1.1.5 Disposiciones de carácter general aplicables a las instituciones de crédito

Finalmente, para enriquecer los requerimientos y controles de seguridad dentro de las aplicaciones del sector financiero, se analizó la circular única bancaria expedida por la CNBV [26], cuyo objetivo es:

"Compilar en un solo instrumento jurídico, las disposiciones aplicables a las Instituciones de Crédito expedidas por la CNBV, a fin de brindar certeza jurídica en cuanto al marco normativo, al que las entidades financieras deberán sujetarse para el desarrollo de sus operaciones".

La circular, modificada recientemente para cumplir con las mejores prácticas internacionales (sección: "Exposición de Motivos" del 27 de enero de 2010), ha agregado una serie de normativas encaminadas a:

"Definir controles específicos que deberán observar las instituciones de crédito de acuerdo con el grado de riesgo en la realización de operaciones a través del uso de medios electrónicos, tales como operaciones en cajeros automáticos, pagos mediante terminales punto de venta, pagos y operaciones mediante teléfonos móviles, operaciones mediante banca por Internet, operaciones a través del servicio host to host, operaciones mediante banca telefónica audio respuesta y voz a voz u otros medios electrónicos, a fin de proteger tanto a los usuarios como a las propias instituciones de crédito."

Conforme a ello, a grandes rasgos los mecanismos de seguridad que esta circular establece son:

Artículo	Requerimiento de seguridad
11. Las Instituciones en el desarrollo de la Actividad Crediticia, deberán contar con procesos, personal y sistemas de cómputo que permitan el logro de sus objetivos. El director general, deberá asegurarse que la infraestructura de apoyo, no contravenga los objetivos, lineamientos y políticas aprobados por el Consejo.	Involucrar a la alta dirección Contar con una infraestructura de TI adecuada a las funciones
12. Las Instituciones deberán contar con sistemas de información de crédito, los cuales deberán: permitir la debida interrelación e interfaces entre las distintas áreas; generar reportes confiables; evitar entradas múltiples y la manipulación de datos; así como permitir la conciliación automática, oportuna y transparente de la contabilidad, y mantener controles adecuados que garanticen la confidencialidad de la información, procuren su seguridad tanto física como lógica, así como medidas para la recuperación de la información en casos de contingencia.	Integración segura con otros sistemas Contar con una infraestructura de seguridad Recuperación de información
31. El área responsable de la función de auditoría interna deberá revisar los sistemas de información de crédito, respecto de: el cumplimiento a las modificaciones, actualizaciones, mejoras e innovaciones propuestas; la calidad y veracidad de la información emitida, y la oportunidad y periodicidad de los reportes.	Realización de auditorías Verificar que los sistemas cumplan con los requerimientos Garantizar tiempo de respuesta en la divulgación de información
47. Las Instituciones, cuando pacten con terceros deberán prever mecanismos y controles que les permitan verificar la adecuada integración de expedientes por parte de dichos terceros; así como su conservación e inmediata disposición.	Garantizar la integridad de información por parte de terceros Garantizar disponibilidad de información por parte de terceros
48. Los expedientes se podrán mantener en papel o mediante archivos electrónicos, siempre y cuando estén disponibles para consulta del personal debidamente facultado. Se deberán implementar controles que permitan conocer la ubicación de cada documento y la identidad del funcionario responsable.	Conservar la información Autenticación del personal facultado para realizar operaciones
49. La información y documentación deberá mantenerse actualizada; contando con mecanismos de control, verificación y procedimientos que permitan detectar faltantes y su regularización.	Garantizar disponibilidad de la información Detección de faltantes
60. Las Instituciones deberán establecer sistemas automatizados de información que permitan la obtención de reportes periódicos y oportunos sobre los riesgos totales a cargo de sus deudores.	Garantizar disponibilidad de la información Garantizar tiempo de respuesta en la divulgación de información
66. Entre los riesgos a los que se encuentran expuestas las Instituciones (y que debe considerarse en la Administración Integral de Riesgos), se encuentra el riesgo tecnológico, el cual se define como: la pérdida potencial por daños, interrupción, alteración o fallas derivadas del uso o dependencia en el hardware, software, sistemas, aplicaciones, redes y cualquier otro canal de distribución de información en la prestación de servicios bancarios con los clientes de la Institución.	Administración de riesgos Clasificar tipos de riesgos

76. Las instituciones de banca múltiple deberán contar con un área de auditoría interna, que lleve a cabo una auditoría de Administración Integral de Riesgos que contemple: la suficiencia, integridad, consistencia y grado de integración de los sistemas de procesamiento de información y de su documentación.	Realización de auditorías Integración segura con otros sistemas Documentación técnica del sistema
78. Los manuales para la Administración Integral de Riesgos deberán ser documentos técnicos que contengan, los diagramas de flujo de información, modelos y metodologías para la valuación de los distintos tipos de riesgo, así como de los requerimientos de los sistemas de procesamiento de información.	Administración de riesgos Clasificar tipos de riesgos Documentación de requerimientos
86. Sobre riesgos cuantificables no discrecionales las Instituciones deberán asegurar:	Administración de riesgos Clasificar tipos de riesgos
II. a) La implementación de controles internos que procuren la seguridad en las operaciones, verificar la delimitación de funciones y los niveles de autorización.	Manejo de niveles de autorización y acceso al sistema
II. c) La existencia de sistemas de procesamiento de información para la administración de riesgos, que permitan restablecer los niveles mínimos de la operación del negocio ante fallas técnicas, eventos fortuitos o de fuerza mayor.	Restablecimiento de operaciones
III. b) 1. Evaluar la vulnerabilidad en el hardware, software, sistemas, aplicaciones, seguridad, recuperación de información y redes, por errores de procesamiento u operativos, fallas en procedimientos, capacidades inadecuadas e insuficiencias de los controles instalados.	Pruebas para la evaluación de vulnerabilidades
III. b) 2. i. Mantener políticas y procedimientos que aseguren en todo momento el nivel de calidad del servicio y la seguridad e integridad de la información.	Contar con políticas de seguridad Garantizar disponibilidad de la información Garantizar la integridad de la información
III. b) 2. ii. Asegurar que cada operación realizada por los usuarios deje constancia electrónica que conforme registros de auditoría.	Constancia electrónica auditable de las operaciones
III. b) 2. iii. Implementar mecanismos que midan y aseguren niveles de disponibilidad y tiempos de respuesta, que garanticen la adecuada ejecución de las operaciones y servicios.	Garantizar disponibilidad de los servicios
III. b) 3. i. Al realizar operaciones con clientes a través de Internet, cajeros automáticos, la banca telefónica o sucursales, se deberá establecer medidas y controles que permitan asegurar la confidencialidad en la generación, almacenamiento, transmisión y recepción de claves de identificación y acceso para los usuarios.	Garantizar la confidencialidad de la información transmitida
III. b) 3. ii. Implementar medidas de control que garanticen la protección, seguridad y confidencialidad de la información generada por la realización de operaciones a través de cualquier medio tecnológico.	Garantizar la integridad de la información Garantizar la confidencialidad de la información
III. b) 3. iii. Contar con esquemas de control y políticas de operación, autorización y acceso a los sistemas, bases de datos y aplicaciones implementadas para la realización de operaciones a través de cualquier medio tecnológico.	Manejo de niveles de autorización y acceso al sistema Contar con políticas de seguridad
III. b) 3. iv. Incorporar los medios adecuados para respaldar y recuperar la información que se genere de las operaciones.	Manejo de respaldos de la información Recuperación de información

III. b) 3. v. Diseñar planes de contingencia, a fin de asegurar la capacidad y continuidad de los sistemas implementados.	Administración de planes de contingencia
III. b) 3. vi. Establecer mecanismos para la identificación y resolución de aquellos actos o eventos que puedan generar riesgos derivados de: actos u operaciones fraudulentas a través de medios tecnológicos, contingencias generadas en los sistemas y el uso inadecuado por parte de los usuarios de los canales de distribución.	Monitoreo y control de riesgos Restablecimiento de operaciones
141. El Consejo, a propuesta del Comité de Auditoría deberá conocer y, en su caso, aprobar los objetivos del Sistema de Control Interno y los lineamientos para su implementación, dentro de los cuales se incluirán los que regulen y controlen lo relativo a la instalación y uso de los sistemas automatizados de procesamiento de datos y redes de telecomunicaciones.	Involucrar a la alta dirección Contar con una infraestructura de seguridad
160. El área de Auditoría Interna deberá verificar que los sistemas informáticos, cuenten con mecanismos para preservar la integridad, confidencialidad y disponibilidad de la información; eviten su alteración y cumplan con los objetivos para los cuales fueron implementados o diseñados. Asimismo, vigilarlos a fin de identificar fallas potenciales y verificar que éstos generen información suficiente, consistente y que fluya adecuadamente. Verificar que la Institución cuente con planes de contingencia y medidas necesarias para evitar pérdidas de información, así como para, su recuperación o rescate.	Realización de auditorías Garantizar la integridad de la información Garantizar la confidencialidad de la información Garantizar disponibilidad de la información Administración de planes de contingencia Recuperación de información
164. La Dirección General será la responsable de la debida implementación del Sistema de Control Interno, la cual deberá:	Involucrar a la alta dirección
IV. a) 3. Delimitar las facultades entre el personal que autorice, ejecute, vigile, evalúe, registre y contabilice las transacciones, evitando su concentración en una misma persona.	Asignación de roles y responsabilidades
IV. b) 2. Proporcionar información en forma oportuna al personal que corresponda conforme a su nivel jerárquico y facultades.	Garantizar disponibilidad de la información Manejo de niveles de autorización y acceso al sistema
IV. b) 3. Procesar, utilizar y conservar información relativa a cada transacción, con grado de detalle suficiente; utilizando mecanismos de seguridad que permitan su consulta sólo al personal autorizado y que limiten su modificación.	Conservar la información Autenticación del personal facultado para realizar operaciones Garantizar la integridad de la información
IV. f) Proteger la integridad y adecuado mantenimiento de los sistemas informáticos, incluidos los sistemas automatizados de procesamiento de datos y redes de telecomunicaciones, así como la inalterabilidad, confidencialidad y disponibilidad de la información procesada, almacenada y transmitida, determinando los mecanismos de respaldo de la información, así como los planes de contingencia.	Contar con una infraestructura de seguridad Garantizar la integridad de la información Garantizar la confidencialidad de la información Garantizar disponibilidad de la información Manejo de respaldos de la información Administración de planes de contingencia
IV. g) Proponer medidas para evitar que terceros o personal de la Institución, utilicen a los sistemas para la comisión de actos ilícitos o irregularidades.	Detección y notificación de delitos o uso indebido de la información

IV. h) Procurar que se observen procedimientos, estructuras organizacionales y políticas de seguridad informática adecuadas a la Institución.	Contar con políticas de seguridad
V. a) Prever de medidas, a fin de que los sistemas informáticos de las Instituciones, tanto para realizar sus operaciones como para la prestación de servicios al público, realicen, en todo momento, las funciones para las que fueron diseñados, desarrollados o adquiridos.	Garantizar disponibilidad de los servicios Verificar que los sistemas cumplan con los requerimientos
V. b) Estén debidamente documentadas sus aplicaciones y procesos, incluyendo su metodología de desarrollo, así como los registros de sus cambios.	Documentación técnica del sistema
V. c) Sean probados antes de ser implementados, al realizar cambios sobre los mismos, así como al aplicar actualizaciones, utilizando mecanismos de control de calidad.	Verificar que los sistemas cumplan con los requerimientos
V. d) Cuenten con las licencias o autorizaciones de uso y hayan sido probados antes de ser implementados.	Uso de licencias Verificar que los sistemas cumplan con los requerimientos
V. e) Cuenten con controles tanto de seguridad que protejan la confidencialidad de la información, como de acceso para garantizar la integridad de los sistemas y de la información generada, almacenada y transmitida por éstos. Dichas medidas serán acordes con el grado de criticidad de la información.	Garantizar la confidencialidad de la información Controles para actualizar y acceder a la información Garantizar la integridad de la información Clasificar información
V. f) Minimicen el riesgo de interrupción de la operación con base en mecanismos de respaldo y procedimientos de recuperación de la información, así como de la infraestructura tecnológica para su procesamiento.	Garantizar disponibilidad de los servicios Manejo de respaldos de la información Recuperación de información Restablecimiento de operaciones
V. g) Mantengan registros de auditoría, incluyendo la información detallada de la operación o actividad efectuadas por los usuarios.	Constancia electrónica auditable de las operaciones
V. h) Realizar pruebas tendientes a detectar vulnerabilidades en los medios electrónicos, de telecomunicaciones y equipos automatizados, que prevengan el acceso y uso no autorizado.	Pruebas para la evaluación de vulnerabilidades
VIII. Dictar las medidas necesarias para que en el manejo de la información relativa a los clientes de la Institución, se observe lo relativo al secreto bancario y fiduciario.	Garantizar la confidencialidad de la información
166. Desarrollar funciones de Contraloría Interna para propiciar el correcto funcionamiento de los sistemas de procesamiento de información conforme a las políticas de seguridad, así como la elaboración de información completa, correcta, precisa, íntegra, confiable y oportuna, incluyendo aquélla que deba proporcionarse a las autoridades competentes, y que coadyuve a la adecuada toma de decisiones. Perseverar la seguridad de la información generada, recibida, transmitida, procesada o almacenada en los sistemas informáticos y de telecomunicaciones de las instituciones de crédito, así como la aplicación de las medidas preventivas y correctivas necesarias para subsanar cualquier deficiencia detectada en materia de seguridad informática.	Contar con políticas de seguridad Garantizar disponibilidad de la información Garantizar la integridad de la información

259. Las Instituciones podrán estipular el uso de medios electrónicos, de cómputo o de telecomunicaciones para transmitir a las casas de bolsa las Órdenes. Asimismo, podrán implementar mecanismos que les permitan interconectarse al sistema de recepción y asignación de la casa de bolsa que ejecute las Órdenes respectivas en Bolsa, debiendo, en todo caso, establecer los controles necesarios para impedir que la identidad de los clientes sea del conocimiento de terceros en detrimento del secreto bancario o fiduciario.	Integración segura con otros sistemas
299. Asegurar la inalterabilidad de los datos, cifras y, en su caso, literalidad de los libros, registros y documentos originales objeto de Microfilmación o Grabación	Garantizar la integridad de la información
302. Las instituciones deberán contar con el conjunto de programas (hardware y software), procedimientos y datos del sistema que permitan conocer el contenido de los discos ópticos o magnéticos que contengan libros, registros y documentos contables y, en su caso, los que se requieran en tratándose de procesos de Microfilmación.	Garantizar disponibilidad de la información Contar con una infraestructura de TI adecuada a las funciones
304. Contar con políticas internas que tengan por objeto establecer lineamientos y procedimientos relativos al manejo y destrucción de libros. registros, documentos y demás información relativa a su contabilidad, que sean objeto de Microfilmación o Grabación.	Contar con políticas para la destrucción de información
I. Garantizar el adecuado manejo y control de los documentos con información confidencial de los clientes, a fin de asegurar que accedan a ella las personas que por sus funciones deban conocerla.	Controles para actualizar y acceder a la información Autenticación del personal facultado para realizar operaciones
II. Cumplir, con las disposiciones aplicables en materia de secreto bancario y fiduciario con respecto a la información de los clientes, estableciendo controles estrictos para evitar la sustracción de información.	Garantizar la confidencialidad de la información
III. Evitar proporcionar a terceras personas, información que las Instituciones obtengan con motivo de la celebración de operaciones con sus clientes.	Garantizar la confidencialidad de la información
307. Para la confirmación de la contratación de un servicio adicional de Banca Electrónica, las Instituciones deberán requerir a los Usuarios que ingresen un Factor de Autenticación. Las instituciones podrán permitir la contratación del servicio, mediante firmas electrónicas avanzadas o fiables de sus clientes.	Manejo de contratos de servicios con el usuario Autenticación de los usuarios
308. Las Instituciones, para permitir el inicio de una Sesión, deberán solicitar y validar al menos: el Identificador de Usuario y un Factor de Autenticación. Donde el Identificador de Usuario deberá ser único para cada Usuario y permitirá a la Institución identificar todas las operaciones realizadas por el propio Usuario a través del servicio. La longitud del Identificador de Usuario deberá ser de al menos seis caracteres.	Autenticación de los usuarios

309. Las Instituciones, en el uso del Identificador de Usuario y los Factores de Autenticación, deberán: impedir la lectura en la pantalla del Dispositivo de Acceso, la información de identificación y Autenticación proporcionada por el Usuario; asegurar que en la generación, entrega, almacenamiento, desbloqueo y restablecimiento de los Factores de Autenticación, únicamente sea el Usuario quien los reciba, active, conozca, desbloquee y restablezca;	Autenticación de los usuarios Administración de contraseñas
310. Las Instituciones deberán utilizar Factores de Autenticación para verificar la identidad de sus Usuarios y la facultad de estos para realizar operaciones a través del servicio de Banca Electrónica.	Autenticación de los usuarios Controles para actualizar y acceder a la información
311. Las Instituciones deberán establecer mecanismos y procedimientos para que sus Usuarios del servicio de Banca por Internet, puedan autenticar a las propias Instituciones al inicio de una Sesión.	Autenticación de la propia institución
313. Las Instituciones deberán solicitar a sus Usuarios, para la celebración de operaciones o prestación de servicios a través de Medios Electrónicos, un segundo Factor de Autenticación de las Categorías 3 (Información contenida o generada por medios o dispositivos electrónicos, así como la obtenida por dispositivos generadores de Contraseñas dinámicas de un solo uso) ó 4 (información del Usuario derivada de sus propias características físicas)	Autenticación de los usuarios
316 Bis 2. Las Instituciones deberán proveer lo necesario para que una vez autenticado el Usuario en el servicio de Banca Electrónica, la Sesión no pueda ser utilizada por un tercero, para lo cual deberá establecer al menos, los mecanismos siguientes: dar por terminada la Sesión en forma automática, e informar al Usuario el motivo cuando exista inactividad por más de veinte minutos, o cuando en el curso de una Sesión, se identifique cambios relevantes en los parámetros de comunicación del Medio Electrónico; Impedir el acceso en forma simultánea, mediante la utilización de un mismo Identificador de Usuario a más de una Sesión, informando al Usuario. Cuando el usuario ingrese a servicios de terceros mediante un enlace, informar esta situación y cerrar automáticamente la Sesión abierta con la Institución.	Manejo de sesiones
316 Bis 3. Establecer procesos y mecanismos automáticos para Bloquear el uso de Contraseñas y otros Factores de Autenticación para el servicio de Banca Electrónica, cuando se intente ingresar al servicio de Banca Electrónica utilizando información de Autenticación incorrecta, o cuando el Usuario se abstenga de realizar operaciones o acceder a su cuenta, por un periodo que determine cada Institución en sus políticas de operación. Se podrán Desbloquear el uso de Factores de Autenticación en base a preguntas secretas, cuyas respuestas deben conservarse almacenadas en forma Cifrada.	Autenticación de los usuarios Administración de contraseñas

316 Bis 4. Para el manejo de Contraseñas y otros Factores de Autenticación se deberán mantener procedimientos que proporcionen seguridad en la información contenida en los dispositivos de Autenticación, la distribución, así como en la asignación y reposición a sus Usuarios de los mismos. Así como está prohibido contar con mecanismos, algoritmos o procedimientos que les permitan conocer, recuperar o descifrar los valores de cualquier información relativa a la Autenticación de sus Usuarios.	Administración de contraseñas Garantizar la confidencialidad de la información
316 Bis 10. Las Instituciones que utilicen Medios Electrónicos para la celebración de operaciones y prestación de servicios, deberán implementar medidas o mecanismos de seguridad en la transmisión, almacenamiento y procesamiento de la información, a fin de evitar que sea conocida por terceros. Para los cual se deberán utilizar tecnologías que manejen Cifrado y que requieran el uso de llaves criptográficas.	Garantizar la confidencialidad de la información Garantizar la confidencialidad de la información transmitida
316 Bis 11. Las Instituciones deberán contar con controles para el acceso a las bases de datos y archivos correspondientes a las operaciones y servicios efectuados a través de Medios Electrónicos.	Controles para actualizar y acceder a la información
316 Bis 14. Las Instituciones deberán mantener en bases de datos las incidencias, fallas o vulnerabilidades detectadas en los servicios de Banca Electrónica, así como todas las operaciones efectuadas a través de éste.	Constancia electrónica auditable de las incidencias Constancia electrónica auditable de las operaciones
316 Bis 17. Las Instituciones estarán obligadas a realizar revisiones de seguridad, enfocadas a verificar la suficiencia en los controles aplicables a la infraestructura de cómputo y telecomunicaciones utilizada para la realización de operaciones y prestación de servicios a través de Medios Electrónicos.	Pruebas o evaluación de los controles de seguridad
316 Bis 18. Las Instituciones estarán obligadas a contar con áreas de soporte técnico y operacional, integradas por personal capacitado, las cuales se encargarán de atender y dar seguimiento a las incidencias que tengan sus Usuarios del servicio de Banca Electrónica, así como a eventos de seguridad relacionados con el uso de Medios Electrónicos.	Garantizar disponibilidad de los servicios
316 Bis 19. Las Instituciones deberán procurar la operación continua de la infraestructura de cómputo y de telecomunicaciones, así como dar pronta solución, para restaurar el servicio de Banca Electrónica, en caso de presentarse algún incidente.	Garantizar disponibilidad de los servicios
316 Bis 20. La Dirección General deberá asegurar que la Institución cuente con medidas preventivas, de detección, disuasivas y procedimientos de respuesta a incidentes de seguridad, controles y medidas de seguridad informática para mitigar amenazas y vulnerabilidades relacionadas con los servicios proporcionados a través de Banca Electrónica, que puedan afectar a sus Usuarios o a la operación de la Institución	Contar con políticas de seguridad Monitoreo y control de riesgos Administración de planes de contingencia Garantizar disponibilidad de los servicios

317. Las Instituciones podrán contratar con terceros, incluyendo a otras Instituciones o entidades financieras nacionales o extranjeras, la prestación de servicios necesarios para su operación. Donde las Instituciones deberán cuidar en todo momento, que las personas que les proporcionen los servicios, guarden la debida confidencialidad de la información.	Garantizar la confidencialidad de la información por parte de terceros
328. Verificar que los terceros o comisionistas con los que se contrate residan en países cuyo derecho interno proporcione protección a los datos de las personas, resguardando su debida confidencialidad, o bien, los países de residencia mantengan suscritos con México acuerdos internacionales en dicha materia o de intercambio de información entre los organismos supervisores, tratándose de entidades financieras.	Garantizar la confidencialidad de la información por parte de terceros
Artículo 334. Las políticas relativas a la contratación de servicios o comisiones, contemplarán como medidas de evaluación:	Manejo de contratos de servicios con terceros
I. La capacidad de los terceros para implementar medidas o planes que permitan mantener la continuidad del servicio con niveles adecuados de desempeño, confiabilidad, capacidad y seguridad.	Garantizar disponibilidad de servicios por parte de terceros
II. La integridad, precisión, seguridad, confidencialidad, resguardo, oportunidad y confiabilidad en el manejo de la información generada con motivo de la prestación de los servicios o comisiones, así como el acceso a dicha información, a fin de que sólo puedan tener acceso a ella, las personas que deban conocerla.	Garantizar la confidencialidad de la información por parte de terceros Garantizar disponibilidad de información por parte de terceros Garantizar la integridad de la información por parte de terceros
VII. La capacidad de las Instituciones, en la Administración Integral de Riesgos para identificar, medir, vigilar, limitar, controlar, informar y revelar los riesgos que puedan derivarse de la prestación de los servicios	Administración de riesgos Monitoreo y control de riesgos Garantizar disponibilidad de servicios por parte de terceros
338. Las Instituciones deberán establecer e implementar en todas sus Oficinas Bancarias, las medidas indispensables de seguridad y protección siguientes: seguridad física de las instalaciones donde se encuentran los equipos de cómputo y telecomunicaciones; Políticas y procedimientos para la protección, confidencialidad y adecuado funcionamiento de: redes de datos, aplicaciones, telecomunicaciones, procedimiento de datos, sistemas, programas y medios automatizados y e Información confidencial.	Contar con una infraestructura de seguridad

Tabla 2.4 –Requerimientos de seguridad en la Circular única bancaria

2.1.2 Legislación internacional

Como se ha visto en el apartado anterior, las regulaciones existentes en cuanto a seguridad en las aplicaciones financieras, son pocas y en algunos casos bastante generales. Es por esto que, para tener un enfoque más amplio de los aspectos de seguridad a salvaguardar en las aplicaciones financieras, se amplió el campo de investigación hacia la normatividad extranjera.

Esto adicionalmente se justifica, dentro de la ley de instituciones de crédito en el apartado "Exposición de Motivos" (decreto del 23 de julio de 1993) [23], donde se expresa:

"Buscando una mayor competitividad de los intermediarios y seguridad para los usuarios de servicios financieros, se hace expresa la obligación de las instituciones de proveer lo necesario para que sus filiales en el exterior se sujeten a la legislación extranjera que les sea aplicable y, en su caso a las disposiciones emitidas por las autoridades mexicanas."

Así tenemos que, las aplicaciones desarrolladas por las instituciones financieras pueden llegar a coexistir con sistemas informáticos de otras naciones; por lo que a continuación mostraremos un breve resumen de las leyes internacionales que por su alcance, se consideraron más relevantes y con mayor impacto en nuestro país [27].

2.1.2.1 Sarbanes Oxley Act

Creada en el 2002 por el senador demócrata Paul Spyros Sarbanes y el congresista Michael G. Oxley de los Estados Unidos de América, la ley Sarbanes Oxley Act (SOX) se creó con el fin de restablecer la confianza en los mercados de valores y en los reportes de información financiera. Para ello, estableció la implementación de estándares más elevados a aplicar para el control de fraudes, en la presentación de información financiera en los mercados, por parte de las instituciones que cotizan en la bolsa, y están enlistadas en la SEC (Securities and Exchanges Commission). Más específicamente, su misión es [28]:

"Proteger a los inversionistas al incrementar la exactitud y confiabilidad de la información divulgada por las instituciones de conformidad a las leyes de valores.

Este nuevo marco de regulación generó un fuerte impacto tanto en los auditores como en las gerencias de las compañías, puesto que establece la "responsabilidad personal" del director general y director financiero sobre la "exactitud" de los estados financieros y los "controles internos adecuados" en la información financiera. [29]

Con referencia al término "controles internos", éste involucra una serie de procesos que las compañías deben incorporar para la generación de reportes financieros y la protección de la

información financiera que conlleva a su generación. Esta información debe ser protegida y almacenada en varias localizaciones a través de la empresa (incluyendo aplicaciones empresariales, bases de datos e inclusive las hojas de cálculo contables).

Desde una perspectiva de tecnologías de información, la ley no contiene explícitamente ningún proceso prescriptivo a seguir. Además no expresa qué significa un "control interno adecuado" o qué soluciones pueden ser implementadas con el fin de crearlos. Sin embargo, una rápida revisión a la legislación revela los siguientes requerimientos de seguridad para el control interno:

Artículo	Requerimiento de seguridad
302. a. 4. Los directivos son responsables de establecer y mantener controles internos. Estos controles, deben garantizar que la información sea siempre conocida por los ejecutivos.	Involucrar a la alta dirección Garantizar disponibilidad de la información Controles para actualizar y acceder a la información
302. a. 5. Los directivos deben dar a conocer a los auditores, todas las deficiencias significativas en el diseño y operación de controles internos, los cuales pudieran repercutir adversamente en la habilidad de registrar, procesar, analizar y reportar los datos financieros. Así como también, cualquier tipo de fraude, material o no, que involucre a la dirección u otros empleados, quienes tienen un rol significativo en los controles internos de la institución.	Realización de auditorías Constancia electrónica auditable de las incidencias Detección y notificación de delitos o uso indebido de la información Autenticación del personal facultado para realizar operaciones
404. a. Las instituciones deberán entregar un reporte anual de controles internos a la Comisión, el cual estipule la responsabilidad de los directivos para establecer y mantener una estructura de controles internos adecuados, así como los procedimientos para generar reportes financieros. Este, deberá incluir una valoración de la efectividad de los mismos.	Involucrar a la alta dirección Garantizar disponibilidad de la información Contar con políticas de seguridad Pruebas o evaluación de los controles de seguridad
406. b. La Comisión deberá revisar que las Instituciones mantengan reglas, en las que se estipule que, sí se realizan cambios al código de ética, estas deben inmediatamente divulgarse a través de Internet o por cualquier otro medio electrónico.	Garantizar tiempo de respuesta en la divulgación de información
403. SEc16. a. 4. Si hay cambios en los accionistas o los altos directivos de la institución, sus datos deben ser públicamente accesibles a través del sitio de Internet y del sitio web corporativo, y se deberán ver reflejados al siguiente día hábil en el que entraron en vigor.	Garantizar tiempo de respuesta en la divulgación de información
409. Divulgar en tiempo real cambios en las condiciones financieras u operaciones de la institución, de cualquier información que es necesaria y útil para los inversionistas y es de interés público.	Garantizar tiempo de respuesta en la divulgación de información
501. 15D. a. 3. Para mejorar la objetividad de las investigaciones de valores, se deberán establecer medidas preventivas estructurales e institucionales, que aseguren que los analistas de valores no accedan a información que podría poner en entre dicho su supervisión.	Medidas para prevenir accesos no autorizados

802. 1519. Cualquiera que altere, destruya, mutile, oculte, encubra, falsifique o realice una entrada falsa a un registro, documento u objeto tangible con la intención de impedir, obstruir o influenciar una investigación federal o la administración de la institución, será condenado por la ley.	Detección y notificación de delitos o uso indebido de la información
802. 1520. Se deberán conservar los registros de auditoría por un periodo mínimo de 5 años. Estos registros abarcan: documentos, memorandos, correspondencia, comunicaciones y registros, incluidos los electrónicos.	Constancia electrónica auditable de las operaciones Conservar la información Manejo de respaldos de la información

Tabla 2.5 –Requerimientos de seguridad en Sarbanes Oxley Act

2.1.2.2 Gramm – Leach – Bliley Act

El Gramm – Leach – Bliley Act (GBL), previamente conocido como Financial Services Modernization Act, es una ley federal de los Estados Unidos aprobada en 1999. Se instituyó para revocar las restricciones existentes de los bancos afiliados con firmas seguras, y para obligar a las instituciones financieras a que adopten medidas estrictas de privacidad relacionadas con los datos del cliente. La ley se aplica a cualquier organización que trabaje con: empresas que remitan impuestos sobre la renta, agencias que reportan crédito de los consumidores, los servicios de establecimiento de transacciones en tiempo real, agencias de colección de débito y las empresas que reciban información confidencial de las instituciones financieras. Más específicamente, su objetivo es [30]:

"Incrementar la competencia en la industria de servicios financieros al proveer un marco de trabajo prudencial para la afiliación de bancos, firmas de seguridad, compañías de seguro, y otros proveedores de servicios financieros."

El conjunto de requerimientos de seguridad observados en la ley, puede conllevar a la implementación de diversos controles en las tecnologías de información de la institución. Sin embargo, la mayoría de estos requerimientos pueden ser satisfechos gracias a una política de seguridad robusta que se haga cumplir en toda la empresa. Más específicamente, los artículos que nos ofrecieron estas pautas son:

Artículo	Requerimiento de seguridad
103. 3. A. ii. Se considera que una institución realiza actividades financieras, si ofrece servicios e información que es financiera por naturaleza a través de medios tecnológicos, incluyendo cualquier aplicación necesaria para proteger la seguridad o eficacia de los sistemas para la transmisión de datos o transacciones financieras.	Contar con una infraestructura de TI adecuada a las funciones Garantizar la confidencialidad de la información transmitida

111. 1. A. i. El Consejo puede requerirle a la institución, sus condiciones financieras, sistemas para monitoreo, control financiero y de riesgos operativos, y transacciones con las subsidiarias.	Constancia electrónica auditable de las operaciones Monitoreo y control de riesgos
111. 2. A. ii. La junta podrá realizar auditorías sobre los riesgos financieros y operacionales de los sistemas que maneja la institución, y los cuales pueden ser una amenaza para la seguridad y validez de cualquier repositorio institucional. Así como también, de los sistemas utilizados para el monitoreo y control de tales riesgos.	Realización de auditorías Administración de riesgos Monitoreo y control de riesgos
231. 3. A. i. I. Las instituciones deben mantener registros y reportes de las condiciones financieras, políticas, sistemas para el monitoreo, control financiero y riesgos operacionales, y las transacciones entre sus afiliados.	Constancia electrónica auditable de las operaciones Monitoreo y control de riesgos
501. b. Las instituciones financieras tienen la obligación de respetar la privacidad de sus clientes, así como proteger la seguridad y confidencialidad de la información no pública de los mismos. Para ello deberán establecer estándares administrativos, técnicos y físicos para: asegurar la seguridad y confidencialidad de los registros e información de los clientes; proteger contra cualquier amenaza y peligro no anticipado la seguridad e integridad de los registros; proteger contra accesos no autorizados los cuales puedan resultar en un daño sustancial al cliente.	Contar con políticas de privacidad Contar con políticas de seguridad Contar con una infraestructura de seguridad Medidas para prevenir accesos no autorizados
502. a. Las instituciones financieras no deben otorgar información personal a terceros, a excepción de que el cliente lo haya consentido de manera escrita o electrónica.	Garantizar la confidencialidad de la información por parte de terceros Manejo de contratos de servicios con el usuario
503. b. Cuando se establezca una relación con el cliente, la institución financiera debe indicar las políticas que la institución maneja para proteger la confidencialidad y seguridad de la información personal.	Contar con políticas de privacidad Contar con políticas de seguridad
508. a. La Secretaría del Tesoro debe de conducir un estudio de las prácticas para compartir información entre instituciones financieras y sus afiliados. El cual debe incluir: El propósito de compartir información, extensión y adecuación de su seguridad; los riesgos potenciales sobre la privacidad del cliente, la adecuación de las políticas de privacidad de las instituciones.	Integración segura con otros sistemas Garantizar la confidencialidad de la información transmitida Contar con políticas de privacidad Contar con políticas de seguridad
521. d. Es correcto obtener información de los clientes, con fin de realizar pruebas sobre los procedimientos de seguridad y de los sistemas para mantener la confidencialidad del cliente. Así como para recuperar la misma de otras personas que la hayan obtenido por métodos fraudulentos.	Recuperación de información

Tabla 2.6 –Requerimientos de seguridad en Gramm – Leach – Bliley Act

2.1.2.3 Basel III

El Comité de Supervisión Bancaria de Basilea, creado en 1975 por los Gobernadores de los bancos centrales del Grupo de los Diez, publicó recientemente "Basilea III"; el cual es un conjunto integral de reformas elaborado para fortalecer la regulación, supervisión y gestión de riesgos del sector bancario. De manera general, estas medidas persiguen [31]:

"La optimización de la capacidad del sector bancario para afrontar perturbaciones ocasionadas por tensiones financieras o económicas, mejorar la gestión de riesgos y el buen gobierno en los bancos, así como reforzar la transparencia y la divulgación de información de los mismos."

Más específicamente, las reformas que introdujo Basilea en su tercera versión, son:

- Dimensión microprudencial, la cual permite aumentar la capacidad de reacción de cada institución en periodos de tensión.
- Dimensión macroprudencial, son los riesgos sistémicos que pueden acumularse en todo el sector bancario, así como la amplificación procíclica de éstos a lo largo del tiempo.

Estas dos dimensiones son complementarias entre sí, ya que aumentando la resistencia de cada banco se reduce el riesgo de alteraciones en todo el sistema. Por otro lado, la ley sigue dando una gran importancia a la Administración de Riesgos misma que podemos notar a lo largo de la diversa documentación que compone a esta ley. De esta manera, los artículos más relevantes identificados en cuanto a requerimientos de seguridad, se muestran en la siguiente tabla.

Artículo	Requerimiento de seguridad
Basilea III: Marco regulador global para reforzar los bancos y sistemas bancarios [32]	
5. 117. 42. El banco someterá regularmente su sistema de medición de riesgos a un examen independiente en el marco de sus procesos de auditoría interna. La revisión del proceso general de gestión de riesgos se realizará periódicamente y abarcará: la idoneidad de la documentación del sistema y del proceso de gestión de riesgos; la validación de cualquier modificación significativa del proceso de medición de riesgos; la exactitud y exhaustividad de los datos, entre otros.	Administración de riesgos Realización de auditorías Documentación operativa de procesos y herramientas Verificación de que los sistemas cumplan con los requerimientos
II. A. 3. 106. 51(i). Los bancos que apliquen el método de modelos internos deberán contar con una unidad de gestión de garantías responsable de controlar la integridad de los datos utilizados en peticiones de reposición de márgenes, y garantizar su coherencia y reconciliación frecuente con todas las fuentes relevantes de datos del banco. La alta dirección deberá asignar recursos suficientes a esta unidad para que el rendimiento operativo de sus sistemas sea el adecuado, medido por la puntualidad y exactitud de las peticiones salientes y por el tiempo de respuesta ante peticiones entrantes.	Involucrar a la alta dirección Garantizar la integridad de la información Contar con una infraestructura de TI adecuada a las funciones Garantizar disponibilidad de los servicios

Convergencia internacional de medidas y normas de capital [33]	
III. C. 1. iii. 275. Dentro del método para calcular los requerimientos de capital en concepto de pérdidas inesperadas, y obtener las ponderaciones por riesgo, se deberán tomar en cuenta dentro de los criterios de asignación a: (Dado **Anexo 6**) El riesgo de diseño y tecnológico dentro de las características de operación.	Administración de riesgos Clasificar tipos de riesgos
V. C. 2. iii. 669. Dentro de los criterios cuantitativos a tomar en cuenta para la estimación del riesgo operacional, se deben tomar en cuenta los eventos de pérdida definidos en el **Anexo 9**. Donde, como fraude externo aparece la seguridad de la información (Daños por ataques informáticos y robo de información). Para Incidencias en el negocio y fallos en los sistemas, se debe tomar en cuenta al hardware, software y telecomunicaciones.	Administración de riesgos Clasificar tipos de riesgos
III. C. iii. 671. Los datos internos de pérdida son de máxima relevancia cuando guarden relación con las distintas actividades del negocio, procesos tecnológicos y procedimientos de gestión del riesgo del banco. Por ello, el banco deberá haber documentado los procedimientos para evaluar en todo momento la relevancia de los datos históricos de pérdida, considerando situaciones en que se utilicen excepciones.	Administración de planes de contingencia
VI. A. 2. ii. Cuando la institución desarrolle su propio modelo, éste deberá basarse en supuestos adecuados, que habrán de ser evaluados y probados por personal cualificado que no participe en el proceso de desarrollo. Deberán validarse las fórmulas matemáticas, los supuestos utilizados y la aplicación del software.	Pruebas al sistema por personal diferente al de desarrollo Verificar que los sistemas cumplan con los requerimientos
Principios Básicos para una supervisión bancaria eficaz [34]	
Principio 2. 4. e. El supervisor dispone de recursos adecuados para desempeñar eficazmente su tarea de supervisión y vigilancia, como un presupuesto suficiente en materia de tecnología que permita dotar al personal de las herramientas necesarias para el análisis del sector bancario y la evaluación de los bancos y grupos bancarios	Contar con un presupuesto eficiente de TI Contar con una infraestructura de TI adecuada a las funciones
Principio 15. 7. El supervisor verifica que el banco cuenta con sistemas de información adecuados (tanto en circunstancias normales como en periodos de tensión) para cuantificar, evaluar y notificar el volumen, composición y calidad de las exposiciones del conjunto de la entidad y para todo tipo de riesgos, productos y contrapartes. El supervisor también verifica que estas informaciones reflejan el perfil de riesgo y las necesidades de capital y liquidez del banco y que oportunamente se presentan ante el Consejo y la alta dirección de la entidad en un formato adecuado a su uso.	Contar con una infraestructura de TI adecuada a las funciones Clasificar tipos de riesgos Garantizar disponibilidad de la información

Principio 25. 5. El supervisor verifica que los bancos cuentan con adecuadas políticas y procesos en materia de tecnología informática para identificar, evaluar, vigilar y gestionar los riesgos tecnológicos. El supervisor también verifica que el banco dispone de una sólida y adecuada infraestructura tecnológica para cubrir las necesidades actuales y previstas del negocio (en circunstancias normales y en periodos de tensión), que garantice la integridad, seguridad y disponibilidad de datos y sistemas, así como facilite una gestión integral y exhaustiva del riesgo.	Administración de riesgos Monitoreo y control de riesgos Contar con una infraestructura de seguridad
Principio 25. 6. El supervisor verifica que los bancos cuentan con sistemas de información adecuados y eficaces para: vigilar el riesgo operacional; recopilar y analizar datos sobre el riesgo operacional; y promover la existencia de adecuados mecanismos de notificación a los Consejos, la alta dirección y las líneas de negocio del banco que faciliten una gestión proactiva del riesgo operacional.	Contar con una infraestructura de TI adecuada a las funciones Contar con una infraestructura de seguridad Monitoreo y control de riesgos Garantizar disponibilidad de la información
Principio 26. 1. d. La legislación, la regulación, o bien el supervisor, exigen que los bancos dispongan de adecuados controles internos con el fin de crear un entorno operativo correctamente controlado para la gestión del negocio, teniendo en cuenta su perfil de riesgo. Estos controles deben considerar la salvaguardia de activos e inversiones: incluido el control físico y el acceso a sistemas informanticos.	Contar con políticas de seguridad Contar con una infraestructura de seguridad Clasificar tipos de riesgos Monitoreo y control de riesgos Controles para actualizar y acceder a la información
Principio 26. 4. El supervisor verifica que los bancos cuentan con una función de auditoría interna independiente, permanente y eficaz, encargada de: evaluar si las políticas, procesos y controles internos existentes (incluidos los procesos de gestión del riesgo, cumplimiento y gobierno corporativo) son eficaces, adecuados y continúan siendo suficientes para la actividad del banco; y garantizar el cumplimiento de las políticas y procesos.	Realización de auditorías Contar con políticas de seguridad
Principio 29. 6. El supervisor verifica que los bancos cuentan con suficientes controles y sistemas para prevenir, identificar y denunciar la utilización abusiva de servicios financieros, incluidos el blanqueo de capitales y la financiación del terrorismo.	Detección y notificación de delitos o uso indebido de la información
Principio 29. 9. El supervisor verifica que los bancos cuentan con políticas y procesos claros, y los respetan, para que el personal notifique cualquier problema relacionado con la utilización abusiva de los servicios financieros. El supervisor también verifica que los bancos poseen y utilizan apropiados sistemas de información para notificar correcta y puntualmente dichas actividades a la dirección.	Detección y notificación de delitos o uso indebido de la información

Tabla 2.7 –Requerimientos de seguridad en Basilea III

2.1.3 Requerimientos de seguridad identificados dentro de la legislación

Haciendo una lectura de la normatividad anteriormente expuesta, podemos concretar que los requerimientos y controles de seguridad identificados, presentan un carácter demasiado general para ser implementados inmediatamente por los desarrolladores de sistemas.

Esto se debe, a que estas leyes en su mayoría, se encuentran enfocadas a la elaboración de un análisis de riesgos, así como al establecimiento de una serie de políticas y controles internos a nivel de negocio, los cuales garanticen el resguardo, confidencialidad e integridad de la información manejada por las instituciones. Donde después estas políticas, deberán permearse hacia los sistemas informáticos en la forma de controles computacionales.

Al realizar el presente análisis, tratando de empatar las especificaciones de seguridad manifestadas por las leyes, se obtuvo una lista de requerimientos de seguridad comunes; los cuales, de ser implementados dentro de las aplicaciones, pueden ayudarnos a cumplir con dicha normatividad.

En este sentido, a continuación se exponen los requerimientos identificados durante este estudio. Posteriormente, este análisis nos servirá en el siguiente capítulo, para identificar los controles de seguridad computacionales a implementar dentro de las aplicaciones financieras.

Requerimieutos de seguridad	Inst. Crédito	Transp. Servicios	Protect. Usuarios	Circular CNBV	SOX	GBL	Basel
Administración de certificados		X					
Administración de contraseñas				X			
Administración de planes de contingencia				X			X
Administración de riesgos	X			X		X	X
Asignación de roles y responsabilidades				X			
Autenticación de la propia institución				X			
Autenticación de los usuarios		X		X			
Autenticación del personal facultado para realizar operaciones		X		X	X		
Clasificar información	X			X			
Clasificar tipos de riesgos				X			X
Conservar la información	X			X	X		
Constancia electrónica auditable de las incidencias				X	X		
Constancia electrónica auditable de las operaciones	X		X	X	X	X	
Contar con políticas de privacidad			X			X	
Contar con políticas de seguridad	X			X	X	X	X
Contar con políticas para la destrucción de información				X			
Contar con un presupuesto eficiente de TI							X
Contar con una infraestructura de seguridad	X			X		X	X
Contar con una infraestructura de TI adecuada a las funciones	X			X		X	X
Controles para actualizar y acceder a la información	X			X			X
Detección de faltantes	X			X			
Detección y notificación de delitos o uso indebido de la información	X			X	X		X
Documentación de requerimientos				X			

Documentación operativa de procesos y herramientas	X						X
Documentación técnica del sistema				X			
Garantizar disponibilidad de la información		X		X	X		X
Garantizar disponibilidad de la información por parte de terceros				X			
Garantizar disponibilidad de los servicios	X			X			X
Garantizar disponibilidad de los servicios por parte de terceros	X			X			
Garantizar el no repudio	X	X					
Garantizar la confidencialidad de la información	X			X			
Garantizar la confidencialidad de la información por parte de terceros	X			X		X	
Garantizar la confidencialidad de la información transmitida		X		X		X	
Garantizar la integridad de la información	X			X			X
Garantizar la integridad de la información por parte de terceros				X			
Garantizar tiempo de respuesta en la divulgación de información		X		X	X		
Independencia operativa con otros sistemas	X						
Integración segura con otros sistemas				X		X	
Involucrar a la alta dirección				X	X		X
Manejo de contratos de servicios con el usuario	X	X		X		X	
Manejo de contratos de servicios con terceros				X			
Manejo de niveles de autorización y acceso al sistema				X			
Manejo de respaldos de la información	X			X	X		
Manejo de sesiones				X			

Medidas para prevenir accesos no autorizados					X	X	
Monitoreo y control de riesgos				X		X	X
Pruebas al sistema por personal diferente al de desarrollo							X
Pruebas o evaluación de los controles de seguridad				X	X		
Pruebas para la evaluación de vulnerabilidades				X			
Realización de auditorias	X			X	X	X	X
Recuperación de información				X		X	
Restablecimiento de operaciones				X			
Uso de licencias				X			
Verificar que los sistemas cumplan con los requerimientos				X			X

Tabla 2.8 –Requerimientos de seguridad dentro de la legislación

2.2 Estándares y guías para el desarrollo seguro

Como habíamos mencionado en la introducción del presente capítulo, el segundo recurso en el que se pueden auxiliar las instituciones financieras, para desarrollar o revisar sus controles de seguridad, son los estándares en materia de seguridad de la información publicados por diversas organizaciones nacionales e internacionales.

Recordando los conceptos generales de seguridad, sus principales metas se enfocan hacia la preservación de la **confidencialidad**, **integridad**, y **disponibilidad**, de los activos y recursos de información que las aplicaciones crean, almacenan, procesan o transmiten durante su ejecución. Básicamente, la preservación de la confidencialidad se refiere a evitar la revelación de la información a entidades no autorizadas; la integridad por su parte trata de prevenir alteraciones no autorizadas a la información, y la disponibilidad consiste en evitar la destrucción no autorizada de la información o bien la denegación de acceso al servicio.

Para poder incluir y mantener los anteriores atributos de seguridad dentro de las aplicaciones, los desarrolladores requieren realizar un esfuerzo mayor durante el desarrollo de las mismas, lo cual implica el uso de metodologías y buenas prácticas de seguridad.

De esta manera, el problema de producir software seguro se convierte en un tema multifacético en donde la ingeniería de software, la ingeniería de seguridad y la administración se combinan para lograr este fin [9]. Más específicamente, la ingeniería de software se enfoca a la planeación, control, administración de la calidad, su medición y las tareas de ingeniería dentro del proyecto. Mientras que la ingeniería de seguridad se centra en los métodos y herramientas necesarias para diseñar, implementar y probar la seguridad de los sistemas desarrollados. Finalmente la administración establece las políticas necesarias para el uso de mejores prácticas de seguridad dentro del desarrollo del software. (Véase Figura 2.1)

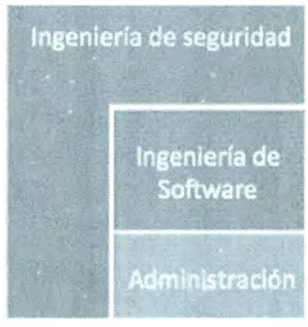


Figura 2.1 – Elementos que conforman la producción de software seguro

Concretando lo anterior, la creación de aplicaciones seguras comienza con el uso de **procesos de ingeniería de software** adecuados, aumentados con **prácticas técnicas de seguridad** y soportadas por **prácticas administrativas** que promueven el desarrollo seguro.

Con estos tres puntos en mente, diversas organizaciones nacionales e internacionales han creado una serie de estándares enfocados a proponer metodologías, guías y controles para el desarrollo de aplicaciones seguras. Debido a ello, a continuación hablaremos de aquellos estándares identificados durante la presente investigación, y los cuales preferentemente incorporan actividades de seguridad dentro del ciclo de vida de desarrollo de sistemas.[4]

2.2.1 Guías para el desarrollo de aplicaciones financieras

Dentro de este apartado se analizan aquellas guías que manifiestan el desarrollo de software seguro dentro de las instituciones financieras. Siendo el primero de ellos, la normatividad interna manejada por el Banco de México, y el segundo los manuales en tecnología promovidos por el Consejo Federal de Examinación de las Instituciones Financieras en los Estados Unidos (FFIEC).

2.2.1.1 Norma Administrativa Interna del Banco de México

Como vimos en la sección anterior, el Banco de México es uno de los principales organismos en promover regulaciones para las Instituciones Financieras en nuestro país. Su "Norma Administrativa Interna en Tecnologías de Información" [5] (NAI) [35], ofrece un ejemplo de cómo involucrar actividades de seguridad dentro del desarrollo de las aplicaciones.

Más específicamente, el Título IV de este documento, establece la normatividad aplicable al desarrollo de sistemas, donde se detalla que el "Líder Informático del Proyecto" debe aplicar un método de desarrollo de software capaz de cumplir con las siguientes actividades:

Etapas del SDLC	Actividades de seguridad
Requerimientos	Documentar los procesos de validación, control y seguimiento para conservar la calidad y seguridad de la información.
	Definir los roles de responsabilidad que tendrán los "Usuarios", a fin de considerar en el diseño los controles de acceso y uso de la información.
Diseño	Para facilitar la disponibilidad de la información, evitar inconsistencias y reducir duplicidades, los "Sistemas" deben usar los "Depósitos Institucionales".

[4] La redacción siguiente comprende un breve resumen de las actividades en cuanto a seguridad que proponen cada uno de los estándares. Para mayor información, favor de consultar directamente cada uno de los estándares.

[5] La información que a continuación se presentará, sobre la normativa interna del Banco de México, se considera de carácter "Confidencial".

	Los "Sistemas" deben contar con los mecanismos para que el "Responsable Operativo", el "Responsable de Soporte", la Contraloría y la Dirección de Auditoría, puedan verificar la información sin poner en riesgo su integridad.
	Incluir el almacenamiento de la información histórica relevante, la cual debe guardarse como parte integral del proceso.
	Integrar facilidades de control de acceso y confidencialidad que permitan el manejo de la información de acuerdo a la clasificación de los "Estándares Institucionales".
	Los "Sistemas" que generen y registren "Contraseñas de servicio" deben validarse conforme a las medidas de seguridad descritas en Apéndice I.1.A (incluye: longitud, caracteres, renovación)
	Contemplar "Procedimientos de Respaldo" y "Procedimientos de Recuperación".
	Descripción de los mecanismos de seguridad y de telecomunicaciones que tendrá el "Sistema".
	El "Grupo de Arquitectura de Sistemas" debe validar que el diseño del "Sistema" no comprometa el funcionamiento de la infraestructura tecnológica del "Banco".
	El "Líder Informático de Proyecto" debe verificar que el código descargado provenga de un sitio de confianza.
Implementación	Utilizar las Bibliotecas de Seguridad Institucionales para la autenticación y autorización de usuarios cuyas cuentas residan en los directorios de recursos de red del "Banco".
	Llevar a cabo pruebas de funcionalidad en operación normal y en situación contingente.
	Pruebas de impacto a la infraestructura de telecomunicaciones y de cómputo (volumen, concurrencia, etc.)
Pruebas	Pruebas de vulnerabilidad a ataques informáticos en coordinación con la Of. De Seguridad Informática.
	Las pruebas deben realizarse en un ambiente independiente del utilizado para el desarrollo y la producción
	Certificar la aplicación.
	Cuando un "Sistema" inicie su operación, deben deshabilitarse, en el ambiente de producción, las cuentas de operación del personal que desarrolló el "Sistema".
	Se debe tener respaldo de los programas fuente utilizados para la generación de la versión de
Implantación	producción del "Sistema" así como de la documentación.
	Los "Sistemas" o programas deben registrarse ante el Registro Público de Derechos de Autor, cuando se concluya que tal registro es conveniente con base en el "Análisis de Riesgos".

Tabla 2.9 – Actividades de seguridad en la NAI del Banco de México

Además de las actividades anteriores, se recomienda la lectura del Título III aplicable a la operación y soporte de sistemas, en donde se da un especial énfasis al "Análisis de Riesgos", la creación de un "plan de contingencia" y la documentación de "procedimientos de respaldo" y de "recuperación".

Otro documento que complementa a la NAI, son los "Lineamientos de Seguridad para Aplicaciones Web del Sitio de Banco de México" [36]. En éste, se establecen los controles de seguridad necesarios para proteger la información manejada por las aplicaciones ofrecidas en los sitios web del Banco de México.

Los tipos de control propuestos permiten cubrir las diferentes características de seguridad con que debe contar cualquier desarrollo de aplicaciones, conforme a la criticidad de la información manejada. En resumen, los controles que se propone son:

Propiedad de seguridad	Criticidad de la información: Pública	Reservada	Sensible
Autenticación	No requerido	**Un factor.** Uso de contraseñas para las aplicaciones que sólo permiten consultas, y están dirigidas a los empleados del banco. **Doble factor**. Empleo de *Token* para aplicaciones donde se pueda modificar información reservada.	Certificados
Control de acceso	No requerido	Control de privilegios por grupos	Control de privilegios por grupos. El acceso sólo podrá realizarse desde sitios confiables
Confidencialidad	No requerido	SSL	SSL
Integridad	Hash	Hash	Hash
Disponibilidad	Requerido	Requerido	Requerido
Validación de seguridad	Validación de la aplicación durante el diseño y pruebas	Validación de la aplicación durante el diseño, pruebas y producción	Validación de la aplicación durante el diseño, pruebas y producción

Tabla 2.10 – Controles de seguridad en los Lineamientos para aplicaciones del Banco de México

2.2.1.2 Manuales en Tecnologías de Información del FFIEC

El Consejo Federal de Examinación de las Instituciones Financieras (FFIEC, Federal Financial Institutions Examination Council) [37], es una agencia formal de los Estados Unidos encargada de prescribir principios uniformes, estándares y reportes para la examinación federal de las instituciones financieras. Recientemente, dicho organismo ha publicado diferentes manuales con referencia al uso de las tecnologías de la información, incluyendo al desarrollo de aplicaciones como uno de ellos.

El "Manual de Desarrollo y Adquisición" [38] que analizaremos a continuación, provee a los auditores y a las instituciones financieras de una guía para identificar y controlar los riesgos en el desarrollo y adquisición de los sistemas. Con referencia al desarrollo de sistemas, esta guía establece una serie de puntos a cubrir en cada una de las fases del ciclo de vida. En donde las actividades más próximas a la inclusión de seguridad dentro de estas etapas son:

Etapas del SDLC	Actividades de Seguridad
Iniciación	Dentro de los "Requerimientos Funcionales", considerar los controles internos y requerimientos de seguridad de la información. Así como los requerimientos de respaldo del sistema (tipo, capacidad, desempeño).
	Establecer la metodología de Administración de riesgos
Planeación	Establecer roles y responsabilidades. En donde se deberá definir las responsabilidades primarias del personal clave dentro del proyecto.
	Dentro del control de requerimientos, considerar el diseño y construcción de controles automatizados y características de seguridad dentro de las aplicaciones.

	Administrar riesgos, estableciendo procedimientos para evaluar, monitorear y administrar riesgos internos y externos, a través del ciclo de vida del proyecto.
	El plan del proyecto debe incluir estándares para: descuido, control del sistema y aseguramiento de la calidad. En descuido, incluir los procedimientos de administración de riesgos. En controles del sistema, incluir los requerimientos funcionales, de seguridad y controles automatizados.
Diseño	Realizar el diseño, en cuanto sea posible, de las características de seguridad, auditoría y controles automatizados de la aplicación.
	Diseñar controles de entrada de datos, como: dígitos de corroboración, chequeo de integridad, duplicación, límites, rangos, chequeo de sensatez, secuencia y validación.
	Diseñar controles de proceso, como: controles *batch*, reporte de errores, *logs* de transacciones, chequeos de secuencia, archivos provisionales y archivos de respaldo.
	Diseñar controles de salida de datos, como: *logs* de *batch*, controles de distribución y controles de destrucción.
Desarrollo	Establecer estándares de código, ya que estos permiten reducir los defectos de código y así incrementar la seguridad, confiabilidad y mantenimiento de las aplicaciones.
	Establecer controles en las librerías como: controles automáticos para contraseñas y aplicaciones para librerías automáticas.
	Establecer controles para el manejo de versiones, los cuales permiten identificar rápidamente los errores en la programación.
	Las organizaciones deben mantener documentación detallada de las aplicaciones. Dado que la documentación, permite a las organizaciones comprender más fácilmente, las características funcionales, de seguridad y los controles manejados; así como mejorar el uso y mantenimiento del software.
	Considerar controles de acceso y cambios para las actividades de documentación. Para asegurar que los individuos solo tengan acceso a las secciones de la documentación necesarias para su trabajo.
Pruebas	Hacer uso de planes de pruebas detallados, los cuales incrementan significativamente la probabilidad de identificar debilidades en las aplicaciones.
	Las pruebas funcionales deberán asegurar que la funcionalidad esperada, la seguridad y los controles internos, se encuentren presentes y operen apropiadamente.
Implementación	Se deberán introducir datos a la aplicación y verificar los mismos; así como configurar y probar los parámetros del sistema y de seguridad.
	Verificar la exactitud de los datos de entrada y las configuraciones de seguridad.
	Llevar a cabo revisiones al final del proyecto para validar el cumplimiento de los objetivos. Este puede incluir, cambios al sistema para corregir problemas e incrementar la seguridad.
Mantenimiento	Establecer estándares administrativos y procedimientos para asegurar que los cambios realizados no afectan a las operaciones o degradan el desempeño del sistema o la seguridad.
	Se deberán establecer controles que permitan realizar cambios de emergencia debido a cuestiones de seguridad o problemas en el procesamiento.
	Una parte crítica de la administración del proceso de versiones, involucre mantener una constante alerta sobre las vulnerabilidades externas y disponibilidad de generar una nueva versión.
	El aseguramiento de la calidad, la seguridad, las auditorias, el cumplimiento con las leyes, redes y los usuarios finales, deberán ser incluidos apropiadamente en el proceso de administración de cambios.
Retiro	Deberán realizarse revisiones sobre riesgos y seguridad, tanto si se realizan modificaciones al sistema, como si ese permanece sin cambios.
	Se deberán transferir datos de los sistemas de producción de una manera planeada y controlada que incluya procedimientos de respaldos y pruebas.

Tabla 2.11 – Actividades de seguridad en SDLC del manual de desarrollo del FFIEC

En adición al anterior escrito, el FFIEC elaboró el "Manual de Seguridad de la Información" [16], el cual define una serie de guías y controles de seguridad a implementar para establecer una cultura de seguridad de la información dentro de la organización, y que permite a las instituciones cumplir con el artículo 501(b) de la ley Gramm-Leach-Bliley.

Dentro de este documento, en la sección "Implementación de Controles de Seguridad", se ofrece un punto dedicado a la seguridad en el "Desarrollo de Sistemas, la Adquisición y su Mantenimiento". Dichos controles no se encuentran definidos dentro del contexto del ciclo de vida del sistema, como en el anterior manual, y a grandes rasgos promueven el uso de:

- Controles en los requerimientos de seguridad.
- Controles de seguridad en la aplicación del software.
- Verificar la fiabilidad en el software.
- Proceso de desarrollo seguro.
- Revisión del código fuente.

2.2.2 Estándares y guías generales

A continuación presentaremos aquellos estándares y guías para el desarrollo de software seguro aplicables a cualquier tipo de industria. Las cuales sin embargo, son usadas actualmente por diversas instituciones financieras para el cumplimiento de las políticas de seguridad dictaminadas por la legislación nacional e internacional.

2.2.2.1 NIST SP800-64

El Instituto Nacional de Estándares y Tecnología (NIST, National Institute of Standards and Technology), elaboró la guía denominada "Consideraciones de Seguridad en el Ciclo de Vida del Desarrollo de Sistemas de Información" [39]. Este documento, presenta un modelo de referencia que permite incorporar aspectos de seguridad en todas las fases del proceso de desarrollo del software, abarcando desde su inicio hasta el retiro del mismo.

En general, el SDLC manejado por el NIST se divide en cinco etapas; cada una de las cuales incluye un conjunto mínimo de actividades requeridas sobre seguridad, para garantizar que la seguridad se incorpore efectivamente en el sistema. (Véase Figura 2.2). De esta manera, una organización puede optar por utilizar el ciclo de vida propuesto por el NIST o bien, adaptar el propio a través de la implementación de las actividades de seguridad descritas en la norma.

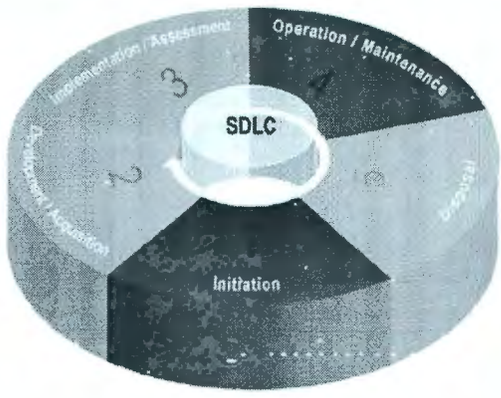


Figura 2.2 - SDLC del NIST

Conforme a lo anterior, las actividades de seguridad a aplicar dentro del ciclo de vida de desarrollo de software son las siguientes:

Etapas del SDLC	Actividades de Seguridad
Iniciación	**Planeación inicial de la seguridad**. Incluye preparaciones para todo el ciclo de vida, incluyendo la identificación de requerimientos de seguridad, roles, entregables, herramientas y tecnologías.
	Categorización de la seguridad. Analizar el nivel del impacto potencial de la información dentro de la organización y los clientes, el cual puede derivarse en perdida de confidencialidad, integridad o disponibilidad de la información.
	Evaluación del impacto en el negocio. Relacionar componentes específicos del sistema con los servicios críticos del negocio. Lo cual permite evaluar las consecuencias en el negocio, si un componente llegase a fallar.
	Evaluación del impacto en la privacidad. Considerar si el sistema transmitirá, almacenará o creará información considerada como privada.
	Usar un proceso seguro para el desarrollo del sistema. Deberá considerar un concepto de operaciones seguras para el desarrollo, estándares y procesos, capacitación en seguridad para el equipo de desarrollo, administración de la calidad, un ambiente seguro y prácticas seguras de codificación y su repositorio.
Desarrollo o Adquisición	**Valoración de riesgos.** Evalúa el conocimiento actual del diseño del sistema y los requerimientos de seguridad, para determinar su efectividad para mitigar riesgos anticipados.
	Seleccionar y documentar controles de seguridad. Consiste en: la selección de controles base de seguridad, adaptar estos a la aplicación e integrar controles adicionales basados en una evaluación de riesgos.
	Diseñar una arquitectura de seguridad. A nivel de sistema deberá tomar en consideración los servicios obtenidos externamente, conexiones entre sistemas y los roles de los usuarios en el sistema. Así como incluir registros de auditoría para los diversos componentes.
	Ingeniería en seguridad y controles de desarrollo. Los controles en seguridad son lógicamente planeados e implementados para formar parte del sistema. Con el objetivo conocer su impacto en el desempeño del sistema.
	Desarrollar documentación de seguridad. Crear el plan de seguridad del sistema y documentación adicional como: el plan de la administración de seguridad, plan de contingencia, plan de monitoreo continuo, plan de capacitación, plan de respuesta a incidentes y valoración del impacto de la privacidad.
	Conducir Pruebas. Los sistemas deben ser evaluados antes de ser implementados para validar que el desarrollo del sistema cumple con los requerimientos funcionales y de seguridad.
Implementación	**Crear un plan detallado para certificación y acreditación**. Deberá identificar los roles clave, limitaciones del proyecto, componentes principales, alcance de las pruebas y nivel de rigor esperado. Creándose un paquete de certificación.

	Integrando seguridad en ambientes y sistemas establecidos. El establecimiento de controles de seguridad debe estar en concordancia con las instrucciones del fabricante, guías de implementación de seguridad y las especificaciones documentadas de seguridad.
	Evaluación de la seguridad del sistema. Validar que el sistema cumple con los requerimientos funcionales y de seguridad y operará con un nivel aceptable de riesgos residuales de seguridad.
	Aprobar el sistema de información. Es una autorización de seguridad del sistema de información para procesar, almacenar o transmitir información. La cual debe verificar la efectividad de los controles de seguridad y estar de acuerdo con el nivel de riesgo residual asociado.
	Preparación de revisiones operacionales. Cuando los cambios son significativos, es necesario realizar pruebas sobre los controles de seguridad afectados, para asegurar la integridad de los mismos.
Operación y Mantenimiento	**Administración y control del desempeño de la configuración.** Las políticas y procedimientos para establecer los componentes base de hardware, software y firmware del sistema de información, y subsecuentemente para controlar y mantener un inventario de los cambios al mismo que pueden tener un impacto significativo en la seguridad.
	Conducir un monitoreo continuo. Determinar si los controles de seguridad en el sistema de información, continúan siendo efectivos a través del tiempo, dado los cambios que ocurren en el sistema así como en el ambiente en el cual opera.
	Construir y ejecutar un plan de retiro o transición. Da a conocer a todos los involucrados, los planes futuros para el sistema y su información. Deberá informar sobre el status de los componentes críticos, servicios e información.
	Asegurar la preservación de la información. Considerar los métodos que podrán ser requeridos para obtener información en el futuro.
Retiro	**Limpieza de los medios**. Limpiar o destruir los medios de información digital antes de su retiro o liberación para su reúso afuera de la organización. Con el fin de prevenir que individuos no autorizados, obtengan acceso y usen la información contenida en los medios.
	Retiro de hardware y software. El retiro de software deberá cumplir con las licencias y otros acuerdos con el desarrollador o con las regulaciones de los gobiernos. En situaciones donde los medios de almacenamiento no pueden ser limpiados apropiadamente, es posible su destrucción física.
	Cierre del sistema. La información del sistema es formalmente dada de baja y desmontada.

Tabla 2.12 – Actividades de seguridad en SDLC del NIST

2.2.2.2 ISO/IEC 27002:2005

La Organización Internacional de Normalización (ISO, International Organization for Standardization) publicó el estándar de "Tecnologías de la Información, Técnicas de Seguridad y Prácticas de Código para la Administración de la Seguridad de la Información" [40]. Anteriormente conocido como ISO/IEC 17799:2005, este documento provee guías y principios generales para iniciar, implementar, mantener y mejorar, la administración de la seguridad de la información en una organización.

En general, podemos calificar a este estándar como un marco de gestión de la seguridad basado en riesgos, donde la seguridad de la información abarca tanto a los sistemas como a los activos de información. Se encuentra estructurado en 11 dominios, con 39 categorías y 133

controles de seguridad. Cada uno de estos dominios, conforma un capítulo de la norma (describiéndose a partir de la sección cinco), y se centra en un determinado aspecto de la seguridad de la información. En la siguiente ilustración, Figura 2.3, se muestra la distribución de dichos dominios y el aspecto de seguridad a cubrir.

ISO/IEC 27002:2005 no obliga a los usuarios a implementar todos y cada uno de los controles descritos; sino más bien permite a la empresa, a través de un proceso de análisis de riesgos, identificar, seleccionar e implementar los controles que mejor se adapten a sus requerimientos específicos.

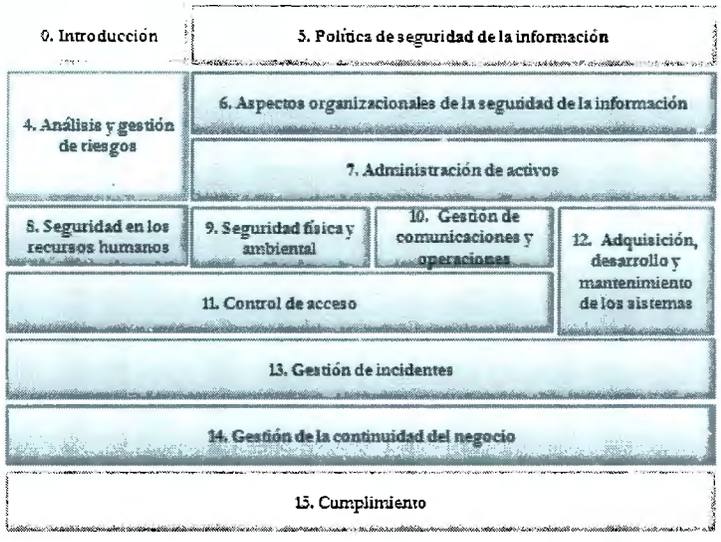


Figura 2.3 - Distribución de los dominios de la norma ISO 27002:2005

Ahora bien, el capítulo que nos interesa para su estudio, es con referencia a 12. "Adquisición, Desarrollo y Mantenimiento de los Sistemas de Información". Las categorías de seguridad y los controles a implementar, se describen a continuación.

Categoría de seguridad	Controles de seguridad
Requerimientos de seguridad de los sistemas de información	**Análisis y especificación de los requerimientos de seguridad**. Los requerimientos para los sistemas de información deberán especificar los requerimientos para los controles de seguridad. Reflejando el valor comercial de los activos de información involucrados, y el daño comercial potencial resultante de una falla o ausencia de seguridad.
Procesamiento correcto en las aplicaciones	**Validación de los datos de entrada.** Los datos de entrada de las aplicaciones deberán ser validados para asegurar que los datos son correctos y apropiados. Para ello, se deberá realizar chequeos en los datos de entrada de las transacciones del negocio, los datos fijos, y las tablas de parámetros.
	Control del procesamiento interno. Deberán incorporarse chequeos de validación para detectar cualquier corrupción de la información por errores del procesamiento o actos deliberados. Asegurando que se minimicen los riesgos de fallas en el procesamiento que lleven a la pérdida de integridad.

	Integridad de los mensajes. Los requerimientos para autenticidad y proteger la integridad del mensaje en las aplicaciones, deberán identificarse así como los controles a ser implementados. Para ello se realizará una evaluación sobre los riesgos de seguridad para proteger la integridad de los mensajes.
	Validación de los datos de salida. Validar los datos de salida de una aplicación para asegurar que el procesamiento de la información almacenada sea correcto y apropiado a las circunstancias.
Controles criptográficos	**Política de uso de los controles criptográficos**. Desarrollar e implementar una política sobre el uso de controles criptográficos, considerando: un enfoque gerencial, la evaluación del riesgo, uso de criptografía, gestión de claves, roles y responsabilidades, entre otros.
	Gestión de claves. Deberá permitir el uso de técnicas criptográficas a la organización. Para ello, todas las claves criptográficas deberán estar protegidas contra su modificación, pérdida y destrucción. Las claves secretas y privadas deberán protegerse contra la divulgación no-autorizada.
	Control del software operacional Establecer procedimientos para el control de la instalación del software en los sistemas operacionales. y minimizar el riesgo de corrupción en los mismos.
Seguridad de los archivos del sistema	**Protección de los datos de prueba del sistema**. Los datos de prueba deberán seleccionarse cuidadosamente, así como ser protegidos y controlados. Evitando el uso de bases de datos operacionales que contienen información personal o cualquier otra información confidencial para propósitos de pruebas.
	Control de acceso al código fuente de los programas. Se deberá controlar estrictamente su acceso para evitar la introducción de una funcionalidad no-autorizada y evitar cambios no-intencionados.
	Procedimientos para el control de cambios. La implementación de cambios deberá controlarse a través del uso de procedimientos formales de control de cambios. Incluyendo una evaluación de riesgos, un análisis de los impactos del cambio y las especificaciones de los controles de seguridad necesarios.
	Revisión técnica de aplicaciones tras efectuar cambios en el sistema. Cuando se cambian los sistemas operativos, las aplicaciones críticas del negocio deberán revisarse y probarse, asegurando que no exista un impacto adverso en las operaciones de la organización o en la seguridad.
Seguridad en los procesos de desarrollo y soporte	**Restricciones sobre cambios en los paquetes de software.** Utilizar los paquetes de software suministrados por vendedores sin modificaciones. Si es necesario realizar cambios, limitarlos y controlarlos.
	Fugas de información. Deberán prevenirse las oportunidades o limitar el riesgo para que se dé fuga de información. Para lo cual se deberá escanear el flujo de salida de los medios y las comunicaciones en busca de información escondida, enmascarar y modular las comunicaciones del sistema, y monitorear las actividades del personal y sus recursos, entre otros.
	Desarrollo de software externo. El desarrollo de software por *out-sourcing*, deberá ser supervisado y monitoreado por la organización. Se deberán considerar los contratos de licencias, propiedad de códigos y derechos de propiedad intelectual; la certificación de la calidad y exactitud del trabajo realizado, requerimientos contractuales y pruebas, entre otros.
Gestión de la vulnerabilidad técnica	**Control de las vulnerabilidades técnicas**. Obtener oportunamente información sobre las vulnerabilidades técnicas de los sistemas de información que se están utilizando, se deberá evaluar la exposición de la organización a dichas vulnerabilidades, y las medidas para tratar los riesgos asociados.

Tabla 2.13 – Controles de seguridad en ISO/IEC 27002:2005

2.2.2.3 *Guía de desarrollo de OWASP*

El proyecto de seguridad abierto para las aplicaciones web (OWASP, Open Web Application Security Project), es una comunidad abierta y libre a nivel mundial enfocada a mejorar la seguridad en las aplicaciones de software. Organiza diversos proyectos de código abierto dedicados a determinar y combatir las causas que hacen al software inseguro, encontrándose entre ellos el "Proceso de Seguridad, Extenso y Ligero para las Aplicaciones" (CLASP, Comprehensive, Lightweight Application Security Process), el "Modelo de Madurez para el Aseguramiento del Software" (SAMM, Software Assurance Maturity Model) y la "Guía para Construir Aplicaciones y Servicios Web Seguros".

Con referencia al proyecto CLASP [41], éste provee un enfoque para mapear conceptos de seguridad dentro de las primeras etapas del ciclo de vida de desarrollo del software. Para ello, establece un conjunto de actividades de seguridad que deben realizar los roles involucrados en el desarrollo del sistema (como son: el administrador del proyecto, el arquitecto, el ingeniero en requerimientos, el auditor, etc.), para posteriormente, ajustar estas actividades al proceso de desarrollo de la aplicación.

Una actualización significativa del proyecto CLASP, es el modelo SAMM [42]; el cuál es un marco de trabajo para ayudar a las organizaciones a formular e implementar una estrategia de seguridad en el software. Se encuentra construido alrededor de las 4 funciones de negocio principales relacionadas con el desarrollo de sistemas. Estas funciones a su vez definen 3 prácticas o actividades de seguridad (véase Figura 2.4.). Finalmente, para cada práctica de seguridad se definen tres niveles de madurez como objetivos, los cuales permiten ir incrementando el nivel de aseguramiento del software.

Figura 2.4 – Niveles del marco de trabajo SAMM de OWASP

De manera general, las prácticas de seguridad que propone, se exponen a continuación:

Funciones de Negocio	Prácticas de Seguridad
Gobierno	**Estrategia y métricas.** Involucra la dirección estratégica global del programa de aseguramiento de software e instrumentación de procesos y actividades para recolectar métricas acerca de la postura de seguridad de una organización.
	Política y cumplimiento. Establece una estructura de control y auditoria para seguridad y cumplimiento de regulaciones a lo largo de una organización para alcanzar un aseguramiento superior en software bajo construcción y en operación.
	Educación y orientación. Incrementa el conocimiento de seguridad entre el personal de desarrollo de software a través de entrenamiento y orientación en temas de seguridad pertinentes a funciones del trabajo individual.
Construcción	**Evaluación de amenazas.** Identificar y caracterizar con precisión los ataques potenciales contra el software de una organización, con el fin de comprender mejor los riesgos y facilitar su gestión.
	Requisitos de seguridad. Promover la inclusión de las necesidades de seguridad durante el proceso de desarrollo de software a fin de especificar la funcionalidad correcta desde el principio.
	Arquitectura de seguridad. Implica el fortalecimiento del proceso de diseño con actividades para promover diseños con seguridad en mente y los marcos de trabajo en que se basa el software.
Verificación	**Revisión de diseño.** Inspección de artefactos creados a partir del proceso de diseño para asegurar la provisión de mecanismos de seguridad adecuados y apegados a las expectativas de seguridad de la organización.
	Revisión de código. Evaluación del código fuente de una organización para ayudar en el descubrimiento de vulnerabilidades y actividades relacionadas a la mitigación como es el establecimiento de bases para las expectativas de la seguridad en programación.
	Pruebas de seguridad. Probar el software de la organización en su ambiente de ejecución para descubrir vulnerabilidades y establecer un estándar mínimo para la liberación del software.
Implementación	**Administración de vulnerabilidades.** Establecer procesos consistentes para administrar reportes internos o externos de vulnerabilidades para limitar la exposición, recopilar datos y así mejorar el programa de aseguramiento.
	Fortalecimiento de ambientes. Implementación de controles para el ambiente operativo que rodea a los programas de una organización para reforzar la postura de seguridad de las aplicaciones que han sido implementadas. **Habilitación operativa.** Identificar y capturar información relevante a la seguridad que necesita un operador para configurar, instalar y correr los programas de una organización.

Tabla 2.14 – Actividades de seguridad en el proceso SAMM de OWASP

En adición a los proyectos anteriores, OWASP publicó la "Guía para Construir Aplicaciones y Servicios Web Seguros" [43]. Este documento contiene diversos detalles prácticos de seguridad para la mayoría de las aplicaciones y servicios web. Además, relaciona dichas prácticas con el COBIT, ayudando rápidamente a identificar su cumplimiento con la ley Sarbanes Oxley Act. Por lo que muchas veces esta guía es referenciada tanto por las instituciones gubernamentales como por las instituciones financieras. Su enfoque es principalmente hacia los arquitectos, desarrolladores, consultores y auditores, debido a que ofrece un manual para diseñar, desarrollar, e implementar aplicaciones web seguras.

En su versión más reciente, la cual solo puede consultarse a través del *Wiky* del proyecto [44], se establecen diversos controles de seguridad a implementar dentro de las aplicaciones, mismos que se encuentran enfocados a los siguientes temas:

- Arquitectura de seguridad
- Autenticación
- Administración de sesiones
- Controles de acceso
- Validación de datos de entrada
- Cifrar y escapar salidas de datos
- Criptografía
- Manejo de errores y *logs*
- Protección de datos
- Comunicación segura
- Seguridad en HTTP
- Configuración de la seguridad
- Búsqueda de código malicioso
- Seguridad interna

2.2.3 Actividades de seguridad identificadas dentro de los estándares y guías de desarrollo

Al comparar las etapas propuestas por las guías y estándares presentados para el ciclo de vida de desarrollo de software, se observó que la metodología del "Manual de Desarrollo y Adquisición" del FFIEC, es quien mejor engloba a las diferentes fases de construcción de los sistemas. Sin embargo, se decidió unir en la actividad de "Análisis", las etapas de "Iniciación" y "Planeación" dado que en la práctica, y como se muestra en el resto de las metodologías, estas actividades se ejecutan normalmente a la par. Cabe mencionar, que no se contempló la actividad de "gobierno" de SAMM, porque se encuentra fuera del alcance del proyecto.[6]

Etapas propuestas	Manuales del FFIEC	NAI del Banco de México	NIST SP800-64	SAMM
---	---	---	---	Gobierno
Análisis	Iniciación Planeación	Requerimientos	Iniciación	Construcción
Diseño	Diseño	Diseño		
Desarrollo	Desarrollo	Implementación	Desarrollo	
Pruebas	Pruebas	Pruebas		Verificación
Producción	Implementación	Implantación	Implementación	Implementación
Mantenimiento	Mantenimiento		Operación y Manto.	
Retiro	Desactivación	---	Retiro	---

Tabla 2.15 – Comparación de los SDLC de los estándares para el desarrollo seguro

Bajo estas nuevas etapas, se integraron las diferentes actividades de seguridad expuestas por las guías, identificando así, las siguientes tareas comunes.

[6] Para mayor referencia sobre el alcance del proyecto, favor de consultar la sección 1.3.1 "Alcances".

Etapa de SDLC	Actividades propuestas	BMI	FFIEC	NIST	ISO	SAMM
Análisis	Identificación de los requerimientos funcionales y de seguridad	Definición de roles	Requerimientos de seguridad Establecer roles y responsabilidades		Análisis y especificación de los requerimientos de seguridad	Requisitos de seguridad
	Clasificación de la información y la seguridad	Clasificación de información, control de acceso y confidencialidad		Categorización de la seguridad		
	Análisis de riesgos	Análisis de riesgos	Administración de riesgos	Evaluación del impacto en el negocio Evaluación del impacto en la privacidad Valoración de riesgos		Evaluación de amenazas
	Selección de controles	Documentar procesos de validación, control y seguimiento	Controles de seguridad dentro de las aplicaciones	Seleccionar y documentar controles de seguridad		
	Planeación de la seguridad	Establecer mecanismos para auditoría Usar depósitos institucionales		Planeación inicial de la seguridad Usar un proceso seguro para el desarrollo del sistema	Política de uso de los controles criptográficos	
Diseño	Arquitectura de seguridad	Mecanismos de seguridad y telecomunicaciones Arquitectura de acuerdo a la infraestructura de seguridad	Diseño de las características de seguridad	Diseñar una arquitectura de seguridad		Arquitectura de seguridad Revisión del diseño
	Diseño de controles de autenticación y autorización	Usar medidas para contraseñas			Gestión de claves	
	Diseño de controles de entrada		Controles de entrada			
	Diseño de controles de proceso		Controles de proceso		Integridad de los mensajes	
	Diseño de controles de salida		Controles de salida			
	Diseño de controles de monitoreo y auditoría				Control de vulnerabilidades técnicas	Administración de vulnerabilidades
Desarrollo	Configuración del ambiente de desarrollo		Manejo de versiones Documentación y control de acceso		Restricciones sobre cambios en los paquetes de software Control de acceso al código fuente de los programas	
	Desarrollo seguro y su control	Usar librerías de sitios de confianza Usar bibliotecas de seguridad institucionales	Estándares de código Controles de librerías	Ingeniería en seguridad y controles de desarrollo Desarrollar documentación de seguridad		
	Revisión del código				Validación de los datos de entrada Control del procesamiento interno Validación de los datos de salida	
Pruebas	Configuración del ambiente de prueba	Ambiente independiente de pruebas	Plan de pruebas	Conducir pruebas	Protección de los datos de prueba del sistema	

	Inspección de la seguridad en los módulos de la aplicación					Revisión del código
	Inspección de la seguridad integral de la aplicación	Pruebas en operación normal y de contingencia Pruebas de impacto Pruebas de vulnerabilidad	Pruebas funcionales con seguridad Verificar datos introducidos y su configuración Verificar exactitud de resultados y configuraciones seguridad Validar cumplimiento de objetivos	Evaluación de la seguridad del sistema		Pruebas de seguridad
Producción	Configuración de la seguridad en el ambiente de producción	Eliminar cuentas del equipo de desarrollo Respaldos del código fuente de la versión en producción		Integrar seguridad en ambientes y sistemas establecidos		
	Certificación de la aplicación	Certificar aplicación Registrar derechos de autor		Aprobar el sistema de información Plan para certificación y acreditación		
	Distribución e instalación segura de la aplicación				Control del software operacional	Habilitación operativa
Mantenimiento	Monitoreo de la seguridad y disponibilidad de las aplicaciones	Procedimientos de respaldo y recuperación	Incrementar seguridad Alerta de vulnerabilidades externas Revisiones de riesgos de seguridad	Preparación de revisiones operacionales Administración y control del desempeño de la configuración Conducir un monitoreo continuo	Revisión técnica de aplicaciones tras efectuar cambios en el sistema Fugas de información	Fortalecimiento de ambientes
	Control cambios		Proceso de administración de cambios Procedimientos para asegurar que los cambios no afecten la seguridad Controles para cambios de emergencia		Procedimientos para el control de cambios	
	Plan de contingencia	Plan de contingencia				
Retiro	Plan de retiro de la aplicación		Planeación para transferir datos, respaldos y pruebas	Construir y ejecutar un plan de retiro o transición		
	Preservación de la información			Asegurar la preservación de la información		
	Medidas de limpieza			Limpieza de los medios Retiro de hardware y software		
	Retiro y cierre de la aplicación			Cierre del sistema		

Tabla 2.16 –Actividades de seguridad dentro de los estándares y guías de desarrollo

2.3 Análisis crítico

Como se había comentado anteriormente, la producción de software seguro se ha convertido en un tema multifacético, el cual requiere que la ingeniería de software, la ingeniería de seguridad y la administración se combinen para lograr este fin. [9]

A pesar de esto, actualmente los profesionales en ingeniería de software e ingeniería de seguridad se encuentran desconectados entre sí. [11] Lo cual ha provocado, que los controles de seguridad se apliquen al sistema cuando su construcción ha concluido, o bien se subsane la falta de los mismos, con el empleo de medios externos a la aplicación para su protección (como son los controles de seguridad sobre la red); los cuales no siempre previenen los diferentes tipos de vulnerabilidades presentados por el sistema.

Esto no lleva de ninguna manera, a la falta de implementación de controles externos en las aplicaciones web, sino más bien, a que éstas se diseñen y desarrollen pensando en términos de seguridad. Como hemos visto durante la sección 1.2 "Definición del problema", los ataques dirigidos hacia los sistemas, se han incrementado con el fin de extraer la información manejada por ellos [7]. De esta manera, si alguno de los controles externos fallan, la aplicación debe estar lo suficientemente preparada para bloquear estos ataques.

Es por ello, que el presente trabajo se enfoca principalmente a la integración de la ingeniería de seguridad dentro de la ingeniería de software, dejando a un lado por el momento, conceptos más avanzados de administración de proyectos.

Con este fin en mente, y dado que la presente investigación se centra en los sistemas manejados por las instituciones financieras, se analizaron primero los requerimientos de seguridad para este tipo de instituciones a través de las leyes aplicables a este sector.

Posteriormente se identificaron diferentes estándares para el desarrollo de software, los cuales comprenden esta integración, eligiéndose para el sector financiero la NAI del Banco de México y los manuales del FFIEC. Complementando, se seleccionaron algunos estándares internacionales de carácter general utilizados por las instituciones financieras, como son: la metodología de seguridad que propone NIST SP800-64, el estándar ISO/IEC 27002:2005 y el proceso CLASP, el marco de trabajo SAMM y la guía de desarrollo de OWASP. Finalmente, al analizarlas se obtuvieron los siguientes resultados:

Características	NAI del Banco de México	Manuales del FFIEC	NIST SP800-64	ISO/IEC 27002:2005	CLASP, SAMM y Guía Desarrollo OWASP
Alcance del estándar	Banco de México	Instituciones Financieras	Industria general	Industria general	Industria general
Toma en cuenta regulación financiera	Nacional	GBL	No	No	SOX
Manejo de seguridad en:	Etapas del SDLC	Etapas del SDLC	Etapas del SDLC	Durante el desarrollo	SDLC (SAMM) Desarrollo
Define una metodología de desarrollo de sistemas	No	Si	Si	No	No
Mapea actividades de seguridad en el ciclo de vida	Si	Si	Si	No	Si (SAMM)
Establece un análisis de riesgos	Si	Si	Si	No	Si
Establece actividades generales de seguridad	Si	Si	Si	No	Si (SAMM)
Establece guías para políticas o controles de alto nivel	No	Si	Si	Si	Si (CLASP y SAMM)
Establece prácticas técnicas o controles de bajo nivel	Si	No	No	No	Si (Guía Desarrollo)
Controles de seguridad integrados al cuerpo de la metodología	No	No	No	Si	No

Tabla 2.17 – Características de los estándares para el desarrollo seguro

Así, tomando en cuenta el ciclo de vida establecido por el FFIEC, se analizaron y mapearon los diferentes tipos de mecanismos de seguridad (Actividades, guías y prácticas técnicas) que proponen los estándares de desarrollo de software. Obteniendo las siguientes observaciones.

Características	NAI del Banco de México	Manuales del FFIEC	NIST SP800-64	ISO/IEC 27002:2005	CLASP, SAMM y Guía Desarrollo OWASP
Análisis	Actividades	Actividades	Guías	Guías	Actividades
Diseño	Actividades	Actividades	Actividades	Guías	Actividades
Desarrollo	Prácticas técnicas	Guías	Actividades	Guías	Guías y Prácticas técnicas
Pruebas	Actividades	Actividades	Guías	Actividades	Prácticas técnicas
Producción	Actividades	---	Guías	Guías	Actividades
Mantenimiento	Actividades	Guías	Guías	Guías	Actividades
Retiro	---	Actividades	Guías	---	---

Tabla 2.18 – Tipos de mecanismos de seguridad en el SDLC de los estándares para el desarrollo seguro

Finalmente, al analizar los cuadros anteriores, concluimos lo siguiente:

- Con referencia a las guías específicas para el desarrollo de sistemas financieros (Banxico y FFIEC), las dos establecen actividades a seguir dentro del ciclo de vida de desarrollo del software. Sin embargo, los mecanismos de seguridad establecidos, se manejan en documentos separados, lo cual lleva al lector a tener que mapear estos por su cuenta dentro de la metodología de desarrollo. Además, no se establecen prácticas técnicas para todas las etapas del ciclo de vida, donde la mayoría de ellas, se quedan a nivel general sin concretar las mismas.
- En cuanto a los demás estándares, el NIST SP800-64 ofrece una metodología de desarrollo de sistemas, la cual involucra actividades de seguridad en cada una de las etapas del ciclo de vida propuesto. Sin embargo esta metodología posee una grave deficiencia, la cual es con relación al diseño y desarrollo de sistemas, en donde las guías que se ofrecen para este tema son demasiado generales. Lo cual puede ocasionar que se cometan errores en el desarrollo de controles de seguridad.
- Con referencia al proyecto SAMM de OWASP, ofrece una serie de actividades de seguridad a implementar enfocándolas a funciones de negocio dentro del desarrollo de sistemas, sin necesariamente proponer un ciclo de vida, lo que ocasiona que actividades como el desarrollo del sistema, no se contemplen dentro de esta metodología. Por su parte CLASP, también establece una serie de actividades a implementar durante el ciclo de vida de desarrollo de software. Sin embargo, no mapea estas actividades en las etapas, sino por medio de roles, dejando nuevamente el mapeo subjetivo de estas actividades al lector.
- Finalmente ISO/IEC 27002:2005 y la Guía de Desarrollo de OWASP ofrecen los controles más completos para la etapa de desarrollo (o codificación) del software, lamentablemente carecen de mecanismos prácticos para el diseño seguro. Además éstos no se mapean dentro de las etapas del ciclo de vida del software.

De esta manera podemos concluir, que las metodologías existentes en el mercado, carecen de guías y prácticas técnicas para la incorporación de seguridad en el diseño y el desarrollo de los sistemas. Esto es altamente preocupante, puesto que un gran porcentaje de las vulnerabilidades detectadas en las aplicaciones, son causadas por el mal uso de modelos de seguridad dentro de las etapas de arquitectura y diseño del sistema, conforme a los análisis de vulnerabilidades de aplicaciones de CERT [9]

Así, la meta a alcanzar es incorporar actividades y mecanismos de seguridad en cada una de las etapas del ciclo de vida del software. Evitando su inclusión hasta que el proceso haya concluido o bien, hasta que se detecte una vulnerabilidad.

Debido a esto, la presente propuesta de tesis pretende establecer actividades de seguridad para cada una de las etapas del ciclo de vida del sistema, las cuales puedan integrase a la metodología existente de desarrollo[7]. Con referencia al ciclo de vida, se tomará como base el propuesto en la sección 2.2.3 "Actividades de seguridad identificadas dentro de los estándares y guías de desarrollo", la cual toma como referencia al FFIEC. Agregando además, las actividades de seguridad previamente identificadas en esa misma sección; para las que se tomó como referencia las normas del Banco de México, FFIEC, NIST SP800-64 y SAMM. Finalmente, se complementarán estas actividades al incluir mejores prácticas técnicas para la seguridad en el diseño e implementación de los sistemas con los estándares de ISO y OWASP: las cuales permitan disminuir el número de vulnerabilidades que presentan las aplicaciones web, y en donde los sistemas desarrollados bajo estas prácticas, permitan cumplir con la legislación vigente en cuanto a materia de seguridad dentro del sector financiero mexicano.

[7] Para mayor referencia sobre la selección de esta metodología y su aplicación con el desarrollo del presente trabajo, favor de consultar la introducción del capítulo 4. "Metodología para el desarrollo de aplicaciones financieras seguras".

CAPÍTULO 3

3 Seguridad en las Aplicaciones Financieras

Recordando nuevamente las secciones 1.1 "Antecedentes" y 1.2 "Definición del problema" del presente trabajo, observamos como el sector financiero es quién más atención genera a los criminales informáticos, debido a la remuneración económica a la cual pueden hacerse acreedores si logran perpetuar sus ataques (Véase Figura 1.3 del capítulo 1). Debido a ello, estas instituciones continuamente se encuentran expuestas al robo de información confidencial tanto de los clientes como de la misma compañía. Lo cual motiva a las instituciones, a establecer dentro de sus aplicaciones una serie de controles de seguridad para prevenir dichos incidentes.

Pero, ¿cuáles son los ataques que enfrentan las aplicaciones financieras así como sus vulnerabilidades, para de esta manera, realizar un análisis de riesgos y determinar los controles generales a aplicar en nuestros sistemas?

Dentro de la presente sección revisaremos estos puntos, los cuales al final, nos permitirán identificar los controles computacionales generales de seguridad a establecer en este tipo de aplicaciones. Mismos que nos permitirán en el siguiente capítulo, seleccionar los patrones de seguridad a implementar dentro del desarrollo de nuestras aplicaciones financieras.

3.1 Conceptos básicos de seguridad

Para entrar más formalmente en materia de seguridad, a continuación se revisará la terminología básica con respecto a este tema, la cual se utilizará en diversas ocasiones durante el desarrollo del presente trabajo.

Comenzaremos por definir a la **seguridad de la información** [40], como la protección de la información de un rango amplio de amenazas con el fin de asegurar la continuidad del negocio, minimizar los riesgos y maximizar el retorno de la inversión así como las oportunidades comerciales. La misma se logra implementando un adecuado conjunto de controles; incluyendo políticas, procesos, procedimientos, estructuras organizacionales, y funciones de software y hardware. Los cuales se necesitan establecer, implementar, monitorear, revisar y mejorar continuamente para asegurar que cumplan los objetivos de seguridad y del negocio. Donde estos controles deben realizarse en conjunción con los otros procesos de administración del negocio.

Más específicamente, la **seguridad del sistema de información** [45] es la protección de la información procesada por el sistema de información de una divulgación no autorizada (violación de la confidencialidad), de modificaciones no acreditadas (violación de la integridad) o de la denegación del servicio (violación de la disponibilidad). Adicionando otras propiedades tales como: la autenticidad, la responsabilidad, la no repudiación y la confiabilidad. La razón detrás de esto, se debe a que estos sistemas y su tecnología proveen del medio por el cual las personas acceden, manipulan y transportan la información.

Ahora bien, una **amenaza** a un sistema de software [45], es un actor, agente, circunstancia o evento el cual tiene la posibilidad de causar daño al sistema, a los datos o a los recursos a los cuales el sistema accede. Las amenazas pueden categorizarse de acuerdo a su intención, siendo éstas: no intencionales, intencionales pero no maliciosas o maliciosas.

Un concepto que va de la mano junto con las amenazas, son las vulnerabilidades. De acuerdo con CVE [46], una **vulnerabilidad** en la seguridad de la información, es un error en el software el cual puede ser usado directamente por un hacker para obtener acceso al sistema o a la red. Más específicamente, se considera como un estado en el sistema computacional el cual permite a un atacante: ejecutar comandos como otro usuario, acceder a datos fuera de las restricciones de acceso especificadas, presentarse como otra entidad, y conducir a una denegación del servicio. Cabe mencionar que las vulnerabilidades se pueden tratar de eliminar, mas no así las amenazas.

Un **riesgo** [47] es una medida sobre el alcance de una amenaza en una entidad, ya sea por una circunstancia o evento potencial. Típicamente se encuentra en función a: los impactos adversos que puede llegar a ocasionar si la circunstancia o evento ocurre, y la probabilidad de ocurrencia. En adición, los riesgos de seguridad relacionados con los sistemas de información, son aquellos surgidos de la perdida de confidencialidad, integridad o disponibilidad de la información en los mismos. Reflejándose como impactos adversos potenciales en las operaciones de la organización (incluyendo su misión, funciones, imagen o reputación) y los activos organizacionales, individuales o de la Nación.

Por último tenemos al **análisis de riesgos** [40], proceso mediante el cual, se identifican las amenazas y las vulnerabilidades de una organización, valorando su impacto y la probabilidad de ocurrencia. Permite además, seleccionar medidas de protección con costos adecuados para mitigar los riesgos en los procesos críticos de la organización. Considerándose así, como parte fundamental en la administración de la seguridad.

3.2 Amenazas y ataques informáticos

Actualmente, los sistemas de información de las organizaciones se están enfrentando con una amplia gama de amenazas de seguridad las cuales provienen de diversas fuentes, como son: el fraude asistido por computadora, el espionaje, el sabotaje, el vandalismo, y los desastres naturales. Además de estos problemas, ataques de tipo código malicioso, hackeo y denegación del servicio, han empezado a ser cada vez más comunes, ambiciosos y sofisticados [40].

Mientras todos los tipos de amenazas vistos con anterioridad son capaces de comprometer la seguridad de software, solo las amenazas maliciosas son realizadas a través de ataques. Para ello, la mayoría de las actividades contra el software, toman ventaja o explotan alguna vulnerabilidad o debilidad en el software [45]. Así un **ataque**, es un intento de ganar acceso hacia los servicios del sistema o bien comprometer alguna de sus propiedades (integridad, confiabilidad, disponibilidad).

Como vimos, los sistemas de software pueden ser sujetos a una amplia variedad de amenazas que pueden comprometer su seguridad. Sin embargo, la mayoría de estas actividades siguen una estrategia básica, la cual es construir un ataque a través de una serie de ataques más pequeños o anatómicos.

Conforme al reporte de Symantec sobre "Ataques Basados en Web" [48], la anatomía general de un ataque por internet se refleja en la Figura 3.1

Figura 3.1 – Anatomía de un ataque en web

En general, este tipo de actividades maliciosas siguen la misma secuencia de eventos, los cuales son:

1. **El atacante rompe la seguridad de un sitio web legítimo e introduce código malicioso.** Como vimos en la "Introducción" del documento, el código malicioso ya no es solamente exclusivo de los sitios web dudosos, hoy en día es común que sitios web legítimos actúen como contenedores de este tipo de código.
2. **Se ataca a las máquinas de los usuarios finales.** El código malicioso del sitio web, se baja hacia la computadora del usuario cuando éste visita el sitio infectado. Para ello existen diversas técnicas donde algunas requieren o no la interacción del usuario.
3. **Conectándose a la máquina del usuario final para realizar actividades maliciosas.** Una vez que el código malicioso se ha establecido en la máquina del usuario final, empieza entonces la actividad maliciosa como puede ser entre otras, el robo de la información personal del usuario (número de tarjetas de crédito, contraseñas, etc.)

Ahora bien, con relación a las aplicaciones financieras, y como vimos en la sección 1.1 "Antecedentes", éstas se encuentran expuestas a la mayoría de los ataques por internet, debido al tipo de información manejada, la cual es sumamente valiosa para este tipo de personas. Así, el "Informe Anual Global de Seguridad" realizado por Deloitte [49], nos muestra a las principales actividades maliciosas a las que están expuestas estas instituciones.

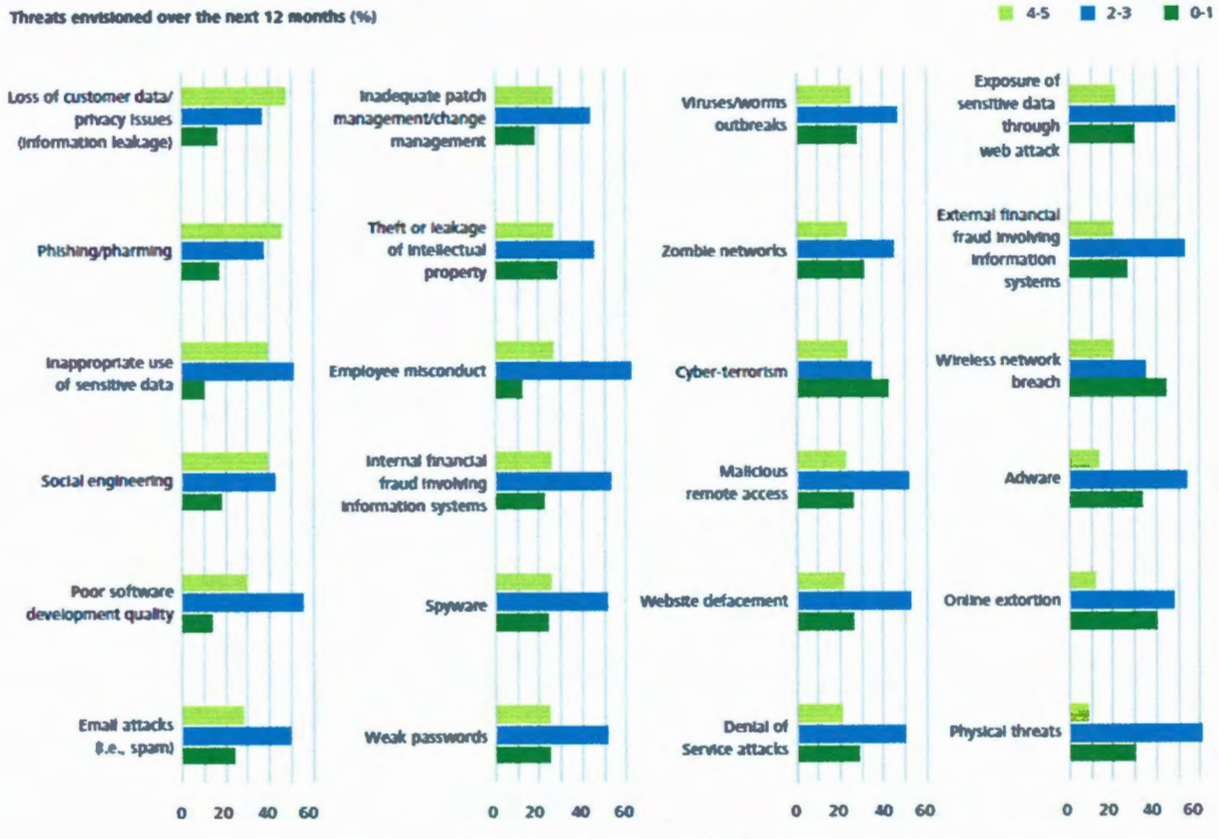


Figura 3.2 – Ataques hacia las instituciones financieras

3.3 Vulnerabilidades

La principal razón que motiva y facilita a los criminales para realizar sus ataques, es la presencia casi garantizada de vulnerabilidades en el software, las cuales pueden ser explotadas para violar alguna de las propiedades de seguridad del mismo. De acuerdo con CERT, la mayoría de los ataques exitosos resultan de la explotación de vulnerabilidades de software conocidas y no arregladas, así como de configuraciones de software inseguras; muchas de las cuales fueron introducidas durante el diseño y la codificación de los sistemas [50].

De igual manera Gartner, en su teleconferencia sobre "Construcción de Aplicaciones Seguras" [51], menciona que en cada una de las diferentes fases del ciclo de vida de desarrollo de software, se introducen una serie de vulnerabilidades; siendo las fases de diseño y construcción donde se concentran los errores más graves.

Revisión de vulnerabilidades	Fases del SDLC	Vulnerabilidades introducidas (no necesariamente de seguridad)
Vulnerabilidades en requerimientos, flujo de procesos de negocio y algoritmos.	Análisis	15%
Vulnerabilidades causadas por relaciones entre módulos, servicios web y flujo de la lógica y datos.	Diseño	40%
Vulnerabilidades en instrucciones del lenguaje e implementación del flujo de la lógica y datos.	Construcción	35%
Vulnerabilidades en ejecutables, Interfaz de usuario y ensamble inseguro de los servicios de seguridad.	Pruebas	10%
Falta de parches, errores administrativos, configuración incorrecta. Si se encuentran vulnerabilidades se deberá regresar al análisis.	Operación	

Tabla 3.1 –Vulnerabilidades a lo largo del ciclo de vida

Es así, como a continuación se identificarán y analizarán las vulnerabilidades relacionadas con el desarrollo de las aplicaciones financieras, en especial aquellas generadas dentro de las etapas de diseño y codificación del sistema. Dado que las mismas, ofrecen un área de oportunidad a los criminales para cometer delitos con y en contra de las aplicaciones.

La justificación de analizar las vulnerabilidades en el presente trabajo, se debe a que estas son vistas como un tipo de error las cuales, de seguir prácticas correctas de seguridad, se pueden eliminar; más no así, las amenazas sufridas por los sistemas.

3.3.1 Vulnerabilidades en el diseño y desarrollo de sistemas

Dado la importancia de las vulnerabilidades en el diseño y desarrollo de sistemas, esta investigación se dio a la tarea de buscar aquellas que son más comunes y afectan en gran medida a las aplicaciones web, debido a su frecuente explotación por parte de los criminales. En este sondeo, se encontró dos fuentes principales: la "Base de Datos Nacional de Vulnerabilidades" de la NIST [52] y los "Errores de Software más Peligrosos" de SANS y MITRE [53].

De las dos fuentes anteriores, quien prioriza de una mejor forma las vulnerabilidades que afectan a las aplicaciones web, es la lista de errores realizada por SANS y MITRE. Quienes evalúan de acuerdo a su prevalencia, importancia y facilidad de utilización, los errores (vulnerabilidades) más frecuentes de programación, diseño y arquitectura. Así el resultado de dicho estudio para el 2011 fue el siguiente.

Nivel	Calificación	Identificador en el CWE	"Vulnerabilidad"
[1]	93.8	CWE-89	Neutralización incorrecta de elementos especiales usados en un comando SQL (Inyección de SQL)
[2]	83.3	CWE-78	Neutralización incorrecta de elementos especiales usados en un comando SO (Inyección de comandos de SO)
[3]	79	CWE-120	Copia de un *buffer* sin checar el tamaño de la entrada (Desbordamiento clásico de *Buffer*)
[4]	77.7	CWE-79	Neutralización incorrecta de entrada durante la generación de páginas web (*Cross-Site Scripting*)
[5]	76.9	CWE-306	Falta de autenticación para funciones críticas
[6]	76.8	CWE-862	Falta de autorización
[7]	75	CWE-798	Uso de credenciales *Hard-coded*
[8]	75	CWE-311	Falta de cifrado de datos sensitivos
[9]	74	CWE-434	Descargas no restringidas de tipos de archivos peligrosos
[10]	73.8	CWE-807	Confiar en entradas no confiables en una decisión de seguridad
[11]	73.1	CWE-250	Ejecución con privilegios innecesarios
[12]	70.1	CWE-352	Falsificación de Peticiones en Sitios Cruzados (CSRF)
[13]	69.3	CWE-22	Limitación incorrecta de una ruta de archivo a un directorio restringido (*Path Traversal*)
[14]	68.5	CWE-494	Descargar código sin revisar su integridad
[15]	67.8	CWE-863	Autorización incorrecta
[16]	66	CWE-829	Inclusión de funcionalidad de un ámbito de control no confiable
[17]	65.5	CWE-732	Asignación incorrecta de permisos para un recurso crítico
[18]	64.6	CWE-676	Uso de funciones potencialmente peligrosas
[19]	64.1	CWE-327	Uso de un algoritmo criptográfico riesgoso o vulnerable
[20]	62.4	CWE-131	Calculo incorrecto del tamaño de un *buffer*
[21]	61.5	CWE-307	Restricción incorrecta de intentos excesivos de autenticación
[22]	61.1	CWE-601	Direccionamiento de una URL a un sitio no confiable (Direccionamiento abierto)
[23]	61	CWE-134	Formato de cadenas no controlado
[24]	60.3	CWE-190	Desbordamiento de un *Integer* o su reciclado (*Wraparound)*
[25]	59.9	CWE-759	Uso de un *Hash* de un solo sentido sin sal

Tabla 3.2 –Vulnerabilidades más comunes en las aplicaciones web.

Ahora bien, con referencia a las aplicaciones financieras, X-Force en su reporte anual de "Amenazas y Riesgos" [54], identificó especialmente las vulnerabilidades más comunes en las aplicaciones web para el sector financiero. Comparando éstas, con los errores identificados por SANS y MITRE, tenemos los siguiente.

Nivel	Tipo de "vulnerabilidad"	Porcentaje de ocurrencia	Identificador en el CWE
1.	Uso incorrecto de SSL	90%	
2.	Falta de control en la información presentada en los mensajes de error	80%	
3.	Falsificación de Peticiones en Sitios Cruzados	75%	[12] CWE-352
4.	Revelación de información	75%	[8] CWE-311
5.	Incorrecta implantación de la aplicación	55%	
6.	Configuración ineficiente del Servidor Web	50%	
7.	Control inadecuado de entradas	40%	[10] CWE-807, [23] CWE-134
8.	Secuencia de Comandos en Sitios Cruzados	30%	[4] CWE-79
9.	Control de acceso incorrecto	30%	[5] CWE-306, [21] CWE-307
10	Cifrado no estándar	10%	[19] CWE-327.
11	Inyección de SQL	5%	[1] CWE-89.

Tabla 3.3 –Vulnerabilidades en las aplicaciones financieras

Como podemos observar, algunas de las vulnerabilidades detectadas para las aplicaciones financieras, no tienen su correspondiente entre la lista de errores comunes para las aplicaciones web. Siendo estas, aquellas relacionadas normalmente con la configuración de la aplicación (uso de SSL, manejo de mensajes de error, incorrecta implantación de la aplicación y configuración del servidor web). Las cuales incluso, se encuentran entre las de mayor recurrencia dentro de las aplicaciones del sector financiero.

Esto nos lleva a observar la necesidad de contar con mecanismos de seguridad no solo para el diseño y codificación de las aplicaciones, sino también para la configuración del ambiente del sistema, teniendo un especial cuidado en las entradas y salidas producidas por la aplicación. Por lo que, este tipo de incidentes sobre la configuración del sistema, deberán adicionarse a la lista de vulnerabilidades a resolver en las aplicaciones financieras.

3.4 Riesgos

Para minimizar el impacto negativo de los ataques hacia los sistemas de información, es necesario integrar actividades de administración de riesgos durante todo el ciclo de vida de la aplicación. Dentro de estas actividades, la identificación de los tipos de riesgo se vuelve parte esencial de este proceso.

Con referencia a este tema, OWASP en su proyecto "Top 10", describe los "Diez Riesgos Más Críticos sobre Seguridad en las Aplicaciones" [55]. Su objetivo es crear conciencia sobre la seguridad en aplicaciones mediante la identificación de algunos de los riesgos más importantes que enfrentan las organizaciones. Proveyendo además, de información genérica acerca de la probabilidad y el impacto técnico causado por los mismos. Así, los riesgos identificados por OWASP se describen a continuación.

Riesgos	Descripción
A1. Inyección	Las fallas de inyección, tales como SQL, OS, y LDAP, ocurren cuando datos no confiables son enviados a un intérprete como parte de un comando o consulta. Los datos hostiles del atacante pueden engañar al intérprete en ejecutar comandos no intencionados o acceder datos no autorizados.
A2. Secuencia de Comandos en Sitios Cruzados (XSS)	Las fallas XSS ocurren cada vez que una aplicación toma datos no confiables y los envía al navegador web sin una validación y codificación apropiada. XSS permite a los atacantes ejecutar secuencia de comandos en el navegador de la víctima los cuales pueden secuestrar las sesiones de usuario, destruir sitios web, o dirigir al usuario hacia un sitio malicioso.
A3. Perdida de Autenticación y Gestión de Sesiones	Las funciones de la aplicación relacionadas a autenticación y gestión de sesiones son frecuentemente implementadas incorrectamente, permitiendo a los atacantes comprometer contraseñas, llaves, *token* de sesiones, o explotar otras fallas de implementación para asumir la identidad de otros usuarios.
A4. Referencia Directa Insegura a Objetos	Una referencia directa a objetos ocurre cuando un desarrollador expone una referencia a un objeto de implementación interno, tal como un fichero, directorio, o base de datos. Sin un chequeo de control de acceso u otra protección, los atacantes pueden manipular estas referencias para acceder datos no autorizados.
A5. Falsificación de Peticiones en Sitios Cruzados (CSRF)	Un ataque CSRF obliga al navegador de una víctima autenticada a enviar una petición HTTP falsificado, incluyendo la sesión del usuario y cualquier otra información de autenticación incluida automáticamente, a una aplicación web vulnerable. Esto permite al atacante forzar al navegador de la víctima para generar pedidos que la aplicación vulnerable piensa son peticiones legitimas provenientes de la víctima.
A6. Configuración Defectuosa de Seguridad	Una buena seguridad requiere tener definida e implementada una configuración segura para la aplicación, marcos de trabajo, servidor de aplicación, servidor web, base de datos, y plataforma. Todas estas configuraciones deben ser definidas, implementadas, y mantenidas ya que por lo general no son seguras por defecto. Esto incluye mantener todo el software actualizado, incluidas las librerías de código utilizadas por la aplicación.
A7. Almacenamiento Criptográfico Inseguro	Muchas aplicaciones web no protegen adecuadamente los datos sensibles, tales como tarjetas de crédito, NSSs, y credenciales de autenticación con mecanismos de cifrado o *hashing*. Atacantes pueden modificar o robar tales datos protegidos inadecuadamente para conducir robos de identidad, fraudes de tarjeta de crédito u otros crímenes.

A8. Falla de Restricción de Acceso a URL	Muchas aplicaciones web verifican los privilegios de acceso a URLs antes de generar enlaces o botones protegidos. Sin embargo, las aplicaciones necesitan realizar controles similares cada vez que estas páginas son accedidas, o los atacantes podrán falsificar URLs para acceder a estas páginas igualmente.
A9. Protección Insuficiente en la Capa de Transporte	Las aplicaciones frecuentemente fallan al autenticar, cifrar, y proteger la confidencialidad e integridad de tráfico de red sensible. Cuando esto ocurre, es debido a la utilización de algoritmos débiles, certificados expirados, inválidos, o sencillamente no utilizados correctamente.
A10. Redirecciones y reenvíos no validados	Las aplicaciones web frecuentemente redirigen y reenvían a los usuarios hacia otras páginas o sitios web, y utilizan datos no confiables para determinar la página de destino. Sin una validación apropiada, los atacantes pueden redirigir a las víctimas hacia sitios de *phishing* o *malware*, o utilizar reenvíos para acceder páginas no autorizadas.

Tabla 3.4 –Riesgos críticos en las aplicaciones web, identificados por OWASP

3.4.1 Riesgos y vulnerabilidades financieras

Analizando el "Top 10" de OWASP, junto con las vulnerabilidades presentadas en la sección anterior (Véase Tabla 3.5), notamos que estos riesgos cubren la mayoría de las vulnerabilidades detectadas por SANS y MITRE, así como las vulnerabilidades para el sector financiero de X-Force. Sin embargo, existen todavía algunas vulnerabilidades que no se mapean totalmente.

Con referencia a SANS y MITRE, las vulnerabilidades que no poseen un riesgo asociado en el "Top 10" de OWASP son las siguientes [56]:

- [nivel 23] CWE-134. Formato de cadenas no controlado
- [nivel 10] CWE-807. Confiar en entradas no confiables en una decisión de seguridad
- [nivel 3] CWE-120. Desbordamiento clásico de *buffer*
- [nivel 18] CWE-676. Uso de funciones potencialmente peligrosas
- [nivel 20] CWE-131. Calculo incorrecto del tamaño de un *buffer*
- [nivel 24] CWE-190. Desbordamiento de un *Integer* o su reciclado

Por otro lado, con relación a X-Force, se identificó que algunas de las vulnerabilidades presentadas, aunque pueden mapearse con los riesgos descritos por la OWASP (2. Falta de control en la información presentada en los mensajes de error y 4. Revelación de información con A6. Configuración Defectuosa de Seguridad; y 7. Control inadecuado de entradas con A1. Inyección, A2. Secuencia de Comandos en Sitios Cruzados y A4. Referencia Directa Insegura a Objetos), y atenderse empleando una metodología de desarrollo seguro, dado su importancia, se consideró tratarlas como puntos aparte. De esta manera, se tomó la decisión de agregar tres nuevos tipos de eventos de riesgo a los proporcionados por la OWASP. Así, la nueva lista se presenta a continuación.

"Riesgos" OWASP (Top Ten)[8]	"Vulnerabilidades" SANS	"Vulnerabilidades" X-Force	Promedio importancia	Nivel de importancia (descendente)
1. Inyección	[1] CWE-89. Inyección de SQL [2] CWE-78. Inyección de comandos de SO	11. Inyección de SQL	3.75	12
2. Secuencia de Comandos en Sitios Cruzados	[4] CWE-79. Secuencia de Comandos en Sitios Cruzados	8. Secuencia de Comandos en Sitios Cruzados	4.66	11
3. Pérdida de Autenticación y Gestión de Sesiones	[5] CWE-306. Falta de autenticación para funciones críticas [7] CWE-798. Uso de credenciales *Hard-coded* [21] CWE-307. Restricción incorrecta de intentos excesivos de autenticación	9. Control de acceso incorrecto	9.00	8
4. Referencia Directa Insegura a Objetos	[6] CWE-862. Falta de autorización [9] CWE-434. Descargas no restringidas de tipos de archivos peligrosos [13] CWE-22. *Path Traversal* [15] CWE-863. Autorización incorrecta [16] CWE-829. Inclusión de funcionalidad de un ámbito de control no confiable	9. Control de acceso incorrecto	10.29	5
5. Falsificación de Peticiones en Sitios Cruzados	[12] CWE-352. Falsificación de Peticiones en Sitios Cruzados	3. Falsificación de Peticiones en Sitios Cruzados	6.66	9
6 Configuración Defectuosa de Seguridad	[11] CWE-250. Ejecución con privilegios innecesarios [17] CWE-732. Asignación incorrecta de permisos para un recurso crítico	5. Incorrecta implantación de la aplicación 6. Configuración ineficiente del Servidor Web	9.00	7
7. Almacenamiento Criptográfico Inseguro	[8] CWE-311. Falta de cifrado de datos sensitivos [19] CWE-327. Uso de un algoritmo criptográfico riesgoso o vulnerable [25] CWE-759. Uso de un Hash de un solo sentido sin sal	4. Revelación de información 10. Cifrado no estándar	12.16	4
8. Falla de Restricción de Acceso a URL	[6] CWE-862. Falta de autorización [15] CWE-863. Autorización incorrecta	9. Control de acceso incorrecto	9.50	6
9. Falta de protección en la Capa de Transporte	[8] CWE-311. Falta de cifrado de datos sensitivos	1. Uso incorrecto de SSL	6.00	10
10. Redirecciones y reenvíos no validados	[22] CWE-601. Direccionamiento abierto	9. Control de acceso incorrecto	13.66	2
Nuevos "eventos de riesgo"	**"Vulnerabilidades" SANS**	**"Vulnerabilidades" X-Force**		
Control inadecuado de entradas	[23] CWE-134. Formato de cadenas no controlado [10] CWE-807. Confiar en entradas no confiables en una decisión de seguridad	7. Control inadecuado de entradas	13.33	3
Control inadecuado de los procesos	[3] CWE-120. Desbordamiento clásico de Buffer [18] CWE-676. Uso de funciones potencialmente peligrosas [20] CWE-131. Calculo incorrecto del tamaño de un buffer [24] CWE-190. Desbordamiento de un *Integer* o su reciclado	5. Incorrecta implantación de la aplicación	14.00	1
Control inadecuado de salidas		2. Falta de control en la información presentada en los mensajes de error 4. Revelación de información	3.00	13

Tabla 3.5 –Eventos de riesgo en las aplicaciones financieras

[8] Cabe mencionar, que entre las diferentes fuentes de información, se manejan diversas interpretaciones sobre las definiciones formales de "riesgos" y "vulnerabilidades", por lo que se solicita al lector tenerlo en consideración.

3.4.2 Riesgos y ataques financieros.

Volviendo al sector financiero, a continuación se presentará un mapeo aproximado entre ataques financieros y sus posibles causas, teniendo por objetivo, mostrar como los anteriores ataques sobre las aplicaciones financieras (Véase Figura 3.2), se canalizan a los eventos de riesgo identificados en el punto anterior por OWASP, SANS y X-Force. Lo cual finalmente es, la explotación exitosa de las vulnerabilidades presentes en los sistemas.

Ataques financieros	Eventos de riesgo
Pérdida de información de clientes / privacidad	Inyección Secuencia de Comandos en Sitios Cruzados Perdida de Autenticación y Gestión de Sesiones Referencia Directa Insegura a Objetos Falsificación de Peticiones en Sitios Cruzado Almacenamiento Criptográfico Inseguro Control inadecuado de entradas
Phising/pharming	Secuencia de Comandos en Sitios Cruzados Perdida de Autenticación y Gestión de Sesiones Falsificación de Peticiones en Sitios Cruzado Poca protección en la Capa de Transporte Redirecciones y reenvíos no validados
Uso inapropiado de datos	Inyección Referencia Directa Insegura a Objetos Falsificación de Peticiones en Sitios Cruzado Control inadecuado de salidas
Ingeniería social	Secuencia de Comandos en Sitios Cruzados Redirecciones y reenvíos no validados
Baja calidad del desarrollo del software	Control inadecuado de los procesos
Ataque vía e-mails (*Spam*)	Configuración Defectuosa de Seguridad
Gestión inadecuada de parches	Configuración Defectuosa de Seguridad
Robo de propiedad intelectual	Inyección Secuencia de Comandos en Sitios Cruzados Pérdida de Autenticación y Gestión de Sesiones Referencia Directa Insegura a Objetos Falsificación de Peticiones en Sitios Cruzado Almacenamiento Criptográfico Inseguro Control inadecuado de entradas Control inadecuado de salidas
Conducta inapropiada por parte de empleados	(Fuera del alcance de esta investigación)
Fraude financiero interno involucrando sistemas de información	Pérdida de Autenticación y Gestión de Sesiones Almacenamiento Criptográfico Inseguro
Spyware	Secuencia de Comandos en Sitios Cruzados Redirecciones y reenvíos no validados
Debilidad en pautas para crear contraseñas	Pérdida de Autenticación y Gestión de Sesiones
Ataques de virus/gusanos	Secuencia de Comandos en Sitios Cruzados Configuración Defectuosa de Seguridad
Redes zombi	Secuencia de Comandos en Sitios Cruzados Configuración Defectuosa de Seguridad Falta de protección en la Capa de Transporte
Ciber-terrorismo	(Fuera del alcance de esta investigación)
Acceso remoto malintencionado	Pérdida de Autenticación y Gestión de Sesiones Falla de Restricción de Acceso a URL Falta de protección en la Capa de Transporte

Alteración de páginas web	Secuencia de Comandos en Sitios Cruzados
Ataques de denegación de servicios	Inyección Secuencia de Comandos en Sitios Cruzados Configuración Defectuosa de Seguridad
Exposición de datos relevantes por vía web	Inyección Pérdida de Autenticación y Gestión de Sesiones Referencia Directa Insegura a Objetos Falsificación de Peticiones en Sitios Cruzados Almacenamiento Criptográfico Inseguro Control inadecuado de salidas
Fraude financiero externo a través de los sistemas de información	Inyección Secuencia de Comandos en Sitios Cruzados Pérdida de Autenticación y Gestión de Sesiones Referencia Directa Insegura a Objetos Falsificación de Peticiones en Sitios Cruzados Falla de Restricción de Acceso a URL Poca protección en la Capa de Transporte
Quebranto de redes inalámbricas	Poca protección en la Capa de Transporte
Adware	Configuración Defectuosa de Seguridad
Extorsión online	(Fuera del alcance de esta investigación)
Amenazas físicas	(Fuera del alcance de esta investigación)

Tabla 3.6 –Eventos de riesgo y ataques en las aplicaciones financieras

Analizando las tablas anteriores (Véase Tabla 3.5 y 3.6) se construyó la siguiente tabla (Véase Tabla 3.7), para identificar aquellos eventos de riesgo con mayor peso o nivel dado la siguiente fórmula:

$$Nivel\ de\ evento\ de\ riesgo = NV * NI * PA \quad (3.1)$$

Dónde:

NV: Es el número de veces (vulnerabilidades diferentes), que el evento fue considerado en OWASP, SANS y X-Force.

NI: Es el nivel de importancia de las vulnerabilidades asociadas a los eventos de riesgo.

PA: Es la participación de los eventos de riesgo en los diferentes tipos de ataques dentro del sector financiero.

Evento de riesgo	Número de veces considerado (NV) (Tabla 3.5)	Nivel de importancia (NI) (Tabla 3.5)	Participación en ataques (PA) (Tabla 3.6)	Nivel
3. Perdida de Autenticación y Gestión de Sesiones	5	8	8	320
6 Configuración Defectuosa de Seguridad	5	7	6	210
4. Referencia Directa Insegura a Objetos	7	5	5	175
9. Falta de protección en la Capa de Transporte	3	10	5	150
1. Inyección	2	12	6	144
2. Secuencia de Comandos en Sitios Cruzados	1	11	10	110
7. Almacenamiento Criptográfico Inseguro	6	4	4	96
13. Control inadecuado de salidas	2	13	3	78
5. Falsificación de Peticiones en Sitios Cruzados	1	9	6	54
8. Falla de Restricción de Acceso a URL	4	6	2	48

10. Redirecciones y reenvíos no validados	3	2	3	18
11. Control inadecuado de entradas	3	3	2	18
12. Control inadecuado de los procesos	5	1	1	5

Tabla 3.7 –Nivel de los eventos de riesgo en las aplicaciones financieras

De esta manera, las anteriores tablas nos confirman que, si podemos eliminar o disminuir las vulnerabilidades en una aplicación web a través del control de sus riesgos, entonces nos estaremos protegiendo contra la mayoría de los ataques dirigidos hacia las aplicaciones financieras. Para aquellos casos en los cuales no se mapeo el ataque hacia uno de los riesgos presentados, significa que estos se subsanan mediante la aplicación de controles físicos y políticas de seguridad.

3.5 Controles de seguridad

Para mitigar y/o eliminar los riesgos identificados en la sección anterior, será necesario implementar una serie de controles que incluyen a políticas, procedimientos, guías, prácticas o estructuras organizacionales [40], las cuales pueden ser de naturaleza administrativa, técnica, legal o física para proteger a un costo razonable los sistemas de información de la empresa.

Estos controles, no necesariamente se deben de implementar durante la codificación o implantación del sistema, sino durante todo el ciclo de vida, tal como lo presenta Gartner [51] en la figura siguiente, donde nos muestra los tipos de control sugeridos a implementar para cada etapa de desarrollo del software.

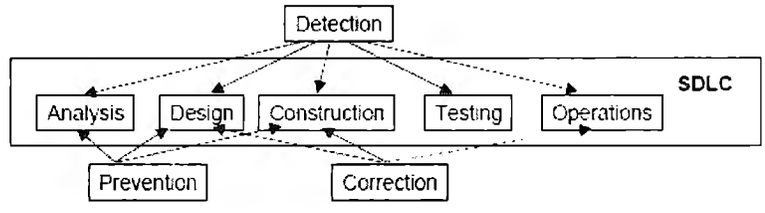


Figura 3.3 – Tipos de controles de seguridad en el SDLC

3.5.1 Controles para los riesgos financieros.

Conforme al "Top 10" de OWASP [55] y las medidas de prevención y mitigación propuestos por SANS y MITRE [53], de forma general los mecanismos mínimos de protección que nos sugieren aplicar para subsanar cada uno de los riesgos identificados, son los siguientes: [9]

[9] Para mayor información consultar la guía "Top 10" de OWASP sobre riesgos en las aplicaciones web y la lista de "Errores de Software más Peligrosos" de SANS y MITRE.

Eventos de riesgo	Controles OWASP, SANS y MITRE	Controles computacionales	Actividades administrativas
1. Inyección	Mantener los datos de entrada no confiables separados del uso de comandos y consultas. Utilizar una API segura que evite completamente el uso del intérprete de comandos, queries o provea una interface parametrizada, que solo acepte ciertos tipos de valores. Detener los caracteres especiales utilizando una sintaxis de escape especial para dicho interprete Usar una validación positiva de entradas con una apropiada canonicalización.	Separación de obligaciones Validación de parámetros	
2. Secuencia de Comandos en Sitios Cruzados (XSS)	Mantener los datos de entrada no confiables separados del contenido activo del navegador. Detener todos los datos no confiables basados en el contexto HTML (cuerpo, atributo, JavaScript, CSS, o URL) donde los mismos serán ubicados. Validación de entradas positiva o *whitelist* con apropiada canonicalización y decodificación. Además, decodificar cualquier entrada codificada, y luego validar la longitud, tipos de caracteres, formato, y reglas de negocio en dichos datos antes de aceptar la entrada.	Separación de obligaciones Validación de datos de entrada	
3. Pérdida de Autenticación y Gestión de Sesiones	Facilitar a los desarrolladores un único conjunto de controles de autenticación fuerte y de gestión de sesiones Evitar vulnerabilidades de XSS	Manejo de sesiones Mecanismos de autenticación Validación de datos de entrada	
4. Referencia Directa Insegura a Objetos	Proteger los objetos accesibles por cada usuario. Utilizar referencias indirectas por usuario o sesión. Comprobar el acceso.	Manejo del procesamiento interno Mecanismo de autorización	
5. Falsificación de Peticiones en Sitios Cruzados (CSRF)	Incluir un testigo no predecible en el cuerpo, o URL, de cada petición HTTP. Dicho testigo debe ser, como mínimo, único por cada sesión de usuario. Incluir el testigo en un campo oculto. Esto genera que el valor sea enviado en el cuerpo de la petición HTTP evitando su inclusión en la URL. El testigo único también puede ser incluido en la URL misma, o en un parámetro de la URL.	Manejo de sesiones	
6. Configuración Defectuosa de Seguridad	Establecer un proceso repetible que permita configurar, rápida y fácilmente, entornos asegurados. Este proceso debe ser automatizado.	Separación de obligaciones Infraestructura de actualización de software	Administración de la configuración Realización de auditorías

	Establecer un proceso para mantener y desplegar todas actualizaciones y parches de software de manera oportuna. Establecer una arquitectura robusta de la aplicación que provea una buena separación y seguridad entre los componentes. Considerar la realización periódica de exploraciones (*scan*) y auditorias para ayudar a detectar fallos en la configuración o parches faltantes. Para todos los datos sensibles considerar las amenazas que afectan a los datos y las cuales se quieran proteger (por ejemplo, ataques internos, usuarios externos) y asegurarse de que todos los datos estén cifrados de manera que se defienda de las amenazas.		
7. Almacenamiento Criptográfico Inseguro	Asegurar que las copias de seguridad almacenadas externamente estén cifradas, y las claves estén gestionadas y almacenadas de forma separada. Asegurar el uso adecuado de algoritmos estándares robustos, que las claves sean fuertes y que existe una gestión de claves adecuada. Asegurar que las contraseñas se almacenan en forma de hash con un algoritmo estándar robusto y con sal. Asegurar que todas las claves y contraseñas son protegidas contra acceso no autorizado.	Manejo de algoritmos criptográficos Manejo de respaldos Administración de contraseñas	Análisis de riesgos
8. Falla de Restricción de Acceso a URL	Planificar un método que requiera autenticación y autorización adecuadas para cada página el cual puede darse por uno o más componentes externos al código de la aplicación. La autenticación y autorización estén basadas en roles. Políticas configurables. La implementación del mecanismo debería negar todo acceso por defecto, requiriendo el establecimiento explícito de permisos a usuarios y roles por cada página. Si la página forma parte de un proceso de varios pasos, verifique que las condiciones de la misma se encuentren en el estado apropiado para permitir el acceso.	Mecanismos de autenticación Mecanismos de autorización Mecanismos de identificación	
9. Protección Insuficiente en la Capa de Transporte	Requerir SSL para todas las páginas sensibles. Las peticiones sin SSL a estas páginas deben ser redirigidas a las páginas con SSL. Establecer el atributo *secure* en todas las cookies sensibles. Configurar el servidor SSL para que acepte únicamente algoritmos considerados fuertes. Verificar que el certificado sea válido, no se encuentre expirado o revocado y que se ajuste a todos los dominios utilizados por la aplicación	Seguridad en las comunicaciones Manejo de algoritmos criptográficos Implementación de certificados	Administración de certificados

10. Redirecciones y reenvíos no validados	Conexiones a sistemas finales (*back-end*) y otros sistemas también deben utilizar SSL u otras tecnologías de cifrado Evitar el uso de redirecciones y reenvíos. Si se utiliza redirecciones y reenvíos, no involucrar parámetros manipulables por el usuario para definir el destino Si los parámetros de destino no pueden evitarse, asegúrese de que el valor facilitado es válido y autorizado para el usuario. Se recomienda que el valor de cualquier parámetro de destino sea un valor de mapeo, en lugar de la dirección, o parte de la dirección, de la URL y en el código del servidor traducir dicho valor a la dirección URL de destino.	Manejo del flujo de procesamiento
11. Control inadecuado de entradas	Utilizar una API o marco de trabajo que provea de métodos para validar los datos de entrada. Revisar que las validaciones que se hagan en el cliente, se realicen de igual manera en el lado del servidor. Asegurarse que las funciones usen cadenas estáticas.	Validación de datos de entrada Validación de parámetros
12. Control inadecuado de los procesos	Usar un lenguaje de programación que no permitan o manejen el desbordamiento de memoria o de buffers. Identificar las funciones no seguras del API y no hacer uso de las mismas. Uso de buenas prácticas de programación.	Manejo del procesamiento interno
13. Control inadecuado de las salidas	Implementar un esquema de manejo de errores. No proporcionar información detallada de errores al usuario.	Manejo de errores Validación de información de salida

Tabla 3.8 –Controles a aplicar para la mitigación de riesgos

3.5.2 Controles para los requerimientos financieros.

Como se ha comentado durante el capítulo 2. "Estado del Arte", las aplicaciones financieras no solo deben establecer aquellos controles que les permitan amortiguar los riesgos sufridos por estas instituciones, sino también, implementar aquellos mecanismos dictados por la normatividad aplicable al sector financiero mexicano y aquella correspondiente con el giro de la institución. De esta manera, al analizar los requerimientos de seguridad, se identificaron los siguientes controles y actividades los cuales permiten satisfacer a los mismos.

Requerimientos de seguridad	Controles computacionales	Actividades administrativas
Administración de certificados	Implementación de certificados	Administración de certificados
Administración de contraseñas	Administración de contraseñas	Administración de contraseñas
Administración de planes de contingencia		Plan de contingencia
Administración de riesgos		Análisis de riesgos
Asignación de roles y responsabilidades		Definición de roles
Autenticación de la propia institución	Mecanismos de autenticación	
Autenticación de los usuarios	Mecanismos de autenticación	
Autenticación del personal facultado para realizar operaciones	Mecanismos de autenticación	
Clasificar información		Clasificación de la información
Clasificar tipos de riesgos		Análisis de riesgos
Conservar la información	Manejo de respaldos	Políticas de respaldo
Constancia electrónica auditable de las incidencias	Manejo de bitácora	
Constancia electrónica auditable de las operaciones	Manejo de bitácora	
Contar con políticas de privacidad		Políticas de privacidad
Contar con políticas de seguridad		Políticas de seguridad
Contar con políticas para la destrucción de información		Políticas para limpieza de información
Contar con un presupuesto eficiente de TI		Presupuesto de TI
Contar con una infraestructura de seguridad	Crear una infraestructura de seguridad	
Contar con una infraestructura de TI adecuada a las funciones	Crear una infraestructura de seguridad	
Controles para actualizar y acceder a la información	Mecanismos de autorización	
Detección de faltantes	Mecanismos de detección y notificación de anomalías	
Detección y notificación de delitos o uso indebido de la información	Mecanismos de detección y notificación de anomalías	
Documentación de requerimientos		Documentación de requerimientos
Documentación operativa de procesos y herramientas		Documentación operativa de procesos y herramientas
Documentación técnica del sistema		Documentación del sistema
Garantizar disponibilidad de la información	Crear una infraestructura de seguridad	Definición de controles de seguridad

Garantizar disponibilidad de la información por parte de terceros		Contratos Definición de controles de seguridad
Garantizar disponibilidad de los servicios	Crear una infraestructura de seguridad	Definición de controles de seguridad
Garantizar disponibilidad de los servicios por parte de terceros		Contratos Definición de controles de seguridad
Garantizar el no repudio	Mecanismo de identificación Mecanismos de autenticación	Definición de controles de seguridad
Garantizar la confidencialidad de la información	Mecanismos de autenticación Mecanismos de autorización	Definición de controles de seguridad
Garantizar la confidencialidad de la información por parte de terceros		Contratos Definición de controles de seguridad
Garantizar la confidencialidad de la información transmitida	Seguridad en las comunicaciones	Definición de controles de seguridad
Garantizar la integridad de la información	Manejo de algoritmos criptográficos	Definición de controles de seguridad
Garantizar la integridad de la información por parte de terceros		Contratos Definición de controles de seguridad
Garantizar tiempo de respuesta en la divulgación de información	Crear una infraestructura de seguridad	Definición de controles de seguridad
Independencia operativa con otros sistemas	Crear una infraestructura de seguridad	
Integración segura con otros sistemas	Crear una infraestructura de seguridad	
Involucrar a la alta dirección		Gobierno de seguridad
Manejo de contratos de servicios con el usuario		Contratos
Manejo de contratos de servicios con terceros		Contratos
Manejo de niveles de autorización y acceso al sistema	Mecanismos de autorización	
Manejo de respaldos de la información	Manejo de respaldos	
Manejo de sesiones	Manejo de sesiones	
Medidas para prevenir accesos no autorizados	Mecanismos de autenticación Mecanismos de autorización	
Monitoreo y control de riesgos	Mecanismos de detección y notificación de anomalías	
Pruebas al sistema por personal diferente al de desarrollo		Pruebas al sistema
Pruebas o evaluación de los controles de seguridad		Pruebas al sistema
Pruebas para la evaluación de vulnerabilidades		Pruebas al sistema
Realización de auditorias		Realización de auditorías
Recuperación de información	Manejo de respaldos	
Restablecimiento de operaciones		Plan de contingencia
Uso de licencias		Políticas de manejo de licencias Contratos
Verificar que los sistemas cumplan con los requerimientos		Pruebas al sistema

Tabla 3.9 –Controles a aplicar para satisfacer los requerimientos de seguridad

3.6 Conclusiones

Recapitulando, en el presente capitulo hemos revisado algunos de los conceptos más importantes y básicos en torno a la seguridad. Además, bajo un enfoque de administración de riesgos, se ha realizado un estudio general sobre las aplicaciones financieras. Lo anterior nos ha llevado por lo consiguiente a identificar, los controles de seguridad mínimos que nuestros sistemas requieren para minimizar o eliminar el riesgo de éstos, debido en gran medida a las vulnerabilidades inmersas en las aplicaciones.

A grandes rasgos este capítulo nos permitió observar, como la mayoría de los ataques hacia las aplicaciones web se dirigen y tienen éxito, debido a la existencia de errores o vulnerabilidades dentro del sistema, las cuales permiten llevar a cabo la actividad criminal. Estas vulnerabilidades tienen principalmente su origen durante el diseño y codificación de los sistemas, así como también en la configuración del ambiente de los mismos.

Por lo tanto, si logramos eliminar o disminuir las vulnerabilidades en una aplicación, entonces nos estaremos protegiendo de la mayoría de los ataques hacia las mismas. En un sentido más específico nos estaremos protegiendo de los ataques hacia las aplicaciones financieras.

Es por ello, que no solo los especialistas en seguridad, sino también los analistas de requerimientos, los arquitectos, los diseñadores, programadores, los especialistas en pruebas y de la configuración del sistema, deben aceptar su responsabilidad para producir y mantener la seguridad del software. De esta manera, deben de estar preparados para: modelar ataques, reconocer las vulnerabilidades las cuales hacen susceptible al software de ser comprometido por estas actividades, identificar las arquitecturas y diseños débiles que exponen a las mismas, así como especificar, diseñar e implementar software no dispuesto a errores [45].

Ahora bien, después de haber identificado las vulnerabilidades más comunes en las aplicaciones web, la segunda tarea fue reconocer aquellos eventos de riesgos presentados en las mismas, tomando en cuenta además, los ataques en los que participan.

Al analizar los riesgos, se asociaron aquellos controles necesarios para minimizar o eliminar los mismos. En general, se trató de identificar los controles computacionales y actividades administrativas capaces de mitigar estos riesgos, sin embargo, es necesario comentar que adicionalmente a ellos, se deben de implementar controles de alto nivel como son las políticas de seguridad así como promover una cultura de desarrollo la cual permita programar

defensivamente, de tal forma que el software siempre opere correctamente bajo condiciones normales, anómalas u hostiles.

Finalmente, en adición a los controles seleccionados para la mitigación de riesgos, se identificaron aquellos que permitan satisfacer los requerimientos en cuanto a seguridad de la información que las instituciones financieras deben de cumplir conforme a la normatividad dictada para este sector. De esta manera, al integrar ambas perspectivas, los controles generales identificados son los siguientes:

Controles computacionales	Controles Administrativos
Administración de contraseñas	Administración de certificados
Crear una infraestructura de seguridad	Administración de contraseñas
Implementación de certificados	Administración de la configuración
Infraestructura de actualización de software	Análisis de riesgos
Macanismo de identificación	Clasificación de la información
Manejo de algoritmos criptográficos	Contratos
Manejo de bitácora	Definición de roles
Manejo de errores	Documentación de requerimientos
Manejo de respaldos	Documentación del sistema
Manejo de sesiones	Documentación operativa de procesos y herramientas
Manejo del flujo de procesamiento	Gobierno de seguridad
Manejo del procesamiento interno	Plan de contingencia
Mecanismo de autorización	Políticas de manejo de licencias
Mecanismos de autenticación	Políticas de privacidad
Mecanismos de detección y notificación de anomalías	Políticas de respaldo
Mecanismos de identificación	Políticas de seguridad
Seguridad en las comunicaciones	Políticas para limpieza de información
Separación de obligaciones	Presupuesto de TI
Validación de datos de entrada	Pruebas al sistema
Validación de parámetros	Realización de auditorías
Validación de información de salida	

Tabla 3.10 – Resumen de controles de seguridad a aplicar en las aplicaciones financieras

CAPÍTULO 4

4 Patrones de Seguridad

Como se analizó durante el capítulo 2. "Estado del Arte", actualmente las guías existentes para el desarrollo de software seguro dentro de las instituciones financieras (la normatividad interna del Banco de México, y los manuales en tecnología del FFIEC), contemplan solamente actividades de seguridad a nivel general. Es así como las instituciones financieras normalmente hacen uso de otros estándares internacionales como NIST SP800-64, ISO/IEC 27002:2005 y los diferentes proyectos de la OWASP, con el fin de implementar controles de seguridad en sus sistemas de información.

Sin embargo, al analizar las mismas, nos dimos cuenta que estos estándares carecen de prácticas técnicas para la incorporación de controles de seguridad durante el diseño y el desarrollo del software. Lo cual se vuelve un factor de riesgo, dado que alrededor del 90% de las vulnerabilidades detectadas en las aplicaciones web, son causadas por errores introducidos en éstas fases [9].

La anterior problemática también se acrecienta, debido a la filosofía actual de desconexión entre los profesionales de seguridad y los desarrolladores de sistemas; tal como se vio durante la sección 1.2 "Definición del problema". En la cual, los profesionales de seguridad se concentran principalmente en la implementación de controles (externos) de seguridad, y los desarrolladores en cómo construir un sistema que funcione [9].

Conforme a este último punto, ha venido ganando terreno un nuevo tipo de prácticas denominadas "patrones de seguridad", las cuales son soluciones probadas a un problema recurrente de seguridad en la información [11]. Permitiendo así, acercar la experiencia de los ingenieros de seguridad al desarrollo de los sistemas de información. Y las cuales además, ofrecen guías para el diseño de software seguro, complementando así, los estándares anteriormente estudiados.

De esta manera, en el presente capítulo se analizarán los patrones de seguridad existentes, los cuales nos permitirán complementar las metodologías de desarrollo de sistemas, así como también, satisfacer los controles de seguridad que las aplicaciones financieras requieren.

4.1 Características

Comenzaremos por definir que un patrón de seguridad [29], describe un problema recurrente y particular de seguridad, el cual se da en un contexto específico, presentando una solución genérica "bien probada" a éste. La solución consiste en un conjunto de roles interactuando entre sí, los cuales son dispuestos en múltiples y concretas estructuras de diseño, o bien describen un proceso para crear una estructura en especial.

De forma concreta, los patrones de seguridad son patrones de software o soluciones genéricas que permiten ofrecer servicios de seguridad [15] (Véase Figura 4.1).

Figura 4.1 – Desarrollo de los patrones de seguridad

¿Y por qué utilizar patrones de seguridad en el desarrollo de sistemas?

Los patrones de seguridad, son vistos como un medio para superar la brecha actual entre los profesionales de seguridad y los desarrolladores de sistemas. Su objetivo es capturar la experiencia sobre seguridad en la forma de soluciones probadas a problemas recurrentes. Así, su intención es que sean usados y entendidos por los desarrolladores, quienes no necesariamente son profesionales de la seguridad. Adicionalmente, capturan las fortalezas y debilidades de la propuesta ofrecida, con el fin de permitir a los desarrolladores tomar una decisión informada entre la aplicación de la seguridad y otros de los requerimientos. [11]

Entre los beneficios obtenidos al utilizar patrones de seguridad, tenemos que [29]:

- Codifican el conocimiento básico de seguridad de una manera estructurada y entendible.
- La representación de los patrones es familiar a los desarrolladores de software.
- Dado que los patrones son usados actualmente para capturar el conocimiento organizacional y de ingeniería de sistemas, al aplicarlos para registrar la experiencia en seguridad, entonces por consiguiente, mejoran la integración de ésta dentro de los sistemas y la empresa.
- Permite llevar la seguridad a términos empresariales y de arquitectura del sistema, y no solo a nivel de implementación.

Además, para considerar una solución como un patrón, debe cumplir con ciertos requisitos como: haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores, y ser reusable o aplicable a diferentes problemas de diseño en distintas circunstancias. Estas características permiten garantizar que el patrón a implementar, ayudará a la solución de nuestro problema de seguridad.

Dado la popularidad de los patrones de diseño en la ingeniería de software, la inclinación natural cuando escuchamos este término, es asumir que los patrones de seguridad debieran usar diagramas UML e incluir código fuente de ejemplo. Mientras es verdad que diversos patrones pueden ser representados de esta manera, existen otros patrones los cuales no comprenden estas características [11].

De esta manera, para unificar y facilitar el uso de los patrones de seguridad, los diversos autores han estructurado los mismos en aspectos como el nombre del patrón, el problema y su solución. Estas plantillas son normalmente diseñadas bajo la ideología de cada autor; sin embargo, los puntos comúnmente manejados, son los siguientes [15].

- **Nombre del patrón.** Captura la esencia del patrón de forma concisa, y de ser posible de manera atrayente.
- **Intención.** Describe lo que hace el patrón, y los temas particulares de diseño al cual se dirige.
- **Contexto.** Describe el contexto del problema.
- **Sinopsis.** Resume el patrón brevemente, en dos a tres enunciados, incluyendo su propósito.
- **Alias.** Enumera otros nombres para el patrón, comprendiendo a los nombres por los cuales se le conoce en otra literatura.
- **Problema.** Describe las condiciones que motivan el uso del patrón.
- **Solución.** Describe a alto nivel cómo el patrón resuelve el problema descrito.
- **Estructura estática.** Presenta los elementos constituyentes involucrados en el uso de este patrón.
- **Estructura dinámica.** Esboza las relaciones entre los diversos componentes de la estructura estática.
- **Cuestiones de implementación.** Proveen de *tips* para la implementación en la forma de pistas detalladas y técnicas.
- **Ataques comunes.** Identifica los ataques que interactúan con el patrón.

- **Usos conocidos.** Cita ejemplos conocidos y usados actualmente para este patrón.
- **Código de ejemplo.** Código utilizado por los desarrolladores para ligar directamente el mismo dentro de la aplicación y empezar a usarlo inmediatamente.
- **Consecuencia.** Describe el posible impacto del uso del patrón con respecto a varios requerimientos funcionales y no-funcionales.
- **Aplicabilidad.** Determina si un patrón es aplicable a un sistema.
- **Delimitaciones.** Contiene las condiciones globales a aplicar con el fin que el patrón logre su meta.

4.1.1 Ejemplos de patrones de seguridad

Con el fin de visualizar cómo son representados los patrones de seguridad en la literatura, se presenta a continuación algunos ejemplos de ellos, mediante el uso de la plantilla común anteriormente expuesta. Esto permitirá comparar el tipo de información proporcionada, entre aquellos patrones que consideramos ofrecen prácticas técnicas (por ejemplo *Intercepting Validator* [57]) y aquellos que se muestran como guías prácticas (por ejemplo *Risk Determination* [29]).

4.1.1.1 Validador interceptor (Intercepting Validator)

Nombre del patrón: Validador interceptor (*Intercepting Validator*)

Intención: Las fuerzas que impulsan el uso de este patrón son:

- Validar una amplia variedad de datos transmitidos por el cliente.
- Contar con un mecanismo común para validar varios tipos de datos.
- Agregar dinámicamente lógica de validación cuando sea necesario para mantener la aplicación segura contra nuevos ataques.
- Las reglas de validación deben de desacoplarse de la lógica de presentación.

Contexto: Diversas estrategias de ataque conocidas involucran comprometer al sistema mandando peticiones que contienen datos inválidos o código malicioso. Estos tipos de ataques requieren que la aplicación intercepte y limpie los datos antes de su uso.

Sinopsis: Limpia y valida los datos antes de su uso dentro de la aplicación, usando dinámicamente validación lógica descargable.

Alias: N/A

Problema: Se necesita un mecanismo simple y flexible para verificar y validar los datos transferidos por el cliente, de código malicioso o contenido malformado. Los datos pueden estar basados en formas, *queries* o con contenido XML.

Solución: Hace uso de un enfoque de filtros los cuales pueden ser aplicados declarativamente basados en una URL permitiendo así que diferentes peticiones sean mapeadas a diferentes cadenas de filtros. En este caso, el filtro pre-procesará las peticiones y contendrá lógica de validación que determina si la petición deberá o no de continuar. Además no se encuentra ligado a un conjunto particular de reglas de negocio pero las validaciones deberán aun así realizarse del lado del servidor.

Estructura estática: El diagrama de clases del patrón se muestra a continuación:

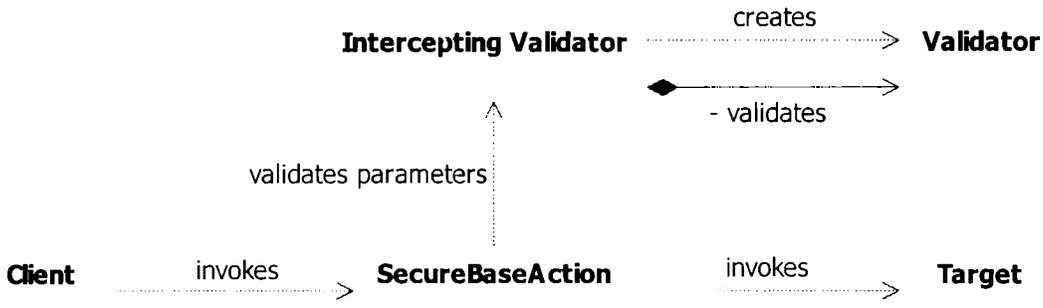


Figura 4.2 – Estructura estática del patrón *InterceptingValidator* [57]

Los participantes y las responsabilidades de estos son:

Client: Un cliente manda una petición a un recurso en particular.

SecureBaseAction: Es usado por el cliente para genéricamente hacer cumplir la validación de las peticiones en la capa web, delegando esta responsabilidad a *InterceptingValidator.*

InterceptingValidator: Es una versión especializada del patrón *InterceptingFilter*, con algunos cambios en la estrategia, como que es el único responsable de la validación de los datos.

Target: El recurso que el cliente requiere.

Validator: Se puede dividir en varios tipos:

- *ParamValidator:* Es responsable de validar todos los parámetros de peticiones; como: la validación de límites, formateo de datos y examinar los parámetros para vulnerabilidades de XSS, URLs malformadas. Las validaciones son específicas de acuerdo al *Target.*
- *SQLValidator:* Es responsable de validar los parámetros para ejecutar sentencias SQL. Examinar los parámetros sobre validaciones de límites, tamaño de datos y formato. Las validaciones son específicas de los *queries* de la base de datos y las transacciones.

Estructura dinámica: El diagrama de secuencia del patrón se muestra a continuación:

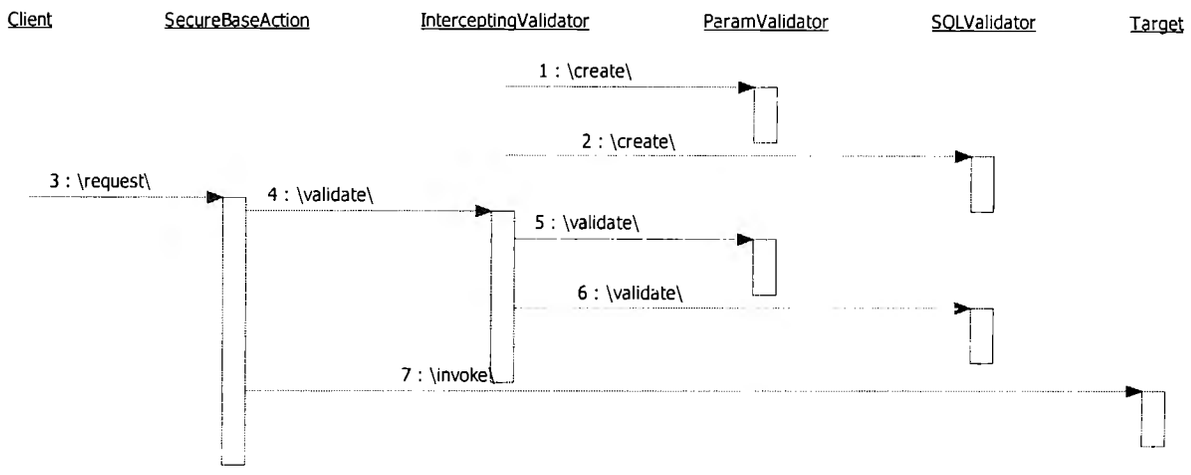


Figura 4.3 – Estructura dinámica del patrón *InterceptingValidator* [57]

Un ejemplo de un escenario de validación de datos malformados es el siguiente:

1. El *Client* realiza una petición a un recurso en particular, especificado como *Target*.
2. *SecureBaseAction* usa al *InterceptingValidator* para validar los datos para el servicio pedido al *Target*.
3. *InterceptingValidator* recupera los validadores apropiados de acuerdo a la configuración del *Target*.
4. *InterceptingValidator* invoca una serie de validaciones configuradas.
5. Cada *Validator* valida y limpia los datos pedidos, si es necesario.
6. Después de una validación exitosa, el *SecureBaseAction* invoca al recurso solicitado.

Cuestiones de implementación: Diferentes validadores serán usados para validar diferentes tipos de datos en una petición. En algunas ocasiones, la lógica de validación puede ser compleja, para lo cual se recomienda el uso de expresiones regulares.

Ataques comunes: Inyección de *scripts* maliciosos sentencias SQL, contenido XML y datos inválidos que hacen uso de campos de una forma que el atacante conoce que pueden ser insertados dentro de la aplicación para causar una falla potencial o denegación del servicio.

Usos conocidos: N/A

Código de ejemplo: Parte del código, usando el *framework* de *Struts*, para la clase *SecureBaseAction* se muestra a continuación:

```
public ActionErrors validate (ActionMapping acxtionmaping,
                              HttpServletRequest request){
   Validator validator = InterceptingValidator.getValidator (actionmapping);
   ValidationErrors errors = validator.process(request);
   If(errors.hasErrors())
      return InterceptingValidator.transformToActionErrors(errors);
   String externalizedProcessingKey = actionmapping.getParameter();
   ExternalizedValidator validatorEx =
   InterceptingValidator.getExternalizedValidator(externalizedProcessingKey)
;
   errors = validatorEx.process(request);
   if(errors.hasErrors())
      return InterceptingValidator.transformToActionErrors(errors);
   try {
      Class cls =
    InterceptingValidatorUtil.loadClass(externalizedProcessingKey);
      Method method =
    InterceptingValidatrUtil.getValidationActionMethod("process");
      InterceptingValidator.invoke(cls, method, new Object[] {request});
   }
   catch(Exception ex) {
      log("Invocation exception", ex);
      return InterceptingValidator.transformToActionErrors(ex);
   }
\
```

Consecuencia: Los beneficios que ofrece el patrón son:

- Centralizar las validaciones de seguridad.
- Desacoplar las validaciones de la lógica de presentación.
- Simplifica la adición de nuevos validadores.

Como factor de riesgo se encuentra la sobrecarga de procesos los cuales pueden causar que la aplicación falle si no se previenen desbordamiento de buffer o ataques de ciclos infinitos.

Aplicabilidad: N/A.

Delimitaciones: Se necesita un marco de trabajo para asegurar que exista un mecanismo para facilitar la adición fácil de nueva lógica de validación para ataques futuros sobre la aplicación.

4.1.1.2 Determinación de Riesgos (Risk Determination)

Nombre del patrón: Determinación de Riesgos (*Risk Determination*)

Intención: Permite evaluar y priorizar los riesgos de los activos.

Contexto: La empresa ha definido los activos a ser incluidos en la valoración de riesgos y ha evaluado la importancia de ellos en una tabla de valoración de activos. Además a ejecutado una valoración de amenazas y de vulnerabilidades, así como colectado combinaciones de amenazas y vulnerabilidades en una tabla.

Sinopsis: Es la fase final del proceso de valoración de riesgos, incorpora los resultados de la valoración de activos, de amenazas y de vulnerabilidades, para evaluar y priorizar los riesgos de los activos.

Alias: Evaluación de riesgos.

Problema: Una vez que se ha determinado el valor de los activos, así como las amenazas y vulnerabilidades que los afectan, se necesita determinar los riesgos involucrados. Necesitando así, una metodología formal para la determinación del riesgo que permita dar a conocer, si el esfuerzo a realizar para proteger los activos es demasiado alto o bajo.

Solución: Determinar sistemáticamente el riesgo que afecta a cada activo de la empresa. El cuál involucra los siguientes pasos:

- Colectar los resultados de la valoración de los activos, las amenazas y las vulnerabilidades.
- Relacionar pares de amenazas y vulnerabilidades con los activos.
- Evaluar el riesgo. Utilizar una ecuación utilizando los resultados numéricos de las tablas anteriores. El resultado presentara el riesgo final propuesto para cada activo.
- Presentar los resultados. Ordenar los resultados en orden descendiente.

Estructura estática: N/A

Estructura dinámica: La secuencia para ejecutar la determinación de los riesgos se muestra en la siguiente figura:

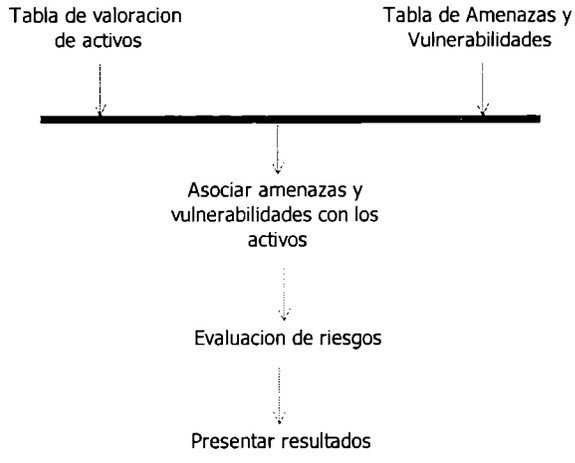


Figura 4.4 – Estructura dinámica del patrón *Risk Determination* [29]

- Colectar la valoración de las activos y la tabla de amenazas-vulnerabilidades.
- Usar una ecuación de riesgos para calcular el riesgo para cada activo.
- Ordenar y presentar los resultados en orden descendiente.

Cuestiones de implementación. Hacer uso de las tablas obtenidas de la valoración de activos, vulnerabilidades y amenazas. Para el cálculo del riesgo utilizar la ecuación:

$$Riesgo(A) = SUM[Amenazas * Vulnerabilidades](A) * ValorActivo(A) \quad (4.1)$$

Ataques comunes. N/A.

Usos conocidos. NIST 800-30, usa una matriz de 3x3 con la probabilidad de la amenaza y el impacto de ésta. Donde los valores cualitativos de la probabilidad de las amenazas (alto, medio, bajo) son convertidos en valores numéricos (1.0, 0.5, 0.1). Así como también los valores del impacto de la amenaza (alto, medio, bajo) son convertidos numéricamente (100, 50, 10). El riesgo es entonces calculado al multiplicar la probabilidad de la amenaza con su impacto.

Código de ejemplo. N/A.

Consecuencia. Los beneficios que ofrece son:

- La empresa es capaz de identificar y manejar los riesgos a sus activos para su mitigación.
- Los resultados cualitativos provistos son más fáciles de calcular, priorizar e interpretar.
- Los resultados pueden ser conseguidos y usados para rastrear el progreso del riesgo de los activo a través de valoraciones consecutivas de riesgos.

Las desventajas son:

- La ecuación de riesgo, podría no considerarse para todas las propiedades de la relación entre amenazas, vulnerabilidades y valor de los activos
- Los resultados son basados en la integridad y subjetividad de las valoraciones de activos, amenazas y riesgos.
- Dado la variedad de métodos para calcular el valor del riesgo, una empresa podría tener dificultades en identificar y aplicar una ecuación en particular.

Aplicabilidad: N/A.

Delimitaciones. N/A.

4.2 Revisión de patrones de seguridad existentes

Actualmente, podemos encontrar diversos estudios e investigaciones publicadas acerca de los patrones de seguridad [15], [58]. Sin embargo, dado las características críticas de las aplicaciones financieras, se decidió acotar a aquellas investigaciones que estuvieran respaldadas por organizaciones gubernamentales, o bien se hayan realizado publicaciones comerciales de las mismas. Bajo estos criterios, los estudios que se consideraron más significativos sobre patrones de seguridad, se describen a continuación.

4.2.1 Repositorio de patrones de seguridad

Este proyecto, respaldado por la Agencia de Proyectos de Investigación Avanzada para la Defensa de los Estados Unidos (DARPA, por sus siglas en ingles), ofrece una de las primeras investigaciones realizadas sobre patrones de seguridad en el desarrollo de aplicaciones web. Dicha investigación produjo un repositorio con 26 patrones y 3 mini-patrones de seguridad, mismos que se describen a continuación [11].

Nombre	Descripción
Patrones estructurales	Conjunto de patrones a implementarse en el producto final, normalmente incluyen diagramas de estructura y descripciones de la interacción.
Bloqueo de cuentas (*Account Lockout*)	Protege las cuentas de los clientes de ataques automáticos que adivinan contraseñas, al implementar un límite de intentos incorrectos antes de deshabilitarla.
Sesión autenticada (*Authenticated Session*)	Permite a un usuario de Web, acceder a múltiples páginas con acceso restringido sobre un sitio web, sin tener que reautentificarse en cada página requerida.
Almacenamiento de datos en el cliente (*Client Data Storage*)	Usa cifrado para almacenar de manera segura datos sensitivos o críticos en el cliente, necesarios para el correcto funcionamiento de la aplicación.
Filtros para las entradas del cliente (*Client Input Filters*)	Protege la aplicación de datos manipulados por clientes inseguros, protegiendo así contra clientes corruptos los cuales podrían causar que la aplicación se comporte de una manera no esperada e insegura.
Sesiones directas (*Directed Session*) (Mini-Pattern)	Asegura que los usuarios no sean capaces de brincar entre una serie de páginas web. Dónde, en lugar de que el sistema exponga múltiples URLs, mantenga la página actual sobre el servidor. Al garantizar el orden en el cuál las páginas son visitadas el desarrollador puede tener la tranquilidad que el usuario no minará o sorteará los puntos de seguridad.
Almacenamiento encriptado (*Encrypted Storage*)	Provee de una segunda línea de defensa contra el robo de datos en los servidores del sistema. Asegura que, aunque se robe información del servidor, lo datos más sensitivos permanecerán a salvo de "ojos entrometidos".
Implementación oculta (*Hidden Implementation*) (Mini-Pattern)	Limita la habilidad de un atacante para discernir el trabajo interno de una aplicación, al ocultar información que puede ser usada para comprometer a la aplicación.

Campo de minas *(Minefield)*	Permite engañar, detectar y bloquear a un atacante durante un intento de interrupción del servicio. Este patrón introduce agresivamente variaciones que podrían contrarrestar los conocimientos en seguridad de los atacantes y ayudar a su detección.
Lista negra de direcciones de red *(Network Address Blacklist)*	Permite rastrear las direcciones de red (direcciones IP) que son fuente de intentos de hackeo y otras actividades maliciosas. Donde cualquier petición proveniente de una dirección dentro de la lista negra es ignorada.
Partición de la aplicación *(Partitioned Application)*	Divide una aplicación larga y compleja en dos o más componentes simples. Cualquier privilegio peligroso es restringido a un único y más pequeño componente. Así, cada componente tiene puntos de seguridad manejables y fáciles de verificar que en una aplicación monolítica.
Autenticación con contraseñas *(Password Authentication)*	Protege contra contraseñas débiles, ataques de adivinación automática y mal manejo de contraseñas.
Propagación de contraseñas *(Password Propagation)*	Requiere que las credenciales individuales de autenticación dé un usuario sean verificadas por la base de datos antes que se de acceso a los datos del usuario.
Aserciones seguras *(Secure Assertion)*	Distribuye verificaciones específicas de la aplicación a través del sistema, mediante el uso de aserciones (*assertions*) seguras que mapean las aserciones convencionales a un sistema de detección de intrusiones.
Aislamiento de procesos en el servidor *(Server Sandbox)*	Construye un muro alrededor del servidor web con el fin de contener el daño que podría resultar de un error no descubierto en el servidor de software.
Proxy confiable *(Trusted Proxy)*	Provee de una interface segura al restringir el acceso a los recursos protegidos, limitando las operaciones que pueden ser ejecutadas o al limitar la vista del usuario a un subconjunto de datos.
Transacciones válidas *(Validated Transaction)* (Mini-Pattern)	Pone todas las validaciones de seguridad relevantes para una transacción específica dentro de una página solicitada. Así, los usuarios pueden navegar libremente a través de páginas, registrándose en diferentes secciones en cualquier orden elegido. La transacción se asegurará de la integridad de la información enviada.
Patrones de procedimiento	Conjunto de patrones que son usados para mejorar los procesos de desarrollo de software con seguridad crítica, los cuales normalmente impactan en la organización o administración del proyecto de desarrollo.
Construir el servidor desde cero *(Build the Server from the Ground Up)*	Entender la instalación por defecto del sistema operativo y las aplicaciones, simplificando la configuración, removiendo los servicios innecesarios e investigar los servicios vulnerables que son parte de la configuración del servidor web.
Escoger los productos correctos *(Choose the Right Stuff)*	Provee de una guía para seleccionar los componentes comerciales apropiados como componentes de seguridad, lenguajes y herramientas, que permiten construir componentes propios.
Documentar las metas de seguridad *(Document the Security Goals)*	Los desarrolladores deberán entender las metas generales de seguridad y los casos de negocio detrás de ellos. Si las metas de seguridad no están documentadas y diseminadas, interpretaciones individuales podrían acarrear políticas inconsistentes y mecanismos inapropiados.
Documentar la configuración del servidor *(Document the Server Configuration)*	Con el fin de ayudar a manejar la complejidad de la configuración del servidor web y las aplicaciones, los desarrolladores y administradores deberán documentar la configuración inicial y todas las modificaciones realizadas a los mismos.
Registrándose al validar fuera de banda (Enroll by Validating Out of Band)	Cuando los usuarios se registran a un sitio web o servicio, algunas veces es necesario validar su identidad usando un canal externo, tal como un correo, teléfono o cara a cara. Lo que permitirá instituir un secreto que puede ser usado para establecer la identidad durante el registro.
Registrándose usando una validación de una tercera parte *(Enroll using Third-Party Validation)*	Cuando el servicio de una tercera parte está disponible y es suficientemente confiable, la aplicación le puede delegar la tarea de autenticar la identidad del usuario.

Registrándose con un secreto compartido preexistente (*Enroll with a Pre-Existing Shared Secret*)	Cuando los usuarios se registran en un sitio web o servicio, algunas veces es suficiente con validar la identidad usando un secreto compartido preexistente. Lo cual permite registrarse sin una comunicación previa para establecer una cuenta.
Registrándose sin validación (*Enroll without Validating*)	Cuando se registran usuarios a un sitio web o servicio, algunas veces no es necesario validar la identidad del usuario. Sin embargo, si el usuario desea crear una cuenta, el sitio establece entonces credenciales de autenticación inicial, como un nombre de usuario y contraseña, los cuales representarán el secreto compartido a utilizar en visitas futuras.
Bitácora para auditoria (*Log for Audit*)	Vincula las bitácoras a la auditoría, asegurando que estas se encuentren configuradas con la auditoría en mente y que sea entendida como parte integral de un registro efectivo.
Parchar proactivamente (*Patch Proactively*)	En lugar de esperar a que el sistema sea comprometido, para entonces aplicar un parche de software de terceros, los administradores de sistemas deberán monitorear los parches para su aplicación inmediata.
Pruebas para atacar al diseño (*Red Team the Design*)	Afecta la evaluación de la seguridad de una aplicación al promover el uso de equipos rojos (los cuales examinan al sistema desde una perspectiva del atacante), durante las fases tempranas del desarrollo cuando es posible arreglar los problemas identificados.
Compartir la responsabilidad por la seguridad (*Share Responsibility for Security*)	Hace a todos los desarrolladores responsables por la seguridad en el sistema, quienes deben entender y manejar estos conceptos. De esta manera, se evita el problema de separación entre el equipo de seguridad y el de desarrollo.
Pruebas con un servidor de pruebas (*Test on a Staging Server*)	Mientras las pruebas unitarias pueden realizarse en las máquinas de desarrollo, una prueba de integral del sistema debe realizarse en máquinas similares a los servidores de producción. Lo cual también evita la configuración excesiva del sistema al momento de su puesta en producción.

Tabla 4.1 –Patrones de seguridad de DARPA

4.2.2 Guía técnica de patrones de diseño seguros

Esta guía técnica publicada por el Open Group, ofrece un catálogo de patrones así como una metodología de diseño seguro basada en éstos, permitiendo a los arquitectos y diseñadores de un sistema desarrollar arquitecturas seguras para soportar los requerimientos particulares. De esta manera, los capítulos que conforman esté reporte, describen la naturaleza y estructura de los patrones de diseño así como su uso, estos son [59]:

Nombre	Descripción
Patrones sobre disponibilidad del sistema	Conjunto de patrones de diseño estructurales los cuales facilitan la construcción de sistemas proveyendo acceso ininterrumpido a los servicios y recursos que se ofrecen a los usuarios.
Puesto de control en el sistema (*Checkpointed system*)	Permite estructurar un sistema de tal manera que su estado pueda ser recobrado o restaurado a un estado conocido válido en caso que un componente falle.
Reserva (*Standby*)	Permite estructurar un sistema de tal forma que el servicio provisto por un componente pueda ser reanudado con un componente diferente capaz de reanudar el mismo servicio que ofrecía el componente que falló.
Comparar-Verificar un sistema tolerante a fallas (*Comparator-checked fault-tolerant system*)	Permite que una falla independiente de un componente pueda ser detectada rápidamente para que un independiente y único componente al fallar, no cause la falla del sistema.

Sistema duplicado (*Replicated system*)	Construye un sistema el cual permite el suministro de servicios de múltiples puntos de presencia y recuperación, en caso de falla de uno o más componentes ligados.
Detección/corrección de errores (*Error detection/correction*)	Agrega redundancia a los datos para facilitar posteriores detecciones y recuperarse de errores.
Patrones de protección para los sistemas	Conjunto de patrones de diseño estructurales los cuales facilitan la construcción de sistemas que protejan los recursos sensitivos contra el uso no autorizado, su revelación o modificación.
Sistema protegido (*Protected system*)	Construir un sistema donde todos los accesos de los clientes a los recursos son dirigidos a un guardia el cual cumple una política de seguridad.
Política (*Policy*)	Aplicar una política aislada a un componente diferenciado de un sistema de información; asegurarse que las actividades para aplicar la política se ejecuten en una secuencia apropiada.
Descriptor de sujetos (*Subject descriptor*)	Provee acceso a los atributos relevantes de seguridad de una entidad a aquellas operaciones que serán ejecutadas.
Comunicación segura *(Secure Communication)*	Asegurar que los objetivos comunes en políticas de seguridad sean satisfechos cuando hay necesidad de que dos partes se comuniquen en la presencia de amenazas.
Contexto Seguro (*Security Context*)	Provee de un contenedor para los atributos de seguridad y datos relacionados a la ejecución de un contexto, proceso, operación o acción en particular
Asociación de seguridad (*Security Association*)	Define una estructura la cual provee a cada participante de una comunicación segura con la información que se usará para proteger los mensajes a trasmitir, y con la información que será usada para entender y verificar la protección aplicada al mensaje a recibir.
Proxy seguro (*Secure Proxy*)	Define la relación entre el guardia de dos instancias del Sistema protegido en el caso cuando una instancia esta enteramente contenida dentro de la otra.

Tabla 4.2 –Patrones de seguridad de Open Group

4.2.3 Patrones de Seguridad: Integrando la seguridad e Ingeniería de sistemas

El libro de patrones de seguridad, es una compilación de diversos patrones de seguridad, los cuales cubren diversas áreas de seguridad y cuyo objetivo es establecer una propuesta sobre la taxonomía de la arquitectura de seguridad en una empresa basándose en el marco de trabajo de Zachman. Para fines prácticos los autores agruparon los patrones de seguridad en diferentes categorías, las cuales corresponden a los capítulos 6 al 13 del libro [29]:

Nombre	Descripción
Seguridad de la empresa y administración de riesgos	Permite que los temas de seguridad se relacionen dentro de las funciones y misión de la empresa. El alcance de este conjunto de patrones incluye políticas, directivas o limitaciones a aplicar a todos los sistemas y las operaciones de la empresa.
Identificación de necesidades de seguridad para los activos de la empresa (*Security Needs Identification for Enterprise Assets*)	Este es el patrón clave para los asuntos de seguridad de la empresa. Ayuda a identificar donde se necesita seguridad, es decir, que propiedades de seguridad deben de aplicarse para una empresa en particular.

Valoración de activos (*Asset Valuation*)	Ayuda a determinar la importancia global de los activos que la empresa posee, si estos se pierden o son comprometidos y el costo de lo mismo.
Valoración de amenazas (*Threat Assessment*)	Identifica las amenazas sobre los activos de la empresa y determina la probabilidad o frecuencia de su ocurrencia.
Valoración de vulnerabilidades (*Vulnerability Assessment*)	Conducir una valoración de vulnerabilidades de la empresa ayuda a identificar las debilidades en los activos y los sistemas que permiten el acceso a ellos, así como evaluar la severidad si ésta es explotada.
Determinación de riesgos (*Risk Determination*)	Es la etapa final del proceso de valoración de riesgos, el cual incorpora los resultados de la valoración de activos, la valoración de amenazas y vulnerabilidades, para evaluar y priorizar los riesgos de los activos.
Enfoques de seguridad en la empresa (*Enterprise Security Approaches*)	Guía a una empresa en la selección de servicios de seguridad: prevención, detección y respuesta. Los cuales son dirigidos por las propiedades de seguridad que los activos requieren y dado la evaluación de los riesgos de seguridad.
Servicios de seguridad de la empresa (*Enterprise Security Services*)	Guía en la selección de servicios de seguridad para proteger los activos, después que se ha identificado el enfoque de seguridad. Permite establecer el grado de robustez y confianza que cada servicio debe ofrecer, basado en prioridades.
Comunicación con otras compañías (*Enterprise Partner Communication*)	Atención a la protección de los datos y los métodos por los cuales son transferidos, entre compañías.
Identificación y autenticación	Estos patrones ofrecen servicios de identificación y autenticación, enfocándose hacia la necesidad de reconocer a un actor y cómo este interactúa con el negocio del sistema.
Requerimientos de identificación y autenticación (*Identification & Authentication Requeriments*)	Un servicio de identificación y autenticación deberá satisfacer un conjunto de requerimientos sobre el servicio y la calidad del mismo. Su función es reconocer a un individuo y validar su identidad individual.
Alternativas de diseño automático de I&A (*Automated I&A Design Alternatives*)	Describe técnicas alternativas para una I&A automática. Ayuda a seleccionar una estrategia apropiada que consiste de una única técnica o combinación de técnicas para satisfacer los requerimientos de I&A.
Diseño y uso de contraseñas (*Password Design and Use*)	Describe mejores prácticas de seguridad para diseñar, crear, manejar y usar componentes de contraseñas para soportar los requerimientos de I&A.
Alternativas de diseño biométrico (*Biometrics Design Alternatives*)	Ayuda a la selección de mecanismos biométricos para satisfacer los requerimientos de I&A.
Modelos para el control de accesos	Se refiere a modelos de alto nivel que representan las políticas de seguridad de la empresa, los cuales definen las limitaciones sobre la seguridad a nivel de la arquitectura y de la aplicación, los cuales se hacen cumplir por niveles más bajos.
Autorización (*Authorization*)	Describe quien está autorizado a acceder a recursos específicos del sistema, en un ambiente en el cual tenemos recursos cuyo acceso necesita controlarse. Así indica para cada entidad activa, cuáles recursos puede acceder y cómo puede acceder.
Control de acceso basado en roles (*Role-Based Access Control*)	Describe cómo asignar derechos basados en las funciones o tareas de la gente en un ambiente en el cual el control de acceso a los recursos computacionales es requerido y donde hay un gran número de usuarios, o una gran variedad de recursos.
Seguridad multinivel (*Multilevel Security*)	Describe como categorizar información sensitiva y prevenir su revelación. Discute cómo asignar clasificaciones a usuarios (autorización), clasificaciones a los datos (niveles de sensibilidad), y separar diferentes unidades organizacionales en categorías.
Monitor de consulta (*Reference Monitor*)	Fuerza la declaración de restricciones de acceso cuando una entidad activa requiere recursos. Describe como definir un proceso abstracto que intercepta todas las peticiones para los recursos y checa su cumplimiento con autorizaciones.
Definición de derechos por rol (*Role Rights Definition*)	Provee de una forma, basada en casos de uso, para asignar derechos a los roles e implementar la política de "menor privilegio".

Arquitecturas para el control de acceso al sistema	Un servicio seguro para el control de acceso es esencial para los sistemas que permiten o deniegan su uso explícitamente. Para ello se presenta un conjunto de patrones que se complementan con la arquitectura de los sistemas de software para proporcionar un control de acceso basado en los requerimientos.
Requerimientos de control de acceso (*Access Control Requeriments*)	Provee de un conjunto genérico común de requerimientos de control de acceso. Los requerimientos dirigen tanto la función de control de acceso como las propiedades del servicio de control de acceso, así como la facilidad de uso y flexibilidad.
Punto de acceso único (*Single Access Point*)	Define un único punto de entrada que otorga o deniega la entrada al sistema después de corroborar el acceso requerido por el cliente.
Punto de chequeo (*Checked Point*)	Define un mecanismo de respuesta a intentos no autorizados de entrada.
Sesión segura (*Security Session*)	Para dar seguimiento a quien usa las funciones y sus correspondientes derechos de acceso, se establece una sesión segura después que el usuario se ha identificado.
Acceso total con errores (*Fuil Access with Errors*)	Provee de una vista de máxima funcionalidad al sistema, pero emite un error al usuario cuando trata de usar una función para la cual no está autorizado.
Límites de acceso (*Limited Access*)	Guía al desarrollador para presentar solo las funciones actualmente disponibles al usuario, mientras oculta todo lo demás para el cual le faltan permisos.
Controles para el acceso al sistema operativo	Patrones arquitectónicos para el control de acceso a los sistemas operativos. Asumen que los recursos se representan como objetos
Autenticador (*Authenticator*)	Maneja el problema de cómo verificar que un sujeto es quien dice ser. Recibe las iteraciones de un sujeto con el sistema y aplica un protocolo para verificar la identidad del sujeto.
Creador controlador de procesos (*Controlled Process Creator*)	Permite definir y garantizar derechos de accesos apropiados para un nuevo proceso.
Fabrica controladora de objetos (*Controlled Object Factory*)	Maneja cómo especificar los derechos de procesos con respecto a un nuevo objeto. Cuando un proceso crea un objeto a través de la fábrica, la petición incluye las características del nuevo objeto. Incluye una lista de derechos para acceder al objeto.
Monitor controlador de objetos (*Controlled Objected Monitor*)	Permite controlar el acceso por un proceso a un objeto. Usa un monitor de referencia para interceptar peticiones de acceso de los procesos. Checa si el proceso tiene el tipo requerido de acceso a el objeto.
Controlador de espacio de direcciones virtuales (*Controlled Virtual Address Space*)	Maneja cómo controlar el acceso por procesos a áreas específicas de su espacio de direcciones virtual de acuerdo a un conjunto predefinido de tipos de acceso.
Ejecución del dominio público (*Execution Domain*)	Define un ambiente de ejecución para los procesos indicando explícitamente todos los recursos que un proceso puede usar durante su ejecución, así como el tipo de acceso a los recursos.
Ambiente de ejecución controlado (*Controlled Execution Enviorenment*)	Define los derechos de un sujeto, y con ellos establecer los permisos hacia los procesos corriendo a cargo del sujeto. Previene que los procesos puedan hurgar información al buscar en la memoria y al acceder a las unidades de disco donde los archivos residen, o tomar control del sistema operativo.
Autorización de archivos (*File Authorization*)	Describe como controlar el acceso a archivos en el sistema operativo. Así describe el acceso a archivos por sujetos. El objeto protegido es un componente que puede ser un directorio o archivo.
Registro	Los eventos de seguridad son tomados como violaciones que ocurren durante las actividades operacionales. Los tomadores de decisiones deben estar conscientes de los eventos de seguridad que involucran a sus activos, esta seguridad es representada por la auditoria de seguridad y de registro.
Requerimientos de seguridad contable (*Security Accounting Requirements*)	Provee de un conjunto genérico de requerimientos de registro de seguridad para rastrear acciones o eventos relacionados con la seguridad.

Requerimientos de auditoría (*Audit Requirements*)	Comprende un conjunto genérico de requerimientos de auditoría y ayuda a priorizarlos. Lo cual incluye análisis de bitácora y rastreo de información de auditoria sobre un evento para encontrar o reportar alguna indicación de violaciones a la seguridad.
Requerimientos de rastreo de auditoria y bitácora (*Audit Trails and Logging Requirements*)	Provee de un conjunto genérico de requerimientos de auditoria trazable, ayudando a aplicarlos en situaciones específicas y a determinar su importancia relativa. Captura bitácoras de auditoría sobre eventos y actividades que ocurren dentro de una organización o sistema, para facilitar la reconstrucción y análisis de esos eventos y actividades.
Requerimientos de detección de intrusiones (*Intrusion Detection Requirements*)	Provee de un conjunto genérico de requerimientos de detección de intrusiones, ayudando a especificar los requerimientos que aplican a cada situación y su importancia relativa. Para automatizar el monitoreo de eventos sobre alguna indicación de violación a la seguridad.
Requerimientos de no repudiación (*Non-Repudiation Requirements*)	Provee de un conjunto genérico de requerimientos de no repudiación, ayudando a su aplicación y a determinar su importancia relativa. Captura y mantiene evidencia para que los participantes de una transacción o iteración no puedan negar que han participado en la actividad.
Arquitecturas con firewall	Existen diversos tipos de firewalls que representan relaciones entre complejidad, velocidad y seguridad, y las cuales son adaptadas para controlar ataques sobre capas específicas de la red. Estos patrones permiten elegir el tipo de firewall que mejor se ajuste al sistema.
Firewall para filtro de paquete (*Packet Filter Firewall*)	Filtra el tráfico de red que entra y sale en un sistema computacional basado en inspección de paquetes a nivel de IP.
Firewall basado en proxy (*Proxy-Based Firewall*)	Interpone un proxy entre las peticiones y el acceso, aplicando controles a través de este proxy. Así inspecciona y filtra el tráfico entrante y saliente de la red basado en el tipo de servicio de aplicación a ser accedido.
Firewall con estado (*Stateful Firewall*)	Filtra el tráfico de red entrante y saliente en un sistema computacional basado en el estado de la información dado comunicaciones pasadas. El estado de la información describe si el paquete entrante es parte de una nueva conexión o una comunicación continua aprobada previamente.
Aplicaciones seguras en internet	Estos patrones se especializan en las aplicaciones sobre internet enfocándose a diversos aspectos o funcionalidades que deben ser protegidas en una aplicación web.
Información anónima (*Information Obscurity*)	Si la información manejada por el sistema es sensitiva, debe ser protegida a través de obscurecer los datos a través de una forma de cifrado así como el ambiente alrededor de los datos.
Canales seguros (*Secure Channels*)	Para la comunicación sensitiva a través de la red pública, crear un canal seguro encriptado para asegurar la confidencialidad de los datos que transitan.
Socios conocidos (*Known Partners*)	Si las iteraciones comerciales son sensitivas o de alto valor, asegurar que el usuario con quien estamos interactuando es quien dice ser, así como proveer mecanismos que permitan identificar a nuestros sistemas.
Zona desmilitarizada (*Demilitarized Zone*)	Separa la funcionalidad del negocio y la información de los servidores web que la publican, colocando los servidores web en un área segura.
Protección de proxy inverso (Protection Reverse Proxy)	Escuda el servidor web de ataques a nivel de red. Además de proteger el software del servidor a nivel de protocolo de la aplicación.
Integración de proxy inverso (*Integration Reverse Proxy*)	Alivia la situación de obtener direcciones invalidas al cambiar la distribución de los servidores, al proveer una vista homogénea de colecciones de servidores sin conocer la distribución física de las maquinas individuales a los usuarios finales.
Puerta frontal (*Front Door*)	Implementa autenticación y autorización al implementar un servidor web de entrada para el *back-end*. Pudiendo incluso acceder a *back-ends* externos al proveer automáticamente el identificador del usuario y su contraseña del repositorio de contraseñas.

Tabla 4.3 –Patrones de seguridad de Integrando la seguridad e Ingeniería de sistemas

4.2.4 Patrones Centrales de Seguridad

Core Security Patterns, es un libro que sirve como guía para construir aplicaciones robustas de seguridad *end-to-end* para aplicaciones empresariales J2EE, servicios web, sistemas de administración de identidad y servicios que proveen soluciones. Su principal objetivo consiste en describir una metodología de diseño seguro, usando un conjunto de patrones de diseño reusables y probados para proporcionar seguridad a las aplicaciones J2EE [57].

Para ello, presenta un catálogo de 23 patrones de seguridad y 101 "mejores prácticas"; que identifican escenarios para casos de uso, modelos arquitectónicos, estrategias de diseño, tecnologías aplicadas y procesos de validación. Estos son:

Nombre	Descripción
Capa web	Los patrones de seguridad de esta capa permiten asegurar la comunicación entre el cliente y el servidor y de servidor a servidor, tanto en la infraestructura como en la aplicación
Autenticación forzosa (*Authentication Enforcer*)	Ilustra como un cliente basado en una aplicación J2EE deberá autenticarse con una aplicación J2EE.
Autorización forzosa (*Authorization Enforcer*)	Ilustra como la autorización debe ser forzada después de la autenticación del usuario con una aplicación J2EE
Validador interceptor (*Intercepting Validator*)	Ofrece mecanismos seguros para validar parámetros antes de invocar una transacción. La validación de los parámetros específicos de la aplicación incluye además la validación de los datos del negocio y sus características.
Acción base segura (*Secure Base Action*)	Es un patrón para centralizar y coordinar tareas relacionadas con la seguridad dentro de la capa de presentación. Sirve como el punto de entrada primario de la presentación y deberá ser usado o extendido por un "Controlador Frontal".
Bitácora segura (*Secure Logger*)	Define como capturar eventos específicos de la aplicación y excepciones de manera segura y confiable para soportar auditorías en seguridad.
Pipa segura (*Secure Pipe*)	Muestra cómo asegurar la conexión entre el cliente y el servidor o entre servidores cuando se conecta a otras instituciones. Agrega valor al requerir autenticación mutua y al establecer confidencialidad y no repudiación entre las partes.
Servicio proxy seguro (*Secure Service Proxy*)	Asegura y controla el acceso a los componentes J2EE expuestos como servicios web. Actúa como un proxy seguro al proveer una interfaz común al servicio subyacente provisto por el componente y al restringir el acceso directo al servicio web que provee del componente.
Administrador seguro de sesiones (*Secure Session Manager*)	Define como crear una sesión segura al capturar la información de la sesión. Usa el patrón de Pipa segura y describe las acciones requeridas para construir una sesión segura entre el cliente y el servidor o entre servidores.
Agente web interceptor (*Intercepting Web Agent*)	Ayuda a proteger aplicaciones web basadas en J2EE a través de un agente web que intercepta las peticiones del contenedor web y provee de autenticación, autorización, cifrado y capacidades de auditoría.
Capa de negocio	Los patrones de seguridad de esta capa soportan los servicios de seguridad del negocio, que normalmente corresponden a los datos y la lógica del negocio
Interceptor auditor (*Audit Interceptor*)	Trabaja junto con el patrón de Bitácora segura. Permite administrar y manejar aspectos de registro y auditoria en el *back-end*.

Administración segura del contenedor (*Container Managed Security*)	Describe cuándo y cómo declarar información relacionada con seguridad para los EJB en un *deployment descriptor*.
Administración dinámica de servicios (*Dynamic Service Management*)	Provee de una instrumentación ajustable dinámicamente de componentes de seguridad para monitorear y activar la administración de objetos de negocio.
Objeto de transferencia ofuscado (*Obfuscated Transfer Object*)	Describe formas de proteger los datos de negocio representados en objetos de transferencia y las cuales se pasan dentro y entre las capas lógicas.
Delegación de políticas (*Policy Delegate*)	Crea, maneja y administra políticas administrativas de seguridad que rigen cómo los objetos de la capa de EJB son accedidos y transferidos.
Fachada de servicios seguros (*Secure Service Façade*)	Provee de una fachada de sesión que puede contener y centralizar interacciones complejas entre componentes de negocio debajo de una sesión segura. Provee de seguridad dinámica y declarativa para los objetos de negocio del *back-end* en la fachada de sesión. Protege de entidades ajenas que puedan ejecutar directamente servicios ilegales o no autorizados.
Objeto seguro de sesión (*Secure Session Object*)	Define formas para asegurar la información de la sesión en los EJBs, facilitando el acceso distribuido y la propagación sin interrupciones de un contexto seguro.
Capa de servicios web	En esta capa los patrones de seguridad permiten establecer diferentes niveles de seguridad para los servicios web
Inspector de mensajes (*Message Inspector*)	Verifica la calidad de los mensajes XML para los mecanismos a nivel de seguridad, tales como Firmas XML, Cifrado XML, en conjunción con un *token* seguro. Además ayuda a verificar y validad los mecanismos de seguridad aplicada en mensajes SOAP cunado son procesados por múltiples intermediarios.
Portal interceptor de mensajes (*Message Interceptor Gateway*)	Provee de un único punto de acceso permitiendo la centralización de la aplicación de la seguridad para mensajes entrantes y salientes.
Enrutador seguro de mensajes (*Secure Message Router*)	Facilita la comunicación XML segura con múltiples puntos de salida que adoptan seguridad a nivel de mensajes y mecanismos de entidad federativa.
Capa de identidad	Dentro de esta capa los patrones de seguridad se encargan de la administración de la identidad y el abastecimiento de servicios.
Constructor de reafirmación (Assertion Builder)	Define como una reafirmación de identidad (por ejemplo reafirmación de autenticación o reafirmación de autorización) puede construirse.
Token para credenciales (*Credential Tokenizer*)	Describe como un *token* principal de seguridad puede ser encapsulado y embebido en un mensaje SOAP, enrutado y procesado.
Delegación de firma única (*Single Sign-on (SSO) Delegator*)	Describe como construir un agente para delegar el manejo de un sistema legado para una autenticación de una sola vez.
Sincronización de contraseñas (*Password Synchronizer*)	Describe como sincronizar principales de manera segura a través de múltiples aplicaciones usando la provisión de servicios.

Tabla 4.4 –Patrones de seguridad de Core Security Patterns

4.3 Patrones de seguridad para las aplicaciones financieras.

4.3.1 Clasificación de patrones de seguridad

Como se pudo observar en la sección 4.2 "Revisión de patrones de seguridad existentes", nos encontramos con diferentes tipos de patrones de seguridad así como diferentes formas de clasificar los mismos. Debido a ello, se decidió catalogar los patrones conforme a su aplicación en cada una de las etapas del ciclo de vida de desarrollo del sistema, tomando en cuenta las actividades identificadas durante la sección 2.2.3 "Actividades de seguridad identificadas dentro de los estándares de seguridad y guías de desarrollo" (Véase Tabla 2.16 del capítulo 2).

De esta manera, se pudo constatar que los patrones de seguridad principalmente se enfocan al establecimiento de guías y prácticas técnicas durante las primeras etapas del ciclo de vida de desarrollo de sistemas; especialmente para las fases de Análisis y Diseño de la aplicación. (Véase tabla 4.5)

Adicionalmente, los patrones estudiados se mapearon con los controles de seguridad identificados en el capítulo anterior 3. "Seguridad en las aplicaciones financieras", encontrando de esta manera, que la mayoría de los controles requeridos por las aplicaciones financieras pueden mapearse a uno o varios patrones de seguridad.

Etapa SDLC	Actividad	Fuente	Nombre	Nivel	Control Computacional	Control Administrativo
Análisis	Identificación de los requerimientos funcionales y de seguridad	Repositorio de patrones de seguridad	Documentar las metas de seguridad (*Document the Security Goals*)	Guía práctica		Políticas de seguridad Documentación de requerimientos
Análisis	Identificación de los requerimientos funcionales y de seguridad	Integrando la seguridad e Ingeniería de sistemas	Identificación de necesidades de seguridad para los activos de la empresa (*Security Needs Identification for Enterprise Assets*)	Guía práctica		Gobierno de seguridad Documentación de requerimientos
Análisis	Identificación de los requerimientos funcionales y de seguridad	Integrando la seguridad e Ingeniería de sistemas	Requerimientos de identificación y autenticación (*Identification & Authentication Requeriments*)	Guía práctica	Mecanismos de identificación Mecanismos de autenticación	Documentación de requerimientos
Análisis	Identificación de los requerimientos funcionales y de seguridad	Integrando la seguridad e Ingeniería de sistemas	Requerimientos de control de acceso (*Access Control Requeriments*)	Guía práctica	Mecanismos de autenticación Mecanismos de autorización	Documentación de requerimientos
Análisis	Identificación de los requerimientos funcionales y de seguridad	Integrando la seguridad e Ingeniería de sistemas	Requerimientos de seguridad contable (*Security Accounting Requirements*)	Guía práctica	Mecanismos de detección y notificación de anomalías Manejo de bitácoras	Documentación de requerimientos
Análisis	Identificación de los requerimientos funcionales y de seguridad	Integrando la seguridad e Ingeniería de sistemas	Requerimientos de auditoría (*Audit Requirements*)	Guía práctica	Mecanismos de detección y notificación de anomalías Manejo de bitácoras	Realización de auditorías Documentación de requerimientos
Análisis	Identificación de los requerimientos funcionales y de seguridad	Integrando la seguridad e Ingeniería de sistemas	Requerimientos de rastreo de auditoria y bitácora (*Audit Trails and Logging Requirements*)	Guía práctica	Manejo de bitácoras	Realización de auditorías Documentación de requerimientos
Análisis	Identificación de los requerimientos funcionales y de seguridad	Integrando la seguridad e Ingeniería de sistemas	Requerimientos de detección de intrusiones (*Intrusion Detection Requirements*)	Guía práctica	Mecanismos de detección y notificación de anomalías	Documentación de requerimientos
Análisis	Identificación de los requerimientos funcionales y de seguridad	Integrando la seguridad e Ingeniería de sistemas	Requerimientos de no repudiación (*Non-Repudiation Requirements*)	Guía práctica	Mecanismos de identificación	Documentación de requerimientos
Análisis	Identificación de los requerimientos funcionales y de seguridad	Integrando la seguridad e Ingeniería de sistemas	Definición de derechos por rol (*Role Rights Definition*)	Práctica Técnica	Definición de roles	Documentación de requerimientos
Análisis	Análisis de riesgos	Integrando la seguridad e Ingeniería de sistemas	Valoración de activos (*Asset Valuation*)	Guía práctica		Análisis de riesgos
Análisis	Análisis de riesgos	Integrando la seguridad e Ingeniería de sistemas	Valoración de amenazas (*Threat Assessment*)	Guía práctica		Análisis de riesgos

Análisis	Análisis de riesgos	Integrando la seguridad e Ingeniería de sistemas	Valoración de vulnerabilidades (Vulnerability Assessment)	Guía práctica		Análisis de riesgos
Análisis	Análisis de riesgos	Integrando la seguridad e Ingeniería de sistemas	Determinación de riesgos (*Risk Determination*)	Guía práctica		Análisis de riesgos
Análisis	Selección de controles	Integrando la seguridad e Ingeniería de sistemas	Enfoques de seguridad en la empresa (*Enterprise Security Approaches*)	Guía práctica		Definición de controles de seguridad
Análisis	Selección de controles	Integrando la seguridad e Ingeniería de sistemas	Servicios de seguridad de la empresa (*Enterprise Security Services*)	Guía práctica		Definición de controles de seguridad
Análisis	Selección de controles	Repositorio de patrones de seguridad	Registrándose al validar fuera de banda (Enroll by Validating Out of Band)	Guía práctica	Mecanismos de identificación	Administración de contraseñas Políticas de privacidad
Análisis	Selección de controles	Repositorio de patrones de seguridad	Registrándose usando una validación de una tercera parte (*Enroll using Third-Party Validation*)	Guía práctica	Mecanismos de identificación	Administración de contraseñas Políticas de privacidad
Análisis	Selección de controles	Repositorio de patrones de seguridad	Registrándose con un secreto compartido preexistente (*Enroll with a Pre-Existing Shared Secret*)	Guía práctica	Mecanismos de identificación	Administración de contraseñas Políticas de privacidad
Análisis	Selección de controles	Repositorio de patrones de seguridad	Registrándose sin validación (*Enroll without Validating*)	Guía práctica	Mecanismos de identificación	Administración de contraseñas
Análisis	Planeación de la seguridad	Repositorio de patrones de seguridad	Escoger los productos correctos (*Choose the Right Stuff*)	Guía práctica		Documentación operativa de procesos y herramientas
Análisis	Planeación de la seguridad	Repositorio de patrones de seguridad	Compartir la responsabilidad por la seguridad (*Share Responsibility for Security*)	Guía práctica		Gobierno de seguridad Políticas de seguridad
Análisis	Planeación de la seguridad	Integrando la seguridad e Ingeniería de sistemas	Comunicación con otras compañías (*Enterprise Partner Communication*)	Guía práctica	Crear una infraestructura de seguridad	Contratos
Diseño	Arquitectura de seguridad	Integrando la seguridad e Ingeniería de sistemas	Alternativas de diseño automático de I&A (*Automated I&A Design Alternatives*)	Guía práctica	Mecanismo de identificación Mecanismo de autenticación	
Diseño	Arquitectura de seguridad	Integrando la seguridad e Ingeniería de sistemas	Alternativas de diseño biométrico (*Biometrics Design Alternatives*)	Guía práctica	Crear una infraestructura de seguridad Mecanismo de identificación Mecanismo de autenticación	
Diseño	Arquitectura de seguridad	Repositorio de patrones de seguridad	Partición de la aplicación (*Partitioned Application*)	Guía práctica	Separación de obligaciones	
Diseño	Arquitectura de seguridad	Guía técnica de patrones de diseño seguros	Reserva (*Standby*)	Práctica técnica	Crear una infraestructura de seguridad Manejo de respaldos Mecanismos de detección y notificación de anomalías	Plan de contingencia

Diseño	Arquitectura de seguridad	Guía técnica de patrones de diseño seguros	Sistema duplicado (*Replicated system*)	Práctica técnica	Crear una infraestructura de seguridad Manejo de respaldos Mecanismos de detección y notificación de anomalías	Plan de contingencia
Diseño	Arquitectura de seguridad	Guía técnica de patrones de diseño seguros	Detección/corrección de errores (*Error detection/correction*)	Práctica técnica	Crear una infraestructura de seguridad Manejo de respaldos	Plan de contingencia
Diseño	Arquitectura de seguridad	Guía técnica de patrones de diseño seguros	Sistema protegido (*Protected system*)	Práctica técnica	Crear una infraestructura de seguridad Separación de obligaciones	
Diseño	Arquitectura de seguridad	Guía técnica de patrones de diseño seguros	Política (*Policy*)	Práctica técnica	Separación de obligaciones Manejo del flujo de procesamiento	
Diseño	Arquitectura de seguridad	Guía técnica de patrones de diseño seguros	*Comunicación segura (Secure Communication)*	Práctica técnica	Crear una infraestructura de seguridad Seguridad en las comunicaciones	
Diseño	Arquitectura de seguridad	Guía técnica de patrones de diseño seguros	Contexto Seguro (*Security Context*)	Práctica técnica	Crear una infraestructura de seguridad Mecanismo de autorización	
Diseño	Arquitectura de seguridad	Guía técnica de patrones de diseño seguros	Asociación de seguridad (*Security Association*)	Práctica técnica	Crear una infraestructura de seguridad Seguridad en las comunicaciones	
Diseño	Arquitectura de seguridad	Guía técnica de patrones de diseño seguros	*Proxy* seguro (*Secure Proxy*)	Guía práctica	Crear una infraestructura de seguridad Seguridad en las comunicaciones	
Diseño	Arquitectura de seguridad	Integrando la seguridad e Ingeniería de sistemas	Firewall para filtro de paquete (*Packet Filter Firewall*)	Práctica Técnica	Crear una infraestructura de seguridad Seguridad en la comunicaciones	
Diseño	Arquitectura de seguridad	Integrando la seguridad e Ingeniería de sistemas	Firewall basado en proxy (*Proxy-Based Firewall*)	Práctica Técnica	Crear una infraestructura de seguridad Seguridad en la comunicaciones	
Diseño	Arquitectura de seguridad	Integrando la seguridad e Ingeniería de sistemas	Firewall con estado (*Stateful Firewall*)	Práctica Técnica	Crear una infraestructura de seguridad Seguridad en la comunicaciones	
Diseño	Arquitectura de seguridad	Integrando la seguridad e Ingeniería de sistemas	Canales seguros (*Secure Channels*)	Práctica Técnica	Seguridad en las comunicaciones Manejo de algoritmos criptográficos	
Diseño	Arquitectura de seguridad	Integrando la seguridad e Ingeniería de sistemas	Zona desmilitarizada (*Demilitarized Zone*)	Práctica Técnica	Crear una infraestructura de seguridad Separación de obligaciones	
Diseño	Arquitectura de seguridad	Integrando la seguridad e Ingeniería de sistemas	Protección de proxy inverso (Protection Reverse Proxy)	Práctica Técnica	Crear una infraestructura de seguridad	
Diseño	Arquitectura de seguridad	Integrando la seguridad e Ingeniería de sistemas	Integración de proxy inverso (*Integration Reverse Proxy*)	Práctica Técnica	Crear una infraestructura de seguridad	
Diseño	Arquitectura de seguridad	Integrando la seguridad e Ingeniería de sistemas	Puerta frontal (*Front Door*)	Práctica Técnica	Crear una infraestructura de seguridad Mecanismo de identificación	

Diseño	Arquitectura de seguridad	Patrones Centrales de Seguridad	Pipa segura (*Secure Pipe*)	Práctica Técnica	Seguridad en las comunicaciones	
Diseño	Arquitectura de seguridad	Patrones Centrales de Seguridad	Servicio proxy seguro (*Secure Service Proxy*)	Práctica Técnica	Crear una infraestructura de seguridad	
Diseño	Arquitectura de seguridad	Patrones Centrales de Seguridad	Agente web interceptor (*Intercepting Web Agent*)	Práctica Técnica	Mecanismo de autenticación Mecanismo de autorización Manejo del flujo de procesamiento	
Diseño	Arquitectura de seguridad	Patrones Centrales de Seguridad	Delegación de políticas (*Policy Delegate*)	Práctica Técnica	Separación de obligaciones Manejo del flujo de procesamiento Manejo de procesamiento interno	
Diseño	Arquitectura de seguridad	Repositorio de patrones de seguridad	Sesión autenticada (*Authenticated Session*)	Guía práctica	Manejo de sesiones	
Diseño	Arquitectura de seguridad	Integrando la seguridad e Ingeniería de sistemas	Punto de acceso único (*Single Access Point*)	Práctica Técnica	Manejo del flujo de procesamiento	
Diseño	Arquitectura de seguridad	Integrando la seguridad e Ingeniería de sistemas	Punto de chequeo (*Checked Point*)	Práctica Técnica	Manejo del flujo de procesamiento	
Diseño	Arquitectura de seguridad	Integrando la seguridad e Ingeniería de sistemas	Sesión segura (*Security Session*)	Práctica Técnica	Manejo de sesiones	
Diseño	Arquitectura de seguridad	Integrando la seguridad e Ingeniería de sistemas	Ambiente de ejecución controlado (*Controlled Execution Enviorenment*)	Práctica Técnica	Manejo del flujo de procesamiento Mecanismos de autorización	
Diseño	Arquitectura de seguridad	Integrando la seguridad e Ingeniería de sistemas	Socios conocidos (*Known Partners*)	Práctica Técnica	Mecanismos de identificación Mecanismos de autenticación	
Diseño	Arquitectura de seguridad	Patrones Centrales de Seguridad	Enrutador seguro de mensajes (*Secure Message Router*)	Práctica Técnica	Mecanismos de autenticación	
Diseño	Arquitectura de seguridad	Repositorio de patrones de seguridad	*Proxy* confiable (*Trusted Proxy*)	Guía práctica	Mecanismos de autorización	
Diseño	Arquitectura de seguridad	Patrones Centrales de Seguridad	Acción base segura (*Secure Base Action*)	Práctica Técnica	Separación de obligaciones Manejo del flujo de procesamiento	
Diseño	Arquitectura de seguridad	Patrones Centrales de Seguridad	Fachada de servicios seguros (*Secure Service Façade*)	Práctica Técnica	Separación de obligaciones Manejo del flujo de procesamiento	
Diseño	Arquitectura de seguridad	Patrones Centrales de Seguridad	Objeto seguro de sesión (*Secure Session Object*)	Práctica Técnica	Manejo de sesiones	
Diseño	Diseño de controles de autenticación y autorización	Integrando la seguridad e Ingeniería de sistemas	Diseño y uso de contraseñas (*Password Design and Use*)	Guía práctica	Administración de contraseñas	Administración de contraseñas
Diseño	Diseño de controles de autenticación y autorización	Patrones Centrales de Seguridad	Administrador seguro de sesiones (*Secure Session Manager*)	Práctica Técnica	Manejo de sesiones	
Diseño	Diseño de controles de autenticación y autorización	Patrones Centrales de Seguridad	Administración segura del contenedor (*Container Managed Security*)	Práctica Técnica	Mecanismos de autenticación	

Diseño	Diseño de controles de autenticación y autorización	Repositorio de patrones de seguridad	Bloqueo de cuentas (*Account Lockout*)	Guía práctica	Administración de contraseñas	
Diseño	Diseño de controles de autenticación y autorización	Repositorio de patrones de seguridad	Autenticación con contraseñas (*Password Authentication*)	Guía práctica	Administración de contraseñas	Administración de contraseñas
Diseño	Diseño de controles de autenticación y autorización	Repositorio de patrones de seguridad	Propagación de contraseñas (*Password Propagation*)	Guía práctica	Administración de contraseñas	Administración de contraseñas
Diseño	Diseño de controles de autenticación y autorización	Guía técnica de patrones de diseño seguros	Descriptor de sujetos (*Subject descriptor*)	Práctica técnica	Mecanismos de autenticación Mecanismos de autorización	
Diseño	Diseño de controles de autenticación y autorización	Integrando la seguridad e Ingeniería de sistemas	Autorización (*Authorization*)	Práctica Técnica	Mecanismos de autorización	
Diseño	Diseño de controles de autenticación y autorización	Integrando la seguridad e Ingeniería de sistemas	Control de acceso basado en roles (*Role-Based Access Control*)	Práctica Técnica	Mecanismos de autorización	
Diseño	Diseño de controles de autenticación y autorización	Integrando la seguridad e Ingeniería de sistemas	Monitor de consulta (*Reference Monitor*)	Práctica Técnica	Mecanismos de autorización	
Diseño	Diseño de controles de autenticación y autorización	Integrando la seguridad e Ingeniería de sistemas	Seguridad multinivel (*Multilevel Security*)	Práctica Técnica	Mecanismos de autorización	Clasificación de la información
Diseño	Diseño de controles de autenticación y autorización	Integrando la seguridad e Ingeniería de sistemas	Acceso total con errores (*Full Access with Errors*)	Práctica Técnica	Mecanismos de autorización	
Diseño	Diseño de controles de autenticación y autorización	Integrando la seguridad e Ingeniería de sistemas	Límites de acceso (*Limited Access*)	Práctica Técnica	Mecanismos de autorización	
Diseño	Diseño de controles de autenticación y autorización	Integrando la seguridad e Ingeniería de sistemas	Autenticador (*Authenticator*)	Práctica Técnica	Mecanismos de identificación Mecanismos de autenticación	
Diseño	Diseño de controles de autenticación y autorización	Integrando la seguridad e Ingeniería de sistemas	Autorización de archivos (*File Authorization*)	Práctica Técnica	Mecanismos de autorización	
Diseño	Diseño de controles de autenticación y autorización	Patrones Centrales de Seguridad	Autenticación forzosa (*Authentication Enforcer*)	Práctica Técnica	Mecanismos de autenticación	
Diseño	Diseño de controles de autenticación y autorización	Patrones Centrales de Seguridad	Autorización forzosa (*Authorization Enforcer*)	Práctica Técnica	Mecanismos de autorización	
Diseño	Diseño de controles de autenticación y autorización	Patrones Centrales de Seguridad	Constructor de reafirmación (Assertion Builder)	Práctica Técnica	Mecanismos de autenticación	
Diseño	Diseño de controles de autenticación y autorización	Patrones Centrales de Seguridad	Token para credenciales (*Credential Tokenizer*)	Práctica Técnica	Mecanismos de autenticación	
Diseño	Diseño de controles de autenticación y autorización	Patrones Centrales de Seguridad	Delegación de firma única (*Single Sign-on (SSO) Delegator*)	Práctica Técnica	Administración de contraseñas Mecanismos de autenticación	
Diseño	Diseño de controles de autenticación y autorización	Patrones Centrales de Seguridad	Sincronización de contraseñas (*Password Synchronizer*)	Práctica Técnica	Administración de contraseñas Mecanismos de autenticación	
Diseño	Diseño de controles de autenticación y autorización	Integrando la seguridad e Ingeniería de sistemas	Monitor controlador de objetos (*Controlled Objected Monitor*)	Práctica Técnica	Mecanismos de autorización	

Diseño	Diseño de controles de entrada	Repositorio de patrones de seguridad	Filtros para las entradas del cliente (*Client Input Filters*)	Guía práctica	Validación de datos de entrada Validación de parámetros	
Diseño	Diseño de controles de entrada	Patrones Centrales de Seguridad	Validador interceptor (*Intercepting Validator*)	Práctica Técnica	Validación de datos de entrada Validación de parámetros	
Diseño	Diseño de controles de proceso	Patrones Centrales de Seguridad	Inspector de mensajes (*Message Inspector*)	Práctica Técnica	Manejo del procesamiento interno	
Diseño	Diseño de controles de proceso	Patrones Centrales de Seguridad	Portal interceptor de mensajes (*Message Interceptor Gateway*)	Práctica Técnica	Separación de obligaciones Manejo del flujo de procesamiento	
Diseño	Diseño de controles de proceso	Integrando la seguridad e Ingeniería de sistemas	Creador controlador de procesos (*Controlled Process Creator*)	Práctica Técnica	Manejo del procesamiento interno Mecanismos de autorización	
Diseño	Diseño de controles de proceso	Integrando la seguridad e Ingeniería de sistemas	Fabrica controladora de objetos (*Controlled Object Factory*)	Práctica Técnica	Manejo del procesamiento interno Mecanismos de autorización	
Diseño	Diseño de controles de proceso	Integrando la seguridad e Ingeniería de sistemas	Controlador de espacio de direcciones virtuales (*Controlled Virtual Address Space*)	Práctica Técnica	Manejo del procesamiento interno	
Diseño	Diseño de controles de proceso	Integrando la seguridad e Ingeniería de sistemas	Ejecución del dominio público (*Execution Domain*)	Práctica Técnica	Manejo del procesamiento interno Mecanismos de autorización	
Diseño	Diseño de controles de salida	Integrando la seguridad e Ingeniería de sistemas	Información anónima (*Information Obscurity*)	Práctica Técnica	Manejo de algoritmos criptográficos Validación de información de salida	
Diseño	Diseño de controles de salida	Patrones Centrales de Seguridad	Objeto de transferencia ofuscado (*Obfuscated Transfer Object*)	Práctica Técnica	Manejo de procesamiento interno Validación de información de salida	
Diseño	Diseño de controles de salida	Repositorio de patrones de seguridad	Almacenamiento de datos en el cliente (*Client Data Storage*)	Guía práctica	Manejo de algoritmos criptográficos	
Diseño	Diseño de controles de salida	Repositorio de patrones de seguridad	Almacenamiento encriptado (*Encrypted Storage*)	Guía práctica	Manejo de algoritmos criptográficos	
Diseño	Diseño de controles de monitoreo, recuperación y auditoría	Guía técnica de patrones de diseño seguros	Puesto de control en el sistema (*Checkpointed system*)	Práctica técnica	Crear una infraestructura de seguridad Manejo de respaldos Mecanismos de detección y notificación de anomalías	Plan de contingencia
Diseño	Diseño de controles de monitoreo, recuperación y auditoría	Guía técnica de patrones de diseño seguros	Comparar-Verificar un sistema tolerante a fallas (*Comparator-checked fault-tolerant system*)	Práctica técnica	Crear una infraestructura de seguridad Manejo de respaldos Mecanismos de detección y notificación de anomalías	Plan de contingencia
Diseño	Diseño de controles de monitoreo, recuperación y auditoría	Patrones Centrales de Seguridad	Interceptor auditor (*Audit Interceptor*)	Práctica Técnica	Manejo de bitácoras	Realización de auditorías
Diseño	Diseño de controles de monitoreo, recuperación y auditoría	Patrones Centrales de Seguridad	Administración dinámica de servicios (*Dynamic Service Management*)	Práctica Técnica	Mecanismos de detección y notificación de anomalías Manejo de procesamiento interno	

Diseño	Diseño de controles de monitoreo, recuperación y auditoría	Repositorio de patrones de seguridad	Bitácora para auditoria (*Log for Audit*)	Guía práctica	Manejo de bitácoras	Realización de auditorías
Diseño	Diseño de controles de monitoreo, recuperación y auditoría	Patrones Centrales de Seguridad	Bitácora segura (*Secure Logger*)	Práctica Técnica	Manejo de errores Manejo de bitácora	Realización de auditorías
Diseño	Diseño de controles de monitoreo, recuperación y auditoría	Repositorio de patrones de seguridad	Campo de minas (*Minefield*)	Guía práctica	Mecanismos de detección y notificación de anomalías	
Diseño	Diseño de controles de monitoreo, recuperación y auditoría	Repositorio de patrones de seguridad	Lista negra de direcciones de red (*Network Address Blacklist*)	Guía práctica	Mecanismos de detección y notificación de anomalías	
Diseño	Diseño de controles de monitoreo, recuperación y auditoría	Repositorio de patrones de seguridad	Aserciones seguras (*Secure Assertion*)	Guía práctica	Mecanismos de detección y notificación de anomalías Manejo del procesamiento interno	
Pruebas	Configuración del ambiente de prueba	Repositorio de patrones de seguridad	Pruebas con un servidor de pruebas (*Test on a Staging Server*)	Guía práctica		Pruebas al sistema
Pruebas	Inspección de la seguridad integral de la aplicación	Repositorio de patrones de seguridad	Pruebas para atacar al diseño (*Red Team the Design*)	Guía práctica		Pruebas al sistema
Producción	Configuración de la seguridad en el ambiente de producción	Repositorio de patrones de seguridad	Aislamiento de procesos en el servidor (*Server Sandbox*)	Guía práctica	Crear una infraestructura de seguridad	
Producción	Configuración de la seguridad en el ambiente de producción	Repositorio de patrones de seguridad	Construir el servidor desde cero (*Build the Server from the Ground Up*)	Guía práctica	Crear una infraestructura de seguridad	
Producción	Configuración de la seguridad en el ambiente de producción	Repositorio de patrones de seguridad	Documentar la configuración del servidor (*Document the Server Configuration*)	Guía práctica	Crear una infraestructura de seguridad	Documentación operativa de procesos y herramientas
Mantenimiento	Monitoreo de la seguridad y disponibilidad de las aplicaciones	Repositorio de patrones de seguridad	Parchar proactivamente (*Patch Proactively*)	Guía práctica	Infraestructura de actualización de software	

Tabla 4.5 –Clasificación de patrones de seguridad conforme al SDLC

4.3.2 Selección de patrones de seguridad

La selección vista a continuación, corresponde principalmente a la elección de aquellos patrones de seguridad considerados más óptimos para satisfacer los controles de seguridad requeridos por las aplicaciones financieras.

Para esto, se tomó como base la clasificación anterior con el fin de comparar los patrones de seguridad que resuelven el mismo control dentro de la misma actividad, eligiendo preferentemente los patrones enfocados a "prácticas técnicas" frente a aquellos que establecen "guías prácticas". Exceptuando además, los "*Mini-Patterns*" descritos por el Repositorio de patrones de seguridad [11] dado el carácter breve de los mismos.

Además, conforme al alcance de la tesis, por el momento se excluyeron de este estudio los patrones de seguridad enfocados a determinar y configurar una infraestructura de seguridad, es decir, aquellos cuyo ámbito es el área de telecomunicaciones y los mecanismos físicos. Lo anterior se debe, a que la metodología propuesta se basa en el establecimiento de un ambiente de seguridad previamente definido en la institución, con el fin de concentrarse exclusivamente al desarrollo de aplicaciones financieras seguras.

Finalmente, aunque se incluyen en este estudio (Véase Tabla 4.6), se tomó la decisión de identificar y separar del resto de los patrones de seguridad, a aquellos correspondientes a la fuente "Patrones Centrales de Seguridad" [57], dado su alcance exclusivo para el diseño y desarrollo de aplicaciones J2EE. Así, conforme a los anteriores criterios, los patrones de seguridad seleccionados fueron los siguientes:

Etapa SDLC	Actividad	Control Computacional o Administrativo	Patrones Generales	Patrones para J2EE
Análisis	Identificación de los requerimientos funcionales y de seguridad	Identificación de los requerimientos funcionales y de seguridad Documentación de requerimientos	Identificación de necesidades de seguridad para los activos de la empresa (*Security Needs Identification for Enterprise Assets*) Requerimientos de identificación y autenticación (*Identification & Authentication Requeriments*) Requerimientos de control de acceso (*Access Control Requeriments*) Requerimientos de no repudiación (*Non-Repudiation Requirements*) Requerimientos de seguridad contable (*Security Accounting Requirements*) Requerimientos de auditoria (*Audit Requirements*) Requerimientos de rastreo de auditoria y bitácora (*Audit Trails and Logging Requirements*) Requerimientos de detección de intrusiones (*Intrusion Detection Requirements*) Definición de derechos por rol (*Role Rights Definition*)	
	Análisis de riesgos	Análisis de riesgos	Valoración de activos (*Asset Valuation*) Valoración de amenazas (*Threat Assessment*) Valoración de vulnerabilidades (Vulnerability Assessment) Determinación de riesgos (*Risk Determination*)	
	Selección de controles	Definición de controles de seguridad	Enfoques de seguridad en la empresa (*Enterprise Security Approaches*) Servicios de seguridad de la empresa (*Enterprise Security Services*) Registrándose al validar fuera de banda (Enroll by Validating Out of Band) Registrándose usando una validación de una tercera parte (*Enroll using Third-Party Validation*) Registrándose con un secreto compartido preexistente (*Enroll with a Pre-Existing Shared Secret*)	
	Planeación de la seguridad	Documentación operativa de procesos y herramientas	Compartir la responsabilidad por la seguridad (*Share Responsibility for Security*) Comunicación con otras compañías (*Enterprise Partner Communication*) Escoger los productos correctos (*Choose the Right Stuff*)	
Diseño Desarrollo	Arquitectura de seguridad	Mecanismo de identificación Mecanismo de autenticación	Alternativas de diseño automático de I&A (*Automated I&A Design Alternatives*) Socios conocidos (*Known Partners*)	
		Separación de obligaciones Manejo del flujo de procesamiento	Partición de la aplicación (*Partitioned Application*) Punto de acceso único (*Single Access Point*) Punto de chequeo (*Checked Point*) Contexto Seguro (*Security Context*) Asociación de seguridad (*Security Association*)	Acción base segura (*Secure Base Action*) Delegación de políticas (*Policy Delegate*) Fachada de servicios seguros (*Secure Service Façade*) Agente web interceptor (*Intercepting Web Agent*)

		Seguridad en las comunicaciones Manejo de algoritmos criptográficos	Canales seguros (*Secure Channels*)	Pipa segura (*Secure Pipe*)
		Manejo de sesiones	Sesión segura (*Security Session*)	Objeto seguro de sesión (*Secure Session Object*)
	Diseño de controles de autenticación y autorización	Mecanismos de identificación Mecanismos de autenticación Mecanismos de autorización	Acceso total con errores (*Full Access with Errors*) Límites de acceso (*Limited Access*) Autenticador (*Authenticator*) Control de acceso basado en roles (*Role-Based Access Control*) Monitor de consulta (*Reference Monitor*)	Autenticación forzosa (*Authentication Enforcer*) Autorización forzosa (*Authorization Enforcer*) Administración segura del contenedor (*Container Managed Security*) Administrador seguro de sesiones (*Secure Session Manager*) Token para credenciales (*Credential Tokenizer*)
		Administración de contraseñas	Diseño y uso de contraseñas (*Password Design and Use*) Bloqueo de cuentas (*Account Lockout*) Propagación de contraseñas (*Password Propagation*)	Delegación de firma única (*Single Sign-on (SSO) Delegator*) Sincronización de contraseñas (*Password Synchronizer*)
	Diseño de controles de entrada	Validación de datos de entrada	Filtros para las entradas del cliente (*Client Input Filters*)	Validador interceptor (*Intercepting Validator*)
	Diseño de controles de proceso	Manejo de procesamiento interno Mecanismos de autorización	Creador controlador de procesos (*Controlled Process Creator*) Fabrica controladora de objetos (*Controlled Object Factory*) Ejecución del dominio público (*Execution Domain*)	
	Diseño de controles de salida	Manejo de algoritmos criptográficos Validación de información de salida	Información anónima (*Information Obscurity*) Almacenamiento de datos en el cliente (*Client Data Storage*) Almacenamiento encriptado (*Encrypted Storage*)	Objeto de transferencia ofuscado (*Obfuscated Transfer Object*))
	Diseño de controles de monitoreo, recuperación y auditoría	Mecanismos de detección y notificación de anomalías Manejo de procesamiento interno	Puesto de control en el sistema (*Checkpointed system*) Comparar-Verificar un sistema tolerante a fallas (*Comparator-checked fault-tolerant system*) Aserciones seguras (*Secure Assertion*)	Administración dinámica de servicios (*Dynamic Service Management*)
		Manejo de bitácoras Realización de auditorías	Bitácora para auditoria (*Log for Audit*)	Bitácora segura (*Secure Logger*) Interceptor auditor (*Audit Interceptor*)
Pruebas	Inspección de la seguridad integral de la aplicación	Pruebas al sistema	Pruebas para atacar al diseño (*Red Team the Design*)	
	Configuración de la seguridad en el ambiente de prueba	Pruebas al sistema	Pruebas con un servidor de pruebas (*Test on a Staging Server*)	

Tabla 4.6 –Patrones de seguridad para las aplicaciones financieras

4.4 Conclusiones

De manera general, este capítulo nos permitió conocer y analizar los principales patrones de seguridad que actualmente existen en la literatura. Como se pudo observar, los patrones de seguridad es un tipo de técnica que proporciona guías y prácticas de manera estructurada para la incorporación de actividades, controles o mejores prácticas de seguridad. Los cuales principalmente, recaen durante las etapas de análisis y diseño, afectando así al desarrollo de la aplicación.

El aspecto clave de este tipo de propuesta, se debe a que proporciona una solución genérica y probada a un problema recurrente de seguridad, la cual puede ser fácilmente comprendida e implementada por los desarrolladores de sistemas. Por lo que los patrones de seguridad son vistos como un medio para superar la brecha actual de desconexión entre los profesionales de seguridad y los desarrolladores de la aplicación.

Además, al analizarlos, nos dimos cuenta que los patrones se pueden mapear dentro de las diferentes etapas del ciclo de vida del software, permitiendo así robustecer las metodologías de desarrollo de sistemas. En especial, ésta técnica permite fortalecer la etapa de diseño, misma que impacta directamente a la fase de desarrollo; ésto se vuelve de suma importancia dado que, cómo lo indica Gartner [51] en su teleconferencia sobre "Construcción de Aplicaciones Seguras" (Véase Tabla 3.1 del capítulo 3), en las etapas de diseño y desarrollo del sistema, es donde un mayor número de vulnerabilidades se introducen a la aplicación. De esta manera, el uso de los patrones de seguridad nos puede ayudar a disminuir el número de vulnerabilidades que las aplicaciones financieras pudieran presentar.

Finalmente esta técnica, permite satisfacer la mayoría de los controles de seguridad que fueron detectados durante la sección 3.5 "Controles de seguridad". Por lo que su uso, no solo permitirá mitigar el impacto de las actividades maliciosas en la aplicación al disminuir el número de vulnerabilidades, sino también cumplir con los requerimientos de seguridad que dicta la normatividad del Sector Financiero Mexicano.

CAPÍTULO 5

5 Metodología para el Desarrollo de Aplicaciones Web Financieras Seguras

Con el fin de producir código seguro, en el que se mitiguen riesgos como los mencionados en los capítulos anteriores, las instituciones necesitan una metodología de desarrollo de sistemas que soporte consistentemente dicho objetivo. El ciclo de vida de desarrollo de software [38], es una técnica de administración de proyectos la cual divide a las actividades involucradas en la creación de una aplicación, en fases o segmentos más pequeños y fáciles de administrar. Esta segmentación de los proyectos, permite a los administradores verificar el cumplimiento exitoso de cada una de las partes antes de continuar con las fases subsecuentes.

Debido a ello, diversas metodologías para el desarrollo de sistemas han surgido a lo largo del tiempo, desde el ciclo de vida tradicional o en cascada, hasta el espiral, RUP (Rational Unified Process), y XP (Programación Extrema)[10], las cuales, como mencionamos, dividen la producción de una aplicación en diversas etapas. Adicionalmente, algunos de los estándares que se han revisado (FFIEC y NIST SP800-64 por ejemplo) en el capítulo 2. "Estado del Arte", también proponen o se basan en una metodología a seguir.

A pesar de la diversidad de fases propuestas por las metodologías de desarrollo de sistemas, la mayoría de los proyectos típicamente incluyen las fases de análisis o iniciación y planeación, diseño, desarrollo, pruebas, implementación, mantenimiento y en algunas ocasiones retiro [38]. Las cuales, pueden integrase o dividirse en otras etapas dependiendo de la institución o la metodología a seguir.

Conforme a ello, la presente metodología no aspira a crear un nuevo ciclo de vida para el desarrollo de software sino más bien, identifica y se integra a la metodología llevada por la organización, bajo la forma de una serie de actividades adicionales a implementar para mantener la seguridad de la información en cada una de las etapas comunes del ciclo de vida de desarrollo de sistemas. Donde además, los patrones de seguridad serán nuestra principal herramienta a utilizar para el establecimiento de actividades e implementación de controles de seguridad dentro del ciclo de vida de desarrollo del sistema.

[10] El desarrollo de la presente tesis no pretende establecer una comparación entre las diferentes metodologías de desarrollo de software. Por lo que se sugiere consultar otros documentos si se desea obtener una mayor información sobre las mismas.

5.1 Aplicación de la metodología

La presente metodología de desarrollo para aplicaciones financieras seguras se enfoca a:

- Proveer una guía para la construcción de software que satisfaga los objetivos mínimos de seguridad de las instituciones financieras.
- Identificar las actividades clave que permiten integrar aspectos de seguridad dentro del ciclo de vida de desarrollo de software
- Ayudar a integrar durante cada una de las etapas del ciclo de vida de desarrollo de software, los controles de seguridad que requiere el sistema para cumplir con sus objetivos de seguridad.
- Disminuir el número de vulnerabilidades que estas aplicaciones puedan presentar, al haberse construido pensando en seguridad.

En cuanto a la selección del caso de estudio, los puntos que se deberán cumplir son aquellos especificados en la sección 5.2.1 "Pre-requisitos para la aplicación de la metodología".

5.1.1 Pre-requisitos para la aplicación de la metodología

Esta metodología toma como base los siguientes puntos para su aplicación:

- La aplicación a desarrollar pertenece a una institución del sector financiero mexicano
- La aplicación requiere del envío y/o recepción de información de carácter confidencial a través de la red.
- Parte o toda la aplicación trabaja en ambiente web.
- Existe actualmente dentro de la institución, una gestión organizacional o gobierno que abogue por la seguridad. Es decir que tanto la alta dirección como el área de sistemas se encuentren preocupados por la seguridad y se hayan establecido previamente una serie de políticas institucionales para lograr este fin.
- Se cuente con una infraestructura de seguridad actualmente establecida y en funcionamiento. Lo cual involucra el previo establecimiento de políticas, controles físicos de seguridad y una infraestructura que permita el desarrollo seguro de sistemas. Sin temor a que personal no autorizado acceda al código fuente de las aplicaciones.
- De preferencia se maneje un estándar internacional como los presentados dentro del capítulo 2. "Estado del Arte" o en su defecto posee su propia normatividad interna para el desarrollo

de sistemas. El cual, ayude a la organización a establecer un gobierno de seguridad en TI, contemplando la seguridad en otras áreas que la presente metodología no abarca, como es la seguridad en redes, en el ambiente de trabajo, administración de inventarios y documentación entre otras.

- En esta primera versión de la metodología, se pide de preferencia apoyarse en una metodología comercial de desarrollo de software.
- Mapear las fases del ciclo de vida con el que actualmente se está trabajando, contra las etapas que la presente metodología propone. Por ejemplo:

Metodología del NIST SP800-64	Metodología propuesta
Iniciación	Análisis
Desarrollo	Diseño
	Desarrollo
	Pruebas
Implementación	Producción
Mantenimiento	Mantenimiento
Retiro	Retiro

Tabla 5.1 – Ejemplo de mapeo entre metodologías

De esta manera, si utilizáramos la metodología del NIST SP800-64, entonces durante la fase de "Iniciación", tendríamos que aplicar las actividades que la metodología propone para su fase de "Análisis".

5.1.2 Aplicación de la metodología

Para lograr una rápida adopción y uso de la presente metodología, esta posee las siguientes características para su aplicación:

- *Presenta la estructura de un ciclo de vida común para el desarrollo de sistemas.* Lo que permite al lector su seguimiento paso a paso para el desarrollo continuo de la aplicación, o en su defecto, permitir dirigirse directamente a la fase de desarrollo del sistema que le interése o desee implementar, o incluso que cada fase se aplique por el equipo o grupo de desarrollo que corresponda, dependiendo de las prácticas de la institución. Cabe mencionar que toma en consideración, que anteriormente ya se mapearon las etapas del ciclo de vida utilizado, contra las fases propuestas por la presente metodología.
- *Establece una serie de actividades de seguridad a implementar.* Dentro de cada fase, la metodología propone las actividades de seguridad a realizar, lo que permite poco a poco ir

estableciendo y desarrollando los controles de seguridad para la aplicación y así reducir el número de vulnerabilidades que esta pueda presentar.

- En algunas ocasiones la guía hará referencia a puntos tratados con anterioridad, como por ejemplo, el uso de un patrón de seguridad que se aplica dentro de la actividad a desarrollar. En este caso, favor de referirse al capítulo y referencia mencionada.
- En otras ocasiones, para una mayor información o claridad sobre el tema, se pedirá referirse a los controles de una guía o estándar en especial, siempre y cuando este sea de dominio público.

5.1.3 Audiencia

Esta guía se encuentra enfocada hacia los líderes informáticos y sus equipos de desarrollo de sistemas que se encuentren laborando dentro de alguna institución del sector financiero mexicano.

Sin embargo esta guía no es completamente exclusiva para este sector, como se puede apreciar a través de una rápida revisión de la misma, las actividades, controles identificados y los patrones de seguridad utilizados, bien pueden emplearse para desarrollar algún otro tipo de aplicación que no necesariamente sea financiera pero que requiere mantener controles básicos en cuanto a seguridad, por ejemplo las aplicaciones e-commerce.

5.2 Metodología de desarrollo de aplicaciones web financieras con patrones de seguridad.

OWASP, dentro de su guía para el desarrollo de aplicaciones [43] nos comenta lo siguiente:

"Las aplicaciones seguras no se dan por sí mismas, son en cambio el resultado de:

- Una gestión organizacional que abogue por la seguridad
- Políticas de seguridad documentadas y apropiadamente basadas en estándares nacionales
- Una metodología de desarrollo con adecuados puntos de control y actividades de seguridad
- Gestión segura de versiones y configuración"

Tomando en cuenta las anteriores observaciones de la OWASP, la presente metodología por lo tanto se sustenta bajo la existencia dentro de la organización, de un gobierno en tecnologías de la información que integre a la seguridad de la información. De esta manera, la metodología propuesta complementa el ambiente de seguridad al establecer una serie de actividades a incorporar dentro del ciclo de vida de desarrollo de sistemas manejado por la institución.

Adicionalmente, la misma tiene por objetivo proponer una serie de controles (y/o patrones de seguridad) los cuales permitan cumplir con los requerimientos generales en cuanto a seguridad de la información dictaminada por la normatividad del sector financiero mexicano, y congruentes con los principales estándares internacionales en materia financiera que las instituciones de este tipo normalmente se encuentran obligadas a seguir (Véase

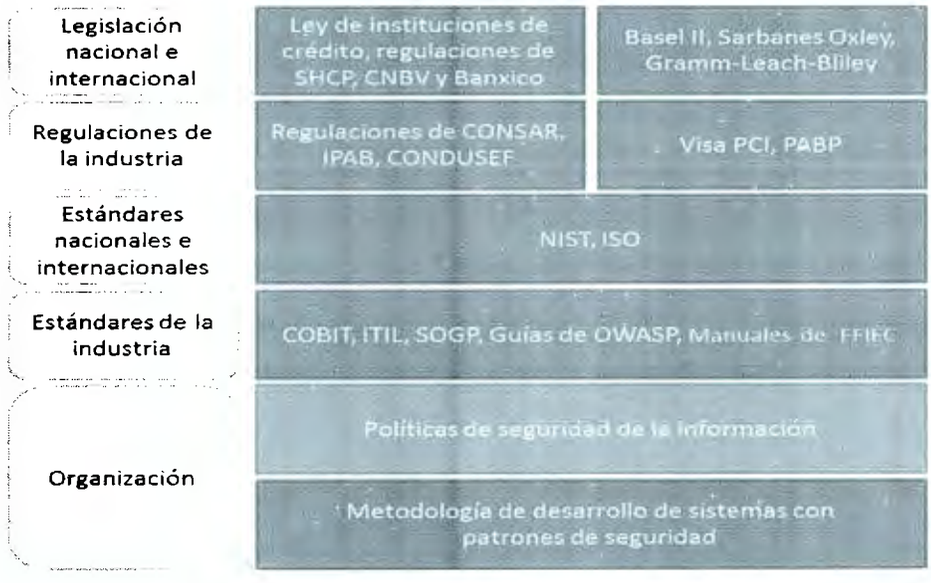


Figura 5.1).

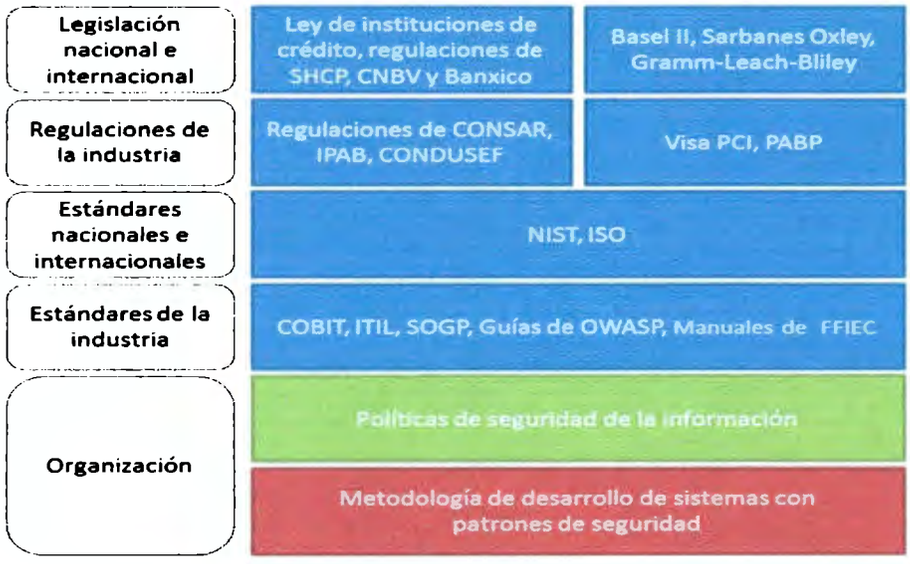


Figura 5.1 – Contexto de la metodología propuesta

Entrando en materia, la presente metodología tomó como punto de partida el ciclo de vida de desarrollo de sistemas propuesto por la FFIEC [38]. Sin embargo, dado el estudio realizado durante la sección: 2.2.3 "Actividades de seguridad identificadas dentro de los estándares y guías de desarrollo", se decidió unir en la actividad de "Análisis", las etapas de "Iniciación" y "Planeación". De esta manera la metodología a presentar se conformará de las siguientes etapas:

Figura 5.2 –Etapas de la metodología propuesta

5.2.1 Análisis

Se definen y expresan las necesidades de información que motivaron el desarrollo del sistema, así como el propósito del mismo. Esta etapa tiene como objetivo, la creación de un primer documento el cual refleje los requerimientos funcionales y no funcionales que ofrecerá el sistema al usuario.

5.2.1.1 Identificación de los requerimientos funcionales y de seguridad.

Conforme al estándar ISO/IEC 27002:2005 [40], es esencial que una organización identifique sus requerimientos de seguridad. Para ello existen tres fuentes principales:

1. El **conjunto particular de principios**, objetivos y requerimientos **comerciales** para el procesamiento. Este punto depende mucho del tipo de institución financiera que este elaborando el sistema y además del tipo de sistema a desarrollar.
2. Los **requerimientos legales**, reguladores, estatutarios y contractuales que tiene que satisfacer una organización. Para ello se recomienda tomar como base la Tabla 2.8 Requerimientos de seguridad dentro de la legislación, e identificar los requerimientos particulares que aplican a la institución y al sistema a desarrollar. Dicha tabla se deberá complementar, de ser necesario, con los requerimientos de seguridad de la información que indica la normatividad adicional aplicable a la institución, dado su giro de especialización dentro del sector financiero mexicano. Así, por ejemplo: si la institución financiera pertenece al ámbito bancario, entonces será necesario incluir los requerimientos que indica la ley sobre la privacidad de la CNBV o bien los estándares internacionales Payment Card Industry (PCI), Data Security Standard y Protect Cardholder Data, entre otros.
3. A través de una **evaluación de riesgos** de la organización, tomando en cuenta la estrategia general y los objetivos de la organización. Identificando amenazas para los activos, evaluando la vulnerabilidad y la probabilidad de ocurrencia de las amenazas para calcular el impacto potencial. Para llevar a cabo esta evaluación, se recomienda consultar la sección 5.1.1.3 "Análisis de riesgos" de la presente metodología.

Con respecto a los primeros dos puntos mencionados, el libro "Patrones de Seguridad: Integrando la seguridad y la ingeniería de sistemas" [29], ofrece una serie de patrones los cuales permiten definir las necesidades de seguridad de la institución. Aunque estos patrones se enfocan

al negocio de la empresa, las actividades proporcionadas permiten además identificar los requerimientos de un sistema en particular.

Analizando, el patrón esencial que permite determinar los requerimientos de seguridad de una aplicación es:

- ✓ **Nombre del patrón:** Identificación de necesidades de seguridad para los activos de la empresa (*Security Needs Identification for Enterprise Assets*)
  Intención: Ayuda a identificar las propiedades generales de seguridad que deben de aplicarse a un sistema en particular
  Solución: Las actividades a seguir son las siguientes. Aunque están descritas a nivel de la empresa, acotarlas a nivel de sistema.

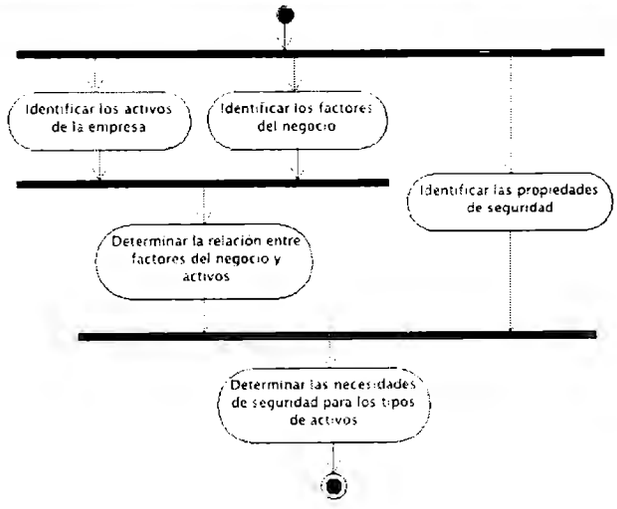


Figura 5.3 –Actividades para la identificación de necesidades de seguridad [29]

Posteriormente, para aquellos sistemas que lo consideren pertinente, pueden adicionalmente refinar los requerimientos identificados con respecto a alguno de los siguientes aspectos, auxiliándose de su correspondiente patrón:

- ✓ **Nombre del patrón:** Requerimientos de identificación y autenticación (*Identification & Authentication Requeriments*)
  Intención: Su función es reconocer a un individuo y validar su identidad individual.

- ✓ **Nombre del patrón:** Requerimientos de control de acceso (*Access Control Requeriments*)
  Intención: Los requerimientos dirigen tanto la función de control de acceso como las propiedades del servicio de control de acceso, así como la facilidad de uso y flexibilidad. Establece los privilegios que puede tener un individuo previamente autenticado

- ✓ **Nombre del patrón:** Requerimientos de seguridad contable (*Security Accounting Requirements*)
  Intención: Provee de un conjunto genérico de requerimientos de registro de seguridad para rastrear acciones o eventos relacionados con la seguridad.

- ✓ **Nombre del patrón:** Requerimientos de auditoría (*Audit Requirements)*
  Intención: Comprende un conjunto genérico de requerimientos de auditoría y ayuda a priorizarlos. Lo cual incluye análisis de bitácora y rastreo de información de auditoria sobre un evento para encontrar o reportar alguna indicación de violaciones a la seguridad.

- ✓ **Nombre del patrón:** Requerimientos de identificación y autenticación (*Identification & Authentication Requeriments*)
  Intención: Su función es reconocer a un individuo y validar su identidad individual.

- ✓ **Nombre del patrón:** Requerimientos de control de acceso (*Access Control Requeriments*)
  Intención: Los requerimientos dirigen tanto la función de control de acceso como las propiedades del servicio de control de acceso, así como la facilidad de uso y flexibilidad. Establece los privilegios que puede tener un individuo previamente autenticado

- ✓ **Nombre del patrón:** Requerimientos de seguridad contable (*Security Accounting Requirements*)
  Intención: Provee de un conjunto genérico de requerimientos de registro de seguridad para rastrear acciones o eventos relacionados con la seguridad.

- ✓ **Nombre del patrón:** Requerimientos de auditoría (*Audit Requirements)*
  Intención: Comprende un conjunto genérico de requerimientos de auditoría y ayuda a priorizarlos. Lo cual incluye análisis de bitácora y rastreo de información de auditoria sobre un evento para encontrar o reportar alguna indicación de violaciones a la seguridad.

- ✓ **Nombre del patrón:** Requerimientos de rastreo de auditoria y bitácora (*Audit Trails and Logging Requirements*)
  Intención: Provee de un conjunto genérico de requerimientos de auditoria trazable, ayudando a aplicarlos en situaciones específicas y a determinar su importancia relativa. Captura bitácoras de auditoría sobre eventos y actividades que ocurren dentro de una organización o sistema, para facilitar la reconstrucción y análisis de esos eventos y actividades.

- ✓ **Nombre del patrón:** Requerimientos de detección de intrusiones (*Intrusion Detection Requirements*)
  Intención: Ayuda a especificar los requerimientos que aplican a cada situación y su importancia relativa. Para automatizar el monitoreo de eventos sobre alguna indicación de violación a la seguridad (pudiendo basarse en horarios, uso, accesos realizados, etc)

- ✓ **Nombre del patrón:** Requerimientos de no repudiación (Non-Repudiation Requirements)
  Intención: Provee de un conjunto genérico de requerimientos de no repudiación, ayudando a su aplicación y a determinar su importancia relativa. Captura y mantiene evidencia para que los participantes de una transacción o iteración no puedan negar que han participado en la actividad

Al identificar los requerimientos funcionales y no funcionales, normalmente se crean casos de uso, o documentos que describirán las diferentes funcionalidades a ofrecer por el sistema. Para cada caso de uso, se deberá además identificar los usuarios que intervienen en el proceso, clasificándolos normalmente por roles. Para esta actividad, se recomienda utilizar el siguiente patrón:

- ✓ **Nombre del patrón:** Definición de derechos por rol (Role Rights Definition)
  Intención: Provee de una forma, basada en casos de uso, para asignar derechos a los roles e implementar la política de "menor privilegio".

Al finalizar esta etapa, lo que se obtendrá será la lista actualizada de requerimientos de seguridad que se definió en la sección 2.1.3 "Requerimientos de seguridad dentro de la legislación"; a la cual se le aumentó o disminuyó los requerimientos de seguridad identificados durante esta etapa.

5.2.1.2 Clasificación de la información y la seguridad

Conforme a NIST SP800-64 [39] la categorización de la información empieza con la identificación del tipo de información que manejará el negocio. Subsecuentemente, se deberá evaluar para identificar aquellas que presentan un mayor riesgo o cuya pérdida afecta en gran medida a la organización. FIPS PUB 199 "Standards for Security Categorization of Federal Information and Information Systems", define un proceso para categorizar la seguridad basada en el impacto potencial dentro de una organización que puedan traer ciertos eventos al comprometer la información o los sistemas de información para cumplir su misión, proteger sus valores, cumplir sus responsabilidades legales, mantener sus funciones del día a día y proteger los individuales.

De manera general, se recomienda que la información se clasifique dentro de alguno de los siguientes rubros dado su criticidad:

1. Pública
2. Reservada
3. Sensible

5.2.1.3 Análisis de riesgos

El objetivo principal del análisis de riesgos es ayudar a seleccionar controles de costo adecuado para mitigar riesgos que pesen sobre procesos críticos de la organización. Para este propósito, debe identificarse amenazas, vulnerabilidades y medidas de protección para reducir el impacto de que una amenaza se concretice.

Existe un sinnúmero de metodologías de análisis de riesgos, algunas cuantitativas, otras cualitativas. Estas últimas son las que actualmente se emplean en materia de Seguridad de la Información.

Análisis de Riesgos Cuantitativo:

- Enfocado a determinar valores numéricos para los componentes objeto del análisis, así como al nivel de posibles pérdidas.
- Los resultados son objetivos, basados en métricas generadas igualmente de forma objetiva. Estos se expresan en porcentajes, probabilidades de ocurrencia de amenazas, pesos, etc.
- Muestra de manera sencilla el costo-beneficio en términos comprensibles a la alta dirección (no técnicos).
- Los cálculos pueden resultar complejos.
- El trabajo previo requiere tiempo y esfuerzos considerables.

Análisis de Riesgos Cualitativo

- No requiere determinar valores numéricos para los componentes objeto del análisis, así como al nivel de posibles pérdidas.
- No es necesario contar con la frecuencia de ocurrencia de las amenazas.
- Los resultados son subjetivos.
- No hay una base para demostrar el costo-beneficio.
- Los cálculos son sencillos.
- La calidad del análisis depende del equipo conformado.

Ahora bien, con respecto a los patrones de seguridad, el libro "Patrones de Seguridad: Integrando la seguridad y la ingeniería de sistemas" [29], define una serie de patrones los cuales en conjunto permiten realizar el análisis de riesgos de la institución. Las actividades que ofrecen estos patrones, a pesar de su carácter general o a nivel empresa, pueden utilizarse para realizar un análisis de riesgos a nivel de la aplicación.

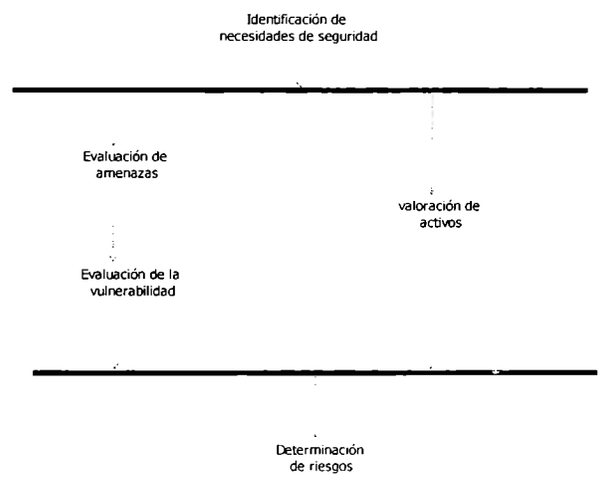


Figura 5.4 –Patrones para la evaluación de riesgos [29]

Al aplicar estos patrones, se sugiere además consultar las secciones: 3.2 Amenazas y ataques informáticos, 3.3 Vulnerabilidades en el diseño y desarrollo de sistemas, y 3.4 Riesgos. Las cuales ofrecen una base general para determinar las amenazas, vulnerabilidades y riesgos que afectan a las aplicaciones financieras; y los cuales, como se podrá observar forman parte de las actividades a realizar durante el análisis de riesgos.

Al finalizar esta etapa, lo que se obtendrá serán los eventos de riesgo detallados que afectan a la aplicación a desarrollar (Véase la sección 3.4 como guía para el establecimiento de los eventos de riesgo)

5.2.1.4 Selección de controles

Una vez obtenidos los eventos de riesgo que afectan a nuestra aplicación (Véase sección 5.1.1.3), así como los requerimientos de seguridad que debe de satisfacer el sistema (Véase sección 5.1.1.1), se deberán seleccionar los controles apropiados a implementar de acuerdo además con la criticidad de la información (Véase sección 5.1.1.2) para asegurar que los riesgos se reduzcan a un nivel aceptable y se cumpla con los requerimientos de seguridad identificados.

Para ello, los controles a implementar se pueden seleccionar a partir de este trabajo (Véase sección 3.5), de otros conjuntos de controles, o bien se pueden diseñar controles nuevos para cumplir con necesidades específicas conforme sea apropiado.

La selección de los controles de seguridad depende de las decisiones organizacionales basadas en el criterio de aceptación del riesgo, opciones de tratamiento del riesgo y el enfoque general para la gestión del riesgo aplicado a la organización, así como también debieran estar sujetas a todas las regulaciones y legislación nacionales e internacionales relevantes.

Independientemente de este trabajo, se recomienda seguir las actividades propuestas por los siguientes patrones:

- ✓ **Nombre del patrón:** Enfoques de seguridad en la empresa (Enterprise Security Approaches)
  Intención: Guía a una empresa en la selección de enfoques de seguridad: prevención, detección y respuesta. Los cuales son dirigidos por las propiedades de seguridad que los activos requieren y dado la evaluación de los riesgos de seguridad. Determinando los servicios de seguridad a establecer por la empresa.

- ✓ **Nombre del patrón:** Servicios de seguridad de la empresa (Enterprise Security Services)
  Intención: Guía en la selección de servicios de seguridad para proteger los activos, después que se ha identificado el enfoque de seguridad. Permite establecer el grado de robustez y confianza que cada servicio debe ofrecer, basado en prioridades.

Adicionalmente, para establecer el ambiente en que se posteriormente trabajará el diseño de controles de autenticación y autorización, se recomienda seguir uno de los siguientes enfoques que proporcionan los siguientes patrones:

- ✓ **Nombre del patrón:** Registrándose al validar fuera de banda (Enroll by Validating Out of Band)
  Intención: Cuando los usuarios se registran a un sitio web o servicio, algunas veces es necesario validar su identidad usando un canal externo, tal como un correo, teléfono o cara a cara. Lo que permitirá instituir un secreto que puede ser usado para establecer la identidad durante el registro.

- ✓ **Nombre del patrón:** Registrándose usando una validación de una tercera parte (*Enroll using Third-Party Validation*)
  Intención: Cuando el servicio de una tercera parte está disponible y es suficientemente confiable, la aplicación le puede delegar la tarea de autenticar la identidad del usuario.

- ✓ **Nombre del patrón:** Registrándose con un secreto compartido preexistente (*Enroll with a Pre-Existing Shared Secret*)
  Intención: Cuando los usuarios se registran en un sitio web o servicio, algunas veces es suficiente con validar la identidad usando un secreto compartido preexistente. Lo cual permite registrarse sin una comunicación previa para establecer una cuenta.

5.2.1.5 Planeación de la seguridad

Conforme al FFIEC [38] una planeación cuidadosa es necesaria dentro de la primera fase del proyecto para coordinar las actividades y administrar los riesgos efectivamente. La profundidad y formalidad de los planes deberán ser apropiados con las características y riesgos del proyecto. Así, se deberán incluir preparaciones para todo el ciclo de vida del sistema, incluyendo la

identificación de herramientas y tecnologías de seguridad. El FFIEC nos propone los siguientes puntos a determinar.

- Vista general del proyecto describiendo, sus metas y estrategias de desarrollo.
- Roles y responsabilidades de los involucrados incluyendo a terceras personas.
- La forma en cómo se reunirá y diseminará la información.
- Criterios de aceptación para cada fase así como los entregables y la documentación.
- Control de requerimientos y administración de cambios.
- Administración de riesgos.
- Aplicación de estándares de la industria para la realización de actividades y aseguramiento de la calidad, tanto para los requerimientos funcionales como de seguridad.
- Programar las tareas a cumplir en cada fase así como el presupuesto asociado.
- Capacitación a los desarrolladores así como planes de prueba

Adicionalmente, los patrones de seguridad nos permiten ampliar algunos de los puntos a analizar durante esta etapa, éstos son:

- ✓ **Nombre del patrón:** Compartir la responsabilidad por la seguridad (*Share Responsibility for Security*)
  Intención: Hace a todos los desarrolladores responsables por la seguridad en el sistema, quienes deben entender y manejar estos conceptos. De esta manera, se evita el problema de separación entre el equipo de seguridad y el de desarrollo.

- ✓ **Nombre del patrón:** Escoger los productos correctos (*Choose the Right Stuff*)
  Intención: Provee de una guía para seleccionar los componentes comerciales apropiados como componentes de seguridad, lenguajes y herramientas, que permiten construir componentes propios..

En caso que el sistema se relacione con sistemas de otras compañías, entonces será necesario aplicar también el siguiente patrón:

- ✓ **Nombre del patrón:** Comunicación con otras compañías (*Enterprise Partner Communication*)
  Intención: Atención a la protección de los datos y los métodos por los cuales son transferidos, entre compañías.

5.2.2 Diseño

Una vez identificados los requerimientos funcionales y no funcionales del sistema, incluyendo a los de seguridad y los controles a aplicar, se deberá determinar cómo será construido el sistema. Para ello, se definirá detalladamente las entidades involucradas y sus relaciones pasando de casos de uso a su definición como casos reales.

5.2.2.1 Arquitectura de seguridad

De acuerdo a Bass, la arquitectura de software de un sistema comprende:

- Los componentes de software,
- Las propiedades visibles externamente de estos componentes, y
- Las relaciones entre ellos

De esta manera, la arquitectura de un sistema consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software de un sistema de información.

Por su parte, la arquitectura de seguridad se refiere a los pilares fundamentales de la seguridad, es decir, la aplicación debe proporcionar controles para proteger la confidencialidad de la información, la integridad de los datos, proporcionar acceso a los datos cuando se requiera y solamente a los usuarios apropiados. [29]

Es así, como la arquitectura de nuestra aplicación se conformará a través del uso de patrones de diseño y patrones de seguridad.

Cabe mencionar, que la mayoría de las aplicaciones existentes dentro del sector financiero, son aplicaciones que se pueden considerar a gran escala. Como buena práctica, las aplicaciones a gran escala necesitan una arquitectura diferente a aquella de un simple formulario de encuesta. Puesto que a medida que las aplicaciones crecen en tamaño, resulta cada vez más difícil implementar y mantener funcionalidades así como una alta escalabilidad. [43]

Debido a ello, una arquitectura de aplicación escalable normalmente se encuentra dividida en niveles, y si se utilizan patrones de diseño, muchas veces se dividen en porciones reutilizables usando diferentes lineamientos específicos para reforzar la modularidad, requerimientos de interface y la reutilización de objetos. Así, una de las arquitecturas de aplicaciones web más comunes es el Modelo Vista Controlador (MVC).

El paradigma MVC es un framework que divide un sistema completo en tres componentes, llamados: modelo, vista y control. En este framework, el modelo es una abstracción del estado y comportamiento del sistema; la vista proporciona alta calidad de visualización en tiempo real y soporta la interacción con el usuario. Finalmente el control, recibe el estado de la información desde el modelo y proporciona comandos para satisfacer algunos objetivos predefinidos, cualitativos y cuantitativos.

Es por esto, que en el presente trabajo se decidió seleccionar los patrones que permiten una mejor separación de intereses. Analizando, se podrá observar que existen patrones de seguridad para la capa de presentación como del negocio, se sugiere por lo tanto seleccionar patrones para ambas capas, sin embargo esto dependerá de las necesidades y características de la aplicación.

Entrando en materia sobre los patrones de seguridad a aplicar, se recomienda primero la ejecución del siguiente patrón para determinar los controles de autentificación a diseñar.

- ✓ **Nombre del patrón:** Alternativas de diseño automático de I&A (*Automated I&A Design Alternatives*)
  Intención: Describe técnicas alternativas para una I&A automática. Ayuda a seleccionar una estrategia apropiada que consiste de una única técnica o combinación de técnicas para satisfacer los requerimientos.

Posteriormente, si el sistema se relacionará con otras aplicaciones, aplicar entonces el siguiente patrón:

- ✓ **Nombre del patrón:** Socios conocidos (*Known Partners*)
  Intención: Si las iteraciones comerciales son sensitivas o de alto valor, asegurar que el usuario con quien estamos interactuando es quien dice ser, así como proveer mecanismos que permitan identificar a nuestros sistemas.

Para definir más específicamente la estructura general de la aplicación, se recomienda elegir uno o varios de los siguientes patrones:

- ✓ **Nombre del patrón:** Partición de la aplicación (*Partitioned Application*)
  Intención: Divide una aplicación larga y compleja en dos o más componentes simples. Cualquier privilegio peligroso es restringido a un único y más pequeño componente. Así, cada componente tiene puntos de seguridad manejables y fáciles de verificar que en una aplicación monolítica.

- ✓ **Nombre del patrón:** Punto de acceso único (*Single Access Point*)
  Intención: Define un único punto de entrada que otorga o deniega la entrada al sistema después de corroborar el acceso requerido por el cliente

- ✓ **Nombre del patrón:** Punto de chequeo (*Checked Point*)
  Intención: Define un mecanismo de respuesta a intentos no autorizados de entrada.

- ✓ **Nombre del patrón:** Contexto Seguro (*Security Context*)
  Intención: Provee de un contenedor para los atributos de seguridad y datos relacionados a la ejecución de un contexto, proceso, operación o acción en particular.

- ✓ **Nombre del patrón:** Asociación de seguridad (*Security Association*)
  Intención: Define una estructura la cual provee a cada participante de una comunicación segura con la información que se usará para proteger los mensajes a trasmitir, y con la información que será usada para entender y verificar la protección aplicada al mensaje a recibir.

Para J2EE

- ➢ **Nombre del patrón:** Acción base segura (*Secure Base Action*)
  Intención: Es un patrón para centralizar y coordinar tareas relacionadas con la seguridad dentro de la capa de presentación. Sirve como el punto de entrada primario de la presentación y deberá ser usado o extendido por un "Controlador Frontal".

- ➢ **Nombre del patrón:** Delegación de políticas (*Policy Delegate*)
  Intención: Crea, maneja y administra políticas administrativas de seguridad que rigen cómo los objetos de la capa de EJB son accedidos y transferidos.

- ➢ **Nombre del patrón:** Fachada de servicios seguros (*Secure Service Façade*)
  Intención: Provee de una fachada de sesión que puede contener y centralizar interacciones complejas entre componentes de negocio debajo de una sesión segura. Provee de seguridad dinámica y declarativa para los objetos de negocio del *back-end* en la fachada de sesión. Protege de entidades ajenas que puedan ejecutar directamente servicios ilegales o no autorizados.

- ➢ **Nombre del patrón:** Agente web interceptor (*Intercepting Web Agent*)
  Intención: Ayuda a proteger aplicaciones web basadas en J2EE a través de un agente web que intercepta las peticiones del contenedor web y provee de autenticación, autorización, encripción y capacidades de auditoría.

Ahora bien, una vez definida la estructura general de la aplicación, será necesario proteger el canal de comunicación, para ello hacer uso del siguiente patrón.

✓ **Nombre del patrón:** Canales seguros (*Secure Channels*)
Intención: Para la comunicación sensitiva a través de la red pública, crear un canal seguro encriptado para asegurar la confidencialidad de los datos que transitan.

Para J2EE

➢ **Nombre del patrón:** Pipa segura (*Secure Pipe*)
Intención: Muestra cómo asegurar la conexión entre el cliente y el servidor o entre servidores cuando se conecta a otras instituciones. Agrega valor al requerir autenticación mutua y al establecer confidencialidad y no repudiación entre las partes.

Adicionalmente, si la aplicación maneja sesiones, entonces aplicar el patrón que se muestra a continuación.

✓ **Nombre del patrón:** Sesión segura (*Security Session*)
Intención: Para dar seguimiento a quien usa las funciones y sus correspondientes derechos de acceso, se establece una sesión segura después que el usuario se ha identificado.

Para J2EE

➢ **Nombre del patrón:** Objeto seguro de sesión (*Secure Session Object*)
Intención: Define formas para asegurar la información de la sesión en los EJBs, facilitando el acceso distribuido y la propagación sin interrupciones de un contexto seguro

5.2.2.2 Diseño de controles de autenticación y autorización

La "Guía de Desarrollo de OWASP" [44] nos indica que el objetivo de los controles de autenticación es:

- Vincular una unidad del sistema a un usuario individual mediante el uso de una credencial
- Proveer controles de autenticación razonables de acuerdo al riesgo de la aplicación.
- Denegar el acceso a atacantes que usan varios métodos para atacar el sistema de autenticación.

Conforme a estos criterios, entonces se recomienda aplicar el siguiente patrón de seguridad:

- ✓ **Nombre del patrón:** Autenticador (*Authenticator*)
  Intención: Maneja el problema de cómo verificar que un sujeto es quien dice ser. Recibe las interacciones de un sujeto con el sistema y aplica un protocolo para verificar la identidad del sujeto.

Para J2EE

- ➢ **Nombre del patrón:** Autenticación forzosa (*Authentication Enforcer*)
  Intención: Ilustra como un cliente basado en una aplicación J2EE deberá autenticarse con una aplicación J2EE.

Posteriormente, definir el nivel de acceso al sistema, para ello elegir entre una de las dos siguientes propuestas:

- ✓ **Nombre del patrón:** Acceso total con errores (*Full Access with Errors*)
  Intención: Provee de una vista de máxima funcionalidad al sistema, pero emite un error al usuario cuando trata de usar una función para la cual no está autorizado.

- ✓ **Nombre del patrón:** Límites de acceso (*Limited Access*)
  Intención: Guía al desarrollador para presentar solo las funciones actualmente disponibles al usuario, mientras oculta todo lo demás para el cual le faltan permisos.

En caso de utilizar contraseñas como medio de autenticación, aplicar los siguientes patrones de seguridad:

- ✓ **Nombre del patrón:** Diseño y uso de contraseñas (*Password Design and Use*)
  Intención: Describe mejores prácticas de seguridad para diseñar, crear, manejar y usar componentes de contraseñas para soportar los requerimientos de I&A.

- ✓ **Nombre del patrón:** Bloqueo de cuentas (*Account Lockout*)
  Intención: Protege las cuentas de los clientes de ataques automáticos que adivinan contraseñas, al implementar un límite de intentos incorrectos antes de deshabilitarla.

Para las aplicaciones J2EE adicionalmente se puede aplicar el siguiente patrón.

Para J2EE

- ➢ **Nombre del patrón:** Delegación de firma única (*Single Sign-on (SSO) Delegator*)
  Intención: Describe como construir un agente para delegar el manejo de un sistema legado para una autenticación de una sola vez.

Para aquellas aplicaciones que se relacionan con otros sistemas, y deben utilizar la misma contraseña para autenticar al cliente, entonces utilizar el siguiente patrón.

- ✓ **Nombre del patrón:** Propagación de contraseñas (*Password Propagation*)
  Intención: Requiere que las credenciales individuales de autenticación de un usuario sean verificadas por la base de datos antes que se dé acceso a los datos del usuario.

Para J2EE

- ➢ **Nombre del patrón:** Sincronización de contraseñas (*Password Synchronizer*)
  Intención: Describe como sincronizar principales de manera segura a través de múltiples aplicaciones usando la provisión de servicios.

Una vez diseñados los controles de autenticación, será también necesario delinear los controles de autorización a utilizar. Un control de autorización, tiene por lo tanto los siguientes objetivos [44]:

- Asegurar que únicamente usuarios autorizados puedan realizar acciones permitidas con su correspondiente nivel de privilegio.
- Controlar el acceso a recursos protegidos mediante decisiones basadas en el rol o el nivel de privilegio.
- Prevenir ataques de escalada de privilegios, como por ejemplo utilizar funciones de administrativas siendo un usuario anónimo o incluso un usuario autenticado.

Para lograr este fin, utilizar uno o varios de los siguientes patrones de seguridad:

- ✓ **Nombre del patrón:** Control de acceso basado en roles (*Role-Based Access Control*)
  Intención: Describe cómo asignar derechos basados en las funciones o tareas de la gente en un ambiente en el cual el control de acceso a los recursos computacionales es requerido y donde hay un gran número de usuarios, o una gran variedad de recursos.

- ✓ **Nombre del patrón:** Monitor de consulta (*Reference Monitor*)
  Intención: Fuerza la declaración de restricciones de acceso cuando una entidad activa requiere recursos. Describe como definir un proceso abstracto que intercepta todas las peticiones para los recursos y checa su cumplimiento con autorizaciones.

Para J2EE

- **Nombre del patrón:** Autorización forzosa (*Authorization Enforcer*)
  Intención: Ilustra como la autorización debe ser forzada después de la autenticación del usuario con una aplicación J2EE.

- **Nombre del patrón:** Administración segura del contenedor (*Container Managed Security*)
  Intención: Describe cuándo y cómo declarar información relacionada con seguridad para los EJB en un *deployment descriptor*.

Adicionalmente, para el caso de las aplicaciones J2EE, si la aplicación maneja sesiones, entonces se puede aplicar los siguientes patrones.

Para J2EE

- **Nombre del patrón:** Administador seguro de sesiones (*Secure Session Manager*)
  Intención: Define como crear una sesión segura al capturar la información de la sesión. Usa el patrón de Pipa segura y describe las acciones requeridas para construir una sesión segura entre el cliente y el servidor o entre servidores.

- **Nombre del patrón:** Token para credenciales (*Credential Tokenizer*)
  Intención: Describe como un *token* principal de seguridad puede ser encapsulado y embebido en un mensaje SOAP, enrutado y procesado.

5.2.2.3 Diseño de controles de entrada

El objetivo de los controles de entrada de acuerdo con OWASP [44] es garantizar que la aplicación sea robusta contra todas las formas de ingreso de datos, ya sea obtenida del usuario, de la infraestructura, de entidades externas o de sistemas de base de datos. Para ello, se deben como mínimo considerar los siguientes lineamientos [40]:

- Revisar los datos de entrada para detectar alguno de los siguientes errores:
  - Valores fuera de rango
  - Caracteres inválidos en los campos de datos
  - Datos incompletos o faltantes
  - Exceder los límites superiores e inferiores del volumen de datos
  - Datos no autorizados o inconsistentes
- Revisión periódica del contenido de los campos claves o archivos de datos para confirmar su validez e integridad.

- Inspeccionar los documentos de entrada en caso de cambios no autorizados.
- Procedimientos para responder a los errores de validación.

De esta manera, el patrón de seguridad que nos guía para modelar los controles de entrada dentro de la aplicación es:

✓ **Nombre del patrón:** Filtros para las entradas del cliente (*Client Input Filters*)
Intención: Protege la aplicación de datos manipulados por clientes inseguros, protegiendo así contra clientes corruptos los cuales podrían causar que la aplicación se comporte de una manera no esperada e insegura.

Para J2EE

➢ **Nombre del patrón:** Validador interceptor (*Intercepting Validator*)
Intención: Ofrece mecanismos seguros para validar parámetros antes de invocar una transacción. La validación de los parámetros específicos de la aplicación incluye además la validación de los datos del negocio y sus características.

5.2.2.4 Diseño de controles de proceso

El "Manual de Desarrollo y Adquisición" del FFIEC [38], nos comenta que los controles de procesamiento automatizado ayudan a asegurar que los sistemas procesen y almacenen la información de manera íntegra, así como permiten rechazar, procesar y registrar errores para una revisión posterior y su correción. Algunos ejemplos de este tipo de controles propuestos por ISO 27002:2005 [40] son:

- Controles de sesión o lote, para conciliar los saldos del archivo de datos después de las actualizaciones de la transacción.
- Controles de saldos, para chequear los saldos de apertura comparándolos con los saldos de cierre anteriores; específicamente usar:
    - Controles corrida-a-corrida
    - Totales de actualización del archivo
    - Controles programa-a-programa
- Validación de los datos de entrada generados por el sistema.
- Chequeos sobre la integridad, autenticidad y cualquier otro dispositivo de seguridad de los datos o software cargado o descargado, entre la computadora central y las remotas.
- Comprobar hash de registros y archivos.

- Chequeos para asegurar que los programas se corran en el momento adecuado.
- Chequeos para asegurar que los programas sean corridos en el orden correcto y terminados en caso de una falla, y que se detenga el procesamiento hasta que se resuelva el problema.
- Crear un registro de las actividades involucradas en el procesamiento.

Para este fin, los patrones de seguridad que nos permiten manejar el control del flujo del procesamiento, son:

- ✓ **Nombre del patrón:** Creador controlador de procesos (*Controlled Process Creator*)
  Intención: Permite definir y garantizar derechos de accesos apropiados para un nuevo proceso.
- ✓ **Nombre del patrón:** Fabrica controladora de objetos (*Controlled Object Factory*)
  Intención: Maneja cómo especificar los derechos de procesos con respecto a un nuevo objeto. Cuando un proceso crea un objeto a través de la fábrica, la petición incluye las características del nuevo objeto. Incluye una lista de derechos para acceder al objeto.
- ✓ **Nombre del patrón:** Ejecución del dominio público (*Execution Domain*)
  Intención: Define un ambiente de ejecución para los procesos indicando explícitamente todos los recursos que un proceso puede usar durante su ejecución, así como el tipo de acceso a los recursos.

5.2.2.5 Diseño de controles de salida

El FFIEC [38], define que los controles de salida ayudan a asegurar que el sistema mantenga y distribuya de forma apropiada la información procesada. Con respecto a este fin, ISO 27002:2005 [40] propone los siguientes controles:

- Chequeos de plausibilidad para comprobar si la salida de datos es razonable.
- Control de conciliación para asegurar el procesamiento de todos los datos.
- Proporcionar la información suficiente para que el sistema de procesamiento subsiguiente determine la exactitud, integridad, precisión y clasificación de la información.
- Procedimientos para responder a las pruebas de validación de salidas de datos.
- Crear un registro de las actividades del proceso de validación de datos de salida.

Con respecto a los patrones de seguridad, aquellos que permiten un control para el manejo de salida de información, son los siguientes:

- ✓ **Nombre del patrón:** Información anónima (*Information Obscurity*)
  Intención: Si la información manejada por el sistema es sensitiva, debe ser protegida a través de obscurecer los datos a través de una forma de encripción así como el ambiente alrededor de los datos.

- ✓ **Nombre del patrón:** Almacenamiento de datos en el cliente (*Client Data Storage*)
  Intención: Usa cifrado para almacenar de manera segura datos sensitivos o críticos en el cliente, necesarios para el correcto funcionamiento de la aplicación.

- ✓ **Nombre del patrón:** Almacenamiento encriptado (*Encrypted Storage*)
  Intención: Provee de una segunda línea de defensa contra el robo de datos en los servidores del sistema. Asegura que, aunque se robe información del servidor, lo datos más sensitivos permanecerán a salvo de "ojos entrometidos".

Para J2EE

- ➢ **Nombre del patrón:** Objeto de transferencia ofuscado (*Obfuscated Transfer Object*)
  Intención: Describe formas de proteger los datos de negocio representados en objetos de transferencia y las cuales se pasan dentro y entre las capas lógicas.

5.2.2.6 Diseño de controles de monitoreo y auditoría

Como se comenta en la "Guía de Desarrollo" de OWASP [44], muchas industrias son requeridas por medio de requisitos legales y regulatorios, que sus sistemas y procesos sean:

- Auditables, todas las actividades que afectan el estado de un usuario o balances deben ser formalmente rastreables.
- Trazables, es posible determinar donde ocurre cada actividad en todas las capas de una aplicación.
- Alta integridad, los logs no pueden ser sobrescritos o modificados por usuarios locales o remotos.

Es así como las aplicaciones bien escritas generan logs de doble propósito para rastrear actividades para la auditoria y el monitoreo, lo que permite fácilmente seguir una transacción sin mucho esfuerzo.

Para ello, los patrones de seguridad que implementan el monitoreo en una aplicación y detección de anomalías, son los siguientes.

- ✓ **Nombre del patrón:** Puesto de control en el sistema (*Checkpointed system*)
  Intención: Permite estructurar un sistema de tal manera que su estado pueda ser recobrado o restaurado a un estado conocido válido en caso que un componente falle.

- ✓ **Nombre del patrón:** Comparar-Verificar un sistema tolerante a fallas (*Comparator-checked fault-tolerant system*)
  Intención: Permite que una falla independiente de un componente pueda ser detectada rápidamente para que un independiente y único componente al fallar, no cause la falla del sistema.

- ✓ **Nombre del patrón:** Aserciones seguras (*Secure Assertion*)
  Intención: Distribuye verificaciones específicas de la aplicación a través del sistema, mediante el uso de aserciones (*assertions*) seguras que mapean las aserciones convencionales a un sistema de detección de intrusiones.

Para J2EE

- ➢ **Nombre del patrón:** Administración dinámica de servicios (*Dynamic Service Management*)
  Intención: Provee de una instrumentación ajustable dinámicamente de componentes de seguridad para monitorear y activar la administración de objetos de negocio.

Con respecto al manejo de bitácora, los patrones que me permiten implementar esta funcionalidad son:

- ✓ **Nombre del patrón:** Bitácora para auditoria (*Log for Audit*)
  Intención: Vincula las bitácora a la auditoría, asegurando que estas se encuentren configuradas con la auditoría en mente y que sea entendida como parte integral de un registro efectivo.

Para J2EE

- ➢ **Nombre del patrón:** Bitácora segura (*Secure Logger*)
  Intención: Define como capturar eventos específicos de la aplicación y excepciones de manera segura y confiable para soportar auditorías en seguridad.

- ➢ **Nombre del patrón:** Interceptor auditor (*Audit Interceptor*)
  Intención: Trabaja junto con el patrón de Bitácora segura. Permite administrar y manejar aspectos de registro y auditoria en el *back-end*.

5.2.3 Desarrollo

Llegado este punto se empieza a codificar los algoritmos y estructuras de datos, definidos en las etapas anteriores, en el correspondiente lenguaje de programación y/o para un determinado sistema gestor de bases de datos.

5.2.3.1 Configuración del ambiente de desarrollo

En esta fase se recomienda facilitar y estar disponible un ambiente de desarrollo integrado (IDE) que permita a los programadores, codificar y compilar programas de manera interactiva con una computadora o un servidor. A través de esta utilidad, los programadores pueden entrar, modificar y eliminar los códigos de programación, así como como compilar y almacenar los programas (fuente y objeto).

Adicionalmente el FFIEC [38] propone el uso de un control de versiones para el código, el cual sistemáticamente retiene copias cronológicas del código de los programas y la documentación. Lo cual facilita la identificación rápida de errores y cambios en el sistema.

5.2.3.2 Desarrollo seguro y su control

Se debe de llevar métodos y técnicas de programación que permitan aumentar la calidad de las actividades de programación y las futuras capacidades de mantenimiento. Para ello el FFIEC [38] recomienda el uso de estándares de desarrollo y el control de librerías.

Con referencia al uso de estándares de desarrollo, utilizar aquellos que tengan un reconocimiento o prestigio internacional, como los descritos durante la sección 2.2.2 "Estándares y guías de desarrollo". Adicionalmente implementar los patrones de seguridad seleccionados durante la fase de diseño de la aplicación.

Recordemos, que los patrones de seguridad son "mejores prácticas" que presentan una solución genérica "bien probada" a un problema recurrente de seguridad. Por lo que durante el desarrollo, seguir las recomendaciones que los patrones ofrecen para su implementación, así como complementar los mismos con los estándares y guías de desarrollo seguros.

Adicionalmente, durante el desarrollo del sistema, controlar el uso de librerías. Las librerías son colecciones de programas con rutinas y módulos reusables. Cuidar que las mismas provengan de agentes confiables y se utilice la versión correcta para la aplicación.

5.2.3.3 Revisión del código

El código fuente también provee de un indicador de confiabilidad. Aquel código que ha sido sujeto a revisiones independientes de seguridad es generalmente más confiable del que no ha sido analizado. Así la revisión del código fuente puede ser manual o automática. [16]

La revisión automática típicamente busca errores comunes de código que pueden tener implicaciones de seguridad. Para este tipo de revisión, se sugiere el uso de las herramientas denominadas de "Caja Blanca", las cuales precisamente buscan errores en el código que pueden generar una vulnerabilidad en la aplicación y la cual puede ser explotada por las actividades maliciosas.

En cuanto a la revisión manual, esta puede ser más detallada para tratar de revisar que se hayan seguido "buenas prácticas" de programación, sin embargo, puede no ser tan confiable dado su naturaleza tediosa y subjetiva.

5.2.4 Pruebas

El objetivo de esta etapa es garantizar que el sistema ha sido desarrollado correctamente, satisfaciendo los requerimientos del sistema sin errores de diseño y/o programación. Además, se verificará que los controles de seguridad previamente establecidos permiten contrarrestar los efectos de ataques externos. Así como corroborar que el número de vulnerabilidades dentro de la aplicación es mínimo o inexistente.

5.2.4.1 Configuración del ambiente de prueba

Se deberá configurar un ambiente de pruebas el cual permita efectuar pruebas o casos de pruebas que permitan validar el funcionamiento del sistema en un ambiente equiparable o igual al ambiente de producción, con la finalidad de corregir aquellos errores omitidos durante la revisión del código de la etapa anterior. Para ello, el patrón que nos permite configurar un ambiente de pruebas es:

- ✓ **Nombre del patrón:** Pruebas con un servidor de pruebas (*Test on a Staging Server*)
  Intención: Mientras las pruebas unitarias pueden realizarse en las máquinas de desarrollo, una prueba de integral del sistema debe realizarse en máquinas similares a los servidores de producción. Lo cual también evita la configuración excesiva del sistema al momento de su puesta en producción.

5.2.4.2 Inspección de la seguridad en los módulos de la aplicación

Realizar pruebas a nivel de cada módulo de la aplicación (aislado del resto). En el que se deberá revisar que el modulo cumpla con los requerimientos funcionales previamente establecidos para el mismo. Revisar tanto la entrada de datos, como el procesamiento y la salida de información.

5.2.4.3 Inspección de la seguridad integral de la aplicación

Se deberá realizar pruebas de integración del sistema. Estas pruebas deberán asegurar que los controles del negocio se traduzcan correctamente en controles de aplicación de manera que el procesamiento sea exacto, completo, oportuno, aprobado y auditable. Los aspectos que se consideran especialmente son: mecanismos de autorización, integridad de la información, control de acceso, respaldo y diseño de pistas de auditoría. Además, según sea la naturaleza del proyecto en cuestión, se podrán efectuar pruebas adicionales para los requerimientos no funcionales, como por ejemplo de rendimiento.

Se deberá abordar también, la seguridad de las aplicaciones y los requerimientos de disponibilidad en respuesta a los riesgos identificados, de acuerdo con la clasificación de los datos, la arquitectura de seguridad en la información de la organización y el perfil de riesgo. Los asuntos a considerar incluyen derechos de acceso y administración de privilegios, protección de información sensible en todas las etapas, autenticación e integridad de las transacciones y recuperación automática.

Adicionalmente, con respecto a la verificación de la seguridad de la aplicación, la subjetividad que presenta el empleo de metodologías cualitativas, puede ser subsanada en cierta medida, empleando tecnologías que validen cuáles son las posibles debilidades o vulnerabilidades que tiene una aplicación. Para ello, utilizar las herramientas denominadas de "caja blanca", las cuales permiten revisar la aplicación a partir de su código fuente; así como herramientas de "caja negra", mismas que se emplean para revisar la aplicación en producción.

Las herramientas de "caja negra", normalmente son utilizadas para atacar la aplicación que se utilizará en producción, las cuales generalmente son realizadas por un equipo de pruebas el cuál simula las actividades de un atacante. Con respecto a este tema, el siguiente patrón de seguridad nos ayuda a definir un equipo de pruebas con estas características.

- ✓ **Nombre del patrón:** Pruebas para atacar al diseño (*Red Team the Design*)
  Intención: Afecta la evaluación de la seguridad de una aplicación al promover el uso de equipos rojos (los cuales examinan al sistema desde una perspectiva del atacante), durante las fases tempranas del desarrollo cuando es posible arreglar los problemas identificados.

Estos equipos pueden adicionalmente hacer uso de los denominados "patrones de ataque", los cuales nos ayudan a determinar que vulnerabilidades pueden afectar a nuestro sistema. Estos patrones, están conformados por la descripción de los métodos utilizados para la explotación del software/hardware, con la desviación particular (desde el punto de vista de diseño) que los orienta a un objetivo relativamente "destructivo" y que son un elemento clave para identificar y comprender las diferentes perspectivas en las que una amenaza se puede convertir en una vulnerabilidad. [60]. De esta manera, los patrones de ataque que nos ayudarán en las pruebas al sistema son los siguientes:[11]

- Abuso de funcionalidad
- Explotación de Autenticación
- Explotación de privilegios
- Ataques a las estructuras de datos
- *Spoofing*
- Abuso de recursos
- Inyección de código y SQL
- Manipulación de recursos

5.2.5 Producción

Durante la fase de producción o implementación, se establece y se prueba la operación efectiva del nuevo sistema de información. Las pruebas de aceptación por parte del usuario final se llevan a cabo en este entorno, finalmente se certifica y pone en producción el sistema desarrollado.

5.2.5.1 Configuración de la seguridad en el ambiente de producción

Establecer procedimientos para el control de la instalación del software en los sistemas operacionales. Para ello, tomar en consideración tanto la configuración lógica como los recursos humanos, es decir, especificar quienes serán los responsables, cómo se verificará el paso a este ambiente y el procedimiento de vuelta atrás si se experimentan problemas.

[11] Para mayor referencia favor de consultar el sitio web http://capec.mitre.org/data/index.html de CAPEC la organización de dicada a la enumeración y clasificación de los patrones de ataque [60]

Así, cuando un sistema inicia su operación, deberán deshabilitarse, en el ambiente de producción, las cuentas de operación del personal que desarrolló la aplicación. Además, tener respaldo de los programas fuente utilizados para la generación de la versión de producción del mismo así como de la documentación. Adicionalmente, la OWASP [44] nos propone los siguientes puntos a tomar en consideración:

- Desactivar todas las opciones innecesarias de manera predeterminada.
- Asegurar que todas las opciones y configuraciones para cada función están inicialmente configuradas para ser la elección más segura posible.
- Inspeccionar el diseño para comprobar si las elecciones menos seguras pudieran ser diseñadas de otra manera.
- No confiar en características instaladas opcionalmente en el código base.
- No configurar nada como preparación para una característica opcional de implantación.

5.2.5.2 Certificación de la aplicación

Realizar pruebas de aceptación por parte del usuario. Una vez obtenido el visto bueno del mismo, certificar y acreditar la implementación efectuada. Se tendrá que involucrar a la gerencia sobre la efectividad del sistema para cumplir los objetivos definidos y para establecer un nivel apropiado de control.

Los gerentes deberán asegurar que los requerimientos y criterios de aceptación de los sistemas nuevos estén claramente definidos, aceptados, documentados y probados. Los sistemas de información nuevos, las actualizaciones y las versiones nuevas deberán migrar a producción después de obtener la aceptación formal. Considerar los siguientes ítems antes de proporcionar la aceptación formal [40]:

- El desempeño y los requerimientos de capacidad de la computadora.
- Procedimientos para la recuperación tras errores y reinicio, y planes de contingencia.
- Preparación y prueba de los procedimientos de operación rutinarios para estándares definidos.
- El conjunto de controles de seguridad acordados y aceptados.
- Procedimientos manuales efectivos.
- Arreglos para la continuidad del negocio.

- Evidencia que la instalación del sistema nuevo no afectará adversamente los sistemas existentes, particularmente en las horas picos del procesamiento, como fin de mes.
- Evidencia que se está tomando en consideración el efecto que tiene el sistema nuevo en la seguridad general de la organización;
- Capacitación para la operación o uso de los sistemas nuevos.
- Facilidad de uso, ya que esto afecta el desempeño del usuario y evita el error

Así, la certificación, es un proceso mediante el cual una organización realiza una alta evaluación en comparación con el estándar de controles técnicos, operativos y de gestión en un sistema de información. Los resultados de una certificación se utilizan para volver a evaluar los riesgos y actualizar el plan de seguridad del sistema, estableciendo las bases para que una autoridad tome una decisión de acreditación.

La acreditación, es la decisión de la gestión oficial para autorizar la operación de un sistema de información y aceptar explícitamente el riesgo de las operaciones, activos e individuos de la organización, basándose en la implementación de un conjunto de requerimientos y controles de seguridad acordados. La acreditación de seguridad ofrece un formulario de control de calidad y desafía a los gerentes y al personal técnico en todos los niveles para que implementen los controles de seguridad más efectivos posibles en un sistema de información, determinados requerimientos de misiones y limitaciones técnicas, operativas y de costo/cronograma.

Finalmente, los sistemas o programas deberán registrarse ante el Registro Público de Derechos de Autor, cuando se concluya que tal registro es conveniente con base en el Análisis de Riesgos.

5.2.5.3 Distribución e instalación segura de la aplicación

Después de probar exitosamente todo el sistema, el sistema estará listo para migrar al ambiente de producción. Asegurar que los programas han sido probados y refinados totalmente, los procedimientos programados y el cronograma para la ejecución de procesos en producción están instalados, todos los datos necesarios han sido convertidos y cargados en el nuevo sistema y los usuarios han desarrollado procedimientos y han sido totalmente capacitados en el uso del nuevo sistema. Se determinará entonces una fecha de migración.

5.2.6 Mantenimiento

A partir del momento que una aplicación entra en producción, también entrará en la etapa de mantenimiento, la cual supondrá pequeñas operaciones tanto de corrección como de mejora de la aplicación (por ejemplo mejora del rendimiento), así como otras de mayor importancia, fruto de la propia evolución (como nuevas opciones para el usuario debidas a nuevas operaciones contempladas para el producto).

5.2.6.1 Monitoreo de la seguridad y disponibilidad de las aplicaciones

Las instituciones financieras deberán asegurar que sus sistemas sean confiables con el paso del tiempo, debido a ello, la OWASP [44] propone que los objetivos de esta etapa aseguren que:

- Los productos son correctamente mantenidos después de su despliegue
- Minimizar el área de ataque atreves del ciclo de vida de producción
- Los defectos de seguridad son arreglados correctamente y en un tiempo adecuado

Más específicamente, el monitoreo de la seguridad y disponibilidad de las aplicaciones de acuerdo a las normas ISO 27002:2005 [40], define que después de que el sistema ha sido establecido en el ambiente productivo, se debe efectuar una revisión posterior a la implementación, previo a un periodo de tiempo determinado para que el sistema se estabilice, de este modo habrá oportunidad para que aparezca cualquier problema significativo. En este sentido:

- Determinar si se lograron los objetivos y requerimientos del sistema
- Determinar si se está midiendo y analizando el costo-beneficio identificado en el estudio de factibilidad y viabilidad
- Revisar las solicitudes de cambio a programas para evaluar el tipo de cambios que requiere el sistema
- Revisar los controles integrados del sistema para asegurarse que se encuentren operando conforme a l diseño
- Revisar los registros de error de operación para determinar si hay algún problema de recursos o de operación inherente en el sistema.
- Revisar los controles de balance de entrada, salida y además informes para verificar que el sistema esté procesando los datos correctamente.

5.2.6.2 Control de cambios

Conforme al ISO/IEC 27002:2005 [40] cuando se cambian los sistemas que se encuentran en operación, se deberá revisar y probar las aplicaciones para asegurar que no exista un impacto adverso en las operaciones o en la seguridad. Para minimizar el riesgo de corrupción de los sistemas operacionales, se deberán considerar los siguientes lineamientos para controlar los cambios:

- La actualización del software operacional, aplicaciones y bibliotecas de programas sólo será realizada por administradores capacitados con la apropiada autorización gerencial.
- Los sistemas operacionales sólo mantendrán códigos ejecutables aprobados.
- El software de las aplicaciones y el sistema de operación sólo se debiera implementar después de una prueba extensa y satisfactoria; las pruebas debieran incluir pruebas de utilidad, seguridad, efectos sobre los sistemas y facilidad para el usuario; y se debieran llevar a cabo en sistemas separados, además asegurar que se hayan actualizado todas las bibliotecas fuente correspondientes del programa;
- Utilizar un sistema de control de configuración para mantener el control de todo el software implementado, así como la documentación del sistema;
- Establecer una estrategia de "regreso a la situación original" (rollback) antes de implementar los cambios;
- Mantener un registro de auditoría de todas las actualizaciones a las bibliotecas del programa operacional;
- Mantener las versiones previas del software de aplicación como una medida de contingencia;
- Archivar las versiones antiguas del software, junto con toda la información requerida y los parámetros, procedimientos, detalles de configuración y software de soporte durante todo el tiempo que se mantengan la data en archivo.

5.2.6.3 Plan de contingencia

El plan de contingencia o de recuperación de desastres, tendrá como objetivo restaurar el servicio de cómputo en forma rápida, eficiente y con el menor costo y pérdidas posibles. Deberá describir los pasos a seguir luego de un desastre para recuperar, aunque sea en parte, la capacidad funcional del sistema. Se entiende por recuperación, "tanto la capacidad de seguir trabajando en

un plazo mínimo después de que se haya producido el problema, como la posibilidad de volver a la situación anterior al mismo, habiendo reemplazado o recuperado el máximo posible de los recursos e información". Para lograr este fin, algunos puntos a considerar son:

- La integridad de la base de datos, número total de registros desde que la base de datos fuente es transferida a la nueva base asumiendo que el número de campos es el mismo.
- La integridad de datos, es decir no sean alterados manualmente, mecánicamente o electrónicamente por personas o programas.
- El almacenamiento y seguridad de los datos sometidos a conversión, es decir tener una copia de seguridad de los datos antes de la conversión para referencias futuras o cualquier emergencia que pudiera surgir como consecuencia de la gestión de programas de conversión de datos.
- Continuidad, es decir la nueva aplicación debe poder continuar con los registros más nuevos como adición y ayudar en asegurar una total continuidad del negocio.
- La última copia de datos antes de conversión desde la antigua plataforma y la primera copia de los datos después de la conversión en la nueva plataforma se deben mantener en los archivos para cualquier referencia futura.

5.2.7 Retiro

El sistema computacional es retirado de producción una vez que la transición a un nuevo sistema se ha completado o bien porque éste es ahora obsoleto.

5.2.7.1 Plan de retiro de la aplicación

Conforme a NIST [39], se deberá construir un plan de transición o retiro, el cual asegure que todo el personal involucrado sea consciente de los planes futuros para el sistema y la información. Este plan deberá contener a todos los componentes críticos, servicios e información, identificando los pasos necesarios, decisiones para mitigar los riesgos que pueden ocasionarse durante esta etapa de transición. Así se sugiere:

- Documentar el retiro
- Notificar el retiro al usuario
- Trabajar con los sistemas en paralelo y estabilizar
- Notificar la terminación del retiro

5.2.7.2 *Preservación de la información*

Cuando se debe de preservar la información, las organizaciones deberán considerar los métodos y tecnologías que requerirán para recuperar la información en el futuro. Debido a que la tecnología pudiera no estar disponible en el futuro. Además los requerimientos legales en cuanto a la conservación de la información, deberán ser considerados al retirar el sistema de producción [39].

5.2.7.3 *Medidas de limpieza*

Asegura que los datos delicados se borren del anterior sistema. De igual manera, destruir los medios de información digital antes de su retiro o liberación para su reúso afuera de la organización. Con el fin de prevenir que individuos no autorizados, obtengan acceso y usen la información contenida en los medios.

5.2.7.4 *Retiro y cierre de la aplicación*

El retiro del software deberá estar acorde con los términos de la licencia y las regulaciones aplicables a la institución. En situaciones donde el hardware que contiene información crítica no pudo ser limpiado correctamente, entonces se podrá proceder a su destrucción. Adicionalmente se recomienda:

- Elaboración del plan de retiro: cese de soporte temporal o permanente.
- Documentar el impacto de retiro.
- Notificación a usuarios del retiro.
- Conducir actividades paralelas.
- Notificar que la terminación está en proceso.
- Asegurar que los datos anteriores permanezcan accesibles.
- Archivar la aplicación y documentación existente.
- Responsabilidad de futuros aspectos residuales.
- Plan de transición a un nuevo producto de software.

5.3 Análisis y validación de la metodología

Para corroborar que la metodología propuesta permite la creación de aplicaciones financieras con un mínimo de vulnerabilidades, los pasos a seguir se dividirá en dos etapas:

5.3.1 Evaluación de la metodología en una aplicación financiera

Objetivo: Corroborar que el uso de la metodología permite disminuir el número de vulnerabilidades de una aplicación financiera, y con ello incrementar el nivel de seguridad de la aplicación

Requisitos: En un primer acercamiento y siempre cuando las condiciones así lo permitan, se deberá utilizar para esta prueba la misma aplicación financiera; primero cuando esta no ha aplicado la metodología (A), y luego comparándola con la misma aplicación pero desarrollada empleando la metodología propuesta (A'). Esto tiene el propósito de realizar la evaluación de la aplicación con el mismo nivel de complejidad, funcionalidades y módulos semejantes.

Pasos para la evaluación:

Para la aplicación financiera A.

1. Desarrollar la aplicación financiera siguiendo los pasos del ciclo de vida que actualmente la empresa ejecuta para el desarrollo de sus aplicaciones.
2. Al llegar a la etapa de pruebas dentro de la metodología original, se deberán realizar las siguientes pruebas, las cuales se describen dentro de la metodología propuesta:
    a. Prueba de caja blanca
    b. Prueba de caja negra
3. Para el cuadro de resultados (Véase Tabla 5.2), llenar si durante las pruebas realizadas se identificó algunos de los eventos de riesgos enlistados dentro de la tabla de resultados. En los casos en que no aplique esta prueba escribir N/A

Para la aplicación financiera A'.

1. Desarrollar la aplicación financiera utilizando la metodología propuesta. Con referencia al uso de la metodología favor de referirse a la sección 5.1.2 "Aplicación de la metodología".
2. Los resultados obtenidos se transcribirán al cuadro de resultados (Véase Tabla 5.2), para las actividades de:
    a. Prueba de caja blanca
    b. Prueba de caja negra

Si alguna de las pruebas no aplican, favor de escribir N/A

Eventos de riesgo	Aplicación A		Aplicación A'	
	Prueba de caja blanca	Prueba de caja negra	Prueba de caja blanca	Prueba de caja negra
1. Inyección				
2. Secuencia de Comandos en Sitios Cruzados				
3. Perdida de Autenticación y Gestión de Sesiones				
4. Referencia Directa Insegura a Objetos				
5. Falsificación de Peticiones en Sitios				
6 Configuración Defectuosa de Seguridad				
7. Almacenamiento Criptográfico Inseguro				
8. Falla de Restricción de Acceso a URL				
9. Falta de protección en la Capa de Transporte				
10. Redirecciones y reenvios no validados				
11. Control inadecuado de entradas				
12. Control inadecuado de los procesos				
13. Control inadecuado de salidas				
Total de eventos de riesgo identificados				
Eventos de riesgo aplicables				
Porcentaje de eventos de riesgo				

Tabla 5.2 – Cuadro de resultados

Evaluación

Una vez que todas las pruebas hayan concluido y el cuadro de resultados (Véase Tabla 5.2), se encuentre completo, se compararán entre sí los resultados obtenidos para ambas aplicaciones. Esto involucra:

1. Llenar el renglón "Total de eventos de riesgo identificados", para lo cual se sumarán el número de eventos de riesgos identificados en cada una de los renglones de las tabla.
2. Llenar el renglón "Eventos de riesgo aplicables". Este rubro nos sirve para valorar más objetivamente los eventos de riesgo identificados, dado que algunos de los resultados que arrojan las pruebas de caja blanca o caja negra, pueden no ser aplicables, dado que la aplicación establece controles que permiten mitigar el riesgo identificado.
3. Llenar el renglón "Porcentaje de eventos de riesgo" a través de la formula

$$P = \left(\frac{n}{N} * 100\right) - 100 \tag{5.1}$$

Dónde:

P: Es el porcentaje de riesgos que la metodología redujo

n: Son el número obtenido en el renglón "eventos de riesgo aplicables" para A'

N: Es el número de eventos de riesgo aplicables para A.

4. La valoración a seguir será la siguiente:

a) Si el porcentaje de riesgos para la aplicación "A'" (a la cual se le aplicó la metodología propuesta) es menor como mínimo en 20 puntos porcentuales al porcentaje de la aplicación "A" (a la cual se le aplicó la metodología propuesta), para las dos columnas (prueba de caja blanca y prueba de caja negra). Entonces se considera las pruebas de seguridad como satisfactorias y la metodología es aprobada
b) Por lo consiguiente, dado el porcentaje de riesgos para alguna de las dos columnas (caja negra y caja blanca), si el resultado de A' es mayor o igual, o menor a 20 puntos porcentuales, entonces la metodología no es aprobada.

5.3.2 Validación general de la metodología

Objetivo: Verificar a nivel general, que la presente metodología permite la creación de aplicaciones financieras seguras dentro del sector financiero.

Requisitos: En un primer acercamiento y siempre cuando las condiciones así lo permitan, se deberá utilizar para esta prueba una muestra de aproximadamente 7 aplicaciones de tipo financiero, las cuales de preferencia pertenezcan a instituciones financieras o áreas de la empresa diferentes. Donde no más de dos de estas aplicaciones se realicen dentro de la misma área de la institución y su nivel de complejidad sea diferente. De no contar con la oportunidad de aplicar la metodología en instituciones o áreas diferentes, entonces el número de muestra aumentará a 10.

Pasos para la evaluación:

1. Para cada una de las aplicaciones financieras dentro de la muestra, desarrollar la misma utilizando la metodología propuesta. Con referencia al uso de la metodología favor de referirse a la sección 5.1.2 "Aplicación de la metodología".
2. Durante la etapa de pruebas, anotar en el cuadro comparativo el porcentaje de eventos de riesgo que la metodología produjo para:
    a. Prueba de caja blanca
    b. Prueba de caja negra

Aplicación	Pruebas de caja blanca	Pruebas de caja negra
Aplicación 1		
Aplicación 2		
...		
Aplicación n		
Porcentaje de eventos de riesgo total		

Tabla 5.3 – Cuadro comparativo entre aplicaciones

Evaluación

Una vez que todas las pruebas hayan concluido y el cuadro comparativo (Véase Tabla 5.3) se encuentre completo, se compararán entre sí los resultados obtenidos para todas las aplicaciones dentro de la muestra. Esto involucra:

1. Llenar el renglón "Porcentaje de eventos de riesgo total" a través del promedio simple del porcentaje de eventos de riesgo anotado para cada aplicación en cada una de las pruebas (caja blanca y caja negra) realizadas.
2. La valoración a seguir será la siguiente:

a) Si el porcentaje de eventos de riesgos para todas las aplicaciones es menor o igual a 20 puntos porcentuales, entonces la metodología es aprobada y se considera que la misma es aplicable para construir aplicaciones financieras seguras dentro del sector financiero mexicano.

b) Por lo consiguiente, si el porcentaje para alguna de las aplicaciones dentro de la muestra es mayor a 20 puntos porcentuales. Entonces se considera que la metodología tiene un alcance parcial para el desarrollo de aplicaciones financieras. Y se debe someter a su revisión.

Cabe mencionar que independientemente del resultado de la aplicación de esta valoración, es conveniente que la metodología se vaya revisando y actualizando. Para poder certificar que la misma cumple con las últimas disposiciones de la legislación existente dentro del sector financiero mexicano, y que además involucre las nuevas tecnologías computacionales que continuamente salen en el mercado.

CAPÍTULO 6

6 Caso de Estudio

Con el fin de evaluar la metodología para el desarrollo de aplicaciones financieras seguras, y de acuerdo con lo especificado durante la sección 5.3.1 "Evaluación de la seguridad en una aplicación financiera", el caso de estudio consistió en crear dos aplicaciones financieras. En donde la primera se realizó bajo el uso de una metodología de desarrollo de software común, y la segunda, incorporando las actividades y patrones de seguridad descritos durante la metodología propuesta en el presente trabajo.

De esta manera, el presente capitulo comenzará con la descripción de los aspectos que influyeron en la selección del caso de estudio, para posteriormente presentar los aspectos generales del mismo. Los aspectos generales, se encuentran enfocados a ofrecer una descripción general de la aplicación, mismos que fueron obtenidos al aplicar la metodología de desarrollo de software original. Finalmente, durante la implementación de la metodología propuesta, se presentarán los resultados conseguidos al aplicar las actividades descritas en la presente metodología.

6.1 Selección del caso de estudio

Conforme a los criterios establecidos en la sección 5.1.1 "Pre-requisitos para la aplicación de la metodología", se decidió desarrollar un portal bancario en línea, para un banco denominado "Banco Comercial de México", el cual permita la transferencia de información confidencial por internet, como son los pagos a cuentas, tarjetas de crédito, y la administración de las mismas.

Además, se inferirá que dado la naturaleza bancaria de la institución, dicha aplicación se desarrollará en un ambiente de trabajo donde se cuenta con políticas de seguridad previamente establecidas y una infraestructura de seguridad.

Así, las características que justifican la implementación y el uso de la metodología propuesta son las siguientes:

Nombre de la aplicación: Banca en Línea.

Institución: Banco Comercial de México

Pre – requisitos	Características
Aplicación perteneciente al sector financiero mexicano	Si
Envío y/o recepción de información de carácter confidencial a través de la red	Si
Trabajo en ambiente Web	Si
Gestión organizacional para la seguridad	Si
Infraestructura de seguridad	Si
Manejo de estándares	No
Metodología de desarrollo comercial	Si

Tabla 6.1 – Cumplimiento de Pre-requisitos para el uso de la metodología propuesta

Por lo tanto, la presente aplicación se tomará favorablemente como Caso de Estudio para la corroboración de la metodología propuesta para el desarrollo de aplicaciones seguras.

6.2 Aspectos generales del caso de estudio

El Banco Comercial de México obtuvo su licencia de la Comisión Nacional Bancaria y de Valores para constituirse como una Institución de Banca Múltiple en México en diciembre del 2010.

Actualmente, la institución busca ampliar sus horizontes, consolidándose como una de las principales entidades financieras y bancarias a nivel nacional. Acorde con lo anteriormente expuesto, el objetivo inicial es ofrecer una amplia gama de servicios financieros de gran calidad a las personas físicas y morales. Para el aseguramiento de su objetivo, dentro de las ventajas competitivas que busca desarrollar dentro de su estrategia integral de negocios, es contar con un servicio al cliente de primer nivel y una banca electrónica sólida, poderosa y segura, que respalde todas las operaciones financieras que ofrece la empresa de manera transparente, integra y confiable.

De esta manera, El Banco Comercial de México desea desarrollar un portal Web que no solo permita a sus clientes conocer los servicios que ofrece la institución, sino también contar con el respaldo de una robusta banca electrónica, en donde el usuario podrá consultar y administrar el estado de sus cuentas así como realizar operaciones financieras de manera fácil y segura.

6.2.1 Aspectos del desarrollo sin la aplicación de la metodología propuesta

En un primer acercamiento, El Banco Comercial de México, inició el desarrollo de su portal y banca en línea haciendo uso de la metodología RUP para el desarrollo de sistemas. De este estudio, se crearon las siguientes vistas arquitectónicas que nos servirán como base para comparar más fácilmente el diseño actual con el resultante de la aplicación de aspectos de seguridad a través del uso de la metodología establecida en el presente documento.

6.2.1.1 Vista de casos de uso

Mediante un levantamiento de requerimientos funcionales y análisis de los mismos, el equipo de desarrollo determinó los siguientes perfiles de clientes:

- **Clásico.** El usuario no tiene derecho a utilizar los servicios de Transferir entre cuentas y Transferir a otros bancos. Además el monto máximo a transferir es de $10,000.00.
- **Avanzado.** El usuario podrá utilizar todos los servicios pero el monto máximo a transferir será de $50,000.00.
- **Premium.** El usuario podrá utilizar todos los servicios sin límite en el monto máximo.

Los casos de uso identificados fueron los siguientes:

- **Ingresar a la Banca en Línea.** El usuario se registrará con el sistema a través de un nombre de usuario y contraseña, para poder entrar a los servicios de banca electrónica.
- **Consultar Saldos y movimientos.** Se mostraran los saldos finales de las cuentas y/o productos que posea el cliente. Además el usuario podrá revisar los movimientos que tuvo en una determinada cuenta.
- **Administrar tarjetas.** Para hacer uso de las opciones de: Transferir entre cuentas, Transferir a otros bancos y Pagar tarjetas de crédito, el usuario deberá primero registrar la cuenta a la que desea transferir. Adicionalmente, se podrá actualizar la descripción de la cuenta de referencia y el monto máximo a transferir una vez dada de alta la misma.
- **Transferir entre cuentas.** El usuario podrá realizar transacciones de traspaso de dinero entre las diferentes cuentas del Banco Comercial de México que tenga registradas.
- **Transferir a otros bancos.** El usuario tendrá la posibilidad de realizar traspaso de dinero a las cuentas de otros bancos.
- **Pagar tarjetas de crédito.** El usuario podrá pagar sus tarjetas de crédito.
- **Salir de la Banca en Línea.** El usuario cerrará su sesión del banco en línea.

Finalmente, el esquema de casos de uso se muestra a continuación:

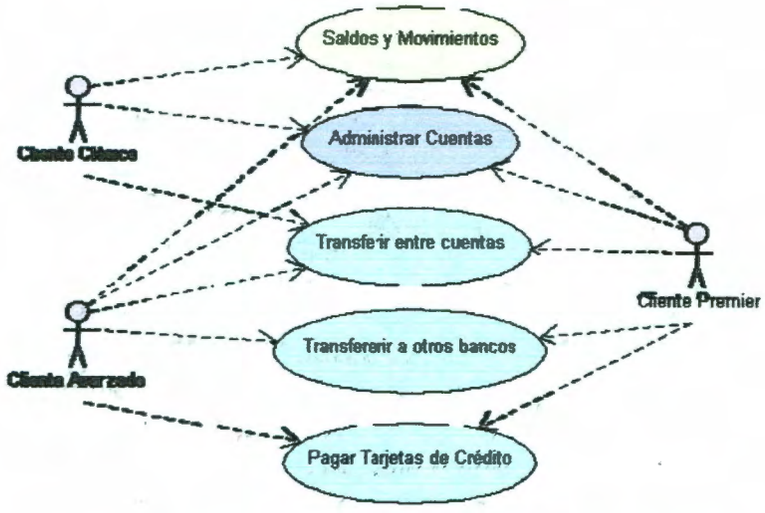


Figura 6.1 –Vista de casos de uso de la aplicación

6.2.1.2 Vista lógica

Con respecto al diseño de clases, se decidió construir la aplicación bajo el enfoque MVC. Utilizando para ello principalmente el patrón de diseño Service to Worker y DAO para la capa de Integración de recursos.

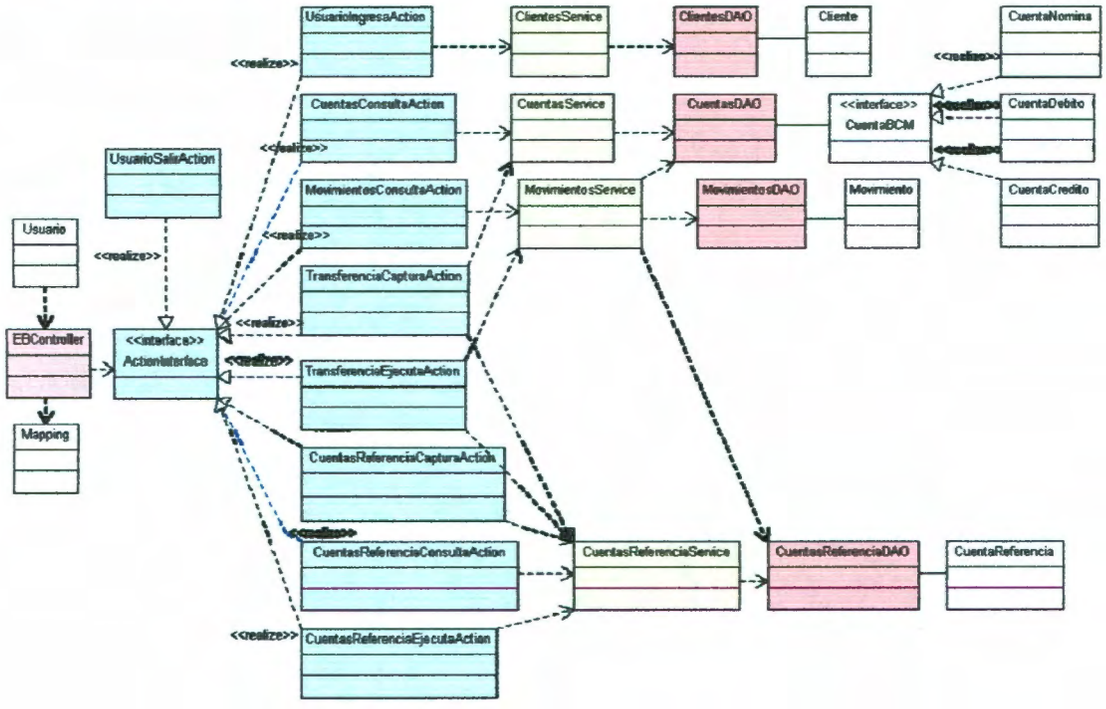


Figura 6.2 –Diagrama de clases de la aplicación

6.2.1.3 *Vista de datos*

Dado el análisis de la información que el sistema manejará, el esquema de Bases de Datos obtenido, se muestra a continuación.

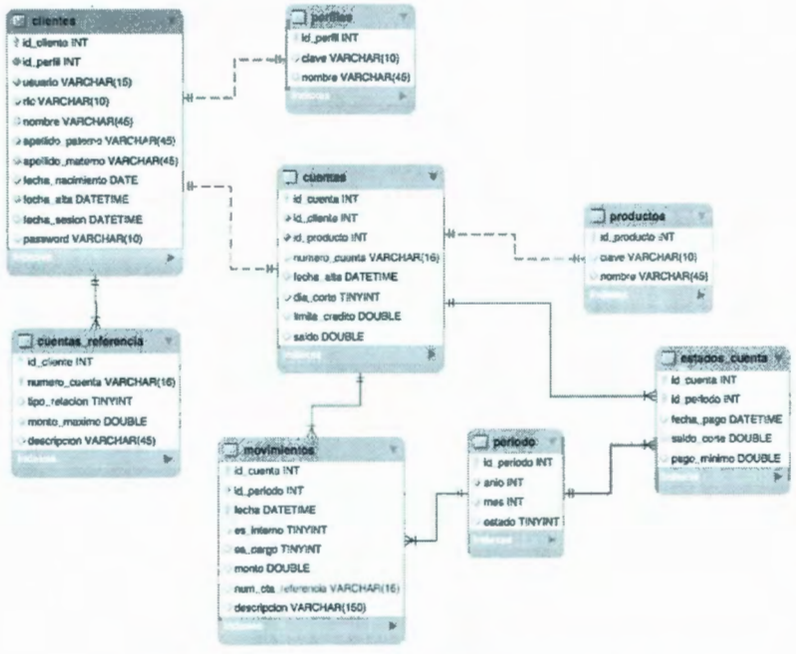


Figura 6.3 –Esquema de base de datos de la aplicación

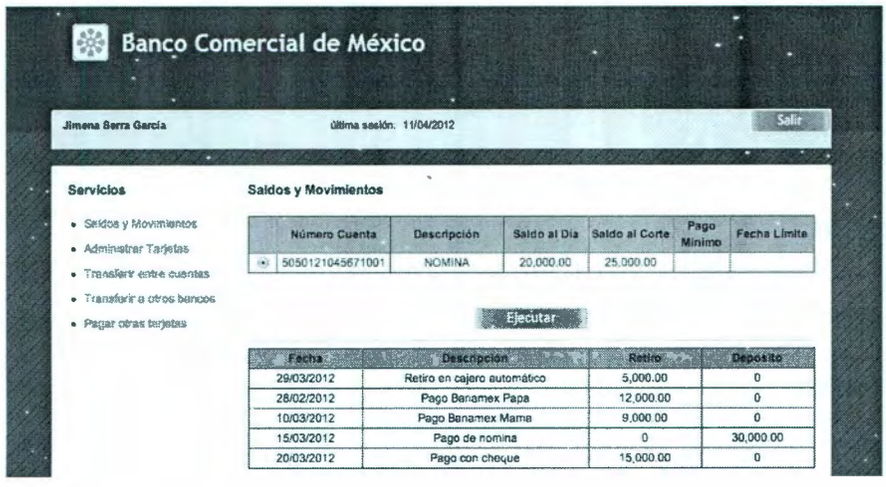
6.2.1.4 *Interfaz del usuario*

Finalmente, con referencia a las interfaces que se mostran a los usuarios, algunos ejemplos de las páginas web diseñadas se muestran a continuación.

- Página principal:

- Página principal para entrar a la banca en línea

- Página de Saldos y Movimientos.



6.3 Implementación de la metodología propuesta

Como bien sabemos, la información hoy en día es uno de los activos más importantes de las organizaciones y como tal requiere de una protección adecuada. Las recientes violaciones a la seguridad de la información y el valor de ésta, enfatizan la creciente necesidad de que las organizaciones protejan su información de manera adecuada y que los sistemas que manejan información crítica, como los sistemas computacionales del Banco Comercial de México, ofrezcan las medidas y controles necesarios para prevenir daños que afecten a la integridad de la información que manejan.

De esta manera, el Banco Comercial de México sabiendo que la integridad de la información de los usuarios es uno de los principales puntos de riesgo para su negocio, ha puesto particular interés en la seguridad de sus sistemas, para ofrecer a sus clientes la plena confianza de que su información está completamente resguardada y que las operaciones que realicen a través de la banca electrónica sean seguras y confidenciales. Entendiendo así, que la administración de la seguridad computacional debe ser parte de la administración del sistema computacional.

Dado los anteriores puntos, el equipo de desarrollo de software decidió utilizar la metodología de desarrollo de aplicaciones web financieras seguras, debido a que incorpora una serie de actividades a desarrollar para integrar la seguridad de la información en todo el ciclo de vida del sistema, y donde además facilita la integración de las seguridad en el diseño y desarrollo a través del uso de patrones de seguridad.

De esta manera, el primer punto consistió en mapear las fases del ciclo de vida normalmente utilizado con aquellas que se describen en la metodología que abarca el presente documento. Obteniendo así el siguiente resultado:

Metodología RUP	Metodología propuesta
Modelado del negocio	Análisis
Requerimientos	
Análisis y Diseño	Diseño
Implementación	Desarrollo
Pruebas	Pruebas
Despliegue	Implementación
	Mantenimiento
	Retiro

Tabla 6.2 – Mapeo de la metodología actual con la metodología propuesta

Posteriormente, aplicando las actividades que propone la metodología, se identificaron los siguientes patrones de seguridad a aplicar durante todo el ciclo de vida de desarrollo del software.

Etapa de SDLC	Actividades propuestas	Patrones aplicados
Análisis	Identificación de los requerimientos funcionales y de seguridad	Identificación de necesidades de seguridad para los activos de la empresa Definición de derechos por rol
	Clasificación de la información y la seguridad	N/A
	Análisis de riesgos	Valoración de activos Valoración de amenazas Valoración de vulnerabilidades Determinación de riesgos
	Selección de controles	Enfoques de seguridad en la empresa Servicios de seguridad de la empresa Registrándose al validar fuera de banda
	Planeación de la seguridad	Compartir la responsabilidad por la seguridad Comunicación con otras compañías Escoger los productos correctos
Diseño	Arquitectura de seguridad	Alternativas de diseño automático de I&A Socios conocidos Acción base segura (J2EE) Pipa segura (J2EE) Sesión segura Token para credenciales (J2EE)
	Diseño de controles de autenticación y autorización	Diseño y uso de contraseñas Bloqueo de cuentas Autenticación forzosa (J2EE) Autorización forzosa (J2EE) Administrador seguro de sesiones (J2EE)
	Diseño de controles de entrada	Validador interceptor (J2EE)
	Diseño de controles de proceso	Fachada de servicios seguros (j2EE) (*Se especifica en Arquitectura de seguridad)
	Diseño de controles de salida	Objeto de transferencia ofuscado (J2EE)
	Diseño de controles de monitoreo y auditoría	Interceptor auditor (J2EE)
Desarrollo	Configuración del ambiente de desarrollo	N/A
	Desarrollo seguro y su control	Implementación de los patrones anteriores
	Revisión del código	N/A
Pruebas	Configuración del ambiente de prueba	Pruebas con un servidor de pruebas
	Inspección de la seguridad en los módulos de la aplicación	Pruebas para atacar al diseño
	Inspección de la seguridad integral de la aplicación	N/A
Producción	Configuración de la seguridad en el ambiente de producción	N/A
	Certificación de la aplicación	N/A
	Distribución e instalación segura de la aplicación	N/A
Mantenimiento	Monitoreo de la seguridad y disponibilidad de las aplicaciones	N/A
	Control cambios	N/A
	Plan de contingencia	N/A

Tabla 6.3 – Implementación de la metodología propuesta

Centrándonos en la etapa de diseño del sistema, que es aquella en dónde se concentra la mayoría de los patrones de seguridad a aplicar, así como también, es una de las etapas que introduce un mayor número de vulnerabilidades en el sistema [51]; a continuación se muestran las vistas arquitectónicas que sufrieron un mayor número de cambios, gracias a la aplicación de los patrones de seguridad.

6.3.1.1 Vista lógica

Con respecto a este punto, la selección de patrones de seguridad fue la siguiente:

Figura 6.4 –Patrones de seguridad aplicados al caso de estudio

Como se podrá observar, la solución consiste principalmente en crear una fachada (*SecureBaseAction*) de servicios de seguridad, la cual intercepte las diferentes peticiones de información hacia la aplicación. Esta fachada entonces, aplicará la lógica de seguridad (autenticación de la petición, autorización, validación de datos y validación de la sesión); si los diferentes controles de seguridad aceptan la petición, entonces ésta será procesada por el negocio, de lo contrario, se mandará un mensaje genérico de error.

Con respecto al negocio, también se utilizará una fachada (*SecureServiceFacade*) la cual controle el flujo de información entre los diferentes procesos del sistema, esta fachada será la que se conecte con el esquema del negocio que se modeló durante la sección anterior (6.2.1.2). En nuestro caso de estudio, esta fachada será el mismo *EBController*, quien ahora además deberá hacer uso del servicio *AuditInterceptor* quien será el encargado de generar logs para cuestiones de auditoría y monitoreo de la aplicación.

De esta manera, esta selección de patrones produjo el siguiente diagrama de clases.

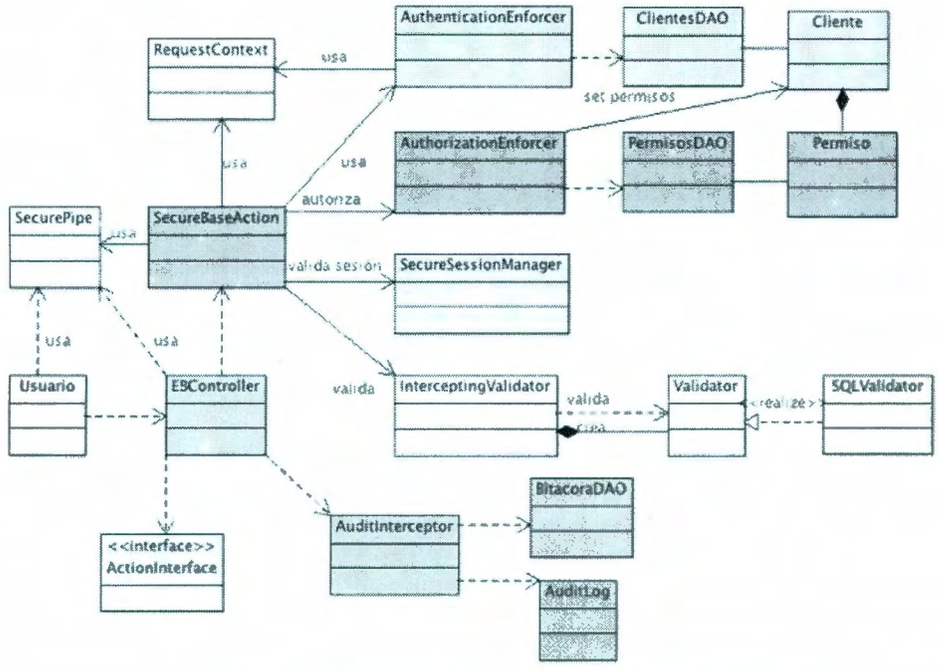


Figura 6.5 –Diagrama de clases de los patrones de seguridad

El esquema generado, no solo nos ayuda a encapsular los servicios de seguridad, si no también, permitirá una mayor flexibilidad si en el futuro la implementación de estos servicios llegara a cambiar. Adicionalmente, un ejemplo del código fuente generado es el siguiente:

```
public class SecureBaseAction {

    public boolean execute(HttpServletRequest request, ArrayList<String> validaciones)

        boolean valido = false;

        //Valida parámetros
        InterceptingValidator validator = new InterceptingValidator();
        if (!validator.valida(request)) {
            throw new ExcepcionSistema("Los parámetros son inválidos");
        }

        //Valida sesión
        String accion = (String) request.getAttribute("accion");
        SecureSessionManager session = new SecureSessionManager();
        AuthorizationEnforcer authorizer = new AuthorizationEnforcer();

        if (accion != null && accion.equals("ingresa")) {
            //Es nueva sesión
            Cliente cliente = null;
            try {
                //Autentica usuario
                AuthenticationEnforcer authenticator = new AuthenticationEnforcer();
                cliente = authenticator.autentica(request);
            } catch (RuntimeException re) {
                throw re;
            } catch (ExcepcionSistema es) {
                validaciones.add(es.toString());
                return false;
            }
            if (cliente == null) {
                throw new RuntimeException("No se recuperó el usuario.");
            }
            //Autoriza usuario
            authorizer.recuperaPermisos(cliente);
            //Genera sesión
            session.generaSession(request, cliente);

            valido = true;

        } else {
            //Ya en sesion
```

Figura 6.6 – Implementación del patrón SecureBaseAction

6.3.1.2 Vista de datos

Finalmente, también fue necesario modificar la estructura de la base de datos. Los cambios más notables es en el manejo de permisos a través de los perfiles, así como la introducción de una tabla de bitácora, la cual registrará los eventos generados por la aplicación y los cuales servirán posteriormente para auditoría.

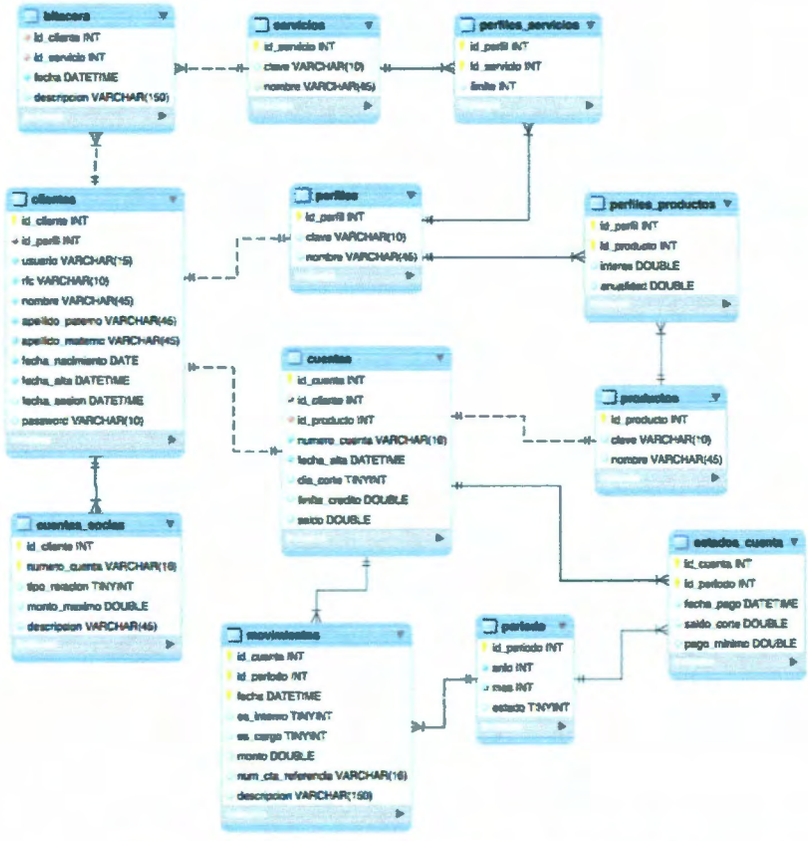


Figura 6.7 –Vista de datos al aplicar los patrones de seguridad

CAPÍTULO 7

7 Análisis e Interpretación de Resultados

En este capítulo se describen las pruebas realizadas al caso de estudio, con el fin de analizar los resultados obtenidos para determinar si la metodología propuesta aporta mejoras en la seguridad de la información para el desarrollo de aplicaciones financieras seguras.

Con este fin en mente, y conforme lo dicta la sección "5.3 Análisis y validación de la metodología" dentro del punto "5.3.1 Evaluación de la seguridad en una aplicación financiera", se desarrollaron dos aplicaciones de tipo financiero. Donde la primera fue desarrollada bajo la metodología de desarrollo de software que utiliza la institución, sin adoptar consideraciones de seguridad que no estuviesen especificadas en los requerimientos, y la otra, incorporando las actividades y patrones de seguridad que dicta la metodología propuesta de desarrollo de aplicaciones financieras seguras.

7.1 Presentación de resultados

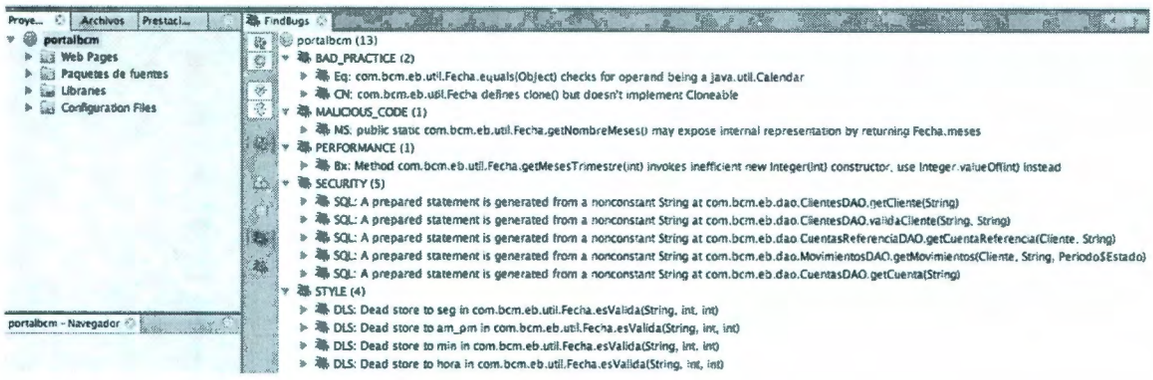
Como se dictó en la sección 5.3.1 "Evaluación de la seguridad en una aplicación financiera", se realizaron dos tipos de pruebas automáticas a las aplicaciones:

a. Prueba de caja blanca. Para ello se hizo uso de la herramienta Fortify, el cuál se emplea para revisar la aplicación a partir de su código fuente.
b. Prueba de caja negra. Donde se hizo uso de la herramienta Skipfish, para aplicar la revisión sobre la aplicación en producción.

7.1.1 Pruebas aplicadas a la aplicación financiera sin la utilización de la metodología

7.1.1.1 Pruebas de caja blanca

Con referencia a la prueba de caja blanca, se utilizó en primera instancia, la aplicación OpenSource FindBugs, la cual arrojó los siguientes resultados. Entre ellos, lo más destacable es la presencia de errores en seguridad que pueden ocasionar Inyección de SQL.



Para complementar la prueba anterior, se utilizó adicionalmente la herramienta Fortify, la cual concluyó el siguiente análisis para la aplicación que fue desarrollada mediante una metodología de desarrollo de software y la cual no contempla actividades de seguridad durante la misma.

Fortify Developer Workbook

FORTIFY

Report Overview

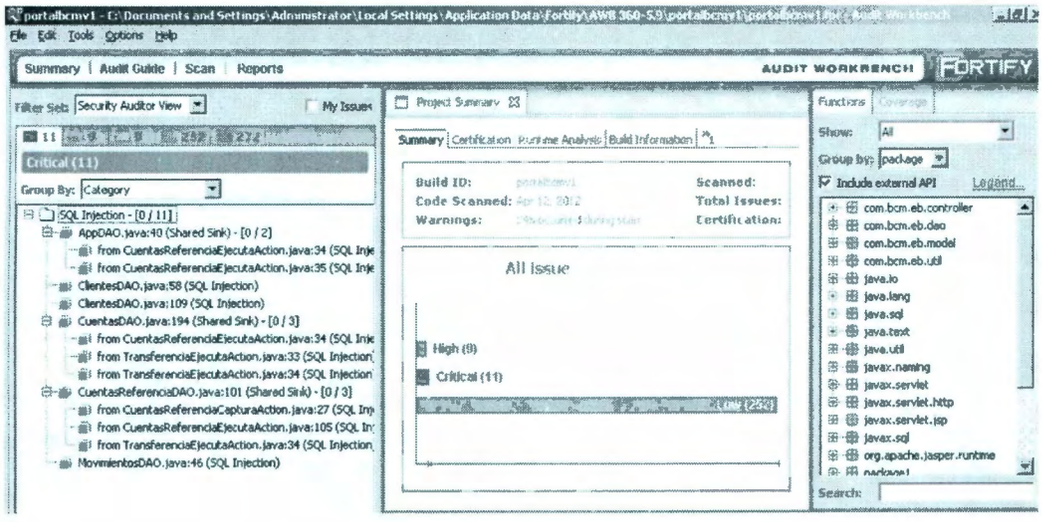
Report Summary

On Apr 12, 2012, a source code review was performed over the portalbcmv1 code base. 82 files, 2234 LOC (Executable) were scanned. A total of 272 issues were uncovered during the analysis. This report provides a comprehensive description of all the types of issues found in this project. Specific examples and source code are provided for each issue type.

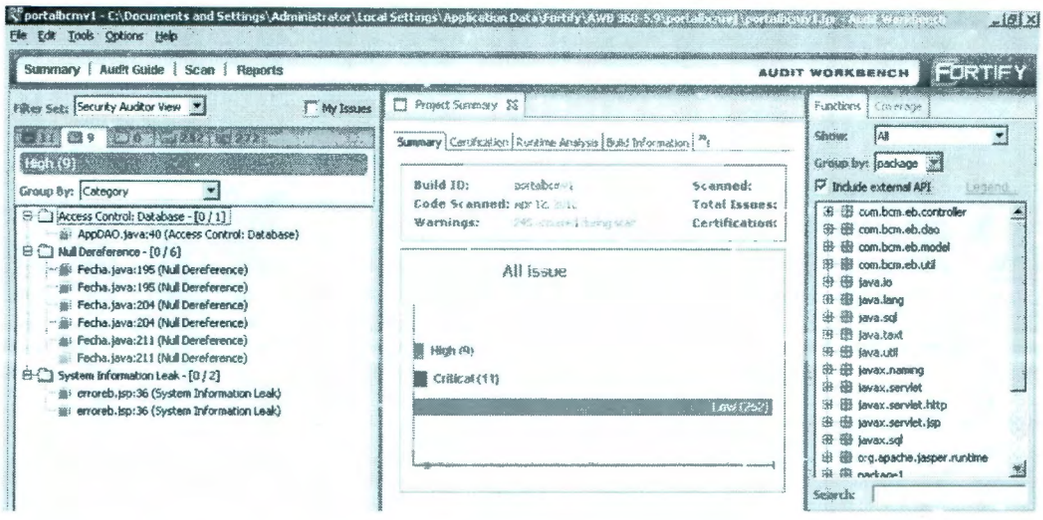
Issues by Fortify Priority Order	
Low	252
Critical	11
High	9

Tabla 7.1 – Resultados de las pruebas de caja blanca sin el uso de la metodología

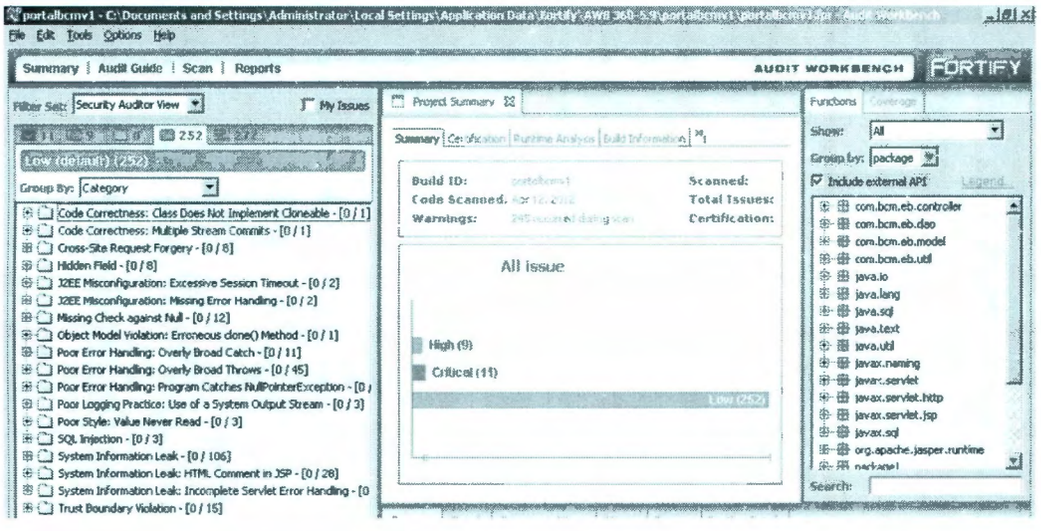
Más detalladamente, las vulnerabilidades críticas que se encontraron correspondieron al riesgo de Inyección de SQL como se puede apreciar a continuación:



En cuanto a las vulnerabilidades de alto nivel, se pudo observar que pertenecen a los eventos de riesgo sobre uso de referencia nulas, control de acceso y fuga de información.



Finalmente, las vulnerabilidades de bajo nivel correspondieron a aquellos riesgos como Secuencia de Comandos en Sitios Cruzados, Fugas de información, Inyección de SQL y Manejo pobre de errores, entre otras.



7.1.1.2 Pruebas de caja negra

Sobre las pruebas de caja negra, se ejecutó un análisis con la herramienta Skipfish, la cual realiza una serie de ataques directamente a la aplicación en producción. Los ataques confirmaron un ataque con Inyección de SQL así como la fuga de información y el compromiso de contraseñas a través de fuerza bruta.

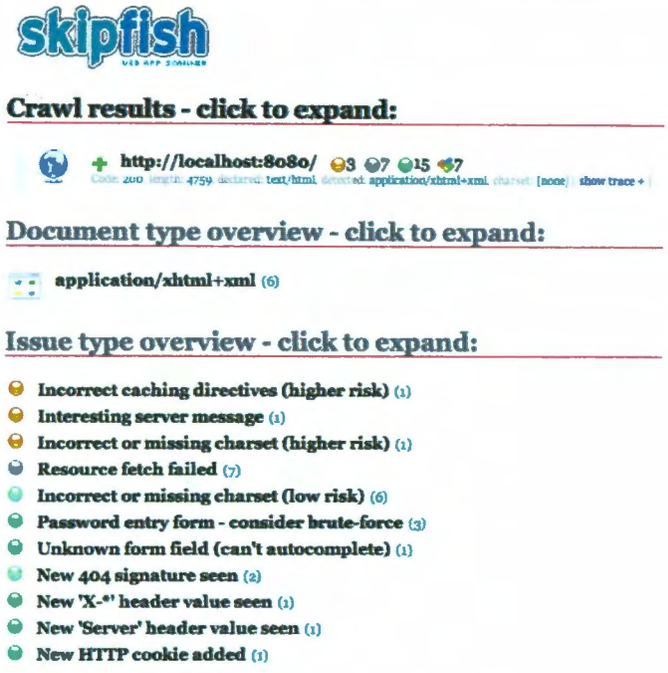
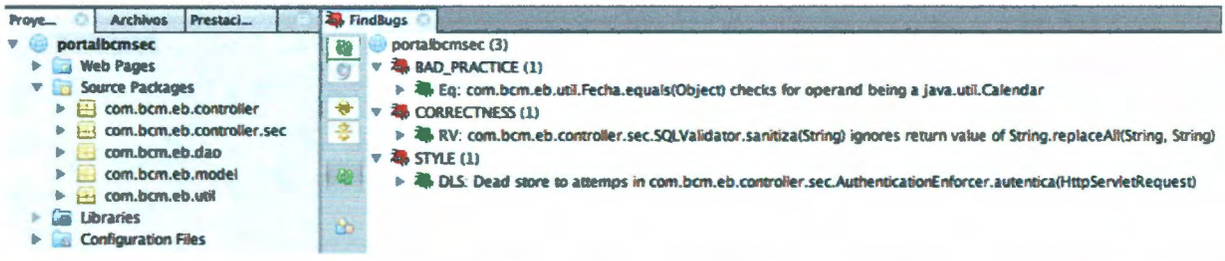


Figura 7.1 –Resultados de las pruebas de caja negra sin el uso de la metodología

7.1.2 Pruebas aplicadas a la aplicación financiera con la utilización de la metodología

7.1.2.1 Pruebas de caja blanca

Al aplicar los patrones de seguridad descritos durante el caso de uso, el resultado del análisis de FindBugs, muestra una reducción significativa en el número de defectos localizados. Así como también, se puede notar la desaparición del error de seguridad con referencia a la Inyección de SQL.



Al igual que en la sección anterior, para complementar el análisis, se sometió la nueva aplicación a las pruebas de caja blanca de la herramienta Fortify, donde el diagnostico generado se presenta a continuación, y en donde como se podrá observar, se eliminaron los errores críticos.

Fortify Developer Workbook

Report Overview

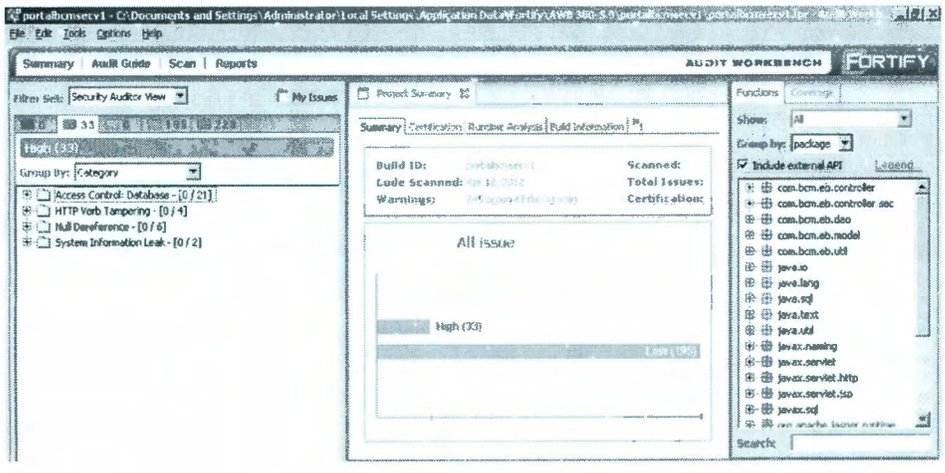
Report Summary

On Apr 12, 2012, a source code review was performed over the portalbcmsecv1 code base. 95 files, 2332 LOC (Executable) were scanned. A total of 228 issues were uncovered during the analysis. This report provides a comprehensive description of all the types of issues found in this project. Specific examples and source code are provided for each issue type.

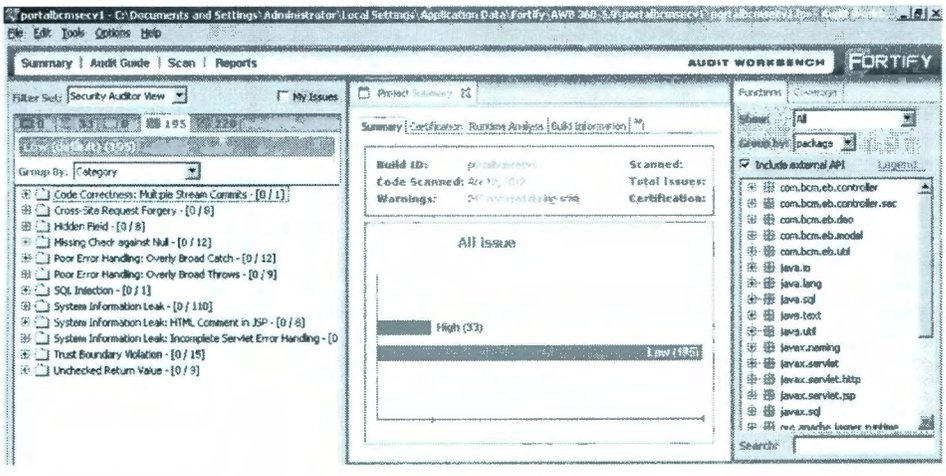
Issues by Fortify Priority Order	
Low	195
High	33

Tabla 7.2 – Resultados de las pruebas de caja blanca con el uso de la metodología

De manera detallada, los errores de alto nivel se presentan a continuación, los cuales normalmente se enfocan al control de bases de datos, referencias nulas y fuga de información.



Finalmente los errores de bajo impacto, se concentraron principalmente en el manejo de errores y de posibles referencias nulas.



7.1.2.2 Pruebas de caja negra

Finalmente, las pruebas de caja negra también demostraron una reducción en el número de ataques con éxito. Cabe mencionar que el único ataque crítico que se presentó, corresponde a la configuración de la página de error del servidor, misma actividad que por el momento se encuentra fuera del alcance de la presente investigación.

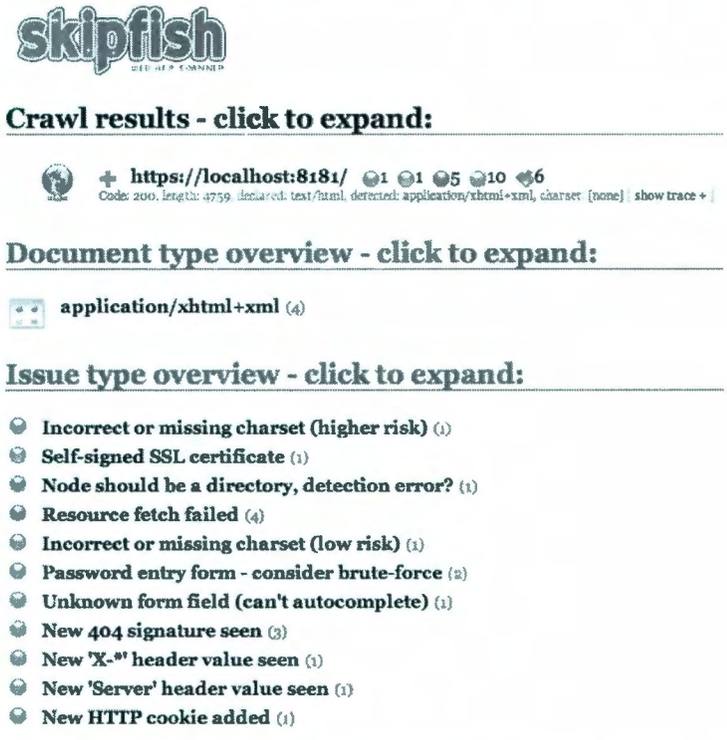


Figura 7.2 –Resultados de las pruebas de caja negra con el uso de la metodología

7.2 Análisis e interpretación de resultados

Finalmente, transcribiendo los resultados obtenidos en las pruebas anteriores, los resultados son los siguientes.

Riesgos	Aplicación A sin aplicar la metodología propuesta		Aplicación A' aplicando la metodología propuesta	
	Prueba de caja blanca	Prueba de caja negra	Prueba de caja blanca	Prueba de caja negra
1. Inyección	14	1	N/A	N/A
2. Secuencia de Comandos en Sitios Cruzados	N/A	N/A	N/A	N/A
3. Perdida de Autenticación y Gestión de Sesiones	2	3	N/A	2
4. Referencia Directa Insegura a Objetos	N/A	N/A	N/A	N/A
5. Falsificación de Peticiones en Sitios Cruzados	8	N/A	8	N/A
6 Configuración Defectuosa de Seguridad	N/A	N/A	N/A	5
7. Almacenamiento Criptográfico Inseguro	N/A	N/A	N/A	N/A
8. Falla de Restricción de Acceso a URL	N/A	N/A	N/A	N/A
9. Falta de protección en la Capa de Transporte	N/A	N/A	N/A	N/A
10. Redirecciones y reenvíos no validados	N/A	N/A	N/A	N/A
11. Control inadecuado de entradas	15	1	N/A	N/A
12. Control inadecuado de los procesos	86	N/A	38	N/A
13. Control inadecuado de salidas	116	4	110	1
Total de riesgos identificados	272	32	228	23
Riesgos aplicables	241	9	156	3
Porcentaje de riesgos			-35.2	-66.6

Tabla 7.3 – Cuadro de resultados

Aplicando la fórmula para caja blanca:

$$-35.2 = \left(\frac{156}{241} * 100\right) - 100$$

Aplicando la fórmula para caja blanca:

$$-66.6 = \left(\frac{3}{9} * 100\right) - 100$$

Dado que tanto las pruebas de caja blanca, como las pruebas de caja negra arrojaron una disminución en el porcentaje de riesgos mayor a 20 puntos porcentuales, se puede concluir que la metodología propuesta permitió reducir el número de vulnerabilidades en la aplicación financiera estudiada.

CAPÍTULO 8

8 Conclusiones y Trabajos Futuros

En este trabajo de investigación, se estudió una propuesta metodológica para disminuir el número de vulnerabilidades en las aplicaciones financieras a través de la aplicación de patrones de seguridad en las diferentes etapas del ciclo de vida del sistema.

El interés de incluir este tipo de prácticas a la metodología se debe, a que los patrones de seguridad ofrecen una forma genérica y probada de resolver un problema recurrente de seguridad, cuya estructura permite a los desarrolladores de sistema su fácil comprensión, adaptación y aplicación.

Como pudimos constatar dentro de la sección 4.3 "Patrones de Seguridad para las Aplicaciones Financieras ", éstos se agrupan principalmente en el área de diseño de la aplicación, lo cual beneficia ampliamente al desarrollo de los sistemas, dado que conforme a la teleconferencia sobre "Construcción de Aplicaciones Seguras" impartida por Gartner [51], en las fases de diseño y construcción de software es donde se concentran los errores más graves en cuanto al número de vulnerabilidades introducidas en las aplicaciones.

Además de estos beneficios, conforme al análisis llevado a cabo dentro de la sección 4.3.2 "Selección de patrones de seguridad", los patrones de seguridad se asocian a la mayoría de los controles que las aplicaciones financieras requieren implementar para cumplir con la normatividad aplicable al sector financiero mexicano.

Dado estas premisas y los resultados mostrados durante los capítulos 6. "Caso de Estudio" y 7. "Análisis e Interpretación de los Resultados" se puedo entonces comprobar, que la aplicación de patrones de seguridad permite mejorar las metodologías de desarrollo de sistemas, así como disminuir el número de vulnerabilidades presentadas por las aplicaciones financieras.

8.1 Contribuciones

Los resultados principales de este estudio se conjugan dentro del capítulo 5. "Metodología para el Desarrollo de Aplicaciones Financieras Seguras". Donde se define una metodología de desarrollo de sistemas en la que se especifican una serie de actividades de seguridad encaminadas a mejorar la metodología de desarrollo de la organización al incorporar controles de seguridad.

Esta metodología nos solo define una serie de actividades, sino que además rompe la barrera entre profesionistas de seguridad y desarrolladores de sistemas, al incorporar el uso de patrones de seguridad en las diferentes fases del ciclo de vida, y en especial durante el diseño de la aplicación. Lo que obliga a pensar en la seguridad durante las primeras etapas de desarrollo del software y no solo hasta que el producto haya sido terminado.

Además de esta metodología, los capítulos 2. "Estado del Arte", 3. "Seguridad en las Aplicaciones Financieras" y 4. "Patrones de Seguridad" proporcionan un análisis de las diferentes fuentes utilizadas para el desarrollo de estos temas.

De esta manera, dentro del "Estado del Arte" se analizaron los artículos de la normatividad aplicable al sector Financiero Mexicano que hablan sobre la seguridad de la información de los sistemas. Lo que permite al lector acercarse y conocer los aspectos legales que rigen al desarrollo de este tipo de aplicaciones. Asimismo, los patrones de seguridad seleccionados para la metodología cubren varios requerimientos de este sector, lo que representa un valor adicional en este tipo particular de desarrollos.

Igualmente, dentro del "Estado del Arte", se hace una comparación entre los estándares de seguridad más comunes, permitiendo contar con una perspectiva de alto nivel sobre el alcance del mismo, para su valoración y futura incorporación como una guía para el desarrollo de sistemas.

Por su parte, el capítulo "Seguridad en las Aplicaciones Financieras" bajo un enfoque de análisis de riesgos determina los ataques, vulnerabilidades y riesgos más comunes que aplican a la generalidad de las aplicaciones financieras, proponiendo controles para la mitigación de los mismos.

Finalmente el capítulo "Patrones de Seguridad" ofrece una clasificación de los patrones comerciales más comunes dentro de las etapas del ciclo de vida de desarrollo de software, facilitando así a los desarrolladores de software incorporar más fácilmente los mismos a sus actividades y, como se mencionó, apegándose a los requerimientos definidos por el sector financiero.

8.2 Conclusiones

Como se pudo comprobar a través del "Caso de Estudio" y el "Análisis e Interpretación de los resultados", el manejo de la metodología proporcionada disminuye cerca de un *35%* el número de vulnerabilidades detectadas por las pruebas de caja blanca y negra, a diferencia de aquellos

sistemas que fueron desarrollados bajo un ciclo de vida de desarrollo de software tradicional en el que no se contemplan actividades específicas de seguridad.

Además, para este caso en particular, se notó que la implementación de patrones de seguridad facilitó al desarrollador la integración de controles de seguridad en la aplicación. Donde, desde un perspectiva arquitectónica, la aplicación de estos patrones mejoran la arquitectura del sistema y la separación de intereses, entre aquellos componentes propios del negocio y aquellos encargados de la seguridad.

8.3 Limitaciones

Como se describe en la sección 1.3.1 "Alcances", la presente metodología no toma en cuenta la creación de un gobierno de Tecnologías de la Información, así como el establecimiento y configuración de una infraestructura de seguridad, a pesar que existen patrones que pueden aplicarse dentro de estas actividades. Esto se debe a que por cuestiones de tiempo, se decidió acotar la misma para abarcar únicamente las actividades exclusivas al desarrollo de la aplicación.

Así mismo, por cuestiones de recursos, tiempo y a falta de un ambiente de trabajo multiempresarial, sólo se pudo llevar a cabo la primera etapa especificada en la sección 5.3.2 "Validación general de la metodología".

8.4 Trabajo futuro

La presente investigación puede dirigirse principalmente a cuatro vertientes de desarrollo:

- Afinar los niveles de riesgos para aplicaciones financieras mediante pruebas de hackeo ético a las aplicaciones producidas con la metodología propuesta.
- La mejora de la metodología propuesta, al aplicarla a varios desarrollos de diferentes propósitos, observando los resultados que arroja al validarla con herramientas de caja blanca y caja negra.
- El uso de aspectos para modelar los patrones de seguridad.
- La creación de una metodología o extensión de la presentada en este trabajo para el desarrollo de aplicaciones móviles con seguridad.

Referencias

[1] ITU, Nielsen Online, «Estadísticas Mundiales del Internet,» 31 marzo 2011. [En línea]. Available: http://www.exitoexportador.com/stats.htm. [Último acceso: 19 febrero 2012].

[2] Deloitte Touche Tohmatsu, «2011 TMT Global Security Study – Key Findings,» 8 diciembre 2011. [En línea]. Available: http://www.deloitte.com/view/en_GX/global/industries/technology-media-telecommunications/c43b53c08adb3310VgnVCM2000001b56f00aRCRD.htm. [Último acceso: 20 febrero 2012].

[3] Kroll, «Informes globales sobre fraude,» 1 febrero 2012. [En línea]. Available: http://es.krollconsulting.com/insights-reports/global-fraud-reports/. [Último acceso: 5 febrero 2012].

[4] Ponemon Institute, «Second Annual Cost of Cyber Crime Study,» agosto 2011. [En línea]. Available: http://www.arcsight.com/webinars/watch/2nd-annual-cost-of-cyber-crime-findings/. [Último acceso: 20 febrero 2012].

[5] IBM X-Force, «IBM X-Force 2011 Mid-year Trend and Risk Report,» septiembre 2011. [En línea]. Available: http://www-935.ibm.com/services/us/iss/xforce/trendreports/. [Último acceso: 20 febrero 2012].

[6] Sophos, «Security Threat Report 2012,» febrero 2012. [En línea]. Available: http://www.sophos.com/en-us/security-news-trends/reports/security-threat-report.aspx. [Último acceso: 19 febrero 2012].

[7] Symantec Corporation, «Symantec Internet Security Threat Report - 2010,» abril 2011. [En línea]. Available: http://www.symantec.com/threatreport/. [Último acceso: 20 febrero 2012].

[8] A. De la Cruz, «Informe de Symantec revela la existencia de ataques maliciosos dirigidos a sitios web confiables,» 8 abril 2008. [En línea]. Available: http://www.symantec.com/es/es/about/news/release/article.jsp?prid=20080408_01. [Último acceso: 20 febrero 2012].

[9] D. Noopur, H. Watts, S. Redwine, G. Zibulski y G. McGraw, «Processes for Producing Secure Software,» *IEEE Security & Privacy,* vol. 2, nº 3, pp. 18-25, mayo-junio 2004.

[10] CSI, «Computer Crime and Security Survey 2010/2011,» 6 junio 2011. [En línea]. Available: http://gocsi.com/survey. [Último acceso: 20 febrero 2012].

[11] D. Kienzle y M. Elder, «Security Patterns Repository Version 1.0,» DARPA, 2002.

[12] J. Meier, «Web Application Security Engineering,» *IEEE Security & Privacy,* vol. 4, nº 4, pp. 16-24, julio - agosto 2006.

[13] Ponemon Institute, «State of Web Application Security,» 26 abril 2010. [En línea]. Available: https://www.imperva.com/ld/ponemon_web_application_security.asp. [Último acceso: 20 febrero 2012].

[14] CERT, «About Cyber Security Engineering,» 12 diciembre 2011. [En línea]. Available: http://www.cert.org/sse/cseoverview.html. [Último acceso: 2 febrero 2012].

[15] K. Suresh Babu y K. Chandrasekharaiah, «Security Patterns: State-of-the-Art Scenario,» *IJCSNS International Journal of Computer Science and Network Security,* vol. XI, nº 4, pp. 131-135, abril 2011.