ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA EN SISTEMAS

IMPLEMENTACIÓN DE SEGURIDAD LÓGICA DE SOFTWARE APLICANDO TÉCNICAS DE PROGRAMACIÓN ORIENTADA A ASPECTOS

Proyecto previo a la obtención del título de ingeniero en Sistemas Informáticos y de Computación

AUTORES:
SALTOS LÓPEZ VALERIA DEL CARMEN

vale.saltos@hotmail.com

VILLACRÉS VEGA DAVID ESTUARDO

david_v_vega@hotmail.com

DIRECTOR: MSC. CARLOS EDUARDO ANCHUNDIA VALENCIA

carlos.anchundia@epn.edu.ec

CODIRECTOR: MSC. JHONATTAN JAVIER BARRIGA ANDRADE

jhonattan.barriga@epn.edu.ec

Quito, noviembre 2017

DERECHOS DE AUTOR

Nosotros, Valeria del Carmen Saltos López y David Estuardo Villacrés Vega, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Firman:

Valeria del Carmen Saltos López	**David Estuardo Villacrés Vega**
C.C. 0201777778	C.C. 1718923434

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Valeria del Carmen Saltos López y David Estuardo Villacrés Vega, bajo mi supervisión.

Msc. Carlos Anchundia

DIRECTOR DE PROYECTO

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Valeria del Carmen Saltos López y David Estuardo Villacrés Vega, bajo mi supervisión.

Msc. Jhonattan Barriga

CO - DIRECTOR DE PROYECTO

AGRADECIMIENTO

Agradezco a Dios por ser la luz que guía mi camino y haberme permitido llegar con salud y vida a este punto, sorteando todos los obstáculos que se me presentaron.

Por haber confiado y creído en mi para poder realizar este proyecto agradezco profundamente a los ingenieros Carlos Anchundía y Jhonattan Barriga.

A mi querida empresa Solutandi, por haberme brindado todas las facilidades para poder desarrollar el presente proyecto.

A mi amado padre Patricio, quien siempre me motivo a no desfallecer y gracias a él, hoy estoy cumpliendo mi más añorado sueño.

A David, mi compañero de vida; le agradezco por toda la ayuda y tantos aportes no solo para el desarrollo de este proyecto sino también para mi vida.

A mis amados hijos Israel y Valentina por todo su amor y paciencia; en este duro y largo caminar han sido el motivo que ha dado impulso a mi vida para poder llegar de su mano a la consecución de esta meta.

A mi querida abuelita Carmelita por haberme apoyado en todo momento, por sus consejos, valores, por la motivación constante que me ha permitido ser una persona de bien y sobre todo por su amor.

A mi hermano Álvaro quien siempre ha estado a mi lado incondicionalmente; dándome ánimo y sobre todo su cariño.

A mi tía Paty por ser la mejor y estar siempre conmigo en las buenas y sobre todo en las malas.

A mis queridos suegros Teddy y Anita quienes en todo este tiempo han sido un gran apoyo para poder culminar esta etapa. Siempre les estaré eternamente agradecida.

A mi cuñada Lis y mi prima Antonella por brindarme su cariño y confianza.

A mis amigas Andrea, Fernanda y Verónica; por ser las mejores e incondicionales amigas, con quienes he compartido momentos inolvidables.

Finalmente, a mis seres queridos que añoraron verme culminar esta etapa; y aunque físicamente no se encuentren aquí; me acompañan desde el cielo.

Valeria.

AGRADECIMIENTO

A Dios por darme la fuerza y la luz para vivir cada día, y por escuchar mis oraciones cuando necesito una guía.

A mis padres por brindarme su apoyo incondicional durante cada día de mi vida tanto en lo personal como en lo profesional.

A mi esposa, gracias por ser mi compañera de vida y mi soporte durante los años que llevamos compartiendo juntos.

A mis hijos por ser mi inspiración, mi razón de vivir y por cada día permitirme disfrutar y aprender con ellos.

A nuestros profesores, director y co-director de este trabajo, ya que gracias a su guía y ayuda me siento mejor preparado para afrontar nuevos retos personales y profesionales.

David

DEDICATORIA

A mi madre Silvia, que desde el cielo con su único e infinito amor ha sido mi fuente de inspiración para poder culminar esta etapa de mi vida.

A mi padre Patricio, por ser pilar fundamental en toda mi vida; sin su apoyo y amor incondicional nada de esto me hubiese sido posible conseguir.

A mi esposo David, mis hijos Israel y Valentina quienes son mis más grandes tesoros por quienes todo esfuerzo y sacrificio vale la pena.

Valeria.

DEDICATORIA

A mis padres, a mi esposa, a mis hijos, a mis hermanos quienes han sido una inspiración en mi vida para llevar a cabo mis propósitos de vida.

A quienes vean en este trabajo una guía que les sirva de ayuda para cumplir los objetivos que se planteen.

David

CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

CAPÍTULO I. INTRODUCCIÓN

1.1. Descripción del problema

La Programación Orientada a Aspectos (POA) representa un prometedor enfoque para mejorar el proceso de desarrollo de software, parece particularmente apropiado cuando los requisitos de aplicación que parecen estar bien separados cruzan la descomposición básica de la aplicación. El dominio de la seguridad del software es un excelente ejemplo de una preocupación en el mundo real que requiere de una solución sofisticada; llegando a convertirse en un desafío la separación de este tipo de preocupaciones. POA es un enfoque que ofrece avanzadas técnicas de modularización. La principal característica de esta tecnología es la capacidad de especificar tanto el comportamiento de una determinada preocupación, así como, su relación con otras preocupaciones [1]. De hecho, POA se ha convertido en un término general que denota varios enfoques para proporcionar soluciones a las preocupaciones y sus relaciones, llegando a ser tomada en cuenta como una herramienta destacada.

La Programación Orientada a Objetos (POO) a través de las clases, modela y diseña objetos del mundo real, yendo desde el más genérico hasta el más específico. Dicho objeto posee atributos que lo definen y acciones que puede ejecutar para llegar a la solución del problema planteado, apalancándose en una de sus características principales denominada abstracción [2].

Precisamente la abstracción como característica esencial de la POO, hace que ésta se enfoque principalmente en la solución del problema más que en su implementación, provocando que se descuiden otros procesos importantes que deben ser tomados en cuenta, tales como; manejo de errores, auditoría, registro, seguridades, entre otros [2]. Por lo tanto, el no separar procesos técnicos que están directamente relacionados con el objetivo principal de un sistema, de otros procesos que generalmente debe tener el mismo, como lo es la seguridad lógica; provoca que su mantenimiento sea complicado, ya que se tienen clases con muchas líneas de código, como consecuencia de su reutilización [3].

Un mantenimiento complicado imposibilita que el software pueda ser modificado de forma rápida y fiable, lo que deriva en el deterioro de la calidad y rendimiento del sistema [3]. Las consecuencias no deseadas de la utilización de POO se pueden atacar con la utilización de técnicas alternativas de programación, las cuales permiten abstraer o separar los procesos no relacionados de la lógica u objetivo principal de un sistema; estás técnicas están inmersas y definidas dentro de POA.

POA define un aspecto, preocupación o incumbencia como una unidad modular que cruza transversalmente la estructura de otras unidades. En base a dicha definición queda claro que con POA se pueden desarrollar componentes que se integren transversalmente en el sistema y resuelvan incumbencias que no están directamente relacionadas con el giro del negocio.

1.2. Justificación del proyecto

1.2.1. Justificación Teórica

El acceso y autorización en un sistema desarrollado con POO, por lo general se encuentra implementado en varios componentes del mismo; puesto que una de las principales limitaciones de la POO es que no puede abstraer la funcionalidad que se repite en distintos módulos y que sus clases no se encuentren relacionadas; por lo tanto, se puede decir que POO carece de mecanismos para abstraer un comportamiento transversal [2].

La teoría de POA cuyas siglas en inglés son AOP; indica que es posible encapsular distintos conceptos que conforman un sistema, por ejemplo, el acceso y autorización del mismo en un solo componente y que se lo implementa de manera modular y separada del resto del sistema, con lo cual se atacaría la limitación de POO ya mencionado [1][2].

Con POO se podrían implementar los métodos propios del giro del negocio y con POA se pueden separar procesos que gestionan la seguridad de un sistema, los cuales no están directamente relacionados con el objetivo principal del mismo.

1.2.2. Justificación Metodológica

Para realizar el proyecto integrador propuesto se utilizará el Modelo de Desarrollo (MD); el cual es considerado un método de investigación eficaz porque permite describir, predecir, probar o entender sistemas o eventos complejos [4].

Adicionalmente los pasos que plantea el MD se ajustan al proceso de investigación que se llevará a cabo para ejecutar el presente proyecto integrador; los cuales se describen a continuación:

a. Revisión de Literatura

Con el objetivo de sustentar teóricamente lo propuesto en el presente proyecto integrador, se consultó bibliografía disponible referente a POA y Seguridad Lógica de Software; lo que permitirá determinar qué tan utilizado es POA en la actualidad y su aplicabilidad en el campo de la seguridad.

b. Cuestionario para encuesta

Como técnica de investigación para determinar el sistema sobre el cual se implementarán los componentes de seguridad lógica, se seguirá todo el proceso de diseño y elaboración de encuestas. Las preguntas de la encuesta serán planteadas en base a lo encontrado en la revisión de la literatura.

c. Recolección y análisis de datos

Una vez aplicada la encuesta se procederá a tabular los resultados para posteriormente analizarlos y como resultado obtener el sistema sobre el cual se implementarán los componentes de seguridad lógica desarrollados con POA.

d. Desarrollo de la propuesta

En base al sistema seleccionado, sus puntos sensibles, y a los resultados de la literatura se procederá a diseñar y construir los componentes de seguridad.

e. Validación del modelo

Se implementarán los componentes de seguridad en un ambiente de pruebas para validar su funcionamiento, emulando el proceso de lectura de datos aplicando pruebas de rendimiento.

f. Evaluación de los resultados

Se compararán los tiempos de respuesta con y sin aspectos, ejecutando procesos del sistema en base a: comunicación interna, comunicación a sistemas expertos y tipo de mensaje; en el que se usarán datos reales en el ambiente de pruebas.

1.2.3. Justificación Práctica

La empresa CORPORACIÓN TECNOLOGÍA DE LA INFORMACIÓN SOLUTANDI CIA. LTDA (TANDICORP) es la beneficiaria de la implementación del sistema.

TANDICORP es una empresa con más de 15 años de experiencia en el mercado informático. Es una organización comprometida con el desarrollo de propuestas tecnológicas innovadoras y soluciones informáticas cuyo objetivo es potencializar el servicio que ofrece su Compañía, posicionándolo como el más eficiente del mercado.

Entre las soluciones que TANDICORP ofrece a sus clientes, se detallan las siguientes:

- **TANDI – BI**: Esta solución está enfocada en brindar el servicio de Bussiness Intelligence.

- **TANDI – INVOICE:** Sistema de Facturación Electrónica, que se integra fácilmente con el SRI y los clientes que consumen este servicio.
- **TANDI – Open ERP:** ERP de código abierto que se adapta a cualquier giro del negocio.
- **TANDI – PAYROLL:** Sistema de gestión de nómina, de fácil parametrización y sencillo de usar.
- **TANDI – ESB:** Es un sistema que permite integrar aplicaciones de software de varios negocios a través de ESB.
- **3CX Comunicaciones Unificadas:** Es un sistema IP PBX de última generación, de estándar abierto SIP basado en Windows que reemplaza a PBX tradicionales.

Se espera que la investigación que se va a realizar para implementar seguridades usando POA y la posterior aplicación en un caso de estudio, sirva de guía para que se lo pueda aplicar en otros sistemas.

1.3. Objetivo General y Específico

1.3.1. Objetivo General

Analizar, diseñar, construir y probar mecanismos de seguridad lógica, aplicando el paradigma de orientación a aspectos, sobre un caso de estudio factible.

1.3.2. Objetivos Específicos

- Definir el porcentaje de componentes de seguridad lógica que se puede implementar con POA en un sistema a través del análisis del estado del arte.
- Demostrar la implementación de componentes de seguridad con POA en un sistema ya construido, mediante un análisis de funcionalidad antes y después de POA.
- Comparar el rendimiento del sistema con y sin la aplicación del componente de seguridad

CAPÍTULO II. MARCO TEÓRICO

2.1. Componentes Transversales

Los componentes transversales son aquellos que se cruzan en otros módulos de un sistema, también son comportamientos que se extienden sobre múltiples módulos de implementación, que a menudo no se relacionan y que no pueden estar claramente separados unos de otros, una idea de esto se visualiza en la Figura 1.

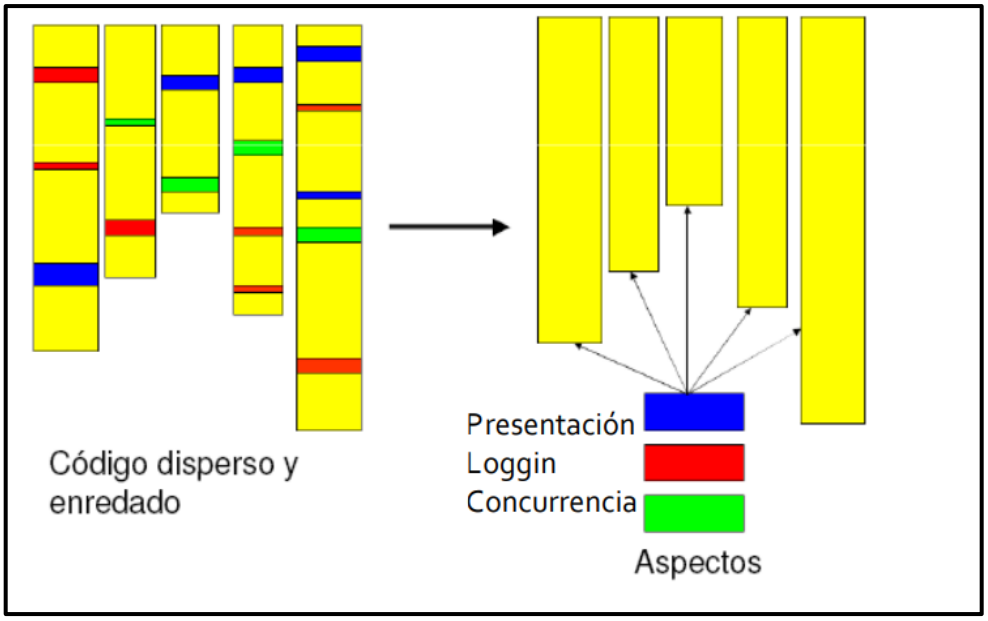


Figura 1 *Separación de componentes Transversales [5]*

Ejemplos de componentes transversales son:

- Seguridad (autorización y auditoría)
- Registro (logging) y depuración
- Sincronización
- Persistencia

Cuando los módulos de un sistema pueden interactuar simultáneamente con varios procesos/requerimientos, lo que implica que los componentes están estrechamente interrelacionados, se produce el *code tangling* (enredo de código). Debido a que los componentes transversales se extienden sobre varios módulos, sus implementaciones también se extienden sobre dichos módulos. Los componentes están mal localizados, y esto se llama *code scattering* (dispersión de código) [6].

2.2. Definición de Aspectos

Un aspecto se considera a cada una de las unidades modulares que cruzan la estructura de otras unidades, además es similar a una clase, que, por tener un tipo, puede heredar clases y otros aspectos, puede ser abstracta o concreta y tener campos, métodos, y tipos como miembros, además encapsula comportamientos que afectan múltiples clases dentro de módulos reutilizables [6].

"POA se esfuerza por ayudar al desarrollador a separar incumbencias para superar los problemas transversales que originan, y proporciona mecanismos de lenguaje que capturan explícitamente la estructura transversal" [6].

Esto demuestra que se hace posible programar componentes transversales de una manera modular y obtener los beneficios usuales de una modularización mejorada: código más sencillo que es más fácil de desarrollar y mantener, y que tiene un mayor potencial de reutilización; esto se logra por

la mejora de la modularización de código usando aspectos. Lo que POO ha hecho para la encapsulación de objetos y la herencia, POA hace para las incumbencias transversales [6].

POA permite tratar la funcionalidad propia del sistema y los aspectos de forma separada. Luego ambos se combinan con un tipo de programa llamado *"Tejedor"* para dar por resultado el sistema final, como se muestra en la Figura 2.

2.3. Principios de POA

POA tiene como único principio el Principio de Separación de Incumbencias.

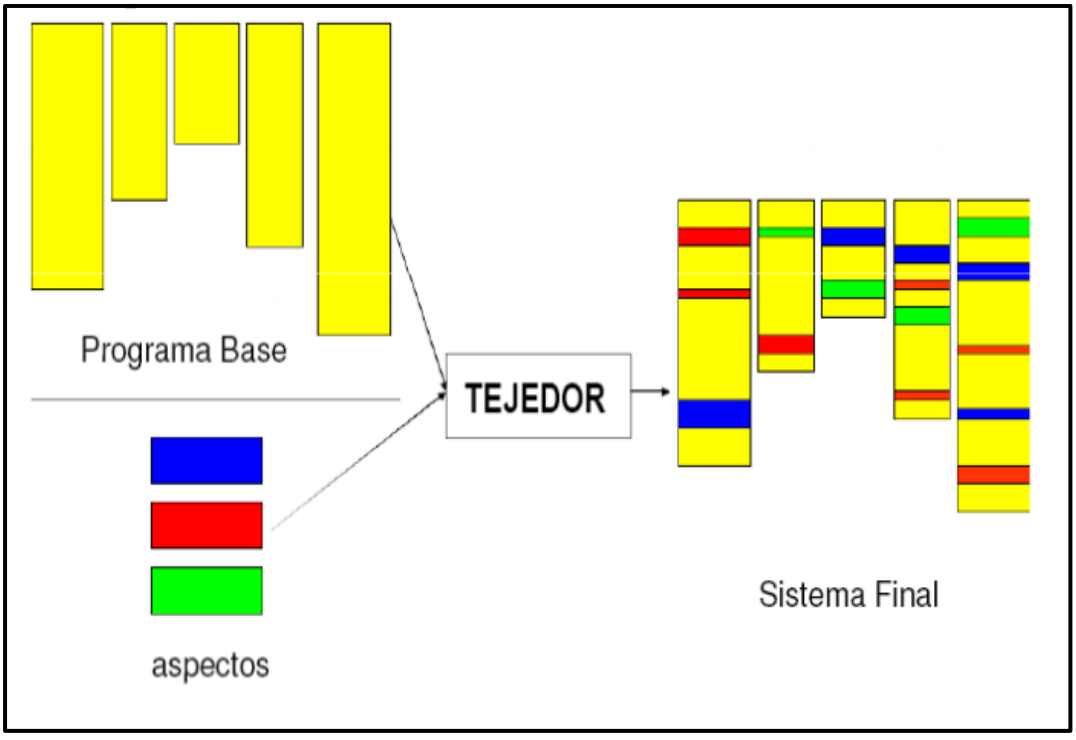


Figura 2 *Tejedor de Aspectos [5]*

2.3.1. Principio de Separación de Incumbencias

La Separación de incumbencias es un importante principio de la ingeniería de Software la cual refiere a la habilidad de identificar, encapsular, y manipular aquellas partes de software que son relevantes a un propósito (concepto, meta, componente, etc.) en particular.

Los componentes pueden ir desde un alto nivel como la seguridad y la calidad de servicios hasta un bajo nivel como el almacenamiento en buffer, almacenamiento en caché y el registro. Pueden ser también componentes funcionales, como lógica de negocio o no funcionales, como sincronización [6].

Este principio plantea también que dentro de un problema se pueden tener varias incumbencias que deben ser identificadas y separadas; separando las incumbencias, se disminuye la complejidad a la hora de tratarlas y se puede cumplir con requerimientos relacionados con la calidad como adaptabilidad, mantenibilidad, extensibilidad y reusabilidad [7].

En un informe detallado por ECURED (2017) sobre las herramientas de programación se menciona:

> *Las técnicas de modelado que se usan en la etapa de diseño de un sistema se basan en partirlo en varios subsistemas que resuelvan parte del problema o correspondan a una parte del dominio sobre el que trata.*
>
> *Estas técnicas sufren en su mayoría la llamada "tiranía de la descomposición dominante" que consiste en guiarse al modelar, implícita*

o explícitamente, por una visión jerárquica determinada de la organización del sistema.

La desventaja de estas particiones es que muchas de las incumbencias a tener en cuenta para cumplir con los requerimientos (en particular, habitualmente, las incumbencias no funcionales) no suelen adaptarse bien a esa descomposición" [8].

El separar incumbencias aplicando POA dentro de un proyecto, permite dividir al mismo en varios aspectos para poderlos desarrollar de manera independiente lo cual facilita el desarrollo ya que se elimina complejidad [8].

2.4. Lineamientos Técnicos de POA

Estructuralmente, un aspecto es una clase con ciertos elementos adicionales para poder abstraer intereses transversales:

- **Join point:** Son los puntos específicos en la ejecución de los componentes base donde es posible el entretejido de aspectos, dependiendo de la herramienta de POA que se utilice y su potencia, se podrá cuantificar los casos que se representarán en un aspecto.

  Algunos ejemplos de join points son: una llamada a un método, la creación de una instancia, el manejo de una excepción, la ejecución de un ciclo, el retorno de un método, la asignación de un valor a una variable, la modificación de un atributo, entre otros [9].

- **Poincut:** Es la agrupación de un grupo de join points dentro de un aspecto y son definidos por el programador, por ejemplo, todas las ejecuciones del constructor de una determinada clase. Todas las herramientas de POA tienen varios designadores de pointcuts los mismos que combinados con expresiones regulares permiten definirlos [9].
- **Advice**: Es un fragmento de código equivalente al contenido de un método, en donde se especifica el comportamiento transversal que contendrá el aspecto. Este comportamiento va acompañado de la indicación del pointcut donde debe ser insertado y la forma de insertarlo. Pueden ser de tres tipos:

  a. Before: Se ejecuta previo a la ejecución del join point asociado

  b. After: Se ejecuta después de la ejecución del join point asociado.

  c. Around: Se ejecuta antes y después del join point asociado,

  pudiendo incluso reemplazar la ejecución del join point [9].

- **Declaraciones de miembros inter-clase (introducción):** Son declaraciones que permiten agregar métodos o atributos a interfaces o clases ya existentes [9].

  Destacándose que las extensiones realizadas por un inter-clase no son visibles para los componentes base, ni tampoco para el propio componente que ha sido extendido.

- **Otras declaraciones:** Son declaraciones de distintos tipos cuya disponibilidad dependen del lenguaje orientado a aspectos que se utilice. Por ejemplo, para el caso de AspectJ se cuenta con los siguientes tipos:

  Parents: Especifica que una clase determinada herede de otra en particular.

  Implements: Agrega la implementación de una interface a una clase dada.

  Soft: Elimina la obligación de capturar las excepciones.

  Warning: Genera un aviso en tiempo de compilación cuando se cumple la condición requerida.

  Error: Genera un error en tiempo de compilación cuando se cumple la condición requerida.

  Precedence: Especifica el orden de aplicación de los distintos aspectos para eliminar conflictos de precedencia de aspectos [9].

2.5. Seguridad Lógica de Software

La seguridad lógica de software es el requisito indispensable, que todo sistema de almacenamiento y procesamiento de datos debe tener, para poder garantizar la disponibilidad, confidencialidad e integridad de los mismos [10].

Asegurando que los recursos del sistema sean utilizados de la manera en la que se espera y los que tienen acceso a la información sean personas o sistemas autorizados para hacerlo [10].

Dentro de la seguridad informática se diseñan los procedimientos, normas, métodos y técnicas destinados a conseguir un sistema seguro y confiable [11]. A continuación, se describe el modelo básico de seguridad con el objetivo de tener una visión general de todos los actores que se involucran dentro del campo de la seguridad en un sistema.

2.5.1. Modelo Básico de Seguridad

La Figura 3 ilustra el modelo básico de seguridad que se deriva de los problemas de seguridad y de sus mecanismos de provisión. En la parte superior del diagrama aparece un proveedor de recursos que ofrece ciertos datos vitales para la seguridad o recursos de aplicación, tales recursos son vitales para la seguridad, ya que alguien con malas intenciones, como el pirata o hacker que se ilustra en la parte inferior del diagrama, podría corromper, hacer referencia (es decir, acceder), sustituir, retrasar el acceso o denegarlo a tales recursos [12].

Un proveedor de servicios de seguridad, que se ilustra en la parte izquierda del diagrama, trata de proteger a tales recursos del ataque, proporcionando servicios de protección, como la integridad, la confidencialidad, la autorización y el acceso, la identidad, la autenticidad, la disponibilidad, la no repudiación y

los servicios de protección de la auditoría. Tales servicios tratan de frustrar los mejores esfuerzos del pirata [12].

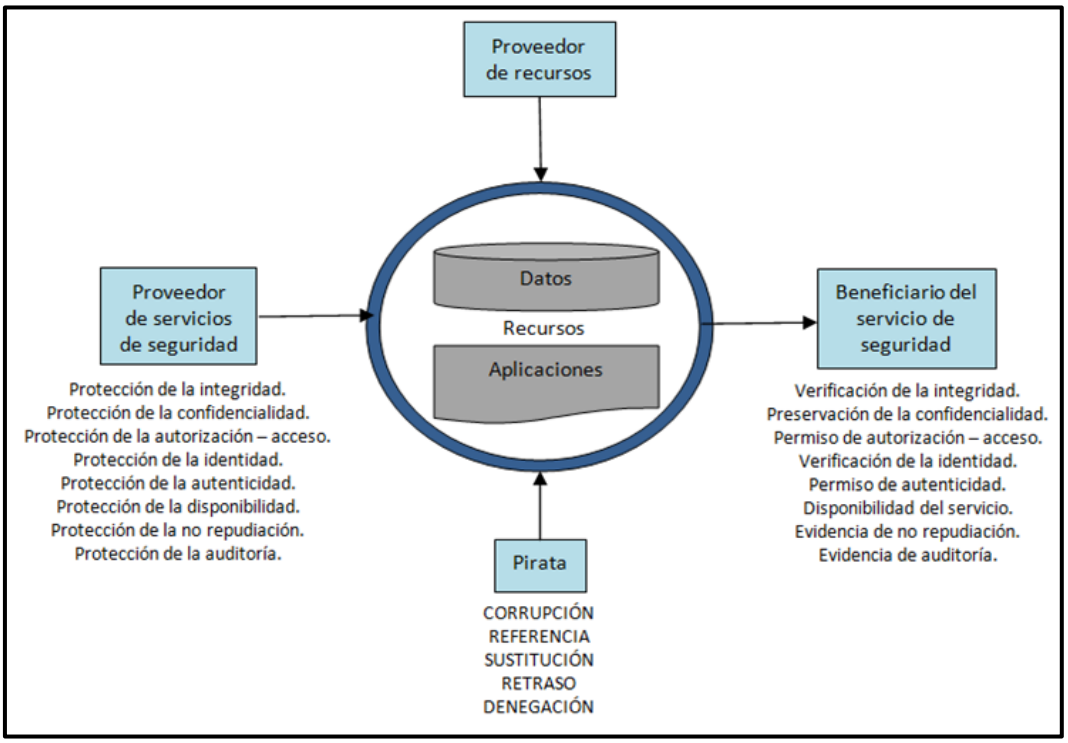


Figura 3 *Modelo básico de Seguridad [11]*

La parte derecha del diagrama muestra el beneficiario de los servicios de seguridad. Estos servicios son los siguientes:

- **Verificación de la integridad:** El beneficiario recibe verificación de que los datos o la aplicación son exactos.
- **Preservación de la confidencialidad:** El beneficiario posee alguna seguridad de que se ha preservado la confidencialidad de los datos o de la aplicación.

- **Permiso de autorización – acceso:** Se ha proporcionado un permiso autorizado para acceder a los datos vitales para la seguridad, a la aplicación o a otro recurso.
- **Verificación de la identidad:** El beneficiario recibe verificación de que la identidad asociada al origen de los datos o de la aplicación es apropiada.
- **Permiso de autenticidad:** Se ha proporcionado un permiso de autenticidad apropiado para acceder a los datos, la aplicación u otro recurso.
- **Disponibilidad del servicio:** El beneficiario tiene acceso a los datos, a la aplicación o a otro recurso cuando espera tenerlo.
- **Evidencia de no repudiación:** El beneficiario tiene cierta evidencia de que un receptor ha recibido datos, en base a la procedencia de los datos o de la aplicación.
- **Evidencia de auditoría:** El beneficiario tiene cierta evidencia de las operaciones vitales que se producen en el sistema para la seguridad.

2.5.2. Principios de la Seguridad Lógica

La mayoría de los daños que puede sufrir un sistema informático no será solo los medios físicos sino contra información almacenada y procesada. El activo más importante que se posee es la información y por tanto deben existir técnicas más allá de la seguridad física que la asegure, estas técnicas las brinda la seguridad lógica [13].

Demostrando que la seguridad lógica consiste en la aplicación de barreras y procedimientos que resguarden el acceso a los datos y solo se permita acceder a usuarios autorizados, teniendo en cuenta los siguientes objetivos:

- Restringir el acceso al arranque (desde la BIOS), al S.O., a los programas y archivos.
- Asegurar que los usuarios puedan trabajar sin una supervisión minuciosa y no puedan modificar los programas ni los archivos que no correspondan.
- Asegurar que se estén utilizando los datos, archivos y programas correctos en y por el procedimiento correcto analizando periódicamente los mismos [13].

2.5.3. Técnicas de Control de Acceso

Estos controles pueden implementarse en la BIOS, el S.O, sobre los sistemas de aplicación, en la base de datos, en un paquete específico de seguridad o en cualquier otra aplicación [13]. Constituyen una importante ayuda para proteger al S.O de la red, al sistema de aplicación y demás software de la utilización o modificaciones no autorizadas. Para mantener la integridad de la información (restringiendo la cantidad de usuarios y procesos con acceso permitido) y para resguardar la información confidencial de accesos no autorizados. De la misma manera es conveniente tener en cuenta otras consideraciones referidas a la seguridad lógica como por ejemplo las

relacionadas al procedimiento que se lleva a cabo para determinar si corresponde un permiso de acceso [13].

a. Identificación y Autorización

Se denomina identificación al momento en que el usuario se da a conocer en el sistema y autenticación a la verificación que se realiza en el sistema sobre esta identificación [13].

La seguridad informática se basa en gran medida en la efectiva administración de los permisos de acceso a los recursos informáticos basados en la identificación, autenticación y autorización de accesos [13].

b. Roles

El acceso a la información también puede controlarse a través de la función perfil o rol del usuario que requiere dicho acceso. Algunos ejemplos de roles serían programador, líder del proyecto, administrador del sistema etc [13].

En este caso los derechos de acceso y política de seguridad asociada pueden agruparse de acuerdo con el rol de los usuarios [13].

c. Limitaciones a los Servicios

Estos controles se refieren a las restricciones que dependen de parámetros propios de la utilización de la aplicación o preestablecidos por el administrador del sistema. Un ejemplo podría ser licencias para la utilización simultánea de un determinado producto software para 5 personas de manera que desde el

sistema no se permita la utilización del producto simultáneamente a un sexto usuario [13].

d. Modalidad de Acceso

Se refiere al modo de acceso que se permite al usuario sobre los recursos y la información esta modalidad puede ser:

- **Lectura**: El usuario puede únicamente leer o visualizar la información, pero no puede alterarla, debe considerarse que la información puede ser copiada o impresa.
- **Escritura**: Este tipo de acceso permite agregar datos, modificar o borrar información.
- **Ejecución**: Otorga al usuario el privilegio de ejecutar programas.
- **Borrado**: Permite al usuario eliminar recursos del sistema como programas, campos de datos o archivos.
- Todas las anteriores.

Además, existen otras modalidades de acceso especiales:

- **Creación**: Permite al usuario crear archivos nuevos, registros o campos.
- **Búsqueda**: Permite listar los archivos de un directorio determinado.

2.6. Análisis de Factibilidad Tecnológica de implementación de seguridades con POA

La Factibilidad Técnica o Tecnológica se enfoca en verificar si se dispone de los recursos técnicos, tanto hardware como software y el personal indicado, para concretar un proyecto tecnológico en una empresa u organización y de ser necesario también se evalúan los requerimientos tecnológicos que deban ser adquiridos para el desarrollo y puesta en marcha del proyecto que pretende realizarse, el cual puede tratarse de algo totalmente nuevo o de la mejora de una solución existente [14].

Entonces, un Análisis de Factibilidad Tecnológica pretende determinar si con la tecnología y la experiencia de un equipo humano disponibles se puede resolver la siguiente interrogante:

¿Se puede implementar el proyecto tecnológico o sistema propuesto?

Para responder la interrogante planteada, un Análisis de Factibilidad Tecnológica debe responder las siguientes preguntas y en base a las respuestas concluir si se puede o no implementar el proyecto propuesto [15]:

1. ¿Actualmente se posee la tecnología necesaria?
2. ¿Se posee la experiencia técnica necesaria?
3. ¿La solución resuelve el problema planteado?

2.7. Estado del Arte de POA con respecto a Seguridad

POA desde sus comienzos fue concebida para modularizar incumbencias transversales (***cross cutting concerns CCC***), no hay un acuerdo sobre si es válido su uso para cualquier CCC. Hay autores que sólo consideran válido su uso para CCC no funcionales, al tiempo que otros proponen su uso para todo CCC. Tampoco faltan los que proponen el uso de POA para cuestiones más allá de la modularización de CCC [9]. Actualmente se cuentan con trabajos en los cuales se aplica POA para modularizar incumbencias transversales porque sus funcionalidades originalmente se encuentran diseminadas en varios módulos y pueden ser agrupadas en una sola unidad, llamada aspecto.

La seguridad evita usos no autorizados y permite el acceso al sistema a quienes tienen el permiso para hacerlo; generalmente todos estos controles se encuentran programados de manera dispersa en todo el sistema y que al ejecutar cada una de estas acciones generan registros en logs o en tablas de auditoría [9]. Es decir que, la **autenticación, autorización y auditoría** son los controles de seguridad más comunes que se consideran al momento de construir un sistema; por lo tanto, al ser controles que no tienen relación con la funcionalidad propia del sistema se convierten en componentes transversales.

Adicionalmente con respecto a la seguridad toda la bibliografía consultada coincide en que la seguridad es un componente transversal en cualquier sistema, ya que se encuentra dispersa pero no es parte funcional del giro del

negocio. Basados en esta premisa se encontraron varios trabajos de investigación como el de Nicolás Martín Paez en su tesis "*Utilización de programación orientada a aspectos en aplicaciones enterprise"* en la cual se demuestra que es posible aplicar POA para implementar seguridades [9].

CAPÍTULO III. DISEÑO

En este capítulo se determinarán los componentes de seguridad que se pueden implementar con POA y su aplicación en un sistema ya construido. Para esto se proponen los siguientes pasos. En base a la teoría se definirán las propiedades de seguridad que se pueden implementar con POA y posteriormente se seleccionará una metodología de desarrollo para el desarrollo de la solución. Con todo lo anterior definido se diseñará la solución y luego se determinará el caso de estudio sobre el cual se validará la misma. Sobre el caso de estudio seleccionado se implementará la solución con la finalidad de evaluar el rendimiento del caso de estudio antes y después de aplicar la solución.

3.1. Definición de Propiedades de Seguridad

Para poder determinar los componentes que pueden ser implementados por aspectos es necesario determinar las propiedades de seguridad. En base al modelo básico de seguridad (Figura 3) se han consolidado las propiedades de seguridad lógica de software como: Integridad, Confidencialidad, Autorización – Acceso, Identidad, Autenticidad, Disponibilidad, No repudio y Auditoría. Sin embargo, tomando como referencia la sección 2.7 del Capítulo II, la autorización, autenticación y auditoría son incumbencias transversales que pueden ser agrupadas y definidas por aspectos, las demás propiedades no pueden ser implementadas en la lógica de un sistema. Por lo tanto, las propiedades de seguridad que pueden ser abstraídos por POA son:

autorización, autenticación y auditoría, lo que equivale a una cobertura del 37.5% de seguridad lógica de un sistema.

Para poder recabar el estado actual de un sistema en cuanto a su cobertura de seguridad se plantea la elaboración de una encuesta. Las preguntas se distribuirán en autenticación, autorización y auditoría, tomando en cuenta el tipo de control, forma de registro y condiciones de negocio. Como resultado se obtuvieron 17, 3 y 12 preguntas de autenticación, autorización y auditoría respectivamente, mismas que fueron depuradas con un experto en seguridad. Estas 32 preguntas fueron evaluadas para poder establecer los procedimientos funcionales que necesitan existir para poder ser implementadas como un aspecto de seguridad, dichos procedimientos se encuentran especificados en la columna de la Tabla 1 *Parámetros POA a cumplirse*. Como resultado se tiene que, las preguntas 11 y 13 no pueden ser implementadas con POA ya que corresponden a actividades no codificables como se detalla en la Tabla 1. Estas preguntas forman parte de la sección I de una encuesta que se utilizará para poder seleccionar el sistema como caso de estudio, misma que se encuentra desarrollada en el Anexo 1.

Tabla 1 Planteamiento de preguntas para Encuesta

Propiedad de Seguridad	Id	Pregunta	Parámetros POA a cumplirse
Auditoría	**1**	¿Son auditados los intentos fallidos de acceso al sistema?	• Existe un módulo de administración de usuarios en el sistema. • Existe un método que se ejecuta para verificar si las credenciales de usuario ingresadas son correctas.
Auditoría	**2**	¿Se auditan los accesos exitosos al sistema?	• Existe un módulo de administración de usuarios en el sistema. • Existe un método que se ejecuta para verificar si las credenciales de usuario ingresadas son correctas.
Auditoría	**3**	¿Se auditan todos los intentos de comunicación con otros sistemas?	• El sistema expone Web Services. • Existe un método que se ejecuta cada vez que los Web Services son consumidos.
Auditoría	**4**	¿Se audita la revocación de perfiles de acceso?	• Existe un módulo de administración de usuarios en el sistema. • Es posible revocar perfiles de acceso. • Existe un método que se ejecuta al momento de revocar el perfil de acceso.

Propiedad de Seguridad	Id	Pregunta	Parámetros POA a cumplirse
Auditoría	**5**	¿Se audita la asignación de perfiles de acceso?	• Existe un módulo de administración de usuarios en el sistema. • Es posible asignar perfiles de acceso. • Existe un método que se ejecuta al momento de asignar el perfil de acceso.
Auditoría	**6**	¿El acceso a información sensible es auditado?	• El sistema procesa o maneja información privada de personas y/o empresas, por ejemplo, ciertos datos personales y bancarios, contraseñas de correo electrónico, domicilio. • Existe una interface que permita visualizar la información sensible. • Existe un método que se ejecuta al momento de acceder a la información sensible.
Auditoría	**7**	¿Se audita la creación de usuarios?	• Existe un módulo de administración de usuarios en el sistema. • Existe un método que se ejecuta al momento de crear un usuario.
Auditoría	**8**	¿Se audita la creación de perfiles de acceso?	• Existe un módulo de administración de usuarios en el sistema. • Es posible crear perfiles de acceso. • Existe un método que se ejecuta al momento de crear el perfil de acceso.

Propiedad de Seguridad	Id	Pregunta	Parámetros POA a cumplirse
Auditoría	**9**	¿Se audita los accesos a tablas de logs?	• Existe una interface para poder consultar el registro de logs. • Existe un método que se ejecuta al momento de consultar el registro de logs.
Auditoría	**10**	¿La información sensible es almacenada en tablas de logs de manera cifrada?	• El sistema procesa o maneja información privada de personas y/o empresas, por ejemplo, ciertos datos personales y bancarios, contraseñas de correo electrónico, domicilio. • Para almacenar la información sensible el sistema lo hace por medio de métodos.
Auditoría	**11**	¿Existe una interface que permita hacer consultas sobre los logs?	• No se pueden definir parámetros de evaluación POA porque hacen referencia a la existencia de una pantalla lo cual con POA no puede ser resuelto.

Propiedad de Seguridad	Id	Pregunta	Parámetros POA a cumplirse
Auditoría	**12**	¿Las tablas de logs cuentan con al menos la siguiente información? Fecha y hora local, dirección IP del usuario que generó el log y opción del sistema que se trató de ejecutar.	• El sistema cuenta con tablas de registros de logs. • Las herramientas con las cuales está desarrollado el sistema permiten capturar la fecha y hora local. • Las herramientas con las cuales está desarrollado el sistema permiten capturar la IP desde la cual se lo invoca. • Las herramientas con las cuales está desarrollado el sistema permiten capturar la opción o proceso del sistema que se trató de ejecutar.
Autenticación	**13**	¿El proceso de autenticación de usuarios usa tablas propias del esquema del sistema?	• No se pueden definir parámetros de evaluación POA porque hacen referencia a una especificación a nivel de base de datos lo cual con POA no puede ser resuelto.
Autenticación	**14**	¿Las contraseñas de los usuarios que acceden al sistema se someten a un proceso de hashing (Algoritmo de Encriptación)?	• Existe un método que se ejecuta al momento de crear las credenciales de usuario de acceso al sistema. • Existe un método que se ejecuta cuando un usuario desea autenticarse en el sistema.
Autenticación	**15**	¿Existe un proceso de verificación de usuarios?	• Existe un método que se ejecuta cuando un usuario desea autenticarse en el sistema.

Propiedad de Seguridad	Id	Pregunta	Parámetros POA a cumplirse
Autenticación	**16**	¿Por intentos fallidos de acceso al sistema el usuario es bloqueado?	• Existe un método que se ejecuta cuando un usuario desea autenticarse en el sistema.
Autenticación	**17**	¿Se cuenta con un proceso de desbloqueo de usuarios?	• Existe un método que se ejecuta cuando un usuario desea autenticarse en el sistema.
Autenticación	**18**	¿Existe un factor de autenticación para procesos que manejan información sensible?	• El sistema procesa información sensible. • El procesamiento de información sensible se lo hace por medio de métodos propios del sistema.
Autenticación	**19**	¿Se controla el uso histórico de contraseñas?	• Existe un módulo de administración de usuarios en el sistema. • Existe un método que se ejecuta al momento de crear las credenciales de usuario de acceso al sistema.
Autenticación	**20**	¿Se controla que la contraseña del usuario no contenga porciones del nombre del usuario?	• Existe un módulo de administración de usuarios en el sistema. • Existe un método que se ejecuta al momento de crear las credenciales de usuario de acceso al sistema.
Autenticación	**21**	¿Para la creación de contraseñas se controla que tenga como un mínimo de 8 caracteres y un máximo de 20 caracteres?	• Existe un módulo de administración de usuarios en el sistema. • Existe un método que se ejecuta al momento de crear las credenciales de usuario de acceso al sistema.

Propiedad de Seguridad	Id	Pregunta	Parámetros POA a cumplirse
Autenticación	**22**	¿Se maneja un estándar de creación de contraseñas? Por ejemplo: La contraseña debe contener al menos una letra mayúscula, números y un carácter especial.	• Existe un módulo de administración de usuarios en el sistema. • Existe un método que se ejecuta al momento de crear las credenciales de usuario de acceso al sistema.
Autenticación	**23**	¿Existe un proceso de recuperación de contraseñas cuando el usuario olvido la misma?	• Existe un módulo de administración de usuarios en el sistema. • Existe un método que se ejecuta al momento de crear las credenciales de usuario de acceso al sistema.
Autenticación	**24**	¿La contraseña tiene un tiempo de caducidad?	• Existe un módulo de administración de usuarios en el sistema. • Existe un método que se ejecuta al momento de crear las credenciales de usuario de acceso al sistema.
Autenticación	**25**	¿Existe un segundo factor de autenticación?	• Existe un módulo de administración de usuarios en el sistema. • Existe un método que se ejecuta al momento de crear las credenciales de usuario de acceso al sistema.

Propiedad de Seguridad	Id	Pregunta	Parámetros POA a cumplirse
Autenticación	**26**	¿El segundo factor de autenticación tiene una duración temporal (minutos)?	• Existe un módulo de administración de usuarios en el sistema. • Existe un método que se ejecuta al momento de crear las credenciales de usuario de acceso al sistema.
Autenticación	**27**	¿Se controla que la baja de un usuario del sistema sea a nivel lógico y no físico?	• Existe un módulo de administración de usuarios en el sistema. • Existe un método que se ejecuta al momento de eliminar un usuario del sistema.
Autenticación	**28**	¿La sesión de usuario se caduca si existe inactividad?	• Cuando un usuario se autentica en el sistema se crea una nueva sesión de usuario. • Los procesos que un usuario puede ejecutar en el sistema se lo hacen mediante métodos.
Autenticación	**29**	¿Existe control de sesión en todos los módulos del sistema?	• Cuando un usuario se autentica en el sistema se crea una nueva sesión de usuario. • Los procesos que un usuario puede ejecutar en el sistema se lo hacen mediante métodos.
Autorización	**30**	¿Se asignan perfiles de acceso a los usuarios?	• Existe un módulo de administración de usuarios en el sistema. • Existen perfiles de acceso creados en el sistema. • Existe un método que se ejecuta al momento de crear o modificar un usuario en el sistema.

Propiedad de Seguridad	Id	Pregunta	Parámetros POA a cumplirse
Autorización	**31**	¿Los perfiles de acceso limitan la ejecución de procesos?	• Existen perfiles de acceso creados en el sistema. • El usuario tiene asignado su respectivo perfil de acceso. • Los procesos que un usuario puede ejecutar en el sistema se lo hacen mediante métodos
Autorización	**32**	¿Se controla que el usuario solamente pueda tener asignado un solo perfil de acceso?	• Existe un módulo de administración de usuarios en el sistema. • Existen perfiles de acceso creados en el sistema. • Existe un método que se ejecuta al momento de crear o modificar un usuario en el sistema.

3.2. Descripción del Proceso Metodológico

3.2.1. Metodología de Desarrollo

La metodología de desarrollo que se va a usar es Test – Driven Development (TDD), debido a que es una metodología ágil y la definición de requerimientos es explícita porque no se necesitan a los usuarios para definirlos y retroalimentar debido a que son requerimientos no funcionales. El proceso de TDD seguirá los siguientes pasos:

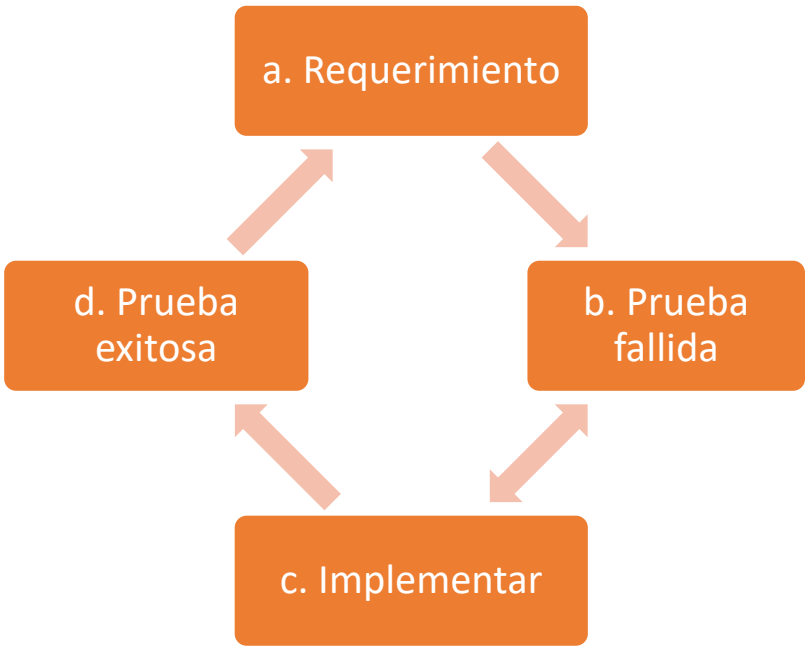


Figura 4 *Ciclo de vida de TDD*

El requerimiento (a) se lo plasmará en una historia de usuario que contará con las siguientes secciones: Requerimiento, identificador, descripción y prueba de aceptación.

La prueba fallida (b) y la prueba exitosa (d) serán los Test Case que representan la escritura del código de la prueba de unidad y que son la razón de ser del TDD que servirán para verificar si se ha desarrollado una funcionalidad en particular y cumple con los criterios de aceptación.

La implementación (c) representa la codificación de los aspectos que equivalen a los requerimientos de seguridad que se han definido.

3.2.2. Ejemplo Práctico de POA

a. Requerimiento

Se define el requerimiento mediante una historia de usuario.

Tabla 2 Requerimiento ejemplo práctico

Se requiere que el sistema imprima un mensaje antes de la ejecución de un método.
Descripción: Utilizando POA se debe interceptar al método de la capa del negocio e imprimir un mensaje antes de la ejecución del mismo.
Pruebas de Aceptación: 1. Antes de la ejecución del método se debe imprimir un mensaje. 2. Durante y después la ejecución del método no se debe imprimir ningún mensaje.

b. Prueba Fallida

En la Figura 5 se muestra la ejecución fallida de la prueba de unidad porque la funcionalidad a evaluar no se encuentra implementada.

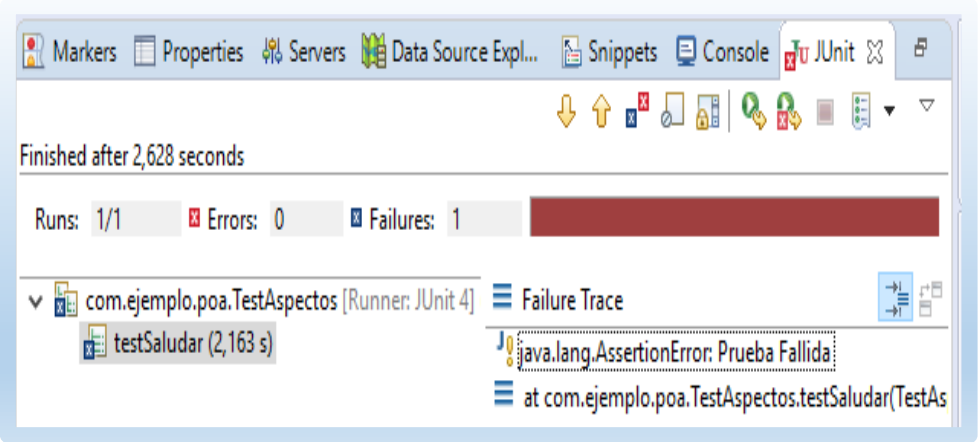


Figura 5 *Falla la prueba de Unidad*

c. Implementar

A continuación, se describe el código y los archivos de configuración que forman parte del ejemplo.

Interface SaludoService.java

```
1  package com.ejemplo.poa;
2
3  public interface SaludoService {
4
5      public String saludar(String nombre, String palabraSaludo);
6
7  }
```

Figura 6 *Interface SaludoService.java*

En la Figura 6 se muestra el código de la interface en la cual se declara el método saludar.

Clase SaludoServiceImpl.java

En la Figura 7 se muestra el código de la implementación de la interface SaludoService.java

```java
package com.ejemplo.poa;

public class SaludoServiceImpl implements SaludoService {

    public String saludar(String nombre, String palabraSaludo){
        String saludo = palabraSaludo + " " + nombre;
        System.out.println(saludo);
        return saludo;
    }

}
```

Figura 7 *Clase SaludoServiceImpl.java*

Clase Main.java

En la Figura 8 que corresponde a la clase Main.java de la cual es importante resaltar:

1. Creación del contexto de Spring.

```java
package com.ejemplo.poa;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class Main {

    public static void main(String[] args) {

        ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext("beans.xml");

        try {
            SaludoService saludo = (SaludoService) context.getBean("saludoService");
            saludo.saludar("Valeria", "buenos dias");


        } finally {
            context.close();
        }
    }

}
```

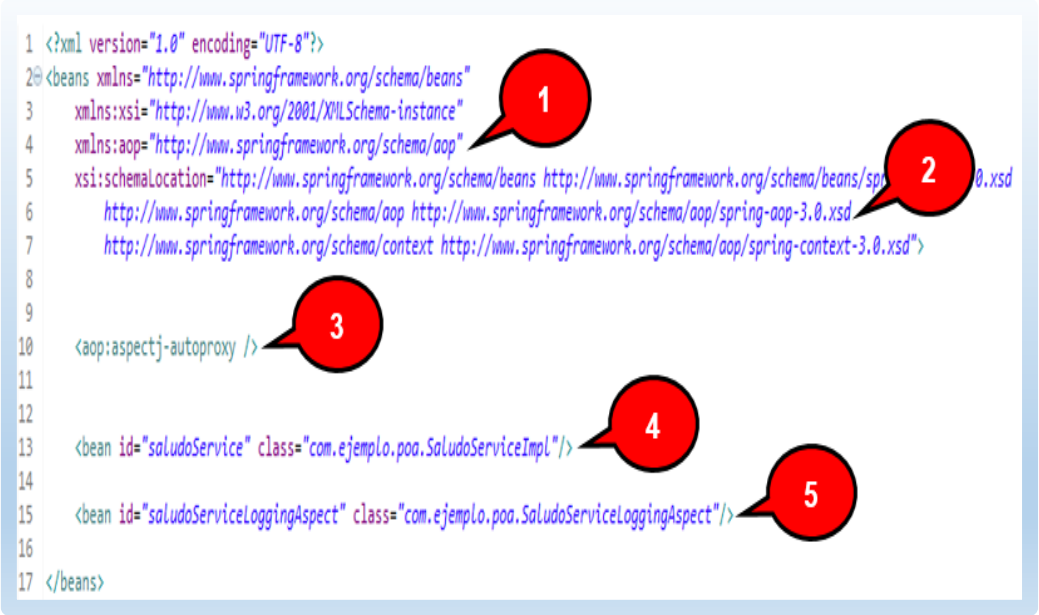
Figura 8 *Clase Main.java*

2. Inyección del bean saludoService.

Archivo beans.xml

Este es el archivo de configuración de Spring el cual se encuentra representado en la Figura 9 y de la que se explican a continuación las configuraciones marcadas en la figura:

1. Se define el namespace de AOP.
2. Se define el esquema de AOP.
3. Se activa el uso de aspectos con el tag **<aop:aspectj-autoproxy />**, para lo cual se debe tener definido el namespace y el esquema de AOP.
4. Se crea el bean **saludoService** que representa la lógica del negocio.
5. Se crea el bean **saludoServiceLoggingAspect** que representa la funcionalidad transversal (aspecto).




```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <beans xmlns="http://www.springframework.org/schema/beans"
3      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4      xmlns:aop="http://www.springframework.org/schema/aop"
5      xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spr                0.xsd
6          http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-3.0.xsd
7          http://www.springframework.org/schema/context http://www.springframework.org/schema/aop/spring-context-3.0.xsd">
8
9
10     <aop:aspectj-autoproxy />
11
12
13     <bean id="saludoService" class="com.ejemplo.poa.SaludoServiceImpl"/>
14
15     <bean id="saludoServiceLoggingAspect" class="com.ejemplo.poa.SaludoServiceLoggingAspect"/>
16
17 </beans>
```

Figura 9 *Archivo de configuración de spring beans.xml*

Clase SaludoServiceLoggingAspect.java

A continuación, se explican los puntos resaltados en la Figura 10 que corresponde a los puntos importantes de la creación de un aspecto:

1. La notación @**Aspect** que es propia de **aspectj** y **spring** la usa, indica que la clase es un aspecto.
2. El método **logAntes** representa un advice que contiene la funcionalidad del aspecto. El advice es notado con @**Before** para que se invoque antes de la ejecución del método y recibe como argumento un **join point**.

3. En el **join point** se debe definir una regla que es una expresión regular que se le conoce como **pointcut** que agrupa uno o más puntos de unión.


```
1  package com.ejemplo.poa;
2
3
4  import java.util.Arrays;
5
6  import org.apache.commons.logging.*;
7  import org.aspectj.lang.JoinPoint;
8  import org.aspectj.lang.annotation.Aspect;
9  import org.aspectj.lang.annotation.Before;
10
11 @Aspect
12 public class SaludoServiceLoggingAspect {
13
14     private = LogFactory.getLog(this.getClass());
15
16     @Before("execution(String SaludoService.saludar(..))")
17     public void logAntes(JoinPoint joinPoint){
18         String nombreMetodo = joinPoint.getSignature().getName();
19         String argumentos = Arrays.toString(joinPoint.getArgs());
20         log.info("Antes: " + nombreMetodo + " con los parametros" + argumentos);
21     }
22 }
```

Figura 10 *Clase SaludoServiceLoggingAspect.java*

d. Prueba Exitosa

Una vez que se implementó lo solicitado en el requerimiento se ejecuta el Test y se tiene como resultado que la prueba es exitosa tal como se muestra en la Figura 11.

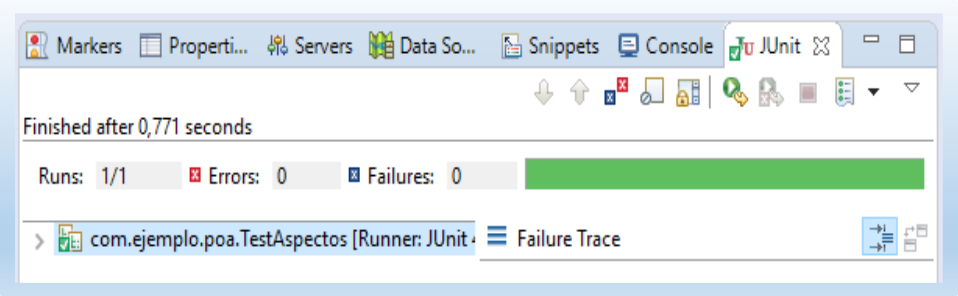


Figura 11 *Prueba de Unidad Exitosa*

Al ejecutar la aplicación en la consola (Figura 12), se imprime el mensaje del método **logAntes** del aspecto que fue implementado para que se ejecute antes del método **saludar** (1) y luego se tiene la impresión del mensaje del método **saludar** que para el ejemplo representa la capa del negocio (2).

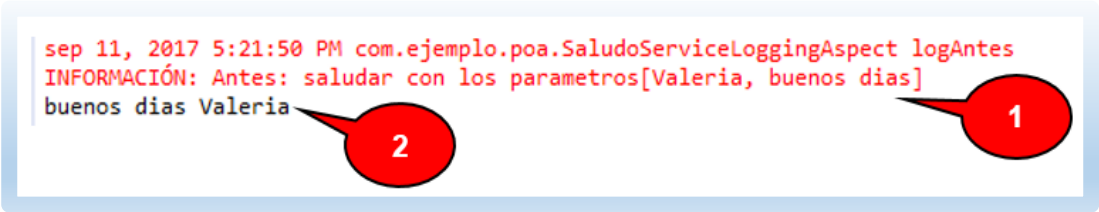


Figura 12 *Resultado ejecución Main.java*

3.3. Caso de Estudio

3.3.1. Encuesta

En la empresa Corporación Tecnología De La Información Solutandi Cia. Ltda. se aplicó una encuesta (Anexo 1) para seleccionar el caso de estudio sobre el cual se implementará el componente de seguridad propuesto en el presente proyecto integrador.

La empresa fue seleccionada porque es el lugar de trabajo de uno de los autores del presente documento y adicionalmente brindó todas las facilidades del caso. Es muy importante recalcar que no se divulgará información crítica.

- **Sujetos Encuestados**

La encuesta se aplicó al Gerente General, Gerentes de Departamento y Comerciales; ya que son quienes conocen a fondo cada uno de los sistemas

que la empresa comercializa porque fueron desarrolladores de los mismos; por lo tanto, cuentan con el conocimiento suficiente para poder responder las preguntas planteadas en la encuesta.

- **Número de Sujetos Encuestados**

El número de personas a encuestar son 6.

- **Objetivo General**

Determinar el sistema sobre el cual se va a implementar el componente de seguridad propuesto en el presente proyecto integrador.

- **Modalidad de la Encuesta**

Debido a que los funcionarios de la Corporación Tecnología De La Información Solutandi Cia. Ltda. se encuentran físicamente trabajando en las instalaciones de la empresa, se aplica personalmente la encuesta.

3.3.2. Caso de Estudio Seleccionado

Se tomó la sección I de la encuesta para poder determinar el sistema que menos cumple con las propiedades de seguridad evaluadas; luego de la tabulación de los resultados (Anexo 2) el sistema que menos cumple con las propiedades de seguridad evaluadas es el Tandi – Invoice como se muestra en la Figura 13.

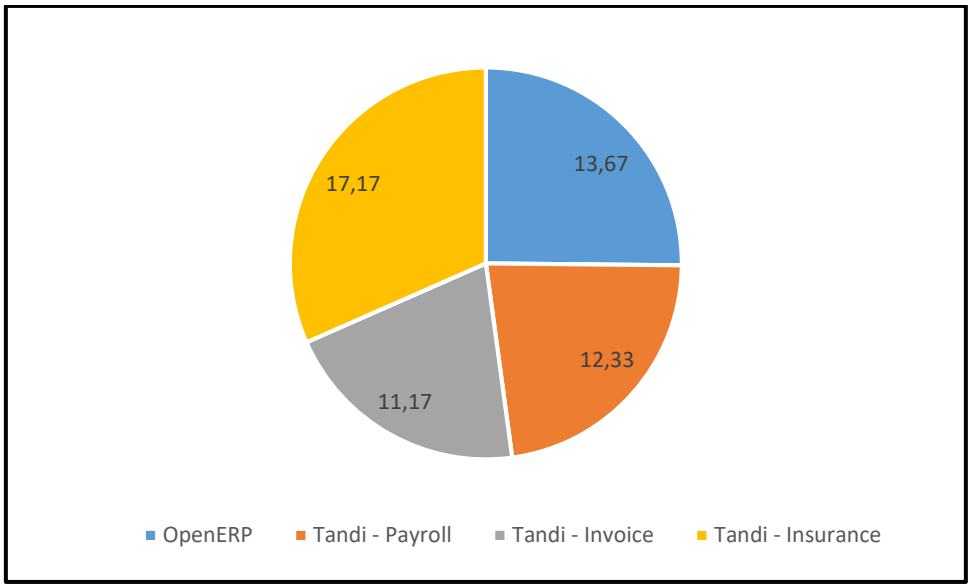


Figura 13 *Cumplimiento de Propiedades de Seguridad*

En base al resultado de la encuesta se tiene la Tabla 3 en la cual se muestran marcadas las preguntas que si tienen la funcionalidad implementada con POO en el sistema Tandi – Invoice.

Tabla 3 Funcionalidad implementada en Tandi – Invoice

Propiedad de Seguridad	Preguntas	Tiene la funcionalidad
Auditoría	1. ¿Son auditados los intentos fallidos de acceso al sistema?	
Auditoría	2. ¿Se auditan los accesos exitosos al sistema?	
Auditoría	3. ¿Se auditan todos los intentos de comunicación con otros sistemas?	✓
Auditoría	4. ¿Se audita la revocación de perfiles de acceso?	
Auditoría	5. ¿Se audita la asignación de perfiles de acceso?	
Auditoría	6. ¿El acceso a información sensible es auditado?	

Propiedad de Seguridad	Preguntas	Tiene la funcionalidad
Auditoría	7. ¿Se audita la creación de usuarios?	✓
Auditoría	8. ¿Se audita la creación de perfiles de acceso?	✓
Auditoría	9. ¿Se audita los accesos a tablas de logs?	
Auditoría	10. ¿La información sensible es almacenada en tablas de logs de manera cifrada?	
Auditoría	11. ¿Existe una interface que permita hacer consultas sobre los logs?	
Auditoría	12. ¿Las tablas de logs cuentan con al menos la siguiente información? Fecha y hora local, dirección IP del usuario que genero el log y opción del sistema que se trató de ejecutar.	
Autenticación	13. ¿El proceso de autenticación de usuarios usa tablas propias del esquema del sistema?	✓
Autenticación	14. ¿Las contraseñas de los usuarios que acceden al sistema se someten a un proceso de hashing (Algoritmo de Encriptación)?	✓
Autenticación	15. ¿Existe un proceso de verificación de usuarios?	✓
Autenticación	16. ¿Por intentos fallidos de acceso al sistema el usuario es bloqueado?	
Autenticación	17. ¿Se cuenta con un proceso de desbloqueo de usuarios?	
Autenticación	18. ¿Existe un factor de autenticación para	

Propiedad de Seguridad	Preguntas	Tiene la funcionalidad
	procesos que manejan información sensible?	
Autenticación	19. ¿Se controla el uso histórico de contraseñas?	
Autenticación	20. ¿Se controla que la contraseña del usuario no contenga porciones del nombre del usuario?	
Autenticación	21. ¿Para la creación de contraseñas se controla que tenga como un mínimo de 8 caracteres y un máximo de 20 caracteres?	
Autenticación	22. ¿Se maneja un estándar de creación de contraseñas? Por ejemplo: La contraseña debe contener al menos una letra mayúscula, números y un carácter especial.	
Autenticación	23. ¿Existe un proceso de recuperación de contraseñas cuando el usuario olvido la misma?	
Autenticación	24. ¿La contraseña tiene un tiempo de caducidad?	
Autenticación	25. ¿Existe un segundo factor de autenticación?	
Autenticación	26. ¿El segundo factor de autenticación tiene una duración temporal (minutos)?	
Autenticación	27. ¿Se controla que la baja de un usuario del sistema sea a nivel lógico y no físico?	✓
Autenticación	28. ¿La sesión de usuario se caduca si existe inactividad?	✓

Propiedad de Seguridad	Preguntas	Tiene la funcionalidad
Autenticación	29. ¿Existe control de sesión en todos los módulos del sistema?	✓
Autorización	30. ¿Se asignan perfiles de acceso a los usuarios?	✓
Autorización	31. ¿Los perfiles de acceso limitan la ejecución de procesos?	✓
Autorización	32. ¿Se controla que el usuario solamente pueda tener asignado un solo perfil de acceso?	

Tomando la sección II de la encuesta se obtuvieron las siguientes características técnicas del sistema Tandi – Invoice:

- Los tipos de usuarios que se pueden autenticar son: Personas y Sistemas.
- Tiene dos subsistemas los cuales se describen brevemente a continuación:

  TandiESB - Portal: Sistema en el cual se visualizan los documentos autorizados y realiza gestión de usuarios.

  TandiESB - Invoice: Sistema que se encarga de la integración con los sistemas empresariales y el SRI; realiza también la notificación de la autorización de los documentos.

- Las capas del negocio de los subsistemas del Tandi – Invoice fueron desarrolladas en los lenguajes de programación que se muestran en la Tabla 4.

Tabla 4 Lenguajes de Programación sobre los cuales fue construido Tandi – Invoice

Subsistema	Lenguaje de Programación
TandiESB - Portal	.net
TandiESB - Invoice	Java

Tandi – Invoice al estar formado por dos subsistemas es necesario determinar sobre cuál de estos se aplicará el componente se seguridad y para ello se define como criterio de selección: *El subsistema en el cual se ejecutan los procesos principales y manejo de información sensible.* Tomando como referencia la Figura 15 en la que se detalla la arquitectura del Tandi - Invoice, se puede concluir que el subsistema que cumple con el criterio de selección es el TandiESB – Invoice.

Por lo tanto, el subsistema **TandiESB – Invoice** del sistema Tandi – Invoice es el caso de estudio sobre el cual se implementará el componente de seguridad desarrollado con POA.

3.3.3. Características Técnicas del Caso de Estudio Seleccionado

El Sistema Tandi – Invoice es un sistema especializado en la gestión de los procesos para la autorización electrónica de documentos tributarios con el SRI, además gestiona el manejo, envío y entrega de estos documentos a los clientes de una manera integrada, fácil y sencilla para las empresas. En la Figura 14 se muestra globalmente el flujo de autorización, notificación, visualización e integración del sistema Tandi – Invoice con los sistemas empresariales emisores de documentos y el SRI.

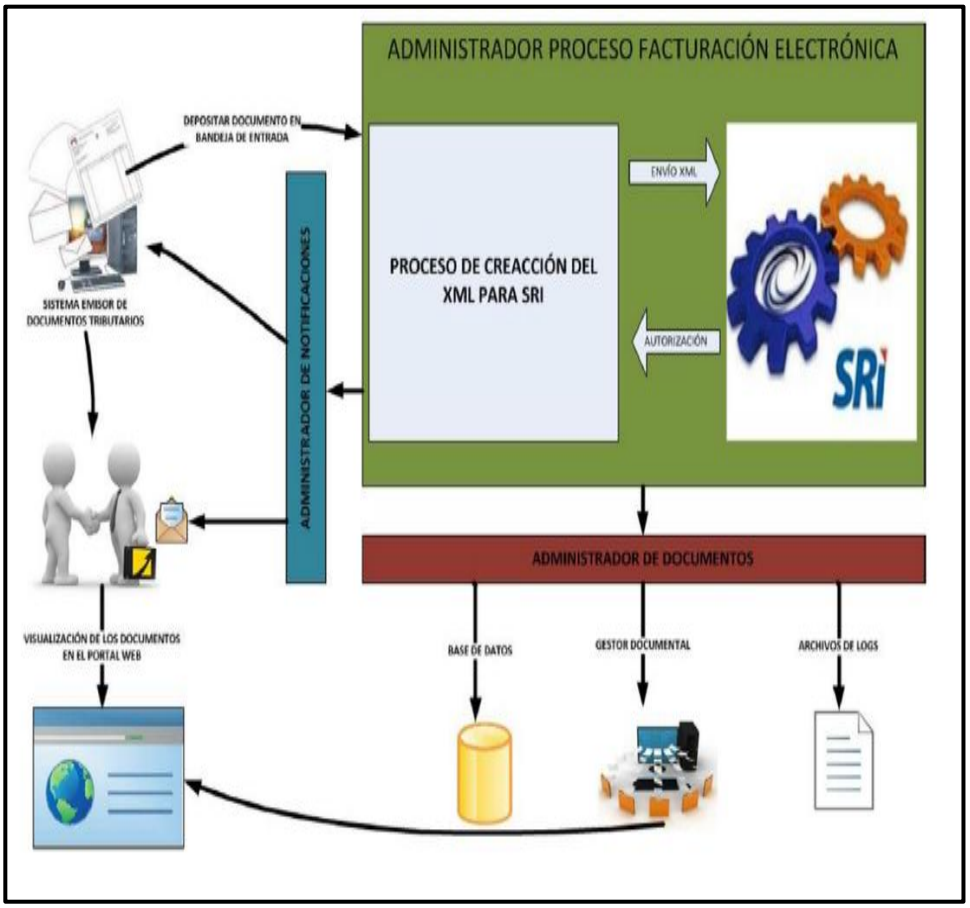


Figura 14 *Flujo de procesos e integración Tandi – Invoice*

3.3.1. Arquitectura del Sistema Tandi – Invoice

La arquitectura del sistema Tandi – Invoice se encuentra representada en la Figura 15 correspondiente al diagrama de componentes.

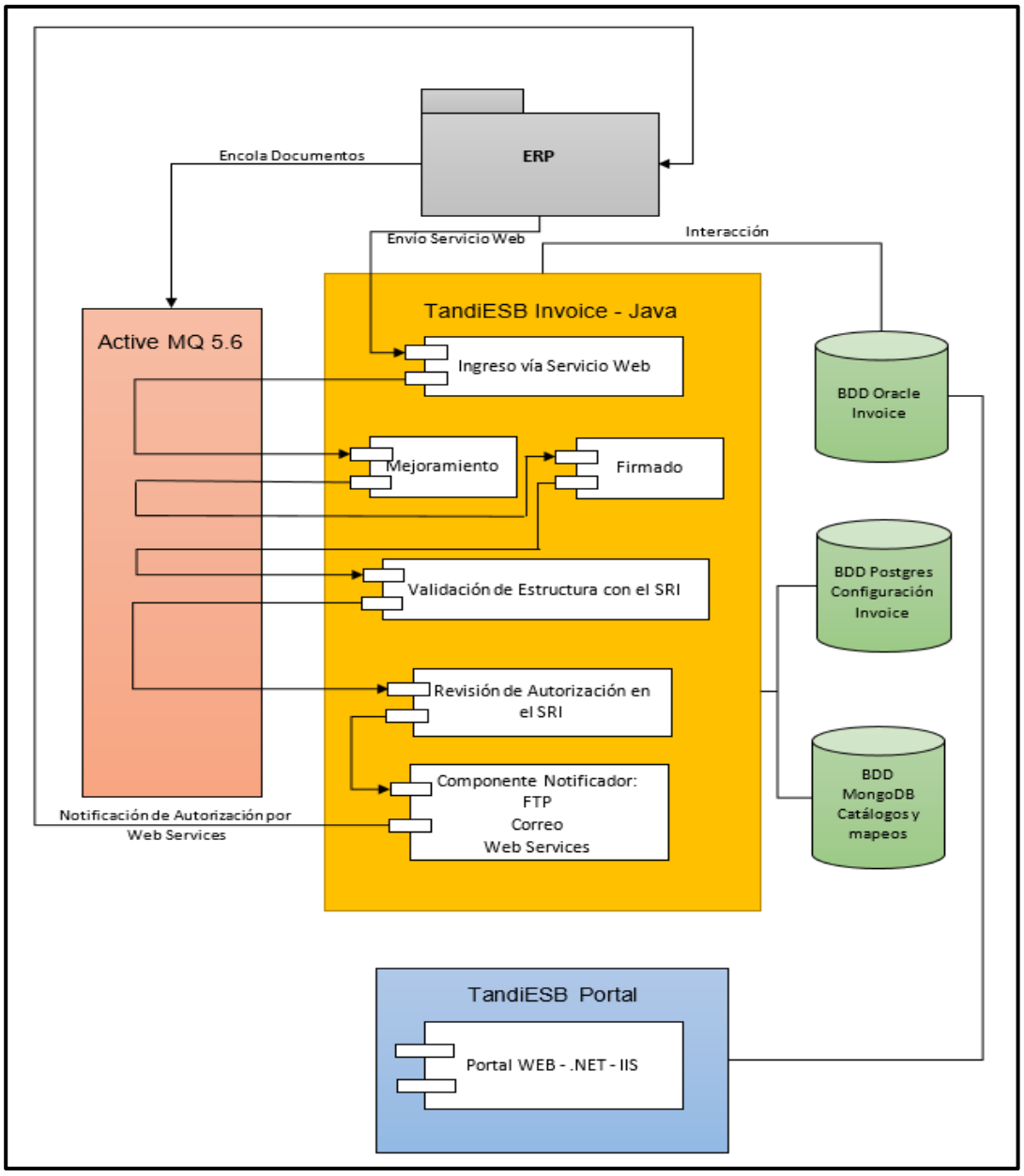


Figura 15 *Diagrama de Componentes Tandi – Invoice*

3.3.2. Herramientas a Usar

Entre las herramientas POA existentes en la actualidad se destacan Aspectj, Jboss-AOP y Spring Framework, aunque más allá de estas tres existe una cantidad importante de herramientas casi todas ellas extensiones a lenguajes Orientado a Objetos [aspectc++], [Loom.NET], [aspectDNG], [Naspect], [aspect#] [9].

Sin duda, gran parte del camino de la evolución de POA lo ha marcado AspectJ, siendo la primera herramienta estable y la de mayor difusión en la actualidad. Al mismo tiempo hay que destacar el aporte de SpringFramework, promoviendo el uso de AOP en ambientes enterprise [9].

A continuación, se explicará brevemente sobre estas dos herramientas.

AspectJ

AspectJ [aspectj] surgió como consecuencia del trabajo de Gregor Kiczales y su equipo en XPARC. A partir de la creciente actividad de la comunidad formada en torno a AspectJ, en 2001 aspectJ pasó a la órbita del proyecto Eclipse [eclipse]. Tal vez por ser la primera herramienta POA, ha marcado en gran parte el rumbo de la comunidad AOSD [9].

AspectJ es una extensión a Java. Desde el punto de vista de la implementación, en un comienzo fue un pre compilador, pero actualmente es un compilador que genera Java Byte Code totalmente compatible con la JVM de Sun Microsystems. En el año 2005, AspectJ se fusionó con aspectwerkz,

incorporando varias funcionalidades de este último relacionadas al entretejido dinámico. La mayoría de los libros sobre POA en la actualidad tratan sobre AspectJ [9].

Spring Framework

Spring Framework es un framework de código abierto creado por Rod Johnson para facilitar el desarrollo de aplicaciones enterprise inicialmente en Java y actualmente también en .NET. Este framework consta de varios módulos, uno de los cuales es SpringAOP [9].

SpringAOP utiliza entretejido dinámico y puede utilizarse independientemente de los otros módulos. Un punto interesante de Spring Framework es que varios de sus módulos hacen uso del módulo POA. En el caso de la implementación Java, ha tenido un gran nivel de adopción convirtiéndose en el estándar de facto para algunas compañías. La versión 2.0 ha dado un paso interesante hacia la integración con AspectJ. En cuanto a la implementación .NET, el módulo POA de Spring, es la implementación POA más completa, estable y utilizada de POA en la actualidad [9].

Como ya se explicó en la sección 3.2 se decidió trabajar con el subsistema de Tandi – Invoice que está desarrollado en java (TandiESB - Invoice) y en base a esta premisa se seleccionaron las siguientes herramientas para la construcción del componente de seguridad:

- Jdk 1.7.

- Spring Framework 3.1
- Maven 4.0
- IDE IntelliJ IDEA 13.3.6

3.4. Desarrollo de Aspectos de Seguridad

3.4.1. Desarrollo del Análisis de Factibilidad Tecnológica

Aplicando la principal interrogante que plantea el Análisis de Factibilidad Tecnológica al presente proyecto, se desea saber lo siguiente:

¿Es posible implementar seguridad lógica en un sistema aplicando POA?

Para concluir si se puede llevar o no a cabo el proyecto propuesto a continuación, se responderán las preguntas que plantea la teoría del punto anterior:

1. **¿Actualmente se posee la tecnología necesaria?**

Para implementar seguridad lógica utilizando POA los recursos tecnológicos mínimos con los que se cuenta son los siguientes:

En cuanto a hardware se dispone de una PC de escritorio que soporta virtualización y tiene las siguientes características:

- Procesador: Intel Core i5-3470 2.90 GHz
- RAM: 12 GB
- Disco Duro: 672 GB Totales – 349 GB Disponibles

En cuanto a software la PC descrita anteriormente está configurada con el siguiente sistema operativo:

- SO: Windows 10 Home Edition 64 bits

Se puede determinar que los recursos tecnológicos descritos son suficientes para poder implementar componentes de seguridad lógica a través de POA, ya que los lenguajes de programación que soportan el paradigma en cuestión – por ejemplo: PHP, Java, .Net, entre otros – requieren como mínimo menos de los recursos disponibles lo cual no es cuestión de estudio del presente proyecto integrador.

2. **¿Se posee la experiencia técnica necesaria?**

Los autores de este proyecto integrador han alcanzado la experiencia técnica necesaria en cuanto a Seguridad Lógica y POA durante su vida estudiantil y laboral, y durante el desarrollo de los capítulos I y II del presente trabajo.

3. **¿La solución resuelve el problema planteado?**

En base a la problemática planteada en el capítulo I y el análisis de esta a través de la teoría descrita en el capítulo II se determina que el problema planteado se puede abarcar y resolver.

Por lo tanto, se concluye que sí es posible implementar componentes de seguridad lógica con POA y que está implementación se realizará en este trabajo.

3.4.2. Requerimientos de Seguridad

a. Análisis de los Requerimientos de Seguridad

A continuación, se muestra el resultado de la evaluación de cada uno de los parámetros definidos en la Tabla 1 sobre el caso de estudio seleccionado; es importante recalcar que este procedimiento se lo hace para poder determinar en base a la revisión del código del sistema cuantas preguntas pueden convertirse en requerimientos y ser implementados con POA.

Por lo tanto, en base a los resultados que se muestran en el Anexo 3 se puede concluir que las preguntas que tienen el más alto nivel de cumplimiento son las que se convertirán en requerimientos para ser implementados con POA y son las siguientes: 1, 2, 10, 12, y 16.

b. Definición de Requerimientos de Seguridad

Tabla 5 Requerimientos de Seguridad a implementar

Propiedad de Seguridad	Id	Pregunta	Requerimiento	Observaciones
Auditoría	**1**	¿Son auditados los intentos fallidos de acceso al sistema?	Auditar los intentos fallidos de acceso al sistema. Almacenando la siguiente información:	

Propiedad de Seguridad	Id	Pregunta	Requerimiento	Observaciones
			• Fecha y hora local • Dirección IP desde la cual se intentó acceder al sistema. • Proceso que se trató de ejecutar.	
Auditoría	**2**	¿Se auditan los accesos exitosos al sistema?	Auditar los accesos exitosos al sistema. Almacenando la siguiente información: • Fecha y hora local • Dirección IP desde la cual	

Propiedad de Seguridad	Id	Pregunta	Requerimiento	Observaciones
			se intentó acceder al sistema. • ID del usuario. • Proceso que se ejecutó.	
Auditoría	**10**	¿La información sensible es almacenada en tablas de logs de manera cifrada?	Almacenar la información sensible de manera cifrada en tablas de logs.	La pregunta 10 y 12 se convierten en un solo requerimiento ya que ambas preguntas están relacionadas con tablas de logs. Se identificará al requerimiento por la combinación de los identificador
Auditoría	**12**	¿Las tablas de logs cuentan con al menos la siguiente información? Fecha y hora local, dirección IP del usuario que genero el log y opción del sistema que se trató de ejecutar.	En los requerimientos que se van a implementar y que se trabaja con tablas de logs controlar que al menos se tenga la siguiente información: • Fecha y hora local	

Propiedad de Seguridad	Id	Pregunta	Requerimiento	Observaciones
			• Dirección IP del usuario que genero el log. • Opción del sistema que se trató de ejecutar o proceso que se ejecutó.	es de sus preguntas es decir 10 - 12
Autenticación	**16**	¿Por intentos fallidos de acceso al sistema el usuario es bloqueado?	Bloquear al usuario por intentos fallidos de autenticación en el sistema.	

En base a los requerimientos definidos en la Tabla 5 se detallan las especificaciones para cada uno.

Tabla 6 Especificación Requerimiento 1

Se requiere auditar los intentos fallidos de acceso al sistema.
Identificador: 1
Descripción: Cada vez que se intente autenticar un usuario no registrado en el sistema o con credenciales incorrectas, se deben auditar cada uno de esos intentos almacenando la siguiente información: • Fecha y hora local. • Dirección IP desde la cual se intentó acceder al sistema. • Proceso que se trató de ejecutar. • Nombre de usuario. • Id de usuario si el mismo se encuentra registrado en el sistema. • Id de la compañía a la que se intenta acceder.
Pruebas de Aceptación: 1. Si el intento de autenticación es fallido se debe insertar un registro en la tabla de auditoría creada. 2. Si el intento de autenticación es exitoso no se debe insertar un registro en la tabla de auditoría creada.

Tabla 7 Especificación Requerimiento 2

Se necesita auditar los intentos exitosos de acceso al sistema.
Identificador: 2
Descripción: Cada vez que se intente autenticar un usuario registrado en el sistema, se deben auditar cada uno de esos accesos almacenando la siguiente información: • Fecha y hora local. • Dirección IP desde la cual se accedió al sistema. • Id de Usuario. • Nombre de usuario. • Proceso que se ejecutó. • Id de la compañía a la que se accede.
Pruebas de Aceptación: 1. Si el intento de autenticación es válido se debe generar un registro en la tabla de auditoría de autenticaciones exitosas. 2. Si el intento de autenticación no es válido no se debe generar registros en la tabla de auditoría de autenticaciones exitosas.

Tabla 8 Especificación Requerimiento 10 - 12

Se requiere almacenar la información sensible en tablas de logs de manera cifrada.
Identificador: 10 - 12
Descripción: Cada vez que se reciba un documento para ser procesado en el sistema se almacenara en una tabla de log el ingreso del documento. En esta tabla se almacenará de manera cifrada los datos personales del cliente y clave de acceso del documento de manera cifrada; ya que esta información se la considera sensible. Dicha tabla deberá contar con la siguiente información: • Fecha y hora local. • Dirección IP del usuario que genero el log. • Proceso que se ejecutó. • Id de la compañía que envía la transacción. • Número de documento. • Detalle de la Transacción.
Prueba de Aceptación: 1. Por cada vez que un documento ingrese en el sistema se debe generar un registro en la tabla de log.

Tabla 9 Especificación Requerimiento 16

Se requiere bloquear al usuario por intentos fallidos de autenticación en el sistema.
Identificador: 16
Descripción: El usuario será bloqueado en el sistema cuando haya intentado por 3 ocasiones autenticarse con credenciales incorrectas. Para controlar el número de intentos fallidos de autenticación se consultará en la tabla de auditoría creada en el Requerimiento 1.
Pruebas de Aceptación: 1. Si los intentos de autenticaciones fallidas son mayores o iguales a 3 el usuario será bloqueado y no podrá autenticarse en el sistema. 2. Si el intento de autenticación es exitoso no se debe bloquear al usuario.

3.4.3. Pruebas Fallidas de los Requerimientos de Seguridad

Se escriben las pruebas unitarias para cada uno de los requerimientos, en esta fase del desarrollo todas las pruebas fallan, porque los requerimientos aún no se encuentran implementados.

Prueba Unitaria Requerimiento 1:

- ***testValidaAutenticacionFallida:*** Verifica si se insertan registros en la tabla de auditoría AUTENTICACION_FALLIDA si las credenciales de usuario no son correctas, cuando no se hayan insertado registros la prueba falla.
- ***testValidaAutenticacionNoFallida:*** Verifica si no se insertan registros en la tabla de auditoría AUTENTICACION_FALLIDA cuando las credenciales de usuario son correctas. En caso de que se hayan insertado registros la prueba falla.

Prueba Unitaria Requerimiento 2:

- ***testValidaAutenticacionExitosa:*** Verifica si se insertan registros en la tabla de auditoría AUTENTICACION_EXITOSA si las credenciales de usuario son correctas, cuando no se hayan insertado registros la prueba falla.
- ***testValidaAutenticacionNoExitosa:*** Verifica si no se insertan registros en la tabla de auditoría AUTENTICACION_EXITOSA cuando las credenciales de usuario no son correctas. En caso de que se hayan insertado registros la prueba falla.

Prueba Unitaria Requerimiento 10 – 12:

- ***testGeneraLogTransaccion:*** Verifica que por cada transacción que ingresa al sistema se inserten registros en la tabla de auditoría

LOG_TRANSACCION. En el caso de que no existan registros la prueba falla.

Prueba Unitaria Requerimiento 16:

- ***testUsuarioBloqueado:*** Verifica si el usuario que tiene 3 intentos de autenticación fallida se encuentre bloqueado. En el caso que el usuario no se encuentre bloqueado la prueba falla.
- ***testUsuarioNoBloqueado:*** Verifica si el usuario que no tiene 3 intentos de autenticación fallida no se encuentre bloqueado. En el caso que el usuario se encuentre bloqueado la prueba falla.

Las pruebas se ejecutan en el IDE y se comprueba que todas fallan como se muestra en la Figura 16.

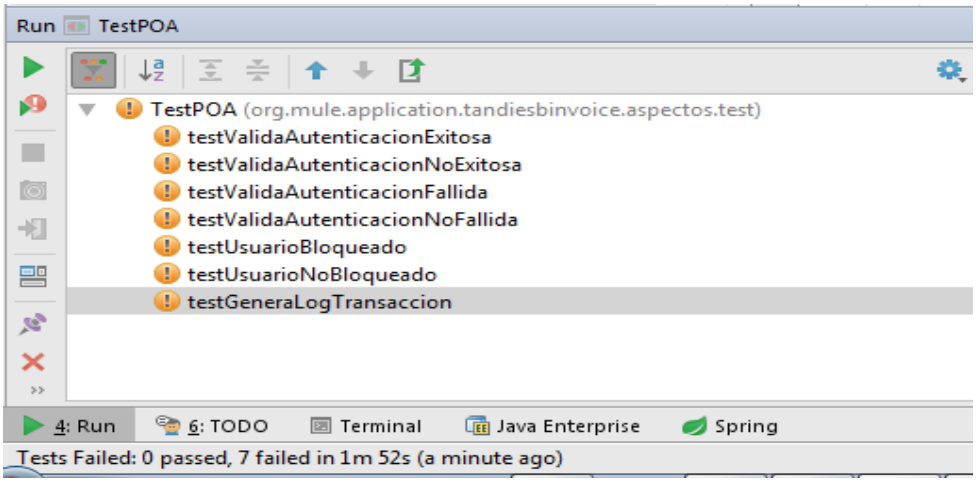
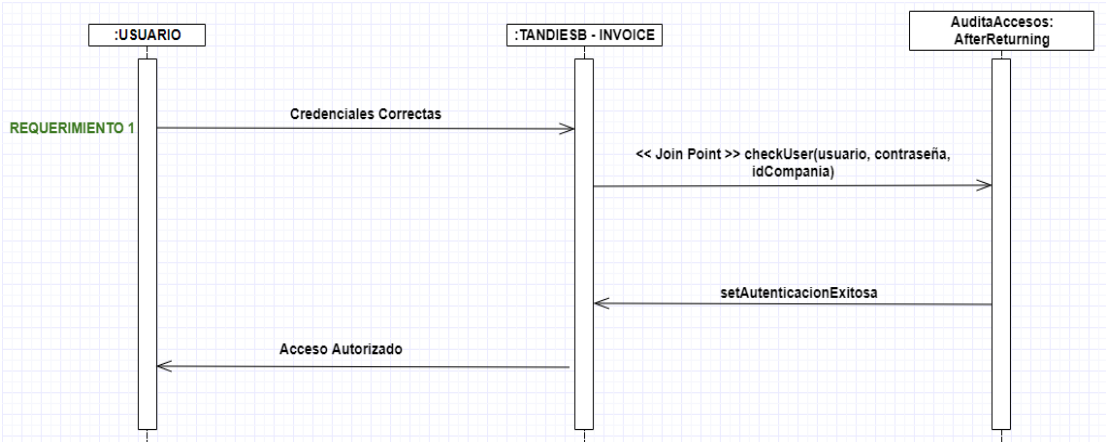


Figura 16 *Ejecución fallida de las pruebas de unidad*

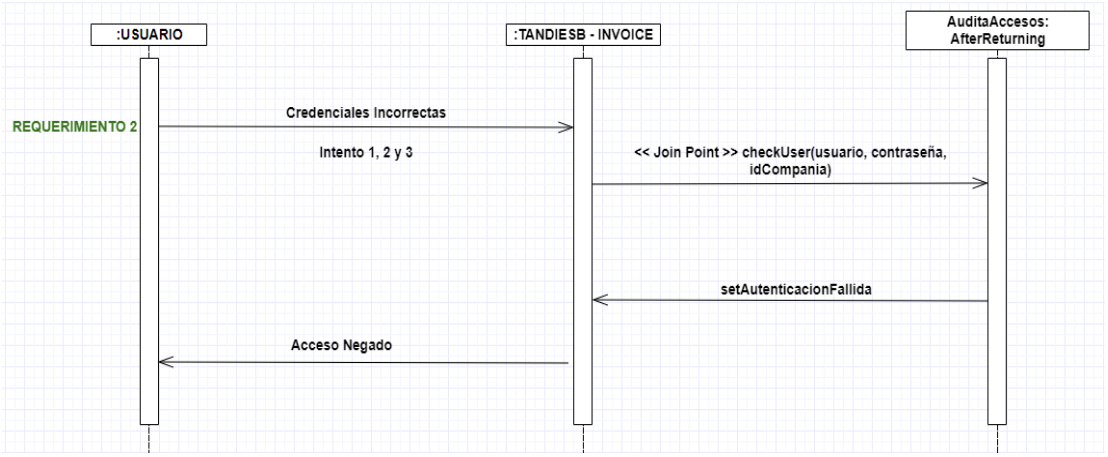
3.4.4. Implementación de Requerimientos de Seguridad

Los aspectos implementados se resumen en el siguiente diagrama de secuencias UML.

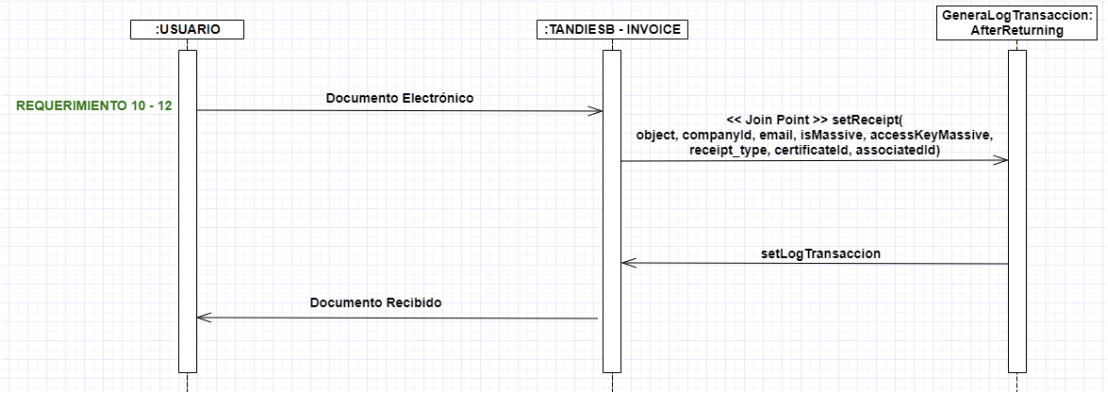
(a)


:USUARIO
:TANDIESB - INVOICE
AuditaAccesos:
AfterReturning
REQUERIMIENTO 1
Credenciales Correctas
<< Join Point >> checkUser(usuario, contraseña,
idCompania)
setAutenticacionExitosa
Acceso Autorizado

(b)


:USUARIO
:TANDIESB - INVOICE
AuditaAccesos:
AfterReturning
REQUERIMIENTO 2
Credenciales Incorrectas
Intento 1, 2 y 3
<< Join Point >> checkUser(usuario, contraseña,
idCompania)
setAutenticacionFallida
Acceso Negado

(c)


:USUARIO
:TANDIESB - INVOICE
GeneraLogTransaccion:
AfterReturning
REQUERIMIENTO 10 - 12
Documento Electrónico
<< Join Point >> setReceipt(
object, companyId, email, isMassive, accessKeyMassive,
receipt_type, certificateId, associatedId)
setLogTransaccion
Documento Recibido

(d)

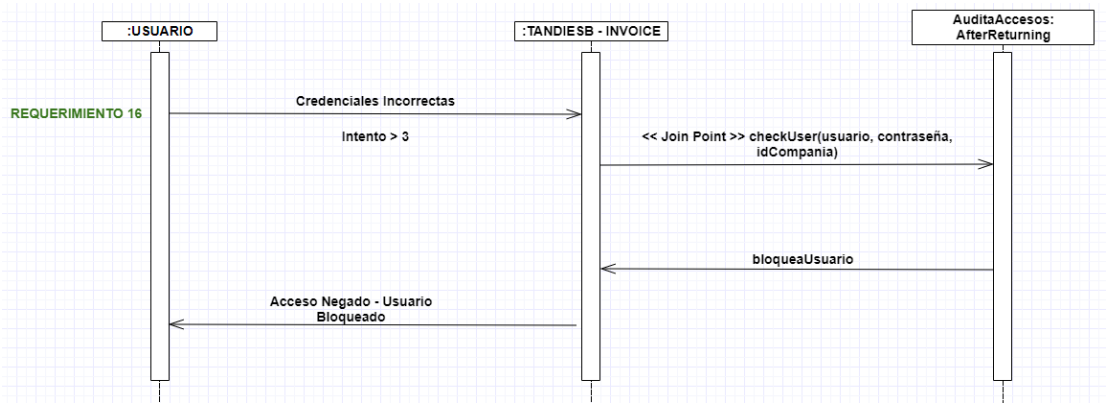


Figura 17 Modelo de Secuencias (a) Requerimiento 1, (b) Requerimiento 2, (c) Requerimiento 10 – 12, (d) Requerimiento 16

3.4.5. Pruebas Exitosas de los Requerimientos de Seguridad

Para validar que los requerimientos se encuentran resueltos correctamente se ejecutan las pruebas de unidad en el IDE y se comprueba que todas pasan como se muestra en la Figura 17.

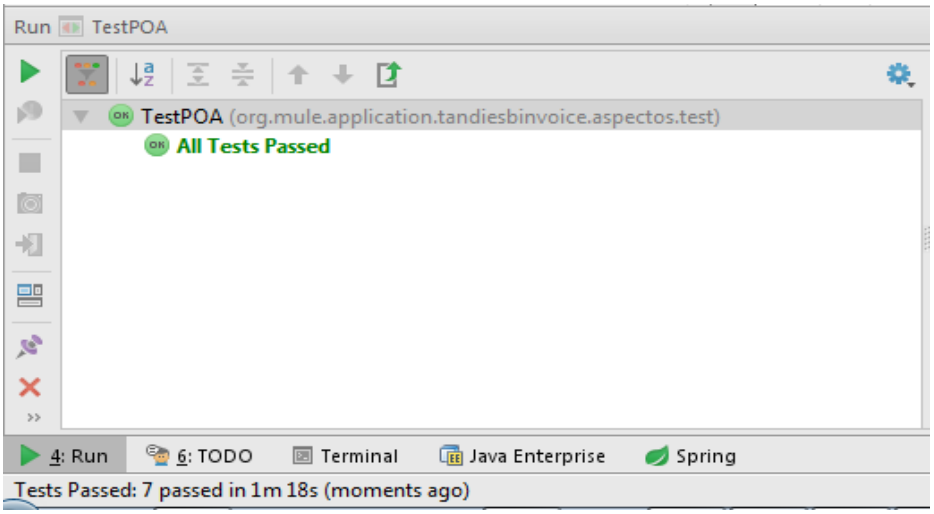


Figura 18 *Ejecución exitosa de las pruebas de unidad*

3.5. Pruebas de Rendimiento

3.5.1. Planificación de las pruebas

Para realizar las pruebas de rendimiento se tomará en cuenta el tiempo que se demore en ejecutar los flujos previo al envío de la transacción al Web Service de recepción de documentos del SRI, es decir, no se toma en cuenta todo el flujo de procesamiento (Figura 14) ya que para completarlo se depende de los tiempos de respuesta del SRI. Los escenarios planteados para las pruebas son los siguientes:

1. Envío de documento que ya se encuentra registrado como autorizado.
2. Envío de documento que no se encuentra registrado en el sistema.
3. Envío de documento con credenciales de usuario incorrectas.

Es necesario especificar que se enviará una transacción por cada escenario planteado con el componente de seguridad activado y el mismo número de transacciones con el componente de seguridad inactivo.

3.5.2. Descripción del ambiente de pruebas

Las pruebas se realizarán sobre un CPU que dispone la empresa para este propósito y cuenta con las siguientes características:

- SO: Windows 7 Ultimate de 64 bits. Service Pack 1.
- Procesador: Intel Core i5-3470T 2.90GHz.
- Memoria: 6 GB de RAM.
- Disco: 124 GB (44 GB disponibles)

3.5.3. Ejecución

Se envían las transacciones en ambiente de pruebas con el componente de seguridad inactivo y posteriormente con el componente se seguridad activado. Los tiempos de procesamiento de cada una de las transacciones se los toma del cliente de Web Service (SoapUI 5.2.1) que se usó para enviar los documentos al sistema TandiESB – Invoice; ya que el sistema responde al cliente antes de enviar el documento al servicio web de recepción de documentos del SRI tal como se lo definió en el punto 3.5.1 (Ver Figura 19)

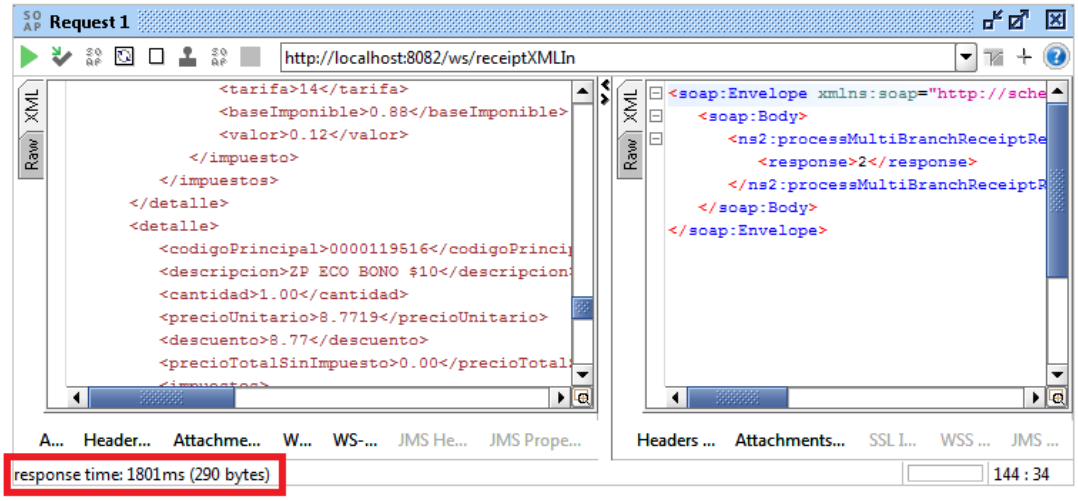


Figura 19 Ejemplo de tiempos de respuesta

Durante el envío de los documentos ejecutó un análisis de rendimiento de software (profilling) con la herramienta Java VisualVM, para evaluar el consumo de recursos del servidor.

3.5.4. Resultados

Una vez que se obtiene el tiempo de procesamiento de cada uno de los documentos se saca un promedio de los tiempos y se los compara dando como resultado que los documentos con el componente de seguridad desactivado se procesan 1.535 segundos más rápido que cuando el componente se encuentra activado, tal como se lo demuestra en la Figura 20.

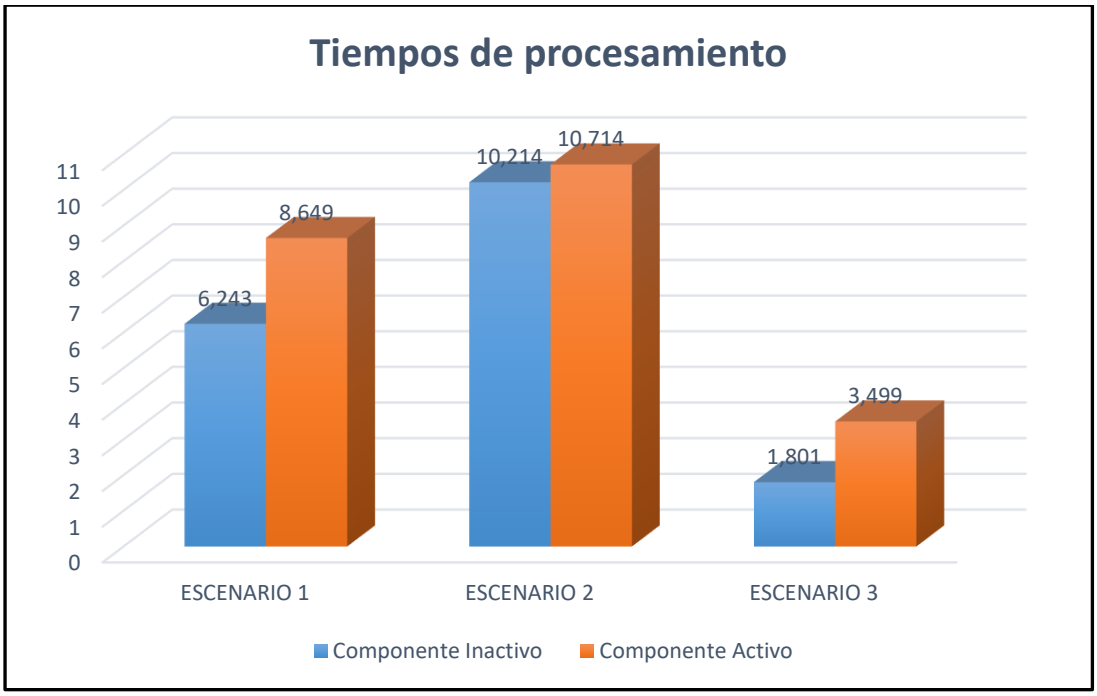


Figura 20 Resultado tiempos de procesamiento

En las Tablas 10 y 11 se muestran los resultados del análisis de rendimiento de software en cuanto a CPU y memoria se refieren; dependiendo del estado del componente de seguridad; comparando los resultados se determina que el consumo de los recursos del servidor aumenta cuando el componente se encuentra activo.

Tabla 10 CPU - Análisis de rendimiento

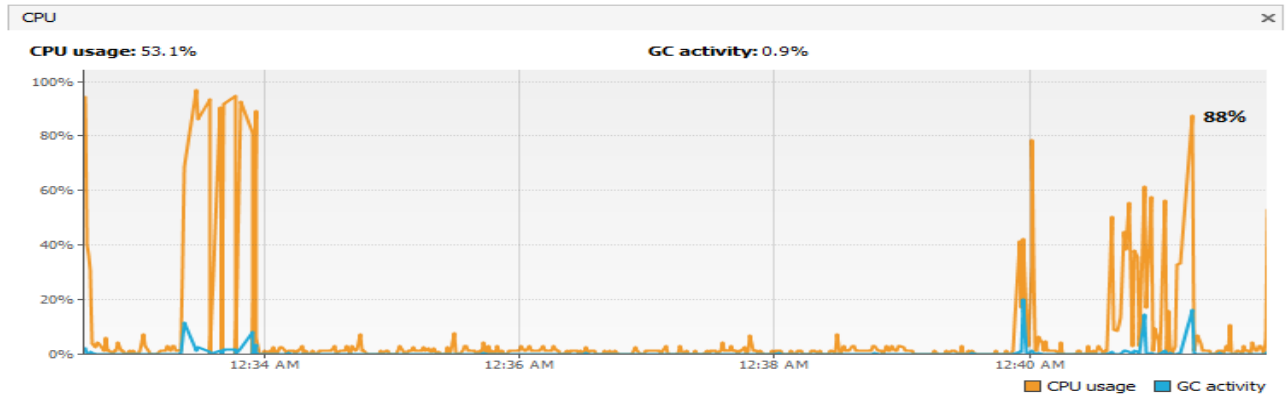
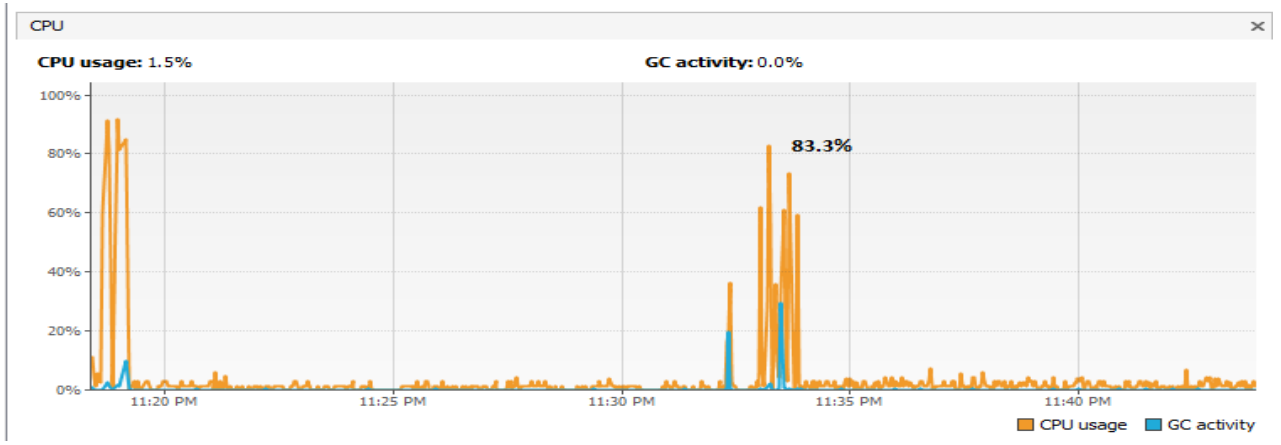
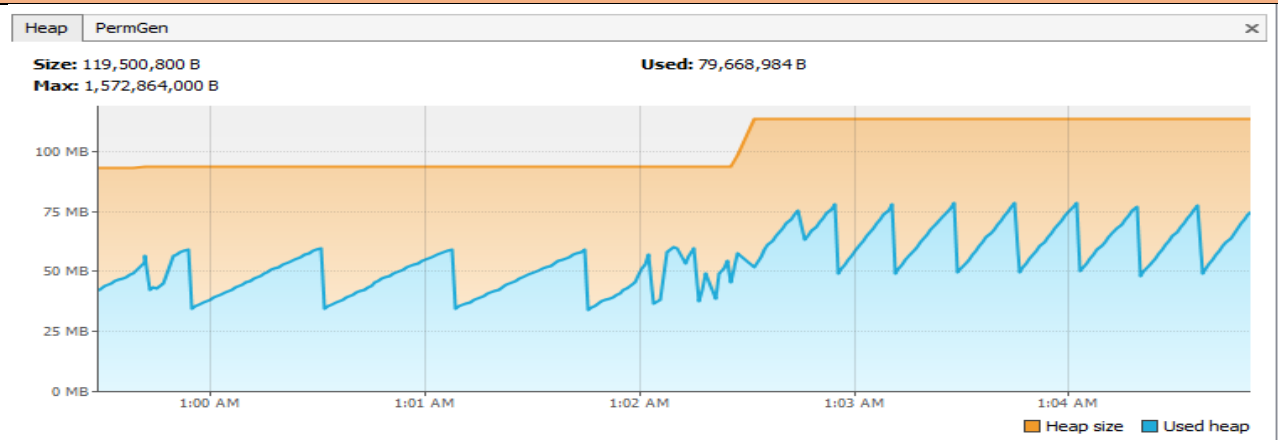
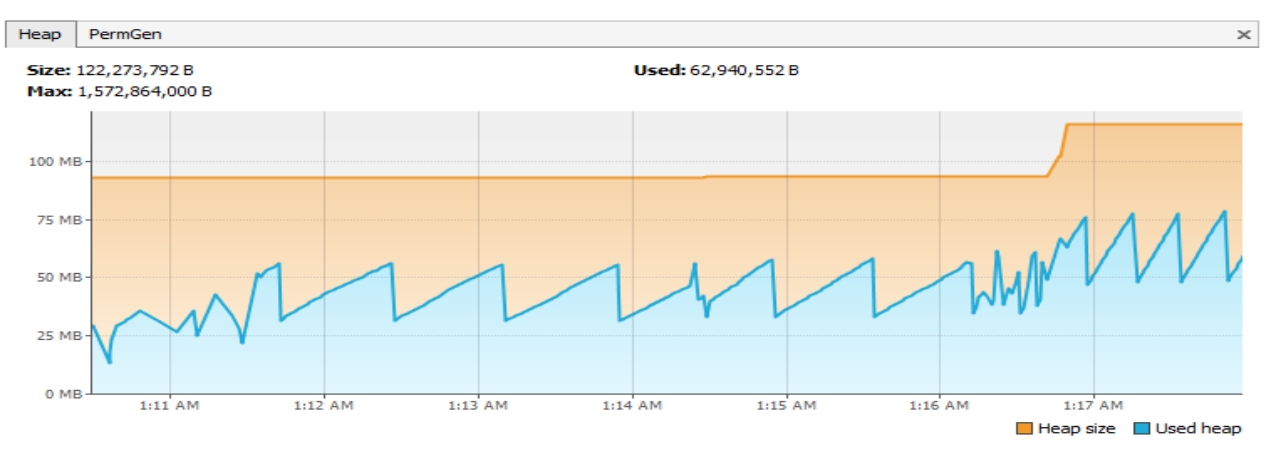
CPU	
Componente Activo	
Componente Inactivo	

Tabla 11 Memoria – Análisis de rendimiento

Memoria	
Componente Activo	
Componente Inactivo	

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

- Para el presente proyecto de manera general se concluye que es posible implementar componentes de seguridad lógica correspondientes a Autorización, Autenticación y Auditoría sobre un sistema existente, aplicando el paradigma de Programación Orientada a Aspectos (POA). Se verifica que este paradigma permite implementar la seguridad de manera separada ratificando lo que su teoría plantea, por ende, la implementación se ha realizado sin la necesidad de modificar el código del sistema que fue seleccionado como caso de estudio.
- En base a la revisión del estado del arte realizada en la sección 2.7 de este trabajo se ha determinado que el porcentaje de componentes de seguridad lógica que se puede implementar con POA recae en el 37.5% del total de componentes definidos en el Modelo Básico de Seguridad (Figura 3). Sin embargo, durante el desarrollo del presente trabajo se logró cubrir con POA el componente de Confidencialidad, (Tabla 5, requerimientos 10 y 12). Consecuentemente, el porcentaje real de componentes de seguridad lógica que se lograron implementar con POA asciende al 50%.
- Para demostrar la implementación de seguridad lógica con POA sobre un sistema existente se construyeron los siguientes componentes:

AuditaAccesos.java y GeneraLogTransaccion.java (el primero implementa los requerimientos 1,2 y 16 y el segundo los requerimientos 10 y 12 - Tabla 5). Los requerimientos que implementan Auditoría no modifican sus flujos funcionales; mientras que, el requerimiento que implementa Autenticación agrega un flujo funcional adicional (bloqueo de usuarios). Consecuentemente se comprueba que luego de las implementaciones realizadas con POA se agrega funcionalidad al sistema sin modificar la original, por ende, el paradigma no es invasivo.

- Al realizar las pruebas de rendimiento descritas en la sección 3.5 se verifica que una vez que se implementaron los componentes de seguridad el rendimiento del caso de estudio seleccionado ha disminuido, ya que los tiempos de ejecución de los procesos que involucran POA han aumentado en promedio 1.535 segundos en un ambiente de pruebas. Por lo tanto, si se opta por agregar seguridad lógica con un módulo totalmente independiente, lo cual se ha demostrado que se puede realizar con la adopción de POA, se debe asumir un mínimo aumento en los tiempos de respuesta del sistema.
- Las bases que se deberían tener para iniciar con un análisis de seguridad son las Políticas de Seguridad, mismas con las que no cuenta la empresa dueña del caso de estudio de este trabajo. Por tal motivo se concluye que bajo este escenario se debe realizar un análisis que

permita definir un punto de partida, lo cual se hizo en este trabajo y se obtuvo como resultado la plantilla presente en la Tabla 1.

- Finalmente se pudo comprobar que al implementar los componentes de seguridad lógica el sistema cubrió consideraciones adicionales de este tipo de seguridad con las que no contaba, sin embargo, esto no garantiza que luego de esta implementación el sistema sea completamente seguro.

4.2. Recomendaciones

- Realizar un trabajo en el cual se pueda realizar la implementación de componentes de seguridad lógica aplicando POO y POA, con el objetivo de realizar un análisis comparativo de rendimiento y complejidad de implementación entre estos dos paradigmas.
- Si es de interés realizar un análisis para implementar componentes de seguridad lógica con POA para cualquier otro caso de estudio que no cuenta con políticas de seguridad, se recomienda utilizar a modo de plantilla la Tabla 1 del presente trabajo. Ésta servirá de ayuda para tener una idea del estado actual de la seguridad lógica del caso de estudio y las posibilidades de aplicar POA para implementar dicha seguridad.
- Realizar el análisis de rendimiento de la implementación de los componentes de seguridad lógica en un ambiente de producción, ya que para este trabajo no se pudo tener acceso a dicho ambiente, el cual

cuenta con mejores recursos que el ambiente de pruebas en el que se realizó este análisis para el presente trabajo.

BIBLIOGRAFÍA

[1] Antonia Ma Reina Quintero. (2000, Dic). [Online]. Visión general de la Programación Orientada a Aspectos. Disponible en: https://www.lsi.us.es/docs/informes/aopv3.pdf

[2] Rohit Sethi. (2010, Nov). Aspect-Oriented Programming and Security. [Online]. Disponible en: http://www.symantec.com/connect/articles/aspect-oriented-programming-and-security

[3] F. Asteasuain, B. Contreras., "Programación Orientada a Aspectos. Análisis del Paradigma" Licenciatura, Depto. de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur, Bahía Blanca, Buenos Aires, Argentina. 2002.

[4] Farzana Shafique, Khalid Mahmood. 2010. Model Development as Research Tool: An Example of PAK-NISEA. [Online]. Disponible en: http://www.webpages.uidaho.edu/~mbolin/shafique-mahmood.htm.

[5] Herramientas Tt. 2002. Exposición Programación Orientada a Aspectos. [Online]. Disponible en: http://ldc.usb.ve/~yudith/docencia/Telematica/TemasHerramientasInfor/Exposiciones/ProgramacionOrientadaAspectosHans.pdf

[6] Oberon. 2002. [Online]. Disponible en: http://oberon2005.oberoncore.ru/paper/np2002.pdf

[7] ECOOP. Origen POA kiczales. 1997.

[8] ECURED. 2017. Programación Orientada a Aspectos. [Online]. Disponible en: https://www.ecured.cu/Programaci%C3%B3n_orientada_a_aspectos#Fundamentos_de_la_Programaci.C3.B3n_Orientada_a_Aspectos. 2017

[9] Nicolás Martín Paez., “Utilización de programación orientada a aspectos en aplicaciones enterprise” Ingeniería, Facultad de Ingeniería, Universidad de Buenos Aires, Buenos Aires, 2007.

[10] Erb. M. 2017. Definición de Seguridad Informática. [Online]. Disponible en: https://protejete.wordpress.com/gdr_principal/definicion_si/

[11] Sergio Ochoa Ovalle, Omar Cervantes Sánchez. 2012. Seguridad Informática. [Online]. Disponible en: http://www.eumed.net/rev/cccss/21/oocs.html#_ftn3 tipos de seguridad

[12] J. Jaworski, P. J. Perrone, Seguridad en Java, edición especial. Madrid, España: Prentice Hall, 2001, ch.1

[13] Seguridad Lógica. [Online]. Disponible en: https://seguridadinformaticasmr.wikispaces.com/TEMA+3+-+SEGURIDAD+L%C3%93GICA

[14] Feasibility Analysis for a Software Project. [Online]. Disponible en: ftp://cosm.sfasu.edu/cs/dcook/CSC426/Feasibility%20and%20CONOPS.ppt

[15] Mukund. 2017. Why a Feasibility Study is Important in Project Management. [Online]. Disponible en: https://www.simplilearn.com/feasibility-study-article

ANEXOS

Anexo 1: Encuesta

Introducción

Buenos días (tardes):

Nosotros, Valeria Saltos y David Villacrés estudiantes de la Escuela Politécnica Nacional solicitamos su valiosa ayuda para contestar el siguiente cuestionario que no llevará mucho tiempo; dada su posición dentro de la empresa usted es importante por su conocimiento sobre los sistemas de la compañía.

Sus respuestas serán de mucha ayuda para evaluar la seguridad de los sistemas que la Corporación Tecnología De La Información Solutandi Cia. Ltda. comercializa. Es muy importante que se conteste con la mayor sinceridad posible. No hay respuestas correctas ni incorrectas.

Lea las instrucciones cuidadosamente.

Instrucciones

Emplee un lápiz o un bolígrafo de tinta azul para rellenar el cuestionario.

Marque con claridad la opción elegida con una cruz o tache, o bien, utilice una “paloma” (símbolo de verificación). Marque así:

X √

Si no puede contestar una pregunta o si la pregunta no tiene sentido para usted, por favor pregúntele a la persona que le entregó este cuestionario.

De antemano: ¡MUCHAS GRACIAS POR SU COLABORACIÓN!

Cuestionario

Nombre: _____ **Cargo**: _____

Años de experiencia dentro de la empresa: _____

Sección I

Por favor marque los sistemas que cumplen con lo planteado en las siguientes preguntas. Las preguntas están clasificadas dependiendo de la propiedad de seguridad a la que pertenecen:

AUDITORÍA				
PREGUNTAS	**SISTEMAS**			
	OpenERP	**Tandi - Payroll**	**Tandi - Invoice**	**Tandi - Insurance**
1. ¿Son auditados los intentos fallidos de acceso al sistema?				
2. ¿Se auditan los accesos exitosos al sistema?				
3. ¿Se auditan todos los intentos de comunicación con otros sistemas?				
4. ¿Se audita la revocación de perfiles de acceso?				
5. ¿Se audita la asignación de perfiles de acceso?				
6. ¿El acceso a información sensible es auditado?				
7. ¿Se audita la creación de usuarios?				
8. ¿Se audita la creación de perfiles de acceso?				
9. ¿Se audita los accesos a tablas de logs?				
10. ¿La información sensible es almacenada en tablas de logs de manera cifrada?				

11. ¿Existe una interface que permita hacer consultas sobre los logs?				
12. ¿Las tablas de logs cuentan con al menos la siguiente información? Fecha y hora local, dirección IP del usuario que genero el log y opción del sistema que se trató de ejecutar.				
AUTENTICACIÓN				
PREGUNTAS	**SISTEMAS**			
	OpenERP	**Tandi - Payroll**	**Tandi - Invoice**	**Tandi - Insurance**
13. ¿El proceso de autenticación de usuarios usa tablas propias del esquema del sistema?				
14. ¿Las contraseñas de los usuarios que acceden al sistema se someten a un proceso de hashing (Algoritmo de Encriptación)?				
15. ¿Existe un proceso de verificación de usuarios?				
16. ¿Por intentos fallidos de acceso al sistema el usuario es bloqueado?				
17. ¿Se cuenta con un proceso de desbloqueo de usuarios?				
18. ¿Existe un factor de autenticación para procesos que manejan información sensible?				

19. ¿Se controla el uso histórico de contraseñas?				
20. ¿Se controla que la contraseña del usuario no contenga porciones del nombre del usuario?				
21. ¿Para la creación de contraseñas se controla que tenga como un mínimo de 8 caracteres y un máximo de 20 caracteres?				
22. ¿Se maneja un estándar de creación de contraseñas? Por ejemplo: La contraseña debe contener al menos una letra mayúscula, números y un carácter especial.				
23. ¿Existe un proceso de recuperación de contraseñas cuando el usuario olvido la misma?				
24. ¿La contraseña tiene un tiempo de caducidad?				
25. ¿Existe un segundo factor de autenticación?				
26. ¿El segundo factor de autenticación tiene una duración temporal (minutos)?				
27. ¿Se controla que la baja de un usuario del sistema sea a nivel lógico y no físico?				

28. ¿La sesión de usuario se caduca si existe inactividad?				
29. ¿Existe control de sesión en todos los módulos del sistema?				
AUTORIZACIÓN				
PREGUNTAS	**SISTEMAS**			
	OpenERP	**Tandi - Payroll**	**Tandi - Invoice**	**Tandi - Insurance**
30. ¿Se asignan perfiles de acceso a los usuarios?				
31. ¿Los perfiles de acceso limitan la ejecución de procesos?				
32. ¿Se controla que el usuario solamente pueda tener asignado un solo perfil de acceso?				

Sección II

33. Indique los tipos de usuarios que se pueden autenticar en el sistema:

TIPOS DE USARIOS	**SISTEMAS**			
	OpenERP	**Tandi - Payroll**	**Tandi - Invoice**	**Tandi - Insurance**
Personas				
Sistemas				

34. ¿Cuántos subsistemas tiene el sistema?

SISTEMAS	**NÚMERO DE SUBSISTEMAS**
OpenERP	
Tandi - Payroll	

Tandi - Invoice	
Tandi - Insurance	

35. Indique el lenguaje de programación utilizado para desarrollar la capa del negocio del sistema.

SISTEMAS	LENGUAJE DE PROGRAMACIÓN
OpenERP	
Tandi - Payroll	
Tandi - Invoice	
Tandi - Insurance	

Anexo 2: Tabulación de la Encuesta

Para la tabulación de las preguntas de la sección I de la encuesta se establece que será representado con el número uno (1) si el sistema evaluado cumple con lo preguntado.

PREGUNTAS	ENCUESTADO 1				ENCUESTADO 2			
	OpenERP	Tandi - Payroll	Tandi - Invoice	Tandi - Insurance	OpenERP	Tandi - Payroll	Tandi - Invoice	Tandi - Insurance
1. ¿Son auditados los intentos fallidos de acceso al sistema?	1				1			
2. ¿Se auditan los accesos exitosos al sistema?	1		1	1	1			1
3. ¿Se auditan todos los intentos de comunicación con otros sistemas?			1				1	
4. ¿Se audita la revocación de perfiles de acceso?								1
5. ¿Se audita la asignación de perfiles de acceso?								1
6. ¿El acceso a información sensible es auditado?								1
7. ¿Se audita la creación de usuarios?	1	1	1	1	1	1	1	1
8. ¿Se audita la creación de perfiles de acceso?	1	1		1	1	1	1	1
9. ¿Se audita los accesos a tablas de logs?								
10. ¿La información sensible es almacenada en tablas de logs de manera cifrada?			1					
11. ¿Existe una interface que permita hacer consultas sobre los logs?				1				
12. ¿Las tablas de logs cuentan con al menos la siguiente información? Fecha y hora local, dirección IP del usuario que genero el log y opción del sistema que se trató de ejecutar.								1
13. ¿El proceso de autenticación de usuarios usa tablas propias del esquema del sistema?	1	1		1	1	1	1	1
14. ¿Las contraseñas de los usuarios que acceden al sistema se someten a un proceso de hashing (Algoritmo de Encriptación)?		1	1	1		1	1	1
15. ¿Existe un proceso de verificación de usuarios?	1	1	1	1	1	1	1	1
16. ¿Por intentos fallidos de acceso al sistema el usuario es bloqueado?	1			1	1			

PREGUNTAS	ENCUESTADO 1				ENCUESTADO 2			
	OpenERP	Tandi - Payroll	Tandi - Invoice	Tandi - Insurance	OpenERP	Tandi - Payroll	Tandi - Invoice	Tandi - Insurance
17. ¿Se cuenta con un proceso de desbloqueo de usuarios?	1			1	1			
18. ¿Existe un factor de autenticación para procesos que manejan información sensible?								
19. ¿Se controla el uso histórico de contraseñas?				1				
20. ¿Se controla que la contraseña del usuario no contenga porciones del nombre del usuario?								
21. ¿Para la creación de contraseñas se controla que tenga como un mínimo de 8 caracteres y un máximo de 20 caracteres?		1				1		
22. ¿Se maneja un estándar de creación de contraseñas? Por ejemplo: La contraseña debe contener al menos una letra mayúscula, números y un carácter especial.								
23. ¿Existe un proceso de recuperación de contraseñas cuando el usuario olvido la misma?	1		1	1	1			1
24. ¿La contraseña tiene un tiempo de caducidad?		1		1		1		1
25. ¿Existe un segundo factor de autenticación?								
26. ¿El segundo factor de autenticación tiene una duración temporal (minutos)?								
27. ¿Se controla que la baja de un usuario del sistema sea a nivel lógico y no físico?	1	1		1	1	1	1	1
28. ¿La sesión de usuario se caduca si existe inactividad?	1	1	1	1	1	1	1	1
29. ¿Existe control de sesión en todos los módulos del sistema?	1	1	1	1	1	1	1	1
30. ¿Se asignan perfiles de acceso a los usuarios?	1	1	1	1	1	1	1	1
31. ¿Los perfiles de acceso limitan la ejecución de procesos?	1	1	1	1	1	1	1	1
32. ¿Se controla que el usuario solamente pueda tener asignado un solo perfil de acceso?		1	1			1		
TOTAL	14	13	12	17	14	13	11	17

PREGUNTAS	ENCUESTADO 3				ENCUESTADO 4			
	OpenERP	Tandi - Payroll	Tandi - Invoice	Tandi - Insurance	OpenERP	Tandi - Payroll	Tandi - Invoice	Tandi - Insurance
1. ¿Son auditados los intentos fallidos de acceso al sistema?	1				1			
2. ¿Se auditan los accesos exitosos al sistema?	1				1			1

PREGUNTAS	ENCUESTADO 3				ENCUESTADO 4			
	OpenERP	Tandi - Payroll	Tandi - Invoice	Tandi - Insurance	OpenERP	Tandi - Payroll	Tandi - Invoice	Tandi - Insurance
3. ¿Se auditan todos los intentos de comunicación con otros sistemas?							1	
4. ¿Se audita la revocación de perfiles de acceso?								
5. ¿Se audita la asignación de perfiles de acceso?								
6. ¿El acceso a información sensible es auditado?				1				
7. ¿Se audita la creación de usuarios?	1	1	1			1	1	1
8. ¿Se audita la creación de perfiles de acceso?	1	1	1			1	1	1
9. ¿Se audita los accesos a tablas de logs?								
10. ¿La información sensible es almacenada en tablas de logs de manera cifrada?								
11. ¿Existe una interface que permita hacer consultas sobre los logs?				1				1
12. ¿Las tablas de logs cuentan con al menos la siguiente información? Fecha y hora local, dirección IP del usuario que genero el log y opción del sistema que se trató de ejecutar.					1			
13. ¿El proceso de autenticación de usuarios usa tablas propias del esquema del sistema?	1	1	1	1		1	1	1
14. ¿Las contraseñas de los usuarios que acceden al sistema se someten a un proceso de hashing (Algoritmo de Encriptación)?		1	1	1		1	1	1
15. ¿Existe un proceso de verificación de usuarios?	1	1	1	1	1	1	1	1
16. ¿Por intentos fallidos de acceso al sistema el usuario es bloqueado?	1			1	1			1
17. ¿Se cuenta con un proceso de desbloqueo de usuarios?	1			1	1		1	1
18. ¿Existe un factor de autenticación para procesos que manejan información sensible?				1				
19. ¿Se controla el uso histórico de contraseñas?				1				1
20. ¿Se controla que la contraseña del usuario no contenga porciones del nombre del usuario?								
21. ¿Para la creación de contraseñas se controla que tenga como un mínimo de 8 caracteres y un máximo de 20 caracteres?		1		1		1		
22. ¿Se maneja un estándar de creación de contraseñas? Por ejemplo: La contraseña debe contener al menos una letra mayúscula, números y un carácter especial.				1				

PREGUNTAS	ENCUESTADO 3				ENCUESTADO 4			
	OpenERP	Tandi - Payroll	Tandi - Invoice	Tandi - Insurance	OpenERP	Tandi - Payroll	Tandi - Invoice	Tandi - Insurance
23. ¿Existe un proceso de recuperación de contraseñas cuando el usuario olvido la misma?	1			1	1		1	1
24. ¿La contraseña tiene un tiempo de caducidad?		1		1		1		1
25. ¿Existe un segundo factor de autenticación?								
26. ¿El segundo factor de autenticación tiene una duración temporal (minutos)?								
27. ¿Se controla que la baja de un usuario del sistema sea a nivel lógico y no físico?	1	1	1	1	1	1	1	1
28. ¿La sesión de usuario se caduca si existe inactividad?	1	1	1	1	1	1	1	1
29. ¿Existe control de sesión en todos los módulos del sistema?	1	1	1	1	1	1	1	1
30. ¿Se asignan perfiles de acceso a los usuarios?	1	1	1	1	1	1	1	1
31. ¿Los perfiles de acceso limitan la ejecución de procesos?	1	1	1	1	1	1	1	1
32. ¿Se controla que el usuario solamente pueda tener asignado un solo perfil de acceso?		1				1		
TOTAL	14	13	10	18	12	13	13	17

PREGUNTAS	ENCUESTADO 5				ENCUESTADO 6			
	OpenERP	Tandi - Payroll	Tandi - Invoice	Tandi - Insurance	OpenERP	Tandi - Payroll	Tandi - Invoice	Tandi - Insurance
1. ¿Son auditados los intentos fallidos de acceso al sistema?	1				1			
2. ¿Se auditan los accesos exitosos al sistema?	1				1			1
3. ¿Se auditan todos los intentos de comunicación con otros sistemas?							1	
4. ¿Se audita la revocación de perfiles de acceso?								
5. ¿Se audita la asignación de perfiles de acceso?								
6. ¿El acceso a información sensible es auditado?				1				
7. ¿Se audita la creación de usuarios?	1	1	1			1	1	1
8. ¿Se audita la creación de perfiles de acceso?	1	1	1			1	1	1
9. ¿Se audita los accesos a tablas de logs?								
10. ¿La información sensible es almacenada en tablas de logs de manera cifrada?								
11. ¿Existe una interface que permita hacer consultas sobre los logs?				1				1

PREGUNTAS	ENCUESTADO 5				ENCUESTADO 6			
	OpenERP	Tandi - Payroll	Tandi - Invoice	Tandi - Insurance	OpenERP	Tandi - Payroll	Tandi - Invoice	Tandi - Insurance
12. ¿Las tablas de logs cuentan con al menos la siguiente información? Fecha y hora local, dirección IP del usuario que genero el log y opción del sistema que se trató de ejecutar.					1			
13. ¿El proceso de autenticación de usuarios usa tablas propias del esquema del sistema?	1	1	1	1		1	1	1
14. ¿Las contraseñas de los usuarios que acceden al sistema se someten a un proceso de hashing (Algoritmo de Encriptación)?		1	1	1		1	1	1
15. ¿Existe un proceso de verificación de usuarios?	1	1	1	1	1	1	1	1
16. ¿Por intentos fallidos de acceso al sistema el usuario es bloqueado?	1			1	1			1
17. ¿Se cuenta con un proceso de desbloqueo de usuarios?	1			1	1		1	1
18. ¿Existe un factor de autenticación para procesos que manejan información sensible?				1				
19. ¿Se controla el uso histórico de contraseñas?				1				1
20. ¿Se controla que la contraseña del usuario no contenga porciones del nombre del usuario?								
21. ¿Para la creación de contraseñas se controla que tenga como un mínimo de 8 caracteres y un máximo de 20 caracteres?		1		1		1		
22. ¿Se maneja un estándar de creación de contraseñas? Por ejemplo: La contraseña debe contener al menos una letra mayúscula, números y un carácter especial.				1				
23. ¿Existe un proceso de recuperación de contraseñas cuando el usuario olvido la misma?	1			1	1		1	1
24. ¿La contraseña tiene un tiempo de caducidad?		1		1		1		1
25. ¿Existe un segundo factor de autenticación?								
26. ¿El segundo factor de autenticación tiene una duración temporal (minutos)?								
27. ¿Se controla que la baja de un usuario del sistema sea a nivel lógico y no físico?	1	1	1	1	1	1	1	1
28. ¿La sesión de usuario se caduca si existe inactividad?	1	1	1	1	1	1	1	1
29. ¿Existe control de sesión en todos los módulos del sistema?	1	1	1	1	1	1	1	1
30. ¿Se asignan perfiles de acceso a los usuarios?	1	1	1	1	1	1	1	1

PREGUNTAS	ENCUESTADO 5				ENCUESTADO 6			
	OpenERP	Tandi - Payroll	Tandi - Invoice	Tandi - Insurance	OpenERP	Tandi - Payroll	Tandi - Invoice	Tandi - Insurance
31. ¿Los perfiles de acceso limitan la ejecución de procesos?	1	1	1	1	1	1	1	1
32. ¿Se controla que el usuario solamente pueda tener asignado un solo perfil de acceso?		1				1		
TOTAL	14	13	10	18	12	13	13	17

Una vez tabulados los resultados se saca un promedio de los totales por sistema y se los ordena de menor a mayor, dando como resultado que el sistema que menos cumple con las propiedades de seguridad evaluadas es el Tandi – Invoice.

Sistema	Promedio
Tandi - Invoice	11.17
Tandi – Payroll	12.33
OpenERP	13.67
Tandi - Insurance	17.17

Anexo 3: Evaluación Caso de Estudio

Propiedad de Seguridad	Id.	Pregunta	Parámetro POA a cumplirse		Cumplimiento de Condición	Funcionalidad Inexistente	Total	Se incluye como Requerimiento
Auditoría	**1**	¿Son auditados los intentos fallidos de acceso al sistema?	El sistema permite la autenticación de usuarios.	1	1	1	1	**SI**
			Existe un método que se ejecuta para verificar si las credenciales de usuario ingresadas son correctas.	1				
Auditoría	**2**	¿Se auditan los accesos exitosos al sistema?	El sistema permite la autenticación de usuarios.	1	1	1	1	**SI**
			Existe un método que se ejecuta para verificar si las credenciales de usuario ingresadas son correctas.	1				
Auditoría	**3**	¿Se auditan todos los intentos de comunicación con otros sistemas?	El sistema expone Web Services.	1	1	0	0	**NO**
			Existe un método que se ejecuta cada vez que los Web Services son consumidos.	1				
Auditoría	**4**	¿Se audita la revocación de perfiles de acceso?	Existe un módulo de administración de usuarios en el sistema.	1	0	1	0	**NO**
			Es posible revocar perfiles de acceso.	0				
			Existe un método que se ejecuta al momento de revocar el perfil de acceso.	0				
Auditoría	**5**	¿Se audita la asignación de perfiles de acceso?	Existe un módulo de administración de usuarios en el sistema.	1	0	1	0	**NO**
			Es posible asignar perfiles de acceso.	0				
			Existe un método que se ejecuta al momento de asignar el perfil de acceso.	0				
Auditoría	**6**	¿El acceso a información sensible es auditado?	El sistema procesa o maneja información privada de personas y/o empresas, por ejemplo, ciertos datos personales y bancarios, contraseñas de correo electrónico, domicilio.	1	0	1	0	**NO**
			Existe una interface que permita visualizar la información sensible.	0				
			Existe un método que se ejecuta al momento de acceder a la información sensible.	1				
Auditoría	**7**	¿Se audita la creación de usuarios?	Existe un módulo de administración de usuarios en el sistema.	0	0	0	0	**NO**
			Existe un método que se ejecuta al momento de crear un usuario.	0				
Auditoría	**8**	¿Se audita la creación de perfiles de acceso?	Existe un módulo de administración de usuarios en el sistema.	0	0	0	0	**NO**
			Es posible crear perfiles de acceso.	0				
			Existe un método que se ejecuta al momento de crear el perfil de acceso.	0				

Propiedad de Seguridad	Id.	Pregunta	Parámetro POA a cumplirse		Cumplimiento de Condición	Funcionalidad Inexistente	Total	Se incluye como Requerimiento
Auditoría	**9**	¿Se audita los accesos a tablas de logs?	Existe una interface para poder consultar el registro de logs.	0	0	1	0	**NO**
			Existe un método que se ejecuta al momento de consultar el registro de logs.	0				
Auditoría	**10**	¿La información sensible es almacenada en tablas de logs de manera cifrada?	El sistema procesa o maneja información privada de personas y/o empresas, por ejemplo, ciertos datos personales y bancarios, contraseñas de correo electrónico, domicilio.	1	1	1	1	**SI**
			Para almacenar la información sensible el sistema lo hace por medio de métodos.	1				
Auditoría	**12**	¿Las tablas de logs cuentan con al menos la siguiente información? Fecha y hora local, dirección IP del usuario que genero el log y opción del sistema que se trató de ejecutar.	Las herramientas con las cuales está desarrollado el sistema permiten capturar la fecha y hora local.	1	1	1	1	**SI**
			Las herramientas con las cuales está desarrollado el sistema permiten capturar la IP desde la cual se lo invoca.	1				
			Las herramientas con las cuales está desarrollado el sistema permiten capturar la opción o proceso del sistema que se trató de ejecutar.	1				
Autenticación	**14**	¿Las contraseñas de los usuarios que acceden al sistema se someten a un proceso de hashing (Algoritmo de Encriptación)?	Existe un método que se ejecuta al momento de crear las credenciales de usuario de acceso al sistema.	0	0	0	0	**NO**
			Existe un método que se ejecuta cuando un usuario desea autenticarse en el sistema.	1				
Autenticación	**15**	¿Existe un proceso de verificación de usuarios?	Existe un método que se ejecuta cuando un usuario desea autenticarse en el sistema.	1	1	0	0	**NO**
Autenticación	**16**	¿Por intentos fallidos de acceso al sistema el usuario es bloqueado?	Existe un método que se ejecuta cuando un usuario desea autenticarse en el sistema.	1	1	1	1	**SI**
Autenticación	**17**	¿Se cuenta con un proceso de desbloqueo de usuarios?	Existe un método que se ejecuta cuando un usuario desea autenticarse en el sistema.	0	0	1	0	**NO**

Propiedad de Seguridad	Id.	Pregunta	Parámetro POA a cumplirse		Cumplimiento de Condición	Funcionalidad Inexistente	Total	Se incluye como Requerimiento
Autenticación	**18**	¿Existe un factor de autenticación para procesos que manejan información sensible?	El sistema procesa información sensible.	1	0	1	0	**NO**
			El procesamiento de información sensible se lo hace por medio de una interface.	0				
			El procesamiento de información sensible se lo hace por medio de métodos propios del sistema.	1				
Autenticación	**19**	¿Se controla el uso histórico de contraseñas?	Existe un módulo de administración de usuarios en el sistema.	0	0	1	0	**NO**
			Existe un método que se ejecuta al momento de crear las credenciales de usuario de acceso al sistema.	0				
Autenticación	**20**	¿Se controla que la contraseña del usuario no contenga porciones del nombre del usuario?	Existe un módulo de administración de usuarios en el sistema.	0	0	1	0	**NO**
			Existe un método que se ejecuta al momento de crear las credenciales de usuario de acceso al sistema.	0				
Autenticación	**21**	¿Para la creación de contraseñas se controla que tenga como un mínimo de 8 caracteres y un máximo de 20 caracteres?	Existe un módulo de administración de usuarios en el sistema.	0	0	1	0	**NO**
			Existe un método que se ejecuta al momento de crear las credenciales de usuario de acceso al sistema.	0				
Autenticación	**22**	¿Se maneja un estándar de creación de contraseñas? Por ejemplo: La contraseña debe contener al menos una letra mayúscula, números y un carácter especial.	Existe un módulo de administración de usuarios en el sistema.	0	0	1	0	**NO**
			Existe un método que se ejecuta al momento de crear las credenciales de usuario de acceso al sistema.	0				
Autenticación	**23**	¿Existe un proceso de recuperación de contraseñas cuando el usuario olvido la misma?	Existe un módulo de administración de usuarios en el sistema.	0	0	1	0	**NO**
			Existe un método que se ejecuta al momento de crear las credenciales de usuario de acceso al sistema.	0				
Autenticación	**24**	¿La contraseña tiene un tiempo de caducidad?	Existe un módulo de administración de usuarios en el sistema.	0	0	1	0	**NO**
			Existe un método que se ejecuta al momento de crear las credenciales de usuario de acceso al sistema.	0				
Autenticación	**25**	¿Existe un segundo factor de autenticación?	Existe un módulo de administración de usuarios en el sistema.	0	0	1	0	**NO**
			Existe un método que se ejecuta al momento de crear las credenciales de usuario de acceso al sistema.	0				
Autenticación	**26**		Existe un módulo de administración de usuarios en el sistema.	0	0	1	0	**NO**

Propiedad de Seguridad	Id.	Pregunta	Parámetro POA a cumplirse		Cumplimiento de Condición	Funcionalidad Inexistente	Total	Se incluye como Requerimiento
		¿El segundo factor de autenticación tiene una duración temporal (minutos)?	Existe un método que se ejecuta al momento de crear las credenciales de usuario de acceso al sistema.	0				
Autenticación	**27**	¿Se controla que la baja de un usuario del sistema sea a nivel lógico y no físico?	Existe un módulo de administración de usuarios en el sistema.	0	0	0	0	**NO**
			Existe un método que se ejecuta al momento de eliminar un usuario del sistema.	0				
Autenticación	**28**	¿La sesión de usuario se caduca si existe inactividad?	Cuando un usuario se autentica en el sistema se crea una nueva sesión de usuario.	1	1	0	0	**NO**
			Los procesos que un usuario puede ejecutar en el sistema se lo hacen mediante métodos.	1				
Autenticación	**29**	¿Existe control de sesión en todos los módulos del sistema?	Cuando un usuario se autentica en el sistema se crea una nueva sesión de usuario.	1	1	0	0	**NO**
			Los procesos que un usuario puede ejecutar en el sistema se lo hacen mediante métodos.	1				
Autorización	**30**	¿Se asignan perfiles de acceso a los usuarios?	Existe un módulo de administración de usuarios en el sistema.	0	0	0	0	**NO**
			Existen perfiles de acceso creados en el sistema.	0				
			Existe un método que se ejecuta al momento de crear o modificar un usuario en el sistema.	0				
Autorización	**31**	¿Los perfiles de acceso limitan la ejecución de procesos?	Existen perfiles de acceso creados en el sistema.	0	0	0	0	**NO**
			El usuario tiene asignado su respectivo perfil de acceso.	0				
			Los procesos que un usuario puede ejecutar en el sistema se lo hacen mediante métodos	1				
Autorización	**32**	¿Se controla que el usuario solamente pueda tener asignado un solo perfil de acceso?	Existe un módulo de administración de usuarios en el sistema.	0	0	1	0	**NO**
			Existen perfiles de acceso creados en el sistema.	0				
			Existe un método que se ejecuta al momento de crear o modificar un usuario en el sistema.	0				