Lineamientos que deben observar los Almacenes Generales de Depósito, para llevar el registro permanente y simultáneo de sus operaciones, de conformidad con el artículo 119, fracción II de la Ley Aduanera.

Anexo

Recomendaciones para la habilitación del portal web

Julio de 2015

Requerimientos No Funcionales de Seguridad

AGCTI/Administración Central de Transformación Tecnológica

Arquitectura de Referencia para aplicaciones Web

1.1 Diseño de Aplicaciones por Capas

Esta sección muestra la estructura general de las aplicaciones en términos de agrupar lógicamente los componentes en capas separadas que se comunican entre ellas, con otros clientes u otras aplicaciones. Las capas están encargadas de dividir lógicamente los componentes y su funcionalidad, y no tienen relación directa con la ubicación física de los componentes. Las capas pueden localizarse en diferentes niveles físicos (tiers), o puede residir en el mismo nivel físico.

Es importante entender la distinción entre capas y niveles. Las capas (layers) describen el agrupamiento lógico de la funcionalidad y los componentes de una aplicación; mientras que, los niveles (tiers) describen la distribución física de la funcionalidad y los componentes en servidores separados, computadoras, redes, o ubicaciones remotas. A pesar de que tanto las capas como los niveles usan el mismo conjunto de nombres (presentación, negocio, servicios, y datos), hay que recordar que solo los niveles (tiers) implican una separación física. Es muy común tener más de una capa en la misma máquina física (en el mismo nivel (tier)). Se puede pensar en el término nivel (tier) refiriéndose a los patrones de distribución física como 2-niveles, 3-niveles y n-niveles.

1.1.1 Diseño de capas lógicas

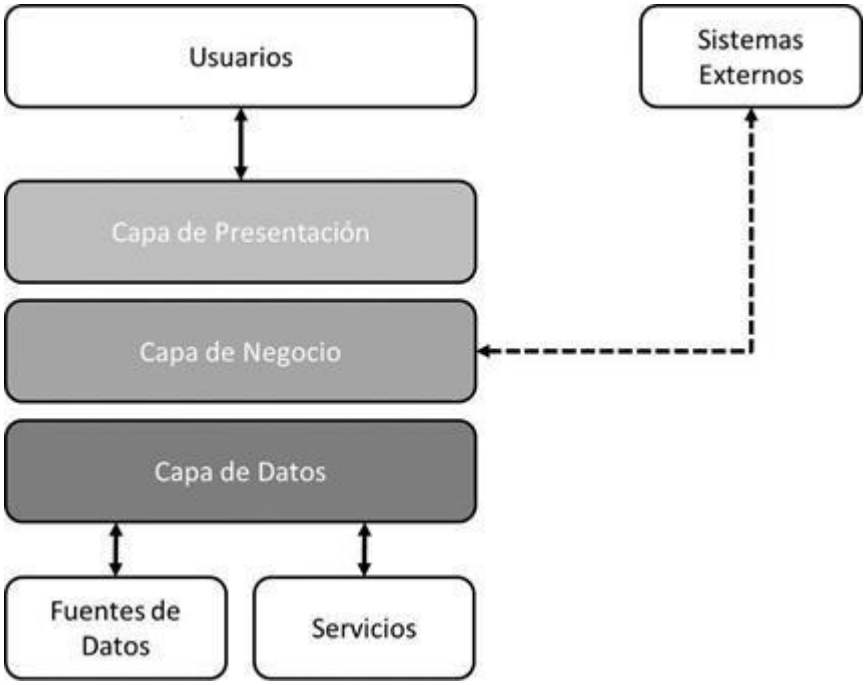
Sin tomar en cuenta el tipo de aplicación que se esté diseñando, y si tiene interfaz de usuario o expone servicios, se puede descomponer el diseño en grupos lógicos de componentes de software. Estos grupos lógicos son llamados capas. Las capas ayudan a diferenciar entre los diferentes tipos de tareas que son ejecutadas por los componentes, haciendo más fácil crear diseños que soportan la reutilización de componentes. Cada capa lógica contiene un número discreto de tipos de componentes agrupados en subcapas, con cada subcapa ejecutando un tipo de tarea específico.

Dividir una aplicación en capas separadas que tienen distintos roles y funcionalidades ayuda a maximizar el mantenimiento del código, optimizando la forma en que la aplicación trabaja cuando se despliega de diferentes formas, y provee una clara distinción entre las ubicaciones donde cierta tecnología o decisiones de diseño se tienen que hacer.

1.1.2 Capas de presentación, negocio y datos

La siguiente figura muestra una forma simplificada, de alto nivel de la representación de las capas y su relación con usuarios, otras aplicaciones que llaman servicios implementados dentro de la capa de negocio de la aplicación, fuentes de datos como bases de datos relacionales o servicios

web que proveen acceso a datos, y servicios externos o remotos que son consumidos por la aplicación.



Capas de presentación, negocio y datos

El diseño más común es el de tres capas que está compuesto por:

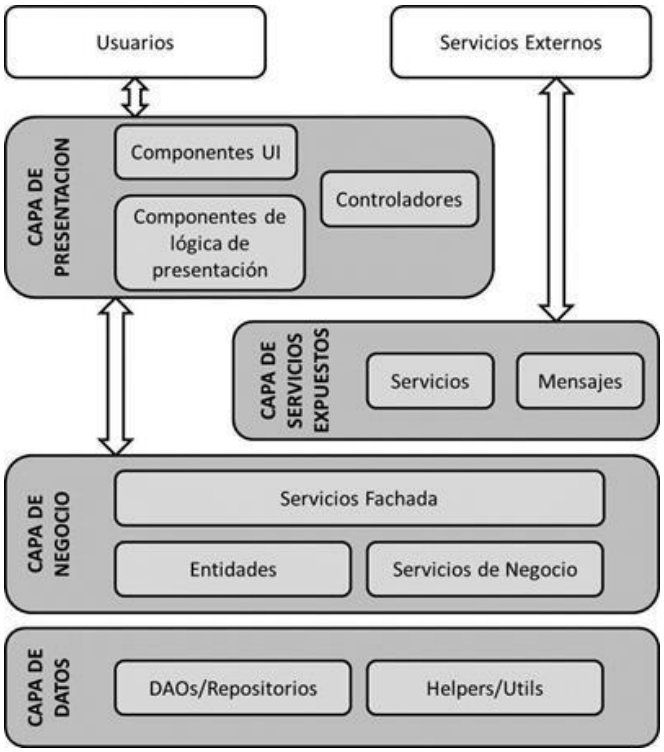
Capa de presentación. Esta capa contiene la funcionalidad orientada al usuario responsable de administrar la interacción del usuario con el sistema, y generalmente consiste de componentes que proveen un puente común hacia la lógica de negocio encapsulada en la capa de negocio.

Capa de negocio. Esta capa implementa la funcionalidad principal del sistema, y encapsula la lógica de negocio. Generalmente consiste de componentes que exponen interfaces de servicios que otros pueden usar.

Capa de datos. Esta capa provee acceso a los datos almacenados del sistema, y a datos expuestos por otros sistemas; quizá accedidos vía servicios. Esta capa expone interfaces genéricas que los componentes en la capa de negocio pueden consumir.

1.1.3 Capa de servicios expuestos (cuando aplique)

Cuando una aplicación debe proveer servicios a otras aplicaciones, así como implementar funcionalidad para soportar a los clientes directamente, una estrategia común es usar una capa de servicios para exponer la funcionalidad del negocio a las otras aplicaciones.



Capa de servicios expuestos

En este escenario, los usuarios pueden acceder a la aplicación vía la capa de presentación, la cual se comunica directamente con los componentes de la capa de negocio o vía una fachada de aplicación (Application Facade/Service Facade) en la capa de negocio. Por otro lado, clientes externos u otros sistemas pueden acceder a la aplicación y hacer uso de su funcionalidad comunicándose con la capa de negocio vía servicios expuestos. Esto da un mejor soporte a muchos tipos de clientes y promueve el re-uso y un nivel alto de composición de funcionalidad entre las aplicaciones.

Se podría dar el caso que la capa de presentación se comunique con la capa de negocio vía la capa de servicios. Sin embargo, esto no es un requerimiento. Si el despliegue de la aplicación ubica la capa de presentación y la capa de negocio en el mismo nivel físico, estas pueden comunicarse directamente. Y aun cuando la capa de presentación y la capa de negocio no están en el mismo nivel físico, con ayuda del framework de Spring se puede crear virtualmente la capa de servicios expuestos y dejar que el framework se encargue de la comunicación remota.

1.2 Niveles físicos (Tiers) y despliegue

Las aplicaciones deben ser desplegadas en ambientes físicos donde las limitaciones de la infraestructura pueden limitar algunas decisiones arquitectónicas. De tal forma, se tienen que considerar los escenarios de despliegue y la infraestructura como parte del proceso de diseño de la aplicación. Al considerar los escenarios posibles de despliegue de la aplicación como parte del proceso de diseño, se previenen situaciones donde la aplicación no puede ser desplegada con éxito o falla en el rendimiento debido a las limitaciones técnicas de la infraestructura.

Seleccionar la estrategia de despliegue impacta en el diseño. Por ejemplo, puede haber restricciones de protocolos o puertos, o topologías de despliegue no soportadas por los centros de datos. Identificar las restricciones de despliegue en fases tempranas ayuda a evitar sorpresas posteriormente; involucrar a los miembros de los equipos de redes e infraestructura ayuda a este proceso. Cuando se seleccione la estrategia de despliegue hay que:

Entender el ambiente físico destino de despliegue.

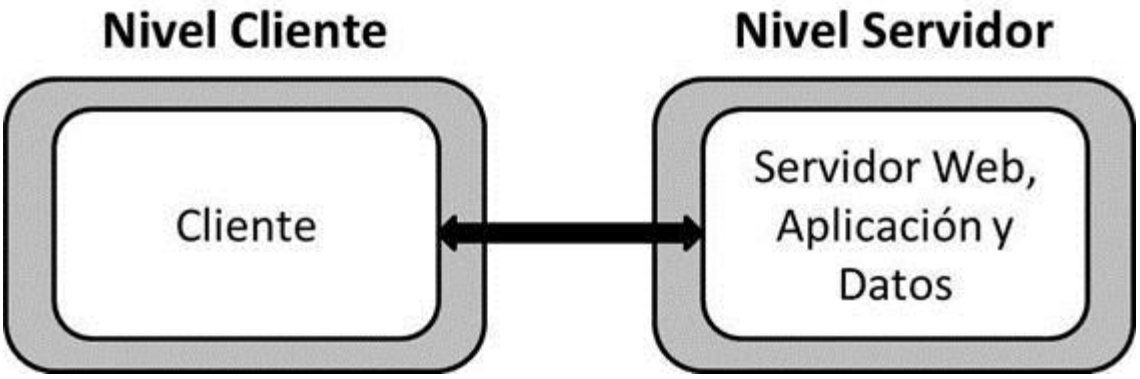
Entender las restricciones arquitectónicas y de diseño basadas en el ambiente de despliegue.

Entender los impactos en seguridad y desempeño del ambiente de despliegue.

1.2.1 Patrones de despliegue

1.2.1.1 Despliegue Cliente-Servidor

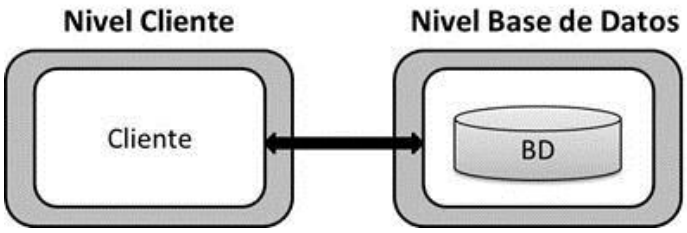
Este patrón representa la estructura básica con dos componentes: un cliente y un servidor. En este escenario, el cliente y el servidor generalmente se localizan en 2 niveles separados.



Despliegue Cliente-Servidor

1.2.1.2 Despliegue 2-Niveles

Como se puede notar la forma es muy parecida al patrón cliente/servidor. Difiere principalmente en la forma en que los componentes en los niveles se comunican. En algunos casos, toda la aplicación se localiza en el cliente y la base de datos se localiza en un servidor separado. El cliente hace uso de procedimientos almacenados en la base de datos.



Despliegue 2-Niveles

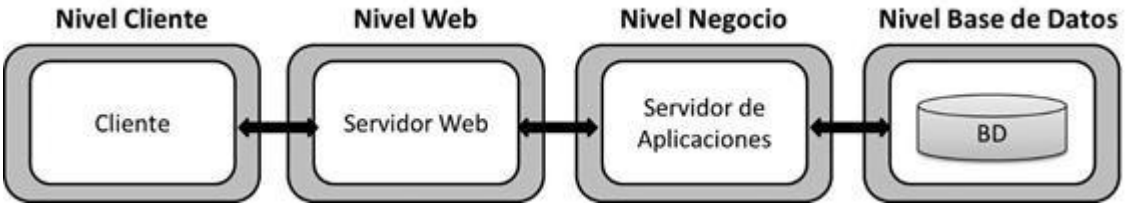
1.2.1.3 Despliegue 3-Niveles

En un diseño de 3-Niveles, el cliente interactúa con el software de la aplicación en un servidor separado, y el servidor de la aplicación interactúa con la base de datos localizada en otro servidor. Este es un patrón muy común en muchas aplicaciones Web o servicios Web, y es suficiente para los escenarios generales. Se puede implementar un firewall entre el nivel cliente y el nivel de Web/Aplicación, y otro entre Web/Aplicación y el nivel de base de datos.

Despliegue 3-Niveles

1.2.1.4 Despliegue 4-Niveles

En este escenario, el servidor Web se encuentra físicamente separado del servidor de aplicaciones/procesamiento. Esto se hace por razones de seguridad, el servidor Web se despliega en una red de perímetro y accede al servidor de aplicaciones ubicado en una diferente subred. En este escenario, se implementa un firewall entre cada uno de los niveles.



Despliegue 4-Niveles

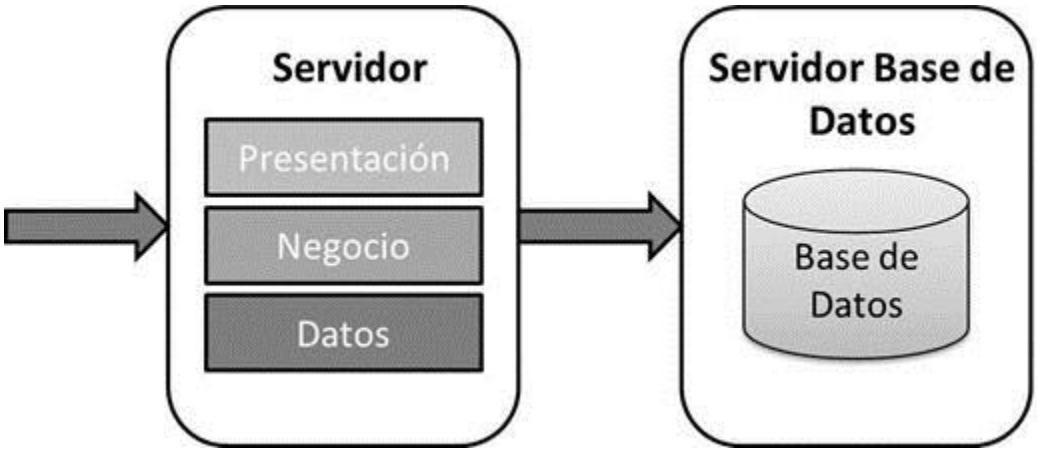
Este escenario es el recomendado por la presente arquitectura de referencia.

1.2.2 Despliegue distribuido y no-distribuido

Si se está construyendo una aplicación simple en intranet para la organización, la cual será accedida por un conjunto finito de usuarios, se puede considerar un despliegue no distribuido. Si se construye una aplicación más compleja, la cual debe ser optimizada para escalabilidad y mantenimiento, se debe considerar un despliegue distribuido.

1.2.2.1 Despliegue no-distribuido

En el despliegue no distribuido, toda la funcionalidad y capas residen en un servidor excepto por la funcionalidad de almacenamiento de datos.



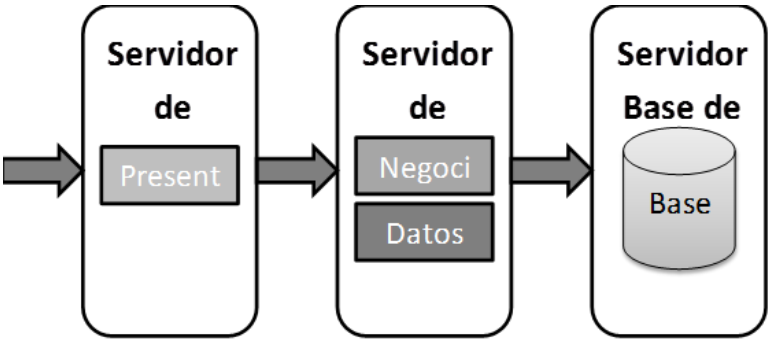
Despliegue no-distribuido

Esta forma tiene la ventaja de simplicidad y minimiza el número de servidores físicos necesarios. También minimiza el impacto de desempeño inherente a la comunicación entre capas que deben cruzar servidores físicos o en clúster en red. Hay que tener en cuenta que al usar un solo servidor,
a pesar de minimizar el impacto en el desempeño por comunicación remota, se puede impactar en el desempeño en otras formas. Debido a que todas las capas comparten recursos, una capa

puede afectar de manera negativa a las demás capas cuando está siendo utilizada de manera constante. Además, los servidores deben ser configurados y diseñados de manera genérica tomando en cuenta los requerimientos de operación más estrictos, y deben soportar el uso máximo de consumo de los recursos del sistema. El uso de un solo nivel reduce de manera general la escalabilidad y mantenimiento debido a que todas las capas comparten los mismos recursos físicos de hardware.

1.2.2.2 Despliegue distribuido

En un despliegue distribuido, las capas de la aplicación residen en niveles físicos separados. Niveles distribuidos organizan la infraestructura del sistema en conjuntos de niveles físicos que proveen ambientes de servidores optimizados para requerimientos específicos funcionales y uso de recursos del sistema. Esto permite separar las capas de la aplicación en diferentes niveles físicos.



Despliegue distribuido

Una forma distribuida permite configurar los servidores de aplicaciones que contienen a las diversas capas para cumplir mejor los requerimientos de cada capa.

Múltiples niveles requieren de ambientes múltiples. Se puede optimizar cada ambiente para ciertos requerimientos operacionales y de uso de recursos del sistema. Así se pueden desplegar componentes en el nivel físico que mejor cumpla las necesidades de uso para maximizar el desempeño y comportamiento del componente. Entre más niveles se usen, más las opciones de despliegue que se tienen para cada componente. Despliegues distribuidos proveen un ambiente más flexible donde se puede escalar cada nivel de manera horizontal o vertical cuando se presenten problemas de desempeño o cuando la demanda de procesamiento aumente.

Otra razón importante para agregar niveles físicos es aplicar políticas específicas de seguridad. Despliegues distribuidos permiten aplicar seguridad más estricta a los servidores de aplicaciones; por ejemplo agregando firewalls entre los servidores web y los servidores de aplicaciones y/o usar diferentes opciones de autenticación y autorización.

1.2.3 Consideraciones de desempeño y diseño de ambientes distribuidos

Componentes distribuidos a través de niveles físicos pueden verse afectados en desempeño debido al costo de llamadas remotas. Sin embargo, tener una distribución de los componentes mejora la escalabilidad, la administración y reduce costos a largo plazo. Se deben tomar en cuenta las siguientes consideraciones cuando se diseña una aplicación que se ejecuta en una infraestructura físicamente distribuida:

Seleccionar formas de comunicación y protocolos entre niveles físicos que aseguren que los componentes pueden interactuar de una forma segura sin degradar el desempeño de los mismos. Se puede tomar ventaja de llamadas asíncronas u opciones de mensajería para minimizar el bloque cuando se hacen llamadas remotas.

Reducir la complejidad de las interfaces de los componentes. Interfaces altamente granulares que requieren de varias llamadas para realizar una tarea trabajan mejor cuando se localizan en la misma máquina física. Interfaces que solo tienen que realizar una llamada para realizar cada tarea proveen un mejor desempeño cuando los componentes están distribuidos en máquinas físicas separadas además de proveer mejores mecanismos de control de transacciones.

Considerar separar procesos críticos que tomen mucho tiempo en su ejecución de otros procesos que podrían fallar usando clúster físicos separados, y determinando alternativas de recuperación cuando ocurran fallos.

Cuando las capas se comunican de manera distribuida, se debe considerar como se administrara el estado entre los niveles físicos, ya que esto afecta la escalabilidad y el desempeño. Las alternativas para la administración de estado típicamente incluyen:

- **Sin estado**. Esta estrategia ofrece más escalabilidad, pero requiere que el cliente maneje el estado de la información.
- **Con estado**. El estado será mantenido o almacenado por cada petición del cliente. Esto tiende a usar más recursos y es una solución menos escalable, el cliente no lleva registro del estado de la información.

1.2.4 Ventajas y desventajas de ambientes con niveles múltiples (N-Tiers)

Los ambientes distribuidos (4-Niveles, N-Niveles) ofrecen ventajas y también tienen algunas desventajas comparados con ambientes no-distribuidos (3-Niveles). Una de las primeras ventajas de un ambiente distribuido es que ofrece la creación de aplicaciones más poderosas debido a la capacidad de escalar las aplicaciones tanto horizontalmente como verticalmente. Ofrece la facilidad de exponer servicios a diversos clientes debido a su naturaleza distribuida donde se pueden dedicar algunos servidores a atender servicios expuestos. Mejoran la seguridad debido a que se pueden agregar firewalls entre los servidores y ubicarlos en diversas zonas protegidas de una red. Además, ofrecen una mejor escalabilidad y alta disponibilidad de las aplicaciones.

Tienen la desventaja de que el diseño de las aplicaciones se hace más complejo debido a que hay que tomar en cuenta la distribución de los componentes.

La siguiente tabla muestra una comparación de una arquitectura simple y una distribuida en varios niveles físicos:

Arquitectura	Ventajas	Desventajas
3-Niveles (Cliente, Servidor, Base de datos)	Desarrollo de componentes más simple. Un solo artefacto de instalación.	Con un mal diseño podría existir alto acoplamiento entre los componentes de software. Poco escalable.
	Operación de los componentes más simple. Comunicación directa entre capas/componentes. Monitoreo más simple.	Balanceo de carga por aplicación. Se dificulta la exposición de servicios a terceros. Mejoras y actualizaciones aplican a la aplicación completa. Se dificulta la implementación de mecanismos de procesamiento asíncrono. Poca tolerancia a fallos. Potenciales problemas de seguridad al ubicar los componentes en una sola

4-Niveles (o mas niveles)	Escalable. Alta disponibilidad. Obliga a tener un diseño desacoplado entre componentes. Se facilita la exposición de servicios a terceros. Balanceo de carga por componentes. Mejoras y actualizaciones por componentes o niveles sin afectar a los demás. Facilidad para implementar mecanismos de procesamiento asíncrono. Alta tolerancia a fallos. Mayor seguridad al tener los componentes distribuidos (la seguridad puede ser personalizada por cada nivel). Mayor flexibilidad al negocio (servicios/componentes pueden ser reemplazados rápidamente para satisfacer las necesidades del negocio). Mayor flexibilidad tecnológica (al ofrecer poco acoplamiento entre servidores la tecnología puede reemplazarse según las necesidades).	Desarrollo de componentes más complejo. Varios artefactos de instalación. Comunicación remota entre capas/componentes. Monitoreo más complejo.

• Apegarse a las especificaciones Técnicas vigentes de Seguridad que están en el Marco Tecnológico de Referencia del SAT

• Aplicación de CAPTCHA en páginas web y móviles.

o Para las soluciones tecnológicas (portal Web o móvil) del SAT que son públicas y no requieren autenticación (por usuario/contraseña, Fiel o ambos u otro mecanismo de autenticación), es obligatorio el uso de mecanismos y componentes CAPTCHA a fin de garantizar que un ser humano es quien hace uso de los sistemas SAT, evitando que robots (bots) exploten ciertos servicios y con ello mitigar ataques DoS (Denial of Service) o DDoS (Distributed Denial of Service) en ambientes web o dispositivos móviles, como teléfonos inteligentes o tabletas.

• SHA256 (SHA–2 256).

o Se deben utilizar como algoritmos de integridad la función criptográfica hash SHA–2 con una longitud (tamaño del bloque) de al menos 256 bits para el intercambio de información entre aplicaciones distribuidas, centro de datos incluyendo los cloud computing, o intercambios de información entre el SAT y otras instituciones.

• SSL 2048

o HTTPS es un protocolo de aplicación basado en el protocolo HTTP, provee un medio seguro de comunicaciones basado en SSL/TSL para crear un canal seguro de comunicación para el tráfico de información. Se debe utilizar para la encripción asociada a HTTP, certificados de clave pública de al menos 2048 bits para el transporte SSL/TSL.

• La información de negocio SAT siempre deberá estar almacenada en la capa de datos.

• Eliminar (Information disclosure) de los nuevos sistemas, mensajes de información a los errores provocados de http como 400 o 500 etc, que proporcionan información de versiones de servidores web o de aplicaciones.

• Pruebas de vulnerabilidad (caja blanca, negra y de penetración). Todas las aplicaciones nuevas del SAT deberán de someterse a pruebas de vulnerabilidad y atender de manera inmediata las denominadas críticas o de color rojo antes de salir a producción.

• Federación de Identidades. Es obligatorio para los nuevos aplicativos que requieran un SSO y un SLO SAT usar para la autenticación por Federación de identidades, la cual se basa en protocolos estándar SAML2, WS-Federation, Liberty Alliance, OAuth u OpenID para el intercambio de autenticación y autorización. Para lo cual es necesario tener en cuenta lo siguiente:

o Se requiere el intercambio de metadata entre los Ids (Identity Providers)y los SPs (Service Providers). Los IdPs del SAT y los SPs de los aplicativos El service provider perteneciente al aplicativo debe utilizar y soportar alguno de los protocolos estándar para la federación.

o Los atributos requeridos del aplicativo deben ser incluidos en el paquete de la federación, conforme a las necesidades de cada aplicación, los cuales serán configurados por el personal a cargo de la administración de identidades del aplicativo con la condición que estos existan en el directorio Institucional.

• El sistema deberá de mitigar y controlar ataques XSS (Cross Site Scripting), deberá adicionar medidas robustas (no solo limitando el número de caracteres utilizando en el caso de java el atributo "maxlenght"), la validación de los parámetros recogidos de los formularios deberá realizarse siempre en el servidor (emplear, por ejemplo, javaScript en el cliente para validar campos de un formulario antes de enviarlos incrementa usabilidad y no esperar a la respuesta del servidor para temas de experiencia del usuario). Se deberá sanitizar los datos recogidos en un formulario para evitar que el atacante juegue con el encoding de los caracteres que estaría enviando. Se deberá utilizar librerías con soporte para sanitizar los parámetros del usuario, por ejemplo OWASP.

• La seguridad de los sistemas debe estar regida por el diseño de Arquitectura de Seguridad del SAT.

• Para la autorización los usuarios serán clasificados en varios tipos, es decir, a través de roles para el acceso a las opciones de trabajo definidas para cada rol, deberán los sistemas restringir acceso derivado del rol obtenido de directorio de identidades institucional, es decir el control de acceso implementado debe permitir asignar y control los perfiles para cada uno de los roles identificados. Los roles deberán generarse de acuerdo a los que negocio SAT defina.

• El sistema deberá contar con mecanismos que permitan el registro de actividades con Identificación de los usuarios que los realizaron.

• El sistema debe contar con pistas de auditoría de las actividades que se realizan sobre el sistema con niveles razonables para su reconstrucción e identificación de los hechos

• El sistema debe validar automáticamente la información contenida en los formularios de ingreso. En el proceso de validación de la información, se deben tener en cuenta aspectos tales como obligatoriedad y coherencia.

• El sistema con control de acceso debe estar restringido, usuario y contraseña, fiel o ambos en base al mecanismo que defina negocio.

• El sistema deberá de rechazar accesos o modificaciones no autorizadas.

• El sistema deberá generar pistas de auditoría referentes a los procesos del sistema. El sistema deberá apegarse a las normas para la construcción de aplicaciones con pistas de auditoría emitidas por la ACSMC, las cuales serán proporcionadas por la administración monitoreo y control.

• Disponibilidad de funciones. Dentro de la interfaz, las funciones disponibles que se mostraran serán aquellas para las cuales tiene permiso el usuario de acuerdo a su perfil. Es responsabilidad de la aplicación la implementación este control.

• El aplicativo debe estar en un esquema de alta disponibilidad.

• Ejecución de pruebas en ambientes apropiados y protección de información. Todas las actividades de pruebas deberán separarse de los sistemas de producción.

• Toda la información usada para probar los sistemas se deberá proteger de acuerdo con las políticas de SAT considerando los convenios de confidencialidad y la responsabilidad sobre el uso de esta.

• Análisis forense de cambios, los cuales deberán apegarse a las definiciones que proporcione la ACSMC por medio de la administración de control y monitoreo.

o Los aplicativos deberán apegarse a las definiciones para la implementación de pistas de auditoría que deberán solicitar para su implementación a la Subadministración de Auditoría de TI, de la ACSMC.

Los aplicativos que requieran consultar información del contribuyente (IdC) deberán utilizar el web service de SISE para el tema de los protección de los contribuyentes políticamente y económicamente expuestos (PPEs).

o Envío de pistas de auditoría a correlacionador. Los servicios deben contar con mecanismos para enviar la información de las pistas de auditoría al correlacionador. Los mecanismos y definiciones deberán de solicitarse a la Subadministración De Auditoría De TI de la ACSMC.

• Administración de roles y permisos por el directorio institucional vigente. Esta es la herramienta que se usará para administrar los roles y permisos, para centralizar y tener más control de las identidades para la autorización y proporcionar SSO Y SLO.

• Los atributos que el directorio institucional puede proporcionar serán solamente los obtenidos de las fuentes oficiales RH y RFC ampliado.

• Para la autenticación y autorización los sistemas deberán integrarse a los componentes de control de acceso y directorio institucional vigent, con el fin de tener centralizada la autenticación y tener más control.

• Establecimiento de controles de seguridad y confidencialidad. Las aplicaciones deben proveer controles de seguridad y confidencialidad apropiados, deberán apegarse a las definiciones que brinde la ACSMC por medio de la Administración de Arquitectura de Seguridad.

• Es responsabilidad del líder de proyecto notificar formalmente si la aplicación decide no integrarse al control de acceso y directorio institucional. Mediante un documento formal deberá de especificar quién es el responsable de esas identidades y quién firmará las responsivas de creación de identidades como roles y perfiles de ese aplicativo fuera de los estándares.

• Control de acceso a aplicaciones por medio de roles. El control de acceso a las aplicaciones y servicios debe ser establecido por medio de roles con apego al directorio institucional.

• Acceso autorizado a datos. Los servicios deberán contar con mecanismos para que los usuarios estén estrictamente limitados a los elementos de los datos a los que tienen acceso autorizado, es responsabilidad de la aplicación control lo anteriormente mencionado.

Las aplicaciones, ya sea por el ciclo de software o productos de caja (commercial off-the-shelf (COTS)), deberán de soportar enmascaramiento de direcciones IP, es decir soportar, NAT (Network Address Translation).

• El responsable del aplicativo deberá de vigilar el parcheo de vulnerabilidades de los sistemas operativos que conforman su aplicativo.

• Evitar el uso de cuentas de administración del sistema operativo para los ambientes de ejecución de productos y tecnologías. Se requiere que el usuario utilizado para la administración de los productos tecnológicos de la Encriptación de cookies. Siempre que viaje una sesión de acceso valida al sistema o arquitectura no sea realizado con cuentas de administración del sistema operativo (SA, root, administrador). Se deberá crear una cuenta específica para la administración del producto restringida a sus responsabilidades.

• Se requiere la definición de los servicios, puertos, protocolos, IP's origen e IP's destino en cada una de las aplicaciones o sistemas, previo a liberarse a producción , el cual deberá de contar con los artefactos de arquitectura (modelo de sembrado físico , flujos unifilares y especificación de arquitectura

• Las aplicaciones deberán tener ambientes no productivos y productivos para que estén en posibilidad de probar o actualizar su aplicación sin afectar producción.

• Las aplicaciones deberán siempre pasar por el análisis de vulnerabilidades y atender los problemas de color rojo para poder salir a producción.

• Se deberá generar las configuraciones de los servidores para solamente habilitar características que se utilizarán, generando un hardening en sus servidores a utilizar.

• Encriptación de información sensible o con clasificación confidencial. Toda comunicación que requiera autenticación o que tenga información sensible o confidencial deberá estar encriptada. El área de negocio del SAT dueña de la información clasificará el tipo de información.

• El enmascaramiento de cookies. Deberá generarse para evitar el traslape de dominios. Deberá ser responsabilidad del control de acceso institucional proporcionar esa funcionalidad y deberá siempre estar en con https.

Información sensible, la encriptación de cookies deberá ser generada por el aplicativo y deberá siempre estar con https.

• Algoritmos de Encriptación de información utilizados en el SAT. Con base en la clasificación de la información por proporcionada por Negocio SAT, deberá determinarse si la información debe ser encriptada, los algoritmos de encriptación utilizados por el SAT son los siguientes:

o Criptografía simétrica - tamaños de llave de 192 bits como mínimo superior. o Diffie

Hellman. Método para generar llaves simetricas de forma aleatoria.

o Criptografía asimétrica RSA - 2048 bits como mínimo o superior

o Firmas digitales. Su razón principal es asegurar el no repudio de las transacciones. Las deberán utilizar los aplicativos que deseen asegurar el no repudio de envió de información por parte del contribuyente.

-La Firma Electrónica Avanzada (Fiel) es un archivo digital que identifica al contribuyente al realizar trámites por internet en el SAT e incluso en otras dependencias del Gobierno de la República.

-La Fiel es única, es un archivo seguro y cifrado que (equivale a la firma caligráfica del contribuyente.

-Por sus características, es segura y garantiza la identidad.

-Se deberán siempre asegurar que su diseño se base en el uso de dos claves o llaves para el envío de mensajes: la llave privada, que sólo deberá de conocer y poseer el titular de la Fiel, (la cual sirve para asegurar el no repudio por parte del contribuyente) y la llave pública, para consulta libre.

o Digestión. Una función hash es un algoritmo que transforma ("digiere") un conjunto arbitrario de elementos de datos, como puede ser un archivo de texto, en un único valor de longitud fija (el "hash"). El valor hash calculado puede ser utilizado para la verificación de la integridad de copias de un dato original sin la necesidad de proveer el dato original. Esta irreversibilidad significa que un valor hash puede ser libremente distribuido o almacenado, ya que sólo se utiliza para fines de comparación. Se deberá utilizar el algoritmo SHA-2 a 256 como mínimo o superior

o Se deberá utilizar para la encripción la biblioteca SgiCripto, la cual se ocupa en diversos proyectos SAT, desarrollada para facilitar y realizar actividades criptográficas como:
-Cifrado asimétrico
-Cifrado simetrico
-Conexión Cliente ARA
-Digestión
-Firma
-Sobres

• Las aplicaciones que requieran la firma electrónica para autenticación autónoma vía firma electrónica avanzada (Fiel) o firmado de documentos para asegurar el no repudio deberán considerar la re-compilación cada determinado tiempo para soportar el crecimiento del tamaño de los certificados. Deberán soportar llaves de 1024, 2048 y 4096 a partir de la presente publicación de este documento.

• Mecanismos de autenticación y protección en Web Services. Deberán ser con apego a los utilizados en los xml Gateway que cuente la institución. Los web services deberán considerar el uso de mecanismos de autenticación como WS-Security 1.1 con WS-Policy o mutual autentication por certificados auto generados por el SAT.

• Encriptación de usuario y password. Estos deberán siempre viajar en un canal seguro de comunicación. Los datos de usuario y password deben viajar encriptados o por un canal seguro cuando sean transmitidos por internet o en red interna.

• Utilización de HTTPS para transacciones expuestas a Internet. Los servicios relacionados con transacciones deben considerar la utilización de HTTPS.

• No es requerido mecanismo de encriptación para sitios de públicos en general. Para las páginas web que son para el público en general que son de sólo consulta no existe el requerimiento de mecanismos de encriptación o canal seguro, solamente uso de CAPTCHA obligatorio.

• Análisis de vulnerabilidades sobre aplicaciones. Se requiere la ejecución de análisis de vulnerabilidades y pruebas de seguridad exhaustivas sobre las aplicaciones para validar que se está entregando una aplicación segura.

• Alineación a las reglas de construcción de contraseñas definida por la ACSMC por la AMIF (Administración de Manejo de identidades y Fiel) Los protocolos de transmisión de datos aprobados por seguridad de la información por canal seguro son:

o Socket de seguridad. o SFTP

o HTTPS

o SMTP / (SSL / TLS.)

o (JMS ó MQ) / (SSL / TLS.)

o SSH (siempre y cuando sea por puerto diferente para el puerto 22 para la red productiva, para la red administrativa si es permitido el puerto 22.

o SSL

• Autorización de usuarios centralizada. Se requieren mecanismos para realizar las autorizaciones de usuarios del sistema de forma centralizada y por medio de la administración central de seguridad monitoreo y control.

• Para el control de acceso en COTS (producto de caja) de MQ deberá ser restricción IP homologada, listas de control de acceso (ACL) o certificados y canal seguro SSL.

o Para el esquema de intercambio de información por MQ se debe cifrar el canal por SSL bajo el algoritmo de cifrado SSL_RSA_WITH_AES_128_CBC_SHA con FIPS.

o La autenticación para el intercambio de información por MQ se debe realizar bajo un esquema de intercambio de certificados.

• Utilización de Selladora Digital para transacciones exitosas a las aplicaciones que requieran entregar un acuse por parte del SAT. Actualmente en el SAT se utiliza la Selladora Digital de tal forma que todos los acuses que salen del SAT implementan este servicio. Se requiere evaluar la implementación de este servicio para las transacciones exitosas, en caso de que aplique.

• Se deben de considerar perfiles para el acceso a los datos.

• Criterios para aplicación de controles de seguridad. Los requerimientos de seguridad estarán basados sobre los requerimientos del negocio, no sobre requerimientos técnicos.

• Aplicación de mecanismos estándares de seguridad. Se requiere el uso de mecanismos de seguridad estándares en la instrumentación de los sistemas de seguridad de la información.

• Vigencia de mecanismos estándares de seguridad y aplicación a sistemas. Todos los sistemas deben trabajar con mecanismos estándares actuales.

• Acceso a sistemas por medio de autenticación. Todo acceso a cualquier sistema de cómputo deberá ser autenticado apegado a los lineamientos de la arquitectura de seguridad institucional.

• En caso de que se requieran hacer respaldos por medio de la red, se deberá utilizar una tarjeta de red específica.

• Las nuevas soluciones tecnológicas deberán de entregar los artefactos de arquitectura e integrar y plasmar las definiciones que proporcione arquitectura de seguridad con apego a Marco Documental vigente.

• Lineamientos para algoritmos de encriptación. El tipo de algoritmos de encriptación deberá definirse en conjunto con Arquitectura de seguridad SAT, tomando en consideración la funcionalidad requerida, esquemas de transferencia, almacenamiento y otros temas.

1.2.5 Componentes de Software

Esta sección lista y describe los distintos componentes de software que pueden ser usados en los aplicaciones web desarrolladas en el SAT y forman parte de la arquitectura de software.

1.2.5.1 Web Browser

Un **web browser** (comúnmente conocido como navegador o browser) es una aplicación de software para obtener, presentar y compartir recursos de información sobre la World Wide Web. Un recurso de información es identificado por un Identificador de Recurso Uniforme (Uniform Resource Identifier o URI) y puede ser una página web, una imagen, un video u otra pieza de contenido. Los hipervínculos que se encuentran los recursos dan a los usuarios facilidad para navegar en sus navegadores recursos relacionados. También puede ser definido como un programa o software de aplicación diseñado para permitir a los usuarios acceder, obtener y ver documentos y otros recursos en la Internet.

Aunque los navegadores están destinados principalmente para usar la World Wide Web, pueden ser usados para acceder información proporcionada por servidores web en redes privadas o archivos en sistemas de archivos.

Los principales web browsers son:

Google Chrome

Firefox

Internet Explorer

Opera

Safari

Es importante señalar que el navegador Institucional para empleados del SAT es Internet Explorer, aunque las aplicaciones Web de contribuyentes deben estar preparadas para funcional en cualquier navegador de los listados.