



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Sistemas para monitorear el consumo de un WEB
SERVICES de generación de timbres fiscales**

TESIS

PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN

PRESENTA:

Abraham Espinal Santos

Arturo Ramos Ortega

Déborah Abni Solís Jiménez

Gabriel Rodríguez Saavedra

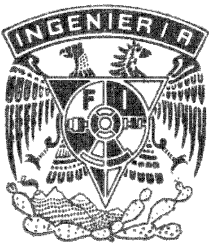
Jaime Alejandro Victoria Miguel

DIRECTOR DE TESIS:

M.I. Juan Carlos Roa Beiza.

MÉXICO D.F.

NOVIEMBRE DE 2014.



Agradecimientos

Déborah Abni Solís Jiménez

A Bety mi mamá

Por apoyarme en todo momento, por los valores que me ha inculcado, por su esfuerzo para darme una excelente educación en el transcurso de mi vida, por darme la oportunidad de estudiar esta carrera y ser mi ejemplo de vida.

A mi hermano Hugo

Por ser parte importante en mi vida, demostrando su apoyo incondicionalmente, llenándome de alegría y amor cuando más lo he necesitado.

A mi familia

Por su apoyo, comprensión y sobre todo su amor.

A mis compañeros, Abraham Espinal, Arturo Ramos, Gabriel Rodríguez, y Alejandro Victoria por estos años de amistad y esfuerzo que nos han permitido lograr diversos objetivos juntos.

Al M.I. Juan Carlos Roa Beiza, director de tesis, por su valiosa guía y asesoramiento a la realización de la misma.

Al PAT por su Orientación en el desarrollo del trabajo

A mí alma mater, Universidad Nacional Autónoma de México, por las lecciones que aprendí en ella, el enriquecimiento de conocimiento y valores que me ha dado.

A todos aquellos que mostraron interés por nuestro trabajo, a quienes han formado parte de mi vida profesional, gracias por su amistad, consejos, apoyo y compañía en todo momento. Algunas personas continúan conmigo, otras en mis recuerdos y en mi corazón; sin importar en donde estén quiero darles las gracias por formar parte de mí y por todo lo que me han brindado.

Abraham Espinal Santos

El presente trabajo de tesis es la culminación de una de las metas más importantes de mi vida, significa para mí un orgullo gigantesco al ver que todo el esfuerzo invertido dio resultados y se encuentra reflejado en dicho trabajo y en el ejercicio de mi profesión.

Doy gracias a Dios por haberme permitido llegar hasta donde estoy, por darme la sabiduría para trazar un camino de bien, por darme la inteligencia para resolver los conflictos y por darme la oportunidad de vivir cada día y ser una mejor persona.

Quiero agradecer profundamente a mis bisabuelos, a mis abuelos, a mi madre, mi hermano y a mi familia quienes son los pilares de mi vida a quienes amo, quiero y respeto porque desde pequeño me ayudaron a crecer con sus consejos, con su constancia, con su dedicación y con la firme confianza en mí, sin ellos y sin todo su esfuerzo esta meta nunca se habría cumplido.

Al mismo tiempo quiero agradecer el apoyo de mis mejores y más cercanos amigos quienes siempre me apoyaron en el transcurso de mi vida como estudiante, fuera de ella y que en la actualidad son parte importante de mí.

Mi más sincero agradecimiento a la Facultad de Ingeniería y a la Universidad Nacional Autónoma de México por haber sido mi segunda casa, por enseñarme lo que se, por darme los mejores amigos y por prepararme para la vida profesional.

Agradezco profundamente el apoyo otorgado por el PAT, al M.I. Juan Carlos Roa Beiza y a mis compañeros de tesis Déborah Solís, Arturo Ramos, Alejandro Victoria y Gabriel Rodríguez por el apoyo, la constancia y el esfuerzo dedicado a lo largo de estos meses.

No hay palabras para describir la felicidad y la satisfacción dentro de mí al ver que he logrado superarme y que pese a las dificultades he salido adelante.

Pero con mucho orgullo hoy puedo decir, lo logre, soy INGENIERO.

Arturo Ramos Ortega

Primero que nada quiero agradecer a Dios por permitirme alcanzar una de mis metas, la primera que me impuse cuando aún tenía pocos años de vida, terminar una carrera en la Facultad de Ingeniería de la UNAM.

Agradezco a mis padres (Olga y Arturo), a mis hermanas (Alin y Fer) y a mis tías (Heri, Nabo, Rosa y Lupe), sin su apoyo, cariño, consejos, regaños no habría alcanzado este objetivo, son la parte importante de mi vida y me esforzare más por ustedes.

A la Facultad de Ingeniería, por darme los conocimientos con los que ahora puedo desenvolverme en el ámbito profesional, y con esto ser alguien y algo en la vida.

Al PAT por tener este programa, y hacer que no solo yo, o mis compañeros, si no muchos estudiantes podamos dar el último paso que nos faltaba para llegar a una de nuestras metas.

Al M.I. Juan Carlos Roa, por la asesoría prestada en la dirección de este trabajo y poder hacer un trabajo que parecía demasiado complicado de una forma sencilla.

Finalmente agradezco a mis compañeros Déborah Solís, Abraham Espinal, Alejandro Victoria y Gabriel Rodríguez. Debbi, Abraham, Alex, Gabo muchas gracias por su tiempo, esfuerzo y dedicación para que pudiéramos llevar este proyecto a cabo, pero sobre todo por su amistad que es aún más valioso para mí.

Jaime Alejandro Victoria Miguel

A mis padres, Eduardo y Epifania.

Por ser un soporte esencial en mi vida, por enseñarme el valor del trabajo, del esfuerzo y por el cariño y la confianza que han depositado en mí. Por el tiempo que siempre me han dado, por el apoyo emocional y económico que siempre me han brindado. Porque sus esfuerzos y sacrificios han sido y serán una motivación para mí, pero sobre todo porque son y serán mis ejemplos de personas a seguir.

A mis hermanos Eduardo y Jorge.

Quienes siempre me han acompañado, que siempre me han enseñado y quienes nunca me han dejado caer o desistir, quienes son ejemplo de admiración por sus esfuerzos y dedicación, por sus conocimientos y por las largas horas de conversación compartiendo conocimiento, compartiendo risas, anécdotas y las vivencias que hemos tenido, que tendremos día a día. Gracias!

A mi familia tanto materna como paterna, pero sobre todo a GRACIAS a mis tíos **Thomas** y **Ninfa**, quienes siempre me han apoyado, quienes han confiado en mí y me han fomentado el seguir y no desistir, brindándome todo tipo de ayuda, consejos y regaños, por lo que por siempre les estaré agradecidos.

Gracias a mi amada Facultad de Ingeniería.

Por las oportunidades, por los retos, por los esfuerzos que siempre ha solicitado y requerido, porque todo ello es lo que genera que algún día, uno se pueda volver digno de llamarse Ingeniero.

A nuestro asesor de tesis, el M.I. Juan Carlos Roa Beiza.

Quien nos ayudado, nos ha marcado el ritmo de trabajo y quien nos ha fomentado seguir y concluir con este proyecto, con este esfuerzo.

GRACIAS A MIS COMPAÑEROS DE PROYECTO: GABRIEL, ABRAHAM, ARTURO Y DEBORA. Gracias por todo su esfuerzo, conocimiento y dedicación llevados a cabo en la realización de esta tesis y porque gracias a

ustedes, a NOSOTROS, se puede tener este escalón para alcanzar este anhelado sueño.

**Y FINALMENTE GRACIAS AL PAT POR DARNOS ESTA OPORTUNIDAD
CON ESTE PROYECTO PARA LOGRAR CERRAR UN CICLO MÁS...**

Gabriel Rodríguez Saavedra

Escribir los agradecimientos es algo muy complicado ya que por fortuna o por desgracia la vida no es una constante, todo está en continuo movimiento y quizás las personas que una vez estuvieron con nosotros ahora simplemente se han ido, dejando en nuestras vidas aprendizaje, consejos, vivencias, lecciones aprendidas, pero también muchas veces tristeza y sufrimiento, es por eso que decidí hacer esta sección muy general y no adentrarme a detalles con los cuales podría escribir un libro completo:

Comenzaré con la familia, doy gracias a mi madre (Lulú), por haberme dado el apoyo económico y moral para poder alcanzar esta meta, por todo tu esfuerzo y sacrificio muchas gracias. A mi hermano (Christian), quien me enseñó que en ocasiones no hay que tomarse la vida tan en serio. A mi abuela materna (Rita) por haberme enseñado las bases de la vida y ser un ejemplo de fortaleza. A mi abuelo materno (Martín), que aunque no lo conocí, siempre quise ser como ese gran ingeniero del que todos hablan maravillas. A todos y cada uno de los integrantes de mi familia, tanto materna como paterna por haber sido parte de mi crecimiento y paso por estos lugares.

A todos mis amigos que me han acompañado en las buenas y en las malas durante toda mi vida, en especial a mi mejor amiga (Iris) quien siempre me ha motivado a seguir adelante. A mis amigos y partícipes de esta tesis: Abraham, Alejandro, Arturo y Déborah, por ser buenos compañeros, amigos inolvidables y por emprender esta travesía juntos.

A mi Alma Mater, la UNAM, por brindarme la oportunidad de pertenecer a ella y por todas las lecciones aprendidas dentro de sus aulas y fuera en sus jardines. Al PAT del palacio de minería por darnos este empujón que necesitábamos para ya por fin titularnos.

Y finalmente, gracias a la vida, Dios, energía, como sea que se le quiera llamar, por darme las oportunidades que he tenido de seguir creciendo en el ámbito profesional.

Índice

Capítulo I. Entorno del problema	1
1.1 Introducción	2
1.2 Leyes y reglamentos del SAT que se deben cubrir.	6
1.3 Conceptos generales de la operación de timbrado de documentos electrónicos	11
1.4 Organigrama funcional de la empresa y responsables de las diferentes areas	18
1.5 Proceso actual para la generación y registro de los CFDI	23
Capítulo II. Marco teórico.	28
2.1 Características, ventajas y desventajas de un WEB SERVICE	29
2.2 Características, ventajas y desventajas de MS-SQL Server	33
2.3 Características, ventajas y desventajas de la metodología ITIL.	37
2.4 Características, ventajas y desventajas Visual Studio.	43
2.5 Características, ventajas y desventajas de perl	46
2.6 Características, ventajas y desventajas de la arquitectura cliente / servidor	51
Capítulo III Estrategia del servicio	57
3.1 Análisis del problema	58
3.2 Recopilación y análisis de datos	65
3.3 Requerimientos de la aplicación	67
3.4 Planteamiento de la solución	70
3.5 Bosquejo de pantallas sugeridas a la empresa	85
Capítulo IV Diseño y construcción de la aplicación	89
4.1 Modelado del sistema	90
4.2 Creación de la base de datos	97
4.3 Diseño y creación de la interfaz de usuario	112

4.4. Pruebas y mantenimiento	117
4.5 Reportes	127
Capítulo V Transición del servicio	132
5.1 Plan de transición y soporte	133
5.2 Aceptación y liberación	135
Capítulo VI Operación del servicio	140
6.1 Eventos, incidentes y alertas del sistema	142
6.2 Accesos, solicitudes y cambios	144
Capítulo VII Mejora continua de los siete pasos	145
7.1 Propósito	146
7.2 Objetivos	146
7.3 Valor al negocio	147
7.4 Proceso de los siete pasos	147
Conclusiones	150
Bibliografía	153

Capítulo I

Entorno del problema



1.1 INTRODUCCIÓN

La facturación electrónica es la elaboración de comprobantes fiscales mediante representaciones digitales, las cuales deben de cumplir con un reglamento establecido lo cual garantiza su validez ante los organismos que las solicitan.

En México el archivo generado es un documento XML(eXtensible Markup Language) el cual contiene todo lo necesario tanto del emisor como del receptor, los montos y desglose de impuestos para que el documento pueda tener efectos fiscales y para que al momento de hacer su declaración de impuestos, el contribuyente pueda hacerlo de una manera más simple, además contiene un sello digital el cual es generado a partir de la combinación de los datos contenidos en el XML (cadena original) y un certificado de sello digital del emisor, haciendo que el documento no pueda alterarse y garantizando su autenticidad.

Este esquema de facturación se ha utilizado desde 2004, y en el transcurso de este tiempo ha tenido diversos cambios hasta llegar al modelo que actualmente se utiliza: Comprobantes Fiscales Digitales por Internet (CFDI).

El CFDI tiene la peculiaridad de que su emisión se realiza a través de un proveedor autorizado de certificación (PAC) el cual se encarga de validar que los documentos generados cumplan con las especificaciones del Sistema de Administración Tributaria (SAT) que es la dependencia que se encarga de gestionar los temas fiscales en México, una vez validado, el PAC debe de generar un folio fiscal (timbre) el cual debe de incluirse en el CFDI como garantía de su validez.

La generación de los timbres fiscales se realizan mediante peticiones a un Web Service (WS) suministrado por el PAC, por lo que la disponibilidad y los tiempos de respuesta que ofrece el WS son de suma importancia para el PAC para brindar los servicios de timbrado, ya que de ello depende la correcta generación de los documentos fiscales de



sus usuarios, y la mínima falla en él, puede representar una importante pérdida económica para ellos, originando multas, indemnizaciones y/o pérdidas de clientes para el PAC.

El objetivo de esta tesis es, mediante la información que provee el PAC acerca del WS de timbrado, poder generar un sistema de monitoreo, el cual permita tener datos en línea y con ellos llevar un control sobre el comportamiento del WS y poder reaccionar oportunamente ante posibles eventualidades, con lo cual se pretende minimizar el riesgo de fallos en el sistema de generación de CFDI. Además dicho sistema debe de cumplir con las especificaciones proporcionadas por el PAC.

Para llevar a cabo este sistema se utilizará la metodología de ITIL (Information Technology Infrastructure Library) ya que es uno de los estándares utilizados por el PAC, además de ser una metodología orientada al ciclo de vida de los servicios, lo cual permitirá establecer un marco de referencia para la estrategia, diseño, transición, operación y mejora continua del sistema desarrollado.

Lo primero que se requiere para entender porque el WS de generación de timbres es tan crítico para un PAC, es conocer los reglamentos emitidos por el SAT para la generación de CFDI, ya que en ellos se indican los parámetros de validación, los tiempos de entrega, los derechos, las obligaciones, así como los castigos por incumplimiento, entre muchos otros puntos que deben de cumplir tanto el emisor, el receptor y el PAC para que un documento sea deducible.

También se requiere conocer los conceptos generales del proceso de timbrado, es decir, lo que implica para un contribuyente poder generar sus comprobantes fiscales a través de un proveedor. Por último es necesario conocer los pasos necesarios para que el PAC pueda generar un folio fiscal y devolverlo al contribuyente, los actores involucrados en cada uno de ellos y la estructura del proceso. Todos estos conceptos están detallados en el primer capítulo.



En el segundo capítulo se habla acerca de las herramientas utilizadas en la creación de este sistema de monitoreo para lo cual primero explicamos que es un WS y cómo funciona la arquitectura cliente-servidor, posteriormente tenemos una reseña del software utilizado y aunque creemos que realmente este sistema podría haberse realizado para otro tipo de sistema operativo, lenguaje de programación, manejador de base de datos e incluso utilizar otro tipo de metodología durante la construcción del mismo, una de las especificaciones del PAC para la construcción del sistema, fue la de utilizar la tecnología de software con la que cuentan actualmente, bajo la cual están desarrollados la mayoría de sus productos y servicios y por lo tanto es la que más se facilita poder dar mantenimiento y mejoras en el futuro además de evitar gastos innecesarios en la adquisición de otro tipo de licencias; es por ello que se optó por el uso de Visual Studio como entorno de programación, y C# como lenguaje de programación por los beneficios de la programación orientada a objetos; la base de datos esta generada en MS SQL Server y aquí se describen los beneficios de la utilización de ITIL como metodología.

En el tercer capítulo se empezará a emplear la metodología de ITIL, haciendo uso del paso *estrategias de servicio*, en el cual se comienza un análisis más a fondo del problema al cual se le dará solución, así mismo se mostrará la información proporcionada por el PAC y la forma en que ésta podrá ser manejada para que sea de utilidad, ya que aunque podemos obtener mucha información acerca de los timbrados, no toda es funcional para su proceso de timbrado, o simplemente es de carácter crítico o confidencial para el PAC o sus clientes y por cuestiones de seguridad no nos permiten utilizarla.

También dentro de este capítulo se presenta la propuesta de solución, para llegar a ella se muestran los requerimientos iniciales propuestos por el PAC, lo que esperan obtener y mediante qué recursos, para esto hacen uso de bosquejos de las pantallas de captura explicando detalladamente cada una de las funcionalidades



Para el capítulo cuarto se toma el segundo paso de la metodología de ITIL que hace referencia al diseño y construcción de la aplicación, en él se detallan cada uno de los pasos empleados en la construcción de la solución, la elaboración de diagramas de contexto, casos de uso, entidad relación y el diccionario de datos, se muestran los scripts de la generación de la base de datos y se muestran pantallas

Una vez finalizada esta parte se muestran los procesos de prueba que se realizaron, así como los errores encontrados durante el proceso y la solución que se implementó en cada uno de ellos, haciendo que la liberación del sistema sea estable durante su liberación a producción. Finalmente se muestran los reportes que se fueron generando a partir del desarrollo del sistema, así como del mantenimiento que se le fue realizando durante el proceso de elaboración del mismo.

El capítulo quinto habla del tercer paso de la metodología de ITIL, la *transición de servicio*, que es el proceso mediante el cual se hacen las pruebas de funcionamiento, la validación, se da visto bueno por parte del usuario final, se concluye con la aceptación y liberación a producción del sistema.

En el capítulo sexto *operación del servicio*, se describe el funcionamiento del sistema ya como parte del trabajo diario del PAC, los accesos y solicitudes que tiene, así como las notificaciones que genera, finalmente se muestran las fallas que presentó durante este proceso y las intervenciones que se hicieron para solucionarlas.

Finalmente en el capítulo séptimo se describe el proceso mediante el cual se realizará una *mejora continua* del servicio, para que de esta manera el sistema pueda ser adecuado a las necesidades futuras del PAC y de este modo no quede inutilizado y su ciclo de vida sea el mayor posible.



1.2 LEYES Y REGLAMENTOS DEL SAT QUE SE DEBEN CUBRIR.

El pasado 30 de diciembre de 2013, se publicó en el Diario Oficial de la Federación (DOF) la Resolución Miscelánea Fiscal (RMF) que está vigente para 2014. El Código Fiscal de la Federación establece la obligación para los contribuyentes a expedir a partir del 1 de enero del 2014, Comprobantes Fiscales Digitales por Internet (CFDI) por todos los actos u operaciones que realicen, por los ingresos que perciban o por las retenciones de contribuciones que efectúen.

Los comprobantes que pueden emitirse en 2014 conforme las disposiciones fiscales son:

- Factura Electrónica a través de Internet (CFDI)
 - Factura Electrónica a través de un Proveedor Autorizado de Certificación de CFDI.
 - Factura Electrónica a través de adquirentes de bienes de los sectores primarios o mineros, o de los servicios de arrendamiento de espacios para la colocación de publicidad o antenas de telefonía.
 - Factura Electrónica expedida por organizaciones y asociaciones que reúnan a productores y comercializadores agrícolas del Sistema Producto.

- Factura Electrónica (CFD)
 - Factura Electrónica (para personas físicas que en el último ejercicio declarado tuvieron ingresos iguales o menores a 500 mil pesos y emitieron CFD, podrán seguir facturando bajo este esquema hasta el 31 de marzo de 2014, siempre y cuando a partir del 1 de abril migren al esquema de facturación electrónica CFDI).
 - Y sus representaciones impresas correspondientes, cuando el receptor así lo solicite.



- Facturación en papel.
 - Factura impresa con Código de Barras Bidimensional (para personas físicas que en el último ejercicio declarado tuvieron ingresos iguales o menores a 500 mil pesos y emitieron CBB, podrán seguir facturando bajo este esquema hasta el 31 de marzo de 2014, siempre y cuando a partir del 1 de abril migren al esquema de facturación electrónica).

Fundamento Legal: Arts. 29, 29-A del CFF y las reglas I.2.7.3.1 y I.2.7.4.1 de la RMF para 2014.

Requisitos que debe contener un CFDI para la Factura Electrónica.

Las personas físicas requieren únicamente para facturar electrónicamente, su Registro Federal de Contribuyentes (RFC) y Firma Electrónica Avanzada (Fiel), a diferencia de las personas morales, no es necesario que tramiten su Certificado de Sello Digital.

Los CFDI deberán cumplir con lo siguiente como está estipulado el DOF vigente para el 2014:

- RFC de los Contribuyentes
- Régimen Fiscal en que tributen
- Deberá señalar el domicilio del local o establecimiento
- Número de folio
- Sello digital
- Lugar y fecha de expedición.
- Cantidad, unidad de medida y clase de los bienes, mercancías o descripción del servicio o del uso y goce que amparen.
- Valor unitario consignado en número.
- Importe total
- Indicar si son parcialidades.
- Impuestos trasladados



- Forma de pago
- Número y fecha del documento aduanero, tratándose de ventas de primera mano de mercancías de importación.

Adicionales a ello, deberán cumplir con el Anexo 20 de la Resolución Miscelánea Fiscal vigente para el 2014 que se define como:

- Código de barras generado conforme al Anexo 20.
- Número de serie del CSD del emisor y del SAT.
- Las leyendas: “Este documento es una representación impresa de un CFDI” o “Este documento es una representación impresa de un Comprobante Fiscal Digital a través de Internet”.
- Referencia bancaria o número de cheque
- Fecha y hora de emisión y de certificación de la Factura Electrónica (CFDI)
- Cadena original del complemento de certificación digital del SAT.

Contenido del Anexo 20 de la RMF para 2014, Medios electrónicos: Estándar de Comprobante fiscal digital a través de Internet.

Formato electrónico único

“...El contribuyente que opte por emitir comprobantes fiscales digitales a través de Internet deberá generarlos bajo el siguiente estándar XSD base y los XSD complementarios que requiera, validando su forma y sintaxis en un archivo con extensión XML, siendo éste el único formato para poder representar y almacenar comprobantes de manera electrónica o digital.

Para poder ser validado, el comprobante fiscal digital a través de Internet deberá estar referenciado al namespace (esquema de elementos y atributos únicos en un documento XML) del comprobante fiscal digital a través de Internet y referenciar la validación del mismo a la ruta publicada por el SAT en donde se encuentra el esquema



XSD objeto de la presente sección:
(http://www.sat.gob.mx/sitio_internet/cfd/3/cfdv32.xsd) de la siguiente manera:

<cfdi:Comprobante

xmlns:cfdi="http://www.sat.gob.mx/cfd/3"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="

<http://www.sat.gob.mx/cfd/3>

http://www.sat.gob.mx/sitio_internet/cfd/3/cfdv32.xsd"

.....

</cfdi:Comprobante>

Adicionalmente a las reglas de estructura planteadas dentro del presente estándar, el contribuyente que opte por este mecanismo de generación de comprobantes deberá sujetarse tanto a las disposiciones fiscales vigentes, como a los lineamientos técnicos de forma y sintaxis para la generación de archivos XML especificados por el consorcio w3, establecidos en www.w3.org...".

Uso del Complemento obligatorio Timbre Fiscal Digital

El resultado de la validación de un CFDI, asignación de un folio fiscal e incorporación del sello digital del SAT se entenderá como el Timbrado Fiscal Digital. El folio fiscal digital será referido como el UUID (Universally Unique Identifier – Identificador Universal Único). Para integrar el complemento "TimbreFiscalDigital" a un comprobante fiscal digital a través de Internet, la estructura resultante deberá integrarse como un nodo hijo del nodo:

Comprobante/Complemento/TimbreFiscalDigital.



Adicional a su inclusión, se deberá definir el namespace correspondiente dentro del nodo Comprobante, así como referenciar la ubicación pública del esquema XSD correspondiente.

Plazo de entrega del CFDI

A partir del 2014 todos los contribuyentes que expidan comprobantes fiscales deberán entregar o hacer llegar a sus clientes en forma inmediata el archivo electrónico del CFDI. Una vez que el cliente solicite el CFDI, se le deberá entregar la representación impresa de dicho comprobante que únicamente presume la existencia del XML original, esto como parte de los requisitos que debe contener un CFDI.

Características adicionales del CFDI

A través de un proyecto de reforma de la RMF 2014, el SAT indica que los contribuyentes podrán incorporar en los CFDI que expidan, la expresión NA (No Aplica) o cualquier otra análoga, en lugar de los siguientes requisitos:

- Régimen fiscal en que tributen conforme a la Ley del ISR.
- Forma en que se realizó el pago.

Todo aquel CFDI que no cumpla con los requisitos establecidos que emanan del artículo 29 del Diario Oficial de la Federación, o cuando los datos se plasmen de forma distinta a la misma, no podrán deducirse o acreditarse fiscalmente.



1.3 CONCEPTOS GENERALES DE LA OPERACIÓN DE TIMBRADO DE DOCUMENTOS ELECTRÓNICOS

Antecedentes

En México, la emisión de CFDI's comenzó a partir del primero de enero de 2011, sin embargo el SAT (Servicio de Administración Tributaria) había estado dando facilidades a ciertos sectores de contribuyentes para emitir comprobantes fiscales por otros medios.

Las facilidades que el SAT había brindado se fueron reduciendo cada vez más y finalmente en su Segunda Revisión Miscelánea Fiscal para el 2013 publicó que el CFD (Comprobante Fiscal Digital), principal medio de emisión alternativo al CFDI, desaparecería, dejando como único medio de facturación al CFDI, dando como fecha límite el 31 de diciembre de 2013 para poder seguir generando CFD's.

En la Resolución Miscelánea Fiscal para el 2014 se publicó que a partir del primero de enero de 2014 todos los contribuyentes fiscales tendrían que migrar al esquema de facturación CFDI, ya que los CFD's y CBB's (Facturas impresas con Código de Barras Bidimensional) dejarían de ser comprobantes fiscales válidos y por ende no serían deducibles, únicamente se brindó una prórroga de tres meses para las personas físicas que ya emitían CFD's o CBB's y cuyos ingresos no hubiesen sido superiores a los quinientos mil pesos en su última declaración y para los contribuyentes que se encuentran dados de alta bajo el Régimen de Incorporación, quedando comprometidos a que a partir del primero de abril del 2014 tendrían que estar totalmente migrados al esquema de facturación CFDI y además estos últimos tendrían que haber facturado todos sus movimientos del año en este mismo esquema.



A partir del primero de abril de 2014 todos los documentos fiscales que se generen deberán ser por medios digitales, el uso del papel desaparecerá para este propósito, es decir, únicamente se utilizará el esquema CFDI.

Conceptos generales

Para comprender mejor el esquema de timbrado de documentos fiscales electrónicos es necesario conocer, de forma general, algunos conceptos sobre los elementos involucrados en el proceso:

- Proveedor Autorizado de Certificación (PAC).

Es una persona moral autorizada y certificada por el SAT, que cumple con la tarea de validar los documentos fiscales digitales que son generados por cierto emisor en cuanto a contenido y estructura, en caso de pasar la validación le regresa el timbre fiscal digital del documento, posterior a esto envía la información recolectada al propio SAT de forma inmediata.

- Timbrado o timbre fiscal digital.

Cadena de caracteres que es asignada a un CFDI, es compuesta por la versión del timbre, el folio fiscal digital (UUID), la fecha de timbrado, el sello del CFD y el número de certificado asignado por el SAT del PAC que certificó el documento.

- Certificado de Sello Digital (CSD).

Es un documento digital a través del cual una entidad certificadora, en este caso el SAT, asegura que una entidad que utiliza dicho certificado es realmente quien dice ser, evitar el robo de identidad es, en su mayoría, responsabilidad del dueño de los datos contenidos en el certificado.



- Firma Electrónica Avanzada (FIEL).

Es un documento digital de identificación, al igual que el CSD, es emitido por el SAT y sirve para asegurar la identidad de una persona física, el archivo contiene la firma caligráfica de la persona a la que pertenece y nuevamente es responsabilidad del dueño evitar el robo de identidad protegiendo el archivo de la FIEL y su contraseña de cifrado.

- Lista de Contribuyentes con Obligación (LCO).

Es un listado generado y publicado por el SAT que contiene el RFC, el número de los Certificados de Sello Digital que manejan, su vigencia e información sobre el estatus ante el SAT de todas las razones sociales dadas de alta, es un archivo XML firmado bajo el estándar PKCS#7.

- Emisor de comprobante.

Es la persona física o moral que incluye su clave de Registro Federal de Contribuyentes (RFC) dentro del campo “Emisor” en un CFDI y que además firma el documento digital con su CSD o su FIEL; usualmente es la entidad que genera el documento fiscal digital quien ejerce el papel de emisor, sin embargo existen esquemas en donde quien genera el documento no es el emisor, sino el receptor.

- Receptor de comprobante.

Persona física o moral cuya clave RFC aparece dentro del campo “Receptor” en un CFDI; usualmente es la entidad que recibe el documento, sin embargo existen esquemas en donde es el receptor quien genera el CFDI a nombre del emisor.



- XML formato SAT.

Es un documento digital con extensión .xml construido bajo un formato ya definido por el SAT, cuyo contenido y estructura ya fue revisado y validado por un PAC y luego de esto fue timbrado, como tal, este archivo es el comprobante con validez fiscal y no el PDF o representación impresa del CFDI.

- Representación impresa de un CFDI.

Es un archivo digital que es entregado usualmente en formato PDF, en el cual se representan de forma comprensible para la mayoría de personas los datos contenidos dentro del XML del CFDI, este documento no tiene validez fiscal.

- Fecha de generación.

Es la fecha y la hora exacta en la que el documento fue generado pero que aún no ha sido timbrado por un PAC.

- Fecha de certificación.

Es la fecha y la hora exacta en la que el documento fue timbrado por un PAC, como validación, la fecha y hora de generación no debe ser mayor a 72 horas del momento en el que se envíe a timbrar, de lo contrario el documento será rechazado.

- Adenda.

Datos adicionales que son requeridos por el receptor del CFDI pero que no son de carácter obligatorio para poder generar el comprobante fiscal.



Componentes mínimos de un CFDI

Para que un CFDI pueda ser válido fiscalmente hablando debe contener como mínimo los siguientes elementos:

- Del emisor, Razón social, RFC, Régimen fiscal, CSD, Sello digital o Número del CSD.

- Del receptor.
 - RFC del receptor.

- Detalles del comprobante.
 - Lugar y fecha de expedición, Fecha y hora de emisión, Descripción del bien o servicio, Valor unitario del bien o servicio, Unidad de medida, Monto de los impuestos trasladados, Monto de los impuestos retenidos o Forma de pago.
 - Método de pago.

- Del timbre.
 - Folio fiscal UUID, Número de serie del certificado del SAT, Fecha y hora de certificación, Sello digital del CFDI.
 - Sello del SAT.
 - Cadena original del complemento de certificación digital del SAT.

Cabe destacar que el XML no podrá ser modificado posterior a que haya sido timbrado, de lo contrario el sello no coincidirá y el comprobante no será fiscalmente



válido además de que generará sanciones sobre quien se adjudique dicho comprobante.

Tipos de documentos que se pueden generar con el esquema CFDI

Listado de documentos que pueden ser generados a través del esquema de CFDI:

- Ingreso:
 - Facturas
 - Recibos de Honorarios
 - Recibos de Pago
 - Recibo de Arrendamiento
 - Recibo de Donativos.
 - Notas de cargo.
- Egreso:
 - Recibos de Nómina
 - Notas de Crédito
 - Notas de Devolución
- Traslado:
 - Comprobantes de Retenciones.
 - Cartas Porte.

Modalidades de facturación CFDI

Se pueden emitir CFDI's de tres formas distintas:

- A través de un PAC.

Es el modelo más conocido, un emisor que genera un bien o servicio genera un CFDI que recibe un receptor que consume dicho bien o servicio.



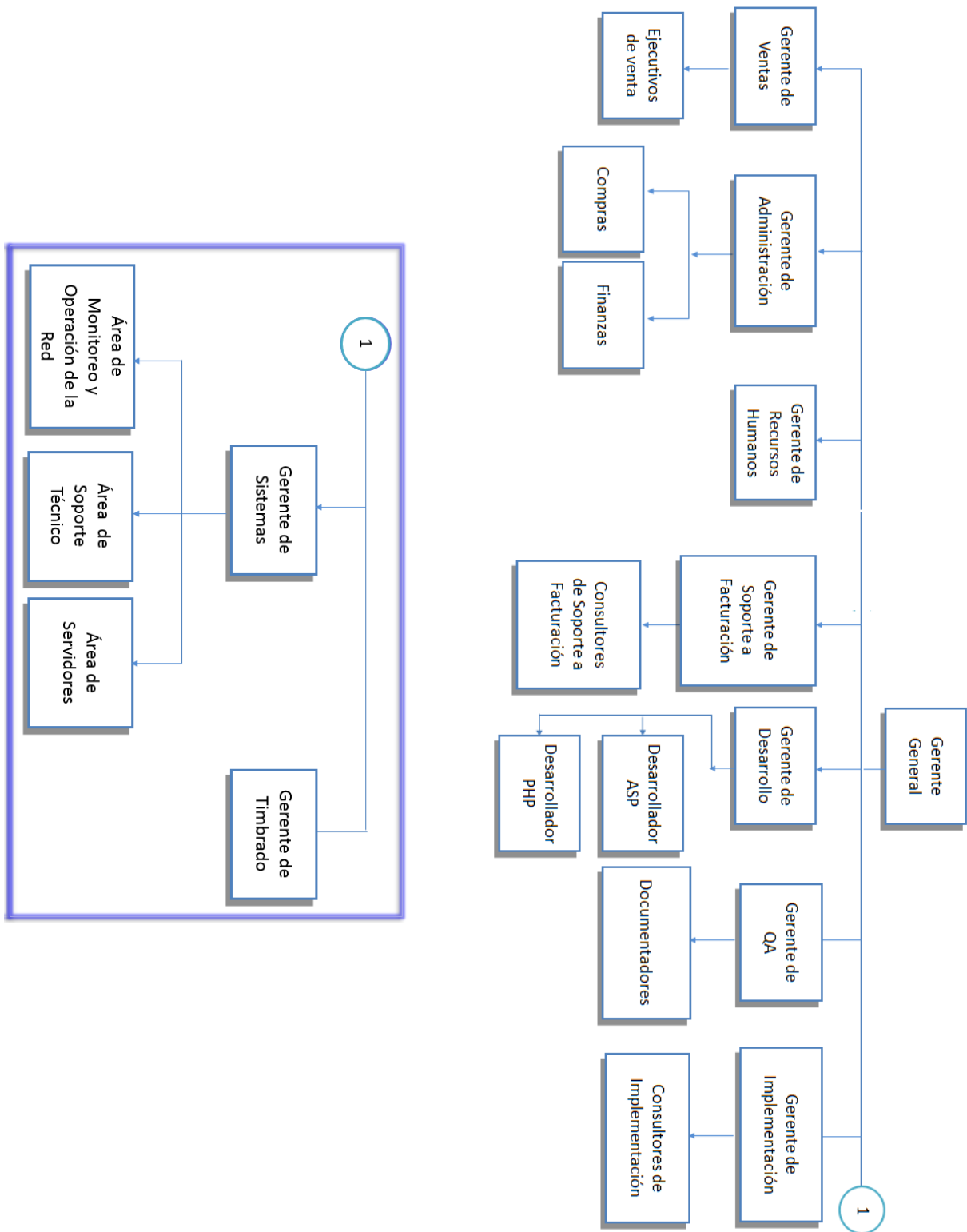
- A través de un adquirente de bienes del Sector Primario.

Un receptor que pretende adquirir bienes de una entidad ubicada dentro del Sector Primario emite un CFDI a través del esquema de Sector Primario a nombre del emisor, es decir, autogenera su comprobante fiscal, se debe cumplir con ciertos requisitos para poder generar CFDI's bajo este esquema.

- A través de organizaciones o asociaciones agrícolas del Sistema Producto. Los productores agrícolas se reúnen en ciertas entidades, las cuales se convierten en el emisor de los CFDI's que generan con los bienes de los productores, las organizaciones deben cumplir con ciertos requisitos para generar CFDI's bajo este esquema.



1.4 ORGANIGRAMA FUNCIONAL DE LA EMPRESA Y RESPONSABLES DE LAS DIFERENTES ÁREAS





Gerente de ventas

Es la persona encargada de organizar, controlar y dirigir el departamento de ventas

Ejecutivos de ventas

Participan en el desarrollo comercial de la empresa para aumentar las ventas, los márgenes definidos y la cuota de mercado actual.

Gerente de Administración y Finanzas

Es quien para mejorar el funcionamiento de las actividades de administración y contabilidad de la empresa, formula y propone normas, políticas y procedimientos. Apoyar a la organización en todas las gestiones legales y reglamentarias.

Compras

Es el departamento encargado de realizar las adquisiciones necesarias en el momento debido, con la cantidad y calidad requerida a un precio adecuado.

Finanzas

Este departamento contribuye en la planeación y seguimiento de programas que ayuden a determinar la estabilidad y desarrollo financiero-económico en la organización.

Gerente de recursos humanos

Es el encargado de coordinar al personal que labora en la empresa así como realizar la búsqueda y selección del personal necesario.



Gerente de soporte a facturación

Responsable de controlar, planear y coordinar las actividades del área de *soporte* a facturación electrónica. Está a cargo de los consultores de soporte a Facturación e interviene en las actividades en caso de ser necesario.

Consultores soporte a facturación

Brindan apoyo a los usuarios cuando se presentan problemas de software o funcionamiento de la herramienta de timbrado.

Gerente de desarrollo

Encargado de Desarrollar e investigar nuevos sistemas para servicio de timbrado, así como innovar las herramientas y aplicaciones actuales.

Desarrollador ASP (Active Server Pages)

Es quien se encarga de diseñar y desarrollar una aplicación en lenguaje ASP

Desarrollador PHP (inicialmente Personal Home Page, ahora PHP Hypertext Preprocessor)

Es quien se encarga de diseñar y desarrollar una aplicación en lenguaje PHP

Gerente de QA (Quality Assurance)

Encargado de asegurar la calidad de las siguientes actividades: diseño, desarrollo, producción, instalación, mantenimiento y documentación, para ser más precisos se asegurará que el producto cumple o excede las expectativas del cliente.



Documentadores

Son aquellos que se encargan de documentar los procesos que deben de llevarse a cabo en cada gerencia. También realizan y actualizan manuales de las diferentes aplicaciones que provee la institución a los clientes.

Gerente de implementación

Organiza y dirige a cada miembro de su equipo para completar los pasos del proyecto o de un nuevo proceso a tiempo. Identifica dificultades tecnológicas y de comunicacionales que se presentan durante el proceso.

Consultores de implementación

Es quien realiza el plan de trabajo de acuerdo a los requerimientos del cliente para poder desarrollar el proyecto. Se inicia con el desarrollo de las actividades basándose en un procedimiento anteriormente definido, debe registrar cada una de ellas en una guía.

Gerente de sistemas

Es el encargado de planear, organizar, dirigir y controlar, el funcionamiento del Área de Sistemas. Sus principales funciones son:

- Determina políticas y procedimientos del uso de HW y SW.
- Implementa nuevos sistemas necesarios en la Institución.
- Coordina y supervisa la elaboración de manuales de uso de las aplicaciones y dispositivos que actualmente están en uso, así como instructivos para HW y SW.
- Tener copias, seguridad y la seguridad de la información en la institución.
- Realiza informes periódicos de las actividades realizadas.



- Supervisa el trabajo del personal a su cargo.
- Está en constante contacto con los gerentes de otras áreas para lograr satisfacer las necesidades de las áreas usuarias, y para reportar el progreso en los proyectos del departamento y uso de recursos

Área de monitoreo y operación de la red

Para garantizar la disponibilidad y eficiencia de la comunicación de las redes locales y de la comunicación con los servicios que se proveen, el área realiza las siguientes actividades:

- Monitoreo proactivo, es el caso de nuestra herramienta de monitoreo para timbrado nos ayudará a identificar el consumo de red en las peticiones al server.
- Tarificación de los enlaces para identificar la cantidad de tráfico generado.
- Administración dispositivos de red que proporcionan la comunicación a Internet, así como de los servidores donde se alojan las herramientas usadas para monitorear
- Instalación y administración de firewalls para los servidores de aplicaciones
- Administración, instalación y uso de las aplicaciones para monitorear la red.

Área de soporte técnico

Tiene a su cargo las siguientes funciones:

- Gestionar el servicio de correo electrónico.
- Administrar la red telefónica institucional.
- Apoyar a usuarios en operaciones de implementación o adecuación de servicios informáticos.
- Detección y eliminación de virus y/o programas espías.
- Instalación y mantenimiento de software propio o programas comerciales.
- Instalación y configuración de componentes internos o externos.
- Recuperación de datos eliminados o destruidos.
- Elaboración de manuales y guías de funcionamiento.



- Desarrollo de planes de mantenimiento.
- Establecer criterios de seguridad.

Área de servidores

Gestiona el sistema operativo del servidor, garantizar que el rendimiento del servidor no se vea afectado, instalar y configurar el software nuevo y las actualizaciones del equipo, solucionar problemas y actualizar la información de cuentas de usuario. Deben realizar copias de seguridad de rutina; integrar las nuevas tecnologías.

Nuestra herramienta de monitoreo apoyará a las actividades del área, en caso de presentar alguna irregularidad en el servicio como lo es tiempos de respuesta altos o disponibilidad del 0%; los administradores deberán revisar a detalle si se trata de un proceso que está consumiendo gran cantidad de recursos del servidor, descartar si se trata de una falla del servicio o detectar si es un error que presenta la red.

Gerente de timbrado

Es quien se encarga exclusivamente del servicio de timbrado, lleva registro, un control y solución de los errores presentados durante el proceso de timbrado. Administra la base de datos que contiene a todos los usuarios de servicio. Realiza las adecuaciones de acuerdo a las normas del SAT.

1.5 PROCESO ACTUAL PARA LA GENERACIÓN Y REGISTRO DE LOS CFDI

Emisión del CFDI.

El primer paso para la generación de un CFDI es que el contribuyente genere un archivo XML con los lineamientos establecidos por el SAT como se muestra en la figura 1.5.1, para dicha generación se puede hacer uso de software propio o software de terceros, el XML deberá contener los requisitos mínimos para su validación como son:



- RFC de los Contribuyentes
- Régimen Fiscal en que tributen
- Deberá señalar el domicilio del local o establecimiento
- Número de folio
- Sello digital
- Lugar y fecha de expedición.
- Cantidad, unidad de medida y clase de los bienes, mercancías o descripción del servicio o del uso y goce que amparen.
- Valor unitario consignado en número.
- Importe total
- Indicar si son parcialidades.
- Impuestos trasladados
- Forma de pago

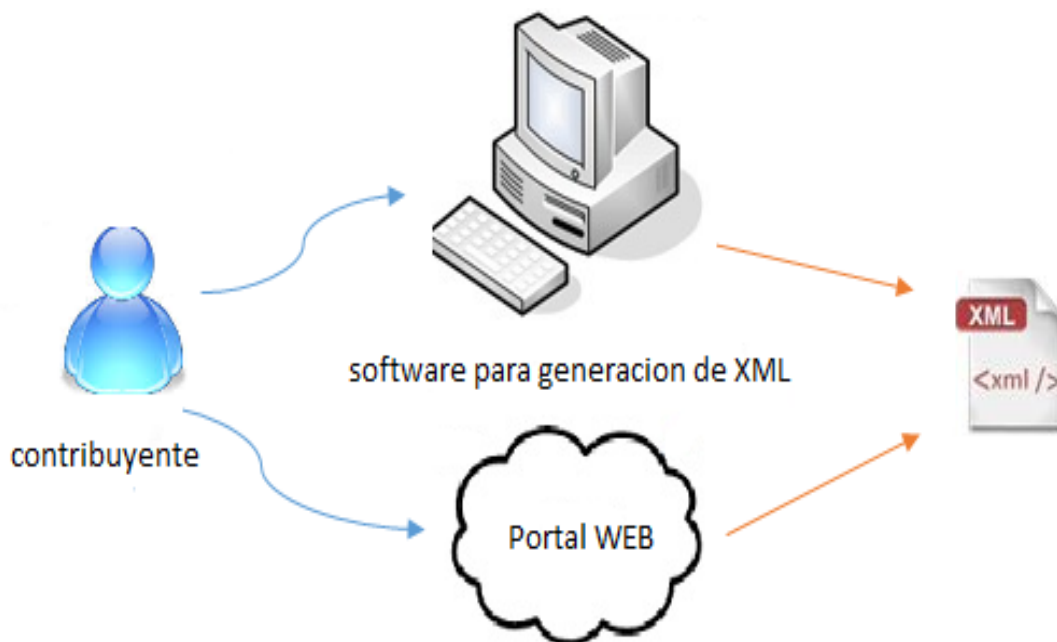


Figura 1.5.1 Emisión de CFDI's



Envío de XML.

Posterior a la creación del XML, el contribuyente procede al envío del mismo a través de distintos protocolos de comunicación definidos y acordados con el proveedor de factura electrónica ver figura 1.5.2. Algunos ejemplos de protocolos para envío son:

- Uso de Webservices.
- Protocolo FTP y SFTP.
- Protocolo HTTP.

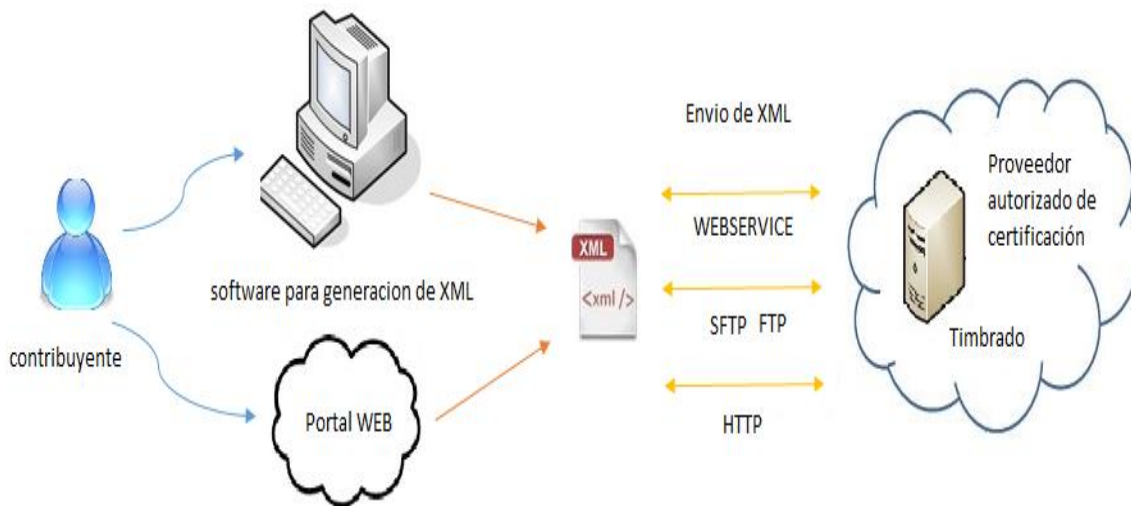


Figura 1.5.2 Envío de XML proveedor.

Recepción y validación de XML.

En este punto el PAC recibe el documento debidamente formado y valida la estructura y los nodos dentro del documento, algunos puntos importantes que valida un PAC son los siguientes:

- Fecha de recepción del documento no mayor a 72 horas posteriores a ser creado.
- Verificación del CSD vigente del contribuyente.
- Verificación de timbre previo del documento.



Una vez que el documento del contribuyente ha sido validado se genera el timbre fiscal digital (TFD) y es regresado al contribuyente mediante el mismo protocolo de comunicación con el que llego ver figura 1.5.3, en este paso el proveedor de factura electrónica añade los siguientes nodos al xml:

- Nodo complemento y nodo TFD que contiene lo siguiente:
 - Versión del timbre digital
 - Sello del proveedor
 - Número del certificado del proveedor
 - Fecha de timbrado
 - UUID

En este punto ya se tiene un CFDI completo y válido el cual es regresado al contribuyente y así mismo puede ser enviado al receptor.

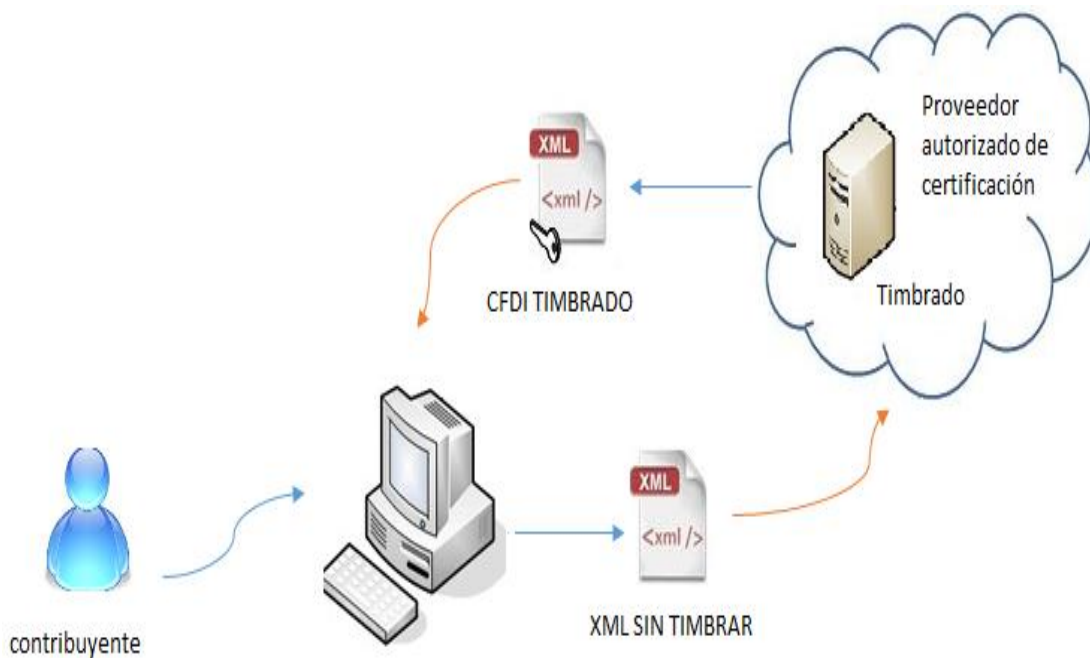


Figura 1.5.3 Recepción y validación de CFDI.



Envío de comprobante al SAT.

Una vez completada la generación del CFDI el PAC tiene la obligación de enviar una copia del mismo al SAT con la finalidad de que el contribuyente se deslinde de las declaraciones mensuales ver figura 1.5.4.

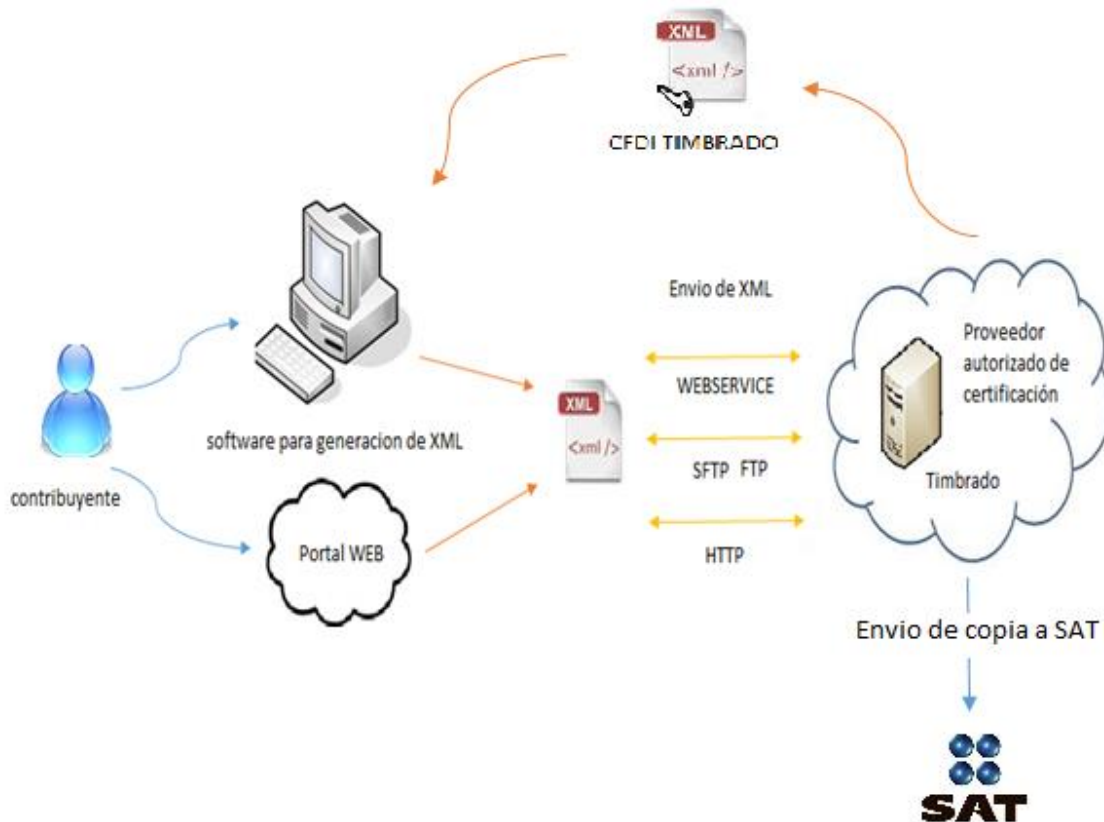


Figura 1.5.4 Envío de comprobante al SAT.

Capítulo II

Marco teórico



2.1 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE UN WEB SERVICE.

Las aplicaciones web actuales ya no son suficientes. El modelo actual de negocio electrónico no facilita la integración de las aplicaciones de Internet con el resto de software de las empresas. Si las compañías quieren extraer el máximo beneficio de Internet, los sitios web deben evolucionar. Este es el contexto en el que surgen los Web Services siendo componentes de software que permiten a los usuarios usar aplicaciones de negocio que comparten datos con otros programas modulares, vía Internet.

“Un Web Services es una interfaz, accesible por protocolos (estándar o no) usados en Internet, que permiten acceder a las funcionalidades de un objeto en concreto, sin importar las tecnologías ni plataformas implicadas en la petición.”¹

El Web Services se coloca entre el usuario y el código usado por éste, y se encargará de abstraer las especificaciones técnicas del programa que atenderá la llamada, para que cualquier lenguaje de programación que tenga soporte para Web Services, tenga acceso a nuestro programa pudiendo ser el elemento cliente un usuario humano, otro Web Services o un programa. Ver figura 2.1.1.



Figura 2.1.1: Esquema del Web Services.

¹ Ribas L. Joan, Web Services (2003). México: Anaya. Pag. 10, 11.



De la definición citada con anterioridad, podemos identificar que los Web Services son aplicaciones independientes de la plataforma que pueden ser fácilmente publicadas, localizadas e invocadas mediante protocolos web estándar, como XML, SOAP, UDDI o WSDL. El objetivo final es la creación de un directorio online de web services, que pueda ser localizado de un modo sencillo y que tenga una alta fiabilidad.

La funcionalidad de los protocolos empleados es la siguiente:

- **XML (eXtensible Markup Language):** Un servicio web es una aplicación web creada en XML. Este estándar permite el intercambio de información estructurada entre diferentes plataformas.
- **WSDL (Web Services Definition Language):** Este protocolo se encarga de describir el web services cuando es publicado. Es el lenguaje XML que los proveedores emplean para describir sus web services.
- **SOAP (Simple Object Access Protocol):** Permite que programas que corren en diferentes sistemas operativos se comuniquen. La comunicación entre las diferentes entidades se realiza mediante mensajes que son enrutados en un sobre SOAP.
- **UDDI (Universal Description Discovery and Integration):** Este protocolo permite la publicación y localización de los servicios. Los directorios UDDI actúan como una guía telefónica de web services.

Características de un Web Services.

Las componentes de los Web Services son:

- **Servicio.** La aplicación es ofrecida para ser utilizada por solicitantes que llenan los requisitos especificados por el proveedor de servicios.
 - La implementación se realiza sobre una plataforma accesible en la red.
 - El servicio se describe a través de un lenguaje de descripción de servicio.



- Tanto la descripción como las políticas de uso han sido publicadas de antemano en un registro.
- **Proveedor de servicio.** Desde el punto de vista comercial, es quien presta el servicio. Desde el punto de vista de arquitectura, es la plataforma que provee el servicio.
- **Registro de servicios.** Es un depósito de descripciones de servicios que puede ser consultado, donde los proveedores de servicios publican sus servicios y los solicitantes encuentran los servicios y detalles para utilizar dichos servicios.
- **Solicitante de servicios.** Desde el punto de vista comercial, la empresa que requiere cierto servicio. Desde el punto de vista de la arquitectura, es la aplicación o cliente que busca y solicita un servicio.

Operaciones de Servicios Web:

- **Publicar / Cancelar.** Los proveedores de servicios publican (publicitan) la disponibilidad de su servicio comercial (e-business) a uno o más registros de servicios, o cancelan la publicación de su servicio.
- **Búsqueda.** Los solicitantes de servicios interactúan con uno o más registros de servicios para descubrir un conjunto de servicios comerciales con los que pueden interactuar para encontrar una solución.
- **Ligar, Unir (Bind).** Los solicitantes de servicios negocian con los proveedores de servicios para acceder e invocar servicios comerciales (e-business).

En la siguiente figura se muestra la Topología del Web Service. Ver figura 2.1.2.



Figura 2.1.2. Topología de un sistema basado en Web Services.

La interacción entre los tres roles queda descrita de la siguiente manera:

1. Describas mediante un lenguaje.
2. Se publica en algún registro.
3. Clientes buscan servicios en registros.
4. Una vez encontrados éstos se ligan.

Ventajas de un Web Services.

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Omnipresencia a los servicios Web puesto que se comunican utilizando HTTP y XML, por lo que cualquier dispositivo que trabaje con estas tecnologías puede tanto ser huésped y acceder a los servicios Web.
- Barrera mínima de participación ya que se ofrecen “Herramientas de Desarrollo” (ToolKits) que permiten a los desarrolladores crear e implementar rápidamente servicios de Web.



- Permiten que los servicios y software de diferentes compañías puedan ser combinados fácilmente para proveer servicios integrados.
- Permiten una invocación sencilla de métodos, mediante SOAP.
- Transparencia para el usuario final.
- Puede ser implementado con herramientas Open Source.
- Grandes empresas lo apoyan como lo es caso de: Microsoft, Sun, Oracle, IBM, Apache.

Desventajas de un Web Services.

- Para realizar transacciones no pueden compararse en su grado de desarrollo con los estándares abiertos de computación distribuida como **CORBA (Common Object Request Broker Architecture)**.
- Su rendimiento es bajo si se compara con otros modelos de computación distribuida, tales como **RMI (Remote Method Invocation)**, **CORBA** o **DCOM (Distributed Component Object Model)**. Es uno de los inconvenientes derivados de adoptar un formato basado en texto.
- Al trabajar con XML no se encuentra la concisión ni la eficacia de procesamiento.
- Al apoyarse en HTTP, pueden esquivar medidas de seguridad basadas en **Firewall (Corta Fuegos)** cuyas reglas tratan de bloquear o auditar la comunicación entre programas a ambos lados de la barrera.
- Dependencia de la disponibilidad de servidores y comunicaciones

2.2 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE MS-SQL SERVER

El Structured Query Language (Lenguaje de consulta estructurado) o comúnmente conocido como SQL, es un lenguaje que sirve para realizar las actividades de definición, manipulación y control de una base de datos relacional. Es un lenguaje de tipo declarativo de alto nivel, es decir, se debe de indicar que es lo que se quiere hacer, y que debido a la gran base teórica que maneja y a su orientación al manejo de registros permite una alta codificación y la orientación a objetos. SQL es el lenguaje



estándar utilizado en la mayoría de los motores de bases de datos ya que permite fácilmente realizar consultas y actualizaciones de la información de manera sencilla haciendo uso de álgebra y cálculo relacional.

La historia de SQL se remonta a inicios de la década de 1970 cuando un grupo de investigación de IBM, encabezado por Donald D. Chamberlin, trabajó en la implementación de un lenguaje con el cual pudieran interactuar con un sistema gestor de base de datos relacional el cual sería muy parecido al lenguaje natural (basado en el idioma inglés), fue así como nació SEQUEL (Structured English Query Language), el cual más tarde, por problemas legales, fue renombrado como SQL.

En la década de 1980 debido a la gran aceptación que presentaba SQL y la necesidad de adoptar un lenguaje estándar para la interacción con bases de datos relacionales, fue adoptado como lenguaje oficial por la ANSI (American National Standards Institute) y por ISO (International Standards Organization), las cuales le hicieron algunas revisiones y modificaciones para finalmente dejarlo en la versión conocida como SQL2 o SQL92, la cual ha servido como base para las mejoras continuas del lenguaje.

SQL se compone de:

- Comandos. Sentencias que sirven para interactuar con la base de datos, se divide en 2 grupos.
 - DLL (Lenguaje de definición de datos). Son comandos que permiten crear, borrar y modificar los esquemas de relación (Bases de datos, tablas, campos e índices). Entre ellos se encuentran las sentencias CREATE, DROP y ALTER.
 - DML (Lenguaje de manipulación de datos). Son comandos que permiten generar consultas basadas en álgebra y cálculo relacional, las cuales sirven



para ordenar, filtrar y extraer datos. SELECT, INSERT, UPDATE y DELETE, son algunas de las consultas que pertenecen a este grupo.

- Cláusulas. Condiciones de modificación que son utilizadas para definir los datos que van a ser seleccionados o modificados. A este grupo pertenecen las sentencias FROM, WHERE y ORDER BY.
- Operadores Lógicos. Evalúan dos condiciones y devuelve un valor de verdad o falsedad, dependiendo de la expresión. Aquí encontramos las instrucciones AND, OR, NOT.
- Operadores de comparación. Instrucciones que sirven para comparar 2 valores. En ellas están incluidas los operadores <, >, <>, <=, >=, =, BETWEEN, LIKE, IN.
- Funciones de agregado. Instrucciones que son utilizadas dentro de la cláusula SELECT con el fin de devolver un valor único dentro de un grupo de datos. Un ejemplo de las instrucciones que pertenecen a este grupo son AVG, COUNT, SUM, MAX, MIN.

A pesar de la robustez de SQL, actualmente se requieren de herramientas adicionales para el manejo y manipulación de los datos, herramientas que son proporcionadas por los sistemas gestores de bases de datos, los cuales presentan características diversas entre sí, por lo que ahora hablaremos de MS-SQL Server, al ser el sistema que se va a utilizar en este proyecto.

MS-SQL Server es un gestor de base de datos basado en el modelo relacional distribuido por Microsoft, y su función principal es la de almacenar y consultar información generada por otras aplicaciones, sin importar si son consultadas de manera local, a través de una red interna o por internet. MS-SQL Server surge, en su versión 1.0, como resultado de la generación de un sistema gestor de base de datos para el sistema operativo OS/2 de IBM, en la cual participaron la propia Microsoft, IBM



y Sybase en el año de 1989, al día de hoy la versión estable más actual es la versión 2012.



Figura 2.2.1 MS SQL Server

Características de MS-SQL

Las principales características de MS-SQL Server son:

- Uso del modelo relaciona de base de datos.
- Uso del lenguaje SQL
- Uso del lenguaje T-SQL
- Manejo de transacciones.
- Manejo de procedimientos almacenados.
- Manejo de triggers.
- Manejo del protocolo cliente servidor.
- Acceso desde línea de comandos o Power Shell (CMDSQL)
- Uso de interfaz gráfica
- Soporte para .NET



- Funciones para la replicación de datos.
- Permite la administración de otros servidores de datos.
- Cuenta con una tecnología bastante robusta.
- Está orientada al manejo de datos de grandes empresas.

Las ventajas que se encuentran en el uso de MS-SQL Server son:

Desventajas de MS-SQL

Como desventajas de MS-SQL Server podemos encontrar:

- Versión gratuita tiene una capacidad muy limitada.
- Alto costo de licencias.
- Para cambiar de versión, es necesario comprar una nueva licencia.
- Se requiere una licencia por cada servidor a utilizar e incluso por cada núcleo a utilizar (dependiendo de la versión adquirida).
- Únicamente puede ser instalado en sistemas operativos Windows.
- Las versiones más modernas no funcionan en sistemas operativos antiguos.
- Utiliza mucho espacio en disco.
- Con una mala administración, utiliza mucha memoria del sistema (puede llegar a hacer uso completo de ella).
- No hay soporte para versiones antiguas.

2.3 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE ITIL.

ITIL son las siglas de Information Technology Infrastructure Library (Biblioteca de Infraestructura de Tecnologías de la Información), la cual contiene un conjunto de mejores prácticas para la gestión de servicios de TI (Tecnologías de la Información) cuyo objetivo principal es apoyar a las organizaciones proveedoras de TI a operar de una forma más eficiente y con mayor calidad.



Cabe destacar que ITIL no es una metodología, tampoco es una receta de cocina la cual deba ser seguida paso a paso para lograr un objetivo, es un conjunto de guías en las cuales las organizaciones se deben de apoyar para alcanzar sus objetivos planteados, ésta es una de sus principales características, la flexibilidad que posee, es tanta que los procesos que documenta pueden ser utilizados incluso para servicios fuera de TI.

ITIL tiene sus inicios a finales de los años 80 con la CCTA, Central Computing and Telecommunications Agency (Agencia de Computo Central y Telecomunicaciones), una agencia del gobierno británico cuyo objetivo fue el de mejorar la entrega de servicios de TI y al mismo tiempo reducir los costos que éstos generaban, al proyecto se le conoció inicialmente como GITIMM, Government Information Technology Infrastructure Management Method (Método del Gobierno para el Manejo de la Infraestructura de las Tecnologías de la Información) y estuvieron involucradas varias casas consultoras para el desarrollo, investigación y documentación de las mejores prácticas que llevarán a la realización de los objetivos planeados. Conforme el proyecto fue evolucionando y la administración gubernamental cambiando se decidió cambiar el nombre a ITIL para remarcar que lo que se pretendía buscar eran buenas prácticas y no una metodología como tal además de que estas buenas prácticas fueran aplicadas en cualquier entorno y no únicamente en el gobierno.

Desde sus primeros años de vida, la documentación que fue generando el proyecto ha sido de carácter público ya que son muchas las organizaciones que han sido partícipes en la construcción de ITIL y nadie puede adjudicarse los derechos sobre todo el conjunto de información.

La primera versión de ITIL fue publicada en 1989 y estaba conformada por 10 libros principales que se centraban principalmente en dos temas: Soporte al servicio y Entrega del servicio, además contaba con algunos otros libros secundarios que tocaban temas fuera de la gestión de servicios de TI.



Fue durante los años 90 cuando ITIL comenzó a ganar popularidad entre las organizaciones prestadoras de servicios de TI y fue adoptada como una poderosa herramienta en el manejo de servicios de TI, esta adopción permitió la creación de estándares como el ISO/IEC 20000, estándar que certifica a una organización en la correcta adopción de los procesos de ITIL ya que la certificación ITIL es únicamente para personas, no para organizaciones.

ITIL v2 fue publicado en 2001, esta versión consistía de 7 libros: Soporte al servicio, Entrega del servicio, Administración de la seguridad, Administración de la infraestructura, Administración de las aplicaciones, La perspectiva del negocio y Planeación para implantar la administración de servicios.

La versión 3 de ITIL fue sacada a la luz en 2007, ésta consistía de cinco libros: Estrategia de servicios, Diseño de servicios, Transición de servicios, Operación de servicios y Mejora continua de servicios, esta nueva organización estuvo enfocada al ciclo de vida de los servicios de TI para su mejor administración, se dejan un poco de lado los procesos para enfocarse más a lo que tiene mayor relevancia, los servicios de TI.

En el año 2011 la versión 3 de ITIL tuvo una actualización en la cual se cubrieron algunos huecos que esta versión tenía, se resolvieron ambigüedades, se aclaran algunos conceptos y además se incluyen algunos ejemplos con la finalidad de orientar de una forma práctica a las personas que consulten la documentación.

En el año 2013 la empresa privada AXELOS Limited compró al gobierno británico los derechos sobre el portafolio BMP, Best Management Practice (Mejores Prácticas de Gestión), entre el contenido de este portafolio se encontraba el manual de ITIL, por lo que el contenido pasó a ser gestionado por una empresa de carácter privado. Aún se desconoce el futuro de ITIL ahora en manos de AXELOS, lo único que han mencionado es que habrá un mayor control sobre la calidad de los Institutos Examinadores (IE)



además de tener en consideración un aumento a los honorarios a pagar, lo que aumentaría el costo de las certificaciones para el usuario final.

Características de ITIL.

- Marco de trabajo libre.
Puede ser adoptado y adaptado por cualquier organización.
- Basado en las mejores prácticas.
Toma procesos que han dado buenos resultados en distintos contextos y que en situaciones similares podrían ayudar a mejorar el desempeño y la entrega de los servicios.
- Independiente de la industria o la tecnología.
Las buenas prácticas de ITIL han dado buen resultado gracias a que no depende de la industria, ni de la tecnología, en cambio, se pueden adaptar fácilmente a los cambios de las mismas.
- Garantiza niveles de servicio.
Debido al enfoque en la entrega de servicios ITIL permite crear distintos niveles de servicio de acuerdo a la complejidad de cada uno y a la exigencia de cada cliente.
- Compuesto por cinco libros.
Dentro de los libros se describen los procesos recomendados para la administración de los servicios.
- Enfoque en la prestación de servicios.
Se enfoca a alinear las necesidades del negocio con los elementos de TI para brindar servicios de manera óptima y eficiente.



Ventajas de ITIL.

- Metodología reconocida a nivel internacional.
Adoptada e implantada por diversas instituciones a nivel internacional.
- Guía constante.
Provee lineamientos para diseñar, desarrollar e implementar la administración del ciclo de vida de un servicio.
- Servicios detallados.
Los servicios requeridos se describen y se detallan de una mejor manera ya que se utiliza un lenguaje entendible para los clientes y usuarios.
- Visión amplia.
Provee un resumen de todos los elementos que componen un servicio, así como la repercusión de algún cambio en ellos.
- Acercamiento con el negocio.
Los servicios son creados a partir de los objetivos perseguidos por la organización.
- Control total de los servicios.
Al llevar un inventario organizado de los servicios activos, retirados y los que están en fase de desarrollo.
- Facilidad en el manejo de cambios.
Debido al control que se tiene sobre los elementos de configuración de los servicios las modificaciones tienden a ser más transparentes para clientes y usuarios.



- Manejo de proveedores.
Se provee una mejora en el trato con los proveedores al establecer contratos entre ambos.
- Niveles de servicio.
Se crean niveles de servicio o SLA's (Service Level Agreements) que ayudan a establecer la calidad del servicio provisto a clientes y usuarios de acuerdo con los objetivos del negocio.
- Mejora continua.
Permite la identificación de mejoras en puntos clave para alinear los objetivos de negocio y dar mayor valor a los clientes en forma de servicios.
- Eficiencia financiera.
El uso de los recursos financieros se centra en el diseño y la mejora continua y se enfocan a los objetivos establecidos en la estrategia del servicio evitando gastos innecesarios.
- Cambio del entorno cultural.
En una correcta adopción de ITIL las áreas involucradas y los participantes se adaptarán a las mejores prácticas.
- Valor a los servicios.
Se crea valor en forma de servicios para clientes y usuarios de acuerdo a la utilidad y garantía que les provee.

Desventajas de ITIL.

- Dificultad para medición.
Debido a que no siempre se definen las métricas correctas.



- Tiempo y esfuerzo requerido.
El tiempo necesario para la implantación de ITIL puede ser alto dependiendo del grado requerido en la organización.
- Resistencia al cambio cultural.
ITIL requiere un cambio cultural en la organización, si el personal no está familiarizado o debidamente capacitado difícilmente se podrá adoptar esta metodología.
- Debido a la falta de entendimiento sobre los nuevos procesos, sus controles e indicadores el personal y las áreas involucradas podrían no observar una mejoría en los servicios.
- Al inicio de la implantación de la metodología, se necesita una inversión en herramientas para la automatización de procesos ya que sin éstas posiblemente no se logren las metas esperadas y no se alcancen las mejoras en la prestación de los servicios.

2.4 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE VISUAL STUDIO.

Visual Studio es un entorno de desarrollo integrado IDE (Integrated Development Environment) desarrollado para sistemas operativos Windows que soporta diversos lenguajes de programación como son C#, C++, .NET, JAVA, PHYTON, Ruby, PHP y ASP.NET para el desarrollo de páginas web y web services.

Visual Studio permite a los desarrolladores crear aplicaciones web, aplicaciones de escritorio y aplicaciones móviles que pueden ser ejecutados en infinidad de sistemas compatibles.

Visual Studio nace en el año 1997 y fue el primer intento que hizo Microsoft por



consolidar un entorno de desarrollo bajo un solo paquete llamado 'Developer studio' las principales características que presento esta versión fue un entorno que permitía la creación de páginas web y el manejo del lenguaje J++.

En años posteriores Visual Studio mejoró con diversas modificaciones y se le adicionaron e incluyeron propiedades nuevas que tuvieron como finalidad apoyar a los desarrolladores a mejorar la creación de sus proyectos y al mismo tiempo crear un entorno sólido, los cambios sobresalientes que sufrió a lo largo de las actualizaciones son:

- Actualizaciones constantes a .net que permitía la integración de las aplicaciones desarrolladas con el sistema operativo.
- Soporte a lenguajes de programación distintos como c#, J++, Java, Python entre otros.
- Facilidad para creación de librerías Dll's que permitían que los archivos ejecutables tuvieran un peso menor y que al mismo tiempo se compartieran con desarrollos distintos, estas dll's son la evolución de las bibliotecas estáticas.
- Integración de editores para código html y css.
- Soporte para diversas bases de datos existentes en el mercado.

El logotipo de Visual Studio es el que se muestra en la figura 2.4.1.



Figura 2.4.1. Logo Visual Studio



Características de Visual Studio 2012.

Las principales características de esta versión son las siguientes:

- Soporte para arquitectura WINRT. Con la nueva versión se permite la creación de aplicaciones para entornos Windows 8 y Windows Phone.
- Colorización de código. Permite la identificación de clases, estructuras, variables etc. Mediante el uso de colores.
- Explorador de soluciones. Permite la visualización total del proyecto como jerarquías y archivos que lo comprenden.
- Herramientas de testing. Permiten la comprobación de cambios en tiempo real previo a la compilación y cierre del proyecto.
- Desarrollo de juegos y aplicaciones en 3D.
- Integración con system center. Enfocado a la administración de servidores.
- Desarrollo de aplicaciones en Windows Azure. Permite que el desarrollo de aplicaciones aprovechando al máximo las ventajas que nos ofrece la nube.
- Plantillas para edición web. Contiene plantillas para el manejo de código HTML5 y CSS3.
- Incorporación de .net framework 4.5.

Ventajas Visual Studio 2012.

- Provee una completa solución para desarrollo en diversos lenguajes de programación.
- Permite la creación rápida y a un menor costo de aplicaciones basadas en Windows.
- Permite la generación de librerías DLL's. Estas bibliotecas ayudan a los desarrolladores a utilizar código incluido en ellas para distintos proyectos.
- Permite la reutilización de código para futuros proyectos como son clases y dll's.
- El soporte y el conocimiento sobre Visual Studio es amplio en la red y es fácil encontrar información y guías para nuevos desarrollos.



- Conexión con bases de datos. Se incluyen diversos conectores para ligar el desarrollo con las bases y ampliar el desarrollo del proyecto.

Desventajas Visual Studio 2012.

- Costo de licenciamiento desde 15,000 hasta 160,000 pesos.
- Software utilizable solamente bajo sistemas operativos Windows.

2.5 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE PERL

El lenguaje de programación Perl tiene sus orígenes en el año de 1987, fecha en que fue publicado, en su versión 1.0, por el ahora célebre programador norteamericano Larry Wall. Desde su aparición fue ampliamente aceptado, utilizado y mejorado por un gran sector de la comunidad de programadores, al grado de que en el año de 1988 ya contaba con una versión 2.0 y para 1989 ya existía la versión 3.0, en cada versión publicada fueron aumentando sus capacidades y fueron incluyendo nuevas funcionalidades, este lenguaje de programación pasó por un proceso de maduración relativamente rápido.

Al atravesar un crecimiento acelerado, el lenguaje de programación Perl, tenía varias deficiencias, una de las principales fue la falta de documentación clara, sin embargo esto cambió en 1991 cuando Perl fue publicado en su versión 4.0 en donde el cambio principal fue el hecho de que ya existía un libro publicado por la editorial norteamericana O'Reilly Media, escrito por el consultor de programación, administrador de sistemas y autor Randal L. Schwartz en conjunto con el propio Larry Wall, en este libro titulado "Programming Perl" (Programando Perl), se describía cómo escribir programas utilizando el lenguaje de programación Perl.

Fue este mismo libro quien aportó el símbolo característico de Perl, un camello, ya que en su portada aparece la imagen de dicho animal, este documento también es



conocido en el mundo de los programadores como “el libro del camello” ya que su portada ha permanecido durante las cuatro ediciones que han sido publicadas. Los derechos sobre el logo pertenecen a la editorial O’Reilly, sin embargo ésta permite su uso sin fines de lucro a la fundación Perl.

En la figura 2.5.1 se muestra el símbolo oficial del lenguaje de programación Perl.



Figura 2.5.1 Logo Perl

En 1993 Larry Wall dejó de trabajar en Perl 4 dejándolo en la versión 4.036 y comenzó a escribir una nueva versión del lenguaje de programación que sería una reescritura completa de todo el código del lenguaje, que incluiría nuevas mejoras como la programación basada en objetos y la posibilidad de utilizar módulos que incluyeran funcionalidades adicionales, esta nueva versión de Perl, 5.0, fue publicada en octubre de 1994 y es la versión que actualmente sigue siendo soportada estando en su revisión 5.18.2.

En 1995 fue puesto en línea el CPAN, Comprehensive Perl Archive Network (Red Integral de Archivos Perl) la cual es una colección de módulos, scripts y documentación de Perl para que puedan ser utilizados por toda la comunidad, actualmente cuenta con más de 25 mil elementos publicados y tiene alrededor de 270 espejos por todo el globo.

En el año 2000 se comenzó a trabajar en la versión 6.0 de Perl, esta edición aún se encuentra bajo desarrollo y promete tener muchas implementaciones, se pretende que



esta versión sea compatible con Perl 5.X, sin embargo no es uno de los objetivos del proyecto, Larry Wall aún se encuentra involucrado con el desarrollo del mismo.

En varias publicaciones se dice que el lenguaje de programación Perl fue nombrado así al ser el acrónimo de “Practical Extraction and Report Language” (Lenguaje Práctico de Extracción y Reportes), sin embargo ésta fue una interpretación que se le dio después de su publicación, inicialmente el lenguaje iba a ser nombrado Pearl (Perla en inglés), sin embargo Larry Wall se percató de que ya existía en ese momento un lenguaje de programación llamado de esa forma, así que lo único que hizo fue quitarle la letra “a” y con ese nombre, Perl, fue publicado. En la sección FAQ, Frequently Asked Questions (Preguntas Usualmente Hechas), de la página oficial de Perl se menciona que nunca se debe hacer referencia al lenguaje como PERL ya que éste no es un acrónimo.

Características de Perl

Se enlistan las características principales del lenguaje de programación Perl:

- Lo mejor de otros lenguajes
Perl toma sus principales características de otros lenguajes de programación, por ejemplo toma de C su estructura, del Shell / Bash Script su sintaxis, de AWK su memoria asociativa o hash, de sed sus expresiones regulares y de Lisp su manejo de listas y los unifica en un único lenguaje.
- Lenguaje interpretado
Perl es un lenguaje interpretado, es decir, se ejecuta a través de un intérprete el cual lee el código del programa cada vez que se pretende correr el programa.
- Scripts
El código de Perl que cumple con cierta función es conocido como Script de Perl, no como programa, esto debido a la característica interpretativa del lenguaje.



- Lenguaje case-sensitive
Al estar sus orígenes en el sistema operativo Unix, el lenguaje hace diferencia entre mayúsculas y minúsculas.
- Uso de módulos
Perl permite el empaquetado de código en módulos para su posterior reutilización, estos módulos pueden ser compartidos a través de la CPAN.
- Ejecución desde línea de comandos
Un script de Perl se ejecuta desde una terminal de línea de comandos del sistema operativo en el que se pretenda correr.
- La comunidad de Perl
Perl cuenta con una gran comunidad alrededor del globo que ofrece, orientación y soporte de forma gratuita en el uso del lenguaje y creación de aplicaciones.
- La CPAN
La Comprehensive Perl Archive Network (Red Integral de Archivos Perl) es un repositorio que contiene más de 25 mil módulos de Perl que también son de código abierto y están disponibles para todo el mundo.
- La fundación Perl
El lenguaje de programación Perl tiene su propia fundación sin ánimos de lucro, se encuentra encargada de las cuestiones legales del lenguaje, de su crecimiento, así como de coordinar el esfuerzo de varias organizaciones de Perl, subsiste a base de donaciones.

Ventajas de Perl

Se enlistan algunas ventajas de Perl sobre otros lenguajes de programación:



- Es multiplataforma
Perl se encuentra presente e instalado de forma nativa en los sistemas operativos Unix y Macintosh, además de que puede ser instalado fácilmente en entornos Windows.
- Es un software de código abierto.
Perl es distribuido con una licencia aún más abierta que la propia GNU GPL, GNU General Public Licence (Licencia Publica General GNU), lo que garantiza el mantenimiento y desarrollo por una gran comunidad de usuarios.
- Es multiparadigma
Soporta los paradigmas de programación imperativo, funcional y orientado a objetos y no obliga a usar alguno en específico al momento de generar código.
- Su flexibilidad
Perl es un lenguaje fácil de aprender además de que permite realizar cosas complicadas de manera sencilla con ayuda de todos los módulos que posee.
- Amplia documentación
Aquella que en el pasado fue su desventaja es ahora uno de sus fuertes, Perl ofrece en su página oficial una gran biblioteca en varios idiomas en donde se documenta el uso del lenguaje, de sus módulos e incluso se ofrecen tutoriales para el aprendizaje.
- Fácil manejo de archivos
Perl permite la escritura de programas pequeños y sencillos que pueden ser utilizados para realizar búsquedas y extraer información concreta de archivos de texto de gran tamaño, esto debido a su potencia en el manejo de cadenas.



Desventajas de Perl

Se enlistan algunas desventajas de Perl sobre otros lenguajes de programación:

- **Tiempo de ejecución elevado**
Al ser un lenguaje interpretado, un programa en Perl es compilado cada vez que se ejecuta, lo cual incrementa el uso de recursos de cómputo y eleva el tiempo de la entrega de resultados, sin embargo cuenta con algunas alternativas para evitar la constante compilación.
- **No es un lenguaje ordenado**
Uno de los propósitos de Perl es que el programador pueda alcanzar su objetivo de la manera más fácil por lo que permite la instalación de cualquier módulo, se pueden crear excepciones a algunas reglas además de que utiliza la heurística para resolver conflictos en la sintaxis, lo que puede dificultar la búsqueda de errores y el correcto funcionamiento al ser difundido.
- **Ambiente poco gráfico**
A pesar de que existen interfaces IDE, Integrated Development Environment (Ambiente de Desarrollo Integrado), es considerado como un lenguaje de programación poco gráfico por lo que su uso puede llegar a dificultarse para los programadores que estén acostumbrados a ambientes más gráficos.

2.6 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE LA ARQUITECTURA CLIENTE / SERVIDOR.

En el mundo de TCP/IP las comunicaciones entre computadoras se rigen básicamente por lo que se llama arquitectura Cliente/Servidor, éste es un modelo que intenta proveer usabilidad, flexibilidad, inter-operabilidad y escalabilidad en las comunicaciones.



La arquitectura Cliente/Servidor es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos.

En la arquitectura Cliente/Servidor, el cliente envía un mensaje solicitando un determinado servicio a un servidor (hace una petición), y éste envía uno o varios mensajes con la respuesta (provee el servicio).

Características de la arquitectura Cliente/Servidor

Los principales componentes del esquema Cliente/Servidor son: los Clientes, los Servidores y la infraestructura de comunicaciones.

En este modelo, las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario.

Cliente

Los clientes interactúan con el usuario, usualmente en forma gráfica. Frecuentemente se comunican con procesos auxiliares que se encargan de establecer conexión con el servidor, enviar el pedido, recibir la respuesta, manejar las fallas y realizar actividades de sincronización y de seguridad.

Las funciones que lleva a cabo el proceso cliente se resumen en los siguientes puntos:

- Administrar la interfaz de usuario.
- Interactuar con el usuario.
- Procesar la lógica de la aplicación y hacer validaciones locales.
- Generar requerimientos de bases de datos.
- Recibir resultados del servidor.
- Formatear resultados.



- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.

Servidor

Es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él.

El servidor normalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos.

Las funciones que lleva a cabo el proceso servidor se resumen en los siguientes puntos:

- Aceptar los requerimientos de bases de datos que hacen los clientes.
- Procesar requerimientos de bases de datos.
- Formatear datos para transmitirlos a los clientes.
- Procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos.

Ventajas la arquitectura Cliente/Servidor

- Existen plataformas de hardware cada vez más baratas. Se pueden utilizar componentes de varios fabricantes, tanto de hardware como de software, esto favorece la flexibilidad en la implementación y actualización de soluciones.
- Facilita la integración entre sistemas diferentes, Por lo que podemos integrar PCs con sistemas medianos y grandes, sin necesidad de que todos tengan que utilizar el mismo sistema operativo.
- El uso de interfaces gráficas bajo este esquema tienen mayor interacción y más intuitiva con el usuario. Esto presenta la ventaja, con respecto a uno centralizado, de que no es siempre necesario transmitir información gráfica por la red pues esta puede residir en el cliente, lo cual permite aprovechar mejor el ancho de banda de la red.



- Los servidores controlan los datos, procesan peticiones y después transfieren sólo los datos requeridos a la máquina cliente. Todo esto reduce el tráfico de la red, lo que facilita que pueda soportar un mayor número de usuarios.
- Tanto el cliente como el servidor pueden escalar para ajustarse a las necesidades de las aplicaciones.
- En una arquitectura como ésta, los clientes y los servidores son independientes los unos de los otros, con lo que pueden renovarse para aumentar sus funciones y capacidad de forma independiente, sin afectar al resto del sistema.
- Permite el uso de ordenadores especializados (Servidores de BD, servidores de aplicaciones, servidores web, etc)

Desventajas de la arquitectura Cliente/Servidor

- Dificulta el diagnóstico de fallas debido a la interacción del software y hardware de diferentes proveedores.
- Los clientes y los servidores, deben utilizar el mismo mecanismo, deben existir mecanismos generales que existan en diferentes plataformas.
- Los problemas de congestión de la red pueden degradar el rendimiento del sistema por debajo de lo que se obtendría con una única máquina.
- Se pueden presentar problemas de seguridad como: control de accesos, confidencialidad, integridad, etc
- El desempeño es otro de los aspectos que se deben tener en cuenta en el esquema Cliente/Servidor.
- La complejidad y el esfuerzo requerido para administrar y soportar un ambiente Cliente/Servidor. Hay pocas herramientas reconocidas que soportan manejo de configuraciones, monitoreo del rendimiento y distribución de versiones de software.



Arquitectura de 1 capa

Entendemos por aplicaciones mono-capa, aquellas que tanto la propia aplicación como los datos que maneja se encuentran en la misma máquina y son administradas por la misma herramienta: podríamos decir que son una sola entidad.

Arquitectura de 2 capas

La arquitectura cliente/servidor tradicional es una solución de 2 capas. Consta de tres componentes distribuidos en dos capas: cliente (solicitante de servicios) y servidor (proveedor de servicios). Los tres componentes son:

- Interfaz de usuario.
- Gestión del procesamiento.
- Gestión de la base de datos.

Arquitectura de 3 capas

La tercera capa está entre la interfaz de cliente y el servidor. La capa intermedia proporciona gestión del procesamiento y en ella se ejecutan las reglas y lógica de procesamiento. Permite cientos de usuarios. La arquitectura de 3 capas es usada cuando se necesita un diseño cliente/servidor que proporcione, en comparación con la arquitectura de 2 capas, incrementar el rendimiento, flexibilidad, mantenibilidad, reusabilidad y escalabilidad mientras se esconde la complejidad del procesamiento distribuido al usuario.

Arquitectura de n capas

Esta añade nuevas capas, La lógica de aplicación se reparte en diferentes capas/niveles ubicadas entre el cliente y los datos.



Las capas intermedias se proporcionan servicios entre sí. Cada nivel se comunica sólo con los niveles contiguos a través de interfaces bien definidos. El funcionamiento de esta capa se define de la siguiente manera: capa k ofrece servicios a capa $k + 1$ y demanda servicios de capa $k - 1$.

Arquitectura distribuida

En un sistema distribuido cada máquina puede cumplir el rol de servidor para algunas tareas y el rol de cliente para otras.

Esta arquitectura le asigna al cliente toda la funcionalidad: la presentación y la lógica, a excepción de los datos, los cuales son repartidos entre el cliente y el servidor; esto permite una ubicación óptima de los mismos.

Capítulo III

Estrategia del servicio



Desde la perspectiva de ITIL la estrategia del servicio nos va a guiar en el proceso de la gestión del ciclo de vida de los servicios con la finalidad alinearlos a las necesidades del negocio creando valor a partir de ellos, priorizando objetivos, oportunidades y dando un uso adecuado a los activos financieros para establecer un mayor retorno de inversión.

En este capítulo se analizará el problema que se quiere resolver, los datos con los que contamos para comenzar con el proyecto, los requerimientos solicitados por la empresa y se propondrá una solución apegándonos a los objetivos del negocio y a las indicaciones otorgadas por la empresa, la finalidad de este capítulo es establecer una guía de referencia para el diseño, transición y operación la cual se apegará a las necesidades del negocio.

3.1 ANÁLISIS DEL PROBLEMA.

Una de las principales partes en el análisis del problema se da con la definición de los requerimientos que nos solicita tanto el SAT en materia legal y la cuestión de los parámetros establecidos para un web service, puesto que nuestro sistema se habrá de encargar de poder realizar un monitoreo en tiempo real sobre datos visuales derivados de la facturación electrónica, para poder establecer un control del comportamiento del mismo, así mismo el tener una reacción ante alguna posible eventualidad, pero sin afectar el servicio ofrecido por la empresa.

Es por ello que para poder definir cuáles son los problemas que solucionará el sistema a desarrollar, debemos considerar en principio los factores que están implícitos tanto para el Web Services, como los requisitos que debe cubrir por parte del SAT con respecto a la facturación electrónica que está establecida en el Diario Oficial de la Federación, por lo que con el siguiente diagrama (ver figura 3.1.1) observamos los pasos que debemos de tener en cuenta, así como ubicar los puntos clave que nos ayuden a encontrar la mejor manera de resolverlos.

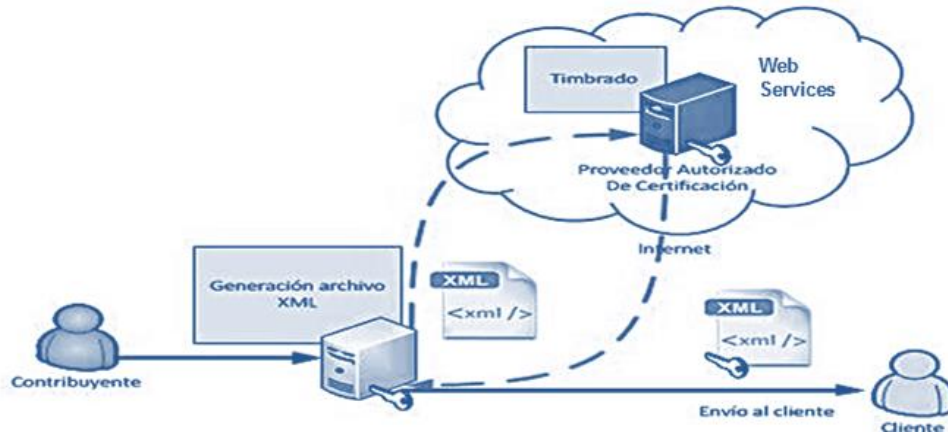


Figura 3.1.1. Diagrama de fases

Disposiciones del SAT

Por disposición del Servicio de Administración Tributaria (SAT), a partir del 1 de enero de 2014, todos los contribuyentes, sin excepción, deberán usar como modelo de facturación el Comprobante Fiscal Digital por Internet (CFDI). Esto significa que, sin importar el giro, los ingresos anuales, el tamaño o régimen fiscal al que pertenezca, los causantes ahora deberán migrar a ese nuevo esquema y dejar de usar el Comprobante Fiscal Digital (CFD), los Comprobantes con Código de Barras Bidimensional (CBB) y los recibos impresos, pues éstos ya no tendrán validez.

Los requerimientos administrativos que el SAT define para poder iniciar con la operación del Comprobante Fiscal Digital son: (Ver figura 3.1.2)

- Obtener la Firma Electrónica Avanzada (FIEL).
- Obtener y controlar los Sellos Digitales, Folios y Series.
- Tener un Sistema Contable Electrónico Automatizado.
- Tener un registro Contable de la Venta simultáneo.
- Contar con una aplicación que permita la generación, impresión, resguardo y transmisión de los Comprobantes Fiscales Digitales (CFD) al SAT.
- Generar un Reporte Mensual de Folios emitidos para el SAT.



Firma Electrónica Avanzada

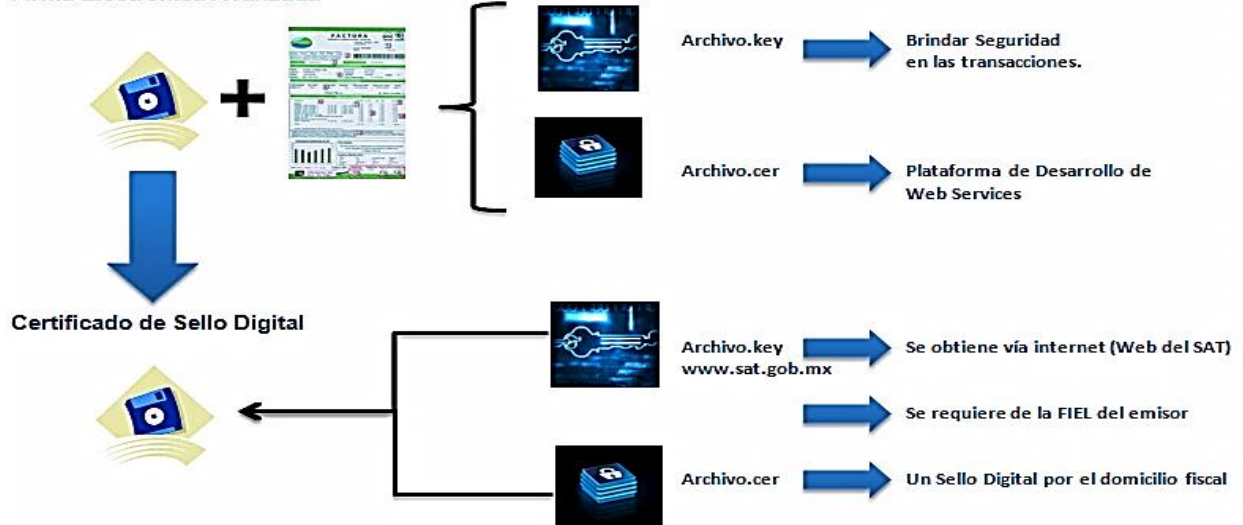


Figura 3.1.2. Elementos comprobantes fiscal digital.

- Resguardar los CFD's (Facturas, notas de crédito, notas de débito).
- Manejar el formato XML estipulado por el SAT en el anexo 20 de la Resolución Miscelánea Fiscal. Ver figura 3.1.3

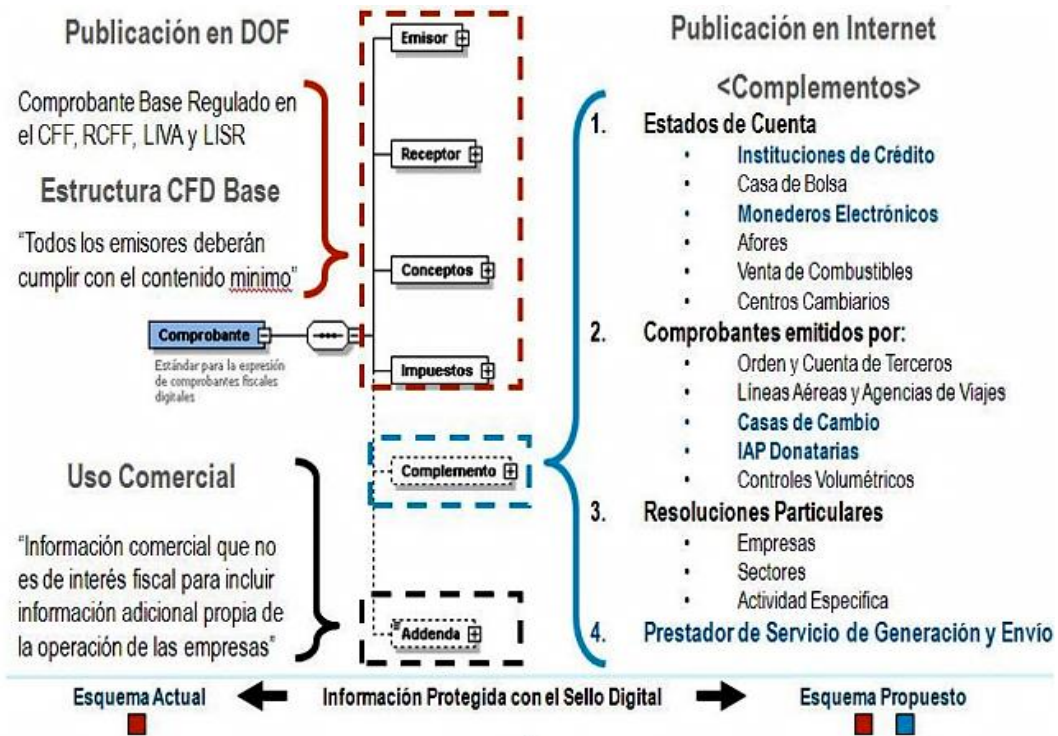


Figura 3.1.3. Formato XML estipulado por el SAT.



Adicionalmente define que es necesario se verifiquen los datos fiscales de los Comprobantes Fiscales Digitales que los Contribuyentes reciban de sus proveedores.

Los lineamientos de Diseño del mismo son:

- Generación electrónica y automática.
- Seguro y estructurado.
- Estándares internacionales gratuitos.
- Para cualquier contribuyente y multiplataforma.
- Neutralidad tecnológica.

La Factura Electrónica CFDI garantiza que cada comprobante expedido es único, infalsificable e inalterable. A través de elemento como la cadena original, el sello digital, la certificación y/o el timbrado realizado por el PAC (Proveedor Autorizado de Certificación de Comprobantes Fiscales Digitales a través de Internet.), el esquema CFDI permite detectar cualquier alteración que se realice sobre un comprobante e impide falsificar los mismos.

Cuando se utilizan comprobantes fiscales en papel, se corre el riesgo de recibir comprobantes apócrifos o falsos que clonan la información de un tercero para generar un documento ilegítimo.

Lineamientos del Web Services.

Los web services permiten que las aplicaciones compartan información y que además invoquen funciones de otras aplicaciones independientemente de cómo se hayan creado las aplicaciones, cual sea el sistema operativo o la plataforma en que se ejecutan y cuáles los dispositivos utilizados para obtener acceso a ellas. Aunque éstos mismos sean independientes entre sí, que es por ello que puedan vincularse y formar un grupo de colaboración para realizar una tarea determinada.



Es por ello que los servicios web se proponen como una alternativa para facilitar la intercomunicación entre diferentes arquitecturas de componentes, ofreciendo una visión de dichas arquitecturas basada en servicios.

Desde un punto de vista general, un servicio web es un componente de software independiente de la plataforma e implementación, que lleva a cabo un servicio concreto y que puede integrarse con otros servicios web para dar un servicio diferente.

Estos componentes de software podrán ser:

- Descrita usando un lenguaje de descripción de servicio.
- Publicada en un registro de servicios.
- Descubierta a través de un mecanismo estándar.
- Invocada a través de un API declarado (usualmente sobre una red)
- Compuesta con otros servicios.

Un Web Services es un sitio en la web que nos ofrece la posibilidad de realizar una o múltiples tareas a través de él, como se muestran en los siguientes esquemas:

- El cliente le envía una petición al servidor que a su vez consulta su repositorio de páginas y se la devuelve al cliente. Ver figura 3.1.4.

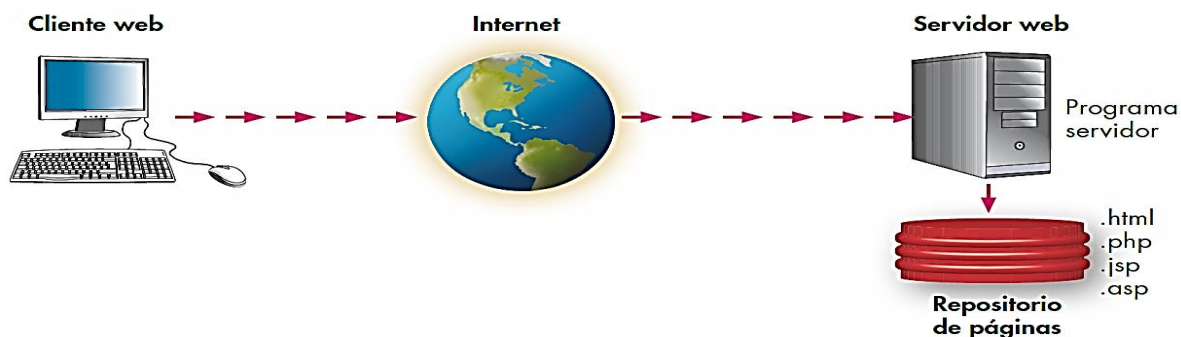


Figura 3.1.4. Esquema básico de funcionamiento del servicio



- El cliente le envía una petición de index.html al servidor que busca en su repositorio el index.html y se lo manda al cliente que muestra la página. Ver figura 3.1.5.

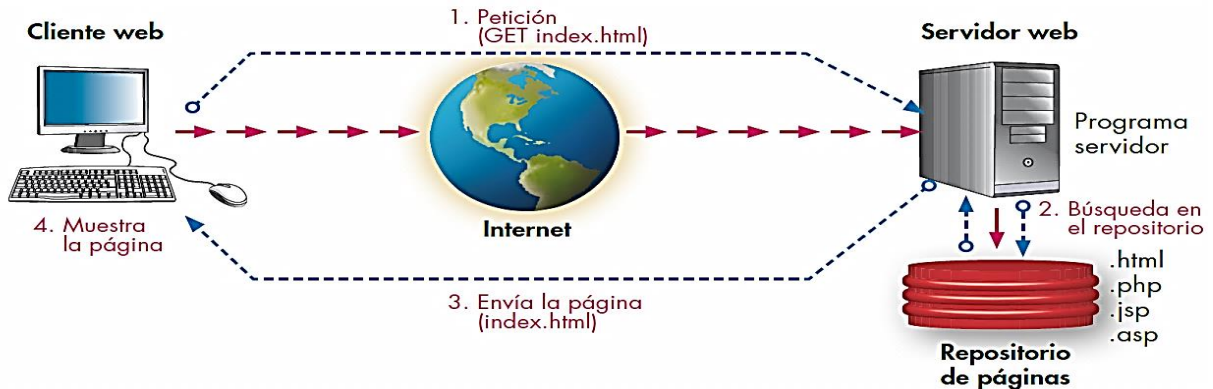


Figura 3.1.5. Esquema de funcionamiento con páginas estáticas.

- El cliente le envía una petición de index.php al servidor que a su vez éste busca en su repositorio el index.php en este momento el servidor crea a partir del código php el código html consultando las bases de datos correspondientes y después el servidor le devuelve el código html correspondiente al index.php que muestra la página. Ver figura 3.1.6

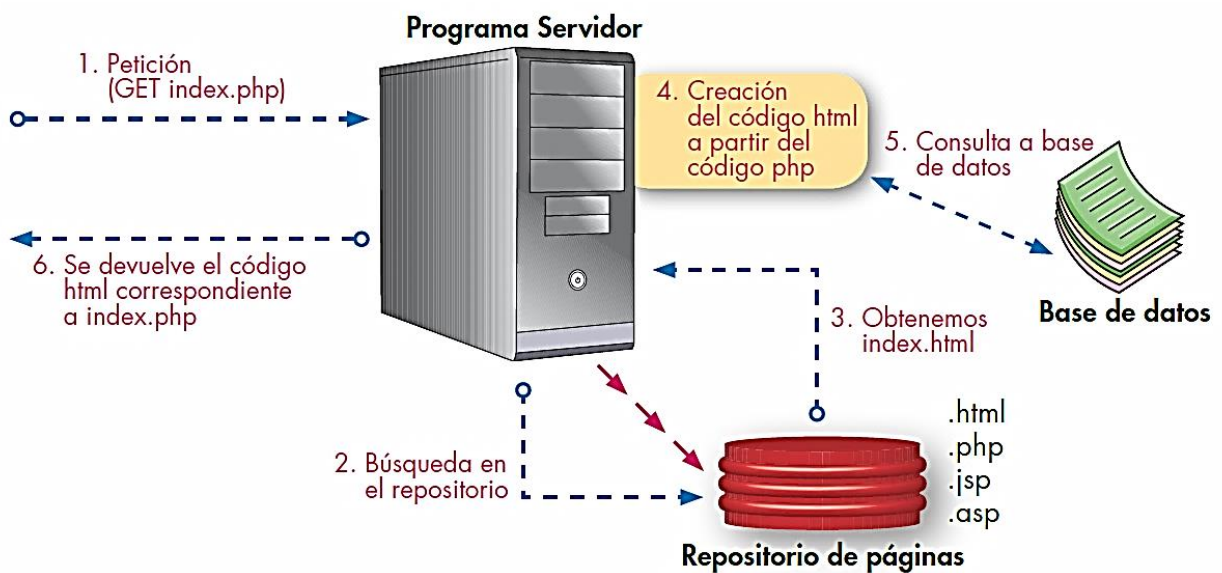


Figura 3.1.6. Esquema de funcionamiento con páginas dinámicas.



En el siguiente diagrama se muestra el esquema de cómo se implementa un Web Services. Ver figura 3.1.7

- Un proveedor de servicio crea un servicio Web
- El proveedor de servicio utiliza WSDL para describir el servicio (a un registro UDDI)
- El proveedor de servicio registra el servicio (en un registro UDDI y/o a un registro/depósito ebXML)
- Otro servicio o usuario localiza y solicita el servicio registrado al consultar los registros UDDI y/o ebXML.
- El servicio o usuario solicitante escribe una aplicación que liga el servicio registrado utilizando SOAP (en el caso de UDDI) y/o ebXML.
- Se intercambian datos y mensajes XML sobre HTTP.

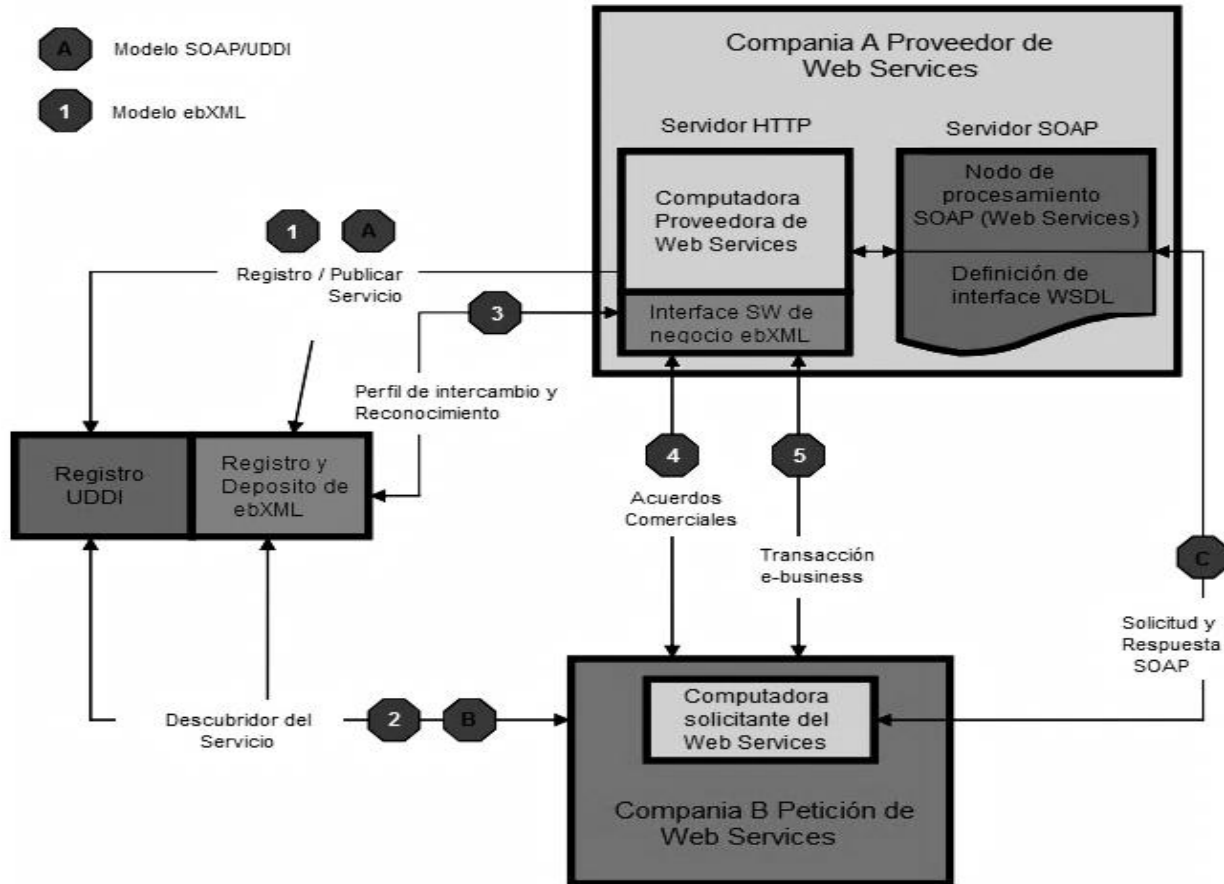


Figura 3.1.7. Esquema de Implementación Web Services.

3.2 RECOPIACIÓN Y ANÁLISIS DE DATOS

El servicio de timbrado de documentos fiscales digitales que la empresa ofrece a través de su Webservice de Timbrado es fundamental e indispensable ya que es la base de muchos otros productos que vende además de ser consumido por clientes que no pueden prescindir de él, es un servicio de alta disponibilidad en el cual se ofrece a los clientes una disponibilidad del 98%, sin embargo, no cuenta con un correcto sistema de monitoreo que regrese datos que puedan ayudar a la toma de decisiones en cuanto al Webservice.

Al ser la disponibilidad de este servicio un dato que debe ser cuidado y dado a conocer a sus clientes, el sistema de monitoreo únicamente cubre este punto haciéndolo a



través de un ping a la dirección del Webservice, este mismo sistema de monitoreo arroja una alarma en caso de que el portal no responda, lo que significaría que el Webservice está caído, el monitoreo utilizado únicamente permite actuar de forma reactiva ante los eventos que se presenten y eso comienza a ser un gran problema para la empresa dado que a raíz de las últimas reformas publicadas por el SAT, sus clientes comienzan a incrementar y el consumo del Webservice se ha incrementado de manera significativa.

En caso de que la empresa requiera conocer datos más específicos sobre el consumo del Webservice, los responsables deben ingresar directamente al servidor que contiene al servicio de Timbrado y abrir manualmente archivos de log que contienen datos como las llamadas al Webservice, las direcciones IP desde donde se realizaron, el tiempo total que duró el proceso de timbrado, el tiempo descompuesto en cada uno de sus módulos y el usuario que utilizó el Webservice.

El proceso de revisar los archivos de log de forma manual se ha vuelto complicado en fechas recientes debido a:

- La aplicación genera y utiliza un único archivo de log por día.
- El consumo del Webservice se ha incrementado en un gran porcentaje y consecuentemente los archivos de log han crecido en tamaño.
- Abrir un archivo de log de gran tamaño y realizar búsquedas de información consume bastantes recursos de cómputo, lo que llega a afectar la disponibilidad del servicio de Timbrado.

Una alternativa para obtener cierta información del Webservice es realizar una búsqueda directamente en la base de datos, sin embargo esta base también tiene un tamaño considerable y se encuentra corriendo en una instalación de Microsoft SQL Server Standard Edition, esta versión del motor de base de datos no permite que las consultas corran en paralelo sino las va encolando, lo cual afecta el tiempo de



respuesta del Webservice y en algunas ocasiones también se ve afectada la disponibilidad del servicio.

Un dato que es requerido y consumido por los clientes de la empresa es la cantidad de timbres que han utilizado, para que sus consultas no afecten el desempeño de la base de datos productiva se optó por dejar una tarea que a través de pequeñas consultas toma la información de los timbres por cliente de la base de datos productiva y la inserta en una base de datos que el cliente ya puede consultar a través de una aplicación Web en donde puede revisar la cantidad de timbres que han consumido según algunos filtros que son ofrecidos.

En los últimos meses se han presentado quejas constantes de los clientes con respecto al portal de consulta de sus timbres, el problema a corregir es que se tienen muchos días de atraso en ese servicio, este atraso es debido a que las consultas que realiza la tarea sobre la base de datos productiva son pequeñas con la finalidad de no afectar el desempeño de la misma y que la cantidad de timbres diarios ha crecido de forma alarmante, lo cual imposibilita a la tarea con la que se cuenta para que tenga la información al día.

La problemática anteriormente descrita y que tiene en este momento la empresa podría ser resuelta aprovechando los archivos de log, tanto para la obtención de datos como el retraso en la aplicación de consulta de los clientes, lo único que se tendría que cuidar es leer el archivo en partes para no consumir recursos de más y no bloquear el archivo ni la aplicación para de no afectar el rendimiento o la disponibilidad del Webservice de Timbrado.

3.3 REQUERIMIENTOS DE LA APLICACIÓN

Los requerimientos de un sistema describen los servicios que ha de ofrecer él mismo y las restricciones asociadas a su funcionamiento.



Sirve para recopilar la información necesaria para establecer la funcionalidad que se quiere alcanzar con el sistema.

Los requerimientos de software pueden dividirse en 2 categorías: requerimientos funcionales y requerimientos no funcionales. Los requerimientos funcionales son los que definen las funciones que el sistema será capaz de realizar, describen las transformaciones que el sistema realiza sobre las entradas para producir salidas. Los requerimientos no funcionales tienen que ver con características que de una u otra forma puedan limitar el sistema, como por ejemplo, el rendimiento.

El sistema por sus características y uso de tecnologías web, será probado para los siguientes navegadores. Estas versiones son:

- Microsoft Internet Explorer 8 ó 9
- Mozilla Firefox
- Google Chrome

El sitio debe incluir la siguiente frase: “AVISO DE CONFIDENCIALIDAD: Este diseño es propiedad intelectual para ser utilizado exclusivamente por esta empresa, quedando estrictamente prohibido la difusión, distribución o reproducción parcial o total.”

Para ingresar al sitio el usuario debe de poseer una cuenta creada únicamente por el administrador de la aplicación.

Requisitos específicos

Esta sección describe detalladamente los requisitos del sistema, entre éstos los funcionales y los no funcionales.

Requisitos funcionales:



- El sistema debe de tener un administrador, encargado de realizar ciertos cambios: a agregar o quitar características al portal de monitoreo. También tiene entre sus funcionalidades crear los usuarios que podrán ingresar mediante autenticación.
- El administrador podrá editar las categorías que el sistema tiene por defecto y agregar nuevos ítems a éstas.
- El portal debe contar con una página de inicio en la cual se deba de garantizar el control en el acceso, utilizando la autenticación de los usuarios para el uso del mismo, esto mediante un usuario y contraseña.
- Las pantallas principales deben contener el logotipo de la empresa.
- Se deberá accesar desde internet, por lo cual debe estar publicada.
- El sistema debe cumplir con todas las validaciones necesarias para evitar datos incorrectos en el evento.
- El usuario al ingresa al sitio mediante una autenticación de usuario, podrá seleccionar las opciones de la gráfica ya sea por un intervalo de tiempo o en tiempo real.

Requerimientos no funcionales:

- La aplicación será multiplataforma desarrollada en ASP y C#
- El motor de base de datos deberá ser MS SQL Server 2008.
- Utilizar para la creación de los Formularios Web hojas de estilos CSS.



- El Protocolo de comunicación con la aplicación WEB deberá ser HTTP.
- Se debe de tener un programa en Perl que permita insertar los datos del archivo .csv a la base de datos en SQL
- El portal web estará formado de menús principales, secundarios, botones y formularios que facilitarán el manejo por parte del usuario, ordenar los contenidos, agrupándolos en conjuntos coherentes y dándoles nombres que los identifiquen.
- Debe de ser una página dinámica en la cual se pueda elegir un intervalo de tiempo desde un mes atrás a solo media hora. En una ventana de lado inferior izquierdo deberá incluirse datos estadísticos como lo son: tiempo de respuesta máximo y mínimo del intervalo de tiempo elegido, conteo de timbres y un porcentaje de uso.
- Se debe de poder visualizar una gráfica en tiempo real de los tiempos de respuesta que ha tenido el servidor durante las últimas 3 horas.
- El sistema deberá estar disponible 24(horas)/7(días) siempre que los servidores donde se alberga estén funcionando correctamente.

3.4 PLANTEAMIENTO DE LA SOLUCIÓN.

En el desarrollo del planteamiento de la solución se hará un estudio que nos permitirá definir las herramientas necesarias para la creación del sistema, se realizarán comparativas del software que nos ayudarán a la construcción del mismo con la finalidad de identificar aquellos que se apeguen a las necesidades del proyecto, el análisis se hará desde tres perspectivas que son:



- Front-End. Parte del proyecto que interactúa directamente con el usuario.
- Back-End. Parte del proyecto que procesa los datos generados por el usuario desde el Front-End.
- Webservices. Canal de comunicación

Análisis Front-End.

Para el análisis del Front-End se estudiarán tres herramientas principales que pueden ser de utilidad para el proyecto y se enumerarán sus ventajas en la tabla 3.4.1.

- Tomcat o apache tomcat es un servidor web desarrollado en la plataforma JAVA lo que permite que sea instalado en ambientes multiplataforma, el proyecto se inició en el año de 1999 con la versión 3, actualmente contamos con la versión 7, en esta última versión se destacan las siguientes mejoras:
 - Limpieza interna de código
 - Mejoras para la detección de fugas de memoria
 - Soporte para servlet 3.0 y JSP 2.1 (Java Servlet Pages).



Figura 3.4.1 Logo Apache

- Cherokee, es un servidor web multiplataforma desarrollado en el año 2001



escrito en su totalidad en lenguaje C y distribuido bajo la licencia GNU, la última versión se dio a conocer en octubre del año 2011, entre sus principales características de destacan:

- Capacidad de realizar redirecciones.
- Permite ser utilizado como balanceador de carga.
- Dispone de un panel de administración desde la web.



Figura 3.4.2 Logo Cherokee

- IIS- servidor web desarrollado para la plataforma Windows con licencia propietaria que permite convertir una pc en un servidor de páginas ya sea en internet o intranet, sus principales características son:
 - Manejo de sitios web basados en asp .net.
 - Administración de sitios centralizada.
 - Manejo de certificados y páginas https.



Figura 3.4.3 Logo IIS 7

	TOMCAT	CHEROKEE	IIS
MULTIPLATAFORMA	XXXX	XXXX	X
MANEJO DE APLICACIONES BASADAS EN .NET	X	X	XXXX
ALINEACIÓN CON HERRAMIENTAS DEL NEGOCIO	X		XXXX
FACILIDAD ADMINISTRACIÓN DEL SITIO	XX	XX	XXX
FACILIDAD DE CONFIGURACIÓN	XX	XX	XX
SOPORTE	XXXX	XXX	XXXX
APROVECHAMIENTO DE RECURSOS	XXX	XXX	XXXX
MANEJO DE CERTIFICADOS	XX	XX	XXXX
EXPERIENCIA EN USO DE LA HERRAMIENTA	XX	X	XXXX
TOTAL	21	18	30

Tabla 3.4.1. Comparativo Front-end



De acuerdo al análisis realizado a los diferentes productos de software que pueden ayudar a la realización del proyecto se concluye que el uso de IIS es la mejor opción para el desarrollo del Front-End debido a que se apega a los estándares utilizados en la organización y a las facilidades que nos ofrece en cuanto a facilidad de uso, configuración y soporte, si bien el uso de IIS se limita solamente a servidores basados en WINDOWS se compensa con la mayoría de servidores basados en dicha plataforma que se maneja dentro de la institución y la experiencia con la que se cuenta en dicha herramienta.

Análisis Back-End.

Para el análisis del Back-End se estudiarán tres lenguajes de programación que ayudarán a la realización del proyecto y se enumerarán sus ventajas en la tabla 3.4.2.

- JAVA- lenguaje de programación desarrollado por SUN en el año de 1995, su sintaxis deriva en mucho de los lenguajes C y C#, las aplicaciones desarrolladas bajo este lenguaje pueden ser ejecutadas desde cualquier máquina virtual de java (JVM) sin importar el sistema operativo, las principales características de este lenguaje es que es un lenguaje multiplataforma y la administración de memoria es de lo mejor ya que adopta un recolector de basura que gestiona el tiempo de vida de los objetos después de su ejecución.
- C -Lenguaje de programación desarrollado en el año de 1972 en los laboratorios Bell y fue la evolución del lenguaje de programación B, es un lenguaje orientado a la implementación de sistemas operativos multiplataforma aunque por su eficiencia en el código es utilizado también para el desarrollo de aplicaciones
- C# - Lenguaje de programación desarrollado en el año de 1980, su creación tuvo el propósito de extender el éxito que tuvo el lenguaje de programación C, con el paso del tiempo se añadieron mejoras al lenguaje para hacerlo un lenguaje multiparadigma.



	JAVA	C	C#
MULTIPLATAFORMA	XXXX	XX	XX
DESARROLLOS REALIZADOS	XX	X	XXXX
PROGRAMACIÓN ORIENTADA A OBJETOS	XXXXX	XXXX	XXXX
ALINEACIÓN CON HERRAMIENTAS DEL NEGOCIO	XX		XXXX
USO DE CLASES	XXXX	XXX	XXXX
SEGURIDAD DENTRO DEL LENGUAJE	XXX	XX	XX
HERRAMIENTAS PARA EL DESARROLLO DENTRO DE LA ORGANIZACIÓN	XX	XX	XXXX
SOPORTE	XXX	XXX	XXXX
PAQUETES ADICIONALES	XXX	XXX	XXXX
EXPERIENCIA EN USO DE LA HERRAMIENTA	XX	X	XXXX
TOTAL	30	21	36

Tabla 3.4.2. Comparativo Back-end.

En el análisis realizado a los lenguajes de programación candidatos a ser usados para el procesamiento de los datos ingresados a partir del Front-End se concluye que el lenguaje de programación C# es el más adecuado para contribuir al desarrollo del



Back-End ya que se cuenta con la experiencia necesaria, el soporte, las herramientas para su utilización además de que el lenguaje se alinea con las políticas de la organización.

Registro de log base de datos.

Para el registro de log se realizará una comparativa con tres lenguajes de programación que serían de utilidad para el proyecto, en la tabla 3.4.3 se enumeraran las ventajas de los productos.

- Perl- lenguaje de programación multiparadigma, orientado a objetos y multiplataforma desarrollado en el año de 1987 tomando como base el lenguaje de programación C ya que utiliza una estructura de programación en bloques, gano fama por su gran capacidad para analizar, procesar texto y extraer información.



Figura 3.4.4 Logo Perl

- Python-Lenguaje de programación interpretado, orientado a objetos y multiplataforma, es decir, requiere de un intérprete para ser ejecutado a diferencia de los lenguajes compilados, nace a finales de los ochenta y es distribuido bajo la licencia GNU.



Figura 3.4.5 Logo Python

- Ruby- Lenguaje de programación interpretado, multiplataforma y orientado a objetos desarrollado a partir del año 2003 y dado a conocer al público en el año de 1995, lenguaje interpretado en una sola pasada y distribuido bajo licencia GNU



Figura 3.4.6 Logo Ruby



	Perl	Python	Ruby
MULTIPLATAFORMA	XXXX	XXXX	XXXX
EXPERIENCIA EN DESARROLLOS	XXXX	X	
ENFOQUE A ANÁLISIS DE ARCHIVOS TXT	XXXX	XX	XXX
ALINEACIÓN CON HERRAMIENTAS DEL NEGOCIO	XXX		
SEGURIDAD DENTRO DEL LENGUAJE	XXX	XX	XX
HERRAMIENTAS PARA EL DESARROLLO DENTRO DE LA ORGANIZACIÓN	XXXX	XX	XX
PROGRAMACIÓN ORIENTADA A OBJETOS	XXXX	XXXX	XXXX
FACILIDAD DE CONEXIÓN CON SQL	XXXX	XXX	XXXX
SOPORTE	XXXX	XXXX	XXXX
PAQUETES ADICIONALES	XXXX	XXX	XXXX
EXPERIENCIA EN USO DE LA HERRAMIENTA	XXX	X	X
TOTAL	41	26	28

Tabla 3.4.3 Comparativo lenguajes para lectura de log.



En el análisis realizado a los lenguajes de programación candidatos a ser usados para el procesamiento de los datos obtenidos a partir del archivo de log se concluye que el lenguaje de programación PERL es el más adecuado para contribuir al desarrollo del programa que realizará la lectura y la inserción de los datos a la base de SQL ya que se cuenta con la experiencia necesaria, el soporte, las herramientas para su utilización además de que el lenguaje se orienta directamente a la lectura y análisis de archivos de texto y se alinea con las políticas de la organización.

Base de datos.

Para el registro de los datos se realizará una comparativa con tres distintos productos de base de datos que serían de utilidad para el proyecto, en la tabla 3.4.4 se enumeraran las ventajas de los productos.

- SQL Server- Sistema para la gestión de base de datos desarrollado por Microsoft en el año de 1989 basado en un modelo relacional, constituye un buen oponente a motores potentes como son Oracle y MySQL, la última versión desarrollada apareció en marzo del año 2012.



Figura 3.4.7 Logo Microsoft SQL Server 2008

- MySQL- Sistema de gestión de datos relacional, multihilo y multiusuario distribuido bajo licencia GPL desarrollado en mayor medida bajo el lenguaje



ANSI C, su última versión estable se dio a conocer el 31 de enero de 2014, es una plataforma estable utilizada por empresas grandes como son wikipedia, google, Facebook, etc.



Figura 3.4.8 Logo MySQL

- PostgreSQL- Es un sistema de gestión de base de datos relacional de código abierto distribuido bajo licencia BSD, el proyecto se inicia en el año de 1982 como uno de los primeros intentos en crear un motor de base de datos relacional, actualmente el proyecto tuvo su último reléase en el año de 2008.



Figura 3.4.9 Logo PostgreSQL



	SQLServer	MySQL	PostgreSQL
CONFIGURACIÓN DE HERRAMIENTAS	XXXX	XX	XX
EXPERIENCIA EN DESARROLLOS	XXXX	XX	
USO DE STORE PROCEDURES Y TRIGGERS	XXXX	XX	XXX
ALINEACIÓN CON HERRAMIENTAS DEL NEGOCIO	XXXX	XX	
SEGURIDAD	XXX	XX	XX
HERRAMIENTAS PARA EL DESARROLLO DENTRO DE LA ORGANIZACIÓN	XXXX	XX	
FACILIDAD DE CONEXIÓN CON VISUAL STUDIO	XXXX	XXXX	XXXX
MULTIUSUARIO MULTIHILOS	XXXX	XXX	XXXX
SOPORTE	XXXX	XXXX	XXXX
PAQUETES ADICIONALES	XX	X	X
EXPERIENCIA EN USO DE LA HERREMIENTA	XXX	X	X
TOTAL	40	25	21

Tabla 3.4.4 Comparativo base de datos.



En el análisis realizado a los motores de bases de datos candidatos a ser usados para el almacenamiento de los datos y de usuarios, se concluye que SQL Server es el más adecuado para contribuir al desarrollo de la base de datos para trabajar y conectar el entorno visual para graficar datos y conectar el módulo de seguridad que contendrá la página, además de que se cuenta con la experiencia necesaria, el soporte, las herramientas para su utilización y se alinea con las políticas de uso de software de la organización.

Marcos de referencia

Para elegir un marco de referencia que se adecue a nuestro proyecto, se realizará una comparativa con tres marcos diferentes, COBIT, ITIL y MOF. En la tabla 3.4.5 se enumeran algunos de las principales características.

- ITIL, es un marco de trabajo de las buenas prácticas destinadas a facilitar la entrega de servicios de tecnologías de la información (TI). Propone el establecimiento de estándares que ayuden en el control, operación y administración de los. Plantea hacer una revisión y reestructuración de los procesos existentes en caso de que lo necesiten, es decir, si el nivel de eficiencia es bajo o si existe una forma más eficiente de hacer las cosas, lo que lleva a una mejora continua.
- COBIT determina, con el respaldo de las principales normas técnicas internacionales, un conjunto de mejores prácticas para la seguridad, la calidad, la eficacia y la eficiencia en TI, aspectos necesarios y básicos para alinear TI con el negocio, identificar riesgos, entregar valor al negocio, gestionar recursos y medir el desempeño, el cumplimiento de metas y el nivel de madurez de los procesos de la organización.

La misión de COBIT es investigar, desarrollar, publicitar y promocionar un marco



de trabajo Gobierno de control de Gobierno de TI autoritativo, actualizado, y Evolución Gestión aceptado internacionalmente que se adopte por las empresas y se Control emplee en el día a día por los gerentes de negocio.

- MOF es un conjunto de prácticas recomendadas a partir de las cuales se pueden diseñar los procedimientos, controles y funciones necesarios para que la infraestructura de TI funcione con eficacia. MOF está basado en la Biblioteca de Infraestructuras de TI (ITIL) y aporta concreción a la plataforma de Microsoft. Ofrece directrices sobre el modo de planear, implementar y mantener procesos operativos de TI que respalden las soluciones de servicio críticas.



	ITIL	COBIT	MOF
CONFIABILIDAD	XXXX	XXXX	XXX
ESCALABILIDAD	XXXX	XXXX	XXX
CAPACITACIÓN	XXXX	XXXX	XX
FLEXIBILIDAD	XXXX	XXXX	XX
ESTADO	XXXX	XXX	XX
HERRAMIENTAS	XXXX	XXX	XX
ENFOQUE	XXXX	XXX	XXXX
SUSTENTO	XXXX	XXX	XXX
TOTAL	32	28	21

Tabla 3.4.5 Cuadro comparativo Marcos de referencia

En el análisis realizado a los marcos de referencia candidatos a ser usados como referencia, se concluye utilizar ITIL, ya que es el más adecuado para contribuir al desarrollo del proyecto, además de que se cuenta con la experiencia y conocimiento para su utilización.



3.5 BOSQUEJO DE PANTALLAS SUGERIDAS A LA EMPRESA

Se hará una breve explicación de cada una de las pantallas propuestas para el proyecto, así como su contenido y funciones de cada una de ellas.

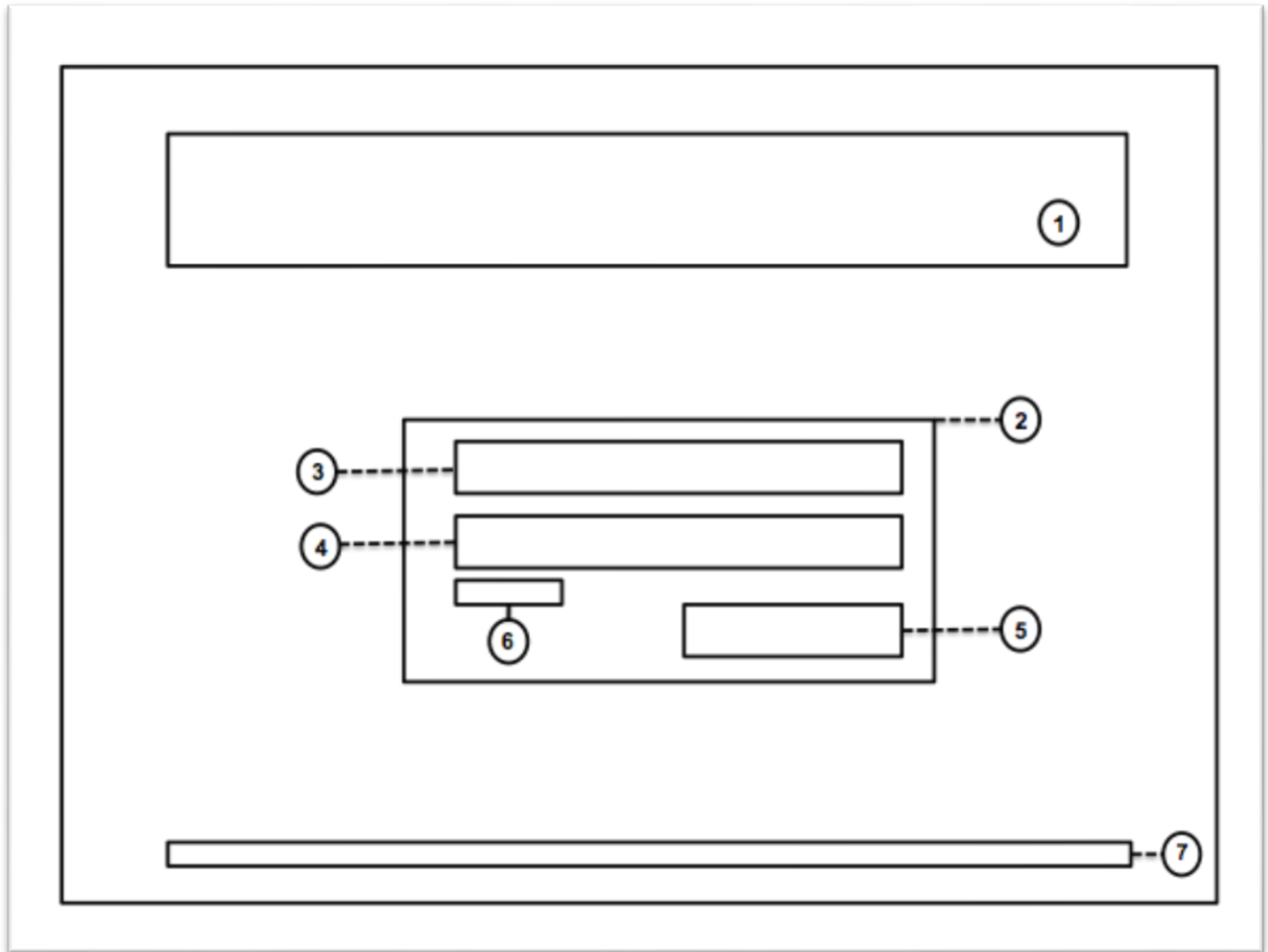


Figura 3.5.1 Pantalla de login

En la figura 3.5.1 podemos ver el bosquejo de la pantalla de login desde la cual el usuario podrá acceder al sistema mediante el uso de sus credenciales, previamente asignadas por un administrador, los elementos que conforman esta pantalla son (1) banner con nombre y logo de la empresa que utilizará este sistema, (2) cuadro para ingresar credenciales de autenticación el cual a su vez contara con los elementos (3) campo para agregar usuario, (4) campo para agregar contraseña, (5) botón para validar



credenciales y dar acceso al sistema y (6) liga de recuperación de contraseña; finalmente en el pie de página se incluiría (7) Copy Right (reservación de derechos) y año de elaboración por parte de la empresa que utilizará la aplicación. Cabe señalar que los elementos (1) y (7) serán constantes en el resto de las pantallas, por lo tanto ya no serán descritos en ellas.

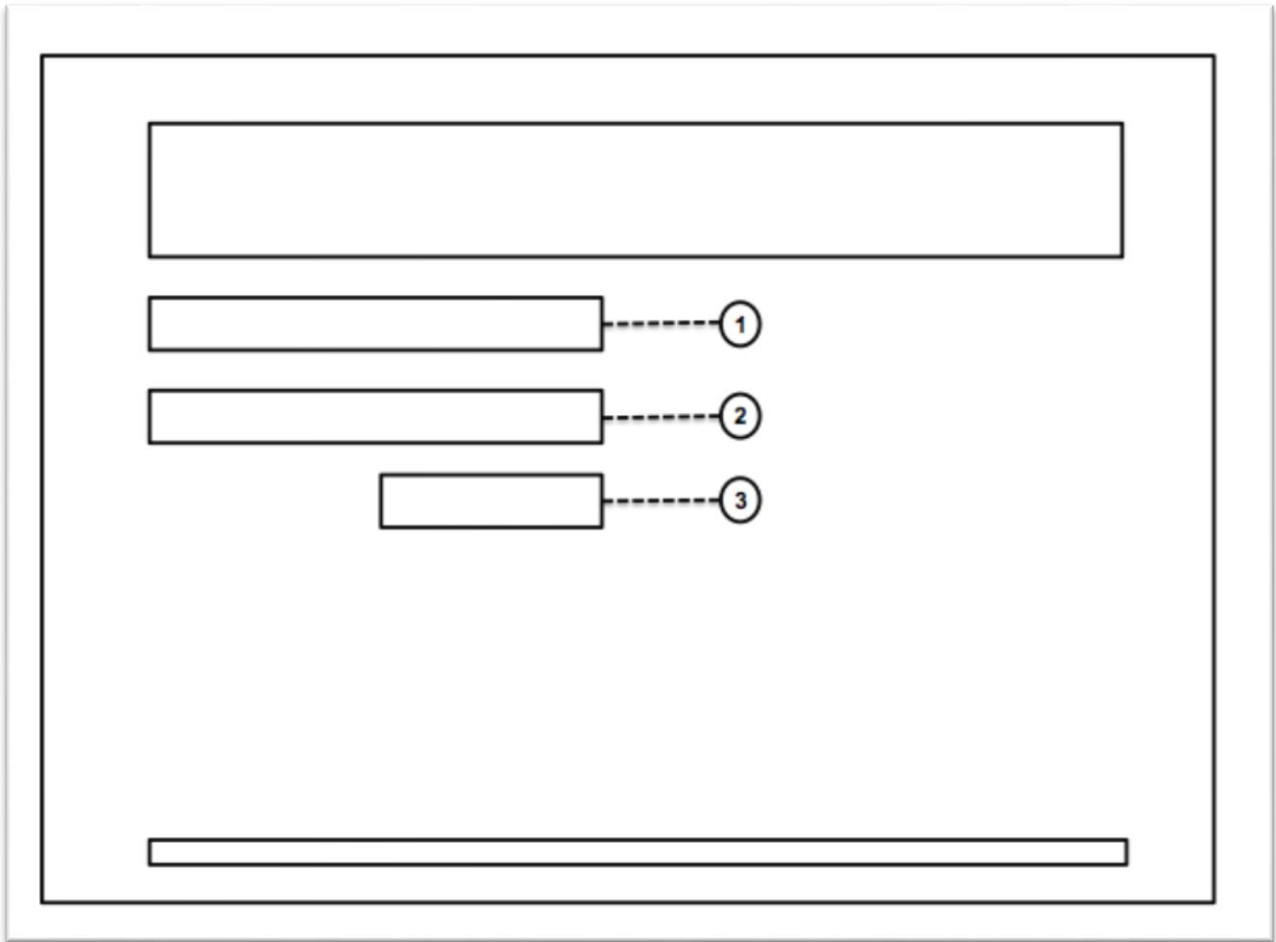


Figura 3.5.2 Pantalla de recuperación de contraseña

En la figura 3.5.2 se muestra la pantalla de recuperación de contraseña, la cual contiene los elementos (1) pregunta secreta, que sería un dato constante, es decir a todos los usuarios se les colocaría la misma pregunta ya que no hay un formulario de inscripción desde el cual cada uno pueda personalizar este dato, (2) campo para respuesta secreta, el cual sería un dato que conozcan tanto la empresa como el usuario y les sirva de identificador (número de empleado, RFC, etc) y (3) botón para



validar información y envío de correo con nueva contraseña, el correo será el que fue proporcionado por la empresa al usuario.

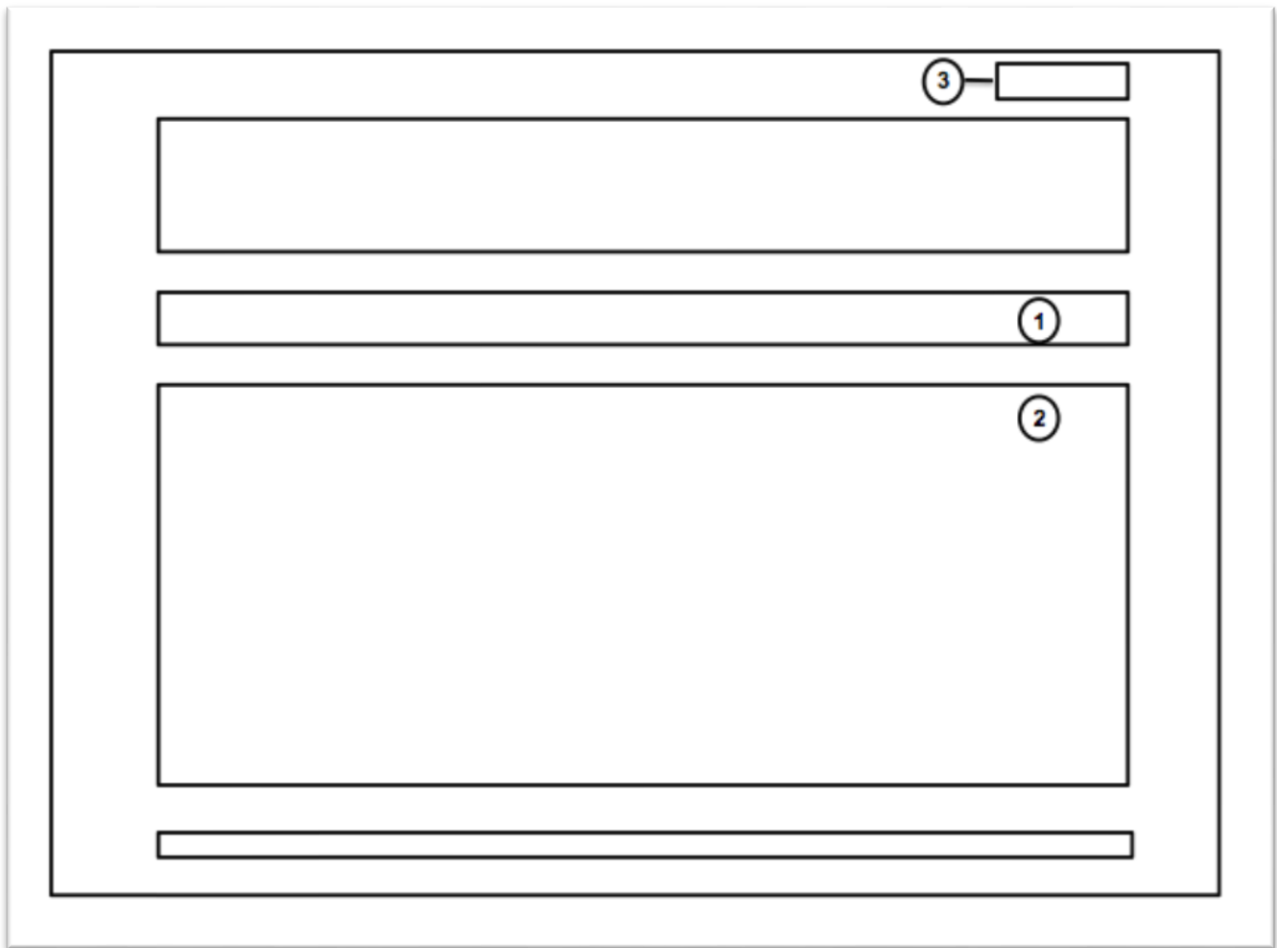


Figura 3.5.3 Pantalla principal

En la figura 3.5.3 se muestra la pantalla principal de la aplicación, desde la cual se vería el monitor de la aplicación, los elementos que se observan en esta pantalla son (1) menú, desde el cual se podría ir accedendo a las distintas opciones que puede tener el sistema y (2) gráfica de rendimiento, desde la cual se verá el comportamiento del WS de acuerdo al rango de tiempo que se haya seleccionado en el menú (tiempo real, día, semana, mes, histórico), como último elemento tenemos (3) botón de cierre de sesión, desde el cual el usuario podrá terminar la sesión en la aplicación, regresándolo a la página de inicio (pantalla de login).

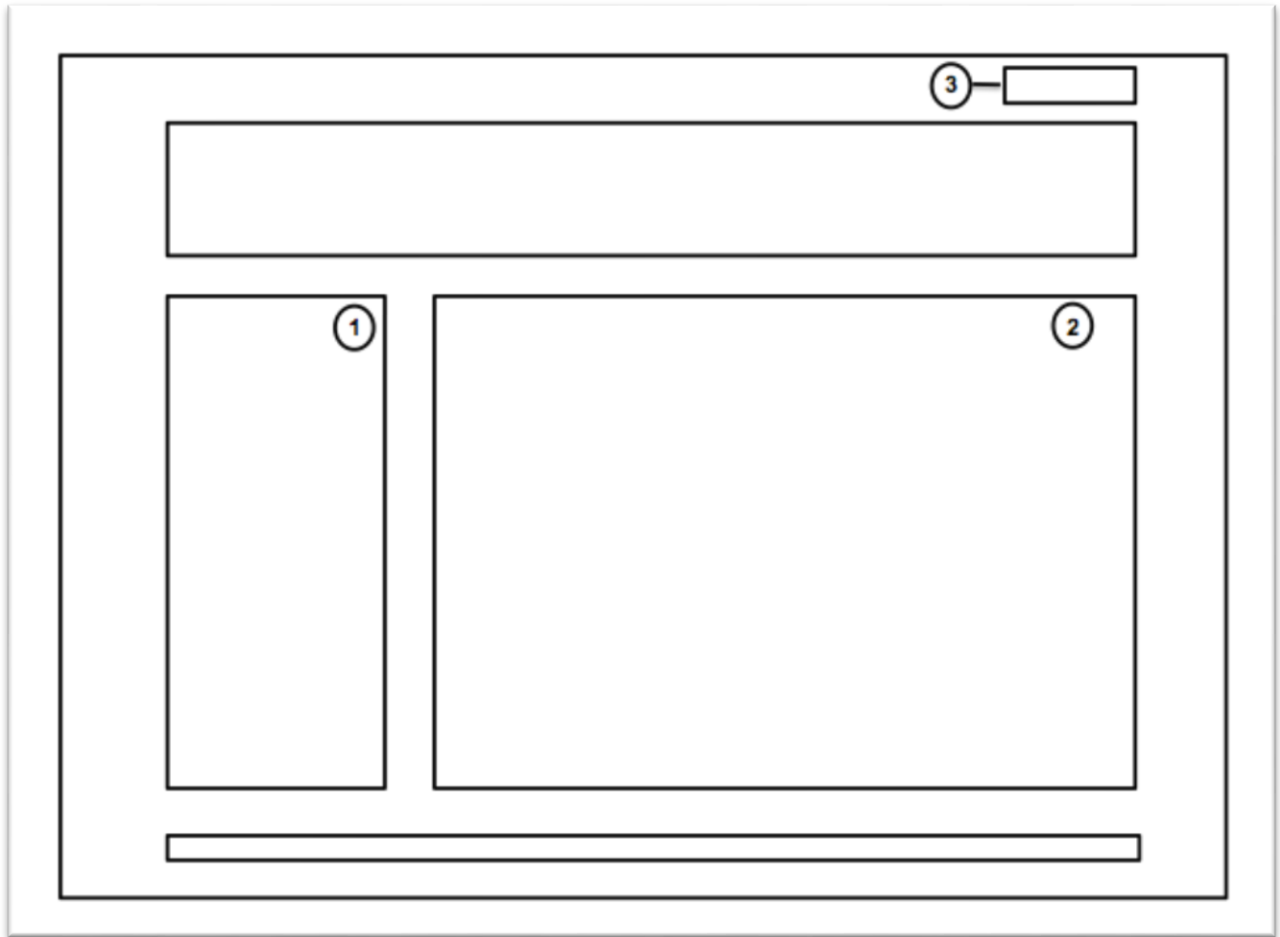


Figura 3.5.4 Pantalla de administración

En la figura 3.5.4, se puede observar el bosquejo de la pantalla de administración del sistema, desde la cual se podrá hacer el alta, modificación y borrado de usuarios, contraseñas, perfiles. El elemento (1) es el menú desde el cual se podrá navegar en las distintas opciones de administración, el elemento (2) será el espacio destinado en el que se mostrarán los distintos formularios desde los cuales se podrá hacer las actividades de administración mencionadas anteriormente, y finalmente el elemento (3) será el botón de cierre de sesión.

Capítulo IV

Diseño y construcción de la aplicación.



El diseño de servicio desde la perspectiva de ITIL se enfoca en la planeación, modificación y construcción de servicios nuevos y modificación a servicios existentes adecuándose a los objetivos establecidos en la estrategia del servicio.

En el siguiente capítulo se tomará la estrategia definida y se diseñará y construirá el servicio a través del uso de diagramas en base a los objetivos del negocio.

4.1 MODELADO DEL SISTEMA

Diagrama de contexto

En este diagrama de la figura 4.1.1 tenemos tres actores; el servidor de timbrado en el cual se timbran los documentos, sus actividades son muy importantes, ya que la información estadística que genere se enviará a la base de datos tim_xlog, esta base de datos es la que almacena los datos que utilizará el sistema para la generación de las consultas.

El Administrador del sistema, quien aparte de modificar el sistema en caso de ser necesario, también administrará a los usuarios que podrán acceder al portal.

El usuario del portal quien mediante un usuario y contraseña podrá ingresar al portal para hacer las consultas ya sea en tiempo real, del día, del mes o el histórico.

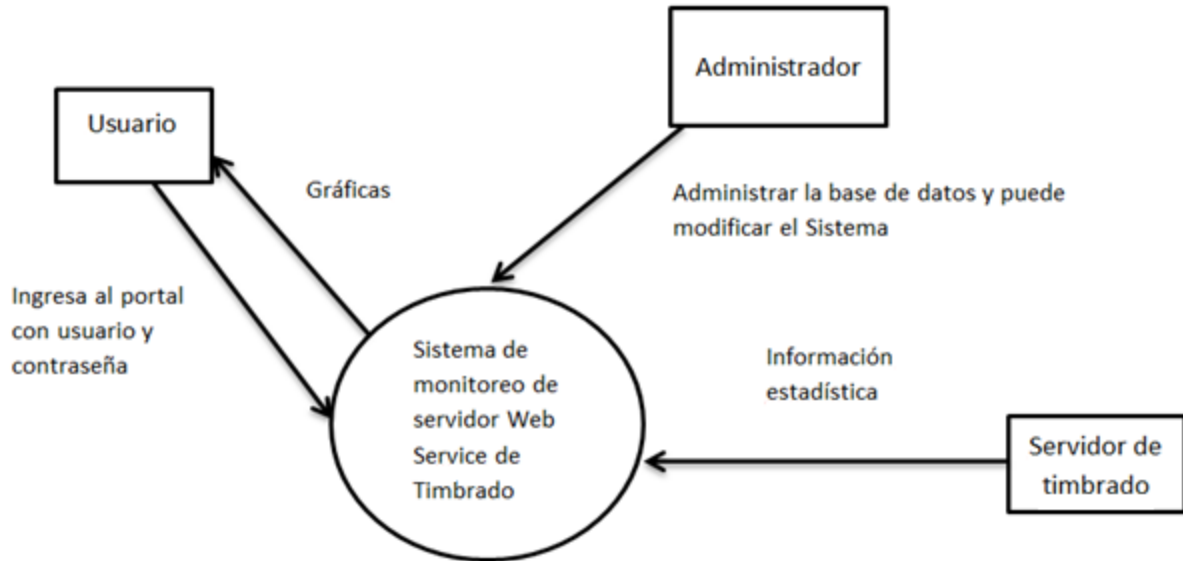


Figura 4.1.1 Diagrama de Contexto

Diagrama Casos de Uso

En el diagrama de casos de uso figura 4.1.2 como actores principales tenemos al usuario quien hará el uso de portal y al administrador del portal, con sus actividades más específicas.

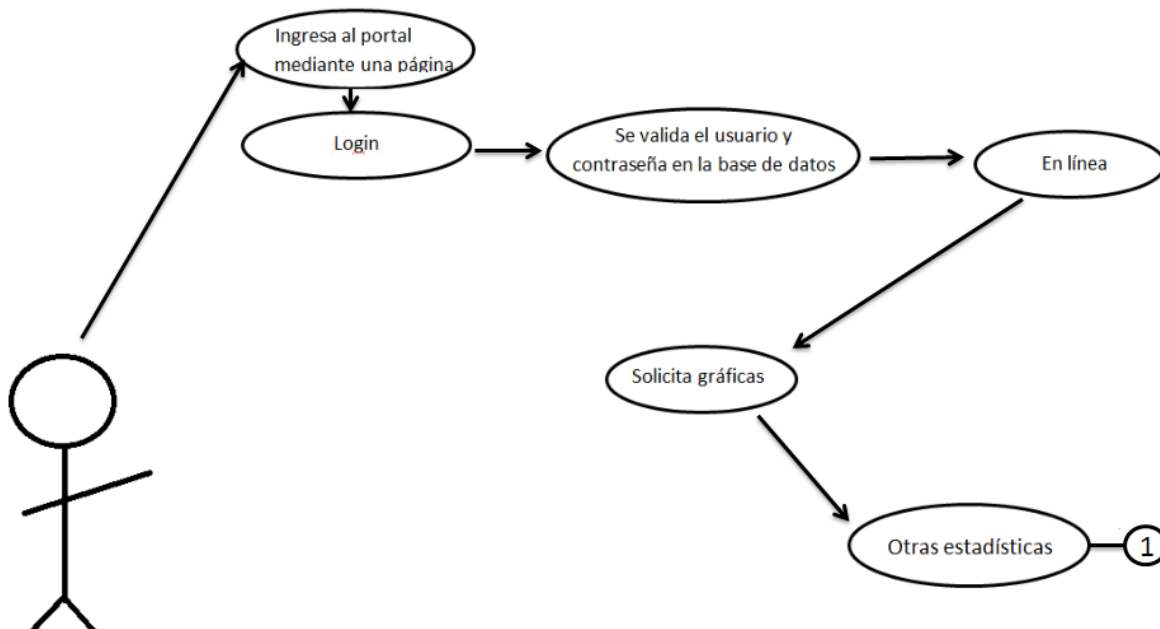


Figura 4.1.2 Diagrama Casos de uso

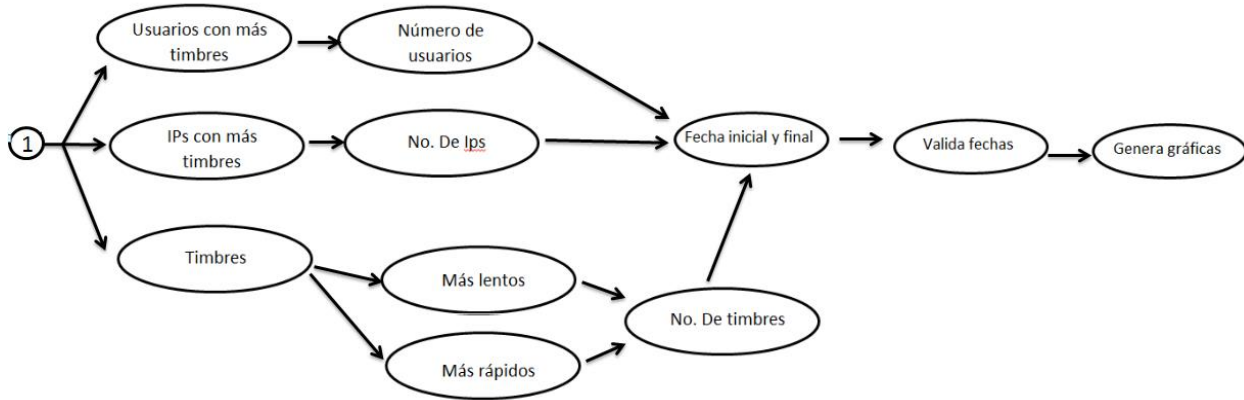


Figura 4.1.2 Diagrama Casos de uso

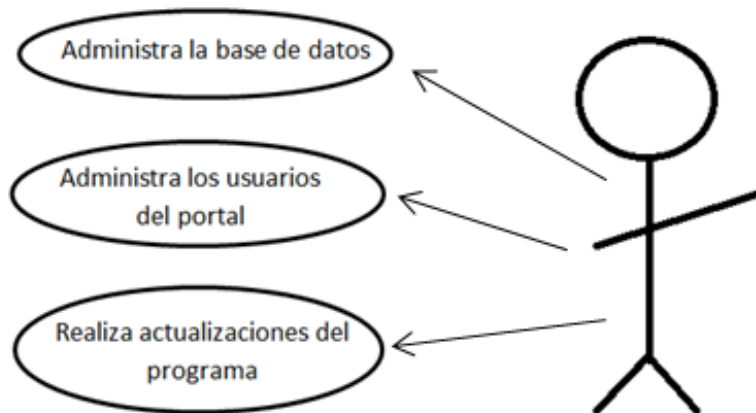


Figura 4.1.3 Diagrama Casos de uso

Diagrama entidad relación

En el diagrama entidad relación figura 4.1.3 de la base de datos contamos con cuatro tablas xlog, tiempos, transacción y usuario.

La tabla xlog guarda información de los timbres que se han generado, cuenta con una llave primaria que es el id, el id es el contador de timbres.



La tabla tiempos guarda los tiempos de respuesta de diferentes transacciones se tienen como llave foránea el id.

La tabla transacción contiene datos de quién y cuándo se realizó el timbrado, se tiene como llave foránea el Tri que es el identificador de la transacción.

La tabla usuario contiene a los usuarios que podrán ingresar al portal mediante la identificación



Figura 4.1.4 Diagrama entidad relación

Diccionario de datos

En las tablas 4.1.1 del diccionario de datos se describe cada tabla que forma parte de la base de datos, así como cada campo.



Xlog					
Descripción : guarda información de los timbres que se han generado					
NOMBRE ORIGINAL	ACRÓNIMO	TIPO DE DATO Y LONGITUD	PK	OPCIÓN NULL	DESCRIPCIÓN DEL CAMPO
FechaHora	Fec_Hra	varchar(MAX)	No	NOT NULL	Fecha y hora de cuando se generó el timbre
Trid	Tri	varchar(MAX)	No	NOT NULL	Identificador de la transacción
Usr	Usr	varchar(MAX)	No	NOT NULL	Usuario de timbrado que generó el timbre
Trtime	Trt	varchar(MAX)	No	NOT NULL	Tiempo de respuesta del server para timbrar
SQL	SQL	varchar(MAX)	No	NOT NULL	Valida que el rfc del cliente
Clidid	Cli	varchar(MAX)	No	NOT NULL	
RFCemi	RFC_emi	varchar(MAX)	No	NOT NULL	RFC del emisor
id	Id	Int	SI	NOT NULL	Contador de timbre

Tabla 4.1.1 Diccionario de datos, tabla Xlog



Tiempos					
NOMBRE ORIGINAL	ACRÓNIMO	TIPO DE DATO Y LONGITUD	PK	OPCIÓN NULL	DESCRIPCIÓN DEL CAMPO
id	Id	Int	Si	NOT NULL	Contador de timbre
total	Tot	varchar(50)	No	NOT NULL	Tiempo total de respuesta del server para timbrar
usuario	Vus	varchar(50)	No	NULL	Tiempo en verificar usuario de timbrado
expedidosat	Exp_SAT	varchar(50)	No	NULL	Tiempo en saber si el documento ya fue expedido
timbradoant	Tim_ant	varchar(50)	No	NULL	Te indica si el documento ya fue timbrado anteriormente
sello	Sel	varchar(50)	No	NULL	Tiempo en verificar el sello
rticfdi	Rti_cfd	varchar(50)	No	NULL	
rtsc	Rts	varchar(50)	No	NULL	
gpaal	Gpa	varchar(50)	No	NULL	
gpsc	Gps	varchar(50)	No	NULL	

Tabla 4.1.2 Diccionario de datos, tabla Tiempos



Transacción					
Descripción : Contiene datos específicos de quién solicitó el timbre					
NOMBRE ORIGINAL	ACRÓNIMO	TIPO DEDATO Y LONGITUD	PK	OPCIÓN NULL	DESCRIPCIÓN DEL CAMPO
trid	Tri	varchar(50)	Si	NOT NULL	Identificador de la transacción
id	Id	Int	No	NOT NULL	Contador de timbre
fecha	Fec	datetime	No	NOT NULL	Fecha de cuando se generó el timbre
usuario	Usu	varchar(50)	No	NOT NULL	Usuario de timbrado que generó el timbre
ip	Ip	varchar(50)	No	NOT NULL	Indica la ip que realizó la transacción
rfc	Rfc	varchar(15)	No	NOT NULL	RFC que generó el timbre

Tabla 4.1.3 Diccionario de datos, tabla Transacción



Usuario					
Descripción: Contiene a los usuarios que podrán ingresar al portal mediante la identificación					
NOMBRE ORIGINAL	ACRÓNIMO	TIPO DE DATO Y LONGITUD	PK	OPCIÓN NULL	DESCRIPCIÓN DEL CAMPO
usuario	Usu_log	varchar(10)	No	NOT NULL	Usuario que podrá ingresar a la página
password	Pass	varchar(20)	No	NOT NULL	Contraseña del usuario
perfil	Per	varchar(20)	NO	NOT NULL	Tipo de usuario
intentos	Intent	Int	NO	NOT NULL	Intentos de acceso fallidos
correo	Mail	varchar(250)	No	NULL	Correo del usuario

Tabla 4.1.4 Diccionario de datos, tabla Usuario

4.2 CREACIÓN DE LA BASE DE DATOS

El proceso de creación de la base de datos comienza con la instalación del programa con el cual se va a crear la misma, en nuestro caso es MS- SQL Server 2008 ya que ésta es la versión que posee la empresa a la cual se está realizando el desarrollo, y tiene licencia. El proceso es muy sencillo con el asistente de instalación, sólo es necesario seleccionar las características que llevará la instalación (figura 4.2.1) y darle siguiente en el menú, será necesario logear los servicios, seleccionar el usuario administrador de la aplicación.

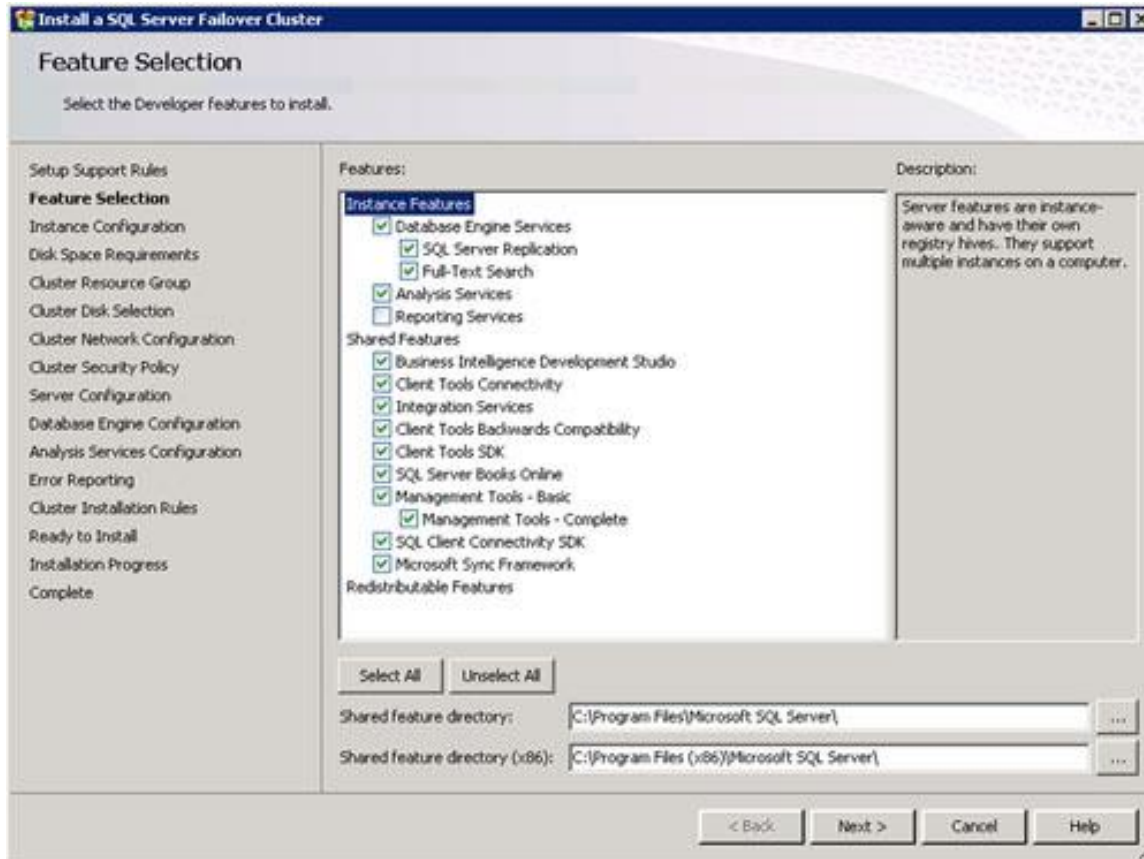


Figura 4.2.1 Opciones de instalación

Después de darle siguiente en todas las ventanas que aparecen, comenzará el proceso de instalación, el cual demorará algunos minutos, al finalizar mostrara un emnsaje indicándonos que el proceso fue concluido satisfactoriamente (figura 4.2.2) y al terminar tendremos instalado el motor de la base de datos, asi como el manejador gráfico del mismo.

Una vez terminado procederemos a realizar la creación de la base de datos, este proceso se puede hacer de dos formas, desde el asistente o desde la línea de comandos, se mostrarán a continuacion ambos procesos.

Desde el admistrador es necesario colocarse en la carpeta base de datos del menu de herramientas que se localiza en el lado izquierdo de la pantalla, posteriormente dale



clic con el botón derecho sobre el y se selecciona la opción nueva base de datos (Figura 4.2.2), y se coloca el nombre de la misma.

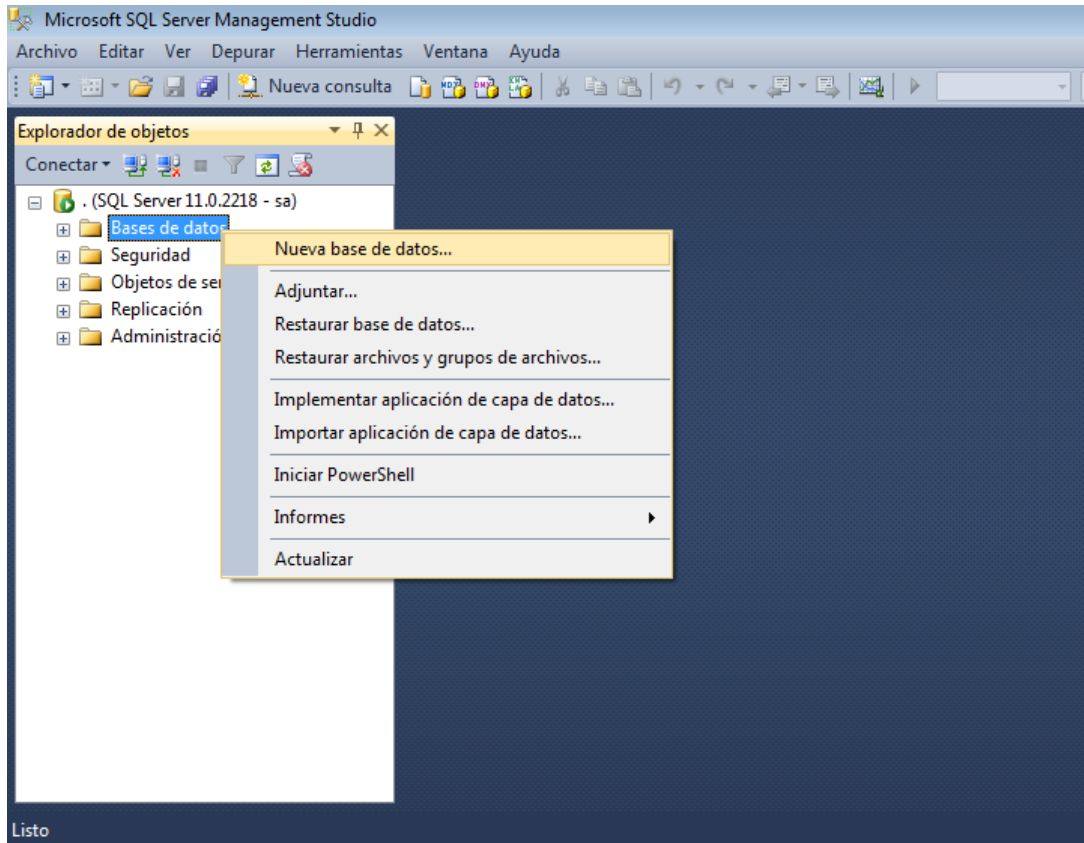


Figura 4.2.2 Creación de base de datos desde administrador

Desde cmd tan solo es necesario ejecutar la instrucción de creación de la base de datos mostrada en la figura 4.2.3, seguida de la instrucción go.



```
C:\Users\Arturo>sqlcmd
1> create database timbrador;
2> go
1>
```

Figura 4.2.3 Creación de base de datos desde CMD

El siguiente paso es la creación de las tablas, desde el administrador, seleccionamos la base de datos en la cual vamos a crearla, desplegamos el menú contextual de esta y el folder que dice tablas damos clic derecho y seleccionamos la opción nueva tabla, esto nos abrirá una nueva ventana en la cual podemos agregar el nombre de cada una de las columnas, así como el tipo de dato que le corresponde (Figura 4.2.4) al terminar de agregar las columnas damos clic en el botón de guardar.

Para hacer lo mismo desde CMD es necesario primero indicar la base de datos a utilizar, esto se hace mediante la instrucción USE y el nombre de la base de datos, una vez seleccionada se realiza la instrucción de SQL para crear y se indican cada una de las columnas con sus respectivos tipos de datos (figura 4.2.5). Las tablas utilizadas en el sistema se muestran en la figura 4.2.6.

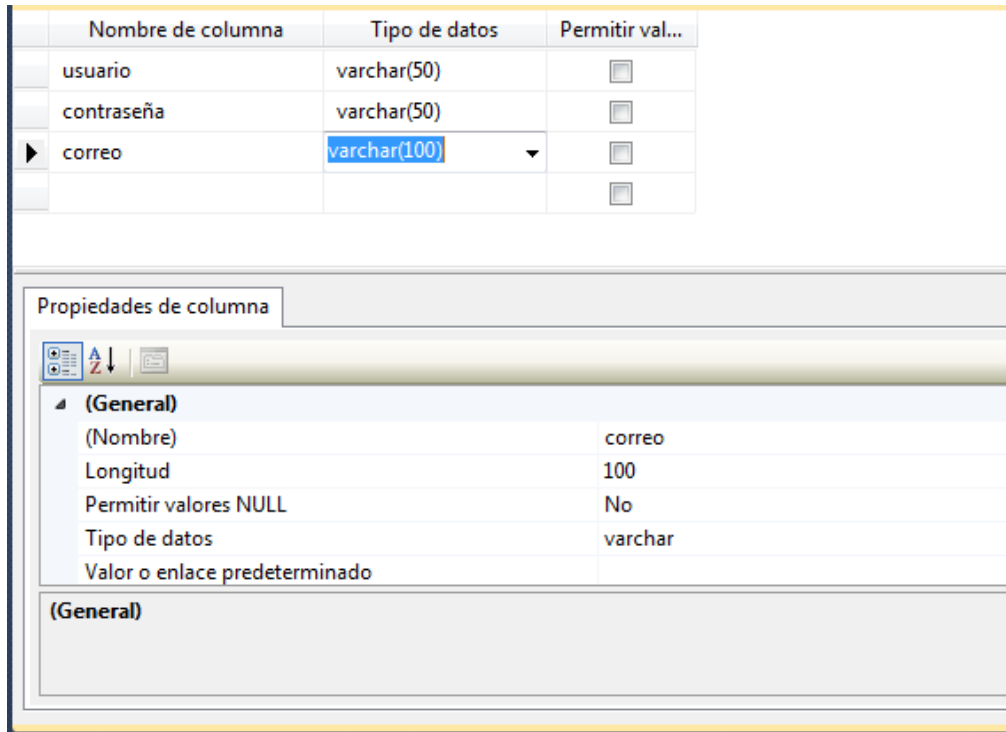


Figura 4.2.4 Creación de una tabla desde el administrador

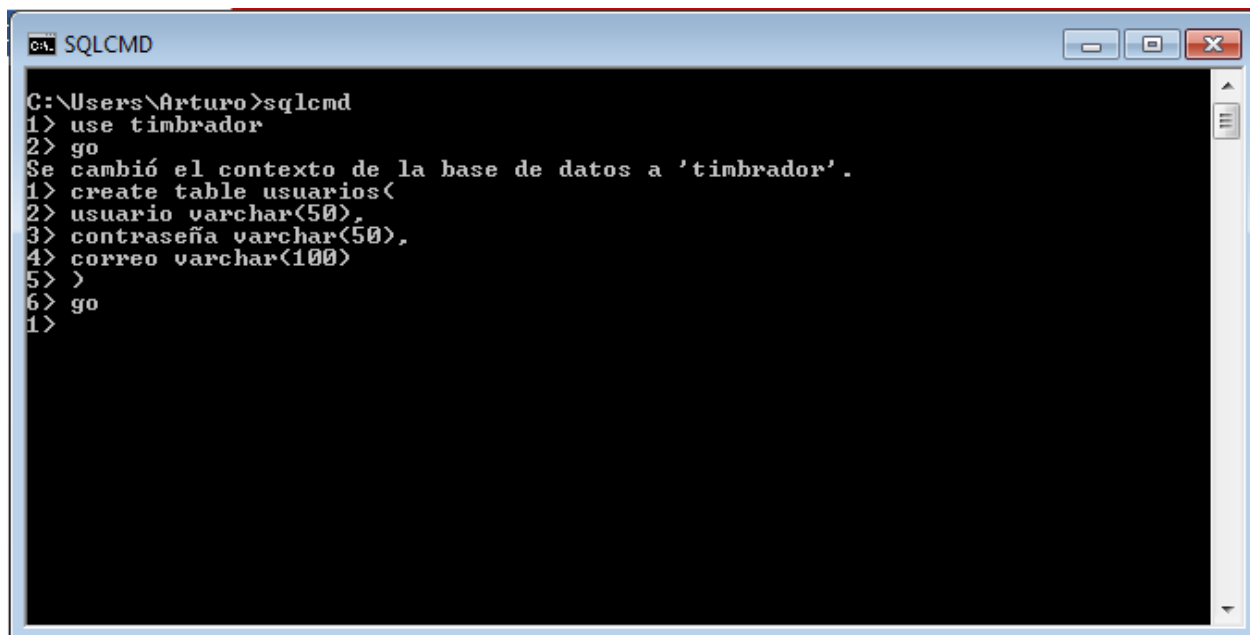


Figura 4.2.5 Creación de una tabla desde CMD

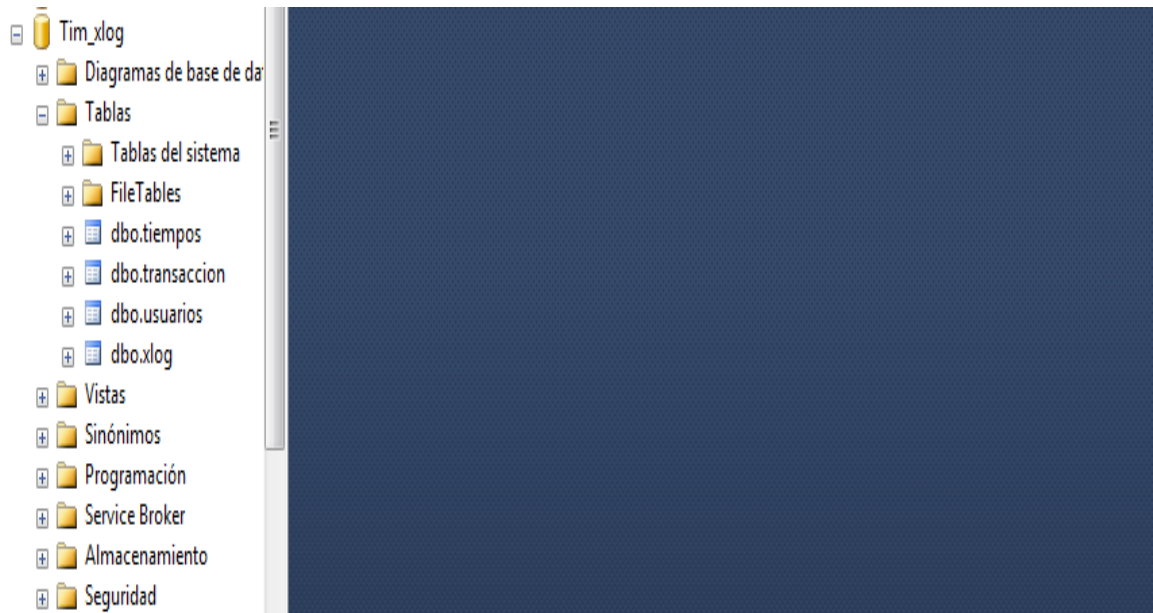


Figura 4.2.6 Árbol de tablas

La creación de procedimientos es muy similar ya sea desde el administrador o desde cmd, para hacerlo desde el manejador, hay que abrir el árbol de objetos de la base de datos, elegir la opción de programación y desplegarla, de ellas se da clic derecho sobre la carpeta procedimientos almacenados y del menú contextual se elige la opción crear (Figura 4.2.7), el resto del procedimiento se hace mediante el uso de programación con SQL.

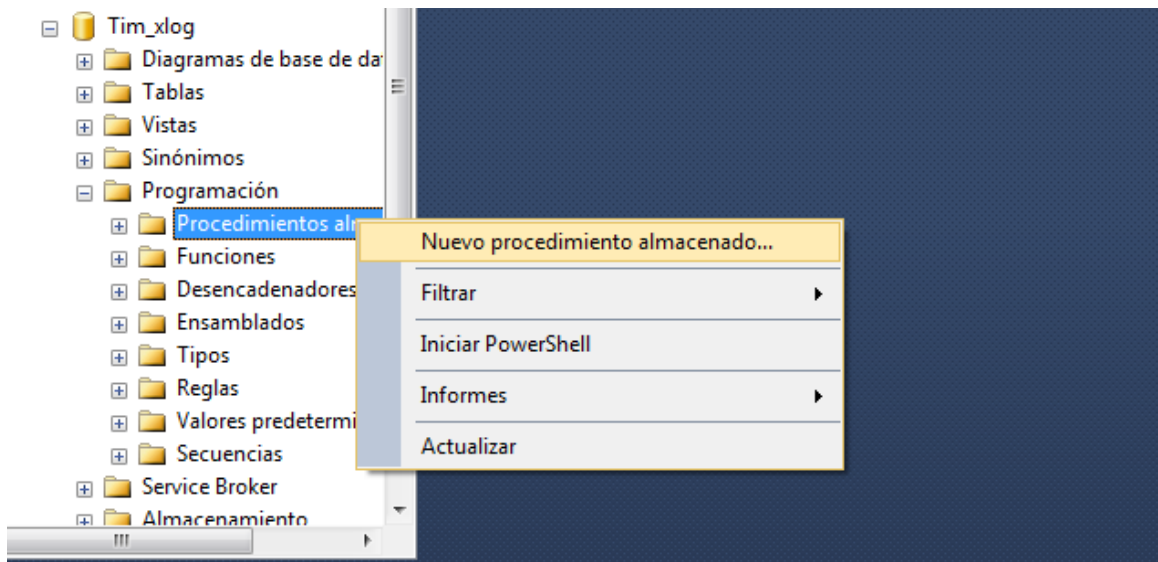
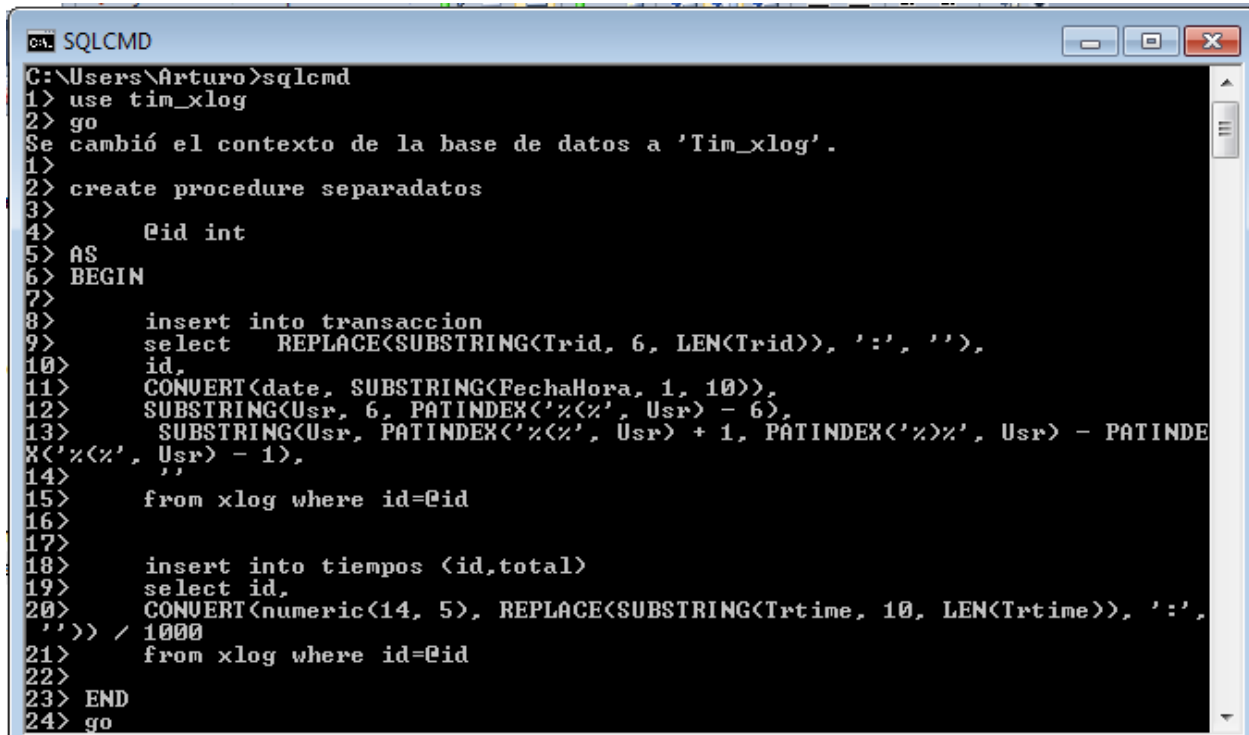


Figura 4.2.7 Creación de un procedimiento almacenado desde el administrador



Para hacerlo desde CMD (Figura 4.2.8), primero es necesario seleccionar la base de datos y mediante el uso de SQL se realiza la programación, cabe destacar que los procedimientos almacenados pueden hacer uso de variables, consultas, parámetros de entrada e incluso otro procedimiento.



```
SQLCMD
C:\Users\Arturo>sqlcmd
1> use tim_xlog
2> go
Se cambió el contexto de la base de datos a 'Tim_xlog'.
1>
2> create procedure separadatos
3>
4>     @id int
5> AS
6> BEGIN
7>
8>     insert into transaccion
9>     select  REPLACE(SUBSTRING(Trid, 6, LEN(Trid)), ':', ''),
10>     id,
11>     CONVERT(date, SUBSTRING(FechaHora, 1, 10)),
12>     SUBSTRING(Usr, 6, PATINDEX('%%', Usr) - 6),
13>     SUBSTRING(Usr, PATINDEX('%%', Usr) + 1, PATINDEX('%%', Usr) - PATINDE
14> X('%%', Usr) - 1),
15>     ''
16>     from xlog where id=@id
17>
18>     insert into tiempos (id,total)
19>     select id,
20>     CONVERT(numeric(14, 5), REPLACE(SUBSTRING(Trtime, 10, LEN(Trtime)), ':',
21> '')) / 1000
22>     from xlog where id=@id
23> END
24> go
```

Figura 4.2.8 Creación de un procedimiento almacenado desde CMD

Realizar un trigger (desencadenador) es básicamente lo mismo que un procedimiento, con la diferencia que para hacerlo desde el manejador elegir la opción de creación se hace desde el árbol de tablas (Figura 4.2.9), mientras que desde CMD es solamente hacer programación teniendo en cuenta que dentro del código de ésta se indica la tabla que hará hacer que el trigger sea iniciado (Figura 4.2.10). El árbol de procedimientos y triggers puede verse en la figura 4.2.11.

Por último se mostrara como realizar una consulta desde el administrador (Figura 4.2.12) y CMD (Figura 4.2.13); desde el administrador basta con dar clic derecho sobre la tabla de la cual se desea hacer la consulta y seleccionar la opción “seleccionar las



primeras 1000 filas”. Para hacerlo desde CMD basta con ejecutar la sentencia “select” y especificar la tabla a la cual se hará la consulta.

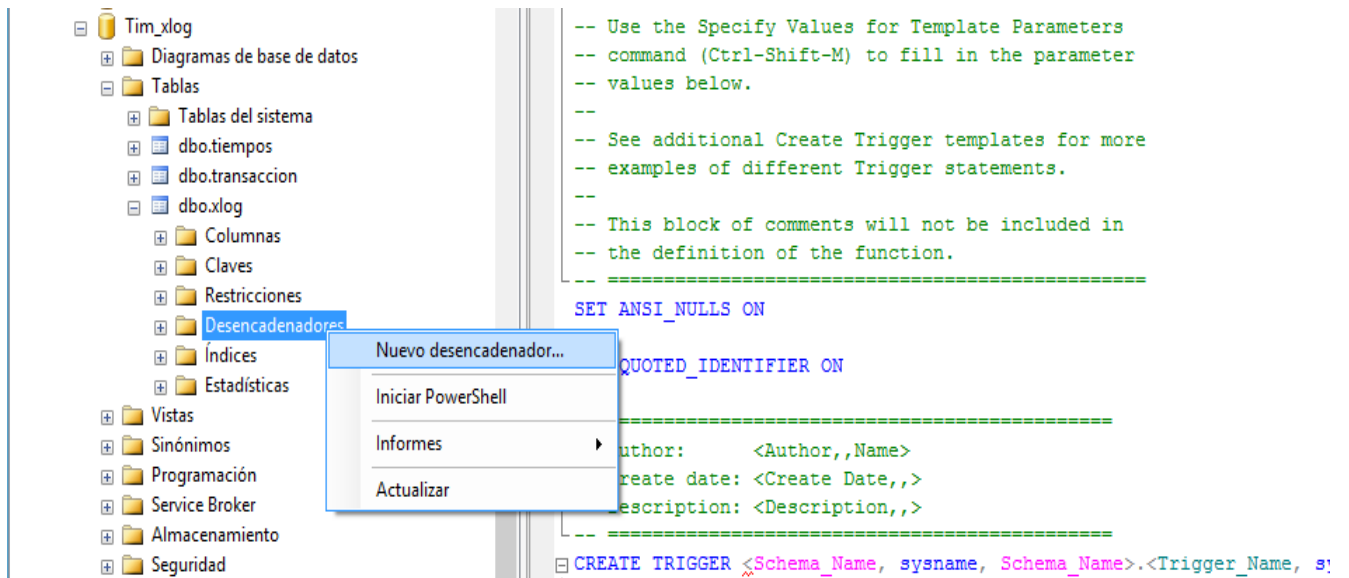


Figura 4.2.9 Creación de un trigger desde el administrador

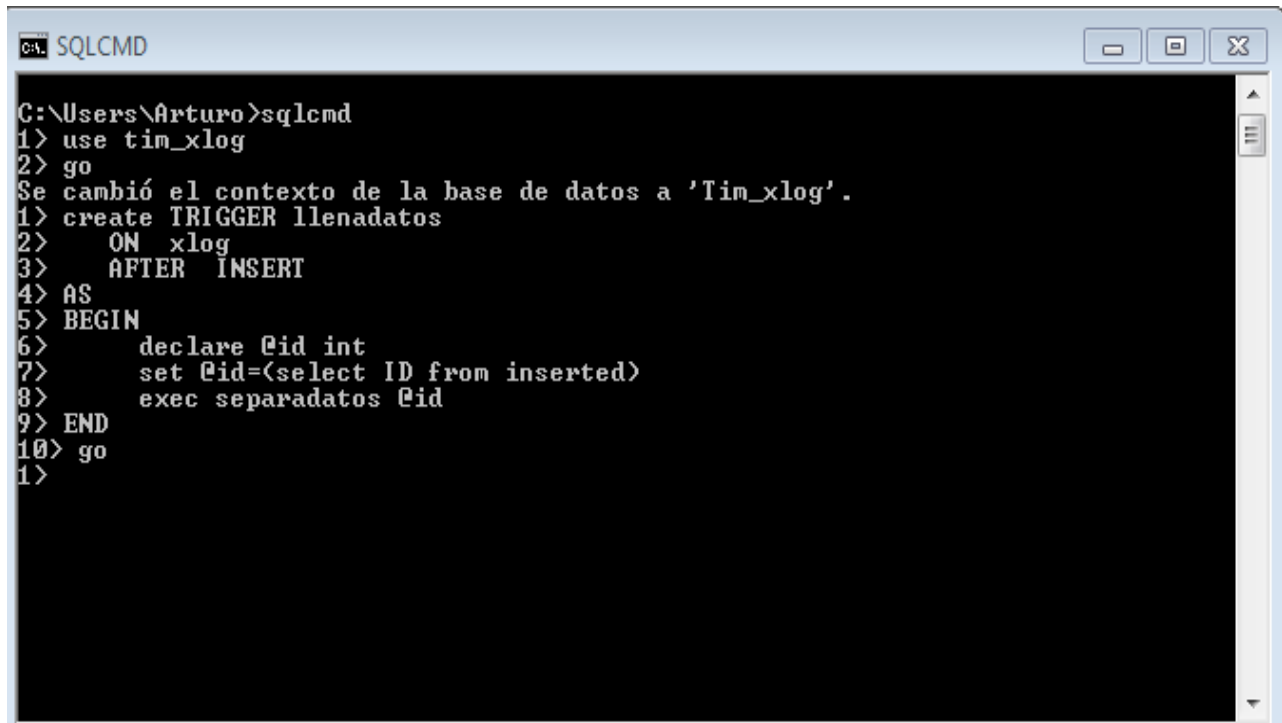


Figura 4.2.10 Creación de un trigger desde CMD

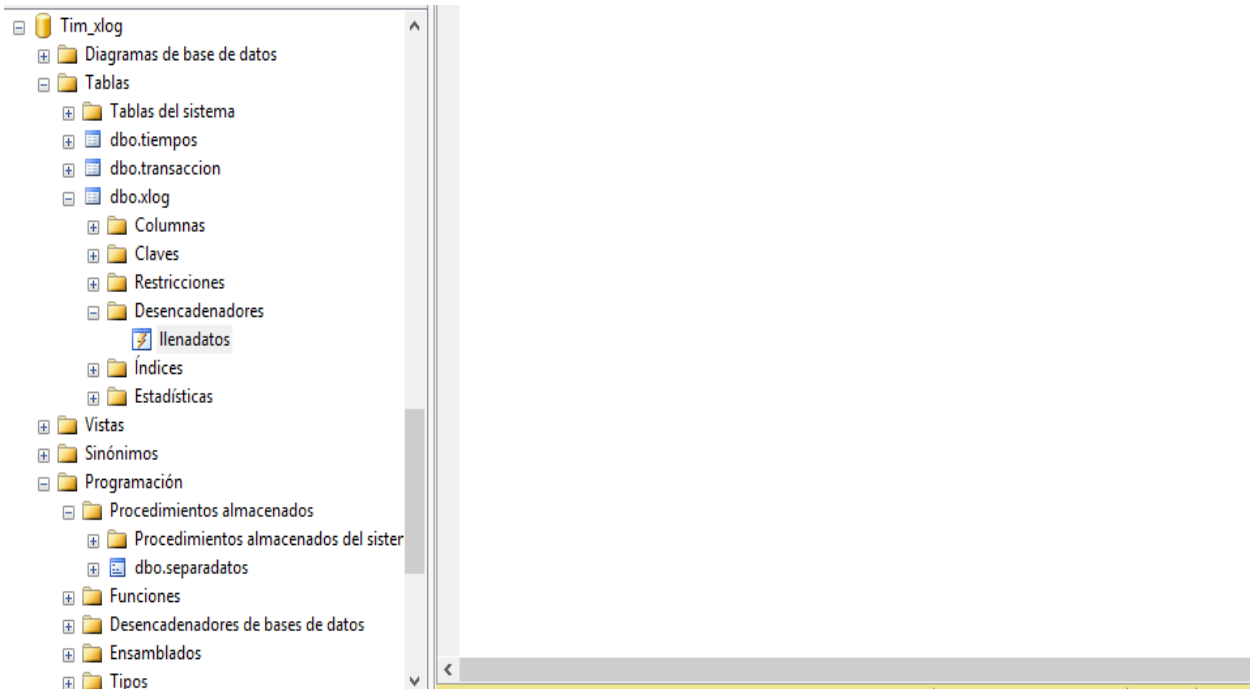


Figura 4.2.11 Árbol de procedimientos y triggers

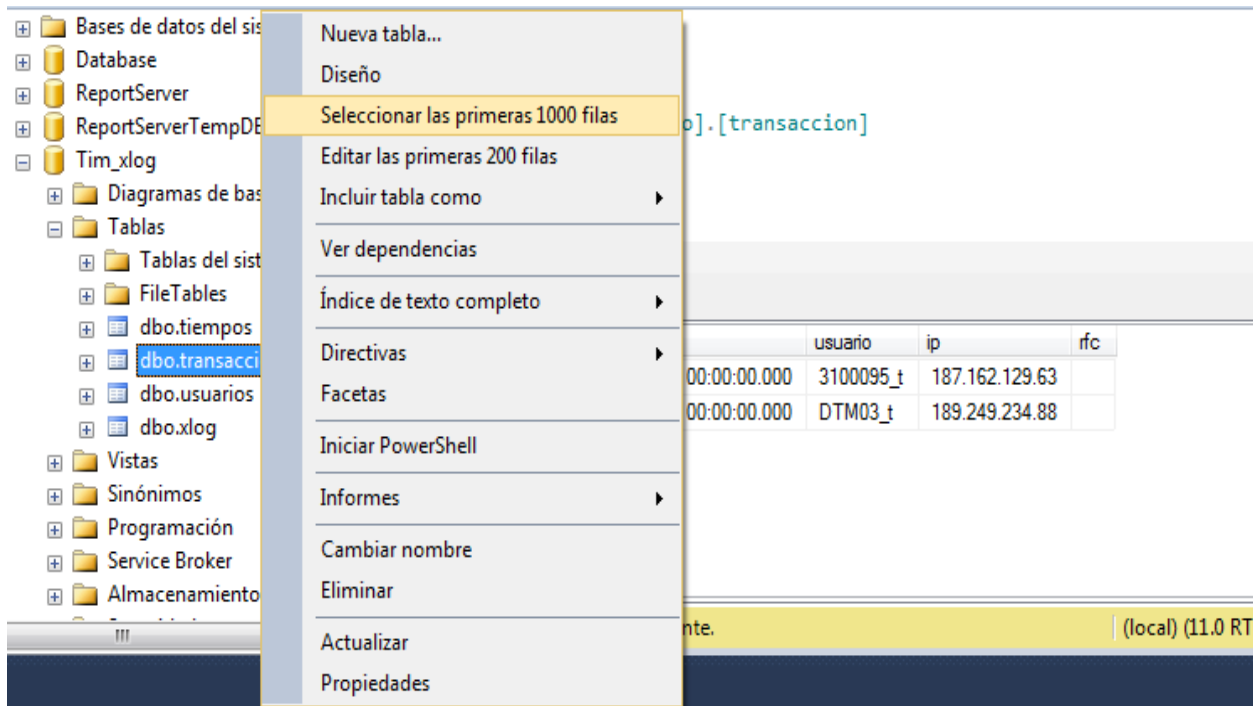


Figura 4.2.12 Consulta desde el administrador



```
C:\Users\Arturo>sqlcmd
1> use tim_xlog
2> go
Se cambió el contexto de la base de datos a 'Tim_xlog'.
1> select * from transaccion
2> go
trnsid          id      fecha
-----
usuario        rfc      ip
-----
64049125      1 2014-04-01 00:00:
00.000 3100095_t 187.162.129.63
64049127      2 2014-04-01 00:00:
00.000 DTM03_t 189.249.234.88

<2 filas afectadas>
1>
```

Figura 4.2.13 Consulta desde CMD

Importación de datos

Los datos que fueron considerados como importantes por la empresa están contenidos dentro de un archivo de texto plano con extensión “txt”, este archivo funciona como un log del Webservice de Timbrado en el cual se almacenan todas las transacciones exitosas.

La empresa no puede cambiar el proceso que sigue el Webservice de Timbrado para que inserte la información requerida directamente en una base de datos ya que esto implicaría notificar de estos cambios al SAT para que la aplicación sea recertificada por sus auditores y no se desea pasar por todo ese largo proceso nuevamente.



Para solucionar esta problemática se decidió utilizar una interfaz de importación de datos la cual leyera el archivo de texto plano e insertara la información contenida dentro de una base de datos.

El lenguaje de programación que se decidió utilizar fue Perl debido a su gran capacidad y facilidad para la lectura de archivos de texto y manejo de cadenas, se analizó el archivo de log y se decidió insertar la información en la base de datos de una forma “cruda” es decir, insertarla tal y como aparece dentro del archivo de texto, únicamente separando los campos que lo componen por el mismo carácter que los separa dentro del txt, el punto y coma (;).

Una vez teniendo los datos dentro de la base, se creó una vista para separar la información por subcampos, para quitarle las unidades de medida e incluso para realizar conversiones de tiempo ya que en el archivo de log aparecen en milisegundos y para su mayor comprensión se requirió que el tiempo fuera presentado en segundos, los datos fueron agrupados de esta manera para que su explotación resultara más sencilla y representara menos código en la interfaz que mostrará al usuario final la información.

En la figura 4.2.14 se muestra el proceso general para la importación de datos desde el archivo de texto plano hacia la base de datos:

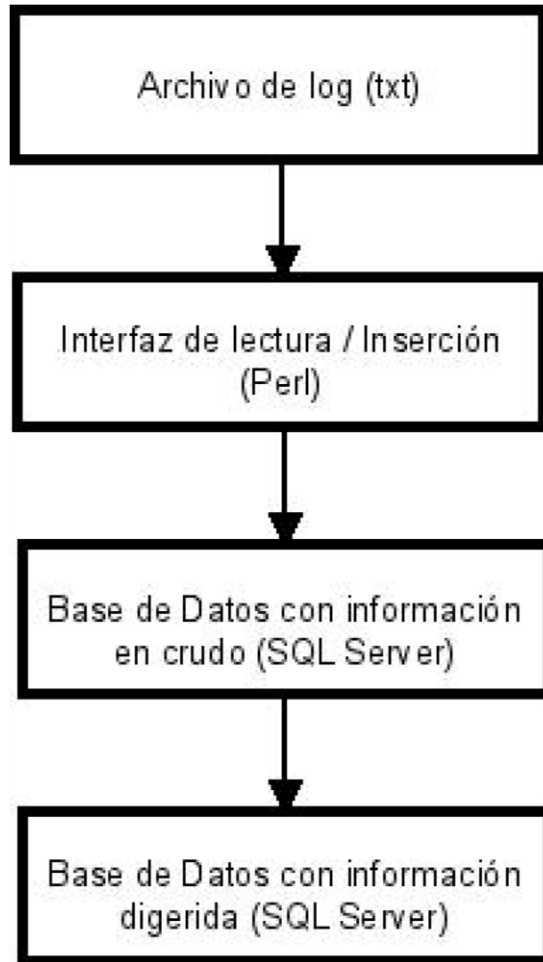


Figura 4.2.14 - Proceso general para la importación de datos

Interfaz de importación

Esta interfaz es un elemento importante para todo el sistema ya que es el módulo que va actualizando la información dentro de la base de datos, si no existiera o presentara fallas, la interfaz web no sería capaz de presentar datos en vivo.

Se requirió que la interfaz que leyera el archivo de texto fuera lo menos invasiva posible ya que si se llegara a bloquear podría afectar la disponibilidad del servicio de Timbrado ofrecido por la empresa por lo que se trabajó en un programa que únicamente tomara el archivo como de sólo lectura y que además lo estuviera



leyendo el menor tiempo posible, esto se logró leyendo el archivo por número de bytes, la primera vez que corre comienza desde el principio del archivo y la segunda desde la última línea que insertó y así sucesivamente.

En la figura 4.2.15 se muestra la lógica simplificada seguida para la construcción de la interfaz de lectura / inserción dentro de un diagrama de flujo:

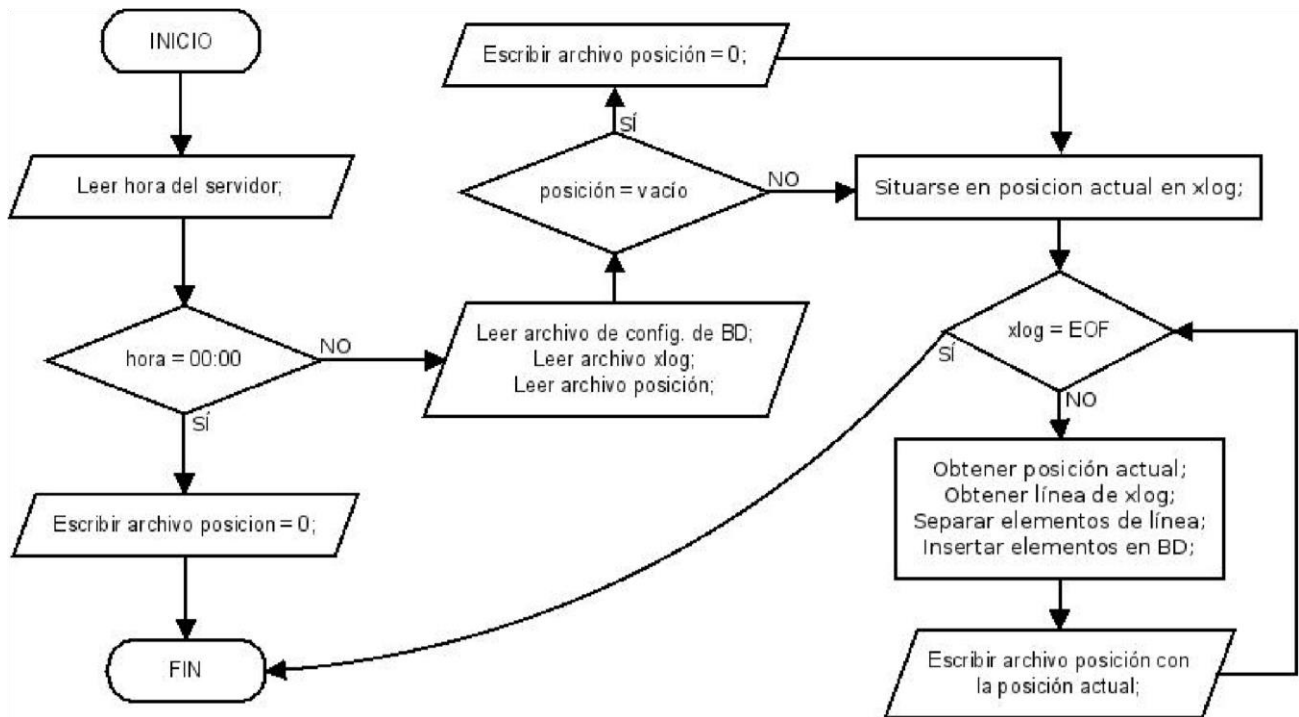


Figura 4.2.15 – Diagrama de flujo simplificado de lógica de la interfaz

A continuación se incluye el código completo de la aplicación programada y que se utiliza como interfaz de lectura / inserción del archivo de log y hacia la base de datos:

```
#!/usr/bin/perl
#Librerias
use Time::localtime;
4
use DBI;
```



```
use CGI;
#Variables
my $log="xlog.txt"; #Nombre del log
my $PosicionActual="posicion.txt"; #archivo con posicion actual en bytes
my $config="config.txt"; #Nombre del archivo de configuracion
#Obtenemos la hora del server, la guardamos y luego la separamos
$tm=localtime;
my ($day,$month,$year,$hour,$minute,$second)=$tm->mday,$tm->mon,$tm->year,$tm->hour,$tm->min,$tm->sec);
#Si la hora es 00:00
if ($hour == 00 and $minute == 00)
{
#Regresa el apuntador a cero
open APUNTADOR, "+>$PosicionActual" or die "No se puede leer ni crear
archivo ".$PosicionActual;
$posicion=0;
printf APUNTADOR $posicion;
close APUNTADOR;
}
#Si la hora es diferente de 00:00, trabaja la carga de log
else
{
#Abrimos archivo de configuracion
open CONFIGFILE, $config or die "No existe ".$config;
#Guardamos el contenido del archivo en variable
$contentido=<CONFIGFILE>;
#Separamos la cadena en los elementos de configuracion
#[0]-Servidor, [1]-Nombre de la base, [2]-Usuario, [3]-Contraseña
@config_data=split(/,/, $contentido);
#Cerramos archivo de configuracion
close CONFIGFILE;
#Conectamos con la DB
my $DSN = "driver={SQL
Server};Server=@config_data[0];Database=@config_data[1];UID=@config_data[
2];PWD=@config_data[3]";
$dbh = DBI->connect("DBI:ODBC:$DSN") or die "$DBI::errstr\n";
5
#Abrimos el archivo de log
open ARCHIVOLOG, $log or die "No existe ".$log;
#Abrimos archivo de apuntador
open APUNTADOR, "+<$PosicionActual" or die "No se puede leer ni crear
archivo ".$PosicionActual;
#Leemos archivo apuntador
#Si archivo vacio escribimos un cero
if ( -s $PosicionActual == 0 )
{
```



```
printf APUNTADOR "0";
$posicion=0;
}
#Si archivo tiene datos, se guarda dato como apuntador
else
{
seek (APUNTADOR, 0, 0);
$posicion=<APUNTADOR>;
}
#Nos colocamos en la ultima posicion leida del log
seek (ARCHIVOLOG, $posicion, 0);
#Comienza la lectura/insercion
while (<ARCHIVOLOG>)
{
#Quitamos espacio al final de la linea
chomp;
#Obtenemos posición actual
$posicion=tell(ARCHIVOLOG);
#Separamos los elementos de la linea
@elementos=split(/,/, $ _);
#Si leemos una linea que sea un espacio no la insertamos
if (@elementos[0] ne " ")
{
#Se arma el query de insercion
my $sth = $dbh->prepare("INSERT into xlog
values('@elementos[0]','@elementos[1]','@elementos[2]','@element
os[3]','@elementos[4]','@elementos[5]','@elementos[6]')");
6
#Se ejecuta el query
$sth->execute();
#Se liberan los recursos de la base
$sth->finish();
}
#Se actualiza apuntador
close (APUNTADOR);
open APUNTADOR, "+>$PosicionActual" or die "No se puede leer ni crear
archivo ".$PosicionActual;
printf APUNTADOR $posicion;
}
#Cerramos el archivo de log
close ARCHIVOLOG;
#Cerramos el archivo apuntador
close APUNTADOR;
#Nos desconectamos de la BD. Mostramos un mensaje si hay fallo
$dbh->disconnect || warn "\nFallo al desconectar.\nError: $DBI::errstr\n";
}
```



4.3 DISEÑO Y CREACIÓN DE LA INTERFAZ DE USUARIO.

El diseño de la interfaz de usuario se realizó bajo el entorno de desarrollo de visual studio 2012 desarrollando una página web utilizando código asp.net que es usado para la creación de formularios web, estos formularios se almacenan en archivos aspx y dentro de ellos se genera y se guarda código HTML y controles que se utilizan para manejar los datos estáticos de la aplicación y los datos dinámicos que genera el usuario.

El manejo de los datos proporcionados por el cliente se complementa con el uso de controles basados en lenguaje *c#*, obteniendo con esto una mayor flexibilidad del sitio web.

El diseño del sitio web se realizó por módulos, estos módulos comprenden el área de imágenes, el área de opciones, el área de graficación y el área de registro y logout.

El área de imágenes está compuesta por código HTML que permite la modificación e integración de archivos, entre las modificaciones que pueden sufrir son cambio de tamaño, cambio de posición y adición de nuevos elementos, la colocación de la imagen, así como el menú se dejaron de manera predeterminada a lo largo de todas las pantallas para que siempre se tuviera a la mano las distintas opciones que nos ofrece el desarrollo de la aplicación es decir, en todas las pantallas a las que el usuario ingrese por default seguirá el menú y la imagen institucional, en el diseño de la aplicación final se tomaron consideraciones del capítulo III de la parte de requerimientos de la aplicación y de los bosquejos de la misma. Ver figura 4.3.1.



Figura 4.3.1. Área de imágenes.

La imagen 4.3.1 requirió de un panel para la colocación de imágenes y de un menú de opciones los cuales tienen una redirección a las páginas correspondientes mediante codificación HTML y son fijas a lo largo de la navegación en la aplicación.

El área de opciones/menú está compuesto por botones los cuales nos llevan a diferentes páginas basadas en código aspx que cumplen con objetivos distintos dependiendo de las necesidades del cliente, éstas fueron diseñadas en base a los requerimientos establecidos en el capítulo III estrategia del servicio y en los bosquejos de la aplicación. Ver figura 4.3.2



Figura 4.3.2. Panel de opciones.

El área de graficación se comparte con todas las páginas que proveen reportes de los datos, éstos se dibujan en un panel que dependiendo del requerimiento del cliente muestra información distinta, esta información puede ser:

- En vivo. Muestra datos en tiempo real, se dibujan los últimos 200 registros.
- Diario. Muestra el reporte de los datos que coinciden con el día en que se



solicita el mismo.

- Mensual. Muestra y dibuja los valores de 30 días atrás.
- Histórico. Se muestra y se dibujan los datos escogidos por el usuario haciendo uso de los calendarios escogiendo la fecha inicial y final deseada.

Para el desarrollo del área de graficación, se utilizaron dos métodos distintos para ello, el primero fue un panel básico en el cual se puede dibujar gráficas de distintos tipos, pay, barras lineales etc., para el segundo tipo de panel se utilizó un timer basado en AJAX que permite que se refresque el panel y la gráfica cada cierto tiempo el cual es configurable y nos de estadísticas en tiempo real, en el caso de la opción histórico se colocaron dos cuadros de texto los cuales despliegan calendarios basados en AJAX que permiten indicar las fechas inicial y final requeridas. Ver figura 4.3.3. y 4.3.4.



Figura 4.3.3. Área de graficación.

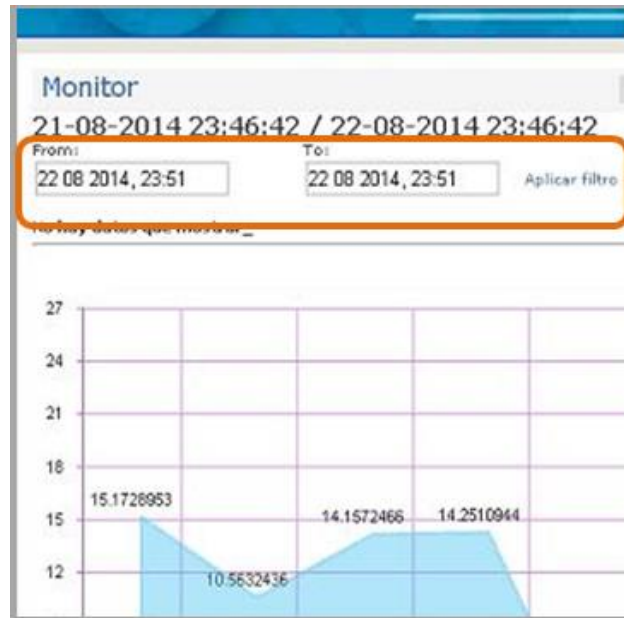


Figura 4.3.4. Graficación datos históricos.

Para la consulta de los datos de las áreas de graficación se utilizó un conector SQL para ligar la base de datos desde código asp, en cada área se configuro una consulta sql para manejar los datos de acuerdo a las necesidades del usuario.

El área de registro nos lleva a una página para dar de alta clientes nuevos solicitando:

- Nombre de usuario
- Correo electrónico
- Password y verificación del mismo.

Para el desarrollo de esta interfaz se utilizan celdas que almacenan la información ingresada por los usuarios y un botón el cual al dar click el usuario inserta los datos a una base que contiene a los usuarios de la aplicación, Ver figura 4.3.5.



The image shows a registration form titled "Registro". It contains four input fields: "Usuario:*", "Contraseña:*", "Perfil:*" (a dropdown menu with "Administrador" selected), and "Correo electronico:". Below the fields are two buttons: "Aceptar" and "Cancelar". The entire form area is enclosed in an orange rounded rectangle.

Figura 4.3.5. Área de registro.

El área de Logout nos lleva a la página principal login donde nos pedirá usuario y contraseña para ingresar nuevamente, el diseño está compuesto por una etiqueta la cual tiene una redirección mediante HTML hacia la página principal ver figura 4.3.6.

The image shows a login page titled "Acceso Monitor de Timbres". It features two input fields: "Usuario:" with the text "sa" and "Contraseña:" with masked characters "*****". Below the fields is a button labeled "Acceder". The form is set against a blue background with a grid pattern and is enclosed in an orange rounded rectangle.

Figura 4.3.6. Página de Login.



4.4 GENERACIÓN DE PRUEBAS Y MANTENIMIENTO.

Pruebas.

Las pruebas son básicamente un conjunto de actividades dentro del desarrollo de software. Dependiendo del tipo de pruebas, estas actividades podrán ser implementadas en cualquier momento de dicho proceso de desarrollo.

De esta manera definimos el concepto de prueba como: “El un proceso crítico en el que un software o uno de sus componentes se ejecutan en circunstancias previamente planificadas con el objetivo básico de encontrar errores; los resultados se observan, se guardan y se evalúan².” Ver figura 4.4.1.

Los objetivos de las pruebas son:

- Verificar y validar el software
- Descubrir defectos no detectados con anterioridad
- Encontrar defectos con poco esfuerzo y tiempo
- Encontrar defectos con una alta probabilidad

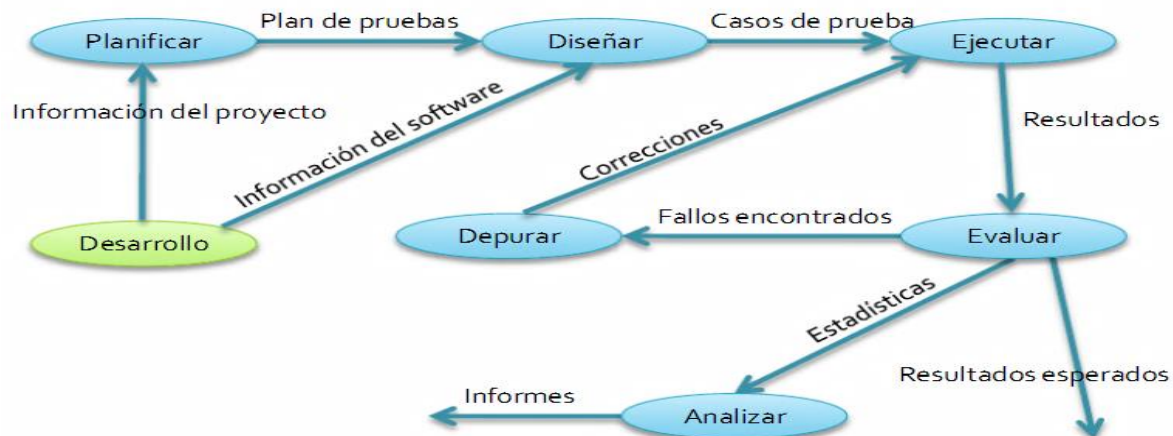


Figura 4.4.1 Ciclo completo de las pruebas.

² <http://es.scribd.com/doc/50206963/Pruebas-en-la-ingenieria-del-software>



- **Prueba de la caja blanca.**

Se denomina cajas blancas a un tipo de pruebas de software que se realiza sobre las funciones internas de un módulo. Siendo un método de diseño de casos de prueba que usa la estructura de control del diseño procedural para derivar los casos de prueba. Ver figura 4.4.2.

Las pruebas de caja blanca intentan garantizar que:

- Se ejecutan al menos una vez todos los caminos independientes de cada módulo.
- Se utilizan las decisiones en su parte verdadera y en su parte falsa.
- Se ejecuten todos los bucles en sus límites.
- Se utilizan todas las estructuras de datos internas.

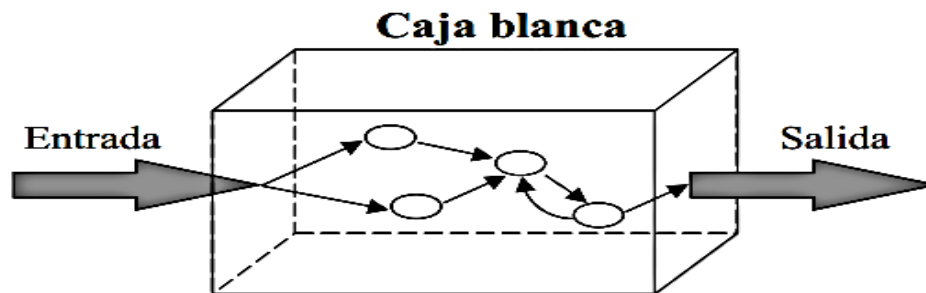


Figura 4.4.2. Diseño de Caja blanca.

- **Prueba de la caja negra.**

También conocidas como pruebas de comportamiento, estas pruebas se basan en la especificación del programa o componente a ser probado para elaborar los casos de prueba. El componente se ve como una “Caja Negra” cuyo comportamiento sólo puede ser determinado estudiando sus entradas y las salidas obtenidas a partir de ellas. No obstante, como el estudio de todas las posibles entradas y salidas de un programa sería impracticable se selecciona un conjunto de ellas sobre las que se realizan las pruebas. Ver figura 4.4.3.



Las pruebas de caja negra se llevan a cabo sobre la interfaz del software, notificando el comportamiento interno y la estructura del programa.

Los casos de prueba de la caja negra pretenden demostrar que:

- Las funciones del software son operativas
- La entrada se acepta de forma correcta
- Se produce una salida correcta
- La integridad de la información externa se mantiene

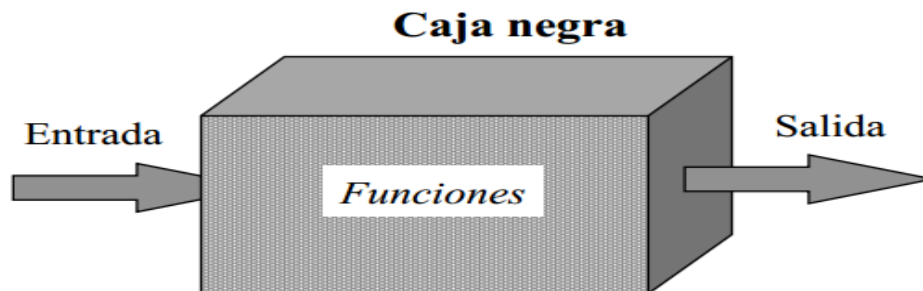


Figura 4.4.3. Diseño de caja negra.

Entre las pruebas realizadas de caja negra al sistema, se tiene en las figuras 4.4.4 y 4.4.5 las pruebas aplicadas al proceso de autenticación inicial del sistema, donde se ingresan los datos correctos e incorrectos.



Figura 4.4.4. Ingreso de datos incorrectos al sistema.



Figura 4.4.5. Ingreso de datos correctos al sistema.



- **Pruebas de regresión.**

El objetivo de las pruebas de regresión es eliminar el efecto onda, es decir, comprobar que los cambios sobre un componente de un sistema de información, no introducen un comportamiento no deseado o errores adicionales en otros componentes no modificados.

- **Pruebas de estrés.**

Esta prueba se utiliza normalmente para romper la aplicación. Se va doblando el número de usuarios que se agregan a la aplicación y se ejecuta una prueba de carga hasta que se rompe. Este tipo de prueba se realiza para determinar la solidez de la aplicación en los momentos de carga extrema y ayuda a los administradores para determinar si la aplicación rendirá lo suficiente en caso de que la carga real supere a la carga esperada.

- **Prueba alpha.**

Se lleva a cabo por un cliente en el lugar del desarrollo. Se usa software de forma natural con el desarrollador, guiando o asesorando al usuario, y registrando los errores y problemas de uso. La prueba alfa se desarrolla en un entorno controlado.

- **Prueba beta.**

Se llevan a cabo por los usuarios finales del software en los lugares de trabajo de los clientes. A diferencia de las pruebas alfa, el desarrollador no está presente normalmente. Así, la prueba beta es un aplicación en vivo en un entorno que no puede ser controlado por el desarrollador. El cliente registra todos los problemas que encuentra durante la prueba beta, e informa a intervalos regulares al desarrollador.



- **Pruebas de integración.**

La prueba de integración consiste en unir los distintos módulos que ya fueron probados de manera independiente y realizar un proceso que abarque toda la funcionalidad de los distintos módulos para certificar que la información que se procesa en uno pase correctamente como entrada a otro módulo, y a su vez, éste proporcione la información correcta al siguiente módulo, abarcando todos los módulos que contenga el sistema

- **Pruebas de módulo o unidad.**

Su enfoque es probar cada módulo o unidad de manera independiente introduciendo datos de entrada con el fin de que los procese y entregue datos de salida, preferentemente ya estimados o conocidos con antelación, para poder certificar que el módulo o unidad está operando y procesando de manera correcta. Ver figuras 4.4.6. y 4.4.7



Figura 4.4.6. Consulta correcta de datos.



Figura 4.4.7. Consulta incorrecta de datos.



Mantenimiento.

En ingeniería del software, el mantenimiento de software atiende a la modificación de un producto de software después de la entrega, para corregir errores, y mejorar el rendimiento principalmente. El mantenimiento del software es una de las actividades más comunes en sistemas, por lo que es necesario tener un control documental de todos los cambios que se soliciten y de los que se aprueben y apliquen al sistema.

La fase de mantenimiento de software es una parte explícita del modelo en cascada del proceso de desarrollo de software. En un ambiente formal la organización o equipo de desarrollo tendrán algún mecanismo para documentar, rastrear defectos y deficiencias.

El software es lanzado con esos defectos conocidos porque la organización de desarrollo en las utilidades y el valor del software en un determinado nivel de calidad compensan el impacto de los defectos y deficiencias conocidas.

Las deficiencias conocidas son normalmente documentadas en una carta de consideraciones operacionales o notas de lanzamiento (release notes), es así que los usuarios del software serán capaces de trabajar evitando las deficiencias conocidas y conocerán cuándo el uso del software sería inadecuado para tareas específicas.

Existen varios tipos de mantenimientos como son:

- **Preventivo.**

El mantenimiento preventivo se realiza a partir del momento en el que se inicia la programación y, que a través de los prototipos y de las pruebas incrementales e iterativas se detectan defectos en la funcionalidad o en el código mismo, sin embargo aún no se presenta como falla en el sistema. Esto permite realizar las modificaciones necesarias y corregir el defecto antes de que se produzca el fallo en la operativa.



- **Correctivo.**

El mantenimiento correctivo tiene lugar cuando ocurre una falla o avería una vez implementado el sistema, lo cual generalmente impacta en la operación, por lo cual este tipo de mantenimientos tienen alta prioridad y deben ser evaluados en el comité de cambios de manera muy oportuna y considerando las implicaciones del cambio.

- **Perfectivo.**

Este tipo de mantenimiento se da en los sistemas como resultado de cambios en las especificaciones iniciales con las que se originó, que normalmente, es debido a cambios en los requerimientos del negocio o de la operativa, y terminan impactando al sistema a efecto de perfeccionarlo, y adecuarlo a cada operación del negocio.

Este tipo de cambios es importante analizarlos con detenimiento, para que el cambio sea en pro de la operativa, y no haya impactos negativos o riesgos sin evaluarse detenidamente; ya que hay ocasiones en donde es necesario rehacer módulos enteros en pro del negocio y en ocasiones dependiendo de la magnitud e implicaciones del cambio, es más fácil cancelar un mantenimiento e iniciar el desarrollo de un nuevo producto de software.

- **Adaptativo.**

Consiste en la modificación de un programa debido a cambios en el entorno (hardware o software) en el que se ejecuta. Desde cambios en el sistema operativo, pasando por cambios en la arquitectura física del sistema informático, hasta en el entorno de desarrollo del software. Este tipo de mantenimiento puede ser desde un pequeño retoque hasta una reescritura de todo el código.



- **Aumentativo.**

Son las incorporaciones, modificaciones y eliminaciones necesarias en un producto de software para cubrir la expansión o cambio en las necesidades del usuario.

- **Horizontal (Hardware).**

Consiste en el mantenimiento realizado a la infraestructura de hardware sobre el cual están instaladas las plataformas que soportan la implementación del sistema. Esto puede abarcar mantenimiento en:

- Disco duro
- Memoria RAM
- Procesadores
- Instalaciones eléctricas
- Instalaciones físicas
- Instalación de redes y cableado

- **Vertical (Software).**

Es el mantenimiento enfocado al software que en conjunto establece la plataforma para la implementación del sistema. Los cambios a la distinta paquetería puede ser con respecto a:

- Cambio a una nueva versión.
- Instalación de fixes (parches de la aplicación).
- Incremento de mejoras y nuevas capacidades del software.
- Homologación de versiones.
- Cambio de sistemas base (sistemas operativos).
- Adaptación a cambios de ambiente (por ejemplo, proveedores de internet).



La fase de mantenimiento de software involucra cambios al software en orden de corregir defectos y dependencias encontradas durante su uso tanto como la adición de nueva funcionalidad para mejorar la usabilidad y aplicabilidad del software, dependiendo del objetivo de la modificación se genera la clasificación de tipos de mantenimiento. Ver figura 4.4.8.

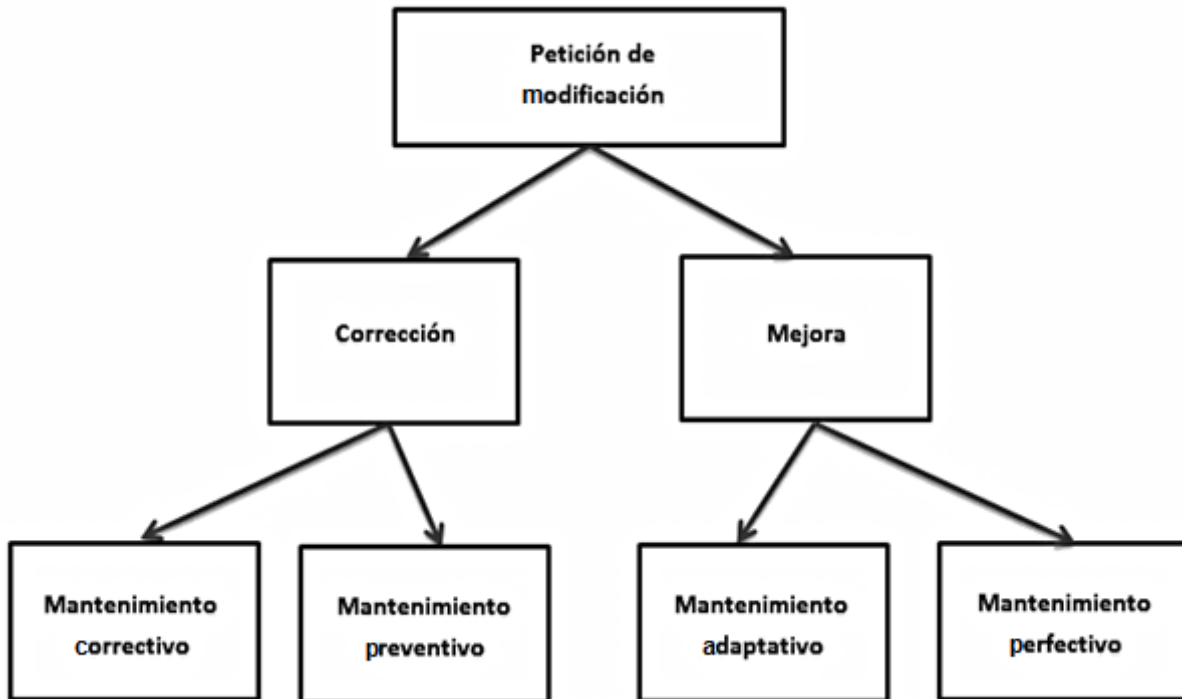


Figura 4.4.8. Tipos de mantenimientos de software.

4.5 REPORTE

Para la solución propuesta se crearon tres tipos de reportes adicionales al monitoreo, contemplándolos como otras estadísticas, las cuales son: Usuarios con más timbres, IP's con más timbres y timbres; los cuales proporcionan los tiempos de respuesta del Web services de timbrado y están basados en periodos de tiempo establecidos por el usuario.



En la figura 4.5.1 se presenta la captura de pantalla de la aplicación que muestra a los usuarios con el mayor número de timbres, es decir, una vez establecido el periodo de tiempo de consulta, un horario definido y el número de resultados (usuarios) que queremos que nos muestre, nos proporcionará y visualizará el número de timbres de cada usuario, ordenándolos de manera descendente.

En la figura 4.5.2 se presenta la captura de pantalla de la aplicación que muestra las direcciones IP con un mayor número de timbres. Estos datos estadísticos nos son sobre salientes, puesto que nos permiten visualizar desde que direcciones se genera un mayor número de peticiones, ya que desde una sola dirección IP se puede estar conectando varios usuarios para realizar el timbrado. Y estos datos nos los brinda una vez que establecemos un periodo de tiempo, así a como un número de resultados deseados.

En la figura 4.5.3 se presenta la captura de pantalla de la aplicación que muestra los tiempos de timbrado del periodo de tiempo establecido, es decir, se grafican los tiempos que tarda en completarse el proceso de timbrado en las transacciones por usuario en el periodo de tiempo seleccionado.

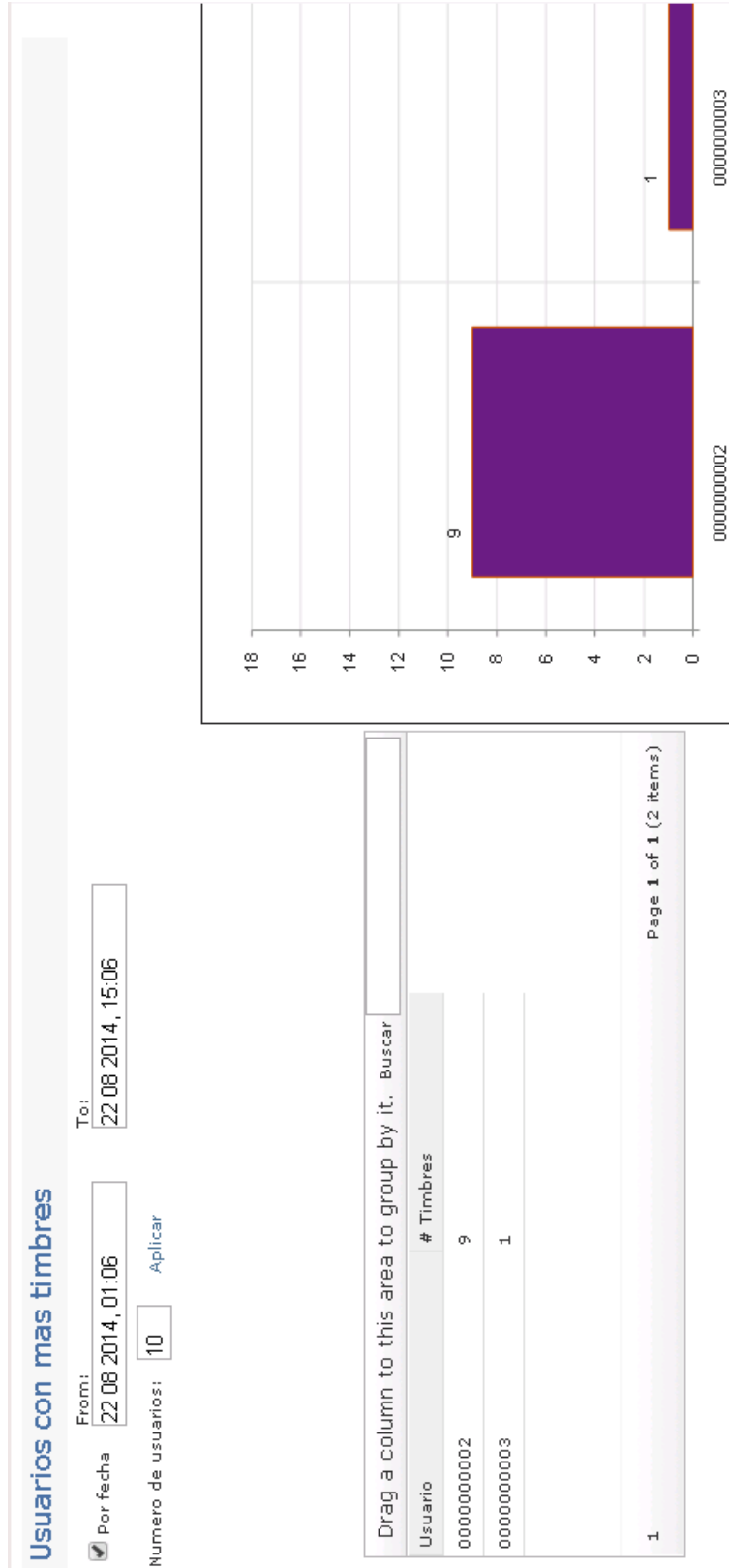


Figura 4.5.1 – Reporte de usuarios con más timbres.

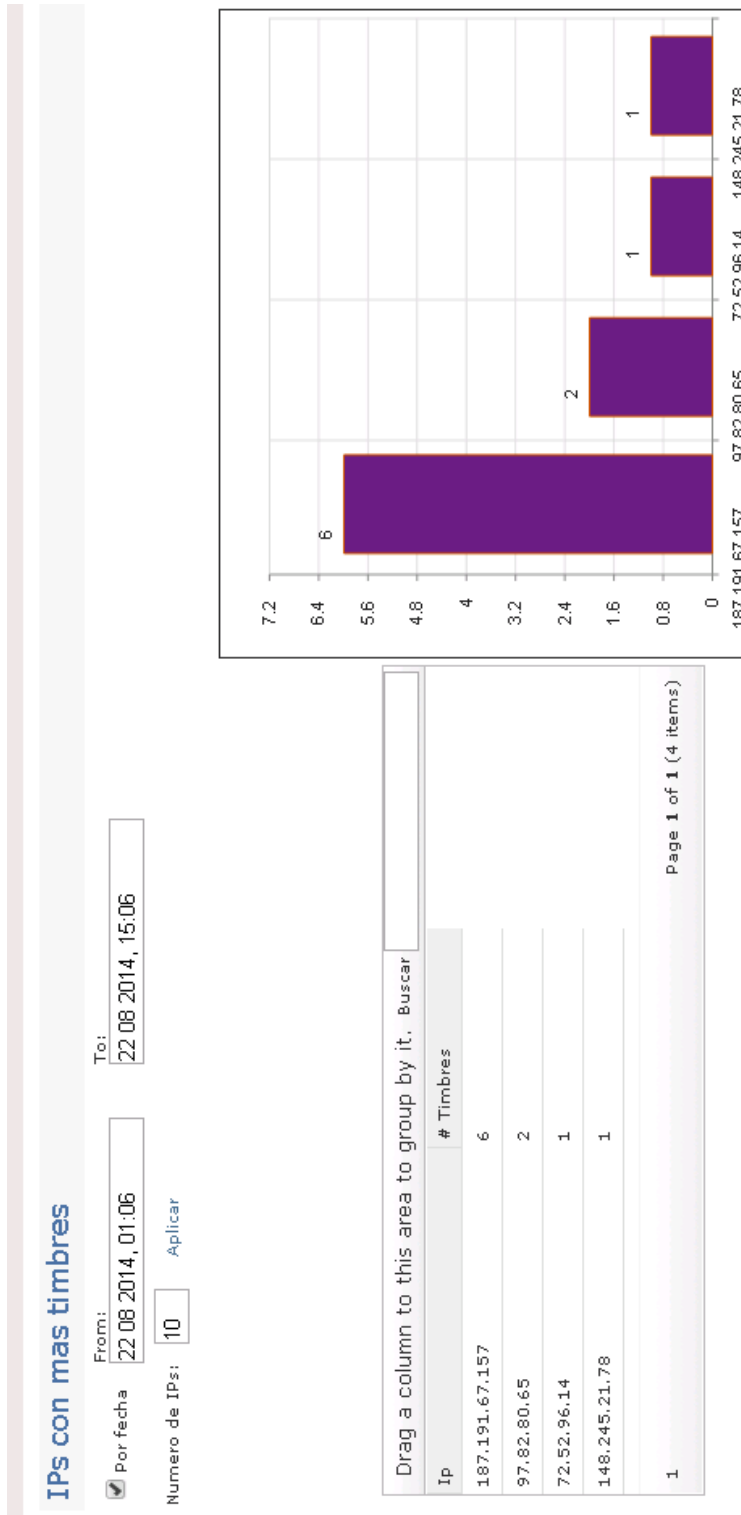


Figura 4.5.2 – Reporte de IP's con más timbres.

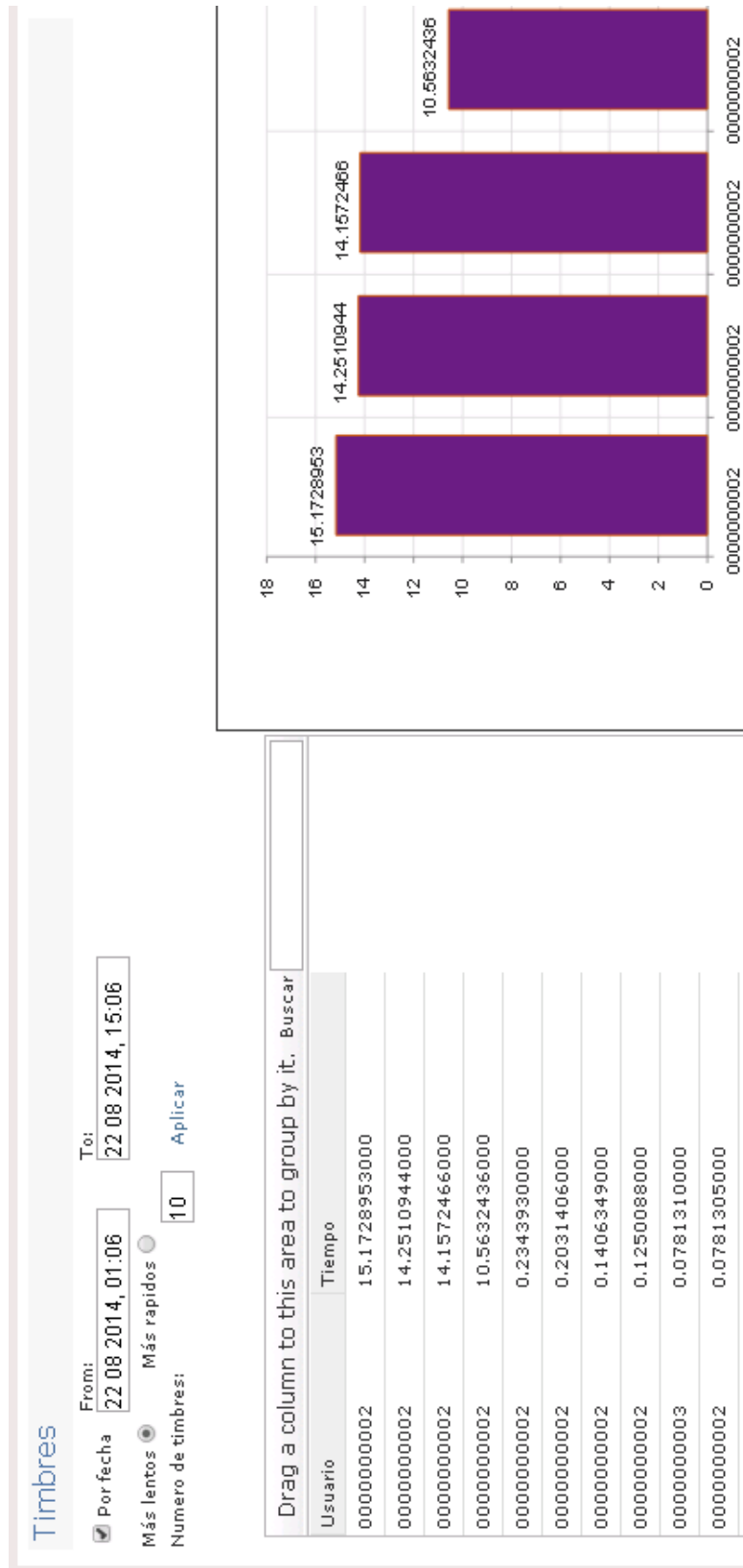


Figura 4.5.4 – Reporte de timbres.

Capítulo V

Transición del servicio.



La transición del servicio desde la perspectiva de ITIL se centra en garantizar que lo establecido en la etapa de estrategia coincida con la etapa de diseño garantizando con esto que los cambios a servicios o la implantación de nuevos se den con un mínimo de riesgo para la organización.

En el siguiente capítulo la transición del servicio ayuda a planear y coordinar la implantación del nuevo servicio minimizando el impacto que pudiera ocasionar a los servicios productivos estableciendo un plan de transición y soporte, una serie de pruebas que ayudaran a determinar la aceptación y liberación del servicio.

5.1 PLAN DE TRANSICIÓN Y SOPORTE

Una vez realizada una nueva implementación en cualquier lugar, generalmente lo más difícil es el proceso de cambio por parte del usuario final, la transición del servicio tiene como finalidad asegurar que los cambios se lleven a cabo de manera coordinada y tengan la aceptación más rápidamente por parte del usuario final.

Los objetivos de la transición es minimizar el impacto negativo de los cambios implementados, reducir el número de incidencias, interrupciones al servicio, re-trabajo causado por un cambio mal realizado, gestionar los riesgos del negocio, realizar un proceso estándar, tener una bitácora de las modificaciones y finalmente implementar el cambio de forma exitosa.

Cada cambio a implementar debe de hacerse en manera conjunta y debe de tener una implementación coordinada, además es importante tener a la mano el plan de retorno para regresar al punto anterior en caso de falla del sistema.

La información que se le debe de indicar al usuario para iniciar el proceso de transición es:

- ¿Quién requiere el cambio?
- ¿Cuál es la razón del cambio?



- ¿Cuál es el resultado esperado?
- ¿Cuáles son los riesgos?
- ¿Qué recursos son necesarios para el cambio?
- ¿Quién es el responsable de la construcción, evaluación e implementación del cambio?
- ¿Qué relación tiene este cambio con otros aplicados?

Para el desarrollo que se trata en esta tesis las respuestas a estas preguntas son: El cambio fue requerido por la dirección general de la organización debido a que se requiere un sistema de monitoreo para garantizar el correcto funcionamiento del sistema de generación de timbres, el resultado esperado es tener información suficiente para prevenir cualquier eventualidad relacionada con el servicio, los riesgos podrían ser una sobre carga de trabajo en el servidor lo cual ralentizaría el servicio final ofrecido al cliente, el cual se irá mejorando conforme se hagan las cargas iniciales y la estabilización del sistema en producción, los recursos a utilizar es el servidor de monitoreo y el personal de sistemas de la empresa quienes a su vez también son los responsables de la construcción e implementación del mismo; la evaluación del sistema estará a cargo del área de calidad del sistema, así como de dirección general, el cambio no interactuara con ningún otro cambio aplicado en la forma de trabajo de la empresa.

Para realizar la transacción es necesario la capacitación de los usuarios, a los cuales se les asignaran 2 perfiles, usuario normal y usuario administrador, a cada uno se les proporcionara el respectivo manual de usuario y una vez lo hayan leído y entendido, se procederá a realizar sesiones de laboratorio para, en conjunto con ellos, hacer el uso de la aplicación y que puedan detallar sus dudas, así mismo se va a realizar un video de la capacitación para que en el futuro en caso de que no haya parte del personal que sepa utilizar el software, se pueda tomar esta capacitación, también en base a las dudas y problemas presentados se realizará un documento con dudas y preguntas frecuentes.



Una vez liberado a producción y que los usuarios hayan sido capacitados, cualquier detalle será tratado como soporte de la aplicación, los soportes incluyen dudas, errores de usuario y errores del programa, el primer nivel de soporte se intentara resolver mediante el documento de dudas y preguntas frecuentes, en caso de que no se resuelva será responsabilidad del área de sistemas de la empresa hacer la revisión pertinente, si se determina que el error es por causa del usuario se le indicara esto y se agregara al documento de dudas, en caso de que requiera modificación de código o del sistema en general, también será responsabilidad del área de sistemas de la empresa hacer la adecuación y reportarlo en la bitácora de cambios de la aplicación para dejarlo documentado en caso de cambios posteriores.

5.2 ACEPTACIÓN Y LIBERACIÓN.

La aceptación y liberación, es la gestión de entregas y despliegues, donde éstas son las encargadas de la implementación y control de calidad de todo el software y hardware instalado en el entorno de producción. Estas mismas deben colaborar estrechamente con la gestión de cambios y la de configuración y activos TI.

La gestión de entregas y despliegues también debe mantener actualizada la biblioteca de medios definitivos (DML), donde se guardan copias de todo el software en producción, y los recambios definitivos (DS), donde se almacenan piezas de repuesto y documentación para la rápida reparación de problemas de hardware en el entorno de producción.

Las interacciones y funcionalidades de la gestión de entregas y despliegues se resumen sucintamente en el siguiente diagrama (ver figura 5.2.1):

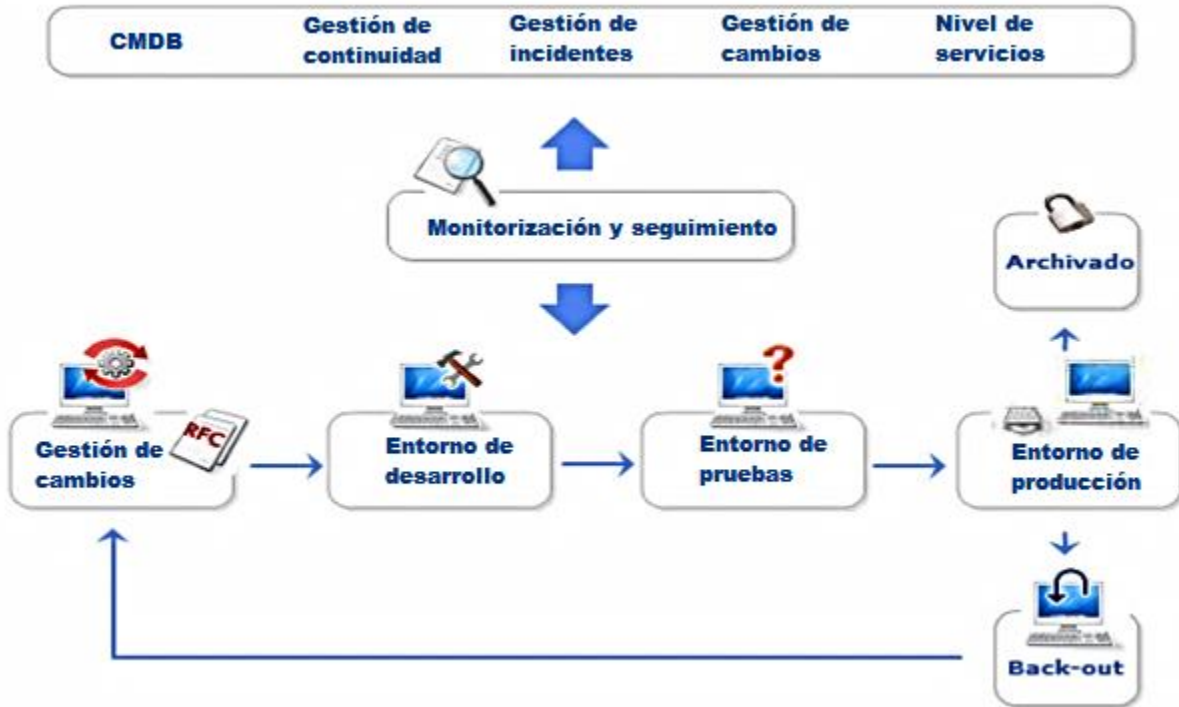


Figura 5.2.1. Gestión de entregas y despliegues.

Las complejas interrelaciones entre todos los elementos que componen una infraestructura TI convierten en tarea delicada la implementación de cualquier cambio. Si la planificación y soporte de la transición es la encargada de diseñar el plan del cambio, la gestión de cambios de aprobarlo y supervisarlo, la validación y pruebas de testear cada nueva versión, es la gestión de entregas y despliegues la que realmente pone en marcha el proceso.

Todo ello requiere de una cuidadosa planificación y coordinación con el resto de procesos asociados a la Gestión de Servicios TI. Entre los principales objetivos de la Gestión de Entregas y Despliegues se incluyen:

- Establecer una política de implementación de nuevas versiones de hardware y software.



- Implementar las nuevas versiones de software y hardware en el entorno de producción después de que la Validación y Pruebas las haya verificado en un entorno realista.
- Garantizar que el proceso de cambio cumpla las especificaciones de la Petición de Cambio (RFC) correspondiente.
- Asegurar, en colaboración con la Gestión de Cambios y la de Configuración y Activos TI, que todos los cambios se ven correctamente reflejados en la Base de Datos de la Gestión de Configuraciones (CMDB).
- Archivar copias idénticas del software en producción, así como de toda su documentación asociada, en la Biblioteca de Medios Definitivos (DML).
- Mantener actualizado el Repuestos Definitivos (DS).

Los beneficios de una correcta Gestión de Entregas y Despliegues se resumen en:

- El proceso de cambio se realiza sin deterioro de la calidad de servicio.
- Las nuevas versiones cumplen los objetivos propuestos.
- El correcto mantenimiento de la DML impide que se pierdan (valiosas) copias de los archivos fuente.
- Se reduce el número de copias de software ilegales.
- Control centralizado del software y hardware desplegado.
- Protección contra virus y problemas asociados a versiones de software incontroladas.

Las principales dificultades con las que topa la gestión de entregas y despliegues son:

- No existe una clara asignación de responsabilidades y/o la organización TI no acepta la figura dominante en todo el proceso de implementación del cambio.
- No se dispone de un entorno de pruebas adecuado en donde se puedan testear de forma realista las nuevas versiones de software y hardware.
- Hay resistencia en los diferentes departamentos a la centralización del proceso de cambio. Es habitual que existan reticencias a adoptar sistemas



estandarizados en toda la organización, sobre todo cuando ésta no ha sido la política tradicional de la misma.

- Se realizan cambios sin tener en cuenta a la gestión de entregas y despliegue argumentado que éstos sólo son responsabilidad de un determinado grupo de trabajo o que su "urgencia" requería de ello.
- La implementación sincronizada de versiones en entornos altamente distribuidos.

La solución a estos problemas pasa por:

- Un firme compromiso de la organización con la gestión de entregas y despliegues y sus responsables.
- Un adecuado plan de comunicación que informe a todos los responsables y usuarios de la organización TI de las ventajas de una correcta gestión de todo el proceso de cambio.

Las principales actividades de la Gestión de Entregas y Despliegues se resumen en (ver figura 5.2.2):

- Establecer una política de planificación para la implementación de nuevas versiones.
- Desarrollar o adquirir de terceros las nuevas versiones.
- Implementar las nuevas versiones en el entorno de producción.
- Llevar a cabo los planes de back-out o retirada de la nueva versión si esto fuera necesario.
- Actualizar la DML, el DS y la CMDB.
- Comunicar y formar a los clientes y usuarios sobre las funcionalidades de la nueva versión.

El siguiente diagrama muestra los procesos implicados en la correcta Gestión de Entregas y Despliegues:



Figura 5.2.2. Procesos implicados en la gestión de Entregas y Despliegues.

Capítulo VI

Operación del servicio.



Desde la perspectiva de ITIL la operación del servicio se enfoca a garantizar que los servicios son entregados, operados y soportados de manera correcta cumpliendo con los requisitos y condiciones acordados en la estrategia del servicio.

En el siguiente capítulo se abordaran temas relacionados con la operación del servicio para garantizar que se cumpla su funcionalidad, y que se apegue a las necesidades del negocio.

Para comprender mejor este tema se presentan los siguientes conceptos definidos por ITIL:

- Evento

Alerta o notificación creada por un servicio o una herramienta de monitoreo.

- Alerta

Algo que sucede y dispara un evento.

- Incidente

Interrupción no planeada o una reducción en la calidad de la entrega de algún servicio, es decir, cualquier evento que pueda afectar a algún servicio.

- Problema

Causa de uno o más incidentes, el momento adecuado para decir que se cuenta con un problema es hasta que se conoce la causa del o los incidentes.

- Solicitud de servicio

Petición formal de un usuario para que algo sea provisto.



6.1 EVENTOS, INCIDENTES Y ALERTAS DEL SISTEMA

Para el manejo de eventos se siguen las mejores prácticas de ITIL basándose en el diagrama que se muestra en la figura 6.1.1, en la cual se muestra el proceso a seguir para la gestión de eventos.

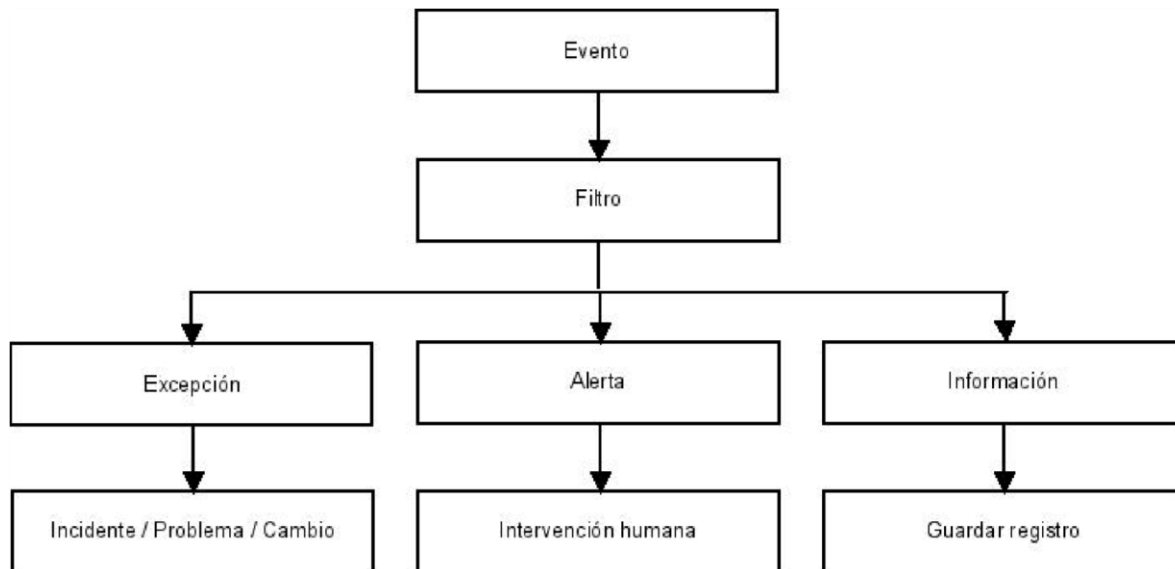


Figura 6.1.1 – Proceso para el manejo de eventos

En la solución propuesta se consideran los siguientes eventos:

- La base de datos no responde

Este evento es considerado como una alerta generada por el sistema de monitoreo con el que cuenta la empresa, una alerta sonora es disparada por la aplicación dentro de su portal y su aplicación móvil además de ser enviado un correo de notificación a las personas encargadas de la base de datos, en la solución propuesta se mostrará un mensaje de error de conexión en lugar de las gráficas.



- El portal web no está disponible

Este evento es considerado como una alerta y es generada por el sistema de monitoreo con el que cuenta la empresa, una alerta sonora es disparada por la aplicación dentro de su portal y su aplicación móvil además de ser enviado un correo de notificación a las personas encargadas de la base de datos.

- Los tiempos de respuesta del webservice rebasan los 3 segundos.

Este evento es considerado como una alerta y es generada por la propia solución propuesta, un correo es enviado a los encargados del webservice.

- Solicitudes para altas, bajas y desbloques de acceso.

Las solicitudes son hechas llegar a la persona encargada del servicio que contiene la solución propuesta, éste evento es considerado como una excepción, se lanza el proceso de cambios.

- Un usuario trata de hacer log in incorrectamente.

Este evento es considerado como informativo, se guarda en la base de datos el número de veces que el usuario ha intentado hacer log in incorrectamente desde su último acceso correcto.

- Se bloquea cuenta de usuario.

Este evento es considerado como informativo, después de 5 log in fallidos por parte de un usuario su cuenta es bloqueada dejando un indicador de ello en la base de datos.



6.2 ACCESOS, SOLICITUDES Y CAMBIOS

Los cambios y las solicitudes, incluyendo las peticiones de acceso, serán realizados a la persona encargada del área de sistemas de la empresa ya que dicha persona será la dueña y la operadora de la aplicación.

Para algún punto de la operación que no haya sido mencionado se consultará con la persona dueña de la aplicación.

Capítulo VII

Mejora continua de los siete pasos.



El ciclo de mejora de los siete pasos es la última etapa de la gestión de servicios que marca ITIL, la última de acuerdo a la estructura pero es la etapa que está presente en todos los módulos del ciclo de vida de un servicio, a diferencia de los otros la mejora continua trabaja de manera constante desde el inicio hasta el fin del proyecto o servicio.

El proceso de mejora continua se basa directamente en el ciclo de Deming el cual establece cuatro etapas que son Plan, Do, Check y Act con las que se da retroalimentación a todas las etapas del ciclo de vida permitiendo que los servicios adquieran un mayor grado de madurez. Las etapas del ciclo de Deming nos ayudan a identificar información para su futuro análisis a través de las siguientes preguntas:

- Plan- ¿Qué hacer y como lo hago?
- Do- Realizar lo planificado
- Check- ¿Se realizaron mejoras según lo planeado?
- Act- ¿Cómo mejorar otra vez?

Ambos procesos, el ciclo de Deming y el proceso de mejora continua de los siete pasos se conjuntan para trabajar durante todo el ciclo de vida de los servicios de TI.

7.1 PROPÓSITO.

El principal propósito de la mejora continua de los siete pasos es contribuir al alineamiento continuo de los servicios de TI con los objetivos de negocio a través de la identificación e implementación de mejoras y métodos para lograr la eficiencia de los procesos durante el ciclo de vida de los servicios.

7.2 OBJETIVOS.

- Hacer recomendaciones en base a análisis y revisiones de cada fase del ciclo de vida de los servicios.



- Revisar cumplimiento de niveles de servicio.
- Mejorar la calidad de los servicios de TI convirtiéndolos en valor hacia el cliente/usuario.
- Mejorar la eficiencia y efectividad de las fases del ciclo de vida de los servicios.
- Alinear en todo momento las necesidades cambiantes del negocio con la provisión de servicios de TI.
- Identificar iniciativas de mejora en los servicios y mejora en los procesos de TI.
- Identificar problemáticas relacionadas con los procesos de TI.

7.3 VALOR AL NEGOCIO.

- Mejorar la calidad de los servicios y elevar la disponibilidad de los mismos.
- Mejorar la información acerca de los servicios existentes y de las áreas de mejora.
- Mejorar la alineación con los objetivos estratégicos del negocio.
- Mejorar la comunicación entre las áreas involucradas.

7.4 PROCESO DE LOS SIETE PASOS.

Los pasos que comprende la mejora de los servicios de ITIL se aplican a detalle a continuación y se enlaza con todas las etapas del ciclo de vida de los servicios. Ver figura 7.1.5.1.

- Identificar estrategia de mejora. Se debe tener una idea clara de la estrategia a seguir para la mejora de un servicio, es decir, se debe tener claro a donde queremos llegar con las nuevas mejoras sin perder de vista los objetivos estratégicos de la organización, las metas operacionales y tácticas y la visión estableciendo una liga con las necesidades del negocio.
- Definir que se va a medir. En base a la estrategia de mejora se deben proponer métricas que ayuden a lograr la estrategia, no todas las métricas propuestas



servirán para el desarrollo de la mejora debido a que no se cuenta con herramientas para hacerlas o ya hay mediciones actualmente corriendo, pero se elegirán las que mejor se adapten a las necesidades de la organización.

- Recopilación de datos. En este paso se debe definir quién será el responsable de recopilar los datos, como se recopilaran y con qué frecuencia se hará, una vez definido esto se almacena y se prepara dichos datos para que sea preparada y accesible de una manera sencilla.
- Proceso de los datos. Para que los datos puedan ser útiles es necesario que sean pasados por un proceso en el cual sean filtrados para ser convertidos en información y que esta pueda ser analizada, este proceso requiere de herramientas que ayuden a seleccionar datos que realmente se apeguen a la estrategia previamente definida, en este punto se define la frecuencia de procesamiento, las herramientas y recursos necesarios y las plantillas de salida que tendrá la información.
- Análisis de datos. Una vez que se tiene información almacenada esta se tiene que explotar para convertirla en conocimiento, este conocimiento proveerá estadísticas de tendencias y ayudara a determinar áreas susceptibles de mejora, y al mismo tiempo ayudara a prever a corto y mediano plazo posibles problemas u oportunidades.
- Creación de informes. Una vez que se obtiene el conocimiento y se identifican problemas u oportunidades se debe realizar un informe que será presentado a las diferentes áreas involucradas, este informe deberá ser claro y de fácil lectura para evitar que sea pasado por alto, el principal objetivo de estos informes será transmitir la información a la organización y clientes con el fin de mejorar la calidad y alinear el servicio con las necesidades de negocio.
- Acciones correctivas. Una vez que la organización y los clientes han sido

Conclusiones

Durante el desarrollo de este trabajo se logró demostrar que la generación de timbres fiscales demoraba más tiempo lo cual era muy elevado de acuerdo a las necesidades de respuesta por parte de los clientes. Lo anterior causa molestia en los consumidores del servicio.

La adaptabilidad es la clave del éxito: ITIL se pudo adaptar para documentar el ciclo de vida del sistema de monitoreo, nos tuvimos que adaptar a las necesidades y requerimientos de la empresa para poder desarrollar el sistema y a su vez la empresa se tuvo que adaptar a los requerimientos del SAT para poder certificar su aplicación.

Se logró el objetivo principal que era el de monitorear el servicio de timbrado de documentos fiscales electrónicos con algo que fuera más allá de únicamente la disponibilidad.

Al terminar la implementación se obtuvo una herramienta que nos apoyó a detectar problemas específicos y se pudo trabajar en una corrección a estos, por parte de las áreas correspondientes.

Consultar una base de datos en lugar de un log para la búsqueda de información, ayudó al área encargada del servicio de timbrado de documentos fiscales electrónicos a desarrollar de una forma más eficiente sus actividades además de evitar la caída del servicio por estar consultando un log que el servicio necesita escribir al mismo tiempo.

El sistema cumple con los requisitos solicitados por la empresa; no se tuvo afectación en el servicio ofrecido, se utilizaron las herramientas disponibles y se cuenta con una base de datos históricos.

El seguir una ideología preestablecida para el manejo de proyectos ayuda bastante para el desarrollo y documentación de los mismos, en este caso fuimos guiados de la mano por ITIL.

El Programa de Apoyo a la Titulación (PAT) es efectivo y tiene buenos resultados ya que te guían en el proceso de elaboración de la tesis además, de cierta manera, presionarte para que termines lo más pronto posible.

Trabajar con personas que tienen conocimientos distintos a los que poseemos, facilita el trabajo y la comprensión de temas en los que nosotros no tenemos tanta experticia.

Los conocimientos adquiridos durante la carrera fueron de vital importancia para poder desarrollarnos en el ámbito laboral, ya que facilitan el análisis de problemas y nos dan un amplio panorama para la solución del mismo.

Nos percatamos que los módulos terminales de la carrera son efectivos, ya que durante el desarrollo del proyecto observamos que todos teníamos los mismos conocimientos pero que cada uno era especialista en su área de trabajo.

Al terminar la carrera, no ir posponer la tesis ya que el tiempo pasa y el trabajo te va absorbiendo, se vuelve complicado obtener el título; es por eso que se agradece que exista el PAT.

El desarrollo de la tesis ayudo a establecer una relación de colaboración estrecha entre los participantes de la misma, dando como resultado la distribución del conocimiento y mejorando el trabajo en equipo.

El sistema desarrollado sienta un precedente para el manejo de futuros proyectos ya que con el apoyo de la metodología utilizada se lleva un control completo a lo largo del despliegue de un nuevo servicio cambio o mejora.

Las buenas prácticas de ITIL y el apartado de mejora continua del servicio ayudará a que en un futuro se realicen cambios en el sistema creado y en los sistemas existentes en la empresa, dejando como resultado un marco de referencia para la creación de nuevos servicios.

Bibliografía



Hotek, Mike "SQL Server 2008 step by step". Microsoft press, 2009

Giardina, Fernando "Desarrollo de sitios y aplicaciones dinámicas". Maestros del web, Abril 2011

MacDonald, Matthew "Beginning ASP.NET in C#". Apress, 2010.

Poe, Curtis "Beginning Perl". John Wile & Sons, INC. 2012.

Pérez César "Microsoft sql server 2008 R2 curso práctico". Editorial Ra-Ma, 2011.

Taylor, Allen "SQL for dummies". Wiley Publishing Inc. 2010

Deitel, Harvey y Deitel Paul, "Como programar C#". Pearson. 2007

Ribas L. Joan, Web Services (2003). México: Anaya. Pág. 10 – 13.

ANEXO 20 de la Resolución Miscelánea Fiscal para 2014, publicada el 30 de diciembre de 2013.

Pressman Roger S. Ingeniería de Software, un Enfoque Practico Mc Graw Hill, México, 2006.

Soriano, Anaid. "Conseptos báscos" Arquitectura cliente Servidor. Universidad Nacional Autónoma de México, agosto 2012

Soriano, Anaid. "Servidores y clientes" Arquitectura cliente Servidor. Universidad Nacional Autónoma de México, septiembre 2012



<https://alternativas.com/2013/alternativas-a-apache-y-iis-servidores-web-similares-y-ligeros/>
<https://blog.jersson.net/12-caractersticas-de-visual-studio-2012-que-no-debes-dejar-pasar/>
<https://blog.tralix.com/2011/04/18/proceso-de-generacion-y-timbrado-de-cfdi/>
https://catarina.udlap.mx/u_dl_a/tales/documentos/lis/marquez_a_bm/capitulo_5.html
<https://cherokee-project.com/>
<https://contrerasygonzalez.blogspot.mx/2011/09/historia-del-visual-net.html>
https://docente.ucol.mx/sadanary/public_html/bd/cs.htm
<https://docstore.mik.ua/orelly/perl/prog3/>
<https://es.scribd.com/doc/50206963/Pruebas-en-la-ingenieria-del-software>
<https://es.slideshare.net/mcdrena/cherokee-11271119>
<https://estratega.org/draft-created-on-09092010-at-1833/>
<https://estrategiasdegestiondeserviciosdeti.blogspot.mx/2014/03/ventajas-y-desventajas-cobit-e-itil.html>
https://itilv3.osiatis.es/transicion_servicios_TI/gestion_entregas_despliegues.php
https://labredes.itcolima.edu.mx/fundamentosbd/sd_u1_6.htm
<https://losimpuestos.com.mx/requisitos-cfdi-2014/>
<https://msdn.microsoft.com/en-us/library/bb545450.aspx>
<https://msdn.microsoft.com/es-MX/library/67ef8sbd.aspx>
<https://msdn.microsoft.com/es-mx/library/kx37x362.aspx>
<https://noticias.facturaxion.com/2013/12/requisitos-que-debe-contener-un-cfdi.html>
<https://oskarito-hurtatiz-32.blogspot.mx/2009/08/historia-y-evolucion-de-visual-studio.html>
<https://perlanaypaty.blogspot.mx/2008/12/lenguaje-perl.html>
<https://prezi.com/cgswghzyz16e/historia-visual-studio/>
<https://quiyojararodriguez.blogspot.mx/2013/02/visual-studio-2010.html>
<https://smr2danielcortes.blogspot.mx/>
<https://www.actualisat.com/component/content/article/25-inicio/150-elementos-de-un-cfdi>
<https://www.actualisat.com/component/content/article/25-inicio/151-proceso-de->



timbrado

<https://www.amexipac.org.mx/factura-electronica/introduccion/>

<https://www.apache.org/>

https://www.dof.gob.mx/nota_detalle.php?codigo=5336743&fecha=13/03/2014

<https://www.elguille.info/NET/ASPNET/tutorialLogin/tutorialLogin.htm>

<https://www.foroz.org/visual-studio-2012-la-nueva-version-de-programacion-de-microsoft.html>

<https://www.globetesting.com/2012/10/nuevas-caracteristicas-de-visual-studio-2012/>

https://www.historianaval.cl/publico/publicacion_archivo/publicaciones/2_4.pdf

<https://www.idconline.com.mx/fiscal/2013/10/17/como-se-elabora-una-factura-electronica>

<https://www.iis.net/>

<https://www.magazcitum.com.mx/?p=50>

<https://www.microsoft.com/es-es/server-cloud/products/sql-server/>

<https://www.mysql.com/>

<https://www.oracle.com/es/technologies/java/overview/index.html>

<https://www.perl.org>

<https://www.postgresql.org/es/>

<https://www.python.org/>

<https://www.ruby-lang.org/es/>

https://www.sat.gob.mx/informacion_fiscal/factura_electronica/Documents/cfdi/PyRFactElect.pdf

https://www.sat.gob.mx/informacion_fiscal/normatividad/Paginas/resolucion_misclanea_fiscal_2014.aspx

https://www.soporteremoto.com.mx/help_desk/articulo04.html

<https://www.uv.mx/personal/jfernandez/files/2012/11/ITIL.pdf>

<http://docs.componentart.com/>

<http://www.aspsnippets.com/Articles/ASPNet-AJAX-Pie-Chart-Control-Populate-from-Database-example.aspx>

<http://jvascriptrules.com/2009/06/30/limitation-on-call-stacks/>