



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Sistema para Comprobantes Fiscales Digitales por Internet

Tesis Profesional
para obtener el título de
INGENIERO EN COMPUTACIÓN

PRESENTAN

Aguilera Hernández Odiseo
Gutiérrez Delgado José Eduardo
Martínez Flores María Yuria
Quintana Gómez Luis Daniel
Vázquez Otero Javier David

ASESOR DE TESIS

M.I. Juan Carlos Roa Beiza



Ciudad Universitaria, México, Noviembre 2014.



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

AGRADECIMIENTOS

Agradezco a **Dios**, por iluminar mi camino y permitirme cerrar este ciclo en mi vida.

Agradezco a mi compañero, amigo y querido esposo, a mi amada hija Victoria, a mi amado hijo Jesús, a mi **hermosa madre**, a mi fallecido padre, a mis hermanos, hermanas, amigas y amigos, por su incondicional apoyo, a todos los llevo en el corazón.

Agradezco a mis compañeros y amigos de tesis, un placer trabajar con todos y cada uno de ellos.

Agradezco a nuestro director de tesis, y a todos los profesores que contribuyeron en mi formación como profesionista.

María Yuria

AGRADECIMIENTOS

Madre.

Difícil es rememorar toda la paciencia, empeño y amor que has expresado en cada uno de los actos que has realizado en la vida por mí y por mis hermanos; para tenerlos ahora presentes haría falta cientos de libros iguales a este. Que sea este trabajo final –Punto final en el título profesional “Ingeniero” por el que tanto nos esforzamos– la expresión de mi insuficiente y más sincero agradecimiento a la valentía y fuerza que le pones a la vida y de la cual me cobijé para poder llegar hasta aquí. No puedo devolverte tus desveladas pero ten por seguro que (casi) todas ellas fueron bien aprovechadas.

Sarah.

Que difícil es escribirte algo cuando usualmente todo lo que tengo que decirte te lo digo con la mirada; tu eres la persona más importante en este muy significativo corte de caja en el largo, y en ocasiones tortuoso, camino de preparación profesional. Apareciste hace tiempo y terminaste siendo la compañera de mi vida. Con tu esfuerzo, tus palabras de aliento, tus regaños, tus expectativas sobre mí y, sobre todo, el deseo de que tengamos una hermosa familia, has permitido que logre alcanzar esta difícil meta, por todo ello tú también estas presente de muchas maneras en este trabajo. Para ti todo mi amor y agradecimiento.

Abuelos Augusto y Cesárea, tío Augusto.

La casa de los abuelos fue el lugar donde siempre me sentí seguro y cobijado, ya sea por mis abuelos, ya sea por mi querido tío. Quiero que sepan que sus cuidados y enseñanzas hicieron que yo pudiera llegar hasta aquí y que por lo mismo no puedo tener más que palabras de agradecimiento porque entre ustedes

nunca me faltó nada. Gracias por enseñarme el valor del trabajo y la honestidad. Gracias por tener tanta paciencia y gracias por tener para mis hermanos y para mí una palabra de aliento y guía. Sin ustedes esto hubiera sido casi imposible.

Luis, Oswaldo, Luis Damián.

La vida nos enseña –a veces a golpes muy duros- que cualquier camino que tomemos siempre será difícil pero al final debemos concluir todo lo que iniciamos. Que este trabajo sea muestra de que el mayor de los esfuerzos merece la mayor de las recompensas y que siempre debemos ir por más porque esa es nuestra naturaleza, siempre hacer el mejor esfuerzo. Gracias a ustedes también porque en ustedes encuentro una familia de la cual me puedo sentir orgulloso y complementado. Cuando leas esto Damián, piensa que tus tíos y tu papá siempre estarán para ayudarte, espero pueda ser un buen ejemplo para ti cuando tomes tus propias decisiones.

Al Programa de Apoyo a la Titulación

Por haberme brindado la oportunidad de desarrollar este trabajo dentro del Palacio de Minería de la Facultad de Ingeniería de la UNAM. A mis compañeros: Yuria David, Daniel y Odiseo por haberme acompañado en esta aventura y al Ingeniero Juan Carlos Roa por habernos proporcionado su experiencia y dedicación. Gracias.

Eduardo

A mi Mami:

Quien durante su estancia en este mundo, me enseñó la importancia de ser un hombre de bien, apoyándome en cada paso que he dado y los siguientes pasos que tenga que dar en este mundo o en cualquier otro, que seguro seguirá a mi lado de cualquier forma posible, quien me dio la vida y las mejores alegrías que se puedan sentir, adorando a mi padre y hermana de la misma manera que lo hizo conmigo, quien me deja sin aliento cada 24, pase el tiempo que pase, te dedico este logro con todo mi amor y sé que donde quiera que estés estarás orgullosa de él; ¡Te amo Madre!.

A mi Padre:

A quien amo y respeto más que nada en el mundo, por saber guiarme por la vida que tiene tantos bemoles, por enseñarme a superar todos los obstáculos, por creer en mí, por dejarme tomar mis decisiones y apoyarlas, por saber darme principios y educación, por apoyarme en esta gran etapa de mi vida donde cierro un ciclo importante, y lo más sorprendente de todo, saber ser papá y mamá al mismo tiempo, gracias por ser el mejor padre del mundo, por habernos amado a mi Mami, Hermana y a mí como lo más grande de este mundo, te dedico este trabajo como uno de tantos logros que te he dedicado y te seguiré dedicando; ¡Te amo Padre!.

A mi Hermana:

Quien es mi mejor compañera, mi amiga, mi confidente y una que otra ocasión como mi madre misma, que ha compartido conmigo el sufrimiento y alegría de nuestra familia, quien a pesar de ser tan diferentes nos amamos con todo el corazón, y lo mejor de todo que eres mi hermana, la mejor de este mundo, gracias

por el apoyo incondicional en todo este tiempo, espero que te guste está dedicatoria por que fuiste parte importante para poder lograrla; ¡Te amo Hermana!.

A mi Pareja:

Quien me acompaña por la vida estando a mi lado y apoyándome en cada decisión que he tomado, rodando a lugares que jamás pensaste, volando por los aires y disfrutando de lo que me apasiona, así como impulsándome a poder terminar esta gran etapa, siendo parte importante de este logro; ¡Te amo Amor!.

Azul:

Espero que algún día tengas curiosidad de leer estas líneas, donde podrás ver que siempre estas presente para mí, que te amo con todo el alma y mi corazón, que sepas que no hay imposibles, que siempre se pueden cerrar ciclos sin importar los obstáculos, que es importante la familia, los amigos, pero lo mejor de todo que hagas lo que te hace feliz, y que también eres y serás parte de mis logros; ¡Te amo Hija!.

Odiseo Aguilera Hernández

AGRADECIMIENTOS

Siempre quise llegar a este momento de escribir estas líneas, ahora me doy cuenta que es difícil realmente agradecer a las personas indicadas, ya que a lo largo de toda mi vida ha habido personas que dejaron una huella imborrable en mi formación personal.

Mis padres, Angélica Gómez Tinajero y Salvador Quintana Hernández como pilar de mis valores y de mi educación, quiero expresarles el orgullo y admiración que tengo hacia ustedes. Siempre serán mi ejemplo a seguir y espero se sientan orgullosos por esta meta alcanzada, estaré agradecido eternamente por el enorme sacrificio.

A mis abuelos, Heriberto Quintana, Mercedes Hernández, José Luis Gómez y Elvira Tinajero que siempre me inculcaron ser un hombre de bien y serán siempre un ejemplo de fortaleza y dedicación.

A mi familia en general, mis hermanas Cynthia Quintana Gómez, Miriam Quintana Gómez, mis tíos y tías, mis primos, por todo su apoyo a lo largo de todos estos años.

A mi futura esposa, Fanny Muñoz García, por apoyarme en estos años de mi vida, por ser un ejemplo para cumplir mis metas y por compartir su vida conmigo, Te Amo.

A mis amigos, que más que amigos los considero mis hermanos, Carlos Hernández, Carlos Jacinto, Juan Carlos López, Luis Alberto López, Miguel Ángel Ramírez y Alejandro Meléndez, por apoyarme a lo largo de mi carrera y ser mis compañeros de aventuras.

A mis compañeros de tesis David Vázquez, Eduardo Gutiérrez, Odiseo Aguilera y Yuria Martínez que siempre mostraron compromiso, trabajo en equipo y que sin ellos no hubiera sido posible realizar este trabajo.

Al profesor Juan Carlos Roa, por dirigir esta tesis con gran profesionalismo y transmitirnos su experiencia para concluir este proyecto.

Luis Daniel Quintana Gómez

Agradecimientos

Quiero expresar mi más sincero agradecimiento a:

A mi madre

Por darme la vida, por estar presente en todos los momentos difíciles, apoyándome a buscar la mejor solución a ellos. Por darme la oportunidad de iniciar, continuar y finalizar mis estudios.

Eres mi apoyo incondicional y mi principal motivación para seguir adelante. Te lo agradezco de todo corazón mamá. Te quiero mucho.

A mi hermana

Por tus consejos, por estar ahí cuando siempre te he necesitado, acompañarme en las buenas y en las malas. ¡Gracias hermanita!

A mi padre

Por sus enseñanzas, por guiarme con tus consejos. Estar presente en todos los momentos importantes de mi vida.

Al PAT

También quiero agradecerle a nuestro asesor de tesis, el M.I. Juan Carlos Roa Beiza, por sus consejos, por su apoyo esencial para alcanzar este sueño.

A mis compañeros

Quiero agradecerle a mis compañeros del seminario: Yuria Martínez, Eduardo Gutiérrez, Daniel Quintana y Odiseo Aguilera. Su amistad y sobre todo por su infalible confianza y apoyo. Juntos compartimos durante este proceso las mismas emociones, el mismo esfuerzo, el mismo sueño. Juntos con nuestras diferentes habilidades organizamos un excelente equipo. ¡Lo logramos chicos!

ÍNDICE TEMÁTICO

CAPÍTULO I ANTECEDENTES.....	1
1.1 INTRODUCCIÓN.....	2
1.2 CONCEPTOS GENERALES DE LOS TÉRMINOS USADOS POR EL SAT	4
1.3 NORMATIVIDAD QUE DEBE SER CUMPLIDA POR LOS PACS.....	7
1.4 DESCRIPCIÓN BÁSICA DE LO QUE DEBE DE CONTENER UN CFDI.....	10
1.5 ACOTAMIENTO DEL PROBLEMA A RESOLVER.....	17
CAPÍTULO II MARCO TEÓRICO.....	23
2.1 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE LAS BASES DE DATOS RELACIONALES Y METODOLOGÍA DE YOURDON/DE MARCO	24
2.2 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE LA PROGRAMACIÓN ORIENTADA A OBJETOS.....	30
2.3 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DEL ENTORNO JAVA,	37
2.4 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DEL MANEJADOR DE BASES DE DATOS MYSQL.....	41
2.5 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DEL AMBIENTE CLIENTE-SERVIDOR	46
CAPÍTULO III PLANTEAMIENTO DEL PROBLEMA.....	59
3.1 ANÁLISIS DEL PROBLEMA.....	60
3.2 REQUERIMIENTOS GENERALES Y PARTICULARES DE LA APLICACIÓN	66
3.3 ANÁLISIS Y POSICIONAMIENTO DE LOS LENGUAJES PARA EL BACK-END Y FRONT- END	69
3.4 APLICACIÓN Y SEPARACIÓN DE LOS MÓDULOS SEGÚN LA METODOLOGÍA DE YOURDON/DE MARCO.....	86
3.5 DISEÑO A NIVEL DE BLOQUES DE LA INTERFAZ DE USUARIO.....	89
CAPÍTULO IV ANÁLISIS Y CONSTRUCCIÓN DE LA APLICACIÓN	93
4.1 MODELADO ESENCIAL DE GENERACIÓN DEL SISTEMA DE GENERACIÓN DE COMPROBANTES FISCALES DIGITALES POR INTERNET (SCFDI)	94
4.1.1 DIAGRAMA DE CONTEXTO DEL SISTEMA SCFDI	96
4.1.2 NOTACIÓN DEL DICCIONARIO DE DATOS.....	98
4.1.3 LISTA DE EVENTOS DEL SISTEMA SCFDI	102
4.1.4 DIAGRAMA DE ENTIDAD-RELACIÓN DEL SISTEMA SCFDI.....	103
4.1.5 DICCIONARIO DE DATOS DEL SISTEMA SCFDI	104
4.1.6 DIAGRAMA DE FLUJO DE DATOS DE RESPUESTA A EVENTOS DEL SISTEMA SCFDI.....	106
4.1.7 ESPECIFICACIÓN DE PROCESOS DEL SISTEMA SCFDI.....	114
4.1.8 MODELO DE IMPLEMENTACIÓN	120
4.2 CREACIÓN DE LA BASE DE DATOS E IMPLEMENTACIÓN DEL SISTEMA SCFDI	185
4.3 CREACIÓN DEL FRONT-END CON LA FUNCIONALIDAD PEDIDA POR EL SAT	196
4.4 GENERACIÓN DE PRUEBAS Y MANTENIMIENTO	209
4.5 GENERACIÓN DE CFDI DE PRUEBA Y EVALUACIÓN DE LOS MISMOS	238
CONCLUSIONES.....	243
BIBLIOGRAFÍA.....	247

CAPÍTULO

I

ANTECEDENTES

ANTECEDENTES

1.1 INTRODUCCIÓN

A medida que la tecnología informática ha evolucionado, ha hecho posible que los equipos de cómputo y el uso de Internet estén más accesibles al público en general, y debido también al avance en temas de seguridad de la información, muchas Instituciones privadas y gubernamentales han sabido aprovechar las ventajas que ofrece la tecnología informática para simplificar sus procesos, optimizar costos y llevar un mejor control en sus tareas administrativas.

Esta evolución tecnológica ha permeado muchas áreas y campos de trabajo, y algunas instituciones han sufrido cambios importantes en los últimos años, tal es el caso de la Secretaría de Hacienda y Crédito Público que a través del Servicio de Administración Tributaria (SAT) ha venido publicando en los últimos años cambios considerables en la forma de manejar los comprobantes fiscales.

El 7 de Diciembre de 2009 se aprobó en el Diario Oficial de la Federación (DOF) los cambios y modificaciones relativas a la emisión de comprobantes digitales que entraría en vigor en enero de 2011, un comprobante fiscal digital en realidad es una factura electrónica, la cual está apegada a los estándares definidos por el SAT en el Anexo 20 de la Resolución Miscelánea Fiscal, en el cual se estipula que una factura electrónica debe contar con un sello digital (Firma Electrónica Avanzada) que corrobora su origen y la validez ante el SAT, una cadena original que funciona como un resumen del contenido de la factura, y un folio que indica el número de la transacción. Durante esta primera fase el CFD se generaba por medios propios, sin embargo, el 14 de septiembre de 2010 se publica en el DOF la creación del Comprobante Fiscal Digital por Internet (CFDI), con este cambio se integran a este proceso los Proveedores Autorizados de Certificación (PAC), que son proveedores que certifica el SAT para que puedan generar los timbres digitales y la asignación de folios fiscales. Hasta el año 2013 las facturas impresas fueron válidas y

también convivieron con el CFD y el CFDI. Pero a partir de este año es obligatorio para los contribuyentes emitir Certificados Fiscales Digitales por Internet (CFDI).

Esta nueva disposición crea la necesidad para muchas empresas de poder contar con sistema que les permita emitir estos CFDI's. Basado en esta necesidad actual es que surge nuestro trabajo de tesis.

Nuestro trabajo de tesis estará conformado por varios capítulos, en el capítulo I abordaremos el tema de la disposición oficial del SAT relacionada con los CFDI's, los detalles técnicos, reglamentos y estatutos, y medios por los que se dan a conocer estos reglamentos, así como la normatividad que rige este tema de Comprobantes Fiscales Digitales por Internet.

En el capítulo II se revisará el Marco Teórico en el que fundamentamos nuestro trabajo de tesis, revisaremos los aspectos técnicos de las herramientas y metodología que utilizaremos para el desarrollo de nuestra aplicación, evaluando las características, ventajas y desventajas de dichas herramientas y lenguajes de programación.

En el capítulo III se realizará el planteamiento del problema, se revisará de manera detallada las posibles soluciones, se discutirán los pros y contras de cada solución planteada, y se irá preparando la información que servirá como base para la aplicación de la Metodología de Análisis y Diseño Estructurado de Yourdon para el desarrollo de nuestro sistema. En esta parte deberán quedar bien documentados los requerimientos del usuario y acotar los alcances del sistema a desarrollar.

En el capítulo IV se concretará la parte del análisis, diseño y construcción de nuestra aplicación, y nos aseguraremos de que la documentación que se genere sea correcta, completa, y entendible por parte de todos los involucrados, para evitar posibles errores u omisiones.

Adicionalmente se desarrollarán los principales aspectos del sistema, por un lado se debe especificar ¿qué debe hacer el sistema? y por otro lado el ¿cómo se

implementará? Este capítulo está apegado a la metodología de Yourdon por lo que se divide básicamente en dos modelos:

- Modelo esencial.- Este modelo ilustra ¿qué debe hacer el sistema?, excluyendo los detalles de ¿cómo debe ser implementado?, este modelo está abarcando la parte del análisis y está conformado a su vez por dos modelos: 1) modelo esencial ambiental y 2) modelo esencial de comportamiento.
- Modelo de implementación.- El cual especifica ¿cómo se implementará el Sistema en forma tangible?, en base en el comportamiento descrito en el modelo esencial dentro de la etapa de análisis

1.2 CONCEPTOS GENERALES DE LOS TÉRMINOS USADOS POR EL SAT

En este capítulo se describen conceptos básicos acerca de los términos usados en el SAT utilizados en el presente proyecto de tesis.

Factura Electrónica.- Se utiliza el término Factura Electrónica o Comprobante Fiscal Digital por Internet indistintamente tanto para este proyecto como dentro de la documentación emitida por el SAT.

Comprobante Fiscal Digital por Internet - Un CFDI Es el mecanismo obligatorio de comprobación fiscal de ingresos, egresos y propiedad de mercancías en traslado por medios electrónicos, el cual hace referencia el artículo 29, primer párrafo y fracción IV del CFF vigente a partir de enero de 2011, cuya particularidad es la remisión del mismo a un proveedor de certificación de CFDI autorizado, para su validación, asignación del folio e incorporación del sello digital del SAT.

Emisor de facturación - Toda persona física o moral está obligada a emitir una factura al momento de realizar alguna operación económica. En este sentido, esta

persona se convertirá automáticamente en el tributario emisor de la factura. Al ser el emisor de la factura, esta persona se ve en la obligación de completar la información básica sobre sí mismo.

Receptor de facturación - El receptor es el encargado de validar las facturas electrónicas o de otra manera no se podrán hacer válidas fiscalmente. Si ellos no aceptan los comprobantes porque hay algún error, los emisores deberán corregirlo.

En el caso que la factura tenga como destinatario a una persona física es necesario completar el nombre y apellidos de esta persona. En el caso que el destinatario sea una persona moral es necesario indicar la denominación social de la misma. Para que una factura sea válida, los datos del emisor y del receptor deben ser completados correctamente.

Impuesto al valor agregado (IVA) - El IVA es un impuesto de bienes o consumo, esto significa que justamente es un impuesto que se estará pagando por el valor que agreguemos a los productos o servicios que hayamos adquirido.

Impuesto trasladado - El IVA trasladado es un impuesto cobrado por una empresa por ventas o ingresos de contado o crédito siempre y cuando estas últimas ya hayan sido liquidadas. Este impuesto se carga a la empresa cada que se devuelve mercancía en su parte proporcional de IVA y se abona cada vez que el negocio o empresa vende un bien o presta un servicio.

Impuesto retenido - La retención es la cantidad que se retiene de un sueldo, salario u otra percepción para el pago de un impuesto, de deudas en virtud de garantizar su declaración.

Impuesto sobre la renta (ISR) - Es un impuesto directo sobre la ganancia obtenida; es decir, por la diferencia entre el ingreso y las deducciones autorizadas obtenido en el ejercicio fiscal.

Pago parcial - Un pago parcial se le dice a cada uno de los importes desembolsados en una obligación a plazo. Un pago parcial equivale a que el

cliente quiere pagar más que la cantidad mínima del plazo, pero menos que la cantidad para pagar el total del documento.

Pedimento - Un pedimento es una especie de declaración fiscal relativa al cumplimiento de obligaciones tributarias en materia de comercio exterior, por medio del cual el importador o exportador manifiestan a la aduana, la mercancía a importar o exportar.

Proveedor autorizado de certificación (PAC) - Es quien cuenta con la autorización del Servicio de Administración Tributaria para generar y procesar fuera del domicilio fiscal de quien lo contrate, los comprobantes para efectos fiscales por medios electrónicos y de manera completamente digital. Dicho proveedor efectúa el proceso de certificación de un comprobante fiscal digital a través de Internet o factura electrónica, mediante la validación, asignación de folio e incorporación del sello digital del SAT.

Régimen fiscal - El régimen fiscal es el conjunto de las normas e instituciones que rigen la situación tributaria de una persona física o jurídica. Se trata, por lo tanto, del conjunto de derechos y obligaciones que surgen del desarrollo de una determinada actividad económica.

Sello digital - Es una firma única de la factura electrónica el cual señala la relación entre emisor y receptor y que permite a la autoridad de certificación comprobar que no han falsificado los datos del comprobante.

El sello digital es una serie de caracteres que se forma como resultado de encriptar la información de la cadena original del comprobante, lo que hace que el comprobante sea infalsificable ya que cualquier cambio en los datos, generaría un sello diferente al original.

Tasa - Las tasas son contribuciones económicas que hacen los usuarios de un servicio prestado por el estado.

Timbre fiscal o UUID - Es uno de los componentes obligatorios que debe de contener su nueva versión de facturación electrónica. Es un estándar el cual es un

identificador único basado en las políticas y lineamientos estipulados por el SAT para el Folio Fiscal el cual es irreplicable.

1.3 NORMATIVIDAD QUE DEBE SER CUMPLIDA POR LOS PACS

Para poder obtener la autorización como Proveedor Autorizado de Certificación – PAC, los interesados deben cumplir con las fichas de trámite del Anexo 1-A de la Resolución de Miscelánea Fiscal vigente, la cual señala los siguientes puntos:

- 114/CFF “Inscripción al taller tecnológico dirigido a contribuyentes interesados en obtener la autorización para operar como proveedor de certificación de CFDI”.
- 115/CFF “Solicitud de validación y opinión técnica del cumplimiento de requisitos tecnológicos y de seguridad para solicitar la autorización para operar como proveedor de certificación de CFDI”, y
- 116/CFF “Solicitud para obtener autorización para operar como proveedor de certificación de CFDI”.

Para el cumplimiento del punto 115/CFF el SAT ha creado la matriz de control que contiene los requisitos que permitirán cumplir con las revisiones de seguridad tecnológica que se hace mención. Esta matriz se encuentra en la dirección electrónica:

ftp://ftp2.sat.gob.mx/asistencia_servicio_ftp/publicaciones/cfdi/MatrizControlesCFDI.xlsx

Así mismo los aspirantes deben firmar una carta compromiso, de confidencialidad, reserva y resguardo de información y datos la cual se encuentra en la dirección electrónica

ftp://ftp2.sat.gob.mx/asistencia_servicio_ftp/publicaciones/cfdi/CartaCompromisoPAC.pdf

Existe también la obligación de declarar los inventarios de información, las características funcionales y servicios generales con los que cuenta el interesado

y para ello se debe llenar el formato especificado que se encuentra en la dirección electrónica:

ftp://ftp2.sat.gob.mx/asistencia_servicio_ftp/publicaciones/cfd/InventariodeActivos.xls

En la fracción VI de la ficha 115/CFF del anexo 1-A de la Resolución Miscelánea Fiscal (RMF) para 2014 se menciona que se debe describir las características funcionales y servicios generales que deberá cumplir la aplicación informática gratuita para la generación y expedición de CFDI; para ello se cuenta con el documento que lo especifica y que se encuentra en la dirección electrónica ftp://ftp2.sat.gob.mx/asistencia_servicio_ftp/publicaciones/cfd/Aplicaciongratuita%28v5%29.pdf

Una vez obtenida la autorización por parte del SAT se adquieren las siguientes obligaciones por parte del PAC

- Estar al corriente en el cumplimiento de sus obligaciones fiscales.
- Mantener un capital social suscrito y pagado de por lo menos \$10'000,000.00 (Diez millones de pesos M.N.) durante el tiempo en el que la autorización se encuentre vigente.
- Contar con la garantía a que se refiere la ficha de trámite 116/CFF “Solicitud para obtener autorización para operar como proveedor de certificación de CFDI”, contenida en el Anexo 1-A.
- Presentar el dictamen de sus estados financieros para efectos fiscales, por el ejercicio en que se les otorgue la autorización que solicita y por todos los ejercicios durante los cuales gocen de la misma.
- Permitir y facilitar la realización de actos de verificación y de supervisión por parte del SAT, de manera física o remota, respecto de tecnologías de la información, confidencialidad, integridad, disponibilidad, consistencia y seguridad de la información y/o cualquier otra de las obligaciones relacionadas con la autorización.
- Permitir que el SAT aplique en cualquier momento evaluaciones de confiabilidad al personal del proveedor de certificación de CFDI,

relacionado con la prestación del servicio para el que fue autorizado.

- Cumplir en términos de la Ley Federal de Protección de Datos Personales en Posesión de los Particulares, con la reserva de la información contenida en las facturas electrónicas que certifique a los contribuyentes, y salvaguardar la confidencialidad de todos los datos proporcionados por los contribuyentes, sean parte o no de las facturas electrónicas.
- Generar y poner a disposición de cualquier contribuyente, que así lo desee, una aplicación gratuita para el timbrado de sus comprobantes, la cual deberá ser una instancia de ejecución de su programa principal cumpliendo con las mismas características de disponibilidad y seguridad a través de un sitio web.
- Validar y certificar de manera gratuita las facturas electrónicas que generen los contribuyentes a partir de la aplicación gratuita, y cumplir con las características funcionales, servicios generales y niveles de servicios mínimos, publicados en la página de Internet del SAT.
- Devolver a los contribuyentes la Factura Electrónica validada conforme a lo que establecen los artículos 29 y 29-A del CFF, con folio asignado y con el sello digital del SAT, emitido para dicho efecto.
- Enviar al SAT las facturas electrónicas, al momento en que realicen su certificación, con las características y especificaciones técnicas que le proporcione el SAT.
- Tener a disposición del SAT en todo momento el acceso a las bases de datos en donde se resguarde la información y las copias de las facturas electrónicas que hayan certificado en los últimos tres meses.
- Proporcionar a los contribuyentes emisores, una herramienta para consultar el detalle de sus facturas electrónicas certificadas.
- Conservar las facturas electrónicas certificadas por un término de tres meses en un medio electrónico, óptico o de cualquier tecnología, aún cuando no subsista la relación jurídica al amparo de la cual se certificaron los CFDI.

-
- Administrar, controlar y resguardar a través de su sistema certificador de facturas electrónicas, los certificados de sello digital (CSD) que les proporcione el SAT para realizar su función.
 - Comunicar por escrito o vía correo electrónico a sus clientes en caso de que suspendan temporalmente sus servicios, con al menos sesenta días de anticipación.
 - Cumplir con los estándares de niveles de servicio establecidos, así como con la “carta compromiso de confidencialidad, reserva y resguardo de información y datos”, que se encuentra publicada en la página de Internet del SAT.
 - Presentar conforme a lo señalado en la ficha de trámite 118/CFF “Aviso para la actualización de datos del Proveedor de Certificación de CFDI, publicado en la página de Internet del SAT”, contenida en el Anexo 1-A, el aviso correspondiente ante la Administración Central de Normatividad de impuestos internos (ACNII), dentro de los tres días siguientes a aquél en que se realice el trámite por cambio de denominación o razón social, nombre comercial, clave del RFC, domicilio fiscal, teléfonos, sitios de Internet o cualquier otro dato que hubiere sido considerado para otorgar la autorización.
 - Cumplir con la matriz de control publicada en la página de Internet del SAT.
 - Comunicar a la Administración General de Comunicaciones y Tecnologías de la Información (AGCTI) sobre los cambios tecnológicos realizados con posterioridad a la obtención de la autorización como proveedor de certificación de CFDI.

1.4 DESCRIPCIÓN BÁSICA DE LO QUE DEBE DE CONTENER UN CFDI


Como tal, los Comprobantes Fiscales Digitales por Internet son los archivos XML que contienen toda la información de la transacción monetaria incluyendo el

complemento de certificación, las partes que debe contener este las describimos a detalle en el tema 4.1.8 MODELO DE IMPLEMENTACIÓN en la sección “Definición del CFDI en formato XML”.

Referente a la representación impresa del CFDI debe contener los requisitos que especificados en los artículos 29 y 29-A del Código Fiscal de la Federación que son las siguientes:

- Datos generales del emisor. Vea figura 1.4.1
 - Clave del Registro Federal de Contribuyente.
 - Logotipo de la empresa.
 - Razón social.
 - Domicilio fiscal.
 - Régimen fiscal en que tributen conforme a la Ley del ISR.
 - Si se tiene más de un local o establecimiento, se deberá señalar el domicilio del local o establecimiento en el que se expidan las facturas electrónicas.

The image shows a printed invoice (CFDI) for 'Industria'. It includes the company logo, a box with the issuer's details, a table with the stamp date and internal folio, and a barcode. Below the invoice, there is a note about the tax regime and payment type, and a page number.

FECHA Y HORA DE TIMBRADO	FACTURA
02/07/2014 12:00:23	JA 10 FOLIO INTERNO 

Este documento es una representación impresa de un CFDI

Régimen General de Ley Personas Morales Tipo de comprobante: ingreso

Forma de pago: Pago en una sola exhibición Método de Pago: NA

Página 1 de 2

Figura 1.4.1 Datos Generales del emisor.

- Datos generales del receptor. Vea la figura 1.4.2
 - Clave del Registro Federal de Contribuyente.
 - Razón social.
 - Domicilio fiscal.

- Régimen fiscal en que tributen conforme a la Ley del ISR.

FAX: 56330870

Este documento es una representación impresa de un CFDI

EMISOR:
INDUSTRIA S.A. DE C.V.
Av. Río Churubusco 385
General Pedro María Anaya
CP: 03340 Benito Juárez
México RFC: XAXX010101000
TEL: 56884233 FAX: 56041364

Régimen General de Ley Personas Morales Tipo de comprobante: ingreso

Forma de pago: Pago en una sola exhibición Método de Pago: NA

Página 1 de 2

RECEPTOR CUENTA 2003/15/05	ENVIAR A		
PAPELERIA EL LAPIZ ROJO, S.A. DE C.V. GUERRERO NO.14 CENTRO PURUANDIRO MICHOACAN 58500, MEXICO RFC. PLR021228J2A 14383830038			
0			
REFERENCIA DE PEDIDO	DE FECHA	TIPO DE PAGO	OBSERVACIONES
			lapizrojo.cdigitales@yahoo.com.mx
ENVIADO POR	PLAZO DE PAGO	NUM GUIA	
	030 DIAS FECHA FACTURA		

Figura 1.4.2 Datos generales del receptor.

- Contener el número de folio asignado por el SAT, ver figura 1.4.3

	Sub Total \$	1551.75
	- Descuentos \$	0.00
	+ Imp. Traslados \$	248.25
	- IVA Retenido \$	0.00
	- ISR Retenido \$	0.00
	= Importe Neto \$	1800.00

IVA Tasa 16.00% \$248.25 IVA Tasa 0.00% \$0.00

(UN MIL OCHOCIENTOS PESOS 00/100 M.N.)

Fecha y hora de generación: 2014-07-02T12:00:13

Número de serie del CSD del Emisor: 0000100000202415670

Número de serie del Certificado del SAT: 0000100000202864883

Folio Fiscal - UUID: b1a23105-66f4-4079-9a89-0357c4d174e3

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor.

ORIGINAL

Para obtener los archivos de este CFDI visite: comercio.com.mx

Figura 1.4.3 folios del SAT.

- Sello digital del contribuyente que lo expide. Vea la figura 1.4.4

Página 2 de 2

Forma de pago: Pago en una sola exhibición Método de Pago: NA

Sello del SAT:
QjvuN2YZZPOH5/7aRnA3COMWg8g3008HZzvi60Q+uSwHW3TOCxPnCIHR7TjaXoVNW3UCFn/DyFpp14CJy4Jl/4IHbC9L2AlrQboXeHN
cSWjGfRdEi6CvYvAKvi0Ry1pAGtmAg78XAwqYuDf57ORq8gzMagVAGkINBw/v1g=

Sello del CFDI:

qtxSBir/fnmRZyF7BPPrNzaXFLifc2fyegmKCneZGpWlmz/J9Yr1IQ1BidDil1W3FOUzd07+XzA2/2Vva9wfoJAUJtb4JbIUuPOI4U0eV06PxK1KMl2/d7aUbB71YTOTzTW1G
63MT4OnbcVegV+IzPjYWR9odL0IkZ3lchE=

Cadena original del complemento de certificación:
[|1.0|b1a23105-66f4-4079-9a89-0357c4d174e3|2014-07-02T12:00:23|qtxSBir/fnmRZyF7BPPrNzaXFLifc2fyegmKCneZGpWlmz/J9Yr1IQ1BidDil1W3FOUzd07+XzA2/2Vva9wfoJAUJtb4JbIUuPOI4U0eV06PxK1KMl2/d7aUbB71YTOTzTW1G63MT4OnbcVegV+IzPjYWR9odL0IkZ3lchE=|0000100000202864883|]

Figura 1.4.4 Sello digital del contribuyente.

- Lugar y fecha de expedición. Vea la figura 1.4.5

The image shows a digital receipt from 'Industria'. On the left is the company logo and name. Below it is the address: 'EMISOR: INDUSTRIA S.A. DE C.V., Av. Río Churubusco 385, General Pedro María Anaya, CP: 03340 Benito Juárez, México RFC: XXCO10101000, TEL: 56884233 FAX: 56041364'. In the center, a blue box highlights the digital stamp area containing: 'Expedido en sucursal: Calz. de la viga 1132, Apatlaco, Distrito Federal México, CP: 09439 Iztapalapa, TEL: 56330995, FAX: 56330870'. To the right, a table shows the stamp details: 'FECHA Y HORA DE TIMBRADO' with the value '02/07/2014 12:00:23' and 'FACTURA' with 'JA 10', 'FOLIO INTERNO', and a barcode. Below the stamp area, it says 'Este documento es una representación impresa de un CFDI'. At the bottom, it includes 'Régimen General de Ley Personas Morales', 'Tipo de comprobante: ingreso', 'Forma de pago: Pago en una sola exhibición', 'Método de Pago: NA', and 'Página 1 de 2'.

Figura 1.4.5 Lugar y fecha de expedición.

- Descripción de mercancía o servicios. Vea la figura 1.4.6
 - Cantidad, unidad de medida y clase de los bienes, mercancías o servicios
 - Señalamiento expreso cuando la contraprestación se pague en una sola exhibición o en parcialidades y forma en que se realizó el pago (efectivo, transferencia electrónica, cheque nominativo o tarjeta de débito, de crédito, de servicio o la denominada monedero electrónico, indicando al menos los últimos cuatro dígitos del número de cuenta o de la tarjeta correspondiente).
 - descripción del servicio o del uso y goce que amparen.
 - Valor unitario consignado en número.
 - Importe total señalado en número o en letra.

NO. IDENTIFICACION	AUTOR TITULO	UNIDAD	CANTIDAD	PRECIO UNITARIO	IMPORTE
0000000202138	CARTEL R-LA DONDE VAS TOMAS	PIEZA	0	0.00	0.00
0000000202145	CARTEL R-LAL MAL TIEMPO	PIEZA	5	20.69	103.45
0000000202152	CARTEL R-L CUENTAME UN CUENTO	PIEZA	0	0.00	0.00
0000000202169	CARTEL R-LEN EL MAR LA LECTURA	PIEZA	0	0.00	0.00
0000000202176	CARTEL R-L FALTA UN LIBRO	PIEZA	0	0.00	0.00
0000000202190	CARTEL R-L LECTURA ES ALTURA	PIEZA	5	20.69	103.45
0000000202206	CARTEL R-L LEER BUEN CONSEJO	PIEZA	0	0.00	0.00
0000000202213	CARTEL R-L PUEBLO QUE LEE	PIEZA	0	0.00	0.00
0000000202220	CARTEL R-L RATON DE BIBLIOTECA	PIEZA	0	0.00	0.00
0000000202237	CARTEL R-L SABIO LEER A DIARIO	PIEZA	5	20.69	103.45
0000000202244	CARTEL R-L SI NO LEO ME ABURRO	PIEZA	1	20.69	20.69
0000000202251	CARTEL R-L TIENE UN LIBRO	PIEZA	5	20.69	103.45
0000000202268	CARTEL R-V GENEROSIDAD	PIEZA	5	20.69	103.45
0000000202275	CARTEL R-V HONESTIDAD	PIEZA	5	20.69	103.45
0000000202282	CARTEL R-V JUSTICIA	PIEZA	5	20.69	103.45
0000000202299	CARTEL R-V LEALTAD	PIEZA	5	20.69	103.45
0000000202305	CARTEL R-V ORDEN	PIEZA	5	20.69	103.45
0000000202312	CARTEL R-V PERSEVERANCIA	PIEZA	5	20.69	103.45
0000000202329	CARTEL R-V RESPETO	PIEZA	5	20.69	103.45
0000000202336	CARTEL R-V RESPONSABILIDAD	PIEZA	5	20.69	103.45
0000000202343	CARTEL R-V SOLIDARIDAD	PIEZA	5	20.69	103.45
0000000202107	CARTEL R-G CUERPOS GEOMETRICOS	PIEZA	3	20.69	62.07
0000000202114	CARTEL R-G FIGURAS GEOMETRICAS	PIEZA	3	20.69	62.07
0000000202121	CARTEL R-G LAS LINEAS	PIEZA	3	20.69	62.07


Sub Total \$		1551.75
- Descuentos \$		0.00
+ Imp. Traslados \$		248.25
- IVA Retenido \$		0.00
- ISR Retenido \$		0.00
= Importe Neto \$		1800.00

IVA Tasa 16.00% \$248.25 IVA Tasa 0.00% \$0.00

(UN MIL OCHOCIENTOS PESOS 00/100 M.M.)


Figura 1.4.6 Descripción de mercancía o servicios

- Métodos de pago. Vea la figura 1.4.7
 - Cuando proceda, se indicará el monto de los impuestos trasladados desglosados por tasa de impuesto y, en su caso, el monto de los impuestos retenidos.
 - Número de referencia bancaria o número de cheque con el que se efectuó el pago (opcional).



Expedido en sucursal:
Calz. de la viga 1132
Apatlaco
Distrito Federal México
CP: 09439 Ixtapalapa
TEL: 56330995
FAX: 56330870

EMISOR:
INDUSTRIA S.A. DE C.V.
Av. Río Churubusco 385
General Pedro María Anaya
CP: 03340 Benito Juárez
México RFC: XAAX010101000
TEL: 56884233 FAX: 56041364

FECHA Y HORA DE TIMBRADO	FACTURA
02/07/2014 12:00:23	JA 10 FOLIO INTERNO 

Este documento es una representación impresa de un CFDI

Régimen General de Ley Personas Morales Tipo de comprobante: ingreso

Forma de pago: Pago en una sola exhibición Método de Pago: NA

Figura 1.4.7 Métodos de pago e impuestos

- Código de barras bidimensional generado conforme al Anexo 20, el cual deberá ser impreso en un cuadrado con lados no menores a 2.75 cm. Figura 1.4.8

Sello del SAT:

QjvuN2YZZPOH5/7aRnA3COMWg8g3008HZzvi60Q+uSwHW3TOCxPnCiHR7TjaXoVNW3UCFh/DyFpp14CJqy4jji/4IHbc9L2AlrQboXeHN
cSWjGRdDEI6CvYvAKwi0Ry1pAGtmAg78XAwqYuDf57ORq8gzMagVAGkiNBw/1v1g=

Sello del CFDI:



qbxSBir/nmRZyF7BPpRiNzaXFLifc2fyegmKcNeZGPwlmz/J9Yr1IQ1BidDil1W3FOUzd07+XzA2/2Vva9wboJAUJtb4JbiUuPOI4U0oV06PxK1KM2/d7aUbBi71YT0TzTW1G
63MT4OnbcVegV+lzPjYWR9odL0kZ3lchE=

Cadena original del complemento de certificación:

||1.0|b1a23105-66f4-4079-9a89-0357c4d174e3|2014-07-02T12:00:23|qbxSBir/nmRZyF7BPpRiNzaXFLifc2fyegmKcNeZGPwlmz/J9Yr1IQ1BidDil1W3FOUzd07+XzA2/2Vva9wboJAUJtb4JbiUuPOI4U0oV06PxK1KM2/d7aUbBi71YT0TzTW1G63MT4OnbcVegV+lzPjYWR9odL0kZ3lchE=|00001000000202864883|

Figura 1.4.8 Código de barras bidimensional

- Campos requeridos por el SAT. Vea la figura 1.4.9
 - Número de serie del CSD del Emisor y del SAT.
 - Fecha y hora de emisión y de certificación de la factura electrónica (CFDI) en adición a lo señalado en el artículo 29-A fracción III del CFF.
 - Cadena original del complemento de certificación digital del SAT.

Fecha y hora de generación:	2014-07-02T12:00:13	
Número de serie del CSD del Emisor:	00001000000202415670	
Número de serie del Certificado del SAT:	00001000000202864883	
Folio Fiscal - UUID:	b1a23105-66f4-4079-9a89-0357c4d174e3	

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor.

ORIGINAL

Para obtener los archivos de este CFDI visite: comercio.com.mx



Deposito en nominal:
Caj. de Lav. y Hig. 1132
Apdo. 100
Distrito Federal, Mexico
C.P. 06702
TEL: 56984235 FAX: 56984234

FECHA Y HORA DE TIMBRADO

FACTURA

02/07/2014 12:00:23

JA 10

FOLIO INTERNO



Este documento es una representación impresa de un CFDI

EMISOR:

INDUSTRIA S.A. DE C.V.
AV. RFO Chimalco 185
Guzmania Pedregal Mexico Anaya
C.P. 01240, PUEBLA, Mexico
Régimen FISC. PASADO DE PASADO

Régimen General de Ley Personas Morales

Tipo de comprobante: Ingreso

Página 2 de 2

Sello del SAT:

QjvuN2YZZPOH5/7aRnA3COMWg8g3008HZzvi60Q+uSwHW3TOCxPnCiHR7TjaXoVNW3UCFh/DyFpp14CJqy4jji/4IHbc9L2AlrQboXeHN
cSWjGRdDEI6CvYvAKwi0Ry1pAGtmAg78XAwqYuDf57ORq8gzMagVAGkiNBw/1v1g=

Sello del CFDI:





qbxSBir/nmRZyF7BPpRiNzaXFLifc2fyegmKcNeZGPwlmz/J9Yr1IQ1BidDil1W3FOUzd07+XzA2/2Vva9wboJAUJtb4JbiUuPOI4U0oV06PxK1KM2/d7aUbBi71YT0TzTW1G
63MT4OnbcVegV+lzPjYWR9odL0kZ3lchE=

Cadena original del complemento de certificación:

||1.0|b1a23105-66f4-4079-9a89-0357c4d174e3|2014-07-02T12:00:23|qbxSBir/nmRZyF7BPpRiNzaXFLifc2fyegmKcNeZGPwlmz/J9Yr1IQ1BidDil1W3FOUzd07+XzA2/2Vva9wboJAUJtb4JbiUuPOI4U0oV06PxK1KM2/d7aUbBi71YT0TzTW1G63MT4OnbcVegV+lzPjYWR9odL0kZ3lchE=|00001000000202864883|

Figura 1.4.9 Campos requeridos por el SAT

- Cualquiera de las siguientes leyendas: “Este documento es una representación impresa de un CFDI”, “Este documento es una representación impresa de un comprobante fiscal digital” o “Este documento es una impresión de un comprobante fiscal digital”. Vea la figura 1.4.10

 <p>Expedido en sucursal: Calz. de la viga 1132 Apatlaco Distrito Federal México CP: 02439 Ixtapalapa TEL: 56330995 FAX: 56330870</p>	<p>FECHA Y HORA DE TIMBRADO</p> <p>02/07/2014 12:00:23</p>		<p>FACTURA</p> <p>JA 10 FOLIO INTERNO</p> 
	<p>Este documento es una representación impresa de un CFDI</p>		

EMISOR:
INDUSTRIA S.A. DE C.V.
Av. Río Churubusco 385
General Pedro María Anaya
CP: 03340 Benito Juárez
México RFC: XAAX010101000
TEL: 56884233 FAX: 56041364



Regimen General de Ley Personas Morales Tipo de comprobante: ingreso

Forma de pago: Pago en una sola exhibición Método de Pago: NA

Página 1 de 2

Figura 1.4.10 Leyendas de documento impreso

- Sello digital del SAT. Vea la figura 1.4.11

 <p>Expedido en sucursal: Calz. de la viga 1132 Apatlaco Distrito Federal México CP: 02439 Ixtapalapa TEL: 56330995 FAX: 56330870</p>	<p>FECHA Y HORA DE TIMBRADO</p> <p>02/07/2014 12:00:23</p>		<p>FACTURA</p> <p>JA 10 FOLIO INTERNO</p> 
	<p>Este documento es una representación impresa de un CFDI</p>		

EMISOR:
INDUSTRIA S.A. DE C.V.
Av. Río Churubusco 385
General Pedro María Anaya
CP: 03340 Benito Juárez
México RFC: XAAX010101000
TEL: 56884233 FAX: 56041364

Regimen General de Ley Personas Morales Tipo de comprobante: ingreso

Forma de pago: Pago en una sola exhibición Método de Pago: NA

Página 2 de 2

Sello del SAT:
OjvuN2YZZPOH5/7aRnA3COMWG8g3008HZzvi60Q+uSwHW3TOCxPnCIHR7TjaXoVNW3UCFnDyFpp14CJqy4lJl/4IHbc9L2AtrQboXeHN
cSWjGfRdDEl6CvYvAKvi0Ry1paGtmAg78XAwqYuDF57ORq8gzMagVAGkINBw1v1g=

Sello del CFDI:




Figura 1.4.11 Sello digital del SAT

1.5 ACOTAMIENTO DEL PROBLEMA A RESOLVER

El sistema que se desarrollará deberá resolver el mandato por la Secretaría de Hacienda a través del Sistema de Administración Tributaria, el cual, dictamina que como medio de comprobación de las transacciones monetarias es necesario emitir comprobante fiscal digital por Internet.

De manera global el problema se visualiza con los siguientes aspectos y se muestra en la figura 1.5.1

- Captura de datos de producto o servicios (Front-End).
- Generación de información en un archivo de tipo texto.
- Transformación de archivo de texto a archivo XML.
- Envío del archivo XML a través de Internet hacia el PAC.
- Recepción del timbre fiscal después de aprobar archivo XML por el PAC.
- Transformación del archivo XML a PDF.
- Envío del CFDI (XML y PDF) a impresión solo formato PDF y por correo electrónico ambos.

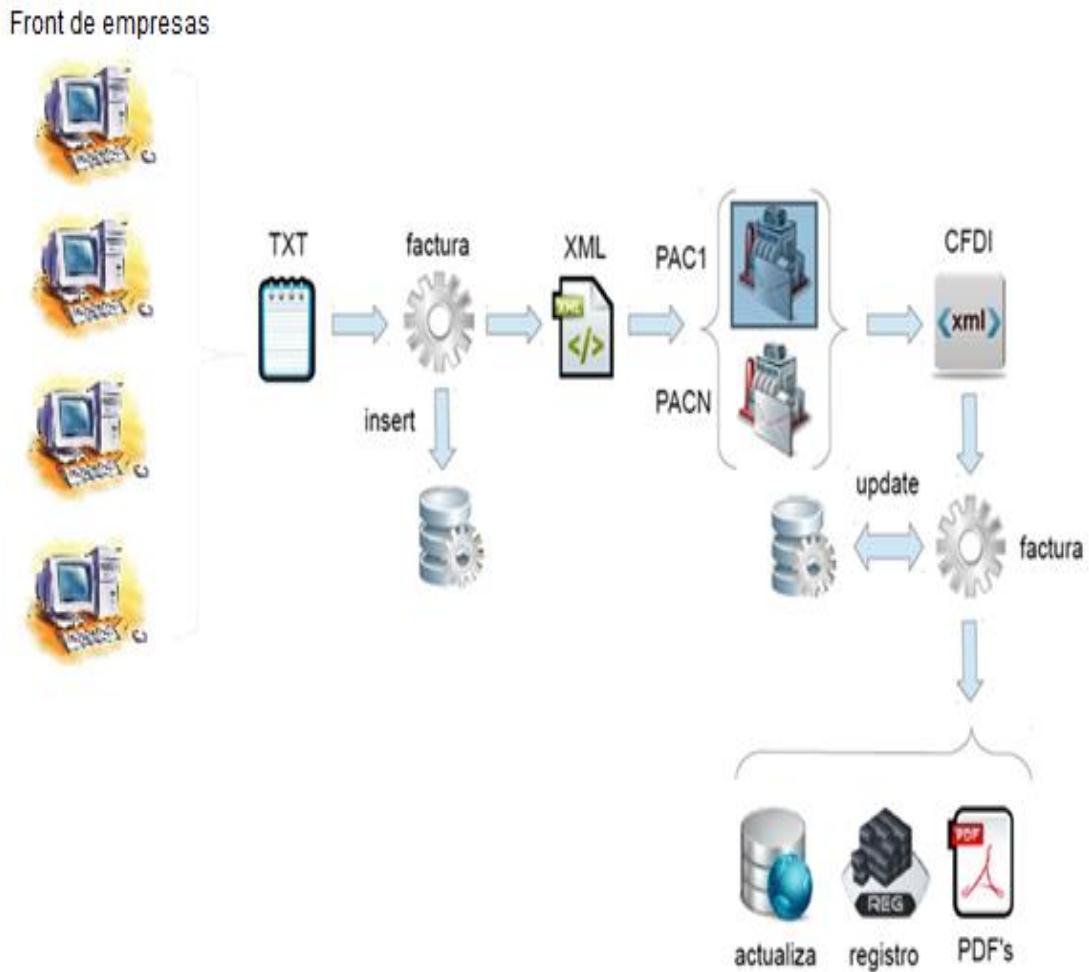


Figura 1.5.1 Proceso de generación de CFDI's

Tomando en consideración que hoy en día la mayoría de las empresas cuenta con un sistema que le permite realizar la comprobación de transacciones monetarias a nivel de impresión sin ser vía Internet o por lo menos contienen un sistema que lleva el control de los comprobantes digitales fiscales, y que cada una de ellas tiene un giro comercial diferente o similar, decidimos desarrollar un Front-End, donde podremos capturar los datos necesarios de productos que nos otorgara el inicio del sistema de la generación de los comprobantes fiscales digitales por Internet.

La figura 1.5.2 muestra entonces, como queda acotado el problema.

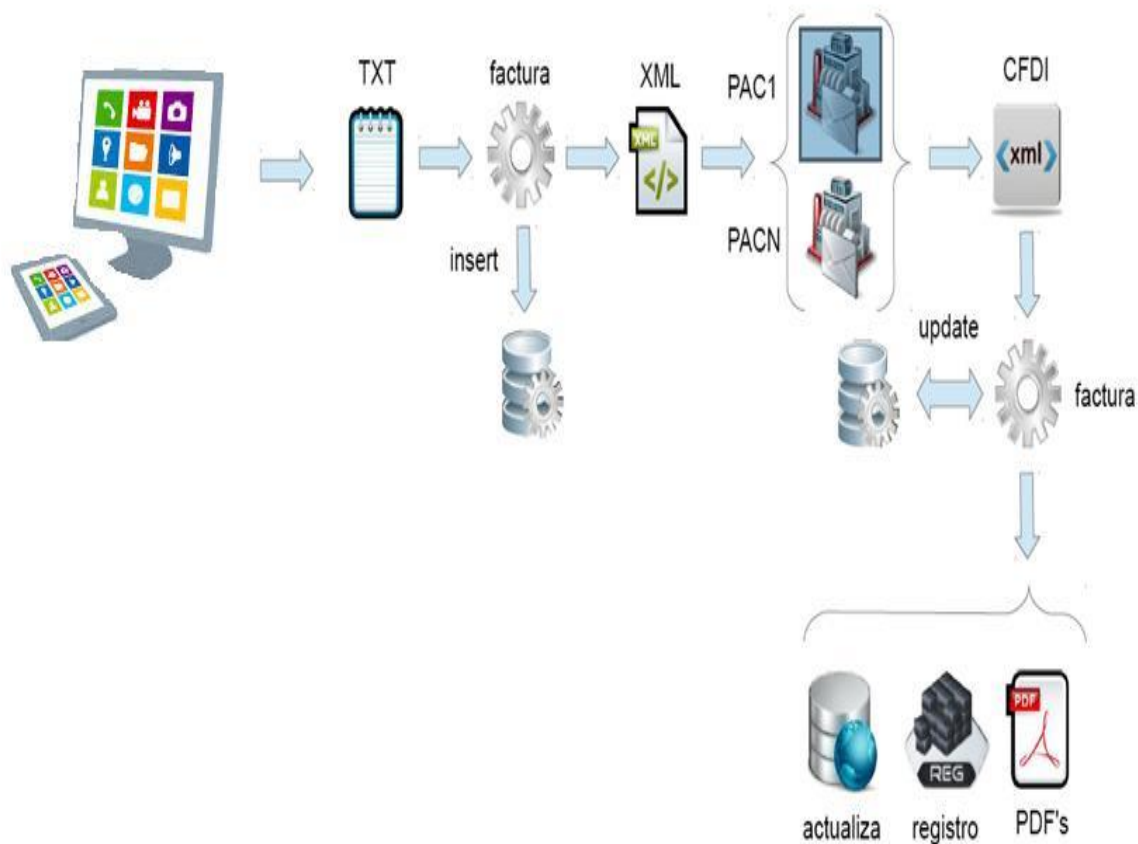


Figura 1.5.2 Proceso acotado de generación de CFDI's

La entrada del sistema deberá ser un archivo de texto (formato .txt), el cual debe contener la información mínima requerida para poder ser procesado, así mismo, se realizará la transformación de los datos al formato correspondiente, en este caso XML para que sea aceptado por el PAC una vez que sea enviado para su validación y certificación, para lograr obtener el producto final CFDI en los formatos de PDF y XML.

Para realizar la transformación de datos a XML, se utilizará programación orientada a objetos (JAVA), así mismo se utilizara un ambiente cliente servidor para poder realizar la transferencia y recepción de datos, ya que es necesario incluir en el proceso la validación de un Proveedor Autorizado de Certificación (PAC), el cual regresará el documento XML timbrado y de esa manera se realizará

un documento con formato PDF para impresión y envío vía correo electrónico de los documentos XML y PDF para el usuario final.

Para poder generar un tipo de archivo PDF para el usuario, esto con el propósito de la legibilidad del documento, se utilizará nuevamente lenguaje de programación orientada a objetos (JAVA).

Se utilizará un manejador de base de datos que no tenga costo, en este caso MS SQL, para poder guardar toda la información generada por el sistema que realizará la generación de los CFDI's, estos datos se guardaran a partir del evento de transformación del archivo de texto a XML, hasta la generación de los comprobantes fiscales. Adicionalmente se incluirá los mensajes de error para cada una de las transacciones realizadas en caso de que exista alguna no exitosa.

Como apoyo para realizar pruebas y validación del sistema se desarrollara un Front-End de manera que solo se encuentre habilitado los cuadros de texto que sean necesarios para poder generar la entrada principal, en este caso el archivo de texto, para esto se utilizará lenguaje de programación orientada a objetos (JAVA).

La ejecución del sistema tendrá que ser automática a partir de que el usuario determine que ha finalizado la captura de los datos y requiera generar la factura electrónica, el evento que genera factura, lanzará o ejecutará todo el sistema que valida, aprueba y genera el comprobante fiscal digital por Internet (CFDI) de tal forma que se tendrá los documentos correspondientes tanto en PDF como XML.

Por tanto, decimos que el problema constará de la construcción o desarrollo de un sistema informático que a partir de la captura de datos (producto o servicio) genere un archivo de texto el cual deberá validar los requerimientos mínimos que solicita la administración tributaria en México (SAT) sea transformado a un tipo de archivo valido solicitado por un proveedor autorizado (PAC) el cual solicita que sea de tipo XML, para poder ser validado, autorizado, timbrado y regresado, de igual manera poder realizar la recepción del mismo, para que se pueda hacer la entrega al cliente final de su comprobante fiscal digital por Internet (CFDI), éste se enviará

por correo electrónico de dos maneras, uno de tipo XML que ya se tiene construido y otro de tipo PDF, el cual también se podrá imprimir.

CAPÍTULO

II

MARCO TEÓRICO

2.1 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE LAS BASES DE DATOS RELACIONALES Y METODOLOGÍA DE YOURDON/DE MARCO

Características, ventajas y desventajas de las bases de datos relacionales

Una base de datos es un conjunto de datos organizados en tablas.

Las bases de datos relacionales están definidas en el uso de relaciones, las cuales son tablas representadas gráficamente como una estructura de filas y columnas.

Las columnas contienen información referente a una propiedad determinada, éstas reciben el nombre de atributos. Los valores de los datos son representados por medio de la intersección de entre filas y columnas.

Las bases de datos relacionales cumplen con el modelo relacional, la cual es la más utilizada en la actualidad para planificar bases de datos.

Entre las características de las bases de datos relacionales se encuentran las siguientes:

- Su modelo se basa en el álgebra relacional
- Independencia lógica y física de los datos
- Una base de datos relacional se compone de varias tablas o relaciones.
- No pueden existir dos tablas con el mismo nombre ni registro
- Cada atributo tiene un nombre diferente en las tablas
- Cada tabla es a su vez un conjunto de registros (filas y columnas).
- El orden de las tuplas y de los atributos no importa.
- El número de relaciones que hay entre entidades recibe el nombre de cardinalidad.
- El número de atributos de una tabla recibe el nombre de grado.
- Al conjunto de valores válidos representados por un atributo recibe el nombre de dominio.

-
- La relación entre una tabla padre y un hijo se lleva a cabo por medio de las claves primarias y ajenas (o foráneas).
 - Las claves primarias son la clave principal de un registro dentro de una tabla y éstas deben cumplir con la integridad de datos.
 - La información es atómica, es decir, cada campo de un registro tiene uno y sólo un valor.
 - Debe haber consistencia en la información, es decir, no debe haber incongruencias en los datos que se almacenan.
 - La información debe ser no redundante, es decir, se debe de eliminar al máximo la existencia de duplicidad de los datos.
 - Los valores que posean los datos deben ser válidos de acuerdo a las funciones entre tablas. A esto se refiere la integridad.
 - Las relaciones entre dos o más tablas se establecen en virtud de los valores de datos comunes contenidos en las tablas correspondientes.
 - Los sistemas de bases de datos relacionales suelen tener asociados lenguajes de consulta de alto nivel, que facilitan y optimizan la realización de búsquedas y actualizaciones en la base de datos.

Entre sus ventajas se encuentran:

- Podemos acceder a los datos a través de llaves.
- Está el hecho de que podemos agregar posteriormente datos al sistema sin tener que reescribir lo que ya tenemos.
- Provee herramientas que garantizan evitar la duplicidad de registros.
- Eficiencia, nos referimos al hecho de que no tenemos duplicación de datos, y tampoco tenemos grandes cantidades de “celdas vacías”. De esta manera, no se tienen que almacenar más datos de los necesarios, ni gastar recursos al revisar las áreas vacías en nuestras tablas.
- Garantiza la integridad referencial, así, al eliminar un registro elimina todos los registros relacionados dependientes.
- Favorece la normalización por ser más comprensible y aplicable.

Entre sus desventajas se encuentran:

- El acceso a los datos puede llegar a ser lento, dependiendo de la robustez de la información.
- Presentan deficiencias con datos gráficos, multimedia y sistemas de información geográfica.
- No se manipulan de forma manejable los bloques de texto como tipo de dato.

Metodología Yourdon/DeMarco

Durante el ciclo de vida del software es fundamental el análisis, diseño y planeación del sistema a desarrollar. Dentro de este proceso es claro el uso de diagramas que nos ayuda a crear representaciones o interpretaciones con precisión de los distintos componentes del sistema. Un buen seguimiento de cada de estos pasos podría aumentar la calidad de los resultados.

Dicho lo anterior, será útil basarnos de una metodología de procesos en el que podamos cumplir con los requisitos pertinentes para modelar un nuevo sistema.

Por sus ventajas y las herramientas disponibles, dentro de este proyecto se optó por el uso de la Metodología Yourdon/DeMarco.

Antecedentes de la Metodología Yourdon/DeMarco

El desarrollo de grandes sistemas de Información es una de las actividades más complejas a las que se han enfrentado muchas organizaciones desde la aparición de la computadora; desde entonces muchos investigadores se han preocupado por el desarrollo de métodos y herramientas que faciliten esta tarea.

El análisis y diseño estructurado fue el resultado de numerosos esfuerzos de estos investigadores. Básicamente consiste en la descomposición de sistemas complejos de una manera organizada en unidades manejables.

Los primeros trabajos sobre el análisis y diseño estructurado fueron realizados por Gane, Searson y De Marco, y fueron posteriormente completados y ampliados por Yourdon, Constantine, Myers y Jackson.

Características de la metodología Yourdon/DeMarco

El Método Estructurado de Yourdon contempla todas las fases del ciclo de vida de desarrollo de un sistema de información, entendiendo como ciclo de vida el proceso de planear, ejecutar y controlar el proyecto de desarrollo:

- **Fase 1: Encuesta**
Es el estudio de factibilidad que determina que los objetivos del sistema sean realistas.
- **Fase 2: Análisis**
El análisis consiste en identificar las necesidades del cliente, evaluar la viabilidad del sistema, realizar un análisis técnico y económico, asignar funciones al software, al hardware, a la gente, a la base de datos y a otros elementos del sistema, estableciendo restricciones de costo y tiempo, creando una definición del sistema que sea la base para todo el trabajo posterior de Ingeniería.
- **Fase 3: Diseño**
La fase de diseño está orientada a la planificación y organización de los datos, así como la definición de acceso a esta información. En el diseño también se contemplan las interfaces de entrada y salida del sistema que cumplan con las especificaciones del usuario.

-
- Fase 4: Implementación
Esta actividad incluye la codificación y la integración de módulos en un esqueleto progresivamente más completo del sistema final.
 - Fase 5: Generación de prueba de aceptación
La especificación estructurada debe contener toda la información necesaria para definir un sistema que sea aceptable desde el punto de vista del usuario.
 - Fase 6: Garantía calidad
Es la prueba final o la prueba de aceptación.
 - Fase 7: Descripción de Procedimientos
Generación de una descripción formal de las partes del sistema.
 - Fase 8: Conversión de bases de datos
Esta actividad requiere como entrada la base de datos actual del usuario, al igual que la especificación del diseño, es posible que no exista una base de datos a convertir, sin embargo, en algunos proyectos esta actividad involucra más trabajo que el mismo desarrollo de programas.
 - Fase 9.- Instalación
En la fase de instalación se integran los productos de salida de fases anteriores, llevando a cabo la instalación del sistema en ambientes productivos.

El método estructurado de Yourdon, se basa en la generación de diferentes modelos, vea figura 2.1.1, con ayuda de estos modelos se cubre la parte del análisis y diseño de un sistema.

Los modelos que abarcan la parte del análisis son:

-
- Modelo esencial ambiental
 - Modelo esencial de comportamiento
 - Modelo de implementación del usuario

Los modelos que abarcan la parte del diseño son:

- Modelo de implementación del usuario
- Modelo de implementación del programa

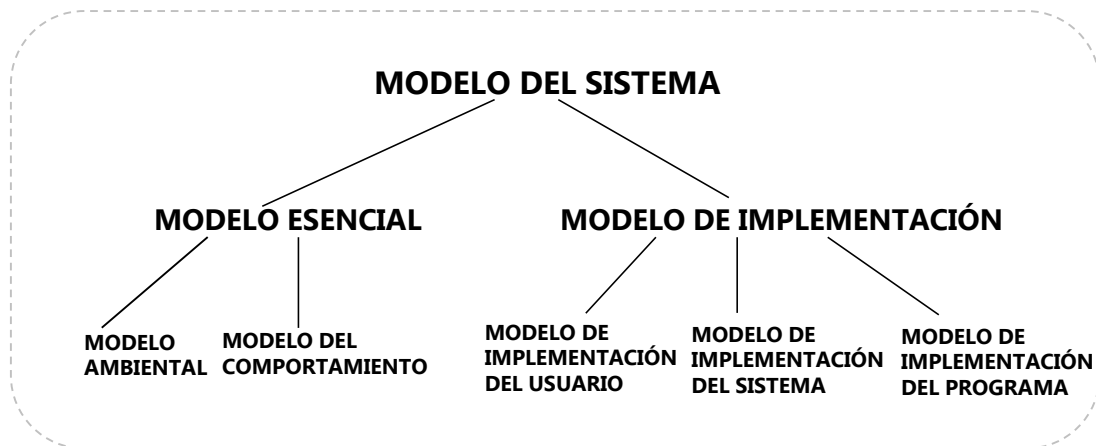


Figura 2.1.1 Modelos de la Metodología de Yourdon

El modelo esencial resalta las funciones del sistema y se enfoca en definir qué es lo que hará el sistema, mientras que el modelo de implementación describe la forma en que se implementará el sistema.

Cada uno de estos modelos hace uso de diferentes herramientas, la mayoría de estas herramientas son gráficas, tales como:

- Diagrama de contexto
- Diagrama de Entidad-Relación
- Diagrama de flujo de datos
- Diagrama de estructuras

Ventajas

- Permiten discutir cambios y correcciones en los requerimientos del usuario, a bajo costo y con riesgo mínimo.

-
- Permite verificar que los requerimientos del usuario sean consistentes y completos.
 - Permiten identificar los componentes de un sistema y su interfaz.
 - La generación de sus diversos modelos ayudan a :
 - validar, comunicar y organizar las ideas del diseñador sobre el funcionamiento del sistema.
 - Se puede controlar redundancia y nos permiten predecir el comportamiento del sistema.
 - Ayudan a enfocar características importantes del sistema.
 - Ayudan a disminuyen los posibles errores, malas interpretaciones u omisiones.
 - Ayudan a documentar el sistema

Desventajas

- La teoría de esta metodología es amplia.
- Toma tiempo aprender la metodología.
- Toma tiempo adquirir experiencia.
- Las actividades de documentar son tediosas

2.2 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE LA PROGRAMACIÓN ORIENTADA A OBJETOS

La programación orientada a objetos es un paradigma de programación, que usa un conjunto de objetos que se comunican entre ellos para realizar tareas y que es un modelo que representa un subconjunto del mundo real, tan fielmente como sea posible, de modo fácil y natural, donde los objetos van a tener características (atributos) y comportamientos (métodos). A diferencia de los lenguajes procedurales, en donde los datos y los procedimientos se encuentran separados y sin relación alguna.

Los lenguajes procedurales utilizan funciones a las que se les pasan datos, en tanto que los lenguajes orientados a objetos definen objetos y después se les envían mensajes a los objetos diciendo que realicen alguno de los métodos especificados para el objeto.

La programación orientada a objetos es una forma de programar que trata de encontrar una solución a estos problemas, introduce nuevos conceptos, que superan y amplían conceptos antiguos ya conocidos.

Conceptos fundamentales de la programación orientada a objetos

Para entender este modelo a continuación se describen sus mecanismos básicos:

- Objeto.- Es el concepto principal sobre el cual se fundamenta la tecnología orientada a objetos. Un objeto puede ser visto como una entidad que posee atributos y efectúa acciones. En el mundo real podemos encontrar cientos de ejemplos que cumplen con esta definición, algunos de ellos son: un árbol, una persona, un automóvil, una computadora etc.

Estos objetos son casos particulares de entidades llamadas clases en donde se definen las características comunes de tales objetos. Si vemos un caso en particular del objeto bicicleta, podemos mencionar como atributos de éste: la marca, el color, el material con el que está construida, las velocidades, entre otro. Algunas acciones que es capaz de realizar un automóvil son; acelerar, virar, frenar y cambiar de velocidad. Este objeto, es una instancia particular de la entidad automóvil. En términos de la programación orientada a objetos, se dice que todo objeto tiene un estado (atributos) y un comportamiento (acciones). La programación orientada a objetos nos permite modelar estos objetos del mundo real en objetos de software de forma eficaz. Un objeto de software mantiene sus atributos o estado en variables e implementa las acciones o comportamientos por medio de métodos o funciones.

-
- Clases.- La clase es un modelo o prototipo que define las variables y los métodos comunes a todos los objetos de cierta clase. También se puede decir que una clase es una plantilla genérica para un conjunto de objetos de similares características.

Por otro lado, una instancia de una clase es otra forma de llamar a un objeto. En realidad no existe diferencia entre un objeto y una instancia. Sólo que el objeto es un término más general, objetos e instancias son ambas representaciones de una clase.

Se tienen 3 tipos de clase que son:

- Abstracta.- es muy general (ej. Animal).
- Común.- es intermedia (ej. Mamíferos).
- Final.- es muy específica (ej. Perro Labrador).

La superclase, se puede definir en términos sencillos como la clase padre de alguna clase específica y puede tener cualquier número de subclases.

La subclase, es la hija de alguna clase específica y sólo puede tener una superclase.

- Herencia.- La herencia básicamente consiste en que una clase puede heredar sus variables y métodos a varias subclases (la clase que hereda es llamada superclase o clase padre). Esto significa que una subclase, aparte de los atributos y métodos propios, tiene incorporados los atributos y métodos heredados de la superclase. De esta manera se crea una jerarquía de herencia.
- Método.- Es una acción que determina cómo debe actuar un objeto cuando recibe un mensaje. En analogía con el lenguaje procedural se le llamaría "función". Un método también puede enviar mensajes a otros objetos, para realizar una acción o para pedir información.

-
- Mensajes.- En la programación orientada a objetos, los objetos se comunican a través de señales o mensajes , siendo estos mensajes los que hacen que los objetos respondan de diferentes maneras, por ejemplo un objeto en Linux como una ventana de alguna aplicación, puede cerrarse, maximizarse o restaurarse (métodos) de acuerdo al mensaje que se le ha enviado. En otras palabras, un mensaje es una comunicación dirigida a un objeto, que le ordena que ejecute uno de sus métodos pudiendo o no llevar algunos parámetros.

Características

Modularidad

Se denomina modularidad a la propiedad que permite subdividir una aplicación en partes más pequeñas (llamadas módulos), cada una de las cuales debe ser tan independiente como sea posible de la aplicación en sí y de las restantes partes. Estos módulos se pueden compilar por separado, pero tienen conexiones con otros módulos.

La modularidad no permite poder modificar las características de clase que definen a un objeto, de forma independiente de las demás clases en la aplicación. En otras palabras, si nuestra aplicación puede dividirse en módulos separados, normalmente clases y estos módulos pueden compilarse y modificarse sin afectar a los demás, entonces dicha aplicación ha sido implementada en un lenguaje de programación que soporta la modularidad. La tecnología orientada a objetos nos brinda esta propiedad para hacer uso de ella en el software que se desee desarrollar.

Abstracción

La abstracción es el proceso por el cual separamos las propiedades más importantes de los objetos, de las que no son. Es decir, por medio de este proceso definimos las características esenciales de un objeto del mundo real, los atributos

y comportamientos que lo definen como tal para después modelarlo en un objeto de software.

La abstracción es clave en el proceso de análisis y diseño orientado a objetos, ya que mediante ella podemos llegar a armar un conjunto de clases que permitan modelar la realidad o el problema que se quiera atacar.

Encapsulamiento

El encapsulamiento es la propiedad de programación orientada a objetos que nos permite asegurar que la información de un objeto le sea desconocida a los demás objetos en la aplicación. Es decir cada objeto está aislado del exterior, esta característica permite verlo como una caja negra que contiene toda la información relacionada con ese objeto.

Este asilamiento protege a los datos asociados a un objeto para que no puedan modificar por quien no tenga derecho a acceder a ellos. Permite manejar los objetos como unidades básicas, dejando oculta su estructura interna.

Polimorfismo

El polimorfismo es la propiedad por la cual una entidad puede tomar diferentes formas, generalmente esta entidad es una clase y la forma en que se consigue que tome diferentes formas es por medio de nombrar a los métodos de dicha clase con un mismo nombre pero con diferentes implementaciones.

Existen dos tipos principales de polimorfismo, que son;

- Por reemplazo.- dos o más clases diferentes con el mismo nombre del método pero haciéndolo de forma diferente.
- Por sobrecarga.- es el mismo nombre del método ocupado varias veces, ejecutándolo de diferente forma y diferenciándose solamente por el argumento o parámetro.

Características adicionales de la programación orientada a objetos

Adicionalmente de las características antes mencionadas como esenciales de los lenguajes de programación orientada a objetos, es deseable que éstos cumplan también con las siguientes:

- Tipificación fuerte.-Esto es, que durante la fase de diseño e implementación se declare que tipo de datos soporta cada variable.
- Manejo de excepciones.-Dentro de la misma definición del lenguaje se deberá establecer la forma de detectar y manipular excepciones que puedan surgir durante la ejecución de un programa.
- Generalidad.- Se refiere principalmente a que las clases se definan lo más generalizadas posible para que sean fácilmente reusables. Para generar este tipo de clases, normalmente se definen parámetros formales que son instanciados por parámetros reales.
- Multitarea.-Es conveniente que el lenguaje permita la creación de procesos que se ejecuten de forma simultanea independientemente del sistema operativo.
- Persistencia.-Los objetos deben de poder permanecer, si así se desea, después de la ejecución de un programa.
- Datos compartidos.-Los objetos pueden necesitar referirse a la misma localidad de memoria o bien comunicarse mediante mensajes.

Ventajas

- Flexibilidad.-Si partimos del hecho que mediante la definición de clases establecemos módulos independientes, a partir de los cuales podemos definir nuevas clases, entonces podemos pensar en estos módulos como bloques con los cuales podemos construir diferentes programas.
- Reusabilidad.-Cuando hemos diseñado adecuadamente las clases, se pueden usar en distintas partes del programa y sus numerosos proyectos.

-
- **Mantenibilidad.**- Debido a la sencillez para abstraer el problema, los programas orientados a objetos son más sencillos de leer y comprender, pues nos permiten ocultar detalles de implementación dejando visibles sólo aquellos más relevantes.
 - **Modificabilidad.**- La facilidad de añadir, suprimir o modificar nuevos objetos nos permite hacer modificaciones de una forma muy sencilla.
 - **Fiabilidad.**- Al dividir el problema en partes más pequeñas podemos probarlas de manera independiente y aislar mucho más fácilmente los posibles errores que puedan surgir.

Desventajas

- **Limitaciones del programador.**-Un gran porcentaje de los programadores no está familiarizado con los conceptos de dicha tecnología, aunque la lógica de la programación estructurada sigue siendo predominante en la mayoría de los desarrolladores de software, cuando se revisan los principios de la programación orientada a objetos, no es claro en que esta requiere una lógica de pensamiento totalmente diferente a lo lógica comúnmente utilizada para la programación estructurada.
- **Tamaño excesivo en las aplicaciones resultantes.**- Esto sucede cuando se heredan clases a partir de clases existentes, éstas heredan de forma implícita todos los miembros de dicha clase aun cuando no todos se necesiten, lo que produce aplicaciones muy grandes que no siempre encajan en los sistemas con los que se disponga.
- **Velocidad de ejecución.**- Una aplicación innecesariamente pesada en muchas ocasiones es más lenta de ejecutar que una aplicación conformada únicamente por los módulos necesarios.

2.3 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DEL ENTORNO JAVA,

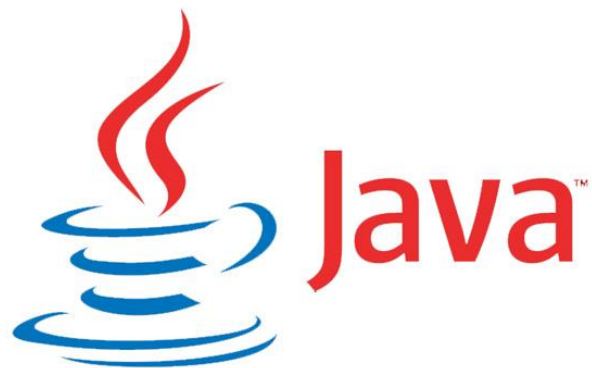


Figura 2.3.1 Logotipo de Java

Características

- Hereda la sintaxis del lenguaje C++ pero eliminando algunas de sus características más oscuras¹, la más significativa de ellas, el uso de punteros.
- Sobrecarga de operadores: varios operadores llamados de la misma manera dentro del mismo contexto (una clase) pero con distintos argumentos.
- Bien definido: el lenguaje cuenta con una documentación concisa que deja

1 El lenguaje de programación C++ heredó a su vez el uso de punteros del lenguaje de programación C. Los punteros hacen referencia dentro del código fuente a locaciones de memoria, con información útil para el programa pero difícil de interpretar por un programador que no conoce el código a detalle.

claro cuáles son los criterios y reglas a seguir en la utilización del código.

- Estructuras de datos: Java permite generar tipos de datos más complejos definidos a partir de tipos de datos más simples, cuenta con algunos mecanismos muy poderosos para tener colecciones de datos muy homogéneos
- Modularidad: Se pueden desarrollar componentes de manera independiente que eventualmente interactúan.
- Facilidad de entrada – salida: Java tiene clases diseñadas a la recepción y envío de datos mediante teclado (sólo entrada), archivos, y streams que viajan por una red determinada. Puede también obtener y enviar datos mediante puertos y otros medios.
- Interacción con bases de datos: se ha escrito un estándar para el acceso de una base de datos el cual es sólo un modelo, con este modelo cada fabricante puede escribir su propia particularización de este modelo aplicado a su propia base de datos.
- Independencia de la plataforma: El bitcode que genera el compilador de Java se ejecuta dentro de una máquina virtual, cada plataforma cuenta con su propia máquina virtual que hace la interpretación de este bitcode a instrucciones propias del procesador donde está corriendo.

VENTAJAS

- Al eliminar los punteros de la sintaxis las aplicaciones en Java se han hecho menos obscuras y más seguras.
- La sobrecarga le ha dotado a Java de una expresividad² muy importante, una característica también heredada del lenguaje de programación C++
- La eliminación de punteros dentro de su sintaxis, permite que el código sea mucho más seguro al tener una relación clara entre la llamada a otra

2 La expresividad dentro de un lenguaje de programación es la capacidad que tiene el lenguaje de programación, de dar a entender al programador no relacionado con ese código en particular, las acciones realizadas por el mismo.

sección del código o variable mediante operadores totalmente entendibles al ojo humano.

- Bien definido: permite la estandarización, mediante la definición de éste, los tipos, longitudes y uso de los distintos operadores, tipos de datos y la forma de utilización de éstos dentro de un código.
- Niveles de empaquetamiento: Java permite tener métodos y variables visibles sólo para una sección del código bien definido y éste puede ser sólo dentro del mismo método, dentro de la misma clase, dentro del mismo paquete o para toda la aplicación, no es necesario redefinir variables en cada sección del código.
- Los programas desarrollados en Java son susceptibles de ser actualizados mediante la sustitución de sus componentes que son soportados por archivos independientes, en un desarrollo grande se obtiene grandes ventajas del el hecho de sólo tener que cambiar algunos archivos en una actualización.
- Permite el uso y manipulación de cadenas mediante la clase String que significa un avance comparándolo con sus antecesores, en los cuales la manipulación de cadenas, se realiza mediante el uso de una colección de caracteres.
- Se cuenta con algunas clases auxiliares para permitir la manipulación fluida y clara de los tipos de datos auxiliares. Arreglos, vectores y listas de tipos de datos primitivos como clases muy complejas son elementos que se pueden manejar de forma muy clara y sencilla dentro del lenguaje.
- De ser necesario se puede generar un driver de acceso a una base de datos si no se cuenta con éste, siguiendo las reglas generales y los modelos ya establecidos por la comunidad Java.
- Los programas compilados en Java generan un pseudo código máquina que debe ser interpretado por la Java Virtual Machine (JVM) lo que permite que este código sea fácilmente transportable a cualquier plataforma, teniendo únicamente correctamente instalada la JVM particular para este sistema. La portabilidad es una ventaja esencial de Java sobre otros

lenguajes.

- Facilidad de aprendizaje: desde el principio, Java fue pensado para que aquellos interesados encontraran en él una forma sencilla de aprenderlo al liberar los aspectos más confusos y la sintaxis oscura de otros lenguajes.
- Java cuenta con la JCP (Java Community Program), la cual se encarga de realizar por consenso las adecuaciones al programa y mejoras a la nueva versión, siempre teniendo en cuenta la experiencia obtenida por los programadores Java de versiones anteriores e implementando nuevas características que surgen a partir de sus necesidades.
- Java genera “librerías estandarizadas” como un subconjunto de tecnologías orientadas a distintos tipos de aplicaciones.
 - Java SE (Standard Edition) aplicaciones de propósito general.
 - Java EE (Enterprise Edition) para desarrollo de aplicaciones empresariales en servidores,
 - Java ME (Micro Edition) para la programación de dispositivos móviles.

DESVENTAJAS

- Las operaciones matemáticas, en especial las operaciones matemáticas básicas, se convierten en acciones realizadas por una clase de Java en particular lo que podría reducir la expresividad del código.
- Si no se hace una definición correcta de los tipos de datos desde el inicio, se tendrá que redefinir el mismo ya que los tipos y capacidades de las variables no dan oportunidad de crecimiento “dinámico”.
- La manipulación de cadenas mediante la clase String puede resultar costosa en términos de procesamiento y memoria.
- Se hizo patente la necesidad de tener tipos de datos auxiliares a los tipos de datos primitivos, es decir, clases que envuelven a un tipo primitivo, esto crea confusión en los programadores menos experimentados.
- Al no compilarse directamente en lenguaje máquina del CPU, sino en un pseudo lenguaje máquina llamado “byte code”, éste debe ser traducido una

vez más por la máquina virtual, haciendo que las implementaciones particulares en cada CPU sean, en general, de mayor tamaño y menos especializadas en el CPU que se corre, en otros lenguajes como C++ no se requiere este paso.

- Los programas, aunque sean pequeños, compilados en Java suelen ocupar mucha memoria, esto se debe a que la JVM debe cargar diferentes características y bibliotecas para un programa aunque no las utilice.
- Existe un componente de la JVM llamado “Garbage collector” encargado de liberar recursos de memoria que han sido ocupados pero que no tienen más utilidad ni referencia útil dentro del ciclo de vida del programa. Este componente realiza su función pero es poco predecible cuándo es lanzado y este puede consumir mucho tiempo de CPU contribuyendo a la lentitud.

2.4 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DEL MANEJADOR DE BASES DE DATOS MYSQL

MySQL es un sistema de administración de bases de datos relacionales, que maneja el estándar SLQ (**Structured Query Language** – lenguaje de consulta estructurado).

El sistema MySQL fue desarrollado por la empresa llamada MySQL AB, pero, hace un par de años fue adquirido por Oracle, y actualmente MySQL forma parte de los productos de bases de datos que ofrece la compañía Oracle Corporation. Ver en la figura 2.4.1 el logo del producto MySQL.



Figura 2.4.1 Logo de MySQL

Existen varias ediciones, pero sólo una versión es gratuita y de código abierto denominada: MySQL Community Edition, este software tiene una amplia demanda, y actualmente es usado por algunos sitios web muy populares como: Wikipedia, Google (aunque no para búsquedas), Facebook, Twitter, Flickr y YouTube. El software MySQL en su versión gratuita usa la licencia **GPL** (GNU **General Public License** – licencia pública general), para definir lo que puede y no puede hacer con el software en diferentes situaciones.

MySQL soporta varios motores de almacenamiento que tratan con distintos tipos de tabla. Los motores de almacenamiento de MySQL incluyen algunos que tratan con tablas transaccionales y otros que no lo hacen:

- MyISAM trata tablas no transaccionales. Proporciona almacenamiento y recuperación de datos rápida, así como posibilidad de búsquedas fulltext. MyISAM se soporta en todas las configuraciones MySQL, y es el motor de almacenamiento por defecto a no ser que tenga una configuración distinta a la que viene por defecto con MySQL.
- El motor de almacenamiento MEMORY proporciona tablas en memoria. El motor de almacenamiento MERGE permite una colección de tablas MyISAM idénticas ser tratadas como una simple tabla. Como MyISAM, los motores de almacenamiento MEMORY y MERGE tratan tablas no transaccionales y ambos se incluyen en MySQL por defecto.
- Los motores de almacenamiento InnoDB y BDB proporcionan tablas transaccionales. BDB se incluye en la distribución binaria MySQL-Max en aquellos sistemas operativos que la soportan. InnoDB también se incluye por defecto en todas las distribuciones binarias de MySQL 5.0. En distribuciones fuente, puede activar o desactivar estos motores de almacenamiento configurando MySQL a su gusto.
- El motor de almacenamiento EXAMPLE es un motor de almacenamiento "tonto" que no hace nada. Puede crear tablas con este motor, pero no puede almacenar datos ni recuperarlos. El objetivo es que sirva como

ejemplo en el código MySQL para ilustrar cómo escribir un motor de almacenamiento. Como tal, su interés primario es para desarrolladores.

- NDB Cluster es el motor de almacenamiento usado por MySQL Cluster para implementar tablas que se particionan en varias máquinas. Está disponible en distribuciones binarias MySQL-Max 5.0. Este motor de almacenamiento está disponible para Linux, Solaris, y Mac OS X.
- El motor de almacenamiento ARCHIVE se usa para guardar grandes cantidades de datos sin índices con una huella muy pequeña.
- El motor de almacenamiento CSV guarda datos en ficheros de texto usando formato de valores separados por comas.
- El motor de almacenamiento FEDERATED se añadió en MySQL 5.0.3. Este motor guarda datos en una base de datos remota. En esta versión sólo funciona con MySQL a través de la API MySQL C Client. En futuras versiones, será capaz de conectar con otras fuentes de datos usando otros drivers o métodos de conexión clientes.

Hay tres tipos de compilación del servidor MySQL:

- Estándar: los binarios estándar de MySQL son los recomendados para la mayoría de los usuarios, e incluyen el motor de almacenamiento InnoDB.
- Max: los binarios incluyen características adicionales que no han sido lo bastante probadas o que normalmente no son necesarias.
- MySQL-Debug: son binarios que han sido compilados con información de depuración extra, este modo no es recomendable usar en sistemas de producción porque el código de depuración puede reducir el rendimiento.

A continuación se listan algunas de las principales características de MySQL

- Escrito en C y en C++
- Probado con un amplio rango de compiladores diferentes
- Funciona en diferentes plataformas: AIX, HP-UX, GNU/Linux, MacOSX, NetBSD, OpenBSD, Solaris, SunOS, y Windows entre otros.
- Usa GNU Automake, Autoconf, y Libtool para portabilidad.

-
- API's disponibles para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, y Tcl.
 - Opera a gran velocidad, debido al uso completo de **multi-thread** (multihilos) mediante threads del kernel. Pueden usarse fácilmente múltiples CPU's si están disponibles.
 - Proporciona sistemas de almacenamientos transaccionales y no transaccionales.
 - Usa tablas en disco B-tree (MyISAM) muy rápidas con compresión de índice.
 - Joins muy rápidos usando un multi-join de un paso optimizado.
 - Tablas hash en memoria, que son usadas como tablas temporales.
 - Las funciones SQL están implementadas usando una librería altamente optimizada y el acceso a la información es ágil.
 - El servidor está disponible como un programa separado para usar en un entorno de red cliente/servidor. También está disponible como biblioteca y puede ser incrustado en aplicaciones autónomas. Dichas aplicaciones pueden usarse por sí mismas o en entornos donde no hay red disponible.
 - Es un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está cifrado cuando se conecta con un servidor.
 - Los clientes pueden conectar con el servidor MySQL usando sockets TCP/IP en cualquier plataforma. En sistemas Windows de la familia NT (NT, 2000, XP o 2003), los clientes pueden usar named pipes para la conexión. En sistemas Unix, los clientes pueden conectar usando ficheros socket Unix.
 - La interfaz para el conector ODBC (MyODBC) proporciona a MySQL soporte para programas clientes que usen conexiones ODBC (Open Database Connectivity).
 - La interfaz para el conector J MySQL proporciona soporte para clientes Java que usen conexiones JDBC. Estos clientes pueden ejecutarse en Windows o Unix. El código fuente para el conector J está disponible.

-
- El servidor puede proporcionar mensajes de error a los clientes en muchos idiomas.
 - MySQL server tiene soporte para comandos SQL para validar, optimizar, y reparar tablas. Estos comandos están disponibles a través de la línea de comandos y el cliente. MySQL también incluye, una utilidad de línea de comandos muy rápida para efectuar estas operaciones en tablas MyISAM.
 - Existen varias interfaces de programación de aplicaciones que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo C, C++, C#, Pascal, Delphi (vía dbExpress), Eiffel, Smalltalk, Java (con una implementación nativa del driver de Java), Lisp, Perl, PHP, Python, Ruby, Gambas, REALbasic (Mac y Linux), (x)Harbour (Eagle1), FreeBASIC, y Tcl; cada uno de estos utiliza una interfaz de programación de aplicaciones específica.

Ventajas del Manejador de Bases de Datos MySQL

Una de las principales ventajas de MySQL es que maneja una versión gratuita y es de código abierto, aunque tiene muchas otras más ventajas:

- Su instalación es fácil y rápida
- Su configuración y uso es muy sencillo
- Es un software confiable, ya que ha sido “probado” por muchos desarrolladores durante varios años, y los dueños de dicho software han ido corrigiendo los problemas de las versiones anteriores en las diferentes versiones de Producción.
- Es un Manejador de Bases de Datos muy robusto, soporta grandes volúmenes de información (soporta bases de datos de tamaño de terabytes), y tiene baja probabilidad de corromper datos.
- Es un software multiusuario, y de alta velocidad para acceso a los datos.
- Es un sistema multiplataforma, puede ser compilado en más 20 plataformas diferentes.

-
- Replicación MySQL para mejorar el rendimiento de las aplicaciones y la escalabilidad
 - Procedimientos almacenados para mejorar la productividad de los desarrolladores
 - Esquema de rendimiento para la supervisión a nivel de usuario / aplicación de consumo de recursos
 - Conectores MySQL (ODBC, JDBC, NET, etc) para la creación de aplicaciones en múltiples idiomas
 - Cuenta con el módulo MySQL Workbench para el modelado visual, desarrollo y administración de SQL

Desventajas del Manejador de Bases de Datos de MySQL

Existen ciertas limitantes en el software de MySQL, dentro de las más importantes se tienen:

- El procesamiento de vistas (View) no está optimizado
- Los privilegios para una tabla no se eliminan automáticamente cuando se borra una tabla

Por otro lado, aunque actualmente existe un excelente nivel de soporte que proporciona el proveedor para las versiones comerciales de MySQL, esta es una de las desventajas de la versión gratuita, ya que no existe el mismo nivel de Soporte.

2.5 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DEL AMBIENTE CLIENTE-SERVIDOR

Introducción.

El procesamiento de datos se efectúa de muchas maneras y de acuerdo a algunos expertos, aún se encuentra en una etapa de evolución y tardará algunos años en que cambie la forma de como los almacenamos, obtenemos y utilizamos electrónicamente.

El principal motivo detrás de esta evolución es la necesidad que tienen las organizaciones (empresas, instituciones públicas o privadas) de realizar sus operaciones más ágiles y eficientemente, debido a la creciente presión competitiva a la que están sometidas, lo cual, se traduce en la necesidad de que su personal sea más productivo, se reduzcan los costos y gastos de operación, al mismo tiempo que se generan productos y servicios más rápidamente y con mejor calidad.

En este contexto es necesario establecer una infraestructura de procesamiento de información que cuente con los elementos requeridos para proveer información adecuada, exacta y oportuna en la toma de decisiones y para proporcionar un mejor servicio a los clientes.

El ambiente Cliente/Servidor reúne las características necesarias para proveer esta infraestructura, independientemente del tamaño y complejidad de las operaciones de las organizaciones públicas o privadas y consecuentemente desempeña un papel importante en este proceso de evolución. Conocer un ambiente que en este momento es una de las más importantes como arquitecturas y utilizadas en el ámbito de enviar y recibir información, también es una herramienta potente para guardar datos en una base de datos como servidor.

El ambiente Cliente/Servidor es un sistema distribuido entre múltiples procesadores donde hay clientes que solicitan servicios y servidores que los proporcionan. La tecnología Cliente/Servidor es una arquitectura que implica productos y servicios enmarcados en el uso de la tecnología de punta, y permite la distribución de la información en forma ágil y eficaz a las diversas áreas de una organización (empresa, institución pública o privada), así como también fuera de ella.

Elementos principales

Los elementos principales de la arquitectura cliente servidor son justamente los elementos llamados cliente y servidor, y su función es el siguiente:

- Cliente/Servidor es una relación entre procesos corriendo en máquinas separadas.
- El servidor es un proveedor de servicios.
- El cliente es un consumidor de servicios.

Cliente y servidor interactúan por un mecanismo de pasaje de mensajes: Pedido de servicio (petición) y respuesta, como se muestra en la figura 2.5.1.



Figura 2.5.1 Cliente/Servidor, petición y respuesta de servicios.

Estos elementos proporcionan al usuario final, el acceso transparente a los servicios (aplicaciones, datos, etc.) o cualquier otro recurso del grupo de trabajo y/o, a través de la organización, en múltiples plataformas. La arquitectura soporta un medio ambiente distribuido en el cual los requerimientos de servicio hechos por estaciones de trabajo inteligentes o "clientes", resultan en un trabajo realizado por otros computadores llamados servidores", se muestra en la figura 2.5.2.

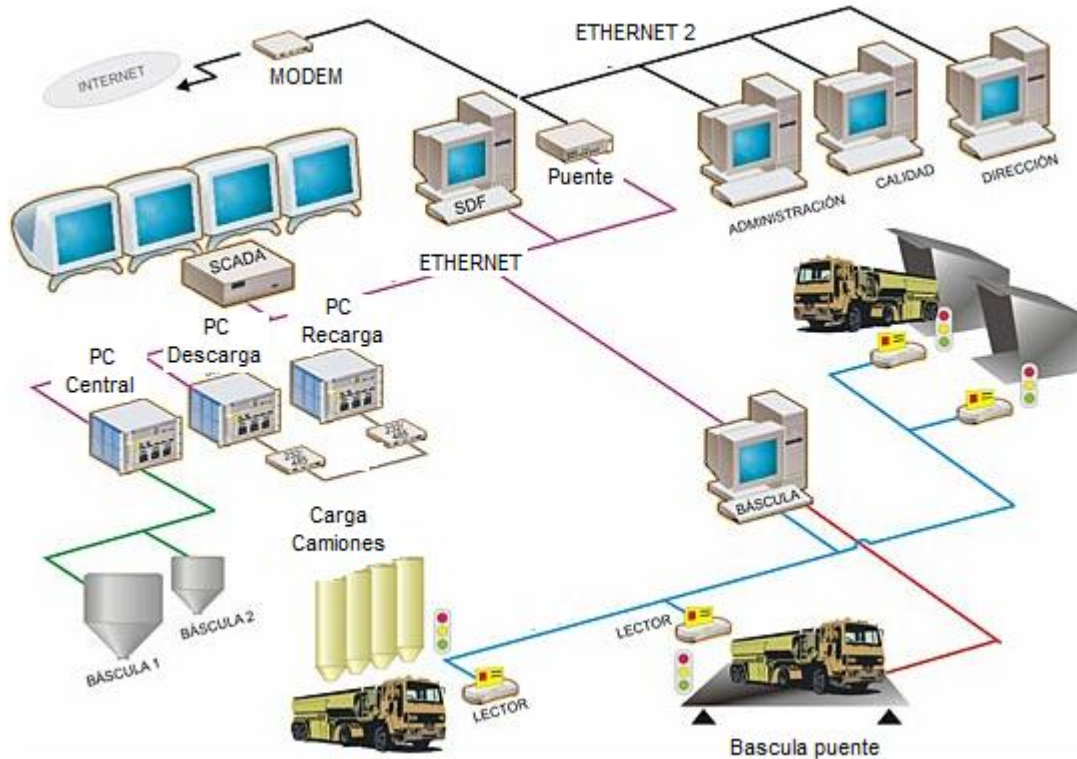


Figura. 2.5.2 Servicios y servidores en múltiples plataformas.

Tradicionalmente la arquitectura cliente / servidor está basada en dos capas el cual contiene un cliente y un servidor. El procesamiento de tareas y la lógica de la aplicación son compartidas entre el servidor de base de datos y el cliente.

Esta arquitectura es donde reside mucho el poder de procesamiento y de la lógica de la aplicación, los clientes pueden estar operando en diferentes plataformas, dando como resultado una distribución compleja de versiones de aplicaciones específicas de las plataformas. Esto hace que el mantenimiento del cliente sea costoso. Ver figura 2.5.3 Arquitectura cliente servidor 2 capas.

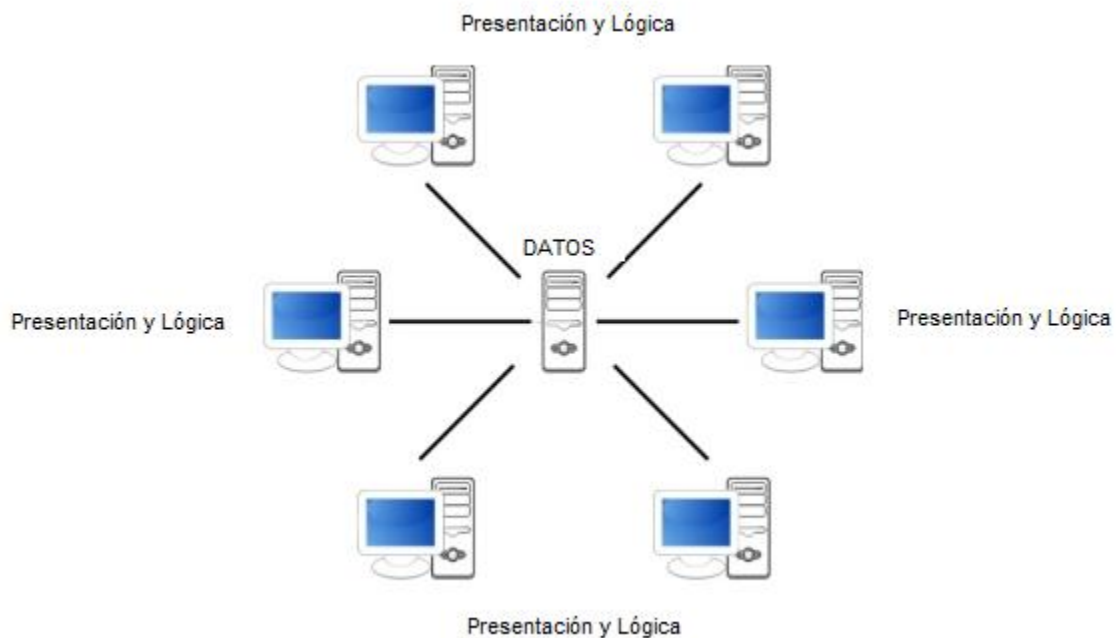


Figura 2.5.3 Arquitectura Cliente Servidor 2 capas.

Las desventajas que caracterizan esta arquitectura son:

- El nivel de las aplicaciones se recargan, entremezclando aspectos típicos del manejo de la interfaz con las reglas del negocio.
- Las reglas del negocio quedan dispersas entre el nivel de aplicación y los procedimientos de la base de datos.
- El nivel de aplicación puede ser demasiado pesado para el cliente.
- Puede existir un cliente delgado y un servidor robusto; esto es lo que generalmente se utiliza en la práctica.

La arquitectura de tres capas es un modelo evolucionado del modelo de dos capas. En este existe una capa intermedia entre el cliente y el servidor, está consiste de un servidor de aplicaciones que contiene el grueso de la lógica de la aplicación, los clientes son livianos o ligeros. Con esta arquitectura la lógica de la aplicación reside en una sola capa que puede ser fácilmente mantenida. El diseño arquitectónico de la capa media puede también ser optimizada en funciones del servidor puesto que éste no tiene que contener u hospedar los servicios.

El software del cliente (capa cliente) es ligeramente pesado, suficiente para ser descargado bajo demanda y lo suficientemente pequeño como para presentar la interfaz del usuario. El grueso de la lógica de la aplicación está implementada ya sea en la capa media (servidor local) o está almacenada en una base de datos. Los clientes son conectados vía LAN a un servidor de aplicaciones local, el cual a su vez se comunica con un servidor central. El servidor local tiene un comportamiento dual, dado que actúa como cliente o servidor en función de la dirección de la comunicación. Algo muy importante y que es la mayor ventaja de esta arquitectura es que ahora únicamente se cambia la regla en el servidor intermedio y ésta actuará en todos los clientes, cosa que no sucedía con la arquitectura en dos capas ya que si alguna regla se cambia, es necesario ir a cada cliente a realizar el cambio. Ver figura 2.5.4 arquitectura cliente servidor 3 capas.

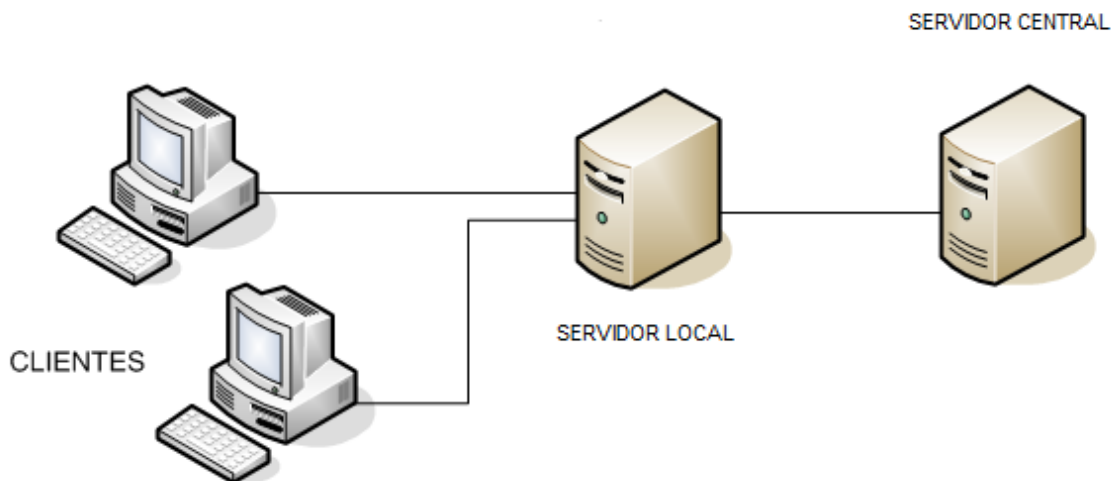


Figura 2.5.4 Arquitectura Cliente Servidor 3 capas.

La mayor desventaja que tiene esta arquitectura es que no se puede realizar un repositorio de datos distribuido, un ejemplo de este tipo de repositorio es el de los bancos que necesitan de un repositorio de datos distribuido para poder tener sus sucursales en otros países y ciudades.

El modelo n-tier (n-capas) de informática distribuida ha emergido como la arquitectura predominante para la construcción de aplicaciones multiplataforma en la mayor parte de las empresas. Como tecnología, las arquitecturas de n-capas proporcionan una gran cantidad de beneficios para las empresas que necesitan soluciones flexibles y fiables para resolver complejos problemas inmersos en cambios constantes.

Todas las aplicaciones basadas en n-capas permitirán trabajar con clientes ligeros, tal como navegadores de Internet, WebTV, Teléfonos Inteligentes, PDAs (Personal Digital Assistants o Asistentes Personales Digitales) y muchos otros dispositivos preparados para conectarse a Internet. De este modo, las arquitecturas de n-capas se están posicionando rápidamente como la piedra angular de los desarrollos de aplicaciones empresariales y las compañías están adoptando esta estrategia a una velocidad de vértigo como mecanismo de posicionamiento en la economía emergente que tiene su base en la red (lo que se ha venido a denominar "nueva economía"). Las arquitecturas basadas en n-capas permiten a los componentes de negocio correr en una LAN, WAN o Internet. Esto significa que cualquiera con un ordenador y conexión a la red posee toda la funcionalidad que tendría si se encontrase delante de su sistema de escritorio.

Los sistemas distribuidos de n-capas proporcionan un conjunto de avances tecnológicos sin precedentes, como pooling de conexiones, multiplexado de conexiones, balanceo de carga dinámico y rendimientos excelentes en hardware trabajando en cluster. Las aplicaciones industriales basadas en n-capas pueden ayudar a las compañías a desarrollar un nuevo núcleo de habilidades en prácticamente todo, desde la gestión del conocimiento hasta los sistemas relacionados con comercio electrónico.

Las arquitecturas basadas en n-capas son el siguiente paso lógico en un proceso de evolución, está basado en las arquitecturas convencionales cliente-servidor (2 y 3 capas) más la convergencia de dos tecnologías tremendamente potentes:

-
- Desarrollo de aplicaciones basadas en componentes.- relacionado directamente con la programación orientada a objetos (lenguajes y técnicas)
 - Internet.- primer ejemplo de un sistema complejo de n-capas cliente-servidor.

Estas y otras tecnologías en rápida evolución proporcionan la infraestructura necesaria y la fontanería relacionada que permite a las compañías operar en un entorno complejo, multiplataforma y con capacidades de computación distribuida, tanto interna como externamente según se requiera en cada caso.

El desarrollo de aplicaciones en n-capas es un proceso iterativo de división del problema en piezas manejables denominadas componentes. Estos componentes, o "componentes de negocio - Business Objects" son "modelos software" basados típicamente en la "vista" de un objeto real, evento o proceso de negocio. Los componentes software individuales pueden formar parte y adaptarse tanto de estructuras independientes como de sistemas colaborativos.

El diseño de aplicaciones en n-capas es ideal para la creación de sistemas adaptables, donde cada componente puede ser utilizado y reutilizado en nuevas combinaciones para satisfacer requisitos de negocio dinámicos. Esto permite a los desarrolladores y a las nuevas aplicaciones reutilizar componentes existentes que modelan la lógica del negocio sobradamente probada. En un entorno tremendamente cambiante como el actual, utilizar aplicaciones basadas en diseños de n-capas posibilita a las empresas ser más ágiles y adaptables en proporcionar valor a sus clientes. Los sistemas basados en n-casps tienen el potencial de reducir drásticamente tanto el time-to-market para las nuevas aplicaciones de negocio, como el coste total de mantenimiento, adaptando estos complejos y caros sistemas a las siempre cambiantes necesidades empresariales. Ver la figura 2.5.5 arquitectura cliente servidor n capas.

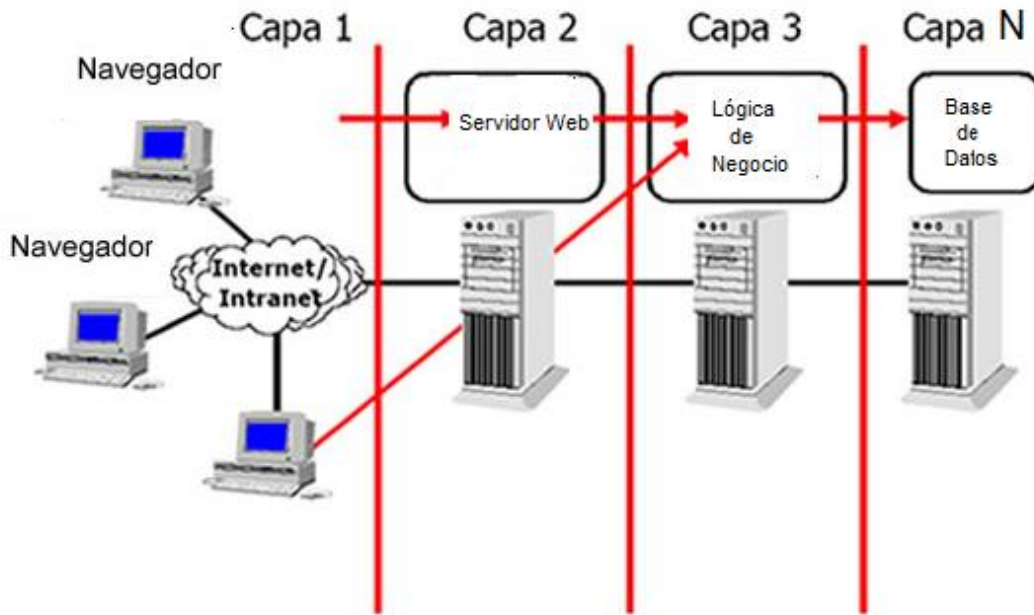


Figura 2.5.5 Arquitectura Cliente Servidor N capas.

Características.

Las características básicas de una arquitectura cliente/servidor son:

- El cliente y el servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas, realizando actividades o tareas independientes.
- Interacción que existe entre un cliente y el usuario, y un servidor que interactúa con los recursos compartidos. El proceso del cliente proporciona la interfaz entre el usuario y el resto del sistema. El proceso del servidor actúa como un motor de software que maneja recursos compartidos tales como bases de datos, impresoras, módems, etc.
- Las tareas del cliente y del servidor tienen diferentes requerimientos en cuanto a recursos de cómputo como velocidad del procesador, memoria, velocidad y capacidades del disco y dispositivos de entrada - salida.
- Se establece una relación entre procesos distintos, los cuales pueden ser ejecutados en la misma máquina o en máquinas diferentes distribuidas a lo largo de la red.

-
- Existe una clara distinción de funciones basada en el concepto de "servicio", que se establece entre clientes y servidores.
 - La relación establecida puede ser de muchos a uno, en la que un servidor puede dar servicio a muchos clientes, regulando su acceso a recursos compartidos.
 - Los clientes corresponden a procesos activos en cuanto a que son éstos los que hacen peticiones de servicios a los servidores. Estos últimos tienen un carácter pasivo ya que esperan las peticiones de los clientes.
 - No existe otra relación entre clientes y servidores que no sea la que se establece a través del intercambio de mensajes entre ambos. El mensaje es el mecanismo para la petición y entrega de solicitudes de servicio.
 - El ambiente es heterogéneo. La plataforma de hardware y el sistema operativo del cliente y del servidor no son siempre la misma. Precisamente una de las principales ventajas de esta arquitectura es la posibilidad de conectar clientes y servidores independientemente de sus plataformas.
 - El concepto de escalabilidad tanto horizontal como vertical es aplicable a cualquier sistema Cliente/Servidor. La escalabilidad horizontal permite agregar más estaciones de trabajo activas sin afectar significativamente el rendimiento. La escalabilidad vertical permite mejorar las características del servidor o agregar múltiples servidores.
 - La interrelación entre el hardware y el software están basados en una infraestructura poderosa, de tal forma que el acceso a los recursos de la red no muestra la complejidad de los diferentes tipos de formatos de datos y de los protocolos.
 - Un sistema de servidores realiza múltiples funciones al mismo tiempo que presenta una imagen de un solo sistema a las estaciones Clientes.
 - Además se constituye como el nexo de unión más adecuado para reconciliar los sistemas de información basados en mainframes o minicomputadores, con aquellos otros sustentados en entornos informáticos pequeños y estaciones de trabajo.

Ventajas.

Entre las principales ventajas del esquema cliente/servidor están:

- La existencia de plataformas de hardware cada vez más baratas. Esta constituye a su vez una de las más palpables ventajas de este esquema, la posibilidad de utilizar máquinas considerablemente más baratas que las requeridas por una solución centralizada, basada en sistemas grandes. Además, se pueden utilizar componentes, tanto de hardware como de software, de varios fabricantes, lo cual contribuye considerablemente a la reducción de costos y favorece la flexibilidad en la implementación y actualización de soluciones.
- Facilita la integración entre sistemas diferentes y comparte información permitiendo, por ejemplo que las máquinas ya existentes puedan ser utilizadas pero utilizando interfaces más amigables al usuario. De esta manera, podemos integrar PCs con sistemas medianos y grandes, sin necesidad de que todos tengan que utilizar el mismo sistema operacional.
- Favorece el uso de interfaces gráficas interactivas, los sistemas construidos bajo este esquema tienen mayor interacción y más intuitiva con el usuario debido a que en el uso de interfaces gráficas para el usuario, el esquema cliente/servidor 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.
- Es rápido el mantenimiento y el desarrollo de aplicaciones, pues se pueden emplear las herramientas existentes (por ejemplo los servidores de SQL o las herramientas de más bajo nivel como los sockets o el RPC).
- La estructura inherentemente modular facilita además la integración de nuevas tecnologías y el crecimiento de la infraestructura computacional, favoreciendo así la escalabilidad de las soluciones.
- Proporciona a los diferentes departamentos de una organización, soluciones locales, pero permitiendo la integración de la información relevante a nivel global.

-
- Es más fácil de administrar una arquitectura cliente/servidor cuando es grande porque la administración se encuentra centralizada.

Desventajas.

Entre las principales desventajas del ambiente Cliente/Servidor están:

- El mantenimiento de los sistemas es más difícil pues implica la interacción de diferentes partes de hardware y de software, distribuidas por distintos proveedores, lo cual dificulta el diagnóstico de fallas.
- Se cuenta con muy escasas herramientas para la administración y ajuste del desempeño de los sistemas.
- Es importante que los clientes y los servidores utilicen el mismo mecanismo (por ejemplo Sockets o RPC), lo cual implica que se deben tener mecanismos generales que existan en diferentes plataformas, además hay que tener estrategias para el manejo de errores y para mantener la consistencia de los datos.
- La seguridad es otra preocupación importante, dado que las verificaciones se tienen que realizar en el cliente y en el servidor.
- El desempeño es otro de los aspectos que se deben tener en cuenta. Problemas de este estilo pueden presentarse por congestión en la red, dificultad de tráfico de datos, etc.

CAPÍTULO

III

PLANTEAMIENTO DEL PROBLEMA

PLANTEAMIENTO DEL PROBLEMA

3.1 ANÁLISIS DEL PROBLEMA.

El análisis del problema es la fase fundamental para resolver este. Esta fase requiere normalmente el máximo de imaginación y creatividad.

El gobierno electrónico o e-gobierno es un concepto que incluye el uso de tecnologías de la información para proporcionar a los ciudadanos facilidad en procesos de gestión pública y así aumentar la eficiencia de los servicios públicos reduciendo costos en consulta de información, trámites y servicios burocráticos.

En México a nivel federal y estatal se está participando activamente en importantes esfuerzos de iniciativas de e-gobierno para integrar soluciones de tecnología que ayuden a sustituir funciones tradicionales para así facilitar las operaciones de los gobiernos con el uso de tecnologías de la información. Como parte de estas iniciativas en el año 2004 el Servicio de Administración Tributaria (SAT) propuso un esquema masivo para la emisión de Comprobantes Fiscales Digitales en su formato de factura electrónica que permite otorgar un medio de respaldo de las operaciones comerciales que realizan los contribuyentes y así remplazar las facturas de papel que se utilizan en la actual normativa, vea la figura 3.1.1 Generación de CFD`s.

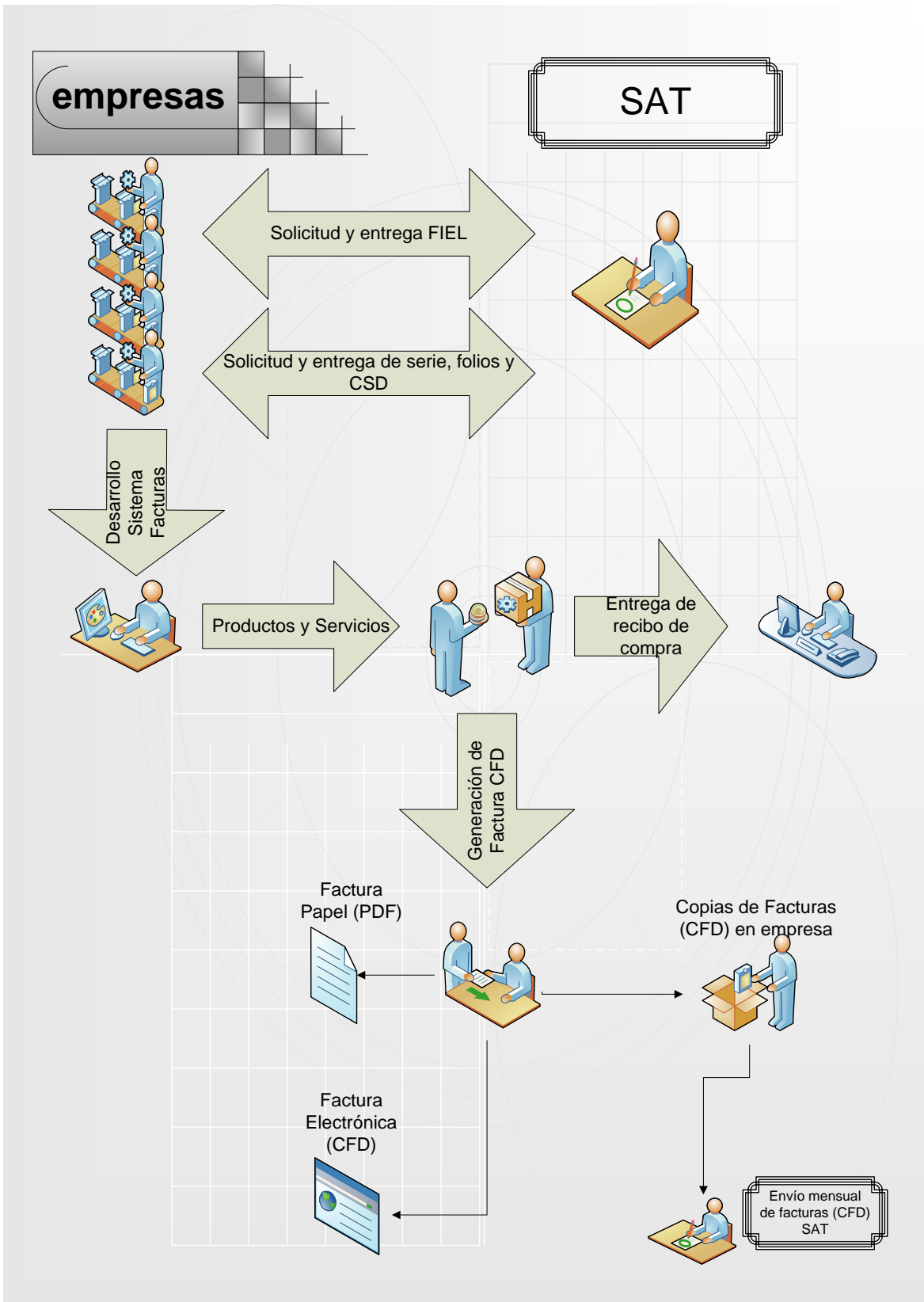


Figura 3.1.1 Generación de CFD's

Contar con documentos digitales tiene muchos beneficios entre los que se encuentran:

- Agilizan la información contable.
- Almacenamiento en formato digital.
- Reducción de costos y errores en el proceso de generación captura.
- Entrega y almacenamiento.

Sin embargo, debido a la aparición de este tipo de documentos digitales existe la obligación de replantearse muchas cuestiones de los documentos tradicionales en papel, surgiendo nuevos problemas, e incluso agudizando algunos de los ya existentes.

En esta lista de problemas, se plantean cuestiones que van, desde la validez legal de los documentos digitales, la alteración de la información contenida en estos o que la información sea comprometida. Debido a esto el Sistema de Administración Tributaria ha incluido ciertos mecanismos para verificar la legitimidad de una factura electrónica. Sin embargo, el uso de estos mecanismos no han resuelto todos los problemas y siguen presentándose casos de comprobantes apócrifos. De aquí que actualmente se está implementando la Facturación Electrónica por Internet o bien la emisión del Comprobante Fiscal Digital por Internet (CFDI) como medida obligatoria. Figura 3.1.2 Generación de CFDI`s.

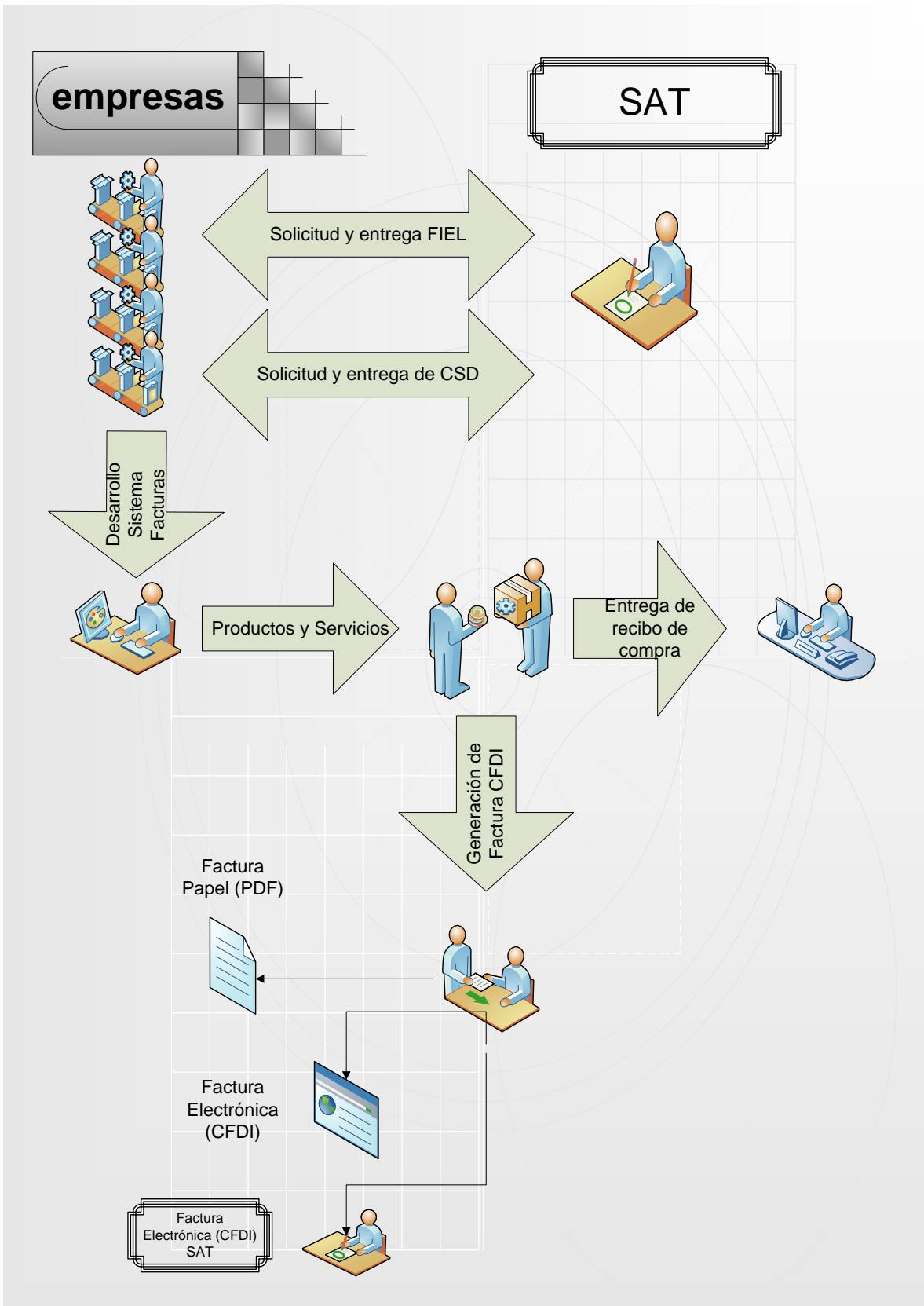


Figura 3.1.2 Generación de CFDI's

Nuestro sistema deberá de realizar la entrega de los CFDI's en la manera solicitada por el SAT, la cual incluye una certificación por un proveedor autorizado de certificación, el cual regresará un timbrado que da validez para que de esta manera poder reportar la operación y realizar la entrega correspondiente por Internet, así mismo emitirá un documento impreso y digital el cual se enviara al solicitante solo para su fácil entendimiento y al SAT para los impuestos correspondientes y la declaración. Vea la figura 3.1.3 Detalle de generación de CFDI's.

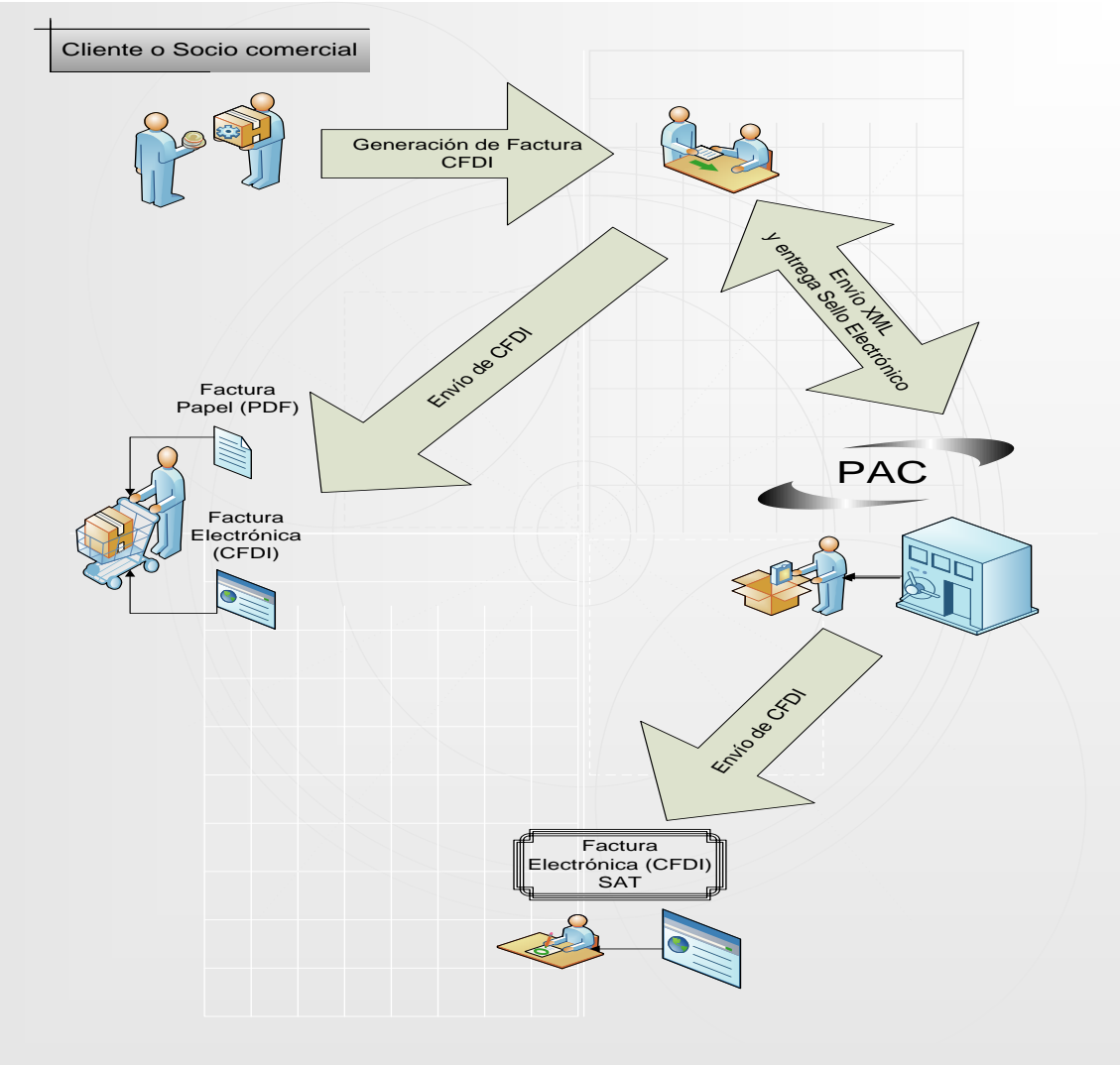


Figura 3.1.3 Detalle generación de CFDI's

Durante el proceso para la generación y expedición de comprobantes fiscales digitales por Internet se encontraron algunos inconvenientes, los cuales, se enumeran a continuación:

- Generación de los datos necesarios a partir del Frond-end de la empresa.
- Formato de datos de la compra de servicio o producto al que se requiere realizar factura o CFDI.
- Captura de datos para su validación y generación de archivo en el formato correspondiente .xml.
- Envío de datos al PAC para su validación y timbrado.
- Recepción de datos al PAC validado y timbrado.
- Proceso de transformación de datos o CFDI de formato .xml a formato .pdf.
- Resguardo de información del proceso de generación de los CFDI`s

Se proponen las siguientes soluciones a los inconvenientes encontrados.

- Realizar la revisión de datos generada en el archivo de tipo texto, y realizar cotejo con los datos que solicita como obligatorios el SAT para poder realizar o expedir CFDI`s.
- Se propone dejar todos los datos en un archivo de tipo texto, de tal manera que cualquier programa pueda leer y validar, cada uno de los campos que se llenan en el Front-End deberán estar en una línea diferente para poder ser procesados.
- Para realizar la captura/lectura y validación de los datos, se propone utilizar lenguaje de programación JAVA, el cual nos puede validar información por medio de rutinas de comprobación y generar un archivo de tipo .xml.
- Para poder enviar los datos por Internet, se propone utilizar el lenguaje de programación JAVA, el cual puede enviar la información en arreglo de bytes, ya que es la forma en que la solicita el proveedor autorizado de certificación para el timbrado de los datos de facturación.
- Para la recepción de datos, se propone utilizar el lenguaje de programación JAVA, el cual puede recibir la información proveniente del proveedor

autorizado de certificación, ya sea con el timbre fiscal realizado de los datos de la factura o con el error correspondiente.

- Para la conversión de los archivos que se generan en un formato .xml, se propone realizar una conversión con el lenguaje de programación JAVA a través de módulos o librerías que realiza esta acción.
- El resguardo se propone realizar una tabla de historia, es decir, una tabla que contenga todos los movimientos y datos de los CFDI's que sirva como repositorio histórico, para poder acceder en el momento que se requiera.

3.2 REQUERIMIENTOS GENERALES Y PARTICULARES DE LA APLICACIÓN

La definición de requerimiento es el acto o la consecuencia de requerir, este verbo que tiene su origen etimológico en el término latino *requirere*, refiere a solicitar, pedir, avisar o necesitar algo.

En ingeniería de software un requerimiento es una necesidad documentada sobre el contenido, forma o funcionalidad de un producto o servicio. Los requerimientos son declaraciones, atributos, capacidades, características y/o cualidades que necesita cumplir un sistema para que tenga valor y utilidad para el usuario.

En el modelo clásico del desarrollo del software, esta etapa viene antecedida por la etapa de factibilidad del sistema y precedida por la etapa del diseño de software.

Existen diferentes tipos de requerimientos:

-
- **Requerimientos funcionales.-** Este tipo de requerimientos son las declaraciones de los servicios que el sistema deberá proporcionar, la manera en que este tendrá que reaccionar a entradas particulares y como se debe de comportar ante situaciones particulares. Este tipo de requerimientos dependen del tipo de software que se desarrolle, de los usuarios del software y del enfoque general tomado por la organización al redactar los requerimientos.

La especificación de requerimientos funcionales de un sistema debe estar completa y ser consistente, esto quiere decir que todos los servicios solicitados por el usuario deben estar definidos y no deben de tener definiciones contradictorias.

- **Requerimientos no funcionales.-** Este tipo de requerimientos son aquellos que no se refieren directamente a las funciones específicas que proporciona el sistema , si no a las propiedades emergentes de dicho sistema, tales como la calidad, el rendimiento, facilidad de uso, tiempo de respuesta y la capacidad de almacenamiento entre otras. De forma alterna definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y las representaciones de datos que se utilizan en las interfaces del sistema.
- **Requerimientos Externos.-** Este tipo de requerimiento son aquellos a los cuales el sistema se debe de atener con respecto a su entorno, es decir todos aquellos que se derivan de los factores externos los sistemas y su proceso de desarrollo. Estos incluyen:
 - Los requerimientos de interoperabilidad que definen la manera en que el sistema interactúa con los otros sistemas/hardware de la organización.
 - Los requerimientos legales que deben de seguirse para asegurar que el sistema opere dentro de la ley.

-
- Los requerimientos éticos, que son impuestos al sistema para asegurar que será aceptado por el usuario y por el público en general.

Requerimientos generales de la aplicación

Los requerimientos generales con los que deberá contar la aplicación al terminar su desarrollo serán los siguientes:

- La aplicación deberá ser multiplataforma por lo que podrá ejecutarse en cualquier sistema operativo bajo versiones de 32 o 64 bits respectivamente.
- La interfaz de usuario debe estar basada en una interfaz gráfica desarrollada con lenguajes de programación orientados a web, deberá ser una interfaz sencilla, moderna y amigable, que facilitará el llenado de datos requeridos para la facturación así como el manejo de la misma.

Los lenguajes de programación orientados a web que se utilizarán serán los siguientes:

- PHP: Es un lenguaje de código abierto especialmente adecuado para el desarrollo web y puede ser incrustado en HTML.
 - Javascript: Se utiliza principalmente del lado del cliente permitiendo crear efectos atractivos y dinámicos en las páginas web.
- La pantalla deberá tener un aspecto homogéneo, es decir, que se presenten las mismas características en cuanto a posición de los controles, tipo de letra y mensajes informativos.
 - Todos los datos serán almacenados en una base de datos utilizando el manejador de base de datos MySQL, ya que es de uso gratuito y se acopla al desarrollo de la aplicación.

Requerimientos particulares de la aplicación

La aplicación deberá de tener los siguientes requerimientos particulares:

- Dentro de la interfaz gráfica deberá contar con un módulo operativo el cual contendrá el botón de envío de facturas una vez que se hayan llenado los datos requeridos.
- Tendrá la capacidad de generar un archivo XML que cumpla con los requisitos establecidos en el Anexo 20 de la Resolución de Miscelánea Fiscal para 2014 para Comprobantes Fiscales Digitales por Internet.
- Enviará este archivo en formato XML a un Proveedor Autorizado de Certificación para que éste timbre el documento.
- Hará la recepción del documento timbrado (y por tanto convertido en un Comprobante Fiscal Digital por Internet) desde el servicio del PAC.
- Generará una representación de este documento como un archivo PDF listo para imprimir o para enviar por correo electrónico.
- Hará el envío por correo electrónico tanto el archivo XML como el archivo PDF producto de esta transacción al cliente.
- Almacenará en los distintos repositorios destinados los datos generados de la transacción.
- En su caso, hacer el manejo de errores en las distintas etapas del proceso de generación de CFDI's.

3.3 ANÁLISIS Y POSICIONAMIENTO DE LOS LENGUAJES PARA EL BACK-END Y FRONT-END

Análisis y posicionamiento de las bases de datos para el back-end.

A continuación se analizan los tres distintos manejadores de bases de datos con el fin de obtener bases suficientes que nos permitan elegir alguno como la mejor opción para la instalación y ejecución del proyecto SCFDI.

Firebird



Figura 3.3.1 Logo de Firebird

Firebird (véase el logotipo en la figura 3.3.1) es un sistema de administración de base de datos relacional (o RDBMS) (Lenguaje consultas: SQL) de código abierto, basado en la versión 6 de Interbase, cuyo código fue liberado por Borland en 2000. Su código fue reescrito de C a C++. El proyecto se desarrolla activamente y el 18 de abril de 2008 fue liberada la versión 2.1.

Algunas de sus características son:

- Es multiplataforma, y actualmente puede ejecutarse en los sistemas operativos: Linux, HP-UX, FreeBSD, Mac OS, Solaris y Microsoft Windows.
- Ejecutable pequeño, con requerimientos de hardware bajos.
- Arquitectura Cliente/Servidor sobre protocolo TCP/IP y otros (embedded).
- Soporte de transacciones ACID y claves foráneas.
- Es medianamente escalable.
- Buena seguridad basada en usuarios/roles.
- Diferentes arquitecturas, entre ellas el Firebird incrustado (embedded server) que permite ejecutar aplicaciones monousuario en ordenadores sin instalar el software Firebird.
- Bases de datos de sólo lectura, para aplicaciones que corran desde dispositivos sin capacidad de escritura, como CD-ROM's.

-
- Existencia de controladores ODBC, OLEDB, JDBC, PHP, Perl, .net, etc.
 - Requisitos de administración bajos, siendo considerada como una base de datos libre de mantenimiento, al margen de la realización de copias de seguridad.
 - Pleno soporte del estándar SQL-92, tanto de sintaxis como de tipos de datos.
 - Completo lenguaje para la escritura de disparadores y procedimientos almacenados denominado PSQL.
 - Capacidad de almacenar elementos BLOB (Binary Large Objects).
 - Soporte de User-Defined Functions (UDFs).
 - Versión auto ejecutable, sin instalación, excelente para la creación de catálogos en CD-ROM y para crear versiones de evaluación de algunas aplicaciones.

Podría considerarse un tercer tipo, el Embedded. Éste consiste en una única biblioteca de enlace dinámico DLL (de unos 2 MB de tamaño) que contiene todo el servidor. De esta forma se puede tener un DBMS completo disponible y distribuible junto con aplicaciones de usuario sin requerir que este se instale por separado.

MySQL



Figura 3.3.2 Logo de MySQL

Es un sistema gestor de bases de datos relacionales en SQL (véase el logotipo en la figura 3.3.2), permite la gestión de los datos de una Base de Datos relacional usando un lenguaje de consulta estructurado, esto significa que, a partir de una oración MySQL llevará a cabo una determinada acción sobre nuestra base de datos.

MySQL es un excelente gestor de bases de datos que la sitúan después de Oracle cómo la mejor solución a nivel técnico por las características que posee.

- Código abierto
- Facilidad
- Funcionalidad
- Portabilidad
- Velocidad

Código abierto: MySQL es una aplicación de Código abierto y gratuita, nos permite redistribuir una aplicación que la contenga y nos permite incluso modificar su código para mejorarla o adaptarla a nuestras necesidades.

Además, existe la seguridad de contar con una importante cuota de mercado y de saber que es una solución estable, mantenida por un buen equipo de desarrolladores e incluso con soporte de pago.

Facilidad: MySQL es un sistema fácil de instalar y configurar en servidores Windows, Linux... y enseguida nos permite ejecutar sencillas consultas de SQL.

Funcionalidad: quizás, en este apartado, MySQL se encuentra deficiente frente a sus rivales, pero sin embargo dispone de muchas funciones vitales para el desarrollo profesional cómo puede ser el volcado online y la duplicación.

Portabilidad: MySQL puede correr en la inmensa mayoría de sistemas operativos, por lo que junto a otro lenguaje de programación del lado del servidor de alta portabilidad como Java o PHP nos permite el desarrollo de aplicaciones web

fáciles de migrar y el acceso y copia de los datos desde cualquier Sistema Operativo.

Velocidad: MySQL es un sistema con una velocidad superior a sus rivales a la hora de realizar instrucciones SQL.

Postgresql



Figura 3.3.3 Logo de PostgreSQL

PostgreSQL (véase el logotipo en la figura 3.3.3) es una base de datos relacional, distribuida bajo licencia BSD y con su código fuente disponible libremente. Es uno de los motores de bases de datos de código abierto más potente del momento.

Sus características técnicas la hacen una de las bases de datos más potentes y robustas del mercado. Su desarrollo comenzó hace más de 15 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. En los últimos años se han concentrado mucho en la velocidad de proceso y en características demandadas en el mundo empresarial.

PostgreSQL se puede ejecutar en la gran mayoría de sistemas operativos existentes en la actualidad, entre ellos Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows. Algunas características más importantes y soportadas son:

- Es una base de datos 100% ACID

-
- Llaves ajenas (foreign keys)
 - Joins
 - Vistas (views)
 - Disparadores (triggers)
 - Reglas (Rules)
- Funciones/procedimientos almacenados (stored procedures) en numerosos lenguajes de programación, entre otros PL/pgSQL (similar al PL/SQL de oracle)

Otra característica muy a tener en cuenta es lo bien que PostgreSQL funciona con grandes cantidades de datos y una alta concurrencia, con muchos usuarios accediendo a la vez el sistema.

Comparativa entre los distintos SGBD analizados, vea tabla 3.3.1

	FIREBIRD	MYSQL	POSTGRESQL
Número máximo de decimales que puede manejar en tipos decimales	18	65	131072
Longitud máxima en tipos de datos varchar	32767	65535	1 GB
Operación de concatenación	(doble pipe)	CONCAT()	(doble pipe)
Carácter delimitador	N/A	"	N/A
Uso de FOREIGN KEY	Variantes de sintaxis	InnoDB	SQL estándar
Cláusula CHECK	OK	X	OK
Declaración GRANT	grant to PUBLIC	grant TRIGGER	grant TRIGGER

	FIREBIRD	MYSQL	POSTGRESQL
Precisión en el tipo de dato TIMESTAMP	4 dígitos	6 dígitos	14 dígitos
Funciones SQL definidas por el usuario	Sólo procedimientos	Sintaxis Estándar	Múltiples lenguajes
Tabla information_schema	X	OK	X
Característica GET DIAGNOSTICS	X	OK	X
Función TRUNCATE	X	OK	OK
Función CAST	OK directo	X	OK función
Valores por defecto.	CREATE	CREATE UPDATE INSERT	CREATE
Soporte de dominio	OK	X	X
Declaración MERGE	OK	X	OK
Longitud de caracteres en identificadores largos	31	64	63 bytes
Outer join completo	OK	X	OK
Tablas temporales	OK	No estándar	OK
Tipos BINARY y VARBINARY	X	OK	Solo binary
Querys recursivos	OK	X	OK
Declaración LIKE en la definición de tabla	X	OK	OK
Cláusula AS en la definición de tabla	X	OK	OK
Generación de triggers		OK	OK

	FIREBIRD	MYSQL	POSTGRESQL
Otorgamiento de privilegios a nivel columna	X	OK	X
Replicación	X	Master – master master – slave Cluster	Master - slave
Integración con Hibernate Framework	Mantenido con integración inmediate en IDE	Mantenido	Mantenido con integración inmediate en IDE

Tabla Comparativa 3.3.1

A continuación se analizan las tres distintas opciones de lenguajes de programación disponibles para generar la interfaz de usuario para el Sistema para Comprobantes Fiscales Digitales por Internet.

PHP



Figura 3.3.4. Logotipo de PHP

PHP (véase el logotipo en la figura 3.3.4) -acrónimo recursivo de PHP: Hypertext Preprocessor- es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

En PHP el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no sabrá el código subyacente que hay. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga.

PHP está enfocado principalmente a la programación de scripts del lado del servidor, por lo que se puede hacer cualquier cosa que pueda hacer otro programa CGI, como recopilar datos de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies.

Existen principalmente tres campos donde se usan scripts de PHP.

- Scripts del lado del servidor.
- Scripts desde la línea de comandos.
- Escribir aplicaciones de escritorio.

PHP puede emplearse en todos los sistemas operativos principales, incluyendo Linux, muchas variantes de Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS y probablemente otros más. PHP admite la mayoría de servidores web de hoy en día, incluyendo Apache, IIS, y muchos otros. Esto incluye cualquier servidor web que pueda utilizar el binario de PHP FastCGI, como lighttpd y nginx. PHP funciona tanto como módulo como procesador de CGI.

De modo que con PHP, se tiene la libertad de elegir el sistema operativo y el servidor web. Además, se tiene la posibilidad de utilizar programación por procedimientos o programación orientada a objetos (POO), o una mezcla de ambas.

Una de las características más potentes y destacables de PHP es su soporte para un amplio abanico de bases de datos. Escribir una página web con acceso a una base de datos es simple utilizando una de las extensiones específicas de bases de datos, o utilizar una capa de abstracción como PDO, o conectarse a cualquier base de datos que admita el estándar de Conexión Abierta a Bases de Datos por medio de la extensión ODBC.

PHP también cuenta con soporte para comunicarse con otros servicios usando protocolos tales como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (en Windows) y muchos otros. También se pueden crear sockets de red puros e interactuar usando cualquier otro protocolo. PHP tiene soporte para el intercambio de datos complejos de WDDX entre virtualmente todos los lenguajes de programación web. Y hablando de interconexión, PHP tiene soporte para la instalación de objetos de Java y emplearlos de forma transparente como objetos de PHP.

PHP tiene útiles características de procesamiento de texto, las cuales incluyen las expresiones regulares compatibles con Perl (PCRE), y muchas extensiones y herramientas para el acceso y análisis de documentos XML. PHP estandariza todas las extensiones XML sobre el fundamento sólido de libxml2, y amplía este conjunto de características añadiendo soporte para SimpleXML, XMLReader y XMLWriter.

C#



Figura 3.3.5 Logotipo de C#

C# (véase el logotipo en la figura 3.3.5) es un lenguaje orientado a objetos que permite a los desarrolladores compilar diversas aplicaciones sólidas y seguras que se ejecutan en .NET Framework. Se puede utilizar C# para crear aplicaciones cliente de Windows, servicios Web XML, componentes distribuidos, aplicaciones cliente-servidor, aplicaciones de base de datos, y más. Visual C# proporciona un editor de código avanzado, cómodos diseñadores de interfaz de usuario, depurador integrado y numerosas herramientas más para facilitar el desarrollo de aplicaciones basadas el lenguaje C# y .NET Framework.

La sintaxis de C# es muy expresiva, sencilla y fácil de aprender y cualquier programador con alguna experiencia en C++ o Java pueden empezar a trabajar de forma productiva en C# en un plazo muy breve. La sintaxis de C# simplifica muchas de las complejidades de C++ y proporciona características eficaces tales como tipos de valor que admiten valores NULL, enumeraciones, delegados, expresiones lambda y acceso directo a memoria, que no se encuentran en Java. C# admite métodos y tipos genéricos, que proporcionan mayor rendimiento y seguridad de tipos, e iteradores, que permiten a los implementadores de clases de colección definir comportamientos de iteración personalizados que el código cliente puede utilizar fácilmente. Las expresiones Language-Integrated Query (LINQ) convierten la consulta fuertemente tipada en una construcción de lenguaje de primera clase.

Como lenguaje orientado a objetos, C# admite los conceptos de encapsulación, herencia y polimorfismo. Todas las variables y métodos, incluido el método Main que es el punto de entrada de la aplicación, se encapsulan dentro de definiciones de clase. Una clase puede heredar directamente de una clase primaria, pero puede implementar cualquier número de interfaces. Los métodos que reemplazan a los métodos virtuales en una clase primaria requieren la palabra clave override como medio para evitar redefiniciones accidentales.

Además de estos principios básicos orientados a objetos, C# facilita el desarrollo de componentes de software a través de varias construcciones de lenguaje innovadoras.

JAVA

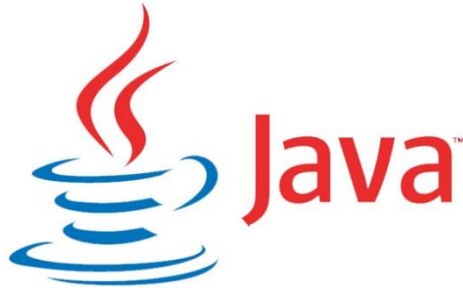


Figura 3.3.6 Logotipo de Java

La principal característica de Java (véase el logotipo en la figura 3.3.6) es la de ser un lenguaje compilado e interpretado. Todo programa en Java ha de compilarse y el código que se genera bytecodes es interpretado por una máquina virtual. De este modo se consigue la independencia de la máquina, el código compilado se ejecuta en máquinas virtuales que si son dependientes de la plataforma.

Java es un lenguaje orientado a objetos de propósito general. Aunque Java comenzara por ser conocido como un lenguaje de programación de applets que se ejecutan en el entorno de un navegador web, se puede utilizar para construir cualquier tipo de proyecto. Su sintaxis es muy parecida a la de C y C++ pero hasta ahí llega el parecido. Java no es una evolución ni de C++ ni un C++ mejorado.

En el diseño de Java se prestó especial atención a la seguridad. Existen varios niveles de seguridad en Java, desde el ámbito del programador, hasta el ámbito de la ejecución en la máquina virtual.

Con respecto al programador, Java realiza comprobación estricta de tipos durante la compilación, evitando con ello problemas tales como el desbordamiento de la pila. Pero, es durante la ejecución donde se encuentra el método adecuado según el tipo de la clase receptora del mensaje; aunque siempre es posible forzar un enlace estático declarando un método como final.

Todas las instancias de una clase se crean con el operador `new()`, de manera que un recolector de basura se encarga de liberar la memoria ocupada por los objetos que ya no están referenciados. La máquina virtual de Java gestiona la memoria dinámicamente.

Una fuente común de errores en programación proviene del uso de punteros. En Java se han eliminado los punteros, el acceso a las instancias de clase se hace a través de referencias.

Además, el programador siempre está obligado a tratar las posibles excepciones que se produzcan en tiempo de ejecución. Java define procedimientos para tratar estos errores.

Java también posee mecanismos para garantizar la seguridad durante la ejecución comprobando, antes de ejecutar código, que este no viola ninguna restricción de seguridad del sistema donde se va a ejecutar.

También cuenta con un cargador de clases, de modo que todas las clases cargadas a través de la red tienen su propio espacio de nombres para no interferir con las clases locales.

Otra característica de Java es que está preparado para la programación concurrente sin necesidad de utilizar ningún tipo de biblioteca.

Finalmente, Java posee un gestor de seguridad con el que se puede restringir el acceso a los recursos del sistema.

A menudo se argumenta que Java es un lenguaje lento porque debe interpretar los bytecodes a código nativo antes de poder ejecutar un método, pero gracias a la tecnología JIT, este proceso se lleva a cabo una única vez, después el código en código nativo se almacena de tal modo que está disponible para la siguiente vez que se llame.

Para el desarrollo del FRONT-END se ha realizado la comparación de tres lenguajes de programación de código abierto mediante los cuales se puede realizar las tareas requeridas. A saber: PHP C# y Java. Vea tabla 3.3.2

	PHP	C#	Java
Madurez	Maduro	Maduro	Maduro
Permite procesos de larga duración	A partir de 5.3	Si	Si
Facilidad de escritura en IDE	NO	SI	SI
Ejecución con pocos recursos	SI	Limitado	SI
Despliegue (deploy)	Up and Go	Múltiples procesos	Deploy
Peso de la biblioteca básica	Simple y potente en cuestiones básicas	Framework integrado con múltiples funciones, muy pesado	Muy ligero con muchas funciones, se agregan librerías para mayor funcionalidad. Ligero
Potencia del lenguaje	Limitaciones en su aplicación más especializada, carencia de bibliotecas de alto rendimiento.	Muy potente	Muy potente

	PHP	C#	Java
Asistencia del IDE	Muchas limitaciones, instalación de plugins para IDE's no nativos. Inexistencia de IDE's nativos pre-construidos.	IDE Gratuito y con mantenimiento mucha asistencia.	Muchos IDE's con mantenimiento y asistencia, múltiples foros y comunidades
Uso asíncrono	No está pensado como un lenguaje de procesamiento asíncrono, no explota el multi-threading del procesador	Escritura tediosa de funciones	Existencia de librerías para uso asíncrono
Paquetes adicionales disponibles	pear, pecl	nuget	Maven
Posibilidades de optimización a nivel de ejecución	Bajas	Altas con el uso de templates con varianza, tipos lightweight, acceso a punteros con código unsafe, rendimiento cercano a C/C++	Optimización basada en la máquina virtual de cada sistema.
Plataforma de ejecución	Windows, Linux, Solaris	Sólo Windows	Múltiples

Tabla Comparativa 3.3.2

La interacción usuario-servicio es una parte muy importante del sistema de generación de comprobantes digitales propuesto, para ello es importante contar con una plataforma capaz de soportar esta interfaz y al mismo tiempo ser ligera, multiplataforma y de fácil instalación así como de bajo o nulo costo con el fin de preservar los principios con los que se pensó este sistema. Para la interfaz web se consideran los siguientes tres servidores web los cuales tienen como características principales ser de código abierto, gratuitos, y multiplataforma. Cualquiera de ellos es una buena opción que nos permitirá ejecutar los procesos relacionados con la interfaz del sistema de generación de Comprobantes Fiscales Digitales. Con el fin de identificar algunas características que nos permitan elegir de mejor forma algún servidor en particular se presenta la siguiente tabla de comparación. Vea tabla 3.3.3

Análisis y posicionamiento servidores web para el front-end

A continuación se analizan tres distintas opciones de servidores web disponibles que pueden ejecutar código PHP, sobre este servidor web se soportará la interfaz de usuario para el Sistema para Comprobantes Fiscales Digitales por Internet.

	APACHE	NGINX	LIGHTTPD
Servidor basado en	Procesos	Eventos	Eventos
Ejecuta funciones de Servidor Proxy	No	Si	No
Ejecución de código	estático/dinámico	estático/dinámico	estático/dinámico
Aprovechamiento de poca memoria	Medio	Alto	Alto
Cantidad de componentes	Muchos	Medio	Pocos

	APACHE	NGINX	LIGHTTPD
Trabajo asíncrono	No	Si	Si
SSL/TLS	Si	Si	Si
Virtual Host	Si	Si	Si
Balanceo de cargas	Si	Si	No
Compresión	Si	Si	Si
Control de acceso	Si	Si	Si
Reescritura de URL	Si, desde .htaccess	Si	No (sólo vía config)
Personalización de log	Si	Si	No
Transmisión de FLV	Si	Si	Si
Ejecución de FastCGI	Si	Si	Si
Estable	Muy estable	Estable	Estable
Popularidad	Muy popular	poca popularidad	poca popularidad
Licenciamiento	Apache Licence 2.0	Simple BSD	Revised BSD
Librería X-Sendfile	3rd party module		X-Sendfile
Uso de memoria RAM	Muy alta	Baja	Baja

Tabla Comparativa 3.3.3

Elección del lenguaje de programación y del manejador de base de datos.

Con base en la tabla anterior y la experiencia con los diferentes manejadores de bases de datos y con los lenguajes de programación evaluados se ha decidido

realizar el desarrollo de la implementación en el lenguaje de programación Java para el back-end y el manejador de bases de datos MYSQL así como el lenguaje de programación PHP para el front-end; esto permitirá un desarrollo ágil, rápida integración con el Framework Hibernate y la implementación de patrones de diseño que facilitan el desarrollo, escritura y mantenimiento del código fuente. Adicionalmente se tiene la posibilidad de ejecutar este código en cualquier plataforma y con base en el desarrollo, mantenimiento y planes a futuro para estas plataformas se tiene amplio margen de seguridad además cada responsable de los distintos lenguajes y sistemas seleccionados mantendrán actualizados estos ampliando el margen de seguridad y vigencia.

3.4 APLICACIÓN Y SEPARACIÓN DE LOS MÓDULOS SEGÚN LA METODOLOGÍA DE YOURDON/DE MARCO

El método de análisis y diseño estructurado de Yourdon contempla todas las fases del ciclo de vida de un sistema de información, desde la fase de estudio y concepción hasta la implementación y puesta en marcha.

Como primer paso, se hace un estudio de factibilidad que determina que los objetivos del sistema sean realistas. El estudio debe presentar en forma detallada los objetivos y alcance del sistema.

El siguiente paso es el análisis que consiste en identificar las necesidades del cliente, evaluar la viabilidad del sistema, realizar un análisis técnico y económico, asignar funciones al software, al hardware, a la gente, a la base de datos y a otros elementos del sistema, estableciendo restricciones de costo y tiempo, creando una definición del sistema que sea la base para todo el trabajo posterior de Ingeniería.

En los primeros apartados de este capítulo se inició justo con el análisis del problema, y se plasmaron los requerimientos generales y particulares de la aplicación, en este primer nivel se ha generado información importante que servirá

como base para llegar a un detalle más fino de definición y poder pasar posteriormente a la parte del diseño.

Así mismo se ha completado el análisis y posicionamiento de los lenguajes para el back-end y front-end, a estas alturas ya hemos evaluado las posibles soluciones y evaluado la opción más viable en cuanto a plataforma, lenguaje de programación, manejador de base de datos y framework para el front, resaltando las ventajas de usar herramientas de código abierto para disminuir costos y no depender de terceros con respecto a las licencias.

Hasta esta parte se tiene identificado que será necesario contar con un front a través del cual se generará un archivo .TXT que será la entrada para nuestro proceso de facturación electrónica, el cual se puede esquematizar en la figura 3.4.1.

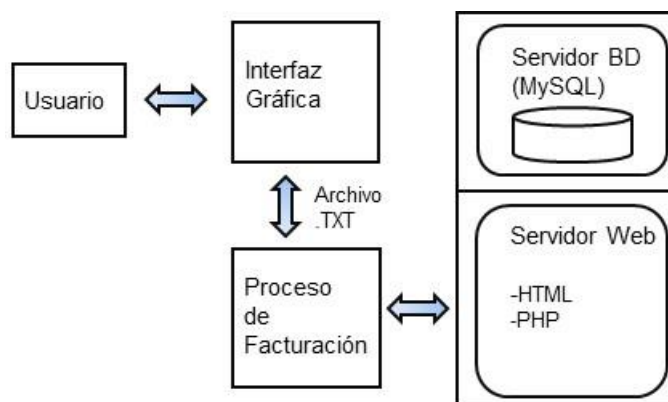


Figura 3.4.1 Esquema del Sistema de Facturación

El propósito del análisis es transformar las entradas principales, a las políticas del usuario y el esquema del proyecto, en una especificación estructurada.

Y el siguiente paso es el diseño, la fase de diseño está orientada a la planificación y organización de los datos, así como la definición de acceso a esta información. En el diseño también se contemplan las interfaces de entrada y salida del sistema que cumplan con las especificaciones del usuario.

La metodología de Yourdon se basa en la generación de diferentes modelos, y cada modelo intenta generar una vista específica del sistema, y a su vez cada modelo se apoya con el uso de diferentes herramientas para esquematizar lo que hará el sistema y cómo lo hará.

Entraremos de lleno al uso de la metodología de Yourdon/De Marco en el capítulo IV, por lo pronto trabajaremos en identificar los módulos que deberá incluir nuestra aplicación, y plantearemos la separación de los mismos a través de un diagrama de bloques, que refleje los requerimientos que se plantearon en los apartados anteriores en este mismo capítulo.

El sistema está enfocado a empresas que se dedican a la compra/venta de algún producto mercantil, por lo que se requiere el front a través del cual se irá registrando cada compra, y los diferentes módulos que intervienen en todo el proceso de emisión de Certificado Fiscal Digital por Internet, la propuesta preliminar de la separación de módulos de nuestro sistema, se muestra en la figura 3.4.2.

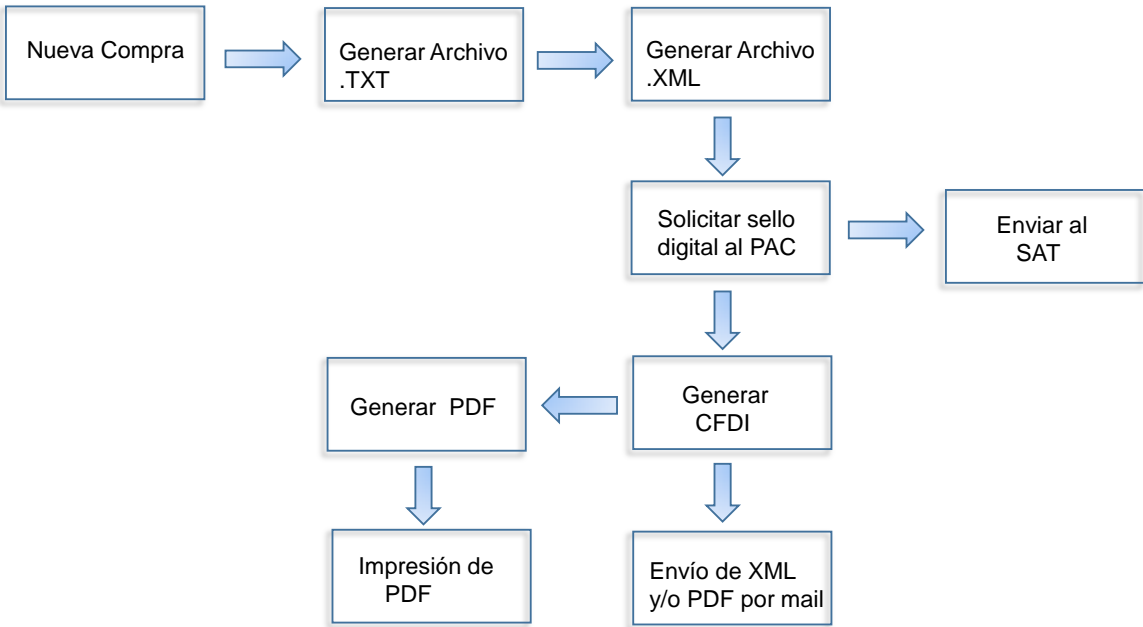


Figura 3.4.2 Diagrama de Bloques del Sistema

El módulo de generación de archivo XML deberá considerar todos los aspectos técnicos del CFDI declarados por el SAT, los cuales se revisaron a detalle en el capítulo II, se debe considerar que el módulo destinado a interactuar con el PAC para solicitar el sello digital debe cumplir con los estándares establecidos para ello y considerar un esquema seguro y robusto.

En términos generales el módulo para generar CFDIs debe considerar el proceso de almacenamiento, consulta y descarga de dichos comprobantes fiscales.

Así mismo, debe considerarse la opción de generar la representación impresa de los CFDIs (formato PDF), y la entrega tanto de este archivo PDF o del CFDI en formato XML por mail o la entrega por algún medio electrónico como USB o CD.

3.5 DISEÑO A NIVEL DE BLOQUES DE LA INTERFAZ DE USUARIO

La entrada de datos al sistema de comprobantes fiscales digitales será a través de un entorno web visible al usuario, el cual será diseñado para un fácil manejo e interacción. Desde aquí este usuario podrá realizar funciones como son: agregar productos, añadir los datos del receptor y añadir los datos de la factura,

Así pues, la interfaz deberá contener las entradas útiles y funcionalidad pertinente que serán importantes para nuestro sistema.

A continuación se muestra el diseño a nivel de bloques de la interfaz del usuario, vea figura 3.5.1:

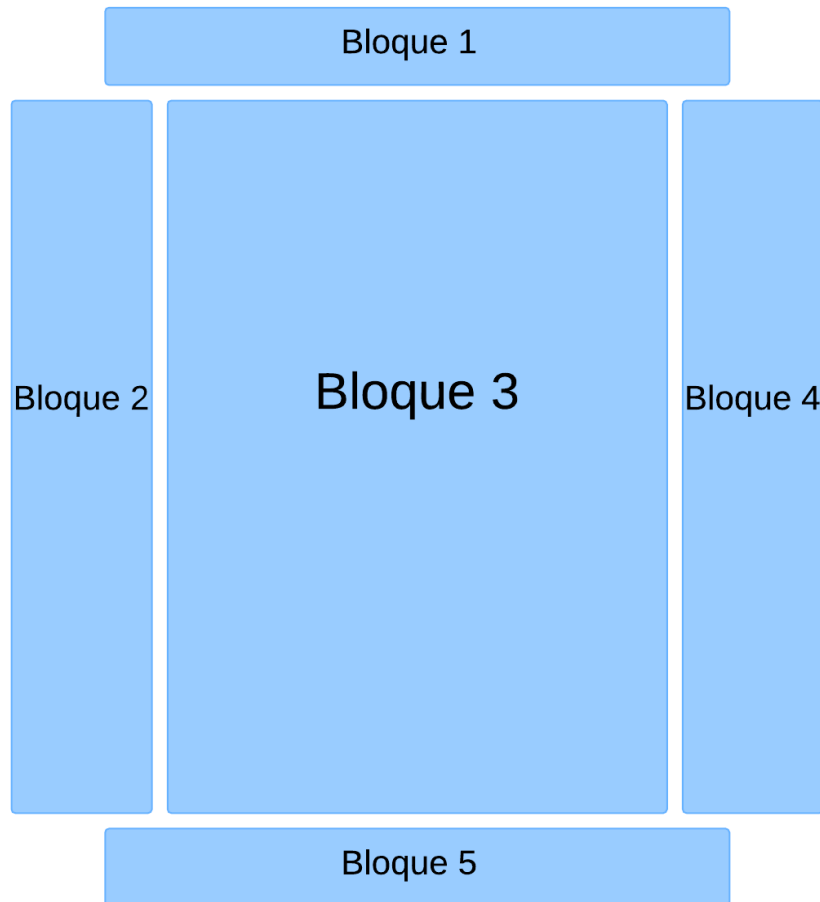


Figura 3.5.1 Diseño a bloques de interfaz de usuario

Descripción de los bloques

A continuación se describe el diseño a nivel de bloques de la interfaz de usuario:

En la parte superior se encuentra el primer bloque. El bloque 1, será el encabezado de la página, en él se colocará el logo de la empresa.

El segundo bloque es una barra lateral que se encontrará en la parte izquierda de la página. En el bloque 2 se colocarán diversos botones con acciones acerca del manejo de la factura.

El tercer bloque que se encuentra al centro de la página, contendrá el área de trabajo principal del usuario, aquí se alojará el formulario con las opciones básicas

de la factura, receptor y de los productos para que sean completados por el usuario.

El bloque 4 será una barra lateral que se encontrará en la parte derecha de la página, esta será el área de notificaciones. Aquí se mostrará información relevante acerca del llenado de los campos y mensajes emergentes.

Por último, el bloque 5 que se encuentra en la parte inferior de la página, será cubierta por los datos relacionados a la empresa, como puede ser dirección, teléfono o contacto.

CAPÍTULO

IV

ANÁLISIS Y CONSTRUCCIÓN DE LA APLICACIÓN

4.1 MODELADO ESENCIAL DE GENERACIÓN DEL SISTEMA DE GENERACIÓN DE COMPROBANTES FISCALES DIGITALES POR INTERNET (SCFDI)

El modelo esencial ilustra qué debe hacer el sistema, excluyendo los detalles de cómo debe ser implementado, establece los requerimientos de las políticas del negocio, permite verificar que estos requerimientos sean consistentes y completos. El modelo esencial está conformado a su vez por dos modelos: modelo esencial ambiental y modelo esencial del comportamiento.

MODELO ESENCIAL AMBIENTAL

El modelo esencial ambiental representa la forma en que el sistema interactúa con su entorno o ambiente.

Para generar el modelo esencial ambiental de nuestro sistema, nos basaremos en el análisis preliminar y estudio de factibilidad que se llevó a cabo en los capítulos anteriores de nuestro trabajo de tesis, y nos basaremos en los modelos y herramientas propuestos por el método estructurado de Yourdon, el modelo esencial ambiental forma parte de la fase del análisis, y en este modelo se utilizarán las herramientas: declaración de propósito, diagrama de contexto, lista de eventos y versión preliminar del diccionario de datos. Vea tabla 4.1.1

	MODELO	HERRAMIENTA
ANÁLISIS	Modelo Esencial Ambiental	-Declaración de propósito -Diagrama de contexto -Lista de eventos -Diccionario de datos
	Modelo Esencial del Comportamiento	-Diagrama de flujo de datos -Diagrama Entidad-Relación -Especificación de procesos
	Modelo de Implementación del Usuario	-Diagrama de flujo de datos -Diagrama de transición de estados -Diccionario de datos -Formato de pantallas
DISEÑO	Modelo de Implementación del Sistema	-Diagrama de flujo de datos -Diagrama de transición de estados -Diccionario de datos
	Modelo de Implementación del Programa	-Diagrama de estructuras -Diagrama de flujo de datos -Diccionario de datos -Especificación de módulos

Tabla 4.1.1 Modelos y herramientas del Método Estructurado de Yourdon

El primer paso para generar el modelo esencial ambiental es escribir la Declaración de propósito, a través del cual se describe el alcance del sistema en una forma clara y concisa. Vea tabla 4.1.2

Sistema de Generación de Comprobantes Fiscales Digitales por Internet (SCFDI).

Declaración de propósito:

El sistema permitirá generar, imprimir y almacenar comprobantes fiscales digitales por Internet, la emisión de dichos CFDIs se apegará a los requisitos impuestos por el Sistema de Administración Tributaria.

Responsabilidades:

El principal objetivo del sistema es generar los CFDIs en forma automática, a través de su entrada principal que será un archivo en formato TXT.

El sistema permitirá capturar los datos requeridos para emitir un Comprobante Fiscal Digital por internet cuando el cliente realice la compra de algún producto.

El sistema SCFDI permitirá generar un archivo XML que cumpla con los requisitos establecidos

en el anexo 20 de la Resolución Miscelánea Fiscal 2014 relacionado con los Comprobantes Fiscales Digitales por Internet.

El sistema enviará la información de Factura Electrónica en formato XML a un Proveedor Autorizado de Certificación (PAC).

El sistema realizará la recepción del documento timbrado (y por tanto convertido en un Comprobante Fiscal Digital por Internet) desde el servicio del PAC.

El sistema podrá generar la representación del CFDI como un archivo en formato PDF y estará disponible para imprimirse o para enviar por correo electrónico.

El sistema permitirá enviar por correo electrónico tanto el archivo .XML como el archivo PDF.

El sistema permitirá almacenar los CFDIs emitidos para cada cliente, al realizar éste la compra de algún producto.

Exclusiones:

El sistema no generará comprobantes fiscales del rubro de pago de Nóminas

El sistema no generará comprobantes fiscales de operaciones de enajenación de inmuebles

El sistema no generará comprobantes fiscales de arrendamiento

El sistema no generará comprobantes fiscales de notas de crédito

Tabla 4.1.2 Declaración de Propósito

4.1.1 DIAGRAMA DE CONTEXTO DEL SISTEMA SCFDI

El siguiente paso del modelo esencial ambiental después de haber generado la declaración del propósito es generar el diagrama de contexto que es la representación gráfica de la forma en que el sistema se relacionará con su entorno o ambiente, muestra las entidades externas con las que se tendrá comunicación representados por rectángulos (TERMINADORES), muestra el proceso principal que recibe el nombre de nuestro sistema, representado por un círculo (PROCESO), y los flujos de información que entrarán al sistema y sus salidas, los

flujos se representan con flechas.

Así mismo, si el sistema recibe algún catálogo o archivo, o consulta información de alguna tabla de algún sistema externo, esta información se representa con dos rallas paralelas (ALMACEN).

En la figura 4.1.1.1, se muestra el diagrama de contexto del SCFDI.

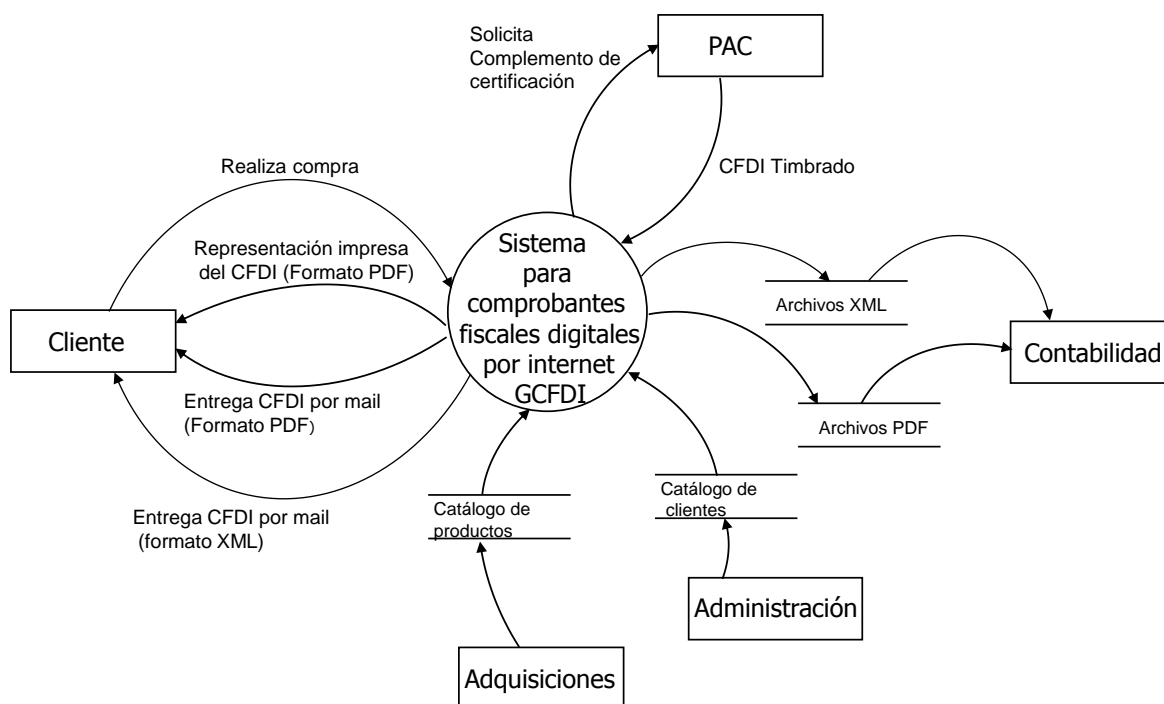


Figura 4.1.1.1 Diagrama de Contexto

Todos los elementos del diagrama de contexto deben documentarse en el diccionario de datos.

El diccionario de datos es un listado organizado de todos los datos pertenecientes al sistema, con definiciones precisas y rigurosas para que tanto el usuario como el analista tengan un entendimiento común de todas las entradas, salidas, componentes de almacenamiento y cálculos intermedios.

4.1.2 NOTACIÓN DEL DICCIONARIO DE DATOS

La notación del diccionario de datos se ha hecho necesaria porque en la mayoría de los sistemas, existen paquetes o elementos de datos lo suficientemente complejos que necesitan ser descritos en términos de elementos más sencillos, a su vez, los elementos sencillos se describen en términos de valores y unidades que pueden asumir.

Se requiere una notación concisa y breve, así como un diccionario normal tiene una definición compacta para definir el significado de las palabras ordinarias. Ver en la tabla 4.1.2.1 los símbolos que se utilizan en el Diccionario de Datos.

Símbolo	Significado
=	está compuesto de
+	y
()	optativo (puede estar presente o ausente)
{ }	iteración
[]	selección
* *	comentario
@	identificador (campo clave) para un almacén
	separa opciones alternativa en la selección

Tabla 4.1.2.1 Notación del Diccionario de Datos

Ver en la tabla 4.1.2.2 el Diccionario de Datos Preliminar del Sistema SCFDI.

Diccionario de Datos	
Especificación de Terminadores	
TERMINADOR	Cliente
DEFINICIÓN	Es una persona física o moral que compra uno o varios productos.

TERMINADOR	PAC
DEFINICIÓN	Proveedor autorizado de certificación, es un proveedor autorizado por el SAT que ofrece servicios para generar Complementos de Certificación.
TERMINADOR	Contabilidad
DEFINICIÓN	Es un área dentro de la empresa que se encarga de llevar la contabilidad de la misma, y requiere consultar los CFDIs de todas las compras que realizan los clientes.
TERMINADOR	Administración
DEFINICIÓN	Es el área de la empresa que se encarga de realizar varias actividades administrativas propias de la empresa.
TERMINADOR	Adquisiciones
DEFINICIÓN	Es el área dentro de la empresa que se encarga de realizar la adquisición de productos a través de los diferentes proveedores.
Especificación de Flujos de Datos	
FLUJO DE DATOS	Realiza compra
SIGNIFICADO	Cuando un cliente realiza la compra de uno o varios productos, el operador necesita capturar sus datos para que el sistema pueda emitir el CFDI (Certificado Fiscal Digital por Internet). Por lo que este flujo es la entrada de varios datos que necesita el sistema, tales como los datos del Receptor o Cliente y datos de los productos que está comprando, así mismo, el cliente debe confirmar si requiere que se le envíe su Factura Electrónica por email en formato XML, en formato PDF, o una representación impresa.
ESTRUCTURA	Grupo
COMPOSICIÓN	-Realiza compra, proporciona sus datos para su factura Se realiza una petición de compra y el cliente debe proporcionar los siguientes datos: Descripcion + Cantidad + rfc_receptor + rfce_receptor + nombre_receptor + calle + exterior + interior + colonia + localidad + referencia + municipio + estado + país + codigoPostal + ciudad + teléfono + fax + emailEnvio + emailContacto -Respuesta para el cliente Se da una respuesta al cliente si su compra es aprobada o declinada.
FLUJO DE DATOS	Entrega CFDI por mail (formato XML)
SIGNIFICADO	Cuando un cliente realiza la compra de uno o varios productos, se le enviará su CFDI por mail en el formato estándar XML- Extensible Markup Language (Lenguaje de Marcas Extensible).

FLUJO DE DATOS		Entrega CFDI por mail (formato PDF)
SIGNIFICADO	Cuando un cliente realiza la compra de uno o varios productos, se le puede enviar su CFDI por mail en formato PDF – Portable Document Format (formato de documento portátil). Esta interfaz se define en el Modelo de Implementación del Usuario.	
FLUJO DE DATOS		Representación impresa del CFDI (formato PDF)
SIGNIFICADO	Cuando un cliente realiza la compra de uno o varios productos, se le puede entregar una representación impresa del CFDI en formato PDF. Esta interfaz se define en el Modelo de Implementación del Usuario.	
FLUJO DE DATOS		Solicita Complemento de Certificación
SIGNIFICADO	Es la petición que la empresa envía al PAC para que genere un CFDI válido ante el SAT. El sistema SCFDI le envía un arreglo de bytes al PAC que contiene el XML para que a su vez el PAC pueda timbrar la Factura Electrónica al integrar el Complemento de Certificación que contiene Sello del SAT, UUID (Folio Fiscal), Fecha y hora de timbrado, y número de Certificado del SAT.	
ESTRUCTURA	Elemento	
VALORES	Cadena en formato estándar XML	
FLUJO DE DATOS		CFDI Timbrado
SIGNIFICADO	Es la respuesta que regresa el PAC al sistema SCFDI una vez que timbró el CFDI. El PAC se encarga de notificar al SAT cada vez que realiza el timbrado de un CFDI, para que a su vez el SAT le genere los datos de control y sea una Factura legítima.	
ESTRUCTURA	Elemento	
VALORES	Cadena en formato estándar XML incluye timbre fiscal.	
Especificación de Almacenes		
ALMACEN	Catálogo de Productos	
DEFINICIÓN	Es un catálogo externo al Sistema SCFDI, que contiene la lista de los Productos que la empresa vende, este catálogo lo deja disponible el área de Adquisiciones de la Empresa, para que el Sistema SCFDI pueda tener acceso a esta información. =@id_concepto + Descripción + Cantidad + Unidad	
ALMACEN	Catálogo de Clientes	
DEFINICIÓN	Es un catálogo externo al Sistema GCFI, que contiene los datos principales de un Cliente, este catálogo lo deja disponible el sistema de	

	Administración de la Empresa, para que el Sistema SCFDI pueda tener acceso a esta información. =@id_cliente + rfc+ rfce + nombre + cuenta + calle_num+ colonia + codigoPostal + estado + localidad + país
ALMACEN	Archivos XML
DEFINICIÓN	Contiene CFDIs en formato XML, son archivos independientes para que el área de Contabilidad pueda tener acceso todos los certificados generadas por el Sistema SCFDI.
ALMACEN	Archivos PDF
DEFINICIÓN	Contiene CFDIs es formato PDF, son archivos independientes para que el área de Contabilidad pueda tener acceso a todas los certificados por el Sistema SCFDI.

Tabla 4.1.2.2 Diccionario de datos preliminar

Para la definición de los flujos de datos en el diccionario de datos, se debe especificar su significado, su composición, y su estructura.

Los tipos de estructura son:

- **Múltiple:** se compone de flujos que existen en instantes diferentes en el tiempo. Estos flujos se utilizan para simplificar los diagramas de nivel más alto.
- **Grupo:** el flujo se compone de otros flujos que existen al mismo tiempo. Estos flujos pueden contener otros flujos.
- **Elemento:** el flujo es una pieza única de datos. Estos flujos son, o bien una propiedad de una entidad, conocida como un atributo, o un mensaje.
- **Par de diálogo:** el flujo tiene dos nombre; el primero un iniciador, y el segundo un flujo de respuesta. Ambos deben ser grupo o elemento con sus propias especificaciones de flujo de datos.

4.1.3 LISTA DE EVENTOS DEL SISTEMA SCFDI

La lista de eventos describe todos los eventos que hacen que el sistema realice una actividad y genere una salida deseada por el usuario, ya sea que regrese información en línea, que genere un archivo, que inicie un proceso, etc.

En la tabla 4.1.3.1 se muestra la Lista de eventos del Sistema SCFDI.

Lista de Eventos	
1.-	Cliente hace una compra de uno o varios productos
2.-	Cliente solicita CFDI de su compra
3.-	Cliente solicita que se le envíe CFDI vía correo electrónico (email)
4.-	Cliente solicita Representación Impresa del CFDI (formato PDF)

Tabla 4.1.3.1 Lista de Eventos

MODELADO ESENCIAL DEL COMPORTAMIENTO

Una vez que se ha concluido el modelo esencial ambiental, se puede iniciar con el modelo esencial del comportamiento, este modelo también forma parte del análisis del sistema, y describe el comportamiento que el sistema requiere para que interactúe de manera exitosa con el ambiente.

En el modelo esencial del comportamiento se utilizan las herramientas: diagrama de flujo de datos (DFD), diagrama de entidad-relación (DER), especificación de

procesos y se va completando el diccionario de datos.

4.1.4 DIAGRAMA DE ENTIDAD-RELACIÓN DEL SISTEMA SCFDI

Un diagrama de entidad-relación (DER) resalta las categorías de información importantes y las relaciones entre estas categorías. Los componentes del diagrama de entidad-relación son:

- Entidades, que representan una colección o conjunto de objetos del mundo real cuyos miembros juegan algún papel en el desarrollo del sistema; deben ser identificados de manera única y ser descritos por uno o más atributos, son representadas por rectángulos.
- Relaciones, que son asociaciones entre entidades, representadas por rombos.
- Supertipo, una entidad clasificada.
- Subtipo, un grupo de ocurrencias de entidades.
- Entidad asociativa, una relación que también actúa como entidad.

Ver en la figura 4.1.4.1 el DER de nuestro sistema, en el cual se duplicó la entidad “Factura” para que fuera más legible el DER, y se representa con un asterisco “*Factura”.

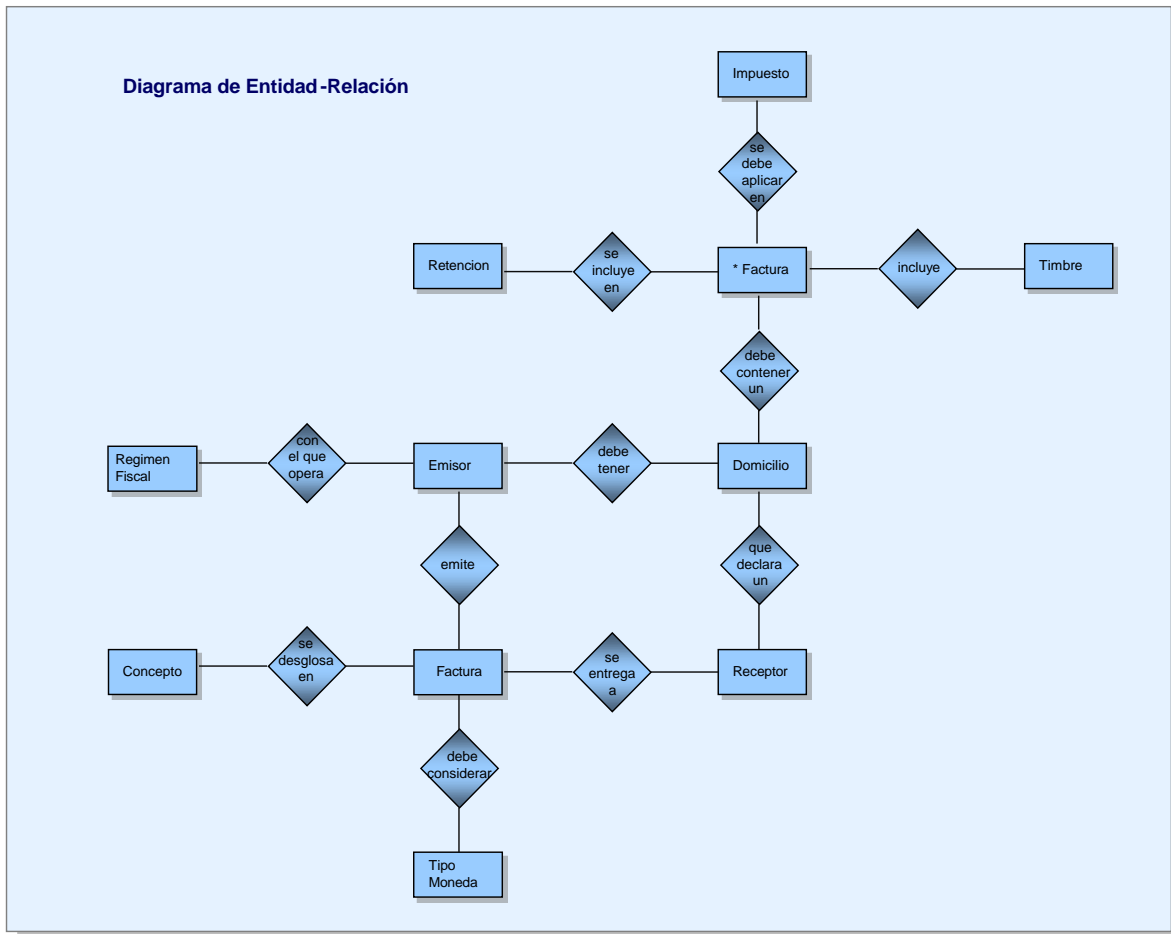


Figura 4.1.4.1 Diagrama de Entidad-Relación del Sistema SCFDI

Con la elaboración del DER, se puede ir complementando el diccionario de datos, ver en la tabla 4.1.5.1 el diccionario de datos con la definición de todas las relaciones del DER.

Es importante comentar que las entidades del DER de nuestro sistema están definidas en la sección 4.2 Creación de la base de datos e implementación del sistema SCFDI de este capítulo.

4.1.5 DICCIONARIO DE DATOS DEL SISTEMA SCFDI

El Diccionario de Datos del Sistema SCFDI se inició a generar en la sección 4.1.2, en donde se definieron todos los elementos del diagrama de contexto, el siguiente

paso es ir complementando el diccionario de datos ahora con la definición de las relaciones del DER, en la tabla 4.1.5.1 se describen los elementos del diagrama de entidad-relación.

Diccionario de Datos	
Especificación de Relaciones	
Relación	Se entrega a
DEFINICIÓN	Es una asociación entre <Receptor> y <Factura>. Una <Factura> se entrega a un <Receptor>, al realizar la compra de uno o varios productos, el Cliente recibirá su factura, por lo que un cliente puede tener una o varias facturas, pero, una factura sólo se entrega a un solo cliente.
Relación	Con el que opera
DEFINICION	Es una asociación entre <Régimen Fiscal> y <Emisor>, un <Emisor> se apeg a un <Régimen Fiscal> para poder realizar la emisión de facturas para cualquier compra que realice cualquier cliente.
Relación	Debe tener
DEFINICION	Es una asociación entre <Emisor> y <Domicilio>, Un <Emisor> emite una factura con los datos del <Domicilio> de un cliente.
Relación	Se desglosa en
DEFINICION	Es la asociación entre <Concepto> y <Factura>, donde una <Factura> contiene la información de cualquier concepto, este pueden ser uno o varios, así como un concepto puede tener una <Factura>.
Relación	Emite
DEFINICION	Es la asociación entre <Emisor> y <Factura>, una <Factura> se entrega a un cliente a través de la emisión del <Emisor>, este puede emitir una o muchas facturas y una <Factura> solo puede emitirse una vez.
Relación	Debe de contener un
DEFINICION	Es la asociación entre <Factura> y <Domicilio>, una <Factura> debe de contener los datos de domicilio y a su vez el <Domicilio> puede estar en muchas facturas.
Relación	Se debe de aplicar en
DEFINICION	Es la asociación entre <Factura> e <Impuesto>, Un <impuesto> se asigna o se agrega a una factura, para su emisión o entrega a un cliente,

	dependiendo del producto o servicio, y una <Factura> puede o no contener un <Impuesto>
Relación	Debe de considerar
DEFINICION	Es una asociación entre <Factura> y <Tipo Moneda>, un <Tipo de Moneda> debe de estar registrado en una o varias <Factura>, y una <Factura> debe de tener solo un <Tipo Moneda>.
Relación	Se incluye en
DEFINICION	Es una asociación entre <Factura> y <Retencion>, una <Retencion> es un impuesto de egreso que se aplica o se ingresa en una o varias <Factura>.
Relación	Incluye
DEFINICION	Es una asociación entre <Factura> y <Timbre>, cuando <Factura> ha sido recibida, verificada y validada entonces se le agrega los datos fiscales, estos datos se incluyen en <Timbre>, una <Factura> debe incluir solo un <Timbre> y un <Timbre> debe incluir sólo una <Factura>

Tabla 4.1.5.1 Diccionario de Datos del Sistema SCFDI incluyendo elementos del DER

4.1.6 DIAGRAMA DE FLUJO DE DATOS DE RESPUESTA A EVENTOS DEL SISTEMA SCFDI

El diagrama de flujo de datos (DFD) ilustra los datos que viajan a través del sistema y las transformaciones que éstos sufren, mostrando el control, la secuencia y algoritmos procedurales que existen en el sistema.

Existen dos tipos de diagramas de flujo de datos que representan vistas alternas del aspecto del comportamiento:

- **Diagrama de flujo de datos de respuesta a eventos**, que representa cada proceso que responde a un evento
- **Diagrama de flujo de datos de alto nivel**, que presentan al sistema desde un nivel general hasta un nivel detallado.

Ver en la figura 4.1.6.1 el DFD de respuesta a eventos del Sistema SCFDI.

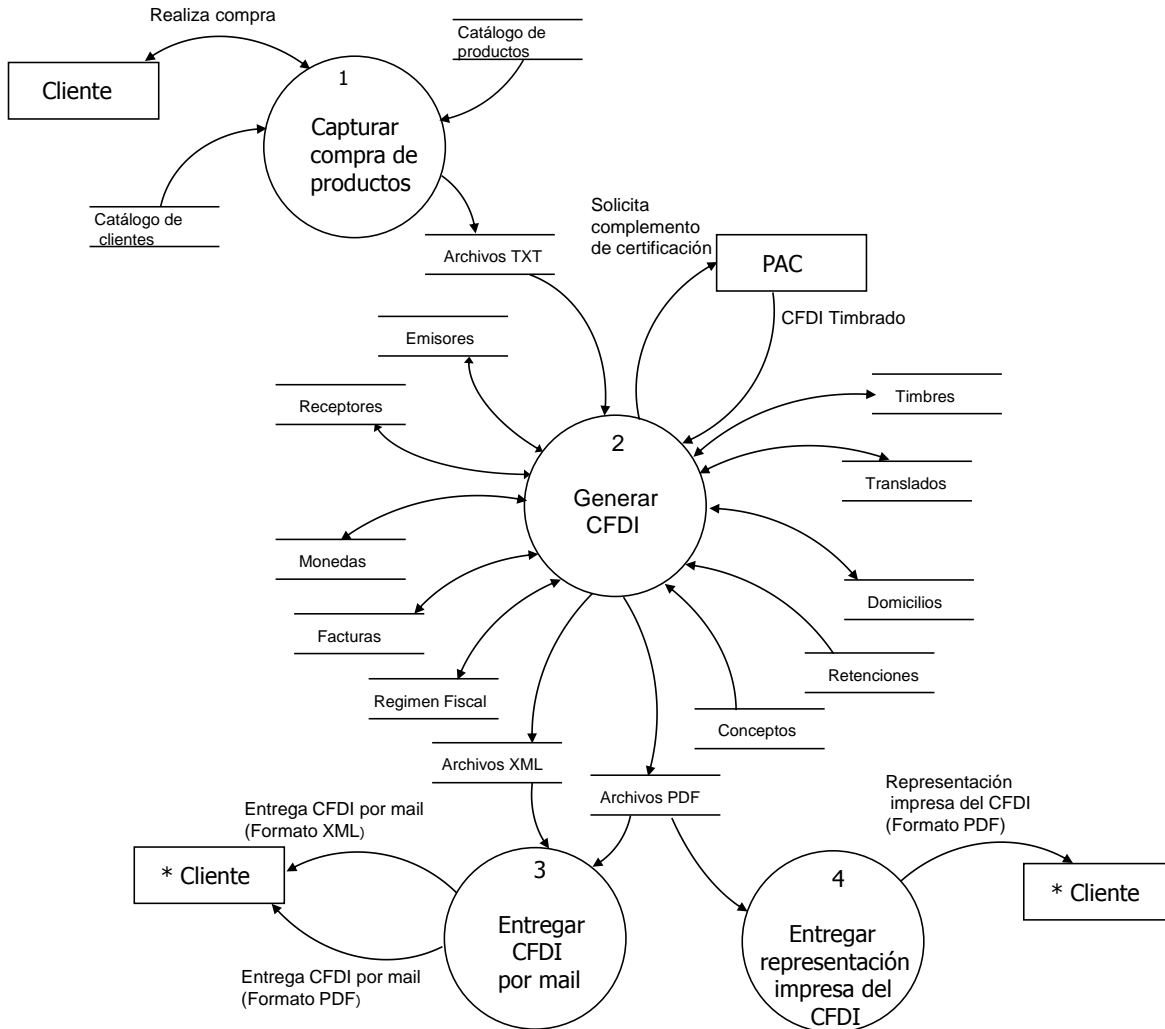


Figura 4.1.6.1 Diagrama de flujo de datos de respuesta a eventos del Sistema SCFDI

Se pueden duplicar los elementos del DFD para que los diagramas queden más claros y no se saturan de flujos entre componentes, pero se deben marcar con un asterisco para indicar que se han duplicado, como se muestra en la figura 4.1.6.1 para el terminador "Cliente".

Determinando procesos de alto nivel

Para determinar los procesos de alto nivel debemos agrupar las respuestas a eventos relacionados. Los procesos se relacionan si se cumple alguna de las siguientes condiciones:

- Si cumplen una función del negocio
- Si todos son revisados por un solo usuario
- Si todos requieren las mismas entidades y relaciones
- Si responden a un grupo de eventos relacionados

Ver en la figura 4.1.6.2 el DFD de primer nivel

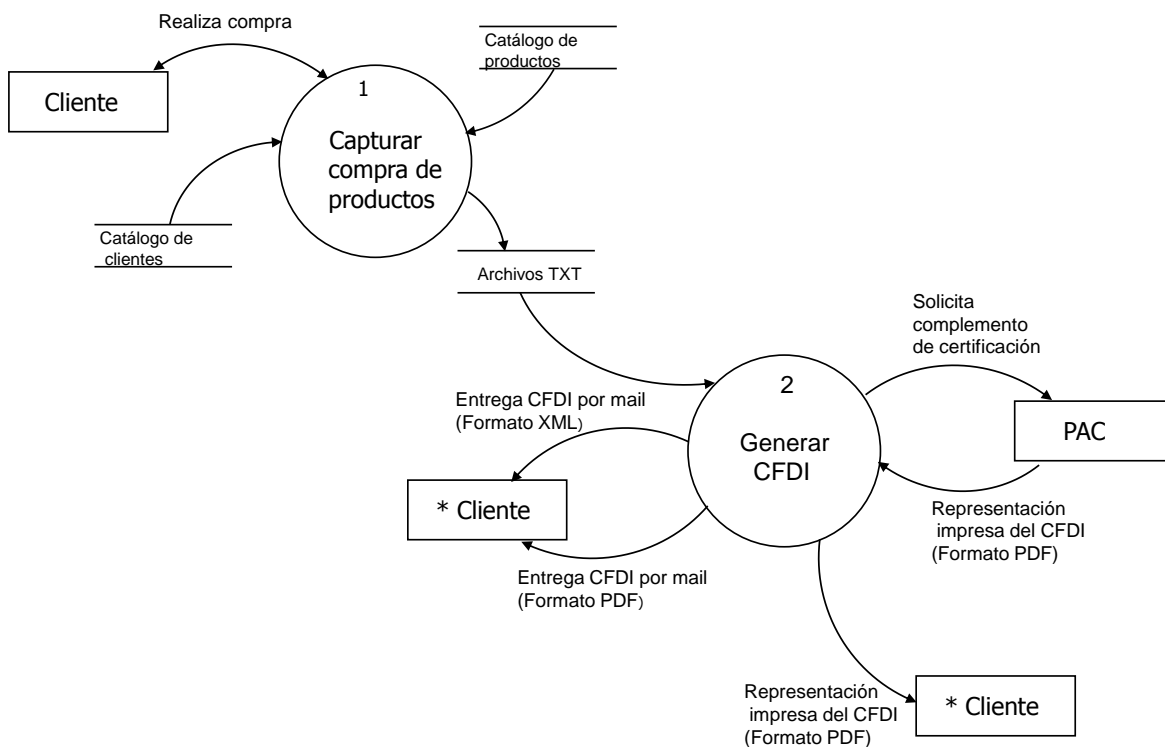


Figura 4.1.6.2 Diagrama de flujo de Datos por niveles (Primer Nivel)

A partir del diagrama de flujo de datos de primer nivel, se puede ir tomando cada proceso para obtener el diagrama de flujo de datos del segundo nivel.

El proceso 1, es a través del cual se recibe la información con la que se podrá generar el CFDI, y en la figura 4.1.6.3 se muestra el DFD de segundo nivel de dicho proceso.

En los diagrama de nivel superior ya no es necesario mostrar los terminadores, por ejemplo en la figura 4.1.6.3 ya no aparece el terminador “Cliente”.

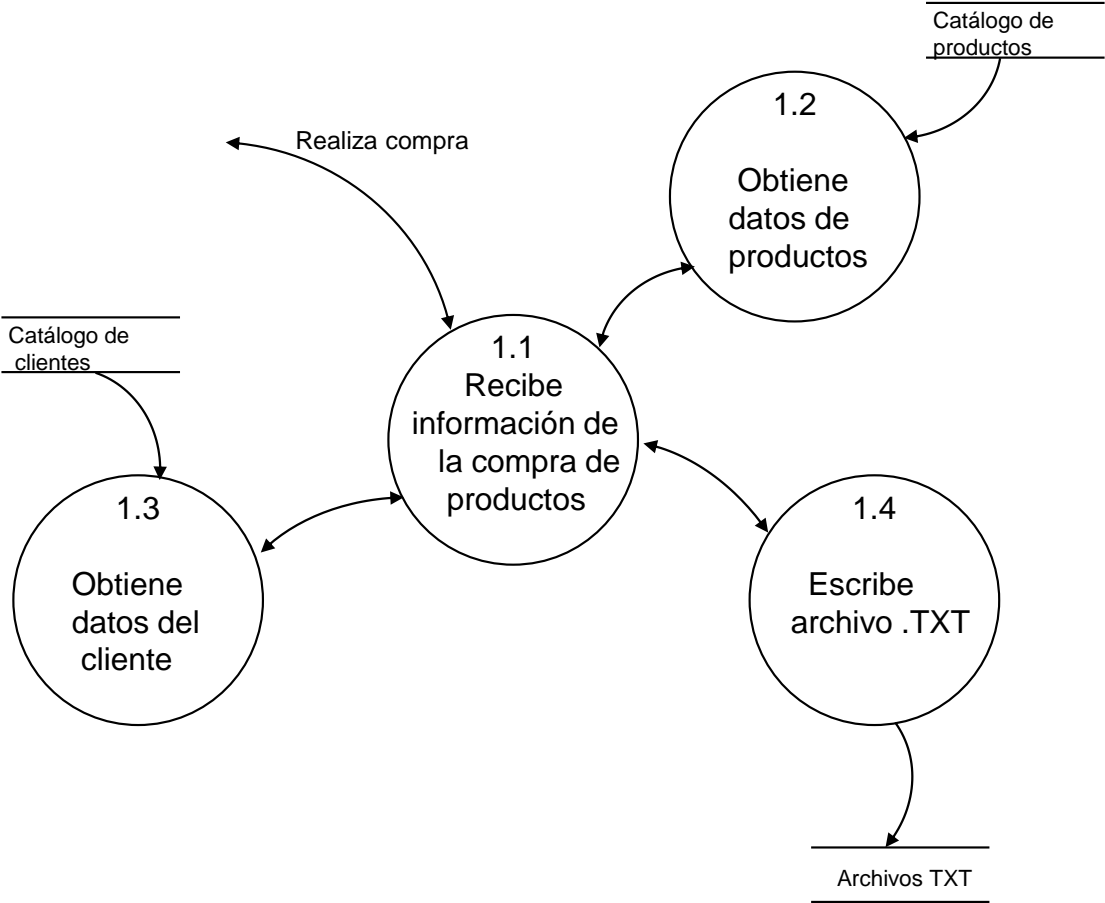


Figura 4.1.6.3 Definición del Proceso 1 Capturar compra de productos

Para este proceso no hace falta bajar a otro nivel, ya que los procesos de los que se compone ya son procesos dedicados a tareas muy específicas, la documentación de estos procesos se verá en la siguiente sección una vez que se concluya todos los diagramas de flujo.

Ahora detallaremos el proceso 2, y elaboraremos el DFD de segundo nivel, como

se muestra en la figura 4.1.6.4.

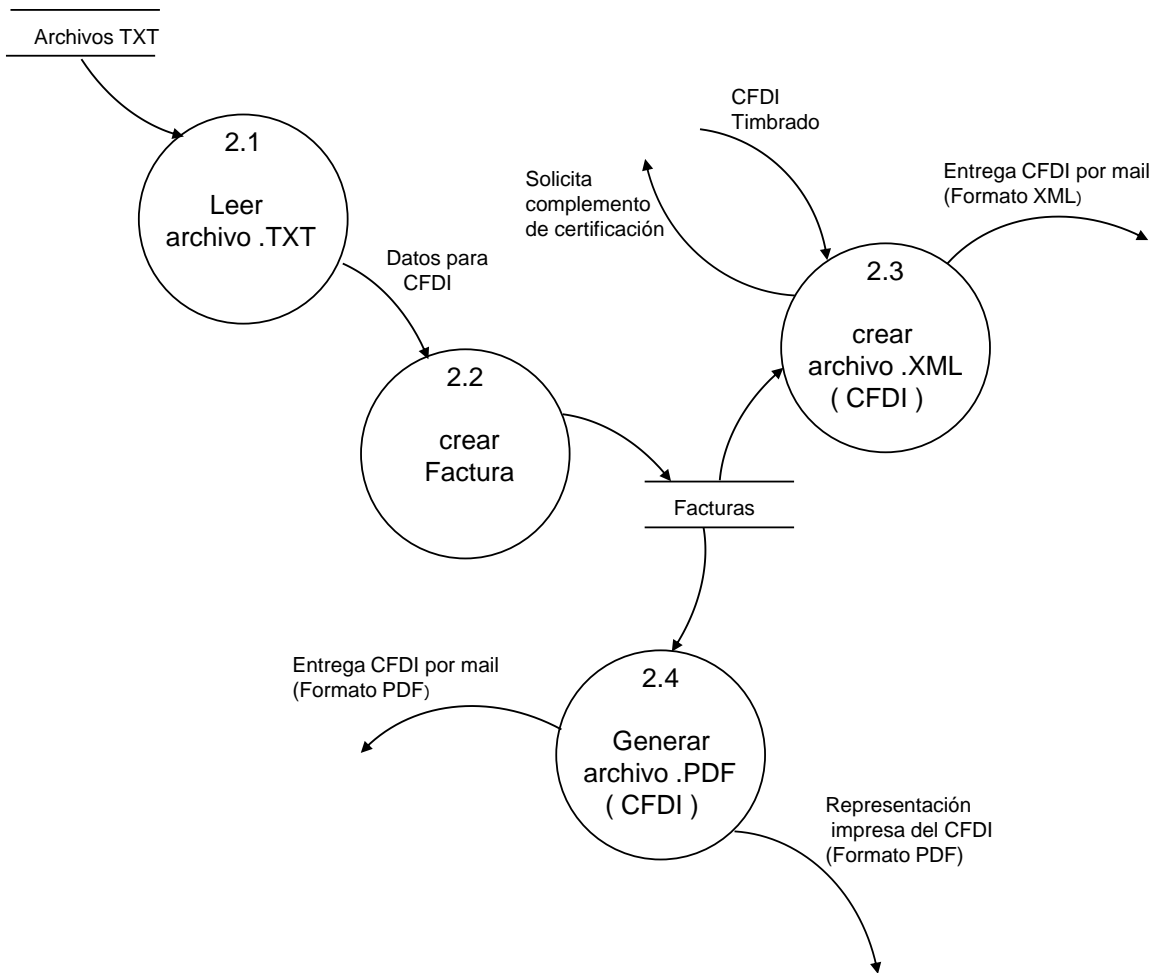


Figura 4.1.6.4 Proceso 2 Generar Factura Electrónica (CFDI)

El siguiente paso es tomar cada proceso del DFD de segundo nivel, y generar los diagramas de flujo de datos ahora a nivel tres, tal como se muestra en la figura 4.1.6.5.

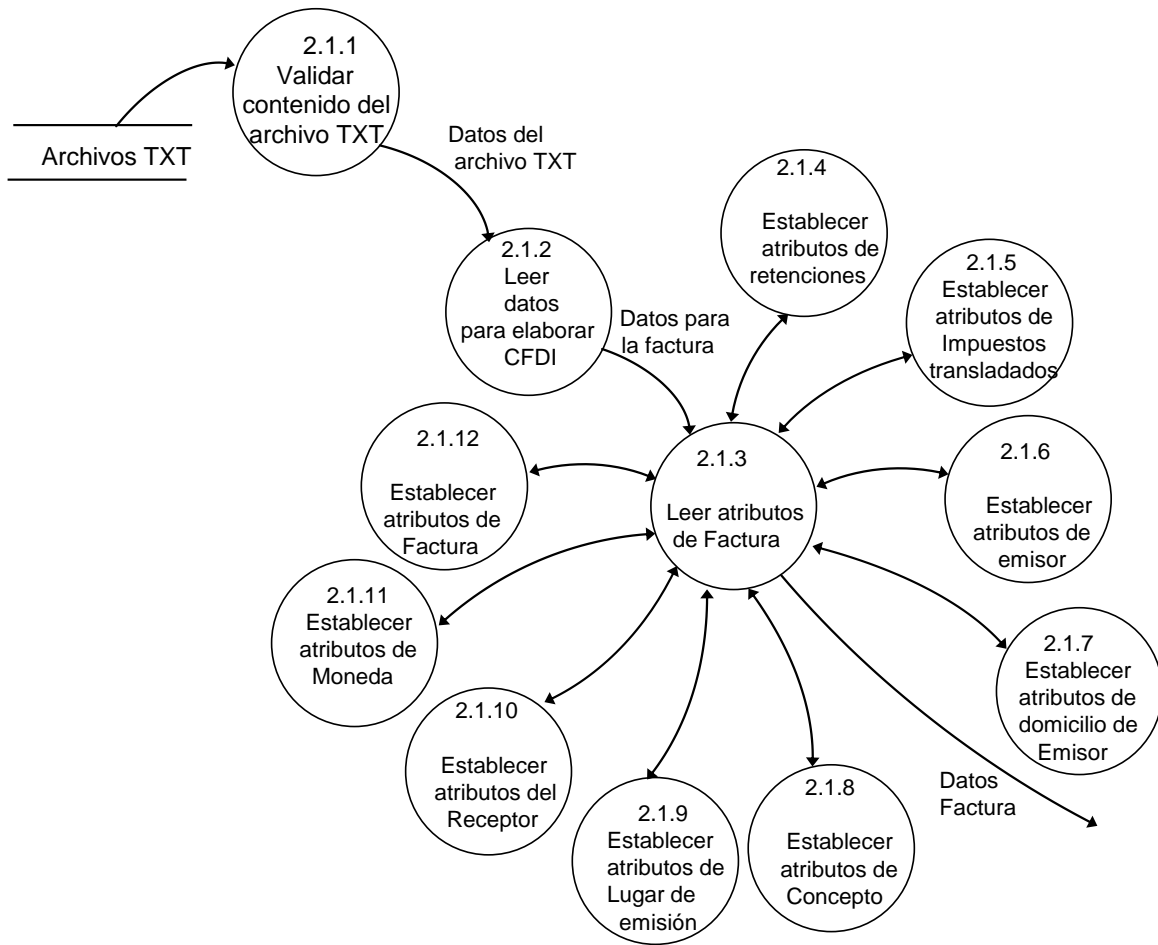


Figura 4.1.6.5 Proceso 2.1 Leer arreglo de archivo TXT

Ahora describiremos el proceso 2.2, ver la figura 4.1.6.6.

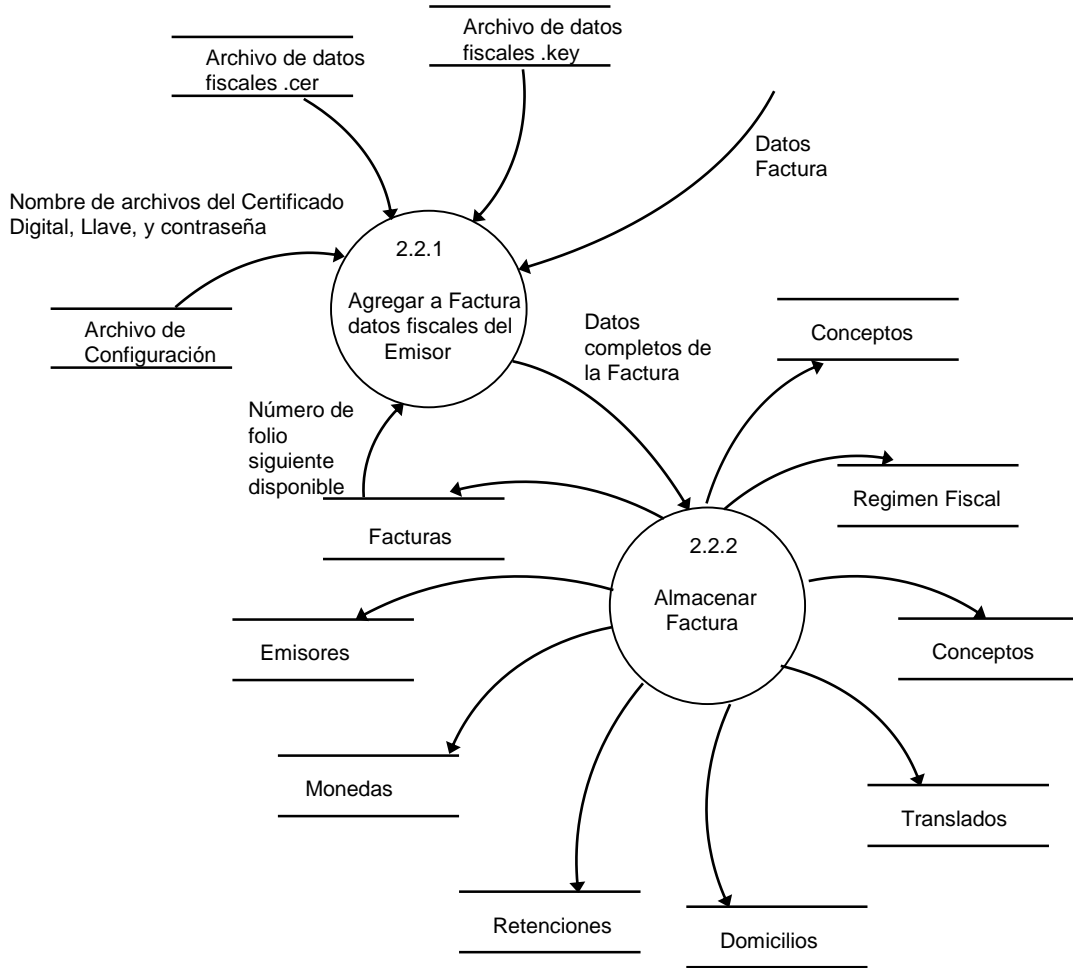


Figura 4.1.6.6 Proceso 2.2 Validar si existe alguna petición para generar CFDI

Se definirá ahora el diagrama de flujo de tercer nivel del proceso 2.3, ver en la figura 4.1.6.7.

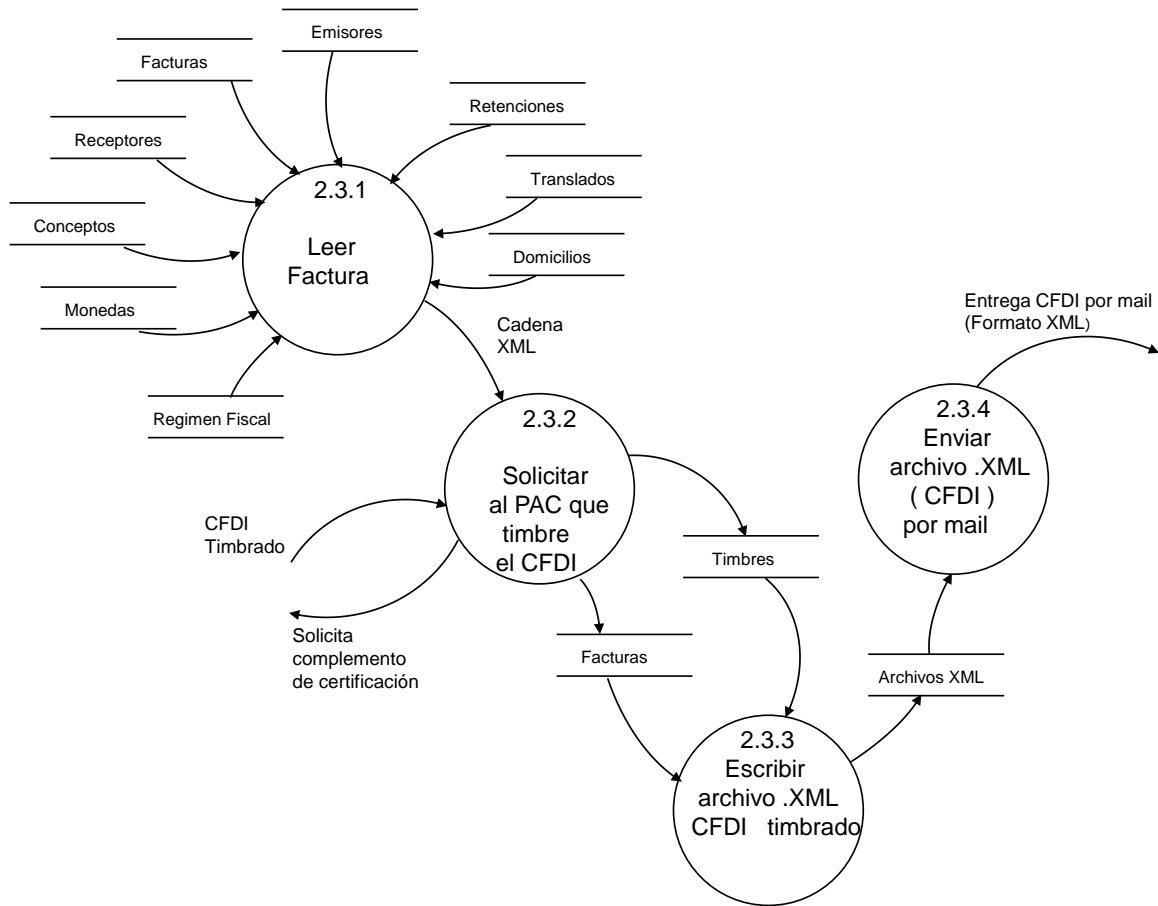


Figura 4.1.6.7 Definición del Proceso 2.3 Crear archivo XML (CFDI)

Ver en la figura 4.1.6.8 el diagrama de flujo de tercer nivel del proceso 2.4.

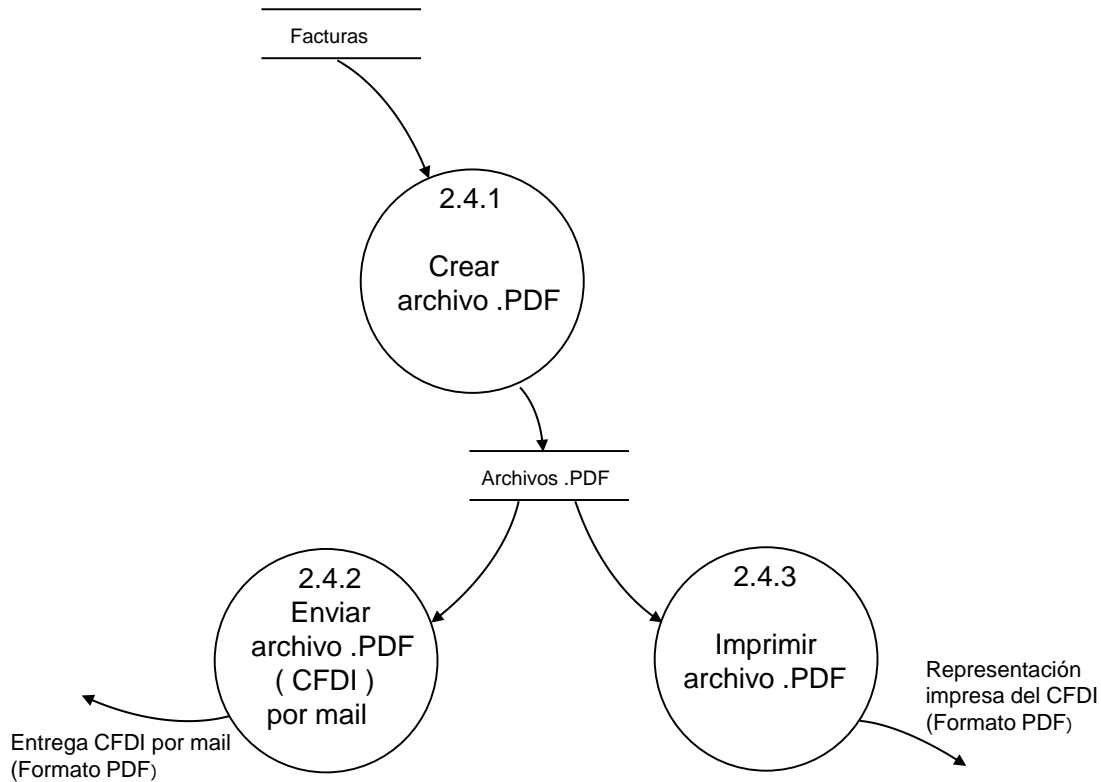


Figura 4.1.6.8 Definición del Proceso 2.4 Generar archivo .PDF (CFDI)

4.1.7 ESPECIFICACIÓN DE PROCESOS DEL SISTEMA SCFDI

El siguiente paso después de haber generado el DFD es documentar cada proceso.

Los procesos de un diagrama de flujo de datos pueden especificarse mediante redacciones textuales, o bien por diagramas de flujo de alto nivel, pero no ambas:

- Los procesos cuyos detalles se muestran por otro DFD se conocen como grupos de proceso o procesos hijo.
- Los procesos cuyos detalles se especifican por una narración textual, se llaman procesos de datos. No existe un DFD de nivel inferior para estos procesos.

Existen diferentes técnicas para la especificación de procesos, en este caso utilizaremos la técnica narrativa.

Especificación del Proceso 1.1 Recibe información de la compra de productos

Recibe la información que el usuario captura en línea, debe validar que el usuario capture los datos obligatorios, los datos que debe recibir son:

- Nombre.- Solo letras y espacios (obligatorio)
- RFC .- Números y letras (obligatorio)
- RFCE.- Números y letras
- Cuenta.- Números y letras
- Email.- Caracteres alfanuméricos, incluye: Puntos, guiones y arroba.
- Calle.- Letras
- Interior.- Números
- Exterior.- Números
- Colonia.- Letras
- CP.- Números
- Estado.- Letras
- Localidad.- Letras
- País.- Letras (obligatorio)

Y la descripción de productos:

- Id producto.- Números y letras
- Descripción del producto.- Alfanumérico
- Precio unitario del producto.- Números
- Cantidad del producto.- Números

Especificación del Proceso 1.2 Obtiene datos de productos

Lee datos del catálogo de productos, recibe el identificador del producto y con ese dato, se busca en dicho catálogo para que el usuario no tenga que capturar todos los datos del producto.

Especificación del Proceso 1.3 Obtiene datos del cliente

Lee datos del catálogo de cliente, recibe el RFC del cliente y con ese dato, se busca en dicho catálogo para que el usuario no tenga que capturar todos los datos del cliente.

Especificación del Proceso 1.4 Escribe archivo .TXT

Verifica que los datos introducidos por el usuario sean consistentes, y genera el archivo .TXT con los datos de emisor, concepto y domicilio del cliente.

Especificación del Proceso 2.1.1 Validar contenido del archivo TXT

Verificar que el archivo tenga por lo menos un producto, y con datos del emisor.

Los datos que debe incluir el archivo .TXT son:

- Nombre del emisor
- Dirección del emisor
- Email del emisor
- Teléfono del emisor

Y la descripción de productos:

- Id producto
- Descripción del producto
- Precio unitario del producto
- Cantidad del producto

Especificación del Proceso 2.1.2 Leer datos para elaborar CFDI

Valida que el archivo .TXT exista en la ruta correspondiente y que contenga los datos y el formato especificados en el apartado 4.1.8, que se titula: Definición del CFDI en formato XML.

Especificación del Proceso 2.1.3 Leer atributos de factura

Dentro del archivo .TXT busca los datos de factura, los cuales deben ser:

- Nombre del archivo
- Serie y folio
- Forma de pago
- Condiciones de pago
- Subtotal
- Descuento
- Motivo del descuento
- Total
- Método de pago
- Cuenta del método de pago
- Tipo de comprobante
- Tipo de pago
- Total de impuesto trasladado
- Total de impuesto retenido

Especificación del Proceso 2.1.4 Establecer atributos de retenciones

Obtiene del archivo .TXT los atributos de retenciones:

- Descripción del impuesto
- Importe

Especificación del Proceso 2.1.5 Establecer atributos de impuestos trasladados

Dentro del archivo .TXT identifica los campos que corresponden a los impuestos trasladados:

- Descripción del impuesto
- Importe
- Tasa

Especificación del Proceso 2.1.6 Establecer atributos de emisor

Dentro del archivo .TXT identifica los campos que correspondan al emisor.

Especificación del Proceso 2.1.7 Establecer atributos de domicilio de emisor

Dentro del archivo .TXT identifica los campos que correspondan al domicilio del emisor.

Especificación del Proceso 2.1.8 Establecer atributos de concepto

Dentro del archivo .TXT identifica los campos que correspondan al concepto, o los conceptos.

Especificación del Proceso 2.1.9 Establecer atributos de lugar de emisión

Dentro del archivo .TXT identifica los campos que correspondan al domicilio donde se realizó la emisión del CFDI.

Especificación del Proceso 2.1.10 Establecer atributos del receptor

Dentro del archivo .TXT identifica los campos que correspondan al receptor.

Especificación del Proceso 2.1.11 Establecer atributos de moneda

Dentro del archivo .TXT identifica los campos que correspondan a la moneda en que se realizó la compra del producto.

Especificación del Proceso 2.1.12 Establecer atributos de factura

Dentro del archivo .TXT identifica los campos que correspondan a la factura.

Especificación del Proceso 2.2.1 Agregar a factura datos fiscales del emisor

Se debe generar un archivo XML a partir de la cadena XML, y se debe pasar por un proceso para obtener la Cadena Original y Sello Digital.

Especificación del Proceso 2.2.2 Almacenar Factura

Debe permitir almacenar el CFDI con todos los componentes que lo conforman, son datos a nivel factura, datos del emisor, datos del receptor, datos del Concepto, moneda, e impuestos.

Especificación del Proceso 2.3.1 Leer Factura

Debe permitir Leer el CFDI con todos los componentes que lo conforman, son datos a nivel factura, datos del emisor, datos del receptor, datos del Concepto, moneda, e impuestos.

Especificación del Proceso 2.3.2 Solicita al PAC que timbre el CFDI

Solicitará al PAC el complemento de Certificación para el Timbrado del CFDI.

Especificación del Proceso 2.3.3 Escribir archivo XML (CFDI Timbrado)

Deberá leer la información de factura y todos sus componentes, para poder generar el archivo XML.

Especificación del Proceso 2.3.4 Enviar archivo XML (CFDI) por mail

Recibirá el nombre y ruta de un archivo .XML y el email destino, y en automático deberá enviar el archivo al email indicado.

Especificación del Proceso 2.4.1 Crear archivo .PDF

Debe leer la información de la factura con todos sus componentes, como receptor, emisor, conceptos, impuestos, etc., y recibir el nombre y ruta para genera el archivo PDF. Al ser llamado este proceso generará el archivo PDF.

Especificación del Proceso 2.4.2 Enviar archivo PDF (CFDI) por mail

Recibirá el nombre y ruta de un archivo .PDF y el email destino, y en automático deberá enviar el archivo al email indicado.

Especificación del Proceso 2.4.3 Imprimir archivo .PDF

Debe recibir el archivo PDF, y poder realizar la representación impresa del CFDI.

4.1.8 MODELO DE IMPLEMENTACIÓN

El modelo de implementación especifica cómo se implementará en forma tangible, el comportamiento descrito en el modelo esencial, dentro de la etapa del análisis.

El modelo de implementación, se divide a su vez en tres modelos:

- Modelo de implementación del usuario: especifica el nivel más alto de diseño en términos de procesadores físicos y las interfaces que existen entre ellos.
- Modelo de implementación del sistema: detalla los componentes de software que tendrán los procesadores y sus interfaces.
- Modelo de implementación del programa: es el nivel más bajo de diseño, ya que describe la organización de los módulos de código en los programas que conforman el sistema.

Modelo de Implementación del Usuario

No existen reglas estrictas para la elaboración del modelo de implementación del usuario, sin embargo, se recomienda seguir un cierto orden con el fin de simplificar esta tarea, tomando como base el modelo esencial y haciendo uso de otros modelos y herramientas enfocados a la implementación:

- Elaboración del modelo de procesadores: declara la forma en que se asignarán los procesos y datos a máquinas, gente y organizaciones. Se utilizan los mismos elementos de un DFD.

-
- Definición de diálogos: diseño de pantallas y reportes que entregará el sistema.
 - Elaboración del diagrama de transición de estados: muestra el flujo de acciones que el usuario deberá realizar dentro del sistema para la generación de sus reportes y ejecución de sus procesos.
 - Generación de prototipos: realización de un ejemplar del sistema (generalmente se simulan las interfaces) para ilustrar la viabilidad y aceptación del diseño.

En el modelo de implementación del usuario se determina la elección de los dispositivos de entrada y salida, el formato de todas las entradas que fluyen desde los terminadores hasta el sistema, y de todas las salidas que fluyen desde el sistema hacia los terminadores, así mismo, se determina la secuencia y los tiempos de entradas y salidas cuando se trata de sistemas en línea.

Diagrama de Procesadores

El primer paso para genera el diagrama de procesadores, es asignar las tareas esenciales a equipos para la parte automatizada, como se muestra en la figura 4.1.8.1.

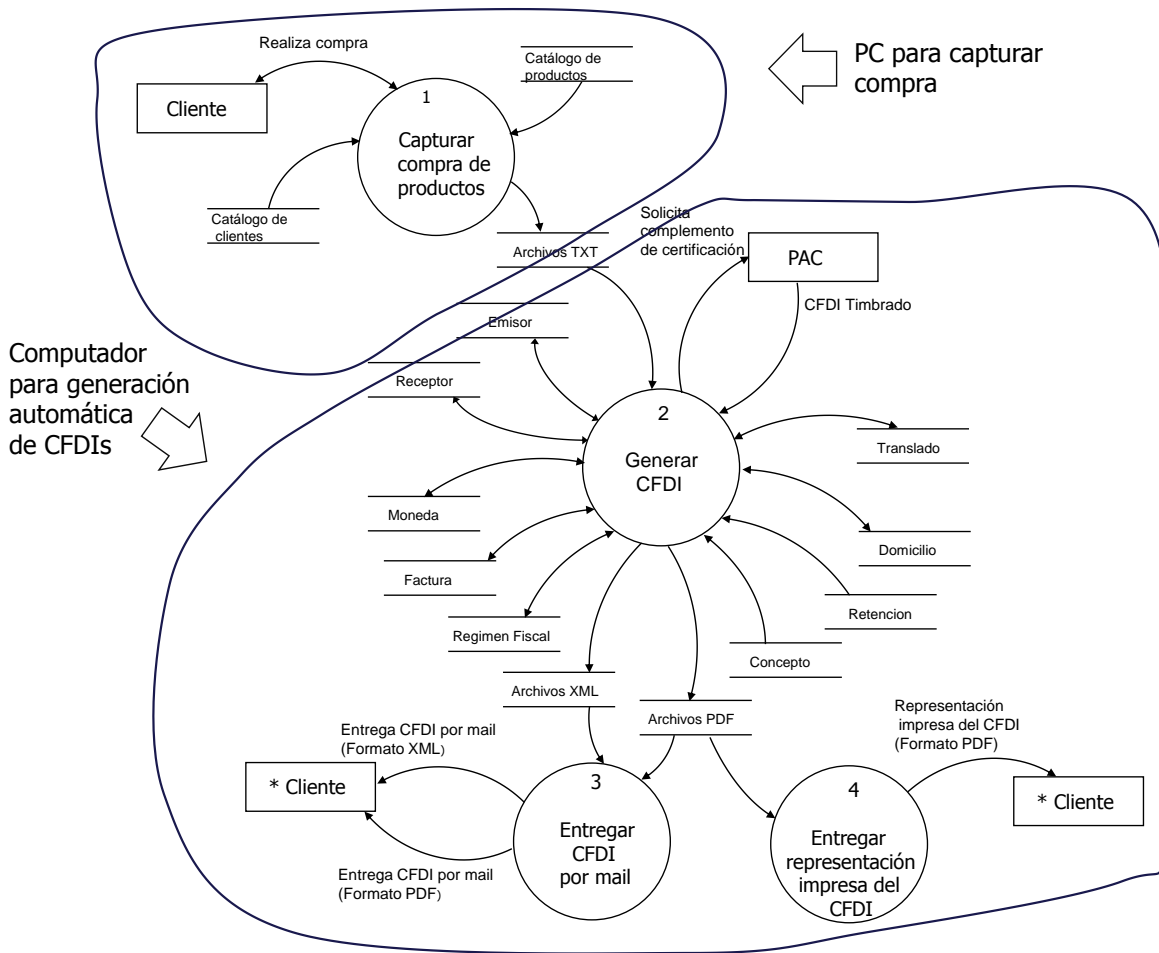


Figura 4.1.8.1 Se identifica los procesadores que soportarán las actividades esenciales

Ver en la figura 4.1.8.2 el diagrama de procesadores del sistema SCFDI

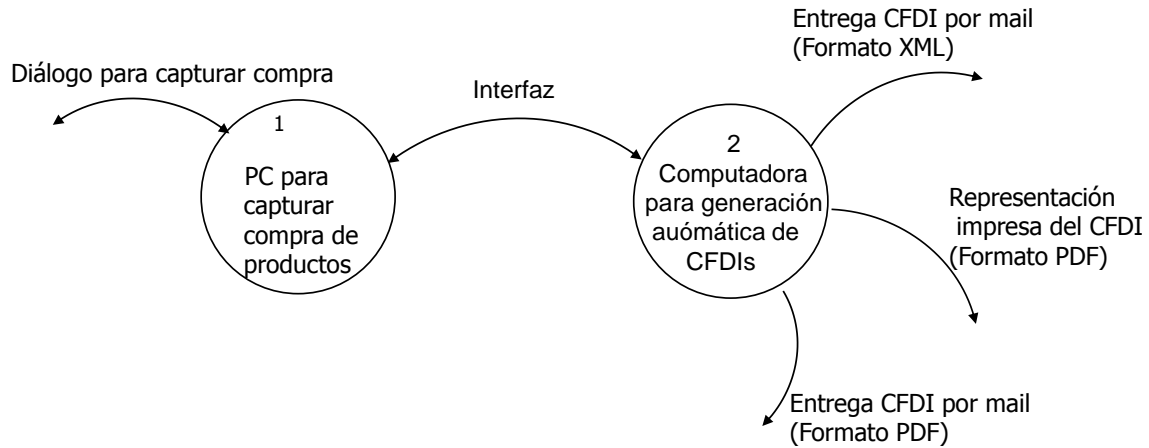


Figura 4.1.8.2 Modelo de Procesadores del Sistema SCFDI

Formatos de Entrada /Salida del Sistema SCFDI

Definición del CFDI en formato XML

Como regla general los documentos XML generan una estructura de árbol en la cual en la parte más alta se encuentra el nodo principal y de este pueden depender n nodos hijos y de ellos otros más; por la forma como se realizan estos archivos XML pueden contener:

- Valores del nodo, si este se encuentra entre la etiqueta de abertura y de cierre del nodo, por ejemplo
 - <Nodo>Valor</Nodo>
- Atributos del nodo, si estos se encuentran dentro de la etiqueta de abertura y tienen un nombre y un valor, pueden contener n atributos un nodo.
 - <Nodo nombreDelAtributo1="valorDelAtributo1"
nombreDelAtributo2="valorDelAtributo2">Valor del nodo</Nodo>

El método que se puede utilizar para restringir la forma de los archivos XML la definición de un esquema el cual se declara dentro de este archivo XML como un atributo más.

Para los archivos CFDI versión 3.2 que es la que se encuentra actualmente vigente y que son los mismos que genera nuestra aplicación generan archivos con la siguiente estructura de árbol. Vea figura 4.1.8.3 y figura 4.1.8.4.

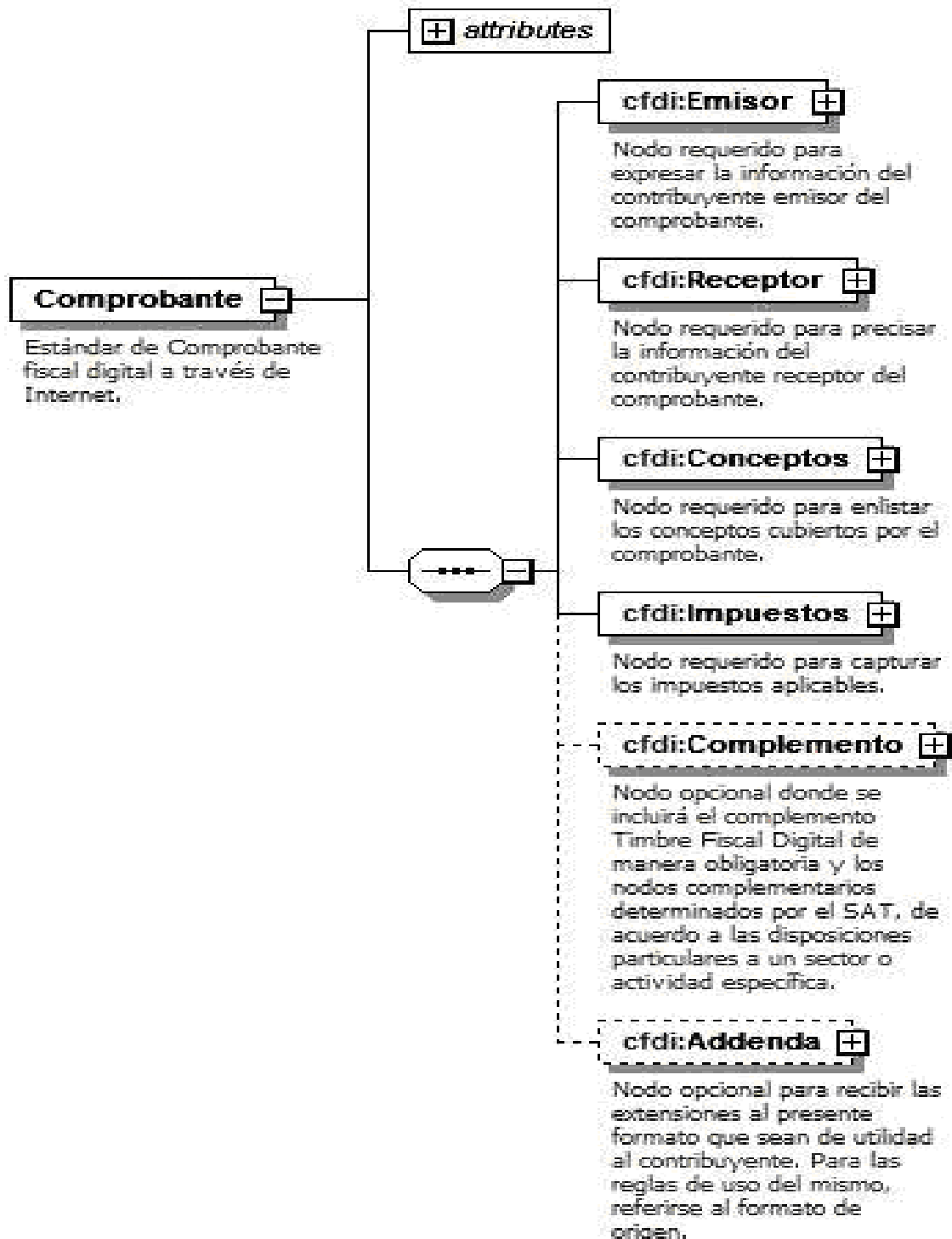


Figura 4.1.8.3 Estructura de árbol para archivos CFDI's

Descripción de los atributos del nodo “cfdi:Comprobante”:

ATRIBUTO	DESCRIPCIÓN	USO	TIPO	Longitud mínima	Longitud máxima
Versión	Atributo requerido con valor prefijado a 3.2 que indica la versión del estándar bajo el que se encuentra expresado el comprobante.	Requerido	cadena	3	3
Serie	Atributo opcional para precisar la serie para control interno del contribuyente. Este atributo acepta una cadena de caracteres alfabéticos de 1 a 3 caracteres sin incluir caracteres acentuados.	Opcional	cadena	1	25
Folio	Atributo opcional para control interno del contribuyente que acepta un valor numérico entero superior a 0 que expresa el folio del comprobante.	Opcional	cadena	1	
Fecha	Atributo requerido para la expresión de la fecha y hora de expedición del comprobante fiscal. Se expresa en la forma aaa-mm-ddThh:mm:ss, de acuerdo con la especificación ISO 8601.	Requerido	cadena	19	19
Sello	Atributo requerido para contener el sello digital del comprobante fiscal, al que hacen referencia las reglas de resolución miscelánea aplicable. El sello deberá ser expresado cómo una cadena de texto en formato Base 64.	Requerido	cadena	1	

formaDePago	Atributo requerido para precisar la forma de pago que aplica para este comprobante fiscal digital a través de Internet. Se utiliza para expresar Pago en una sola exhibición o número de parcialidad pagada contra el total de parcialidades, Parcialidad 1 de X.	Requerido	cadena	1	
noCertificado	Atributo requerido para expresar el número de serie del certificado de sello digital que ampara al comprobante, de acuerdo al acuse correspondiente a 20 posiciones otorgado por el sistema del SAT.	Requerido	cadena	20	20
Certificado	Atributo requerido que sirve para expresar el certificado de sello digital que ampara al comprobante como texto, en formato base 64.	Requerido	cadena		
condicionesDePago	Atributo opcional para expresar las condiciones comerciales aplicables para el pago del comprobante fiscal digital a través de Internet.	Opcional	cadena	1	
Subtotal	Atributo requerido para representar la suma de los importes antes de descuentos e impuestos.	Requerido	importe		
Descuento	Atributo opcional para representar el importe total de los descuentos aplicables antes de impuestos.	Opcional	importe		
MotivoDescuento	Atributo opcional para expresar el motivo del descuento aplicable.	Opcional	cadena		
TipoCambio	Atributo opcional para representar el tipo de cambio conforme a la moneda usada	Opcional	cadena		
Moneda	Atributo opcional para expresar la moneda utilizada para expresar los montos	Opcional	cadena		

Total	Atributo requerido para representar la suma del subtotal, menos los descuentos aplicables, más los impuestos trasladados, menos los impuestos retenidos.	Requerido	importe		
TipoDeComprobante	Atributo requerido para expresar el efecto del comprobante fiscal para el contribuyente emisor.	Requerido	cadena		
MetodoDePago	Atributo requerido de texto libre para expresar el método de pago de los bienes o servicios amparados por el comprobante. Se entiende como método de pago leyendas tales como: cheque, tarjeta de crédito o débito, depósito en cuenta, etc.	Requerido	cadena	1	
LugarExpedicion	Atributo requerido para incorporar el lugar de expedición del comprobante.	Requerido	cadena	1	
NumCtaPago	Atributo opcional para incorporar al menos los cuatro últimos dígitos del número de cuenta con la que se realizó el pago.	Opcional	cadena	4	

Tabla 4.1.8.1 Definición de atributos

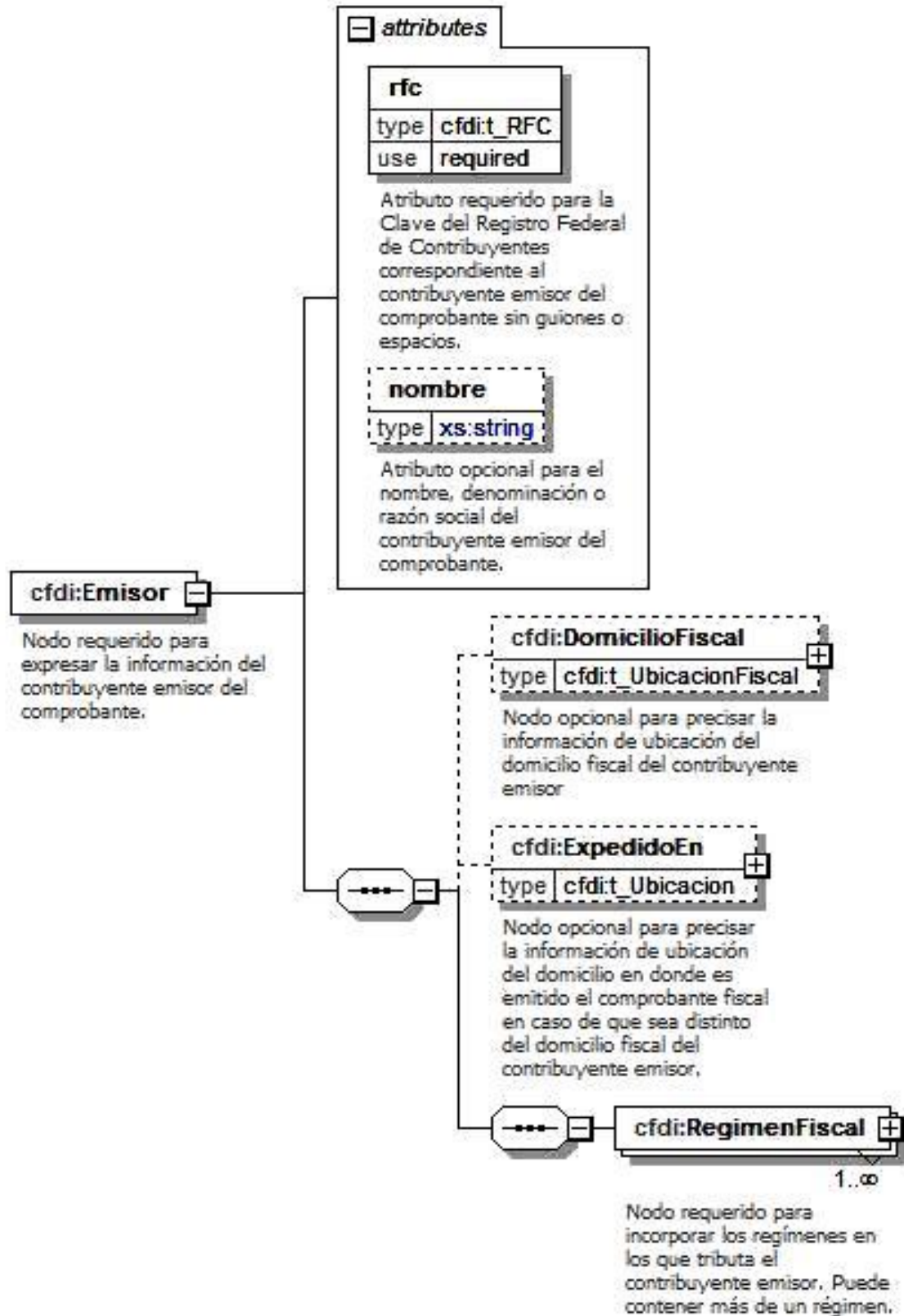


Figura 4.1.8.4 Estructura de árbol para Emisor

Descripción de los atributos del nodo “cfdi:Emisor”: vea tabla 4.1.8.2

ATRIBUTO	DESCRIPCIÓN	USO	TIPO	LM	LX
rfc	Atributo requerido para la Clave del Registro Federal de Contribuyentes correspondiente al contribuyente emisor del comprobante sin guiones o espacios.	requerido	rfc		
nombre	Atributo opcional para el nombre, denominación o razón social del contribuyente emisor del comprobante.	requerido	cadena	1	

Tabla 4.1.8.2 Atributos del nodo cfdi:emisor

Descripción de los atributos del nodo “cfdi:DomicilioFiscal”: vea tabla 4.1.8.3

ATRIBUTO	DESCRIPCIÓN	USO	TIPO	LM	LX
Calle	Este atributo opcional sirve para precisar la avenida, calle, camino o carretera donde se da la ubicación.	opcional	cadena	1	
NoExterior	Este atributo opcional sirve para expresar el número particular en donde se da la ubicación sobre una calle dada.	opcional	cadena	1	
NoInterior	Este atributo opcional sirve para expresar información adicional para especificar la ubicación cuando calle y número exterior (noExterior) no resulten suficientes para determinar la ubicación de forma precisa.	opcional	cadena	1	
Colonia	Este atributo opcional sirve para precisar la colonia en donde se da la ubicación cuando se desea ser más específico en casos de ubicaciones urbanas.	opcional	cadena	1	
Localidad	Atributo opcional que sirve para precisar la ciudad o población donde se da la ubicación.	opcional	cadena	1	
Referencia	Atributo opcional para expresar una referencia de ubicación adicional.	opcional	cadena	1	
Municipio	Atributo opcional que sirve para precisar el municipio o delegación (en el caso del Distrito Federal) en donde se da la ubicación.	opcional	cadena	1	
Estado	Atributo opcional que sirve para precisar el estado o entidad federativa donde se da la ubicación.	opcional	cadena	1	
País	Atributo requerido que sirve para precisar el país donde se da la ubicación.	requerido	cadena	1	
codigoPostal	Atributo opcional que sirve para asentar el código postal en donde se da la ubicación.	opcional	cadena		

Tabla 4.1.8.3 Atributos del nodo cfdi:DomicilioFiscal

Descripción de los atributos del nodo “cfdi:ExpedidaEn”: vea tabla 4.1.8.4.

ATRIBUTO	DESCRIPCIÓN	USO	TIPO	LM	LX
Calle	Este atributo opcional sirve para precisar la avenida, calle, camino o carretera donde se da la ubicación.	opcional	cadena	1	
noExterior	Este atributo opcional sirve para expresar el número particular en donde se da la ubicación sobre una calle dada.	opcional	cadena	1	
noInterior	Este atributo opcional sirve para expresar información adicional para especificar la ubicación cuando calle y número exterior (noExterior) no resulten suficientes para determinar la ubicación de forma precisa.	opcional	cadena	1	
Colonia	Este atributo opcional sirve para precisar la colonia en donde se da la ubicación cuando se desea ser más específico en casos de ubicaciones urbanas.	opcional	cadena	1	
Localidad	Atributo opcional que sirve para precisar la ciudad o población donde se da la ubicación.	opcional	cadena	1	
Referencia	Atributo opcional para expresar una referencia de ubicación adicional.	opcional	cadena	1	
Municipio	Atributo opcional que sirve para precisar el municipio o delegación (en el caso del Distrito Federal) en donde se da la ubicación.	opcional	cadena	1	
Estado	Atributo opcional que sirve para precisar el estado o entidad federativa donde se da la ubicación.	opcional	Cadena	1	
País	Atributo requerido que sirve para precisar el país donde se da la ubicación.	requerido	cadena	1	
codigoPostal	Atributo opcional que sirve para asentar el código postal en donde se da la ubicación.	opcional	cadena		

Tabla 4.1.8.4 Atributos del nodo cfdi:ExpedidaEn

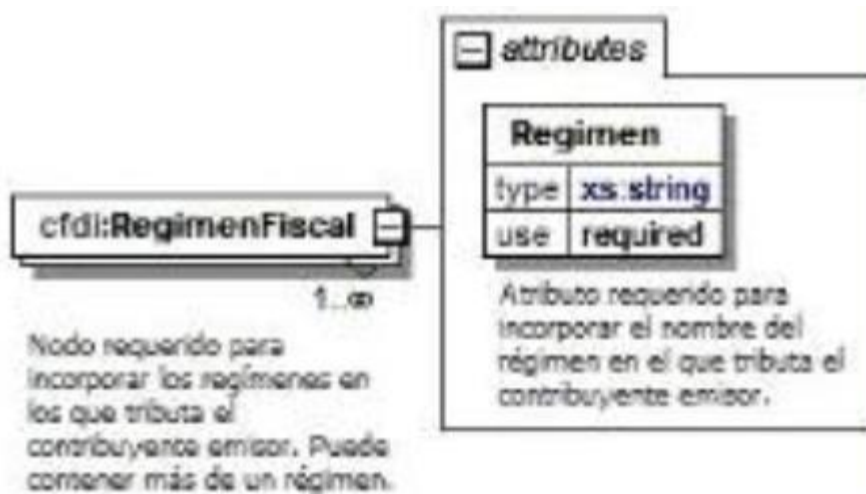


Figura 4.1.8.5 Estructura de árbol para RegimenFiscal

Descripción de los atributos del nodo “cfdi:RegimenFiscal”: vea tabla 4.1.8.5

ATRIBUTO	DESCRIPCIÓN	USO	TIPO	LM	LX
Regimen	Atributo requerido para incorporar el nombre del régimen en el que tributa el contribuyente emisor.	requerido	1		

Tabla 4.1.8.5 Atributos del nodo cfdi:RegimenFiscal

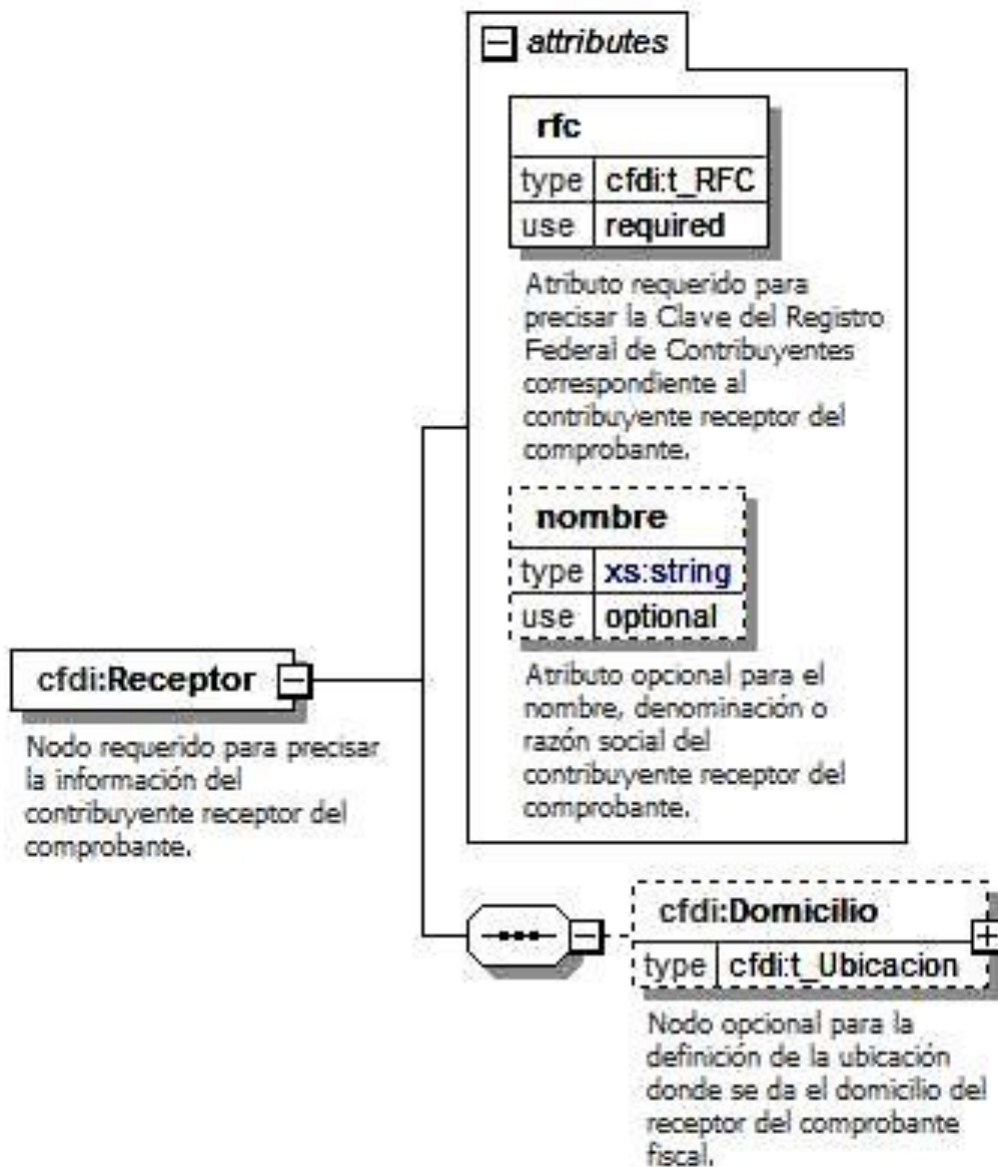


Figura 4.1.8.6 Estructura de árbol para Receptor

Descripción de los atributos del nodo “cfdi:Receptor”: vea tabla 4.1.8.6.

ATRIBUTO	DESCRIPCIÓN	USO	TIPO	LM	LX
rfc	Atributo requerido para precisar la Clave del Registro Federal de Contribuyentes correspondiente al contribuyente receptor del comprobante.	requerido	Cadena	12	13
nombre	Atributo opcional para el nombre, denominación o razón social del contribuyente receptor del comprobante.	opcional	cadena	1	

Tabla 4.1.8.6 Atributos del nodo cfdi:Receptor

Descripción de los atributos del nodo “cfdi:Domicilio”: vea tabla 4.1.8.7.

ATRIBUTO	DESCRIPCIÓN	USO	TIPO	LM	LX
calle	Este atributo opcional sirve para precisar la avenida, calle, camino o carretera donde se da la ubicación.	opcional	cadena	1	
noExterior	Este atributo opcional sirve para expresar el número particular en donde se da la ubicación sobre una calle dada.	opcional	cadena	1	
noInterior	Este atributo opcional sirve para expresar información adicional para especificar la ubicación cuando calle y número exterior (noExterior) no resulten suficientes para determinar la ubicación de forma precisa.	opcional	cadena	1	
colonia	Este atributo opcional sirve para precisar la colonia en donde se da la ubicación cuando se desea ser más específico en casos de ubicaciones urbanas.	opcional	cadena	1	
localidad	Atributo opcional que sirve para precisar la ciudad o población donde se da la ubicación.	opcional	cadena	1	

referencia	Atributo opcional para expresar una referencia de ubicación adicional.	opcional	cadena	1	
municipio	Atributo opcional que sirve para precisar el municipio o delegación (en el caso del Distrito Federal) en donde se da la ubicación.	opcional	cadena	1	
estado	Atributo opcional que sirve para precisar el estado o entidad federativa donde se da la ubicación.	opcional	cadena	1	
pais	Atributo requerido que sirve para precisar el país donde se da la ubicación.	requerido	cadena	1	
codigoPostal	Atributo opcional que sirve para asentar el código postal en donde se da la ubicación.	opcional	cadena		

Tabla 4.1.8.7 Atributos del nodo cfdi:Domicilio

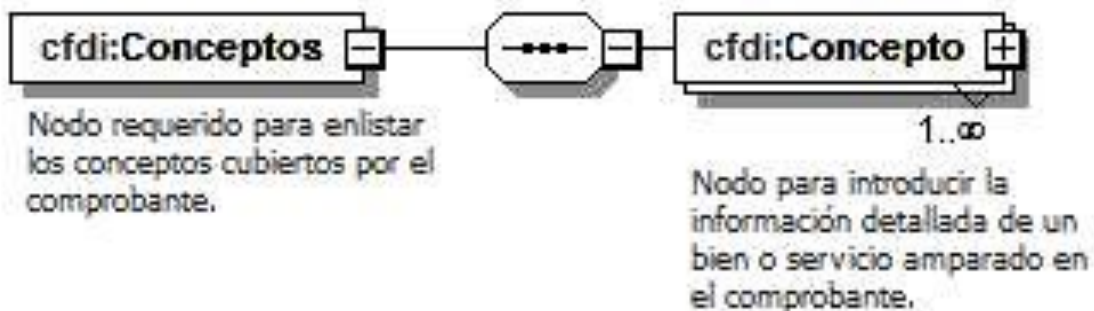


Figura 4.1.8.7 Estructura de árbol para Conceptos

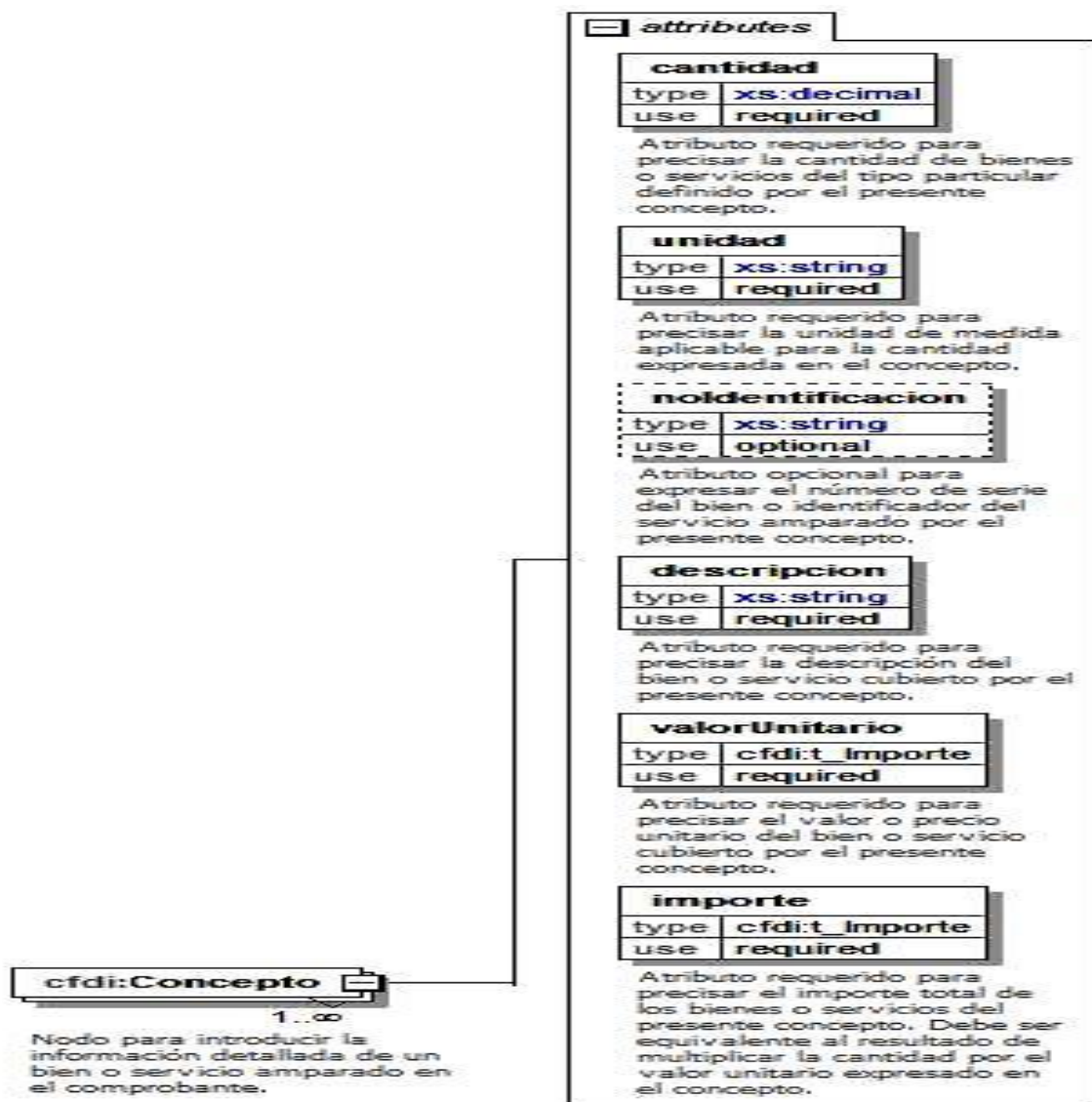


Figura 4.1.8.8 Estructura de árbol para Conceptos

Descripción de los atributos del nodo “cfdi:Concepto”: vea tabla 4.1.8.8

ATRIBUTO	DESCRIPCIÓN	USO	TIPO	LM	LX
Cantidad	Atributo requerido para precisar la cantidad de bienes o servicios del tipo particular definido por el presente concepto.	requerido	decimal		
Unidad	Atributo requerido para precisar la unidad de medida aplicable para la cantidad expresada en el concepto	requerido	cadena	1	
nolidentificacion	Atributo opcional para expresar el número de serie del bien o identificador del servicio amparado por el presente concepto.	requerido	cadena	1	

Descripción	Atributo requerido para precisar la descripción del bien o servicio cubierto por el presente concepto.	requerido	cadena	1	
valorUnitario	Atributo requerido para precisar el valor o precio unitario del bien o servicio cubierto por el presente concepto.	requerido	importe		
Importe	Atributo requerido para precisar el importe total de los bienes o servicios del presente concepto. Debe ser equivalente al resultado de multiplicar la cantidad por el valor unitario expresado en el concepto.	requerido	importe		

Tabla 4.1.8.8 Atributos del nodo cfdi:Concepto

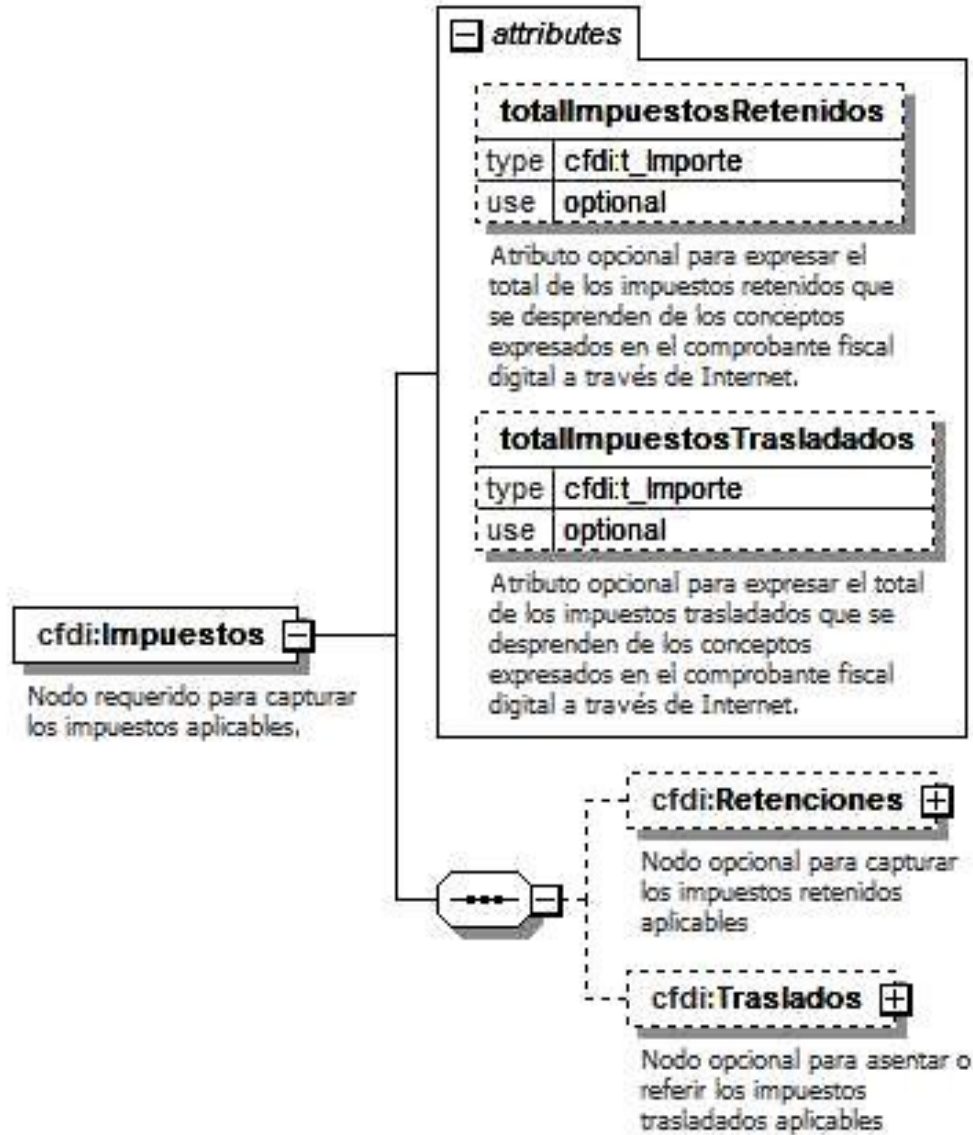


Figura 4.1.8.9 Estructura de árbol para Impuestos

Descripción de los atributos del nodo “cfdi:” vea tabla 4.1.8.9.

ATRIBUTO	DESCRIPCIÓN	USO	TIPO	LM	LX
totalImpuestosRetenidos	Atributo opcional para expresar el total de los impuestos retenidos que se desprenden de los conceptos expresados en el comprobante fiscal digital a través de Internet.	opcional	importe		

totalImpuestosTrasladados	Atributo opcional para expresar el total de los impuestos trasladados que se desprenden de los conceptos expresados en el comprobante fiscal digital a través de Internet.	opcional	importe		
---------------------------	--	----------	---------	--	--

Tabla 4.1.8.9 Atributos del nodo cfdi

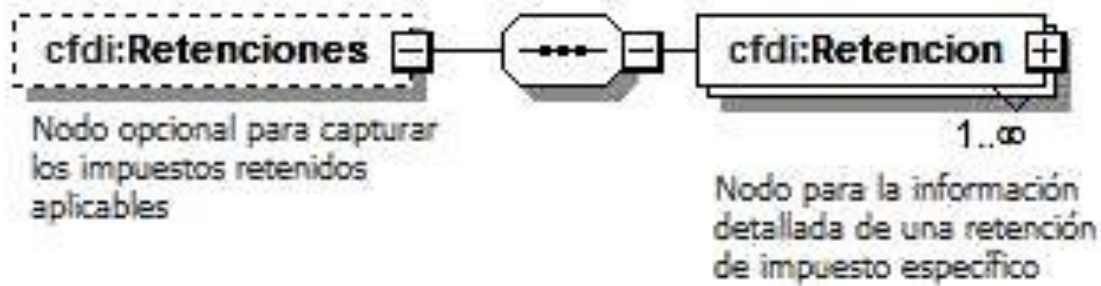


Figura 4.1.8.10 Estructura para Retenciones

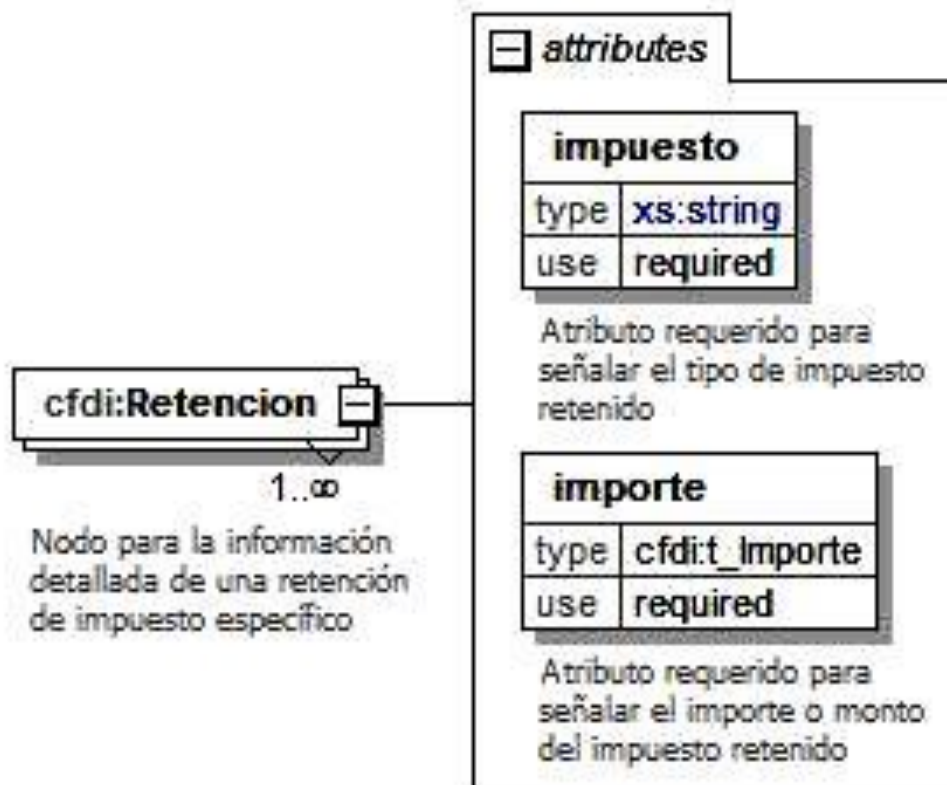


Figura 4.1.8.11 Estructura de árbol para Retenciones

Descripción de los atributos del nodo “cfdi:Retencion”: vea tabla 4.1.8.10.

ATRIBUTO	DESCRIPCIÓN	USO	TIPO	LM	LX
Impuesto	Atributo requerido para señalar el tipo de impuesto retenido	requerido	cadena		
Importe	Atributo requerido para señalar el importe o monto del impuesto retenido	requerido	importe		

Tabla 4.1.8.10 Atributos del nodo cfdi:Retención

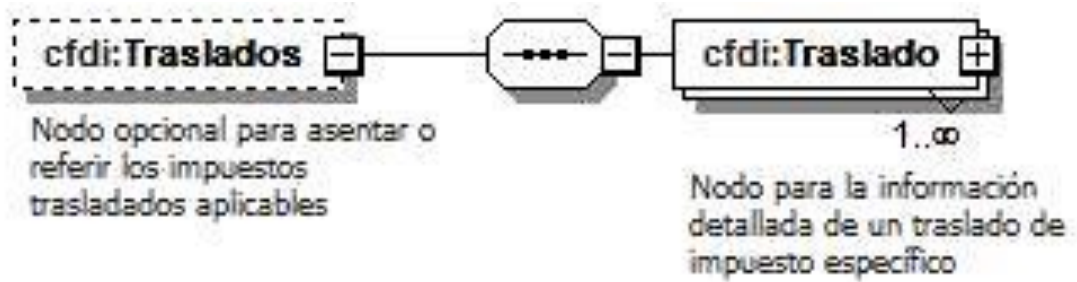


Figura 4.1.8.12 Estructura de árbol para Traslados

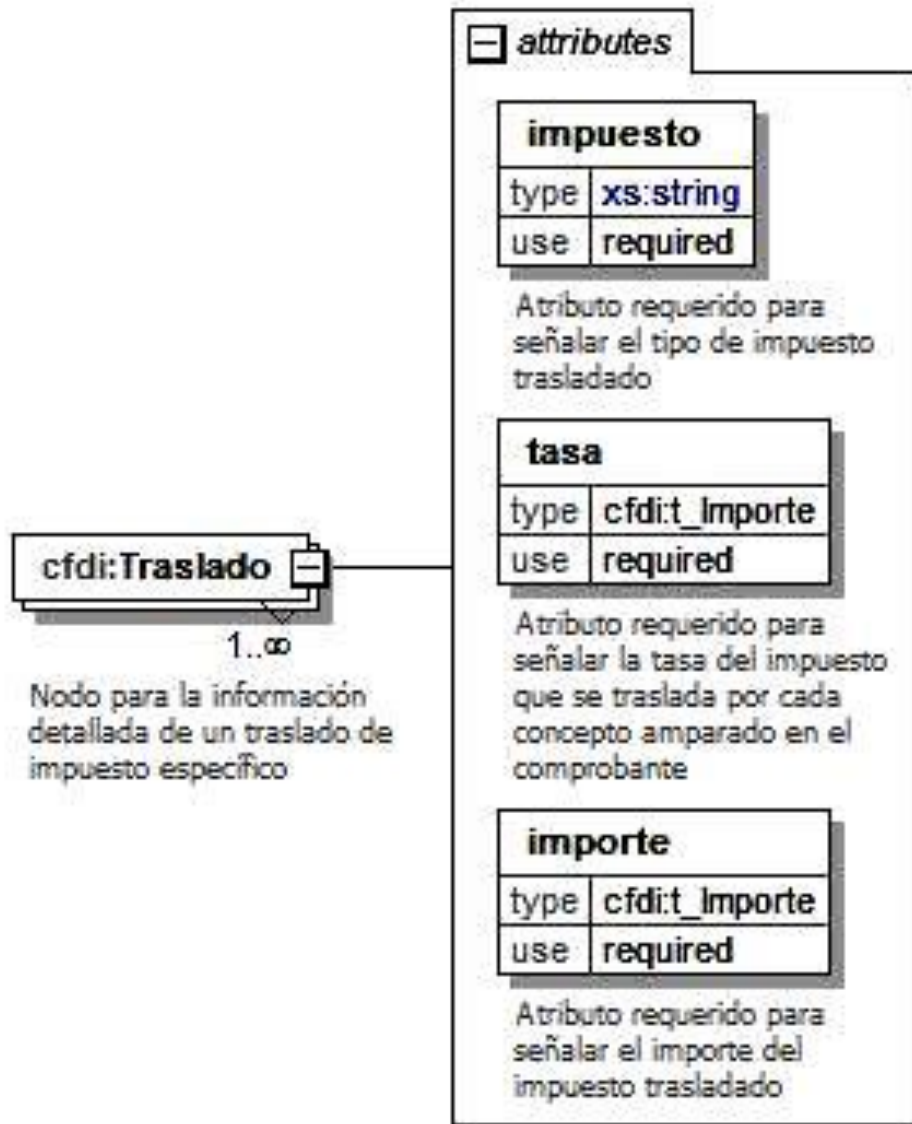


Figura 4.1.8.13 Estructura de árbol para Traslados

Descripción de los atributos del nodo “cfdi:Traslado”: vea tabla 4.1.8.11

ATRIBUTO	DESCRIPCIÓN	USO	TIPO	LM	LX
impuesto	Atributo requerido para señalar el tipo de impuesto trasladado	requerido	Cadena		

tasa	Atributo requerido para señalar la tasa del impuesto que se traslada por cada concepto amparado en el comprobante	requerido	importe		
importe	Atributo requerido para señalar el importe del impuesto trasladado	requerido	importe		

Tabla 4.1.8.11 Atributos del nodo cfdi:Traslado

Definición del CFDI en formato PDF

La especificación del CFDI en formato PDF se encuentra detallada en el capítulo 1.4 Descripción básica de lo que debe de contener un CFDI.

Diagrama de Transición de Estados

El siguiente paso es generar el diagrama de transición de estados de nuestro Sistema SCFDI.

El diagrama de transición de estados es una herramienta de modelado que enfatiza el comportamiento dependiente del tiempo de un sistema.

Los componentes del diagrama de transición de estados son los siguientes:

- Estado. Es pasivo, representa la espera de que algo suceda, un proceso puede estar en un sólo estado a un tiempo. El nombre que se le dé al estado debe reflejar lo que el proceso está esperando, que se despliegue algo (en el caso de iteraciones con pantallas).
- Transición. Una transición representa un cambio de estado. Múltiples transiciones desde o hacia un estado singular son permitidas. Una transición puede dejar y retomar el mismo estado. Las transiciones tienen dos partes:
 - Condición de transición
Indica las causas por las cuales el sistema cambia de un estado a otro, la condición se lee como “cuando....”
 - Condición de acción

Se ejecuta cuando la condición se cumple, la condición se lee como “hacer...”

Nada sucede excepto la acción entre el tiempo que la condición se cumple y la llegada del nuevo estado.

La condición puede ocasionar que un procesamiento se lleve a cabo o que una señal sea producida.

La acción de la transición puede no ser etiquetada si la sola acción es una transición a otro estado. Vea figura 4.1.8.14

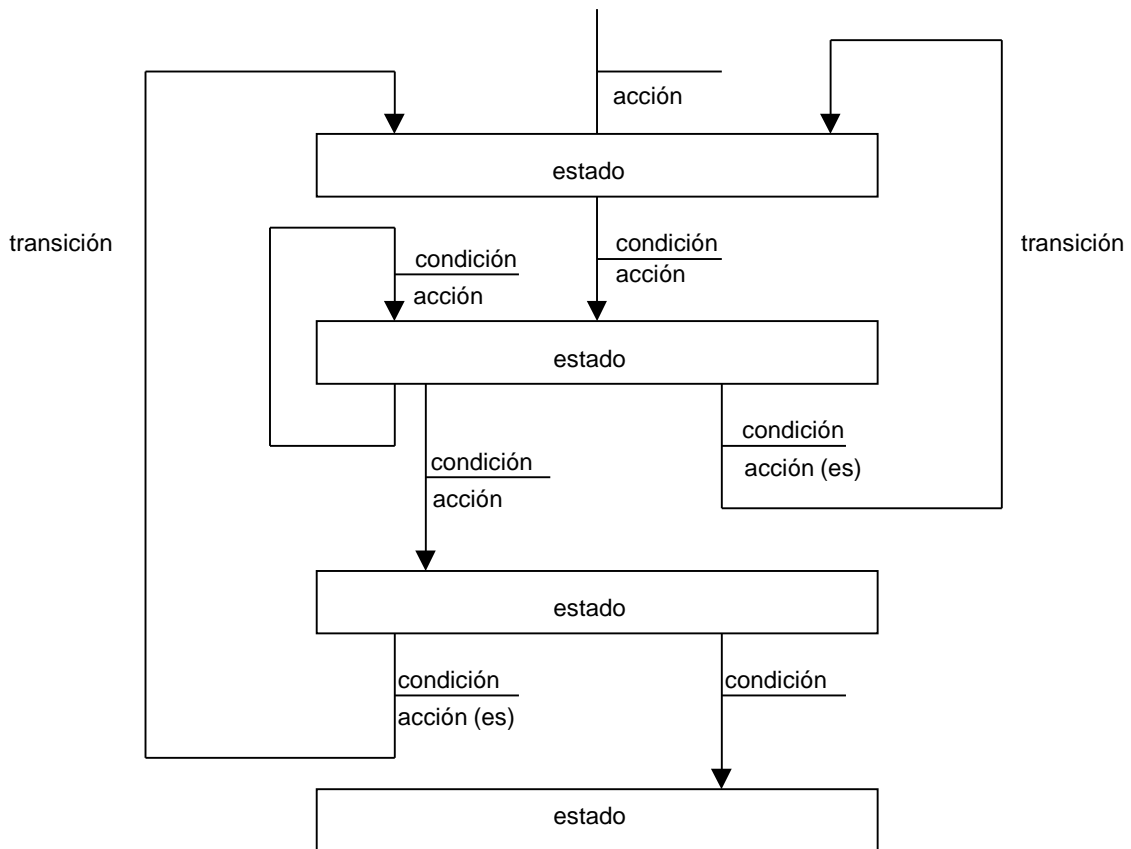


Figura 4.1.8.14 Ejemplo del Diagrama de transición de estados

Ver en la figura 4.1.8.15 el Diagrama de transición de estados de estados del Front-End de nuestra aplicación SCFDI.

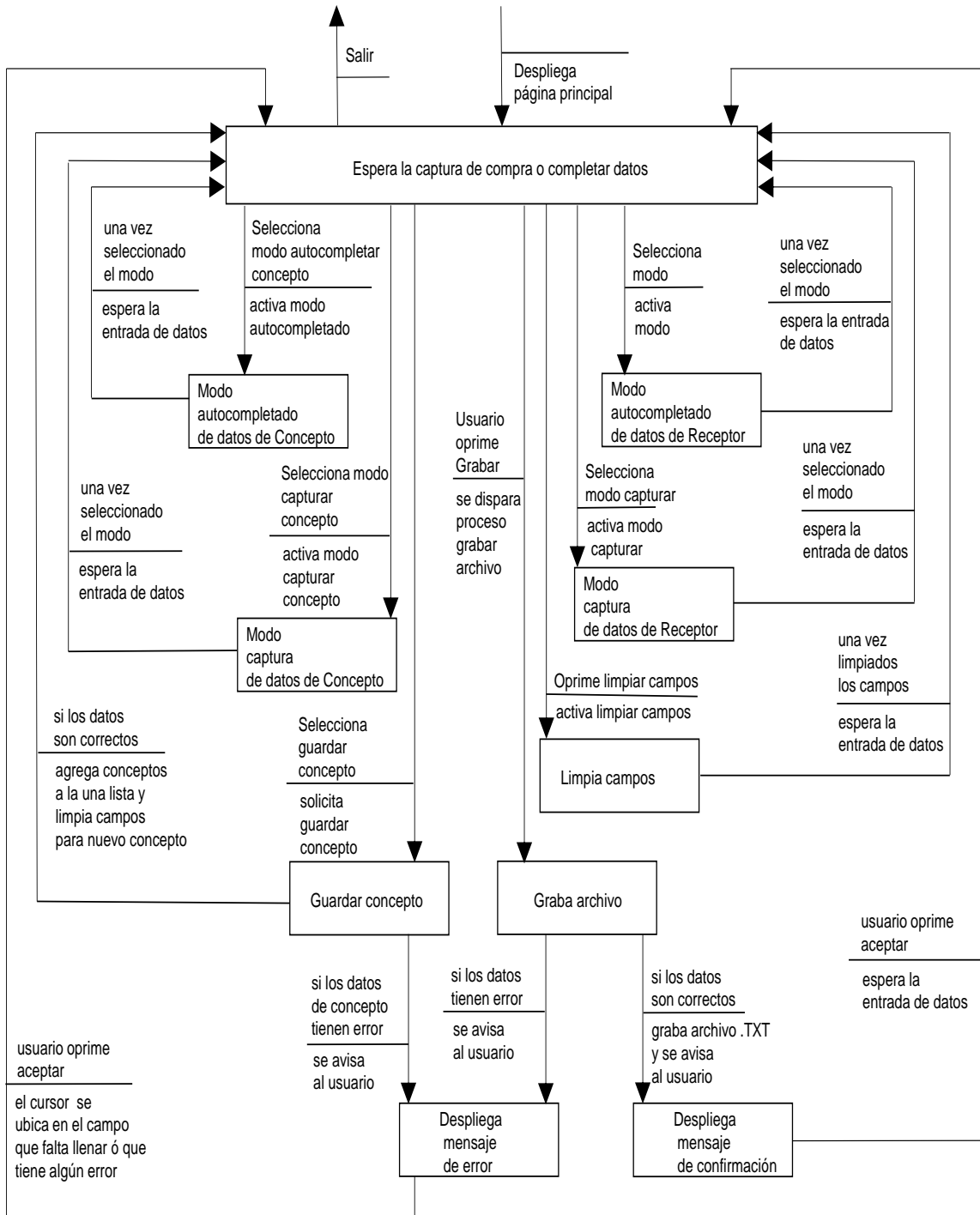


Figura 4.1.8.15 Diagrama de transición de estados del Front-End del Sistema SCFDI

Generación de Prototipos

Para el diseño de interfaz del usuario, se propuso una distribución como se muestra en la figura 4.1.8.16, el Bloque 1 para el encabezado y logo de la empresa.

El bloque2 para ejecutar acciones requeridas por el usuario, el Bloque 3 para leyendas y cajas para itroducir datos y algunos botones de acciones, en el Bloque 4 para Notificaciones, el Bloque 5 para desplegar datos de referencia de la empresa.

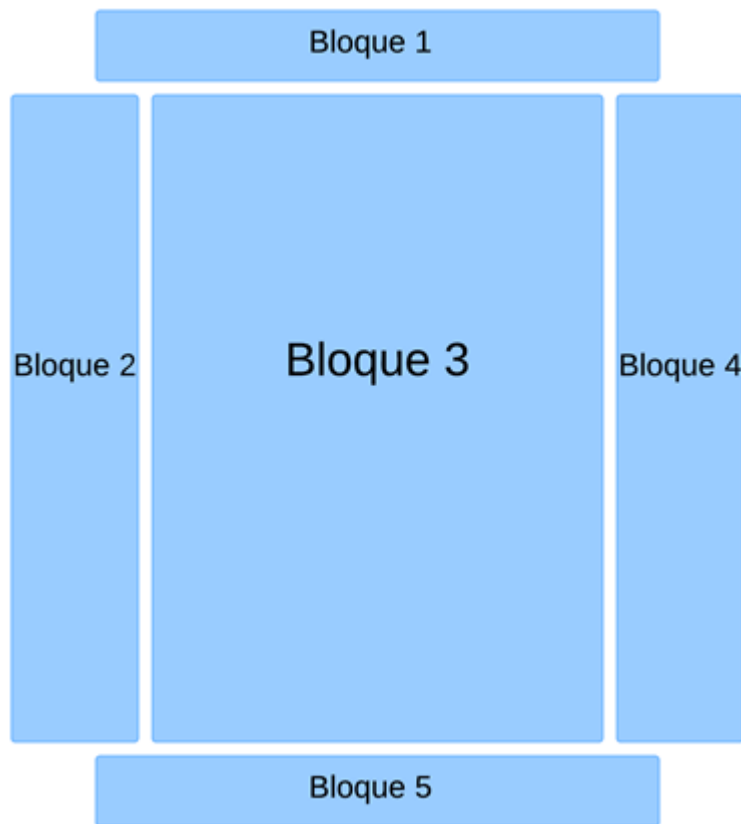


Figura 4.1.8.16 Prototipo para formato de pantallas del Front-End

Ver en la figura 4.1.8.17 y 4.1.8.18 los prototipos que se generaron directamente con las herramientas de desarrollo, en este caso con el lenguaje PHP, HTML, CSS, Javascript y AJAX

- Sistema de Comprobantes Fiscales Digitales por Internet -

Acciones

Limpiar Campos

Concepto

Identificador:

Descripcion:

Cantidad:

Unidad:

Valor Unitario:

Por Identificador

Receptor

Nombre:

RFC:

RFCE:

Cuenta:

Telefono:

Fax:

E-Mail:

Notificaciones

Para iniciar debes
agregar al menos un
producto

Figura 4.1.8.17 Prototipo de Pantalla 1 del Front-End del Sistema SCFDI

Estado:

Localidad:

Pais:

Autocompletado

Factura

Forma de pago:

Cond de pago:

Descuento:

Motivo Descuento:

Metodo de pago:

Comprobante:

Moneda

Tipo de Moneda:

Tipo de Cambio:

Figura 4.1.8.18 Prototipo de Pantalla 2 del Front-End del Sistema SCFDI

Modelo de Implementación del Sistema

El objetivo principal del modelo de implementación del sistema es mostrar cómo se asignarán la parte automatizada del modelo de implementación del usuario a las piezas principales de hardware y software del sistema, y cómo será la comunicación entre procesadores (CPUs).

Existen algunas opciones:

- Se asignan y distribuyen todas las tareas esenciales procesador por procesador.
- Cada proceso del nivel 0, se asigna a un procesador.
- Se puede escoger una combinación de computadoras principales, o mini y micro computadoras.

Es importante mencionar que en la implementación que involucre más de un procesador involucrará también algún mecanismo de comunicación entre procesadores.

Las herramientas que se usan en este modelo son: diagrama de flujo de datos para generar el modelo de tareas.

Modelo de Tareas

La noción de tareas es común casi en cualquier marca de hardware de computadora, aunque la terminología difiera de un proveedor a otro, algunos usan el término partición y otros usan el término punto de control. Se utiliza la herramienta del DFD, ya que el modelo de tareas utiliza los mismos elementos del DFD.

Es casi universal que la comunicación entre tareas se realiza a través del sistema operativo, por lo que en esta parte, también se definirá el sistema operativo de cada procesador.

Piezas de Hardware del sistema SCFDI

Procesador 1

Máquina para capturar compra:

PC

Intel Dual Core ™

1.6 GHz

Memoria 1 GB

Disco Duro 250 GB

Procesador 2

Máquina para generar la Factura Electrónica en forma automática

PC

Pentium 4

2.8 GHz

Memoria 8 GB

Disco Duro 500 GB

El sistema GCFID se puede montar en una arquitectura cliente/servidor, a través de una red LAN, con las siguientes características:

- Ancho de Banda 100 MBPS
- Hub para conexión de 10 nodos
- Protocolo TCP/IP

Piezas de Software del Sistema SCFDI

Procesador 1

Windows 7 home

Navegador Google Chrome con JavaScript Activado

Procesador 2

GNU/Linux Ubuntu 12.04

Servidor Web Apache versión 2

PHP versión 5

Java versión 7

MySQL versión 5

Aunque es posible hacer la implementación en una sola máquina que posea las características del procesador 2

En el modelo de implementación del sistema también se deben definir las interfaces entre procesadores, para el sistema SCFDI la interfaz entre el procesador 1 y procesador 2 será a través de una archivo .TXT, tal como se muestra en la figura 4.1.8.19.

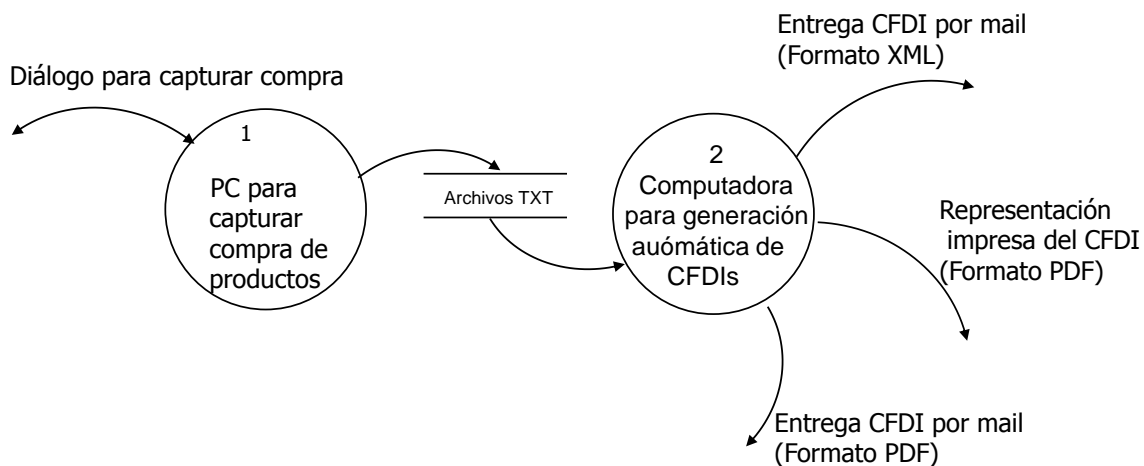


Figura 4.1.8.19 La interfaz entre el procesador 1 y procesador 2 será a través de archivos en formato TXT

Definición de interfaces del Sistema GCFI

Formato del Archivos TXT (es el insumo de entrada al proceso de generación automática de CFDIs)

El archivo .TXT con el que se genera el CFDI es un archivo de texto plano donde en cada línea se pone un valor que pertenece al comprobante en cuestión y con los que en conjunto se forma el CFDI. El formato es:

LLAVE=VALOR

Donde LLAVE es una sola palabra única entre el conjunto de llaves en el archivo y que identifica un campo y VALOR es el contenido de dicho campo. También se

puede incluir comentarios siempre que estos inicien con el símbolo de número (#), el orden en que aparecen estos campos en el documento no es importante sin embargo es importante que de ser marcados como requeridos estos aparezcan y tengan un valor asignado. Vea tabla 4.1.8.12.

LLAVE	CONTENIDO	TIPO
CONCEPTOS		
CANTIDAD_XXX	Cantidad	entero
UNIDAD_XXX	Unidad	cadena
NUMIDE_XXX	Número de identificación	cadena
DESCRIPCION_XXX	Descripción	cadena
VALUNI_XXX	Valor unitario	importe
IMPORTE_XXX	Importe	importe
DOMICILIO		
CALLEX	Calle	cadena
NUMEROX	Número	cadena
COLONIA	Colonia	cadena
LOCALIDADX	Localidad	cadena
LUGARX	Lugar	cadena
MUNICIPIOX	Municipio	cadena
ESTADOX	Estado	cadena
PAISX	País	cadena
CPX	Código postal	cadena
CIUDADX	Ciudad	cadena
TELEFONOX	Teléfono	cadena
FAXX	Fax	cadena
EMAILEX	Email de envío	cadena
EMAILCX	Email de contacto	cadena
EMISOR		
RFCE	RFC	cadena
NOMBREE	Nombre o razón social	cadena
COMPROBANTE		
SERFAC	Serie del comprobante	cadena
FORPAG	Forma de pago	cadena
CONPAG	Condiciones de pago	cadena
SUBTOTAL	Subtotal	importe
DESCUENTO	Descuento	importe
MOTIVO	Motivo del descuento	cadena

TOTAL	Total del documento	importe
METPAG	Método de pago	cadena
CTAMETPAG	Cuenta de pago	cadena
DOCUMENTO	Tipo de comprobante	cadena
TIPPAG	Tipo de pago	cadena
TTRASLADO	Total de impuestos trasladados	importe
TRETENIDO	Total de impuestos retenidos	importe
MONEDA		
MONEDA	Variable que guarda la moneda en la que están expresados todos los valores	cadena
TIPOCAMBIO	Tipo de cambio para las monedas extranjeras	importe
SUFIJO	El sufijo a imprimir posterior a el total expresado con letras	cadena
CARACTER	Código decimal ASCII del carácter de la moneda	entero
TOTALMN	Total en moneda nacional en caso de ser montos expresados en moneda extranjera	importe
RECEPTOR		
RFCR	RFC del receptor	cadena
NOMBRER	Nombre del receptor	cadena
REGIMEN		
REGIMEN	Régimen en el cuál tributa el emisor	cadena
IMPUESTOS RETENIDOS ⁱ		
IMPUESTO_XX	Nombre del impuesto retenido	cadena
IMPORTE_XX	Importe del impuesto retenido	importe
IMPUESTOS TRASLADADOS ⁱⁱ		
TRASIMP_XX	Nombre del impuesto trasladado	cadena
TRASNTAS_XX	Tasa del impuesto trasladado	importe
TRASIMP_XX	Importe del impuesto trasladado	importe

Tabla 4.1.8.12 Formato del Archivos TXT (es el insumo de entrada al proceso de generación automática de CFDI's)

Modelo de Implementación del Programa

El modelo de implementación del programa se genera a partir del modelo de implementación del sistema adicionando detalles como:

- Asignación de actividades de tareas a módulos
- Organización jerárquica de módulos
- Interfaces entre módulos

En este modelo se utiliza las herramientas: diagrama de estructuras, especificación de módulos, y en caso de ser necesario se complementa el diccionario de datos.

Diagrama de estructuras

El objetivo del diagrama de estructuras es modelar los agrupamientos de instrucciones que hacen una tarea, las conexiones de datos y control entre los módulos y la organización de los módulos.

El modelo de implementación del programa se genera a partir del Modelo de Implementación del Sistema adicionando detalles como la asignación de actividades de tareas a módulos.

Para una buena organización de código deberíamos:

- Mantener la parte esencial del sistema de la implementación.
- Separar los controles de las tareas esenciales.
- Procurar una organización descendente de las funciones de las tareas.

La cohesión es una medida de la fuerza de asociación de las instrucciones, lo ideal es que un módulo tenga **alta cohesión**, los módulos con una fuerte relación entre sus componentes:

- Facilita el mantenimiento

- Son más fáciles de entender
- Es más factible que sean reusables.

El acoplamiento es una medida de la interdependencia entre módulos, lo ideal es que un sistema tenga un **bajo acoplamiento**, ya que un acoplamiento excesivo:

- Hace más difícil de entender una módulo sin observar los otros
- Incrementa el número de módulos que serán afectados por un cambio en un campo (item) del dato, y
- Incrementa la probabilidad de que un módulo se vea afectado por el cambio en un campo (item) del dato.

En la figura 4.1.8.20, se muestra del diagrama de estructuras del sistema SCFDI del programa del front-end del sistema SCFDI.

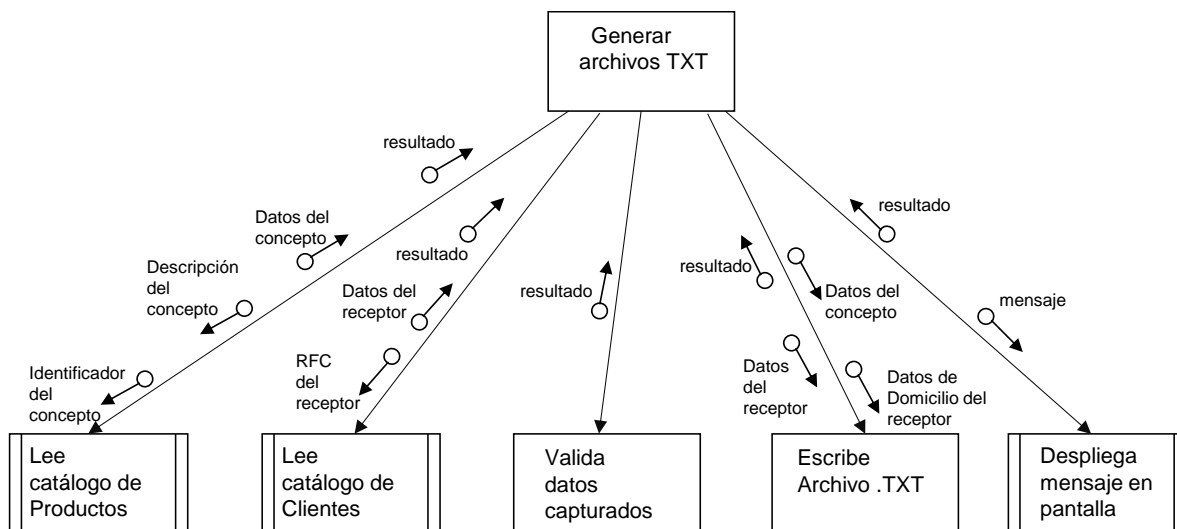


Figura 4.1.8.20 Diagrama de estructuras del programa del front-end del sistema SCFDI

En las figuras 4.1.8.21, 4.1.8.22, 4.1.8.23 y 4.1.24 se muestra el diagrama de estructuras del programa del back-end del sistema SCFDI.

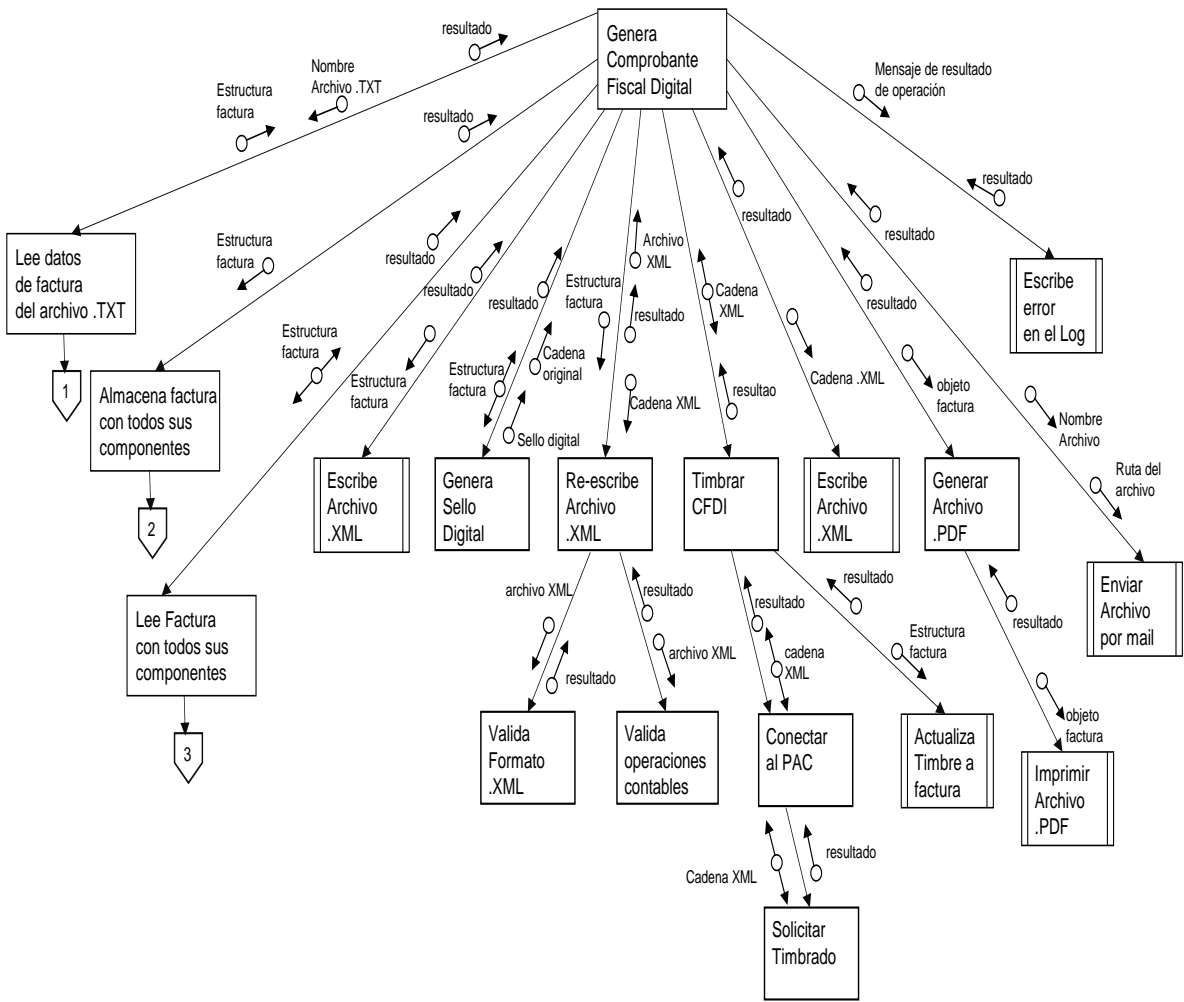


Figura 4.1.8.22 Diagrama de estructuras del módulo lee datos de factura del archivo .TXT

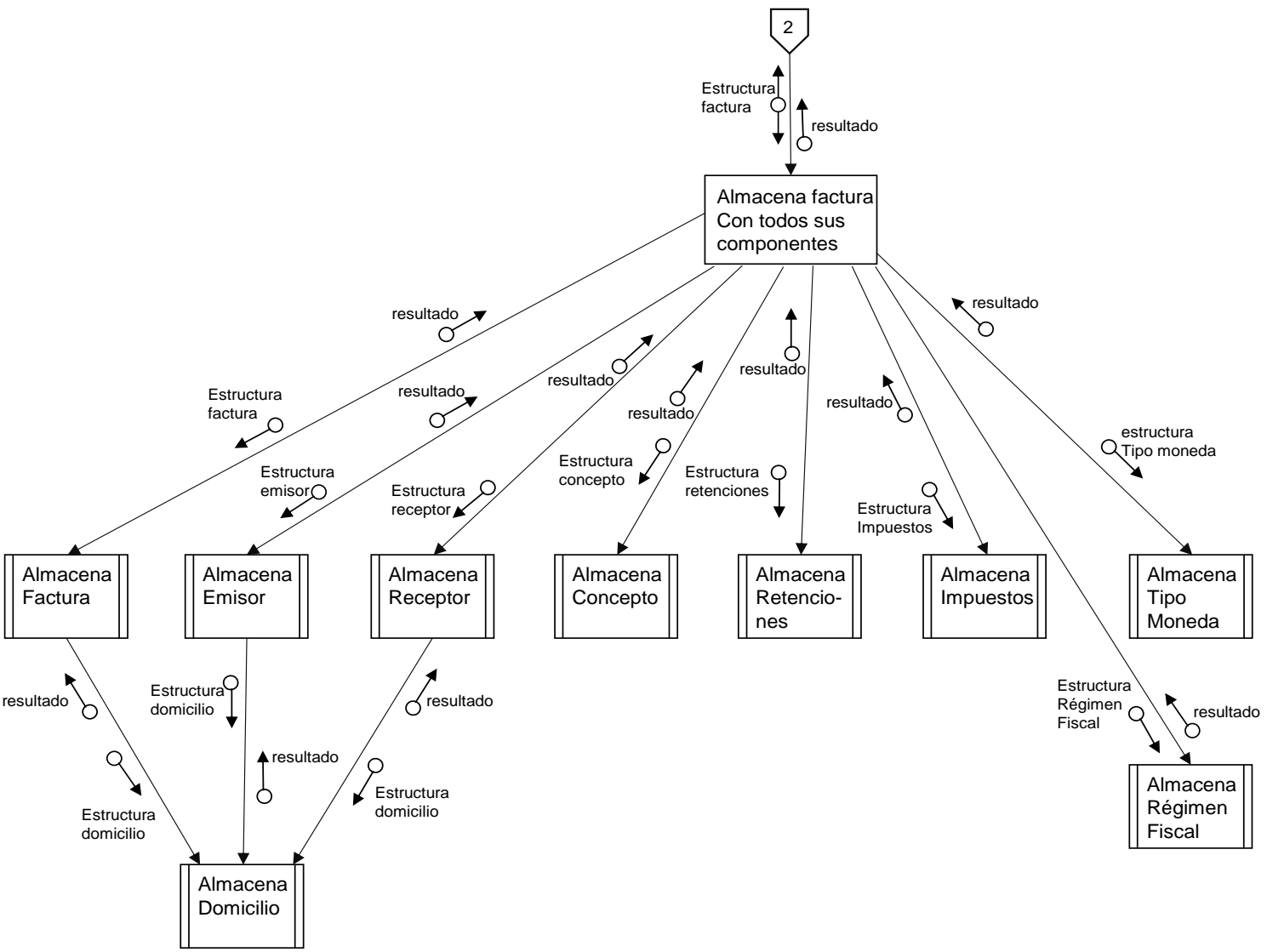
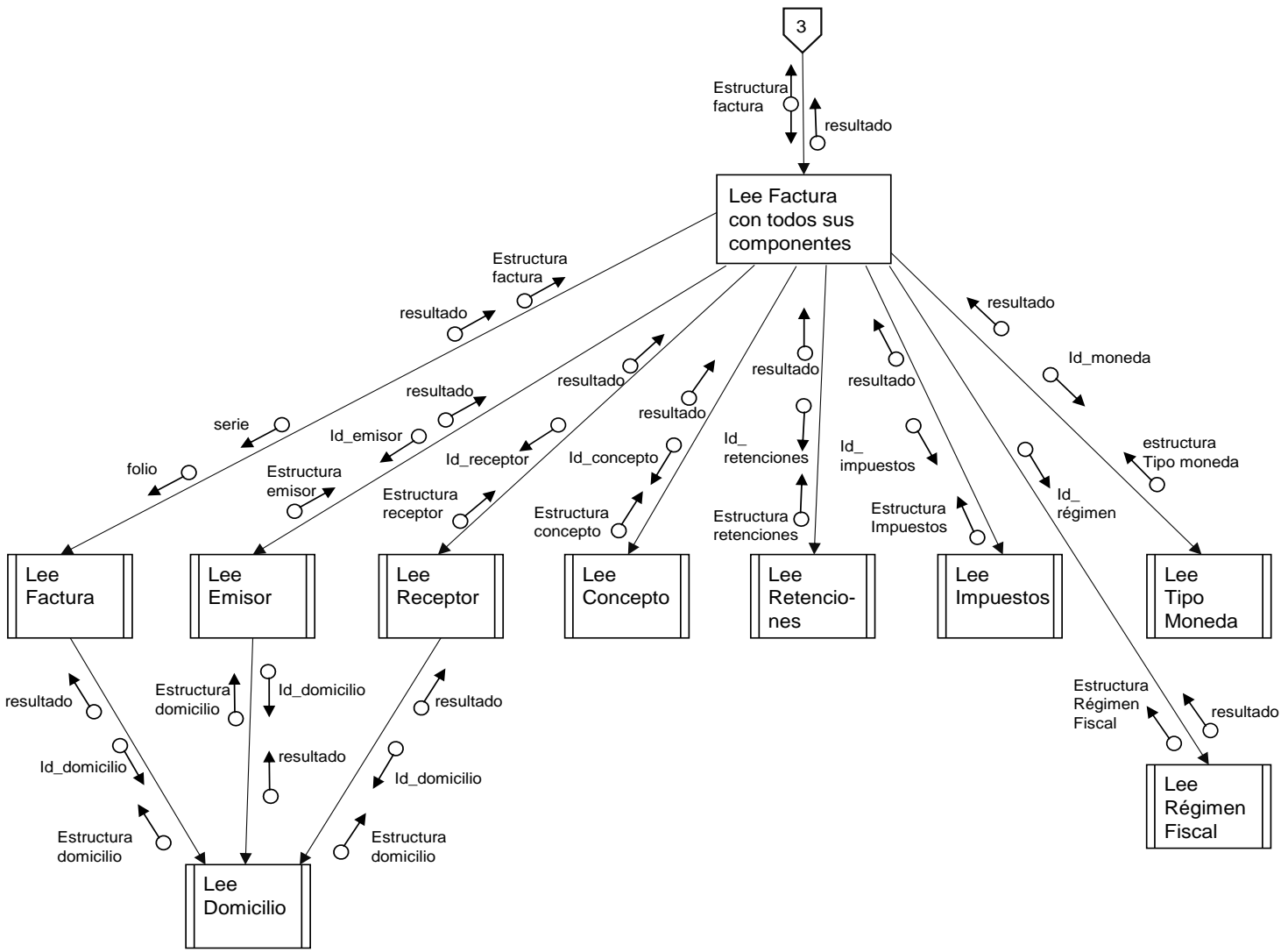


Figura 4.1.8.23 Diagrama de estructuras del módulo almacena factura con todos sus componentes

Figura 4.1.8.24 Módulo lee factura con todos sus componentes.



Especificación de módulos del sistema SCFDI

El siguiente paso después de haber generado el diagrama de estructuras es documentar cada módulo. Existen varias opciones para documentar los procesos, en este caso, utilizaremos la técnica narrativa.

Especificación del módulo generar archivos TXT

Este módulo permitirá generar el archivo TXT que contendrá los datos necesarios para la generación de un CFDI.

Si el usuario selecciona el modo de autocompletado de datos del concepto, se llama al módulo: **lee catálogo de productos**, y se le pasa como parámetros el Identificador del concepto y descripción del concepto (aunque sólo es necesario capturar cualquiera de los dos y sobre uno de ellos se hace la búsqueda).

Y se reciben los siguientes datos:

- Identificador del concepto
- Descripción del concepto
- Unidad
- Valor unitario

Si el usuario selecciona el modo de capturar datos del concepto, se habilitan todas las cajas de texto del bloque concepto para que el usuario pueda capturar todos los datos del concepto:

- Identificador del concepto
- Descripción del concepto
- Cantidad
- Unidad
- Valor unitario

Si el usuario selecciona el modo de autocompletado de datos del receptor, se llama al módulo: **lee catálogo de clientes**, y se le pasa como parámetro el RFC del receptor, y se reciben los siguientes datos:

- Nombre

-
- RFCE
 - Cuenta
 - Teléfono
 - Fax
 - e-mail
 - Calle
 - Número Interior
 - Número Exterior
 - Colonia
 - CP
 - Estado
 - Localidad.- Alfanumérico
 - Fax.- Solo números
 - Teléfono.- Solo números

Si el usuario selecciona el modo de capturar datos del receptor, se habilitan todas las cajas de texto del bloque de datos del receptor para que el usuario pueda capturar todos los datos del receptor.

Se valida que se hayan capturado los campos obligatorios haciendo un llamado al módulo: **valida datos capturados**, y si son correctos manda a llamar al módulo: **escribe archivo TXT**, en caso contrario envía un mensaje al usuario para verifique los datos capturados, haciendo un llamado al módulo: **despliega mensaje en pantalla** con el mensaje de error que se haya encontrado.

Cuando todos los campos fueron correctos y se genera el archivo TXT, se limpian los datos de pantalla para continuar con una nueva captura para la generación de un CFDI.

Especificación del módulo lee catálogo de productos

Si se recibe el identificador de producto, consulta el catálogo de productos para recuperar el registro que coincida con el Identificador de producto introducido, y se recuperan los siguientes datos y se regresan al módulo

que hizo el llamado:

- Descripción
- Unidad
- Valor unitario

Si se recibe la descripción de producto, consulta el catálogo de productos para recuperar el registro que coincida con la descripción del producto introducido, y se recuperan los siguientes datos y se regresan al módulo que hizo el llamado:

- Id
- Unidad
- Valor unitario

Especificación del módulo lee catálogo de clientes

Si se recibe el RFC del receptor, se consulta el catálogo de clientes para recuperar el registro que coincida con el Identificador del RFC introducido, y se regresan los siguientes datos al módulo que hizo el llamado:

- Nombre
- RFC
- RFCE
- Cuenta
- Telefono
- Fax
- e-mail
- Calle
- Número Interior
- Número Exterior
- Colonia
- CP
- Estado
- Localidad

-
- Fax
 - Teléfono

Especificación del módulo valida datos capturados

Este módulo verifica que se hayan capturado todos los datos en forma correcta y que todos los campos obligatorios tengan algún valor:

- Id producto.- Números y letras (obligatorio)
- Descripción del producto.- Alfanumérico (obligatorio)
- Valor unitario del producto.- Números (obligatorio)
- Cantidad del producto.- Números (obligatorio)
- Nombre.- Solo letras y espacios (obligatorio)
- RFC .- Números y letras (obligatorio)
- RFCE.- Números y letras
- Cuenta.- Números y letras
- Email.- Caracteres alfanuméricos, incluye: Puntos, guiones y arroba.
- Calle.- Alfanumérico
- Interior.- Números
- Exterior.- Números
- Colonia.- Letras
- CP.- Números
- Estado.- Letras
- Localidad.- Letras
- País.- Letras (obligatorio)

Dentro del bloque factura se solicitan los siguientes campos:

- Forma de pago.- Debe tener un valor seleccionado de la lista (Una exhibición, Dos exhibiciones)
- Condición de pago.- Debe tener un valor seleccionado de la lista (Ninguna, 30 días, 60 días y 90 días).
- Descuento.- Puede ir vacío, aunque es numérico.
- Motivo descuento.- Es alfanumérico.

-
- Método de pago.- Debe tener un valor seleccionado de la lista (Efectivo, Tarjeta de crédito, Tarjeta de Débito, Tarjeta de servicio, Transferencia electrónica, Monedero electrónico, Cheque nominativo)
 - Tipo comprobante.- Debe tener un valor seleccionado de la lista (Ingreso, Egreso, Traslado)
 - Tipo de moneda.- Debe tener algún valor seleccionado de la lista (Peso, Dólar, Euro)
 - Tipo de cambio.- Numérico (Obligatorio)

Si todos los datos cumplen con el formato y valores establecidos, se regresa una respuesta exitosa, en caso contrario se regresa error al módulo que hizo el llamado.

Especificación del módulo despliega mensaje en pantalla

Este módulo permite desplegar un mensaje en pantalla, recibe la cadena que se debe mostrar al usuario, y espera a que el usuario oprima el botón [ACEPTAR] para eliminar el mensaje y regresar al usuario a modo captura.

Especificación del módulo escribe archivo .TXT

Obtiene de las cajas de texto todos los datos introducidos por el usuario, tanto del bloque de Concepto, del Receptor y de Factura, y escribe los datos en el archivo de Salida, en el formato indicado en la tabla 4.1.8.12 Formato del Archivos TXT (que es el insumo de entrada al proceso de generación automática de CFDIs).

Especificación del módulo lee datos de factura en el archivo .TXT

Valida que el archivo tenga las variables serie

Si no tiene serie, regresa error al módulo que hizo el llamado, y mueve el archivo de solicitud de CFDI a un directorio de errores de la máquina en donde se está ejecutando la aplicación.

Si sólo trae los campos serie y folio, indica que es un archivo para cancelar la factura con esa serie y ese folio.

Si trae serie, folio y otros campos, entonces es un archivo para generar un CFDI y se busca campos conocidos, y manda a llamar a los módulos: **lee atributos de factura**, **lee atributos de receptor**, **establece atributos de concepto**, **establece atributos de retención**, **establece atributos de impuestos** y al módulo **establece certificado de sello digital**.

Regresa el resultado de la operación al módulo que hizo el llamado.

Especificación del módulo escribe error en el log

Recibe un mensaje de error como entrada, y utiliza la librería (loger), la cual escribe un registro con el error reportado, en el archivo de errores de la aplicación.

Especificación del módulo lee atributos de factura

Mapea el archivo de texto, hacia un archivo de propiedades, en este proceso se deben leer los siguientes datos:

- Nombre del archivo
- Serie
- Forma de pago
- Condiciones de pago
- Subtotal
- Descuento
- Motivo del descuento
- Total
- Método de pago
- Cuenta del método de pago
- Tipo de comprobante
- Tipo de pago
- Total de Impuesto trasladado
- Total de impuesto retenido

Debe completar los datos de la factura y hacer el llamado a los siguientes módulos en el siguiente orden:

-
- Módulo establece atributos de factura
 - Módulo establece atributos de emisor
 - Módulo establece atributos del domicilio del emisor
 - Módulo establece atributos del lugar de emisión
 - Módulo establece atributos de tipo moneda

Y finalmente debe ligar el objeto emisor al objeto factura.

Regresa el resultado de la operación al módulo que hizo el llamado.

Especificación del módulo establece atributos de factura

Cuando se reciben los datos de la Factura se debe llenar el objeto Factura con los datos raíz:

- Serie
- Forma de pago
- Condiciones de pago
- Subtotal
- Descuento
- Motivo del descuento
- Total
- Método del pago
- Cuenta del método del pago
- Tipo de comprobante
- Total del Impuesto trasladado
- Total del Impuesto retenido

Regresa el resultado de la operación al módulo que hizo el llamado.

Especificación del módulo establecer atributos de emisor

Se agregan dos atributos al objeto emisor:

- Razón social
- RFC

Regresa el resultado de la operación al módulo que hizo el llamado.

Especificación del módulo establece atributos del domicilio del emisor

Limpiar el objeto atributos de factura (se deja en blanco la llave valor) y luego se leen datos obtenidos del archivo inicial, los datos que se leen del archivo Emisor son:

- Calle
- Número exterior
- Colonia
- Localidad
- Municipio
- Estado
- País
- Código postal
- Ciudad
- Teléfono
- Fax

Se debe crear un objeto domicilio y se deben pasar los datos que se leyeron y establecer dichos datos dentro del objeto emisor

En este punto el objeto emisor ya debe contar con todos los datos de su domicilio, que se pasan como un arreglo de datos encapsulado.

Regresa el resultado de la operación al módulo que hizo el llamado.

Especificación del módulo Establece Atributos del lugar de emisión

Se deben limpiar los atributos objeto factura, y se llenan con algunos datos provenientes del archivo original:

- Calle
- Número exterior
- Colonia
- Localidad
- Municipio
- Estado
- País

-
- Código postal
 - Ciudad
 - Teléfono
 - Fax

Se debe crear un nuevo objeto de tipo domicilio y se le deben pasar esas variables, y una vez que estén llenas esas variables, se le debe pasar el arreglo de datos en forma encapsulada el domicilio del lugar de emisión al objeto emisor. Regresa el resultado de la operación al módulo que hizo el llamado.

Especificación del módulo establece atributos de tipo moneda

Se deben limpiar atributos del objeto tipo moneda.

Se deben leer otros datos que son:

- Documento
- Tipo de cambio
- Total en moneda nacional
- Clave de la moneda que se leyó del archivo original

El objeto tipo moneda se le pasa al objeto factura. Regresa el resultado de la operación al módulo que hizo el llamado.

Especificación del módulo lee atributos del receptor

Se debe limpiar el objeto receptor, y se hace un llamado al módulo

Establece atributos de receptor, debe recibir los siguientes datos:

- Nombre
- RFC
- RFCE
- Cuenta

Se debe crear un objeto domicilio receptor y se establecen los datos del domicilio del receptor, haciendo un llamado al módulo **establece atributos del domicilio de receptor**, se debe regresar el resultado de la operación al módulo que hizo el llamado.

Especificación del módulo establece atributos del domicilio del receptor

Se debe limpiar el objeto domicilio receptor, y se debe actualizar los valores de los siguientes campos:

- Calle
- Número exterior
- Colonia
- Localidad
- Municipio
- Estado
- País
- Código postal
- Ciudad
- Teléfono
- Fax
- email

Se regresa el objeto domicilio receptor y el resultado de la operación.

Especificación del módulo establece atributos del receptor

Se debe limpiar el objeto receptor, y se le deben pasar los siguientes datos del Receptor:

- RFC
- Razón social
- RFC del extranjero

Si el resultado fue exitoso, se regresa el objeto receptor y se regresa respuesta exitosa al módulo que hizo el llamado, en caso contrario, se regresará error.

Especificación del módulo establece atributos de concepto

Es una repetición de pasos, de acuerdo al número de conceptos, y se ejecutan de la siguiente manera:

Se crea índice para leer los conceptos, con ese número de índice se van etiquetando todos los datos de un concepto en forma iterativa tantas veces como conceptos existan:

- Cantidad
- Unidad
- Número de identificación
- Descripción
- Valor unitario
- Tasa del impuesto trasladado

En este proceso se crea un objeto de arreglo de conceptos que se forma de tantos conceptos como compras haya realizado el cliente.

Y este objeto de conceptos se pasa al objeto factura.

Especificación del módulo establece atributos de retenciones

Primero se limpia el objeto atributos factura, se crea un objeto retenciones, el cual es un arreglo de retención de impuesto, es el cálculo de la suma total de los diferentes impuestos (IVA o ISR).

En forma iterativa se agrega al objeto retenciones, una instancia de retención la cual incluye la siguiente información:

- -Descripción del impuesto
- Importe

Es decir, el objeto de retenciones está compuesto por dos instancias, uno para el impuesto de IVA retenido y el otro para impuesto ISR retenido.

Cuando ya se tiene el conjunto de retenciones, este objeto retenciones se agrega como un nodo más del objeto factura.

Especificación del módulo establece atributos de Impuestos trasladados

Se crea un objeto impuestos trasladados, el cual es un arreglo de impuestos trasladados, que está compuesto por las diferentes tasas a las que se encuentran grabados los diferentes conceptos (productos) que

adquirió el cliente.

En forma iterativa se agrega al arreglo de impuestos trasladados, una instancia de impuesto trasladado la cual incluye la siguiente información:

- Descripción del impuesto
- Importe
- Tasa

Esto se hace tantas veces como diferentes tasas se hayan obtenido en los pasos iniciales en este proceso.

Cuando ya se tiene el conjunto de impuestos trasladados, este objeto impuestos trasladados se agrega como un nodo más del objeto factura.

Especificación del módulo establece certificado del sello digital

Para poder emitir facturas electrónicas los contribuyentes deben contar con un certificado de sello digital, su llave (archivo .key) y la contraseña de la llave que les proporcionó el SAT.

El nombre de los dos archivos .cer y .key, así como el password se toman de un archivo de configuración, haciendo un llamado al módulo **Lee archivo de configuraciones**, proporcionando el nombre de la variable que debe leer para cada dato que se requiere.

Con estos insumos, lo que debe hacer el proceso primero es des-encryptar el Certificado del Sello digital (archivo .cer) utilizando la llave y la contraseña de la llave.

Una vez des-encryptado el archivo .cer se debe tomar su información y se pasa a formato base 64 y se agrega al objeto factura en la variable Certificado.

Y el número de Serie del Certificado también se agrega a la factura como número de Certificado, este dato también se toma del archivo .cer mencionado arriba.

Se hace un llamado al módulo **obtiene folio disponible**, se recibe el número de folio disponible y asigna a la factura el número de folio obtenido. Regresa el resultado de la operación al módulo que hizo el llamado.

Especificación del módulo lee archivo de configuraciones

Recibe el nombre de la variable que se debe buscar en el archivo de configuración (config.xml) que está ubicado en la misma ruta donde corre la aplicación principal, y regresa el valor que encontró en el archivo de configuraciones, en caso de presentarse algún problema regresa error al módulo que hizo el llamado.

Especificación del módulo obtiene folio disponible

Consulta la tabla de facturas, para saber cuál es el siguiente número de folio disponible, en caso de que no haya ningún registro en la tabla se asigna el folio 1, si encuentra datos, obtiene el último número de folio asignado y le se suma un 1 a ese folio devuelto y se regresa el número de folio al módulo que hizo el llamado, o regresa error en caso de que hubiera fallado la consulta.

Especificación del módulo almacena factura con todos sus componentes

Se deberán grabar todos los componentes de la factura, para ellos se hace un llamado a los módulos:

- Módulo almacena factura
- Módulo almacena emisor
- Módulo almacena receptor
- Módulo almacena concepto
- Módulo almacena retenciones
- Módulo almacena impuestos trasladados
- Módulo almacena régimen fiscal
- Módulo almacena tipo moneda

Se crea un objeto llave PK que está compuesta por serie y folio, y con esa llave PK se consulta la tabla t_factura.

Y se manda llamar el módulo **Escribe archivo XML**.

Si la transacción fue exitosa regresa resultado exitoso al módulo que hizo el llamado, en caso contrario, regresará el error encontrado.

Especificación del módulo almacena factura

Recibe una estructura factura (que deberá incluir un nodo hijo con la información de lugar de emisión), y graba en la tabla t_factura un registro con todos los campos de factura:

- serie
- folio
- version
- sello
- cadenaOriginal
- noCertificado
- certificado
- formaDePago
- condicionesDePago
- subtotal
- descuento
- motivoDescuento
- total
- metodoDePago
- ctaMedotoDePago
- ctaMetodoDePago
- vigente
- tipoDeComprobante
- f_moneda
- nombreArchivo
- tipoPago
- totalTrasladado
- totalRetenido

-
- f_receptor
 - f_emisor
 - f_timbre
 - f_lugarEmision

Para almacenar los datos del domicilio de lugar de emisión de la factura en la tabla t_domicilo, hace un llamado al módulo **almacena domicilio** y se le pasa como parámetro la estructura del lugar de emisión.

Si la transacción fue exitosa regresa resultado exitoso al módulo que hizo el llamado, en caso contrario, regresará el error encontrado.

Especificación del módulo almacena emisor

Recibe una estructura emisor (que deberá incluir un nodo hijo que contiene el domicilio), identifica los datos del emisor y graba en la tabla t_emisor un registro con todos los campos:

- id
- rfc
- nombre
- f_domicilio
- f_regimenFiscal

Para almacenar los datos del domicilio del emisor en la tabla t_domicilo, hace un llamado al módulo **almacena domicilio** y se le pasa como parámetro la estructura de domicilio del emisor.

Si la transacción fue exitosa regresa resultado exitoso al módulo que hizo el llamado, en caso contrario, regresará el error encontrado.

Especificación del módulo almacena receptor

Se recibe una estructura receptor (incluye un nodo con la información del domicilio del receptor), y se graba en la tabla t_receptor un registro con todos los campos:

- id
- rfc

-
- nombre
 - f_domicilioFiscal
 - f_regimenFiscal

Para almacenar los datos del domicilio del receptor en la tabla t_domicilo, hace un llamado al módulo **almacena domicilio** y se le pasa como parámetro la estructura de domicilio del receptor.

Si la transacción fue exitosa regresa resultado exitoso al módulo que hizo el llamado, en caso contrario, regresará el error encontrado.

Especificación del módulo almacena concepto

Se recibe una estructura concepto y se graba en la tabla t_concepto un registro con todos los campos:

- cantidad
- unidad
- numeroDeIdentificacion
- descripcion
- valorUnitario
- importe
- f_serie
- f_folio

Si la transacción fue exitosa regresa resultado exitoso al módulo que hizo el llamado, en caso contrario, regresará el error encontrado.

Especificación del módulo almacena retenciones

Se recibe una estructura concepto y se graba en la tabla t_retencion un registro con todos los campos:

- id
- impuesto
- importe
- f_serie
- f_folio

Si la transacción fue exitosa regresa resultado exitoso al módulo que hizo el llamado, en caso contrario, regresará el error encontrado.

Especificación del módulo almacena traslados

Recibe una estructura traslado y graba en la tabla t_translado, un nuevo registro con todos los campos:

- id
- impuesto
- tasa
- importe
- f_serie
- f_folio

Si la transacción fue exitosa regresa resultado exitoso al módulo que hizo el llamado, en caso contrario, regresará el error encontrado.

Especificación del módulo almacena régimen fiscal

Recibe una estructura RegimenFiscal y graba en la tabla t_regimenfiscal, un nuevo registro con todos los campos:

- Id
- descripcion

Si la transacción fue exitosa regresa resultado exitoso al módulo que hizo el llamado, en caso contrario, regresará el error encontrado.

Especificación del módulo almacena tipo moneda

Recibe una estructura TipoMoneda y graba en la tabla t_moneda un nuevo registro con todos los campos:

- id
- moneda
- tipo_cambio
- documento
- sufijo

-
- caracter
 - totalMN

Si la transacción fue exitosa regresa resultado exitoso al módulo que hizo el llamado, en caso contrario, regresará el error encontrado.

Especificación del módulo almacena domicilio

Recibe una estructura domicilio y graba en la tabla t_domicilio un nuevo registro con todos los campos:

- id
- exterior
- interior
- colonia
- localidad
- referencia
- municipio
- estado
- pais
- codigoPostal
- ciudad
- fax
- emailEnvio
- emailContacto

Si la transacción fue exitosa regresa resultado exitoso al módulo que hizo el llamado, en caso contrario, regresará el error encontrado.

Especificación del módulo escribe archivo .XML

Se hace la consulta de base de datos y se regenera todo el objeto factura con todos sus nodos y propiedades, para esto tiene que hacer una consulta de todas las tablas que contienen la información de la factura (t_moneda, t_factura, t_emisor, t_retencion, t_domicilio, t_transalado, t_concepto, t_regimen)

Este proceso se auxilia de un objeto “document” que pertenece al estándar w3c, diseñado para elaborar archivos en formato .XML en forma muy sencilla, en donde se le pasan los diferentes nodos y atributos.

El primer nodo que se crea es el nodo de comprobante con sus respectivos subnodos: emisor, receptor, conceptos e impuestos.

Y a su vez en el nodo emisor se le pasan sus subnodos: domicilio fiscal y expedido en.

Así también para el nodo receptor se pasa el subnodo: domicilio, para el nodo conceptos se pasan toda la serie de conceptos, en el nodo Impuestos se le pasan los subnodos de retenciones y traslados como se muestra a continuación:

- Comprobante
 - Emisor ----Domicilio Fiscal
 ----Expedido En
 - Receptor ---- Domicilio
 - Conceptos-----Concepto_1
 -----Concepto_n
 - Impuestos --- Retenciones---Retencion_IVA
 ----Retencion_ISR
 ---Traslados ----Tasa_1
 Tasa_N

Se obtiene la fecha de la factura y con esa información la pasa a formato B31 con una función matemática, y con esta información se armara la ruta completa donde se almacenara el archivo .XML generado y el nombre físico de dicho archivo, como sigue: FACTURAS/rfc_receptor/fechaB31-SerieyFolio.XML

El directorio FACTURAS está a la misma altura donde se almacena la aplicación.

Especificación del módulo genera sello digital

Se crea un objeto de tipo StringWriter que es una biblioteca de Java que sirve para escribir archivos de texto plano, con la posibilidad de que el archivo que lee nos lo regresa como una cadena de texto, o como un arreglo de bytes, para este caso se utiliza la opción de cadena de texto.

Se crea otro objeto tipo llave PK con el cual se vuelve a llamar el proceso **escribe archivo .XML**, a través del cual se obtiene un objeto "DOM" que es un tipo de dato auxiliar representación del .XML

Se crea otro objeto del tipo "TRANSFORMER" que es de la librería Java XML, a la cual se le pasa el objeto StringWriter y "DOM", y nos devuelve la Cadena Original del XML (ver en definiciones del SAT la definición de Cadena Original).

Esta información de la Cadena Original se le agrega al objeto Factura.

Se convierte en formato SHA-1 (algoritmo de encriptación que genera cadenas a 128 caracteres), posteriormente se encripta con el algoritmo RSA y el archivo .key (que es el juego de datos del emisor), con lo cual se obtiene una cadena binaria que se transforma a B64 y es el sello digital.

Una vez que se tiene el sello se agrega al objeto Factura

A través de Hibernate se utiliza la función update para actualizar la tabla factura con los datos de cadena original y sello digital.

Especificación del módulo re-escribe el archivo .XML

Genera el objeto llave PK (Serie y Folio), hace la consulta a la base de datos con la llave primaria y genera el objeto "DOM", reescribiendo el archivo .XML agregando el dato de sello digital.

Especificación del módulo lee factura con todos sus componentes

Se deberán leer todos los componentes de la factura, para ellos se hace un llamado a los módulos:

- Módulo lee factura
- Módulo lee emisor

-
- Módulo lee receptor
 - Módulo lee concepto
 - Módulo lee retenciones
 - Módulo lee impuestos trasladados
 - Módulo lee régimen fiscal
 - Módulo lee tipo moneda

Si la transacción fue exitosa regresa resultado exitoso al módulo que hizo el llamado, en caso contrario, regresará el error encontrado.

Especificación del módulo Lee Factura

Recibe la serie y el folio, y busca en la tabla t_factura un registro que coincida con esos datos, y recupera los campos:

- serie
- folio
- version
- sello
- cadenaOriginal
- noCertificado
- certificado
- formaDePago
- condicionesDePago
- subtotal
- descuento
- motivoDescuento
- total
- metodoDePago
- ctaMedotoDePago
- ctaMetodoDePago
- vigente
- tipoDeComprobante

-
- f_moneda
 - nombreArchivo
 - tipoPago
 - totalTrasladado
 - totalRetenido
 - f_receptor
 - f_emisor
 - f_timbre
 - f_lugarEmision

Con el campo f_lugarEmision del registro recuperado, hace un llamado al módulo **lee domicilio** y se le pasa como parámetro el id de domicilio (f_lugarEmision), para recuperar todos los datos del lugar de emisión.

Si la transacción fue exitosa regresa resultado exitoso al módulo que hizo el llamado, en caso contrario, regresará el error encontrado.

Especificación del módulo lee emisor

Recibe el identificador del emisor, y busca en la tabla t_emisor un registro que coincida con esa llave, y recupera los campos:

- id
- rfc
- nombre
- f_domicilio
- f_regimenFiscal

Con el campo f_domicilio del registro recuperado, hace un llamado al módulo **lee domicilio** y se le pasa como parámetro el id de domicilio (f_domicilio), para recuperar todos los datos del lugar del domicilio del emisor.

Si la transacción fue exitosa regresa resultado exitoso al módulo que hizo el llamado, en caso contrario, regresará el error encontrado.

Especificación del módulo almacena receptor

Recibe el identificador del receptor, y busca en la tabla t_receptor un registro que coincida con esa llave, y recupera los campos:

- id
- rfc
- nombre
- f_domicilioFiscal
- f_regimenFiscal

Con el campo f_domicilioFiscal del registro recuperado, hace un llamado al módulo **lee domicilio** y se le pasa como parámetro el id de domicilio (f_domicilioFiscal), para recuperar todos los datos del domicilio del Receptor.

Si la transacción fue exitosa regresa resultado exitoso al módulo que hizo el llamado, en caso contrario, regresará el error encontrado.

Especificación del módulo lee concepto

Recibe el numeroDeldentificacion del concepto, y busca en la tabla t_concepto un registro que coincida con esa llave, y recupera los campos:

- cantidad
- unidad
- numeroDeldentificacion
- descripcion
- valorUnitario
- importe
- f_serie
- f_folio

Si la transacción fue exitosa regresa resultado exitoso al módulo que hizo el llamado, en caso contrario, regresará el error encontrado.

Especificación del módulo lee retenciones

Recibe el id de retenciones, y busca en la tabla t_retencion un registro que

coincida con esa llave, y recupera los campos:

- id
- impuesto
- importe
- f_serie
- f_folio

Si la transacción fue exitosa regresa resultado exitoso al módulo que hizo el llamado, en caso contrario, regresará el error encontrado.

Especificación del módulo lee traslados

Recibe el id de traslado, y lee en la tabla t_traslado, el registro que coincida con esa llave, y obtiene los campos:

- id
- impuesto
- tasa
- importe
- f_serie
- f_folio

Si la transacción fue exitosa regresa resultado exitoso al módulo que hizo el llamado, en caso contrario, regresará el error encontrado.

Especificación del módulo lee régimen fiscal

Recibe una estructura RegimenFiscal y graba en la tabla t_regimenfiscal, un nuevo registro con todos los campos:

- Id
- descripcion

Si la transacción fue exitosa regresa resultado exitoso al módulo que hizo el llamado, en caso contrario, regresará el error encontrado.

Especificación del módulo lee tipo moneda

Recibe el id de moneda, y busca en la tabla t_moneda un registro que

coincida con esa llave, y recupera los campos:

- id
- moneda
- tipo_cambio
- documento
- sufijo
- caracter
- totalMN

Si la transacción fue exitosa regresa resultado exitoso al módulo que hizo el llamado, en caso contrario, regresará el error encontrado.

Especificación del módulo lee domicilio

Recibe el id de domicilio y lee en la tabla t_domicilio un nuevo registro que coincida con esa llave, y obtiene todos los campos:

- id
- exterior
- interior
- colonia
- localidad
- referencia
- municipio
- estado
- pais
- codigoPostal
- ciudad
- fax
- emailEnvio
- emailContacto

Si la transacción fue exitosa regresa resultado exitoso al módulo que hizo el llamado, en caso contrario, regresará el error encontrado.

Especificación del módulo valida formato XML

En este proceso se utiliza la función `validator` de la librería `Java.xml.validator`, en la cual se le pasa el XSD que es el esquema que publicó el SAT para CDFI y el archivo `.XML` generado, el cual contesta con un `TRUE` o `FALSE`, es decir, regresa el valor de `TRUE` cuando el `.XML` cumple con el formato, y `FALSE` cuando hay algún error.

Especificación del módulo Valida Operaciones Contables

Abre el archivo `.XML` y hace las siguientes validaciones:

- Verifica que el importe sea el valor unitario X cantidad en cada concepto.
- Verifica que la suma de todos los importes de los conceptos sea igual al subtotal
- Que la suma de las retenciones sea igual a la suma de Impuestos Retenidos
- Que la suma de los impuestos trasladados sea igual al total de impuestos trasladados
- Que el subtotal – descuentos + impuestos trasladados– impuestos retenidos = total

Si el valor fue `TRUE`, entonces se manda a llamar el módulo para timbrar CDFI.

Especificación del módulo Timbrar CDFI

Se lee el archivo `.XML`, y se convierte a un arreglo de bytes, y se utiliza la especificación de las librerías que proporciona el PAC. Con esas especificaciones se generan unas librerías locales en lenguaje Java, que permite hacer la conexión al PAC y enviarle la cadena XML y los datos de autenticación (`USER`, `PASSWORD`), para lo cual se hace un llamado al módulo **conectar al PAC** y se pasan ambos datos como parámetros, y también se le pasa la cadena XML.

Actualiza el objeto factura, agregando el nodo timbrado, y graba a la tabla t_timbrado y en la tabla t_factura, y reescribe el archivo .XML final en el directorio FACTURA.

Inserta un registro en la tabla t_timbre con los datos obtenidos al momento de Timbrar el CFDI:

- id
- PAC
- version
- UUID
- fechaTimbrado
- fechaCancelacion
- selloCFD
- noCertificadoSAT
- selloSAT

Regresa el resultado de la operación al módulo que hizo el llamado.

Especificación del módulo conectar al PAC

Recibe los datos de autenticación (USER, PASSWORD) para usar el servicio de timbrado y la cadena XML. Y llama al módulo **solicitar timbrado** pasando la cadena XML.

Regresa el resultado de la operación al módulo que hizo el llamado.

Especificación del módulo solicitar timbrado

Al recibir la cadena XML, el PAC regresa otro arreglo de bytes, que al transformarlo a un formato XML ya contiene los datos del Timbrado, o en su defecto si hubo problemas regresa una cadena XML con datos de error.

Regresa el resultado de la operación al módulo que hizo el llamado.

Especificación del módulo generar archivo PDF (representación Impresa del CFDI)

Este módulo recibe el objeto Factura

En este proceso se utiliza el objeto document de la librería com.itexpdf.text mediante la cual se le van agregando textos provenientes del esquema del documento "PDF" fijos, y datos provenientes del objeto factura variables, ordenándolos de la forma que indican los requisitos del cliente.

Se graba el formato PDF con la ruta y nombre que se describen a continuación:

`/ruta_aplicacion/facturas/rfc_receptor/serie_y_folio.PDF`

Especificación del módulo Imprimir archivo PDF

Del objeto Factura, obtiene el rfc receptor, Serie y folio, y con eso busca en el directorio: `/ruta_aplicacion/facturas/rfc_receptor/serie_y_folio.PDF`

Y valida cada minuto que exista algún archivo .PDF para imprimir, cuando hay un nuevo archivo lo mueve aun archivo de paso, y cuando se imprime éste se borra.

Se usa el objeto PDFFILE de la librería com.sum.pdfview y el objeto com.sum.pdfPrintPage.

Mediante el objeto printerJob de la librería java.awt.print, el cual tiene un método llamado lookupPrintServices que devuelve un arreglo de todas las impresoras disponibles en el sistema operativo.

Se lee el archivo de configuración config.XML y se obtiene el nombre de la impresora a utilizar, y con el nombre de dicha impresora se establece el nombre en el objeto PrinterJob y se le pasa la ruta completa del archivo que se va a imprimir.

Y adicionalmente se le da un tamaño de papel, el área imprimible, y la orientación de la página y se utiliza su método Print.

Especificación del módulo actualiza timbre a factura

Recibe el objeto factura, y el timbre que se obtuvo del PAC, actualiza la tabla t_factura con los datos de timbre que se almacenaron en la tabla t_timbre al hacer la petición de timbrar la cadena XML al PAC.

Regresa el resultado de la operación al módulo que hizo el llamado.

Especificación del módulo genera comprobante fiscal digital

Espera la llegada de un archivo .TXT en el directorio FACTPBAS, en cuanto llega este archivo llama el módulo **lee datos de factura del archivo .TXT**, si el resultado fue exitoso, llama al módulo **almacena factura con todos sus componentes**.

Realiza el llamado de los siguientes módulos en el orden indicado:

- Lee factura con todos sus componentes
- Escribe archivo .XML
- Genera sello digital
- Re-escribe archivo .XML
- Timbrar CFDI
- Escribe archivo .XML
- Generar archivo .PDF
- Enviar archivo por mail

En caso de existir algún error en algunos de los pasos de este módulo, se agrega un error en el log de la aplicación, haciendo un llamado al módulo **escribe error en el log**, al cual se le pasa el mensaje de error correspondiente.

Especificación del módulo valida formato XML

Recibe la ruta y el nombre del archivo del archivo XML, y verifica que tenga la estructura documentada en el apartado 4.1.8 que se titula: Definición del CFDI en formato XML.

Especificación del módulo imprimir archivo PDF

Recibe la ruta y el nombre del archivo que se enviará por mail, así como el email al que se enviarán los archivos, y se envía el correo electrónico con los archivos adjuntos.

Especificación del módulo imprimir archivo PDF

Recibe la ruta y el nombre del archivo que se enviará por mail, así como el email al que se enviarán los archivos, y se envía el correo electrónico con los archivos adjuntos.

4.2 CREACIÓN DE LA BASE DE DATOS E IMPLEMENTACIÓN DEL SISTEMA SCFDI

Como se menciona en capítulos anteriores, la base de datos que se utilizara será MySQL, se mostrará dónde y cómo se debe de realizar la instalación de la misma para poder ser utilizada en el sistema de generación de comprobantes fiscales digitales por Internet (SCFDI).

Como primer paso, debemos de obtener el manejador de base de datos (MySQL), el que nos servirá para poder crear la base de datos donde se resguardará la información que se utilice al generar los CFDI`s y que utilizara nuestro sistema.

Para poder obtener una copia, tenemos que conectarnos a Internet y navegar hasta la siguiente página. Figura 4.2.1 Descarga MySQL

<http://www.mysql.com/downloads/>

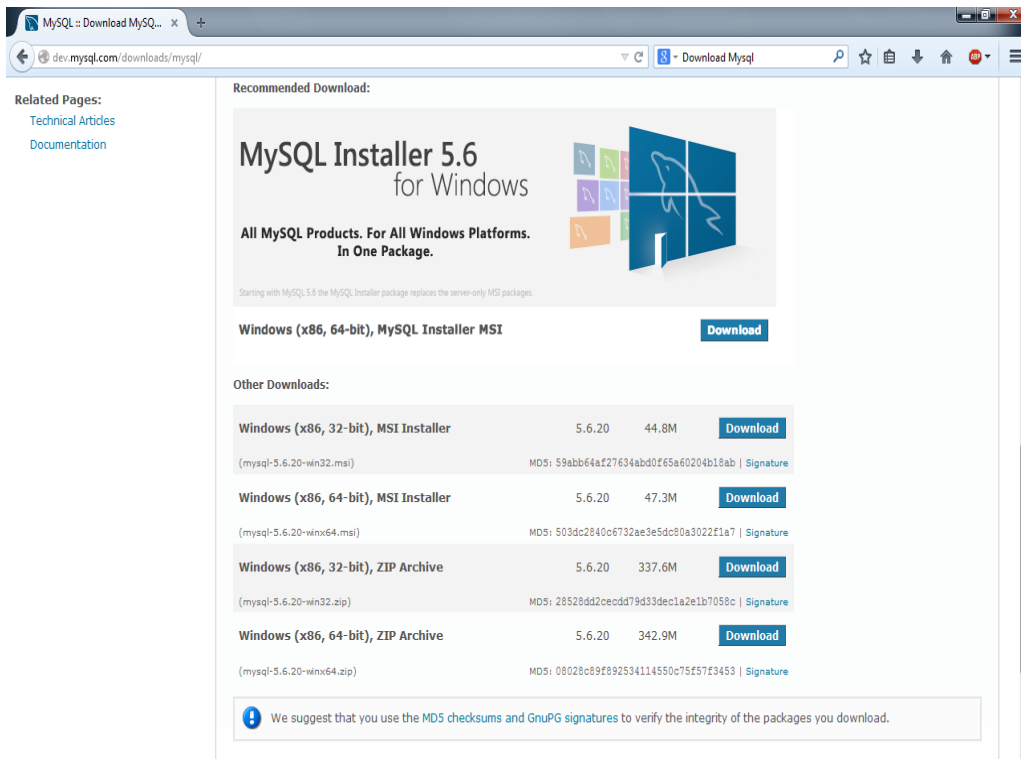


Figura 4.2.1 Descarga MySQL

Descargar y guardar el archivo .zip, para poder ejecutar e instalar el manejador de base de datos (MySQL) adquirido. Figura 4.2.2 Descarga y ejecución

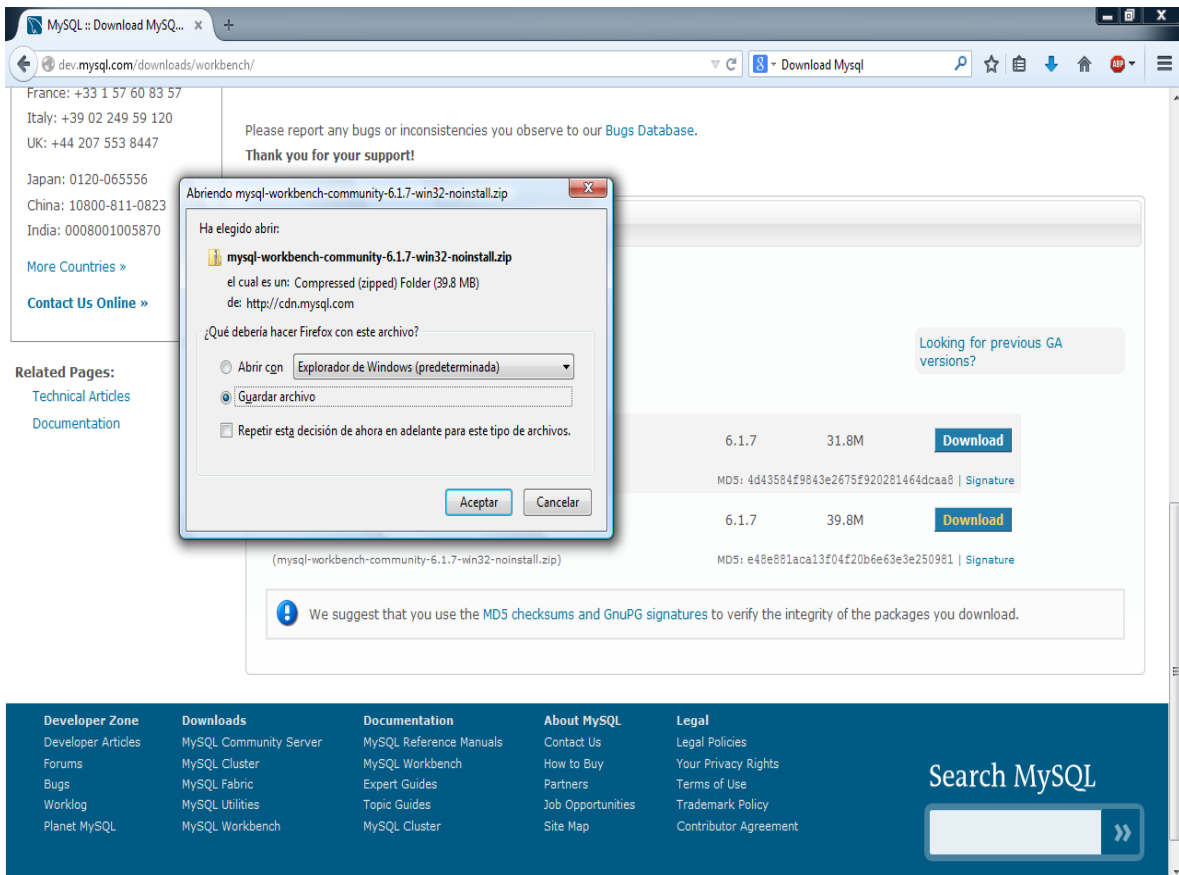


Figura 4.2.2 Descarga y ejecución

Al momento de ejecutar el programa para la instalación de MySQL, se deben seguir las instrucciones que van apareciendo en la pantalla, está nos guiara paso a paso de cómo realizar las acciones necesarias para lograr la instalación requerida.

Elegir la opción de FULL para instalar todos los productos que se encuentran disponibles para MySQL. Vea la figura 4.2.3 Selección de instalación completa.

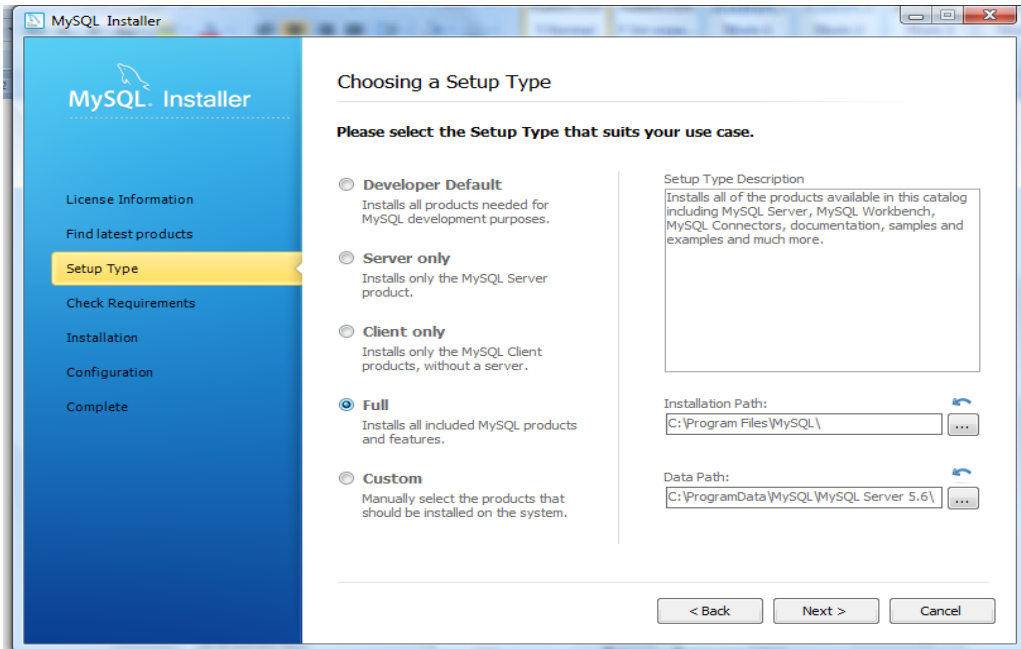


Figura 4.2.3 Selección de instalación completa.

Llenar las contraseñas correspondientes que se soliciten, no olvidarlas ya que son necesarias para la administración de la base de datos. Vea la figura 4.2.4. Asignación de contraseñas.

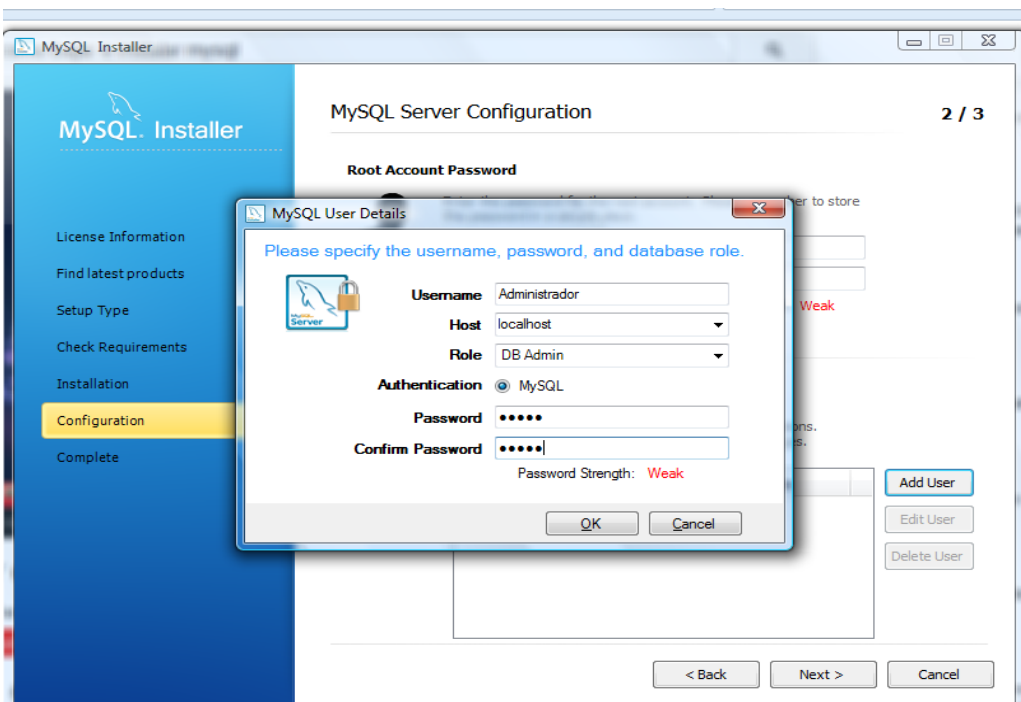


Figura 4.2.4. Asignación de contraseñas.

Cuando MySQL queda instalado correctamente, estaremos listos para poder crear la base de datos correspondiente. Para poder crearla, tenemos que iniciar sesión en MySQL, después en la línea de comando agregar la instrucción CREATE DATABASE CFDIS y de esta manera se estará creando nuestra base de datos que se utilizará. Ver figura 4.2.5. Creación base de datos.

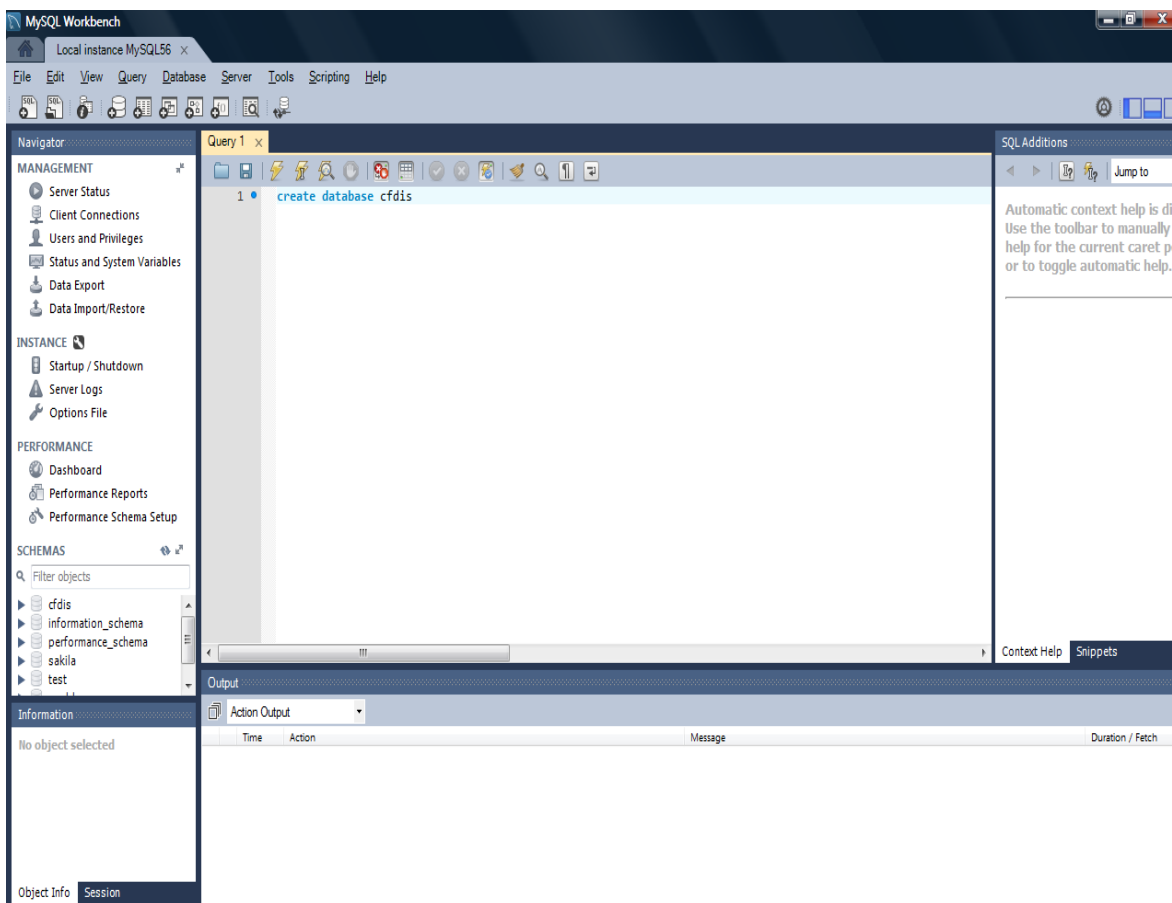


Figura 4.2.5. Creación base de datos.

Una vez que ya definimos nuestra base de datos, lo siguiente es agregar las tablas correspondientes, las cuales estarán almacenando los datos necesarios para poder realizar todas las operaciones requeridas para la generación de comprobantes fiscales digitales por Internet.

Para poder agregar una tabla en nuestra base de datos figura 4.2.6. Creación de tabla, se hace de la forma siguiente:

- Se declara la utilización del uso de nuestra base de datos con el comando “USE”.
- Se crea la tabla en la base de datos seleccionada con el comando “CREATE”.
- Se agregan los campos correspondientes que definirán la tabla con el tipo de dato que le corresponde.

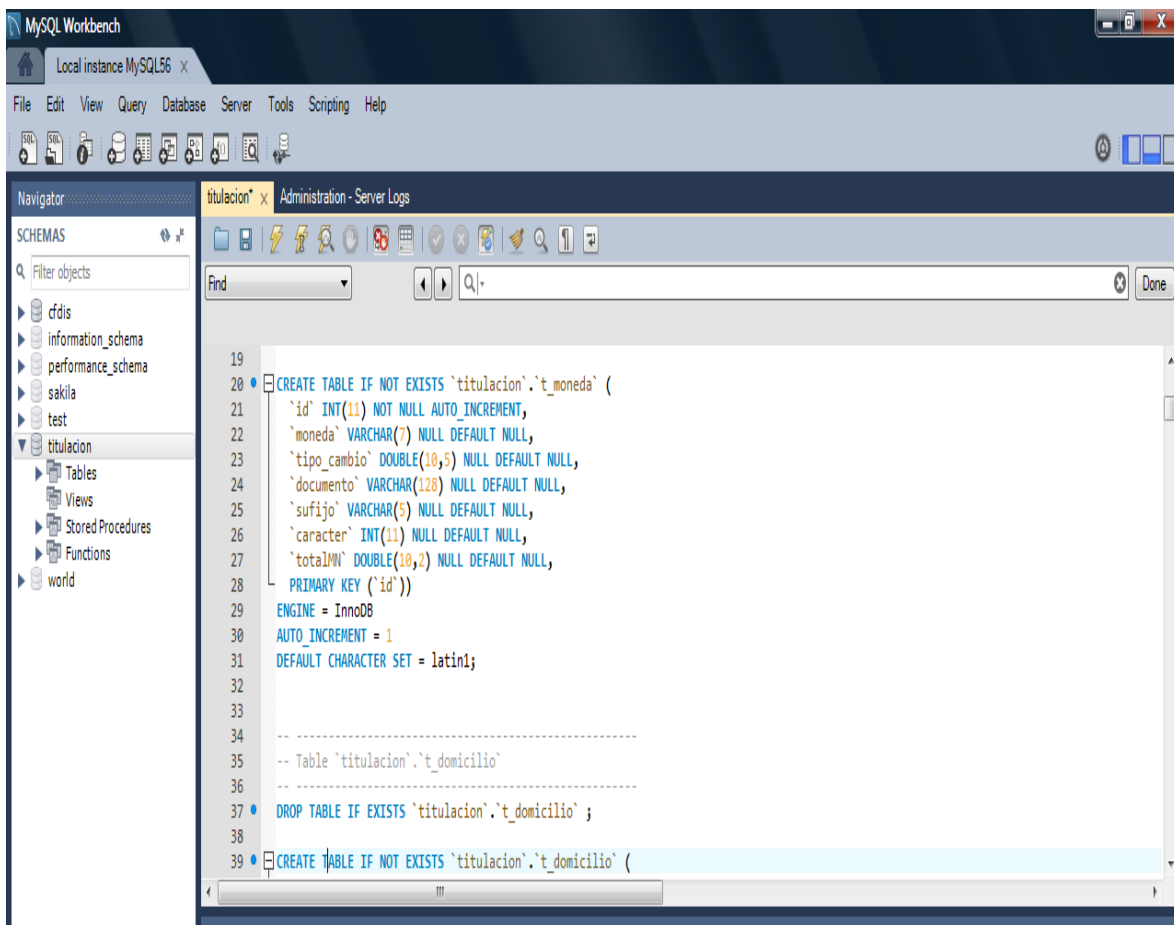


Figura 4.2.6. Creación de tabla.

Posteriormente para poder realizar una consulta a la base de datos se debe de realizar de la siguiente manera, vea la figura 4.2.7. Consulta. Se realizará el ejemplo con la tabla de conceptos que se llama c_conceptos.

La instrucción que dará como resultado una consulta es el comando de SELECT, y debemos de señalar la tabla y la base de datos a la que se requiere realizar una consulta.

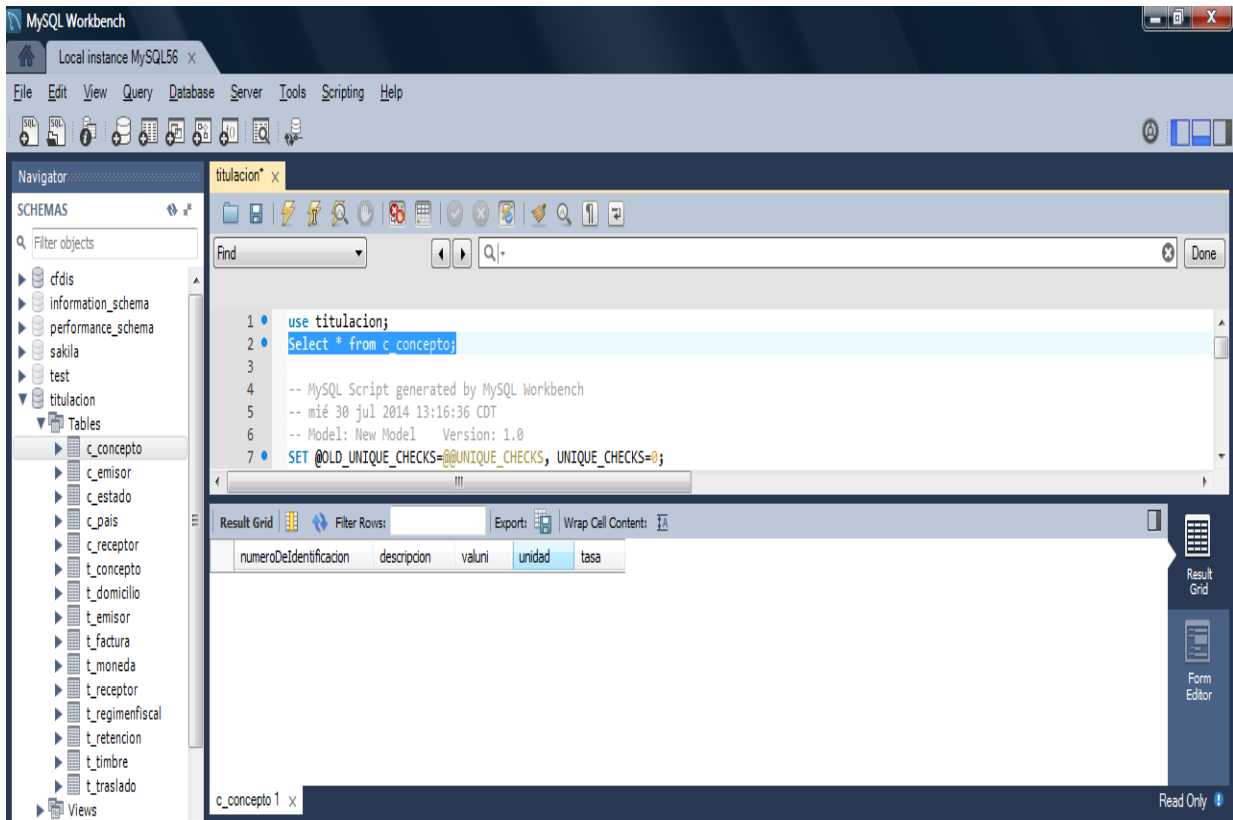


Figura 4.2.7. Consulta.

Un trigger es un objeto con nombre en una base de datos que se asocia con una tabla, y se activa cuando ocurre un evento en particular para esa tabla, la sentencia que se utiliza es la siguiente:

```
CREATE TRIGGER nombre_disp momento_disp evento_disp
ON nombre_tabla FOR EACH ROW sentencia_disp
```

El disparador queda asociado a la tabla, y esta debe ser una tabla permanente, no puede ser una tabla TEMPORARY ni una vista.

Para momento_disp es el momento en que el disparador entra en acción. Puede ser BEFORE (antes) o AFTER (después), para indicar que el disparador se ejecute antes o después que la sentencia que lo activa.

El evento_disp indica la clase de sentencia que activa al disparador. Puede ser INSERT, UPDATE, o DELETE. Por ejemplo, un disparador BEFORE para sentencias INSERT podría utilizarse para validar los valores a insertar.

No puede haber dos disparadores en una misma tabla que correspondan al mismo momento y sentencia. Por ejemplo, no se pueden tener dos disparadores BEFORE UPDATE. Pero sí es posible tener los disparadores BEFORE UPDATE y BEFORE INSERT o BEFORE UPDATE y AFTER UPDATE.

La sentencia_disp es la sentencia que se ejecuta cuando se activa el disparador. Si se desean ejecutar múltiples sentencias, deben colocarse entre BEGIN y END, el constructor de sentencias compuestas. Esto además posibilita emplear las mismas sentencias permitidas en rutinas almacenadas. Vea la figura 4.2.8 Creación de tablas de ejemplo para trigger y la figura 4.2.9 Creación de trigger.

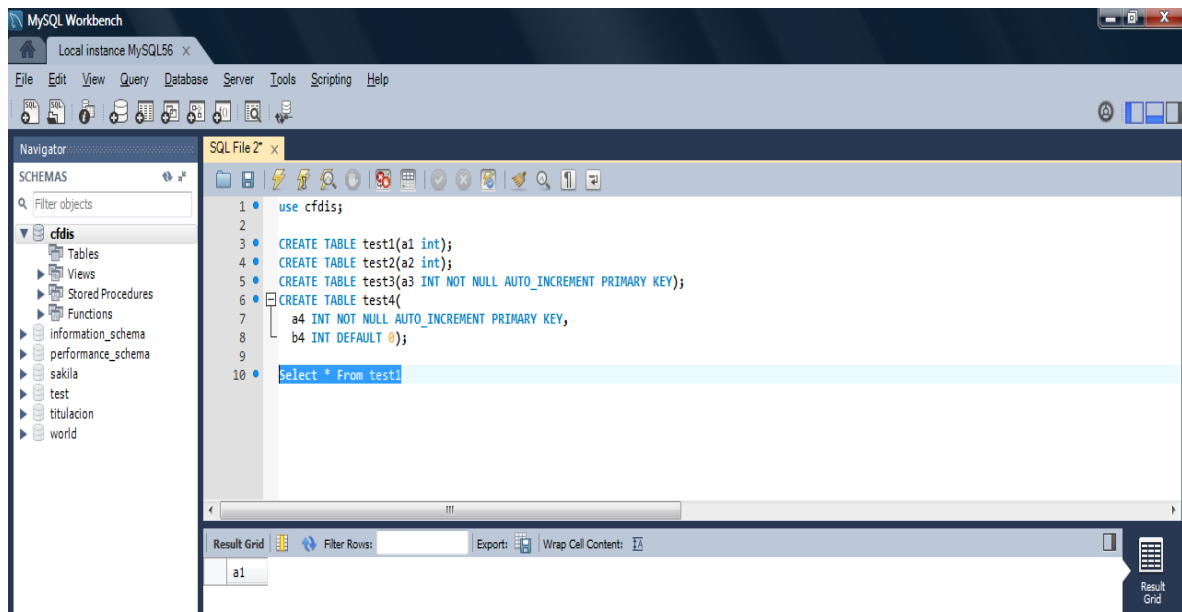


Figura 4.2.8 Creación de tablas de ejemplo para trigger.

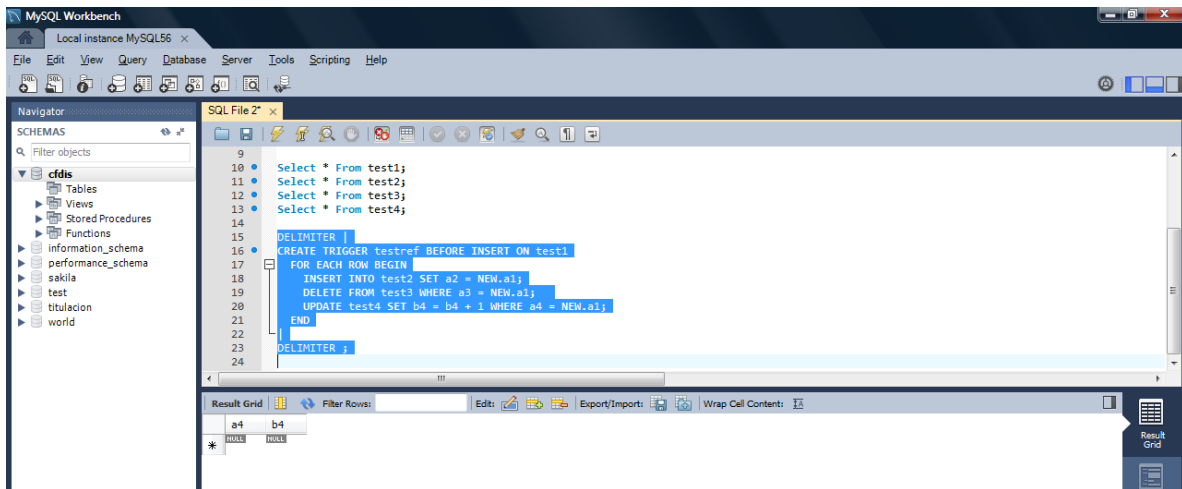


Figura 4.2.9 Creación de trigger.

Las columnas de la tabla asociada con el disparador pueden referenciarse empleando los alias OLD y NEW. OLD.nombre_col hace referencia a una columna de una fila existente, antes de ser actualizada o borrada. NEW.nombre_col hace referencia a una columna en una nueva fila a punto de ser insertada, o en una fila existente luego de que fue actualizada.

El uso de SET NEW.nombre_col = valor necesita que se tenga el privilegio UPDATE sobre la columna. El uso de SET nombre_var = NEW.nombre_col necesita el privilegio SELECT sobre la columna.

Una stored procedure es el procedimiento de una base de datos que está en condición de ejecutar operaciones complejas dentro de la misma base de datos. Es un programa escrito en SQL o en otros lenguajes que provienen de SQL.

La sentencia que se utiliza para poder realizarlo es la siguiente:

```
CREATE PROCEDURE sp_name ([parameter[,...]])
    [characteristic ...] routine_body
```


Vea la figura 4.2.10 creación de store procedure y 4.2.11 Ejecución del store procedure

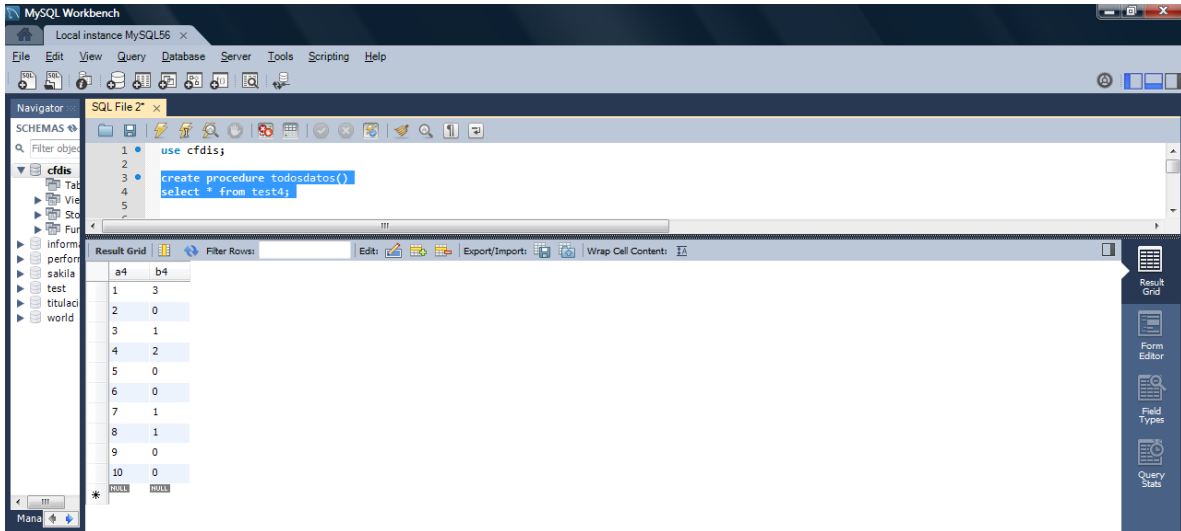


Figura 4.2.10 Creación de store procedure.

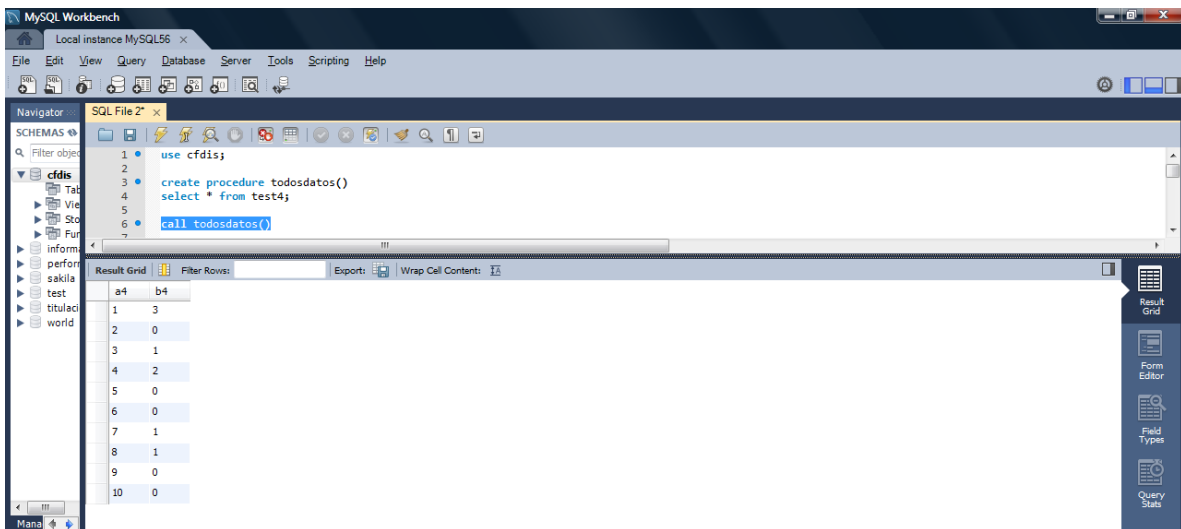


Figura 4.2.9 Ejecución de store procedure.

El desarrollo de la base de datos que se implementó en el sistema para resolver el tema de esta tesis no se requirió el uso de triggers y store procedure.

El árbol de tablas de la base de datos creada queda como se muestra en la figura 4.2.10 Diagrama Entidad Relación del Sistema.

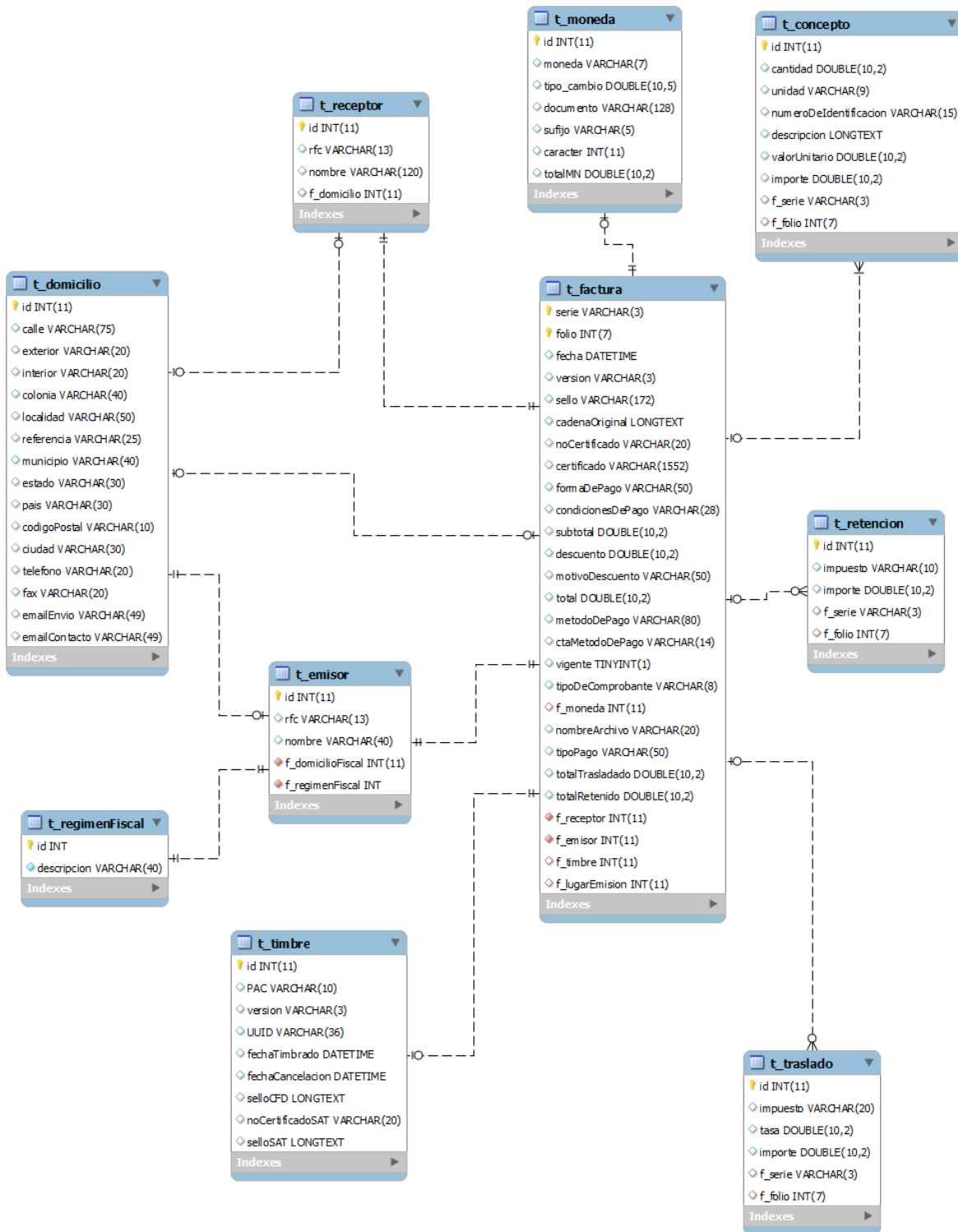


Figura 4.2.10 Diagrama Entidad Relación del Sistema.

4.3 CREACIÓN DEL FRONT-END CON LA FUNCIONALIDAD PEDIDA POR EL SAT

Como se habló en el capítulo 3.5, la interfaz deberá contener los requisitos básicos para la creación de un comprobante fiscal digital y que su vez deberá cumplir con los requisitos solicitados por el SAT para realizar el proceso de facturación electrónica.

A continuación se describirán las funciones con las que cuenta la interfaz del SCFDI.

Funciones básicas

Área de trabajo principal

A continuación se muestra la presentación visual de la interfaz para el SCFDI. Vea figura 4.3.1, 4.3.2 y 4.3.3

The screenshot displays the user interface for the SCFDI system. At the top, there is a logo for 'Industria' and the title '- Sistema de Comprobantes Fiscales Digitales por Internet -'. The interface is divided into three main sections:

- Acciones:** A sidebar on the left containing a button labeled 'Limpiar Campos'.
- Concepto:** A central form section with the following fields:
 - Identificador:
 - Descripcion:
 - Cantidad:
 - Unidad:
 - Valor Unitario:Below these fields are two buttons: 'Modo' and 'Por Identificador', and a 'Guardar' button at the bottom.
- Receptor:** A section below the 'Concepto' section with the following fields:
 - Nombre:
 - RFC:
 - RFCE:
 - Cuenta:
- Notificaciones:** A sidebar on the right containing the text: 'Para iniciar debes agregar al menos un producto'.

Figura 4.3.1 Vista superior de la interfaz del usuario del SCFDI

Receptor

Nombre:

RFC:

RFCE:

Cuenta:

Telefono:

Fax:

E-Mail:

Domicilio

Calle:

Interior:

Exterior:

Colonia:

C.P.:

Estado:

Localidad:

Pais:

Autocompletado

Figura 4.3.2 Vista central de la interfaz del usuario del SCFDI

Factura

Forma de pago:

Cond de pago:

Descuento:

Motivo Descuento:

Metodo de pago:

Comprobante:

Moneda

Tipo de Moneda:

Tipo de Cambio:

Vendedora de libros, Direccion Calle Alamos No. 123 Col. Centro
Mexico D.F.

Figura 4.3.3 Vista inferior de la interfaz de usuario del SCFDI

La interfaz contendrá un formulario que se encontrará ubicado en el centro de la página. Este será el área de trabajo del usuario y aquí será posible completar los datos requeridos de la factura.

Este formulario se dividirá en tres bloques para su organización, el primer bloque será el bloque de concepto, este contiene los datos relacionados con los productos.

Bloque concepto

El bloque concepto contiene los campos para ingresar los datos del producto, estos campos son los siguientes:

- Identificador: Representación numérico del producto.
- Descripción: Descripción del producto.
- Cantidad: Número de productos por descripción
- Unidad: Descripción unitaria del producto.
- Valor unitario: Costo del producto.

Dentro de este bloque habrá dos métodos para agregar productos: por identificador y por descripción. Cada uno de estos métodos contiene una función de autocompletar, que consiste en el autocompletado de los registros que vaya ingresando el usuario en tiempo real. Estos datos serán consultados a través de una tabla distinta de bases de datos.

Para realizar el cambio de método de entrada se añadirá un botón con el nombre de “Modo”, ubicado en la parte inferior de este bloque.

A continuación se mostrará la ubicación de este botón en el formulario. Vea figura 4.3.4

Figura 4.3.4 Ubicación del botón Modo en el bloque concepto.

Dentro de este mismo bloque se añadió un botón de guardar, esta función guardará los nuevos productos de manera temporal, la funcionalidad de la interfaz verificará si los campos de este bloque fueron completados, sino es así, se desplegará un mensaje de error que informará al usuario acerca de este error.

Por último, después de grabar los datos, este informará al usuario sobre el éxito de haber guardado el producto y además se limpiarán los campos del bloque concepto con el fin de preparar este módulo para guardar un nuevo producto. Vea figura 4.3.5

- Sistema de Comprobantes Fiscales Digitales por Internet -

Acciones

Limpiar Campos

Notificaciones

Para iniciar debes agregar al menos un producto

Concepto

Identificador: ej. 9786

Descripción: ej. JAIME BAENA

Cantidad: ej. 2

Unidad:

Valor Unitario:

Modo Por Identificador

Guardar

Has agregado 1 producto(s)

Figura 4.3.5 Mensaje al agregar nuevo producto

Bloque receptor

Próximo al bloque concepto se encuentra el bloque receptor, en este formulario será posible añadir los datos del receptor.

Los campos disponibles dentro de este bloque son los siguientes:

- Nombre: Nombre del receptor.
- RFC: Registro Federal del Contribuyente.
- RFCE: RFC extranjero
- Cuenta: Número de cuenta del receptor.
- E-Mail: Dirección de correo electrónico del receptor
- Fax: Número telefónico del fax del receptor.
- Teléfono: Número telefónico del receptor.

Dentro del apartado "Domicilio" se encuentran los siguientes campos:

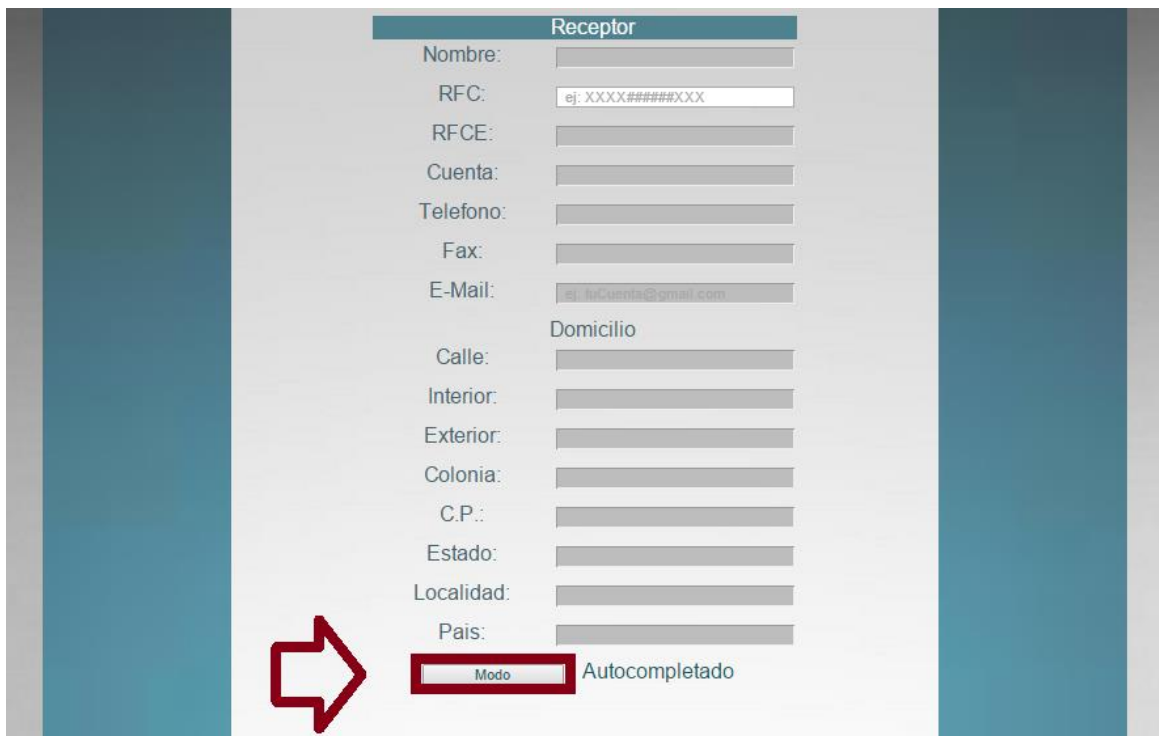
- Calle: Calle del domicilio del receptor
- Interior: Número interior del domicilio
- Exterior: Número exterior del domicilio
- Colonia: Colonia del domicilio del receptor
- C.P.: Código postal del domicilio del receptor

- Estado: Estado federativo del domicilio del receptor
- Localidad: Localidad del receptor
- País: País del receptor

Similar al bloque concepto, el bloque receptor integra dos métodos para agregar receptores: el modo automático y el modo manual.

En el modo automático el usuario podrá ingresar únicamente el RFC deseado a través del campo con el mismo nombre. El campo "RFC" estará ligado a una funcionalidad que le permitirá autocompletar el valor que le es ingresado, esto a través de una consulta a una tabla de base de datos externa. Después de autocompletar el campo RFC, se autocompletarán los datos mediante la consulta antes mencionada.

A continuación se muestra la ubicación del botón "Modo" dentro del formulario. Vea figura 4.3.6



The image shows a web form titled "Receptor" with the following fields:

- Nombre:
- RFC:
- RFCE:
- Cuenta:
- Telefono:
- Fax:
- E-Mail:
- Domicilio section:
 - Calle:
 - Interior:
 - Exterior:
 - Colonia:
 - C.P.:
 - Estado:
 - Localidad:
 - País:

At the bottom of the form, there is a "Modo" button with a red outline and a red arrow pointing to it from the left. The text "Autocompletado" is visible to the right of the button.

Figura 4.3.6 Ubicación del botón modo en el bloque receptor

Al activar el modo “Manual”, la interfaz desbloqueará todos los campos del bloque receptor y el usuario podrá completar todos los datos de este bloque de manera manual.

Bloque factura

Dentro del bloque factura se solicitan los siguientes campos:

- Forma de pago: Forma de pago en exhibición del producto.
 - Una exhibición
 - Dos exhibiciones
- Condición de pago: Condiciones de pago del producto
 - Ninguna
 - 30 Días
 - 60 Días
 - 90 Días
- Descuento: Valor del descuento al producto
- Motivo descuento: Motivo del descuento del producto, este campo de desbloquea al completar el campo Descuento.
- Método de pago:
 - N/A
 - Efectivo
 - Tarjeta de crédito
 - Tarjeta de Débito
 - Tarjeta de servicio
 - Transferencia electrónica
 - Monedero electrónico
 - Cheque nominativo
- Tipo comprobante:
 - Ingreso
 - Egreso
 - Traslado
- Dentro del apartado moneda encontramos los siguientes campos:

-
- Tipo de moneda:
 - Peso
 - Dólar
 - Euro
 - Tipo de cambio: Valor monetario (Obligatorio).

Panel izquierdo

Además la interfaz contendrá en su parte izquierda dentro de la barra lateral un botón que servirá para limpiar todos los campos del formulario. Vea figura 4.3.7

- Sistema de Comprobantes Fiscales Digitales por Internet -

Acciones

Limpiar Campos

Notificaciones

Para iniciar debes agregar al menos un producto

Concepto

Identificador: ej: 9786

Descripcion: ej: JAIME BAENA

Cantidad: ej: 2

Unidad:

Valor Unitario:

Modo: Por Identificador

Guardar

Figura 4.3.7 Vista de los botones de funciones básicas

Panel derecho

El área derecha de la interfaz contendrá el área de notificaciones, aquí se visualizarán diversos mensajes relacionados a errores creados durante el proceso de completado de datos en el formulario.

A continuación se describen algunas notificaciones que se visualizarán en la interfaz. Vea figuras 4.3.8, 4.3.9, 4.3.10 y 4.3.11



Figura 4.3.8 Notificaciones de inicio



Figura 4.3.9 Notificación de verificador de factura

- Sistema de Comprobantes Fiscales Digitales por Internet -

Acciones

Limpiar Campos

Concepto	
Identificador:	9786070107207
Descripcion:	JAIME BAENA LIZBETH - FORMA
Cantidad:	ej: 2
Unidad:	PIEZA
Valor Unitario:	156.00

Modo **Por Identificador**

Guardar

Notificaciones

Completa todos los datos del producto



Figura 4.3.10 Notificación por no guardar al menos un producto

- Sistema de Comprobantes Fiscales Digitales por Internet -

Acciones

Limpiar Campos

Concepto	
Identificador:	ej: 9786
Descripcion:	ej: JAIME BAENA
Cantidad:	ej: 2
Unidad:	
Valor Unitario:	

Modo **Por Identificador**

Has agregado 1 producto(s)

Guardar

Notificaciones

Factura creada exitosamente



Figura 4.3.11 Notificación de factura creada

Funciones extra

Dentro de las funciones extra de la interfaz se encuentra la de ocultar los bloques del formulario haciendo clic sobre la pestaña que contiene el nombre del bloque.

Con esto el usuario será capaz de contraer y desplegar dichos bloques para una mejor visibilidad de los datos. Esta propiedad facilitará la navegación del usuario a la aplicación. Vea figura 4.3.12

The screenshot displays a web interface for a digital fiscal receipt system. The title is '- Sistema de Comprobantes Fiscales Digitales por Internet -'. The interface is divided into three main vertical sections:

- Acciones (Left):** Contains a button labeled 'Limpiar Campos'.
- Notificaciones (Right):** Contains the text 'Para iniciar debes agregar al menos un producto'.
- Formulario (Center):**
 - A dark teal bar at the top is labeled 'Concepto' and is currently collapsed.
 - Below it is another dark teal bar labeled 'Receptor'.
 - Form fields for 'Receptor' include: 'Nombre:', 'RFC:' (with a placeholder 'ej. XXXX#####XXX'), 'RFCE:', 'Cuenta:', 'Telefono:', 'Fax:', and 'E-Mail:' (with a placeholder 'ej. luCuenta@gmail.com').
 - A section titled 'Domicilio' follows, with fields for 'Calle:', 'Interior:', 'Exterior:', 'Colonia:', 'C.P.:', 'Estado:', 'Localidad:', and 'Pais:'.
 - At the bottom of the form is a 'Modo' dropdown menu set to 'Autocompletado'.

Figura 4.3.12 Vista al contraer la cortina “Concepto”

El sistema integrará un verificador para que los campos no contengan errores, esto también servirá para que el sistema no maneje datos erróneos. Vea figura 4.3.13



Figura 4.3.13 .Error al no completar algún campo del producto

Confirmación

Por último al presionar el botón enviar datos, este mostrará una ventana emergente con los datos de la factura recién completada.

Al presionar el botón aceptar enviaremos por completo la factura para su siguiente paso al SCFDI, en contraste al presionar el botón cancelar, cancelaremos la factura con la posibilidad de hacer algunos cambios. Vea la figura 4.3.14

Acciones

Limpiar Campos

Confirma los datos de la factura

- Concepto -

Producto: JAIME BAENA LIZBETH - FORMACION CIVICA Y ETICA 1
Cantidad: 2
Unidad: PIEZA
Precio: 156.00

Producto: MORENO FRACISCO MARTIN - 100 MITOS DE LA HISTORIA DE MEXICO VOL.2
Cantidad: 1
Unidad: PIEZA
Precio: 249.00

- Receptor -

Nombre: GUTIERREZ DELGADO JOSE EDUARDO
Cuenta: 88888888
RFC: GUDE8203295K5
RFCE: EEEE820329EEE
Domicilio
Calle: CARTAGENA SUR Interior: INT Exterior: 66 San Pedro
Zacatenco C.P.: 07360
Gustavo A. Madero Distrito Federa MEXICO

- Factura -

Forma de pago: Una Exhibicion
Condicion de pago: Ninguna
Metodo de pago: N/A
Tipo de comprobante: ingreso
Moneda: PESOS
Tipo de cambio: 13.3

Notificaciones

Para iniciar debes agregar al menos un producto

C.P.:

Estado:

Localidad:

Figura 4.3.14 .Solicitud para confirmar los datos

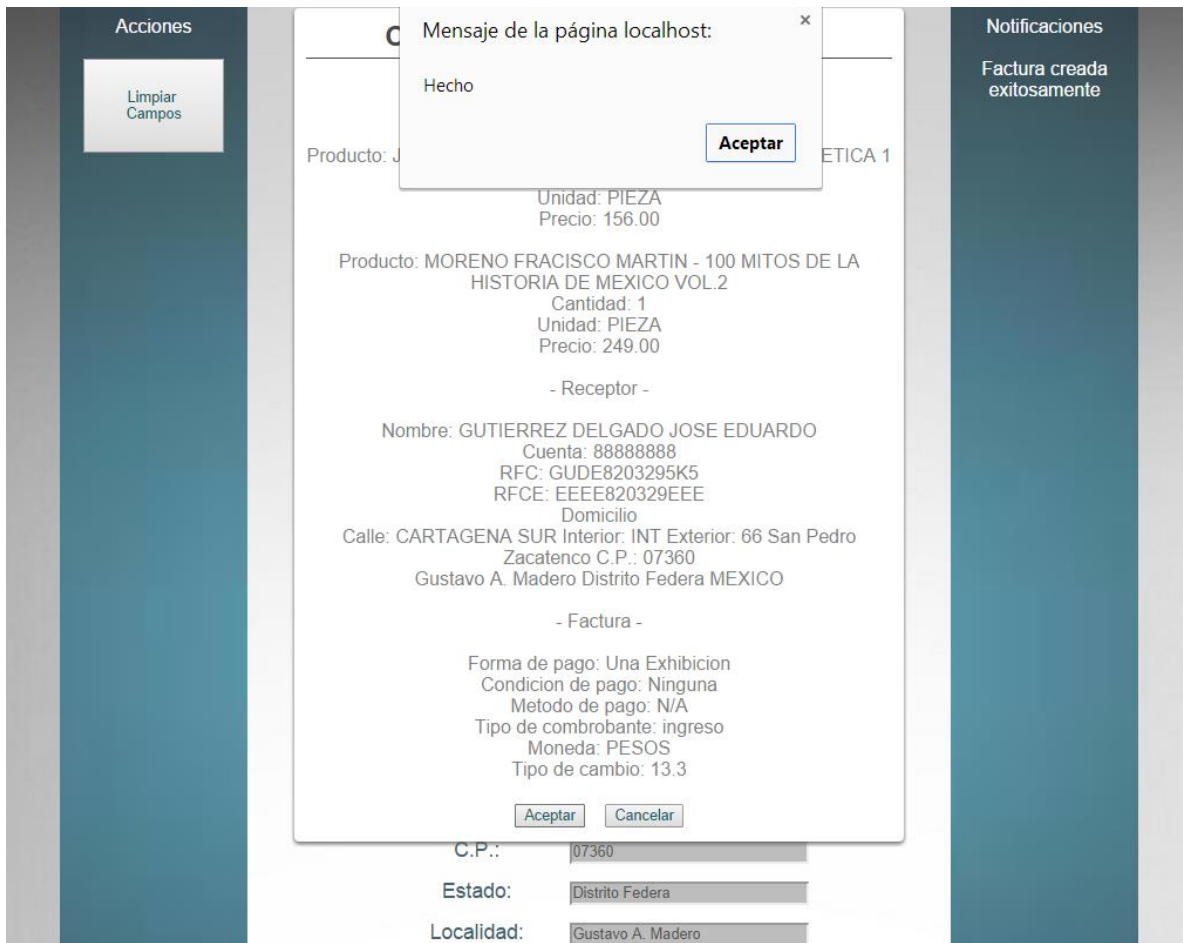


Figura 4.3.15 Vista al confirmar los datos

4.4 GENERACIÓN DE PRUEBAS Y MANTENIMIENTO

Las pruebas y el mantenimiento son fases importantes del ciclo de vida de un sistema, y ambos rubros son tratados en la Ingeniería de Software. La ingeniería de Software es una rama de la Ingeniería en computación cuyo principal objetivo es que el desarrollo de sistemas de Software sea costeable y que se produzcan sistemas confiables y de calidad.

Revisaremos primero la importancia de las pruebas, los tipos de prueba, y cuáles se aplicaron a nuestro sistema, y posteriormente revisaremos la cuestión del mantenimiento.

Pruebas

Las pruebas en el desarrollo de sistemas es una actividad fundamental para asegurar la calidad del software a liberar, permite disminuir riesgos y puede representar un ahorro significativo en los costos, tiempo y esfuerzo al evitar que se presenten fallas en producción.

Para muchas compañías, una falla de alguno de sus sistemas en producción puede implicar pérdidas monetarias cuantiosas, o daños a su reputación y credibilidad antes sus clientes, por lo que es altamente recomendable realizar el mayor número de pruebas posibles a los sistemas antes de liberarlos a producción.

Tipos de pruebas

Existen diferentes tipos de pruebas, cada una cubre diferentes aspectos del sistema, algunas pruebas son responsabilidad del área de tecnología y en algunas pruebas debe participar el usuario ya que es quien conoce las reglas del negocio. Describiremos ahora los tipos de pruebas:

Pruebas unitarias

Son aquellas que se realizan a módulos de código del sistema en forma independiente, es decir, se verifica la funcionalidad de cada módulo en forma aislada. La intención de las pruebas unitarias es que al llegar a las pruebas “integrales” o las pruebas que integran a todos los módulos, las piezas de software ya vayan con cierto grado de confiabilidad. Las pruebas unitarias las realizan por lo general los desarrolladores al momento que van concluyendo cada módulo.

Pruebas de validación

Las pruebas de validación se enfocan en validar que el sistema realice las funciones que el usuario definió en sus requerimientos. Estas pruebas las realizan los desarrolladores o el equipo de pruebas en un ambiente controlado, y es

recomendable que en estas pruebas participe también el usuario que conoce las reglas del negocio, ya que de esta forma se garantiza que no se ha omitido o malinterpretado algún requerimiento.

Pruebas de integración

Son las pruebas en las que se involucran y conectan todos los módulos que conforman un proceso. Consiste en realizar pruebas para verificar que un gran conjunto de partes de software funcionan juntos.

Las pruebas de integración es la fase de la prueba de software en la cual módulos individuales de software son combinados y probados como un grupo.

Pruebas de regresión

Este tipo de pruebas pretende identificar algún problema o inconsistencia adicional al que se está corrigiendo, es decir, por lo general son requeridas cuando ya existe un sistema corriendo en producción y se va agregar nueva funcionalidad o se va a corregir una falla actual de producción.

La intención de este tipo de pruebas es garantizar que los cambios o modificaciones no produzcan efectos negativos sobre la pieza que se está corrigiendo u otros componentes. Esto puede ser generado por ejemplo por una práctica no adecuada de control de versiones, o una solución no robusta o simplemente como consecuencia de un rediseño de la aplicación.

Pruebas de caja blanca

Están enfocadas en probar los posibles flujos de ejecución de un programa y cerciorarse de que se devuelven los valores de salida correctos. Se pueden aplicar a varios niveles: unidad, integración y sistema, por lo general se aplican en las pruebas unitarias. Su principal objetivo es validar los flujos de ejecución dentro de cada unidad de código, ya sea una función, una clase un módulo, etc., sin embargo, pueden también emplearse para probar flujos entre unidades durante las pruebas de integración o incluso entre subsistemas.

Pruebas de caja negra

Se denomina caja negra a aquel elemento que es estudiado desde el punto de vista de las entradas que recibe y las salidas o respuestas que produce, sin tener en cuenta su funcionamiento interno. Es decir, la idea es probar que un conjunto de instrucciones que conforman un módulo sea probado como una unidad, sin enfocarse en la forma en que se ejecutan dichas instrucciones internamente.

Este tipo de pruebas facilitan el entendimiento de un sistema y permiten que al ocurrir una falla, el problema sea identificado y abordado de manera más rápida.

Pruebas alfa

Es la primera versión del programa, la cual es enviada a los verificadores para probarla. Algunos equipos de desarrollo utilizan el término alfa informalmente para referirse a una fase donde un producto todavía es inestable, aguarda todavía a que se eliminen los errores o a la puesta en práctica completa de toda su funcionalidad, pero satisface la mayoría de los requisitos.

Pruebas beta

Es una prueba piloto, difiere de las pruebas alfa en que el desarrollador no está presente normalmente cuando se lleva a cabo la prueba. De esta forma, 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, y envía retroalimentación al desarrollador.

Pruebas de estrés

Se refiere a realizar pruebas con altos volúmenes de información, o alto flujo de transacciones y gran concurrencia de usuarios, con la intención de verificar que el sistema tiene la suficiente capacidad para soportar situaciones reales del ambiente productivo.

Este tipo de pruebas permiten disminuir el impacto en la liberación de sistemas de alta criticidad, ya que permiten identificar posibles cuellos de botella, problemas de arquitectura o infraestructura que pudiera generar una intermitencia que derive en pérdidas significativas para la empresa. Permiten tener elementos para la toma de decisiones sobre configuración a nivel hardware y software, y tener visibilidad acerca del rendimiento del sistema en general y posibles necesidades de escalabilidad.

Pruebas que se aplicaron al sistema SCFDI

Se realizaron pruebas unitarias a los diferentes módulos del sistema, por un lado se realizaron las validaciones en el front, se realizaron validaciones individuales en la captura de los datos de concepto (o el producto que compra un cliente) como se muestra en la figura 4.4.1.

The screenshot displays the web interface for the 'Sistema de Comprobantes Fiscales Digitales por Internet'. At the top, the 'Industria' logo is visible. The main content area is titled '- Sistema de Comprobantes Fiscales Digitales por Internet -'. On the left, there is a sidebar with 'Acciones' and a 'Limpiar Campos' button. On the right, there is a 'Notificaciones' section with the message 'Para iniciar debes agregar al menos un producto'. The central form is divided into two sections: 'Concepto' and 'Receptor'. The 'Concepto' section includes fields for 'Identificador' (9786070107207), 'Descripcion' (JAIME BAENA LIZBETH - FORM), 'Cantidad', 'Unidad' (PIEZA), and 'Valor Unitario' (156.00). Below these fields are radio buttons for 'Modo' (Por Identificador) and a 'Guardar' button. The 'Receptor' section includes fields for 'Nombre', 'RFC', 'RFCE', 'Cuenta', and 'Domicilio'.

Figura 4.4.1 Captura concepto

En caso de que el usuario no capture algún dato mandatorio del concepto, el sistema arroja un error en pantalla como se muestra en la figura 4.4.2.



Figura 4.4.2 Error en pantalla cuando falta un dato del concepto

Se verificó que la interfaz permitiera incorporar diversos productos en el rubro de concepto, cuando se introducen todos los datos mandatorios, el sistema permite ingresar un concepto, tal y como se muestra en la figura 4.4.3 y en la figura 4.4.4.



Figura 4.4.3 Datos completos de Concepto

Cuando el usuario captura los datos completos de concepto y oprime el botón “guardar”, el sistema envía el mensaje de que se ha agregado un producto como se muestra en la figura 4.4.4.

The screenshot displays a web application interface for digital fiscal receipts. The main title is "- Sistema de Comprobantes Fiscales Digitales por Internet -". On the left, under "Acciones", there is a "Limpiar Campos" button. The central "Concepto" form includes the following fields: "Identificador:" with the value "ej: 9786", "Descripción:" with "ej: JAIME BAENA", "Cantidad:" with "ej: 2", "Unidad:" (empty), and "Valor Unitario:" (empty). Below these fields is a "Modo" dropdown menu set to "Por Identificador". At the bottom of the form is a "Guardar" button. On the right, under "Notificaciones", a message reads "Completa todos los datos del producto". Below the "Guardar" button, a confirmation message states "Has agregado 2 producto(s)".

Figura 4.4.4 La aplicación confirma que se ha agregado un producto.

Se verificó que la interfaz permitiera navegar entre los diferentes rubros de información que se deben introducir para generar una factura electrónica, nos permite mover el cursor entre los rubros de concepto, receptor y factura, así como introducir datos en cada celda, como se muestra en la figura 4.4.5.

- Sistema de Comprobantes Fiscales Digitales por Internet -

<p>Acciones</p> <div style="border: 1px solid gray; padding: 5px; text-align: center; width: 80px; margin: 10px auto;">Limpiar Campos</div>	<p>Concepto</p> <p>Identificador: <input type="text" value="ej: 9786"/></p> <p>Descripcion: <input type="text" value="ej: JAIME BAENA"/></p> <p>Cantidad: <input type="text" value="ej: 2"/></p> <p>Unidad: <input type="text"/></p> <p>Valor Unitario: <input type="text"/></p> <p style="text-align: center;"><input type="button" value="Modo"/> Por Identificador</p> <p style="text-align: center; color: green;">Has agregado 2 producto(s)</p> <p style="text-align: center;"><input type="button" value="Guardar"/></p> <p>Receptor</p> <p>Nombre: <input type="text" value="GUTIERREZ DELGADO JOSE EDI"/></p> <p>RFC: <input type="text" value="GUDE8203295K5"/></p> <p>RFCE: <input type="text" value="EEEE820329EEE"/></p> <p>Cuenta: <input type="text" value="88888888"/></p> <p>Telefono: <input type="text" value="5555555555"/></p> <p>Fax: <input type="text" value="6666666666"/></p> <p>E-Mail: <input type="text" value="envio@servidor.com"/></p>	<p>Notificaciones</p> <p>Completa todos los datos del producto</p>
--	--	---

Figura 4.4.5 Permite navegar entre campos e introducir información

Se validó que si falta algún dato mandatorio, la aplicación envía mensaje de que faltan datos por capturar, y el cursor lo posiciona en la celda del dato faltante, tal como se muestra en la figura 4.4.6.

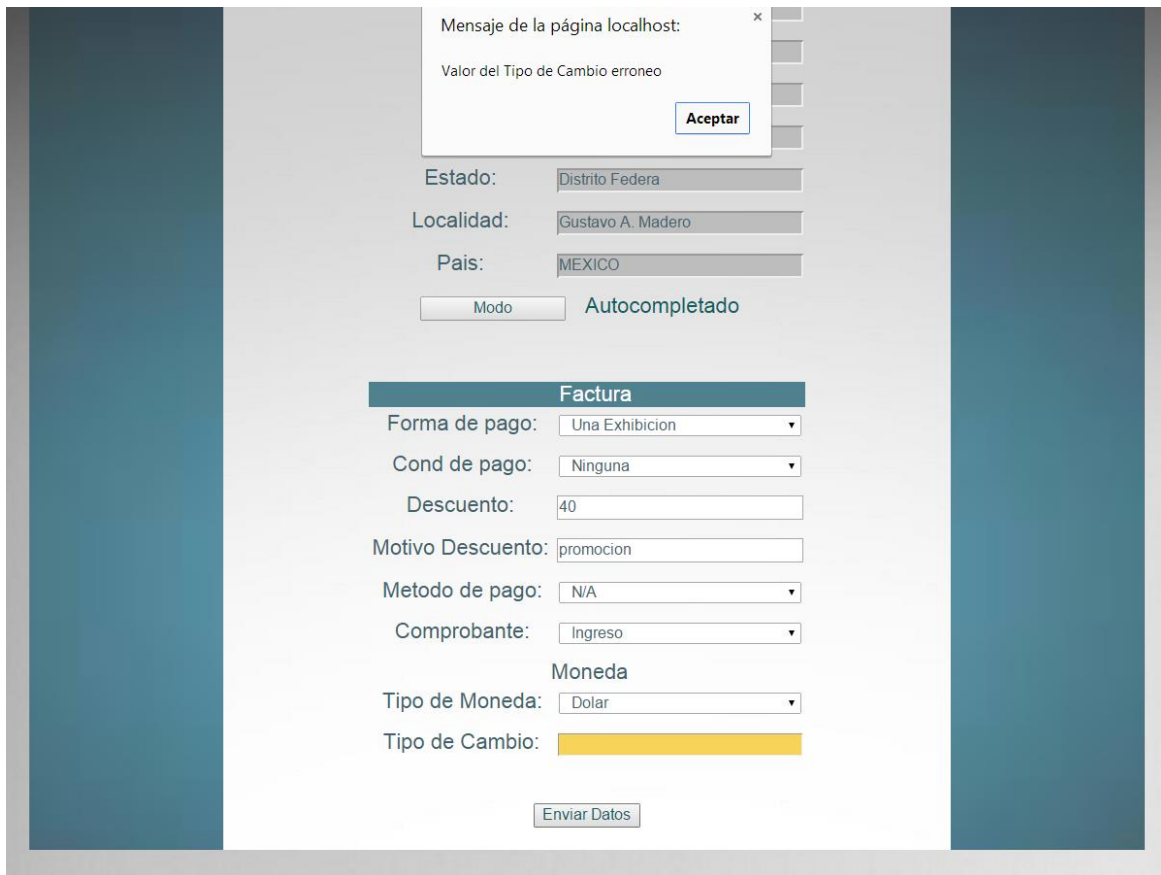


Figura 4.4.6 Cuando falta datos se deja el cursor en la celda del dato faltante

Así mismo, se ejecutó el caso de prueba exitoso, cuando todos los datos obligatorios se han capturado en forma correcta, y se graba el archivo .TXT, ver en la figura 4.4.7.

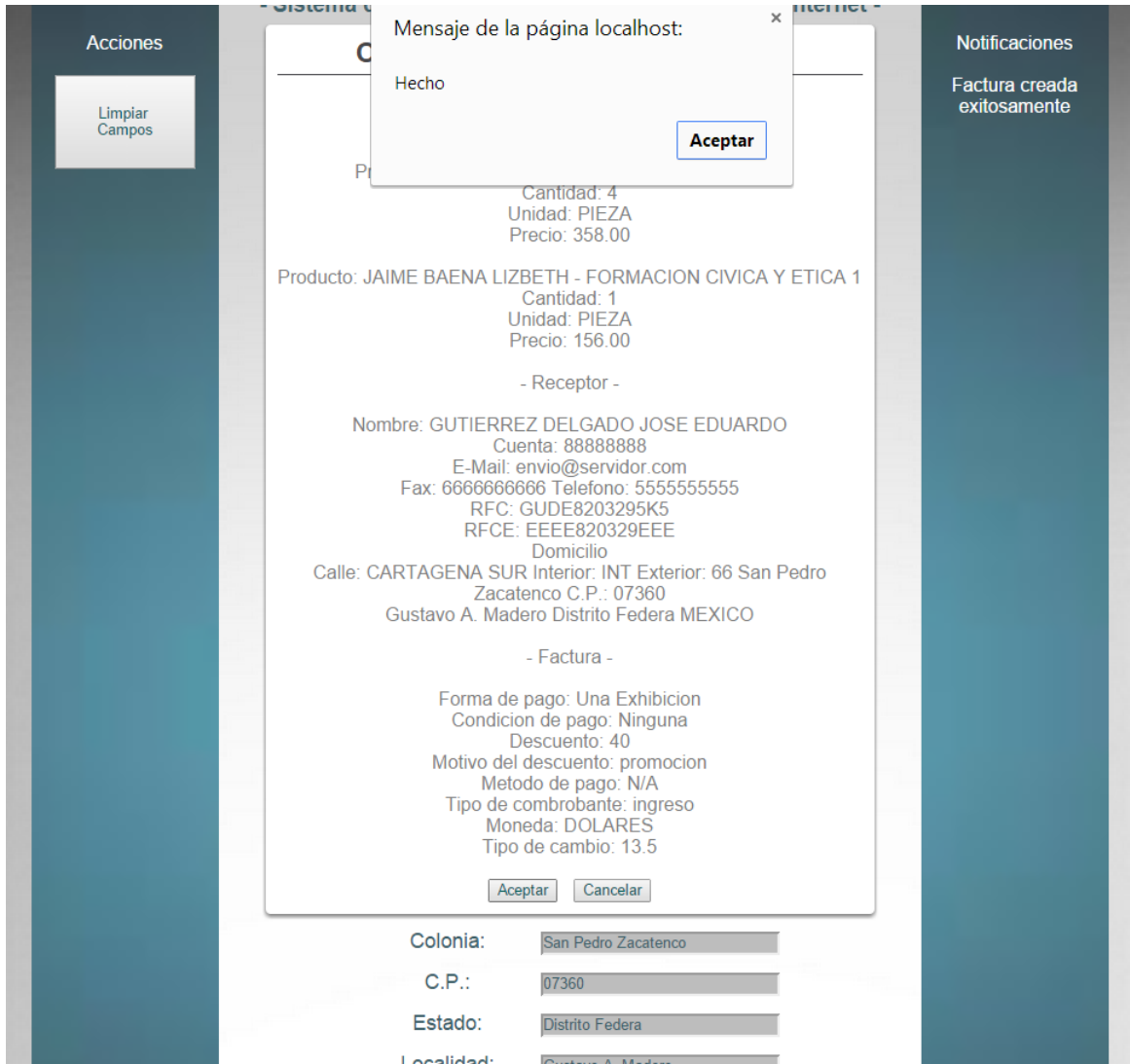


Figura 4.4.7 Se capturan todos los campos obligatorios

Se verificó que el archivo generado .TXT tuviera los campos requeridos que son la entrada para el proceso de generación de la factura electrónica. Ver el ejemplo del archivo .TXT en la figura 4.4.8.

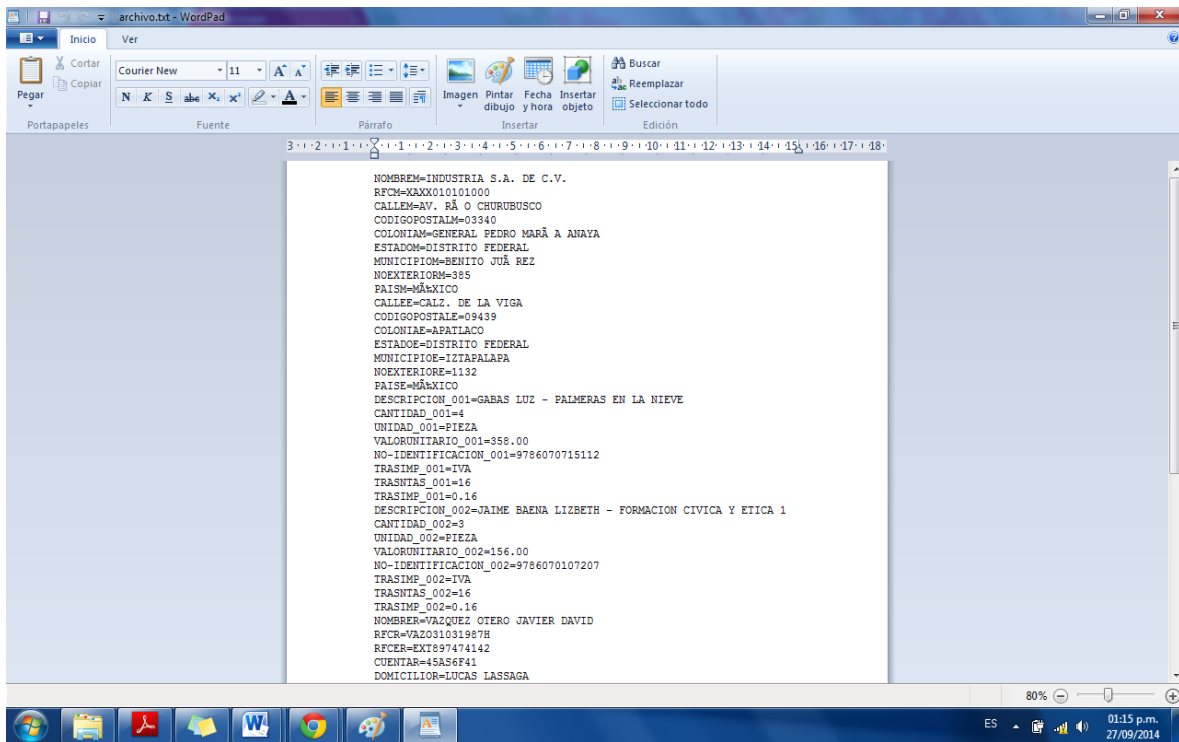


Figura 4.4.8 Salida del archivo .TXT

Se realizaron también pruebas de validación, al probar los dos flujos válidos para limpiar las celdas para capturar un nuevo producto, por un lado cuando se termina de generar un archivo .TXT y cuando es una solicitud del usuario limpiar los campos. Ver figura 4.4.9.

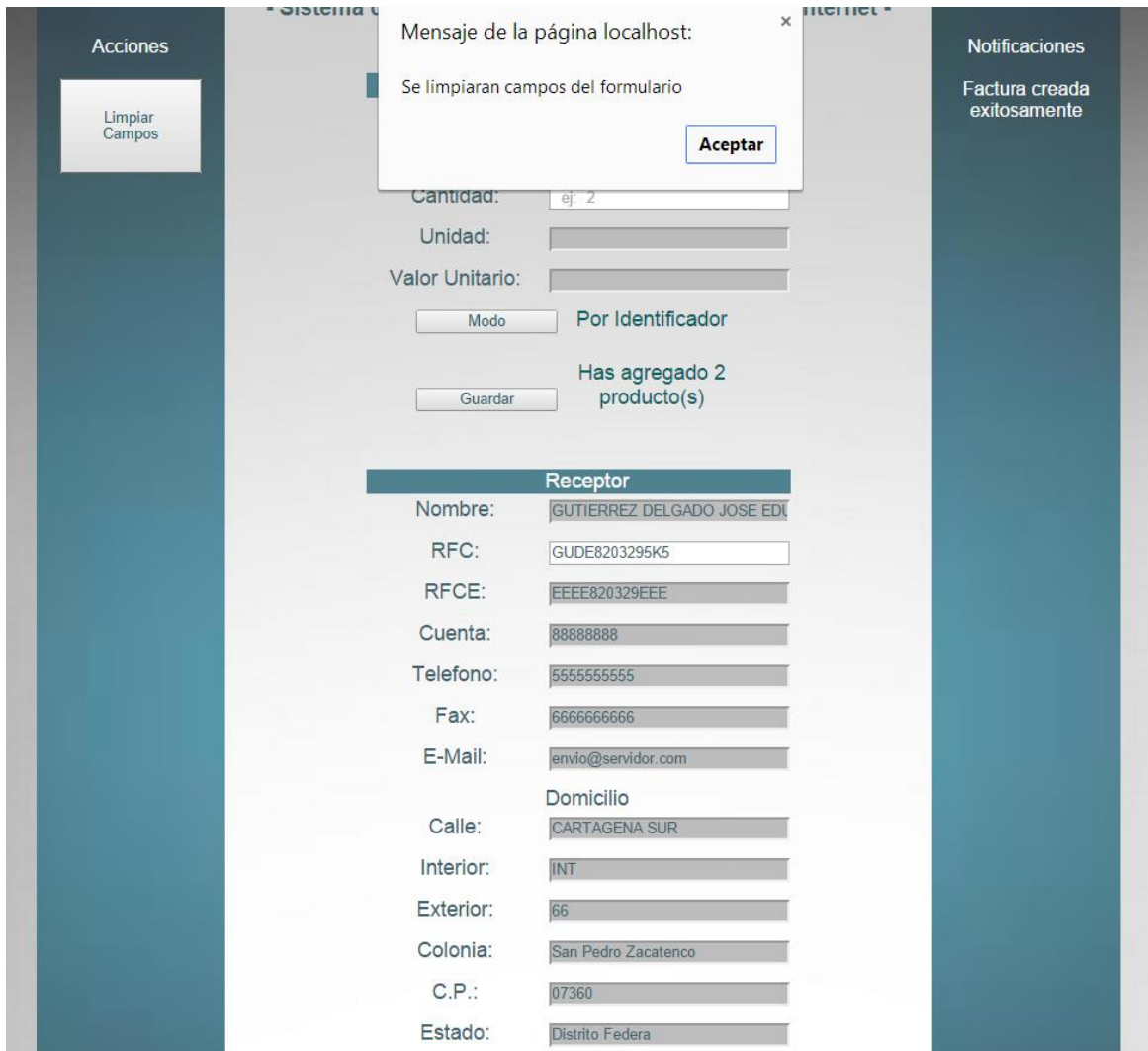


Figura 4.4.9 Función limpiar campos de pantalla (caso exitoso)

Pruebas de caja negra.

Para hablar de “pruebas de caja negra” nos referimos a que se está estudiando el software desde el punto de vista de las entradas que recibe y las salidas que produce, sin tomar en cuenta su funcionamiento interno. En una prueba, de una caja negra nos interesará su forma de interactuar con el medio que le rodea entendiendo qué es lo que hace, pero sin dar importancia a cómo lo hace.

Errores generados por falta de datos requeridos por el SAT.

En el Anexo 20 de la Resolución Miscelánea Fiscal para 2014, publicada el 30 de diciembre de 2013 se establecen los requerimientos de aquellos datos para generar un CFDI y que son obligatorios que aparezcan; concretamente en el nodo Comprobante algunos datos marcados como requeridos son Subtotal, Total o TipoDeComprobante por lo que el archivo TXT que genera el comprobante y que se obtiene del front a utilizar deben estar estos datos. Para generar una prueba por cada uno de los datos mencionados eliminaremos del archivo TXT generado por el front los datos que generan estos componentes del archivo para verificar que se produce un error. Es importante notar que debido a la forma en que da el tratamiento a los archivos de propiedades el lenguaje de programación Java, aquellas líneas iniciadas con el carácter número (#) se consideran como comentarios y no serán procesadas como un conjunto llave-valor para alimentar el sistema. Vea figuras 4.4.10, 4.4.11, 4.4.12, 4.4.13 y 4.4.14

62	#METODODEPAGO=tarjetaDeCredito	62	#METODODEPAGO=tarjetaDeCredito
63	METPAG=tarjetaDeCredito	63	METPAG=tarjetaDeCredito
64	#TIPODECOMPROBANTE=ingreso	64	#TIPODECOMPROBANTE=ingreso
65	DOCUMENTO=ingreso	65	DOCUMENTO=ingreso
66	MONEDA=PESOS	66	MONEDA=PESOS
67	TIPOCAMBIO=1	67	TIPOCAMBIO=1
68	TIPOCAMBIO=1	68	TIPOCAMBIO=1
69	#SUFIJO=''	69	#SUFIJO=''
70	SUFIJO=M.N.	70	SUFIJO=M.N.
71	MONDOCTO=CFDI EN PESOS	71	MONDOCTO=CFDI EN PESOS
72	CARACTER=36	72	CARACTER=36
73	TOTALMN=536.40	73	TOTALMN=536.40
74	SUBTOTAL=500	74	
75	#TOTAL=570	75	#TOTAL=570
76	TOTAL=536.40	76	TOTAL=536.40
77	#AGREGADOS	77	#AGREGADOS
78	SERFAC=JA	78	SERFAC=JA
79	IMPORTE_001=220	79	IMPORTE_001=220
80	IMPORTE_002=280	80	IMPORTE_002=280
81	TTRASLADADO=46.4	81	TTRASLADADO=46.4
82	REGIMEN=REGIMEN GENERAL DE LEY PERSON	82	REGIMEN=REGIMEN GENERAL DE LEY PERSONAS MORALES

Figura 4.410. Eliminación del dato requerido SUBTOTAL para prueba de caja negra

[29-09-14 03:52:56] JA: Factura.subTotal is null and required
[sep 29 03:52:56] [main] ERROR ([Validator.java:231](#)) - Factura.subTotal is null and required

Figura 4.4.11. Error obtenido a partir del dato faltante SUBTOTAL

64	#TIPODECOMPROBANTE=ingreso	64	#TIPODECOMPROBANTE=ingreso
65	DOCUMENTO=ingreso	65	DOCUMENTO=ingreso
66	MONEDA=PESOS	66	MONEDA=PESOS
67	TIPOCAMBIO=1	67	TIPOCAMBIO=1
68	TIPOCAMBIO=1	68	TIPOCAMBIO=1
69	#SUFIJO=''	69	#SUFIJO=''
70	SUFIJO=M.N.	70	SUFIJO=M.N.
71	MONDOCTO=CFDI EN PESOS	71	MONDOCTO=CFDI EN PESOS
72	CARACTER=36	72	CARACTER=36
73	TOTALMN=536.40	73	TOTALMN=536.40
74	SUBTOTAL=500	74	SUBTOTAL=500
75	#TOTAL=570	75	#TOTAL=570
76	TOTAL=536.40	76	
77	#AGREGADOS	77	#AGREGADOS
78	SERFAC=JA	78	SERFAC=JA
79	IMPORTE_001=220	79	IMPORTE_001=220
80	IMPORTE_002=280	80	IMPORTE_002=280
81	TTRASLADADO=46.4	81	TTRASLADADO=46.4

Figura 4.4.12. Eliminación del dato obligatorio TOTAL

[29-09-14 03:58:54] JA: Factura.total is null and required
[sep 29 03:58:54] [main] ERROR ([Validator.java:231](#)) - Factura.total is null and required

Figura 4.4.13. Error obtenido a partir del dato faltante TOTAL

57	FORPAG=Una Exhibicion	57	FORPAG=Una Exhibicion
58	#CONDICIONDEPAGO=Prepago	58	#CONDICIONDEPAGO=Prepago
59	CONPAG=Prepago	59	CONPAG=Prepago
60	DESCUENTO=10	60	DESCUENTO=10
61	MOTIVODESCUENTO=MOTIVO DESCUENTO	61	MOTIVODESCUENTO=MOTIVO DESCUENTO
62	#METODODEPAGO=tarjetaDeCredito	62	#METODODEPAGO=tarjetaDeCredito
63	METPAG=tarjetaDeCredito	63	METPAG=tarjetaDeCredito
64	#TIPODECOMPROBANTE=ingreso	64	#TIPODECOMPROBANTE=ingreso
65	DOCUMENTO=ingreso	65	
66	MONEDA=PESOS	66	MONEDA=PESOS
67	TIPOCAMBIO=1	67	TIPOCAMBIO=1
68	TIPOCAMBIO=1	68	TIPOCAMBIO=1
69	#SUFIJO=''	69	#SUFIJO=''
70	SUFIJO=M.N.	70	SUFIJO=M.N.
71	MONDOCTO=CFDI EN PESOS	71	MONDOCTO=CFDI EN PESOS
72	CARACTER=36	72	CARACTER=36
73	TOTALMN=536.40	73	TOTALMN=536.40
74	SUBTOTAL=500	74	SUBTOTAL=500
75	#TOTAL=570	75	#TOTAL=570
76	TOTAL=536.40	76	TOTAL=536.40
77	#AGREGADOS	77	#AGREGADOS

Figura 4.4.14. Eliminación del dato obligatorio DOCUMENTO que genera el dato TipoDeComprobante

[29-09-14 04:05:12] JA: Factura.tipoDeComprobante is null and required
 [sep 29 04:05:12] [main] ERROR ([Validator.java:231](#)) - Factura.tipoDeComprobante is null and required

Figura 4.4.15 Error obtenido a partir del dato faltante TipoDeComprobante

Cuando existen todos los datos requeridos se genera un comprobante válido.

Errores generados por errores geométricos ingresados desde el front.

Un requerimiento no mencionado por el SAT para los comprobantes fiscales pero válido desde el punto de vista contable y por tanto desde el punto de vista del usuario es que las cifras asentadas sean algebraicamente correctas, entre las validaciones requeridas se encuentran:

- Que por cada concepto el importe reportado sea el precio unitario por la cantidad.
- Que la suma de los importes de los conceptos sea igual al subtotal.

- Que la suma algebraica del subtotal, descuentos, impuestos trasladados e impuestos retenidos sean igual al Total.

Para probar este punto se cambiará el total en el archivo de prueba. Vea las figuras 4.4.16 y 4.4.17

63	METPAG=tarjetaDeCredito	63	METPAG=tarjetaDeCredito
64	#TIPODECOMPROBANTE=ingreso	64	#TIPODECOMPROBANTE=ingreso
65	DOCUMENTO=ingreso	65	DOCUMENTO=ingreso
66	MONEDA=PESOS	66	MONEDA=PESOS
67	TIPOCAMBIO=1	67	TIPOCAMBIO=1
68	TIPOCAMBIO=1	68	TIPOCAMBIO=1
69	#SUFIJO=''	69	#SUFIJO=''
70	SUFIJO=M.N.	70	SUFIJO=M.N.
71	MONDOCTO=CFDI EN PESOS	71	MONDOCTO=CFDI EN PESOS
72	CARACTER=36	72	CARACTER=36
73	TOTALMN=536.40	73	TOTALMN=536.40
74	SUBTOTAL=500	74	SUBTOTAL=500
75	#TOTAL=570	75	#TOTAL=570
76	TOTAL=536.40	76	TOTAL=536.00
77	#AGREGADOS	77	#AGREGADOS
78	SERFAC=JA	78	SERFAC=JA
79	IMPORTE_001=220	79	IMPORTE_001=220
80	IMPORTE_002=280	80	IMPORTE_002=280
81	TTRASLADADO=46.4	81	TTRASLADADO=46.4

Figura 4.4.16. Cambio del campo TOTAL

```
[sep 29 04:22:17] [main] ERROR (validaCFD.java:237) - Error de validación contable,
serie JA, folio 4, subTotal (500.0) - descuentos (10.0) + impuestos trasladados (46.4)
- impuestos retenidos (0.0) = (536.4) no es igual al total (536.0000)
[sep 29 04:22:17] [main] ERROR (FabricaDeCFDI.java:212) - El archivo E9T-JA4.xml
no superó alguna de las validaciones requeridas, no se intenta timbrar.
```

Figura 4.4.17. Error reportado por un error algebraico en las cifras asentadas

Pruebas de caja blanca.

Las pruebas de caja blanca o pruebas de caja de cristal o pruebas estructurales toman en cuenta el funcionamiento y procedimiento del software. Para generar estas pruebas se deben tomar distintos valores de entrada para examinar los

flujos que seguirán el programa y verificar que las salidas son las adecuadas y que corresponden a estas entradas.

La criptografía de dos llaves

La criptografía de dos llaves, de clave pública o de dos llaves es el método criptográfico que usan un par de claves para el envío de mensajes. Las dos llaves pertenecen a la misma entidad. Una llave es pública y se puede entregar a cualquier persona, la otra es privada y el propietario debe guardarla de modo que nadie tenga acceso a ella. Los métodos criptográficos garantizan que esa pareja de claves solo puede generarse una vez de tal modo que se asume que no es posible que dos entidades tengan casualmente la misma pareja de claves.

Aunque la llave pública contiene, en general, cualquier dato esta puede estar validada por alguna autoridad de certificación (CA o certification authority) que garantice que la información que contiene esta es verdadera, esta información va adjunta a esta llave y puede verificarse en cualquier momento de tal modo que se puede verificar que al ver la “firma” de validación de alguna autoridad de certificación podemos saber que la información contenida en la llave pública es real.

Dos utilidades de la criptografía de dos llaves son:

- Generar archivos cifrados y poder enviarlos por un medio no seguro con la seguridad de que aunque este mensaje sea interceptado no podrá ser descifrado.
- Validar la autenticidad e integridad de algún mensaje, es decir, que no ha sido alterado desde que fue firmado y que realmente fue emitido por quien supuestamente lo hizo.

La criptografía de dos llaves es usada dentro del ambiente de generación de CFDI's; el Sistema de Administración Tributaria como autoridad de certificación y basada en el método de criptografía de dos llaves otorga al Emisor un par de archivos y una contraseña para que puedan generar sus CFDI's; por una parte la

llave pública o archivo .cer y por otra parte la llave privada o archivo .key y la contraseña de la clave privada. Este juego de archivos y contraseña constituye el certificado de sello digital (o CSD) mediante los cuales se pueden generar CFDI's

Para ejecutar la prueba de caja blanca se verificó el elemento que permite al SAT comprobar que un CFDI: el sello del comprobante.

Generación y validación del sello digital del comprobante digital.

La generación del sello digital del comprobante se hace mediante los siguientes pasos:

- Una vez formado el XML (sin sello) se extraen cada uno de los datos del mismo y se genera una cadena con cada uno de estos datos separados por un carácter pipe; las particularidades y forma de acomodar los datos contenidos en esta cadena se encuentran en el anexo 20 de la Resolución de Miscelanea Fiscal para el 2014. Para facilitar la generación de esta se puede utilizar algún lenguaje/método computacional para transformar este archivo XML con un archivo de transformación XSLT que se encuentra publicado en http://www.sat.gob.mx/sitio_internet/cfd/3/cadenaoriginal_3_2/cadenaoriginal_3_2.xslt
- Una vez generada esta cadena se hace pasar por un método de digestión SHA1 que como tal hace virtualmente imposible recuperar el mensaje original y se obtiene una cadena de 160 bits.
- A esta cadena de 160 bits se cifra utilizando la llave privada del certificado de sello digital con lo que se obtiene una cadena binaria.
- Finalmente esta cadena binaria se convierte a base 64 con lo cual se obtienen caracteres imprimibles que se agregan al XML como el valor de la propiedad sello del nodo comprobante.

Debido a que el contenido de la llave pública también debe agregarse al XML como el valor del atributo certificado en el nodo comprobante el XML CFDI resulta ser:

- Infalsificable
- Inalterable
- Con un sello único
- Irrepudiable

Para obtener la cadena original del comprobante que es una sucesión de los datos contenidos en el mismo separados por el carácter “pipe” (|). La forma y reglas de generar este se encuentran en el ANEXO 20 de la Resolución Miscelánea Fiscal para 2014, publicada el 30 de diciembre de 2013

A partir de un documento de prueba obtuvimos la siguiente cadena original:

```
||3.2|2014-09-29T12:47:54|ingreso|Pago en una sola exhibición|060 DIAS FECHA  
FACTURA|1890.00|1.00|PESOS|1890.00|NA|Num.1132, Col. Apatlaco  
Deleg.Iztapalapa D.F. Mexico C.P. 09439|ETR820329K89|EMISOR: EDITORIAL  
TRILLAS,S.A.DE C.V.|Av.Rio Churubusco|Num.385|Col. General Pedro Maria  
Anaya|Delegacion. Benito Juarez|D.F.|Mexico|03340|Num.1132|Col.  
Apatlaco|Deleg.Iztapalapa|D.F.|Mexico|C.P. 09439|REGIMEN GENERAL DE LEY  
PERSONAS MORALES|AAB980814MC8|EL ARMARIO ABIERTO, SA. DE  
CV.|AGUSTIN MELGAR NO. 25|CONDESA DELEG.CUAUHTEMOC|D.  
F.|MEXICO|06140|10.00|PIEZA|9789682472640|TORICES-  
ORIENT.SEXUALIDAD|126.00|1260.00|10.00|PIEZA|9786071701459|TORICES-  
SEXUAL.DISCAPACIDAD|63.00|630.00|IVA|0.00|ISR|0.00|0.00|IVA|0.00|0.00|0.00  
||
```

Al aplicar el método de procesamiento SHA-1 a esta cadena original se obtiene la siguiente cadena:

4A2A66761C09B7B7AA06902C104C1B3F7760569F

Se utiliza la clave privada correspondiente al certificado digital y se obtiene el siguiente mensaje ya convertido a caracteres imprimibles:

```
U1+UBYDVzVMOXamRtoePkAxq6tcKGBmAd4i2VqzAN0HUDdpxLAcNGB4GAgK
dz2sJNf4fkiJTDKUcCoVqhRgJddex78vFLSvKI3u9NL0RJbQGOjytk78p+guc5OUU
oReLQQ0MBDguNzEysijpRIamdce4jBulhFUio+XhzxoSAkw=
```

Como ya mencionamos, la criptografía asimétrica permite saber si un mensaje ha sido efectivamente generado por el emisor y que además no ha sido alterado; para ello requiere los siguientes datos:

- El mensaje firmado con la llave privada del emisor, para los CFDI's el valor del campo sello en el nodo comprobante (y para la validación grabado en un archivo mensaje.txt)
- La llave pública perteneciente al emisor (para este ejemplo, leído desde el XML e ingresado a un archivo certificado .pem)

Utilizando el software openssl se utiliza el siguiente comando:

```
openssl dgst -sha1 -verify certificado.pem -signature mensaje.txt
```

Y obtenemos la respuesta "Verify OK" lo que demuestra que el mensaje efectivamente ha sido generado con el certificado en cuestión y que además el mensaje no ha sido alterado.

Para verificar que este certificado pertenece a una entidad válida y certificada por la entidad de certificación, en este caso el SAT escribimos el contenido de la propiedad certificado del nodo Comprobante en un archivo con la extensión CRT, al abrirlo el lector de certificados del sistema operativo muestra a quien pertenece y quien ha firmado este archivo (vea la figura 4.4.18)

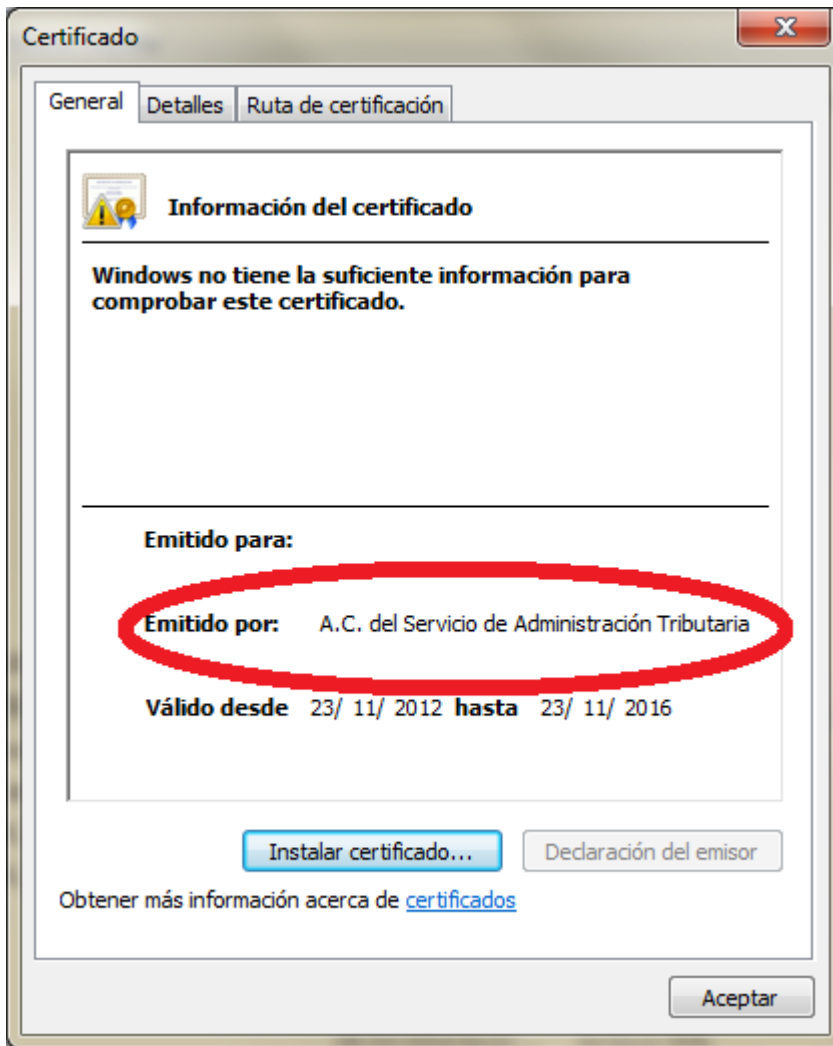


Figura 4.4.18. Información de la llave pública del CSD.

Comprobación Adicional.

Debido a que los validadores del SAT únicamente dan como resultado “comprobante válido o comprobante no válido” para validar este comprobante pasando por alto la salida de estas operaciones hacia el usuario se utiliza el software externo “ValidaCFD” (Freeware distribuido desde la página <http://www.validacfd.com/index.php?load=default>) versión 1.0.140823 obteniendo los mismos resultados. Vea figura 4.4.19

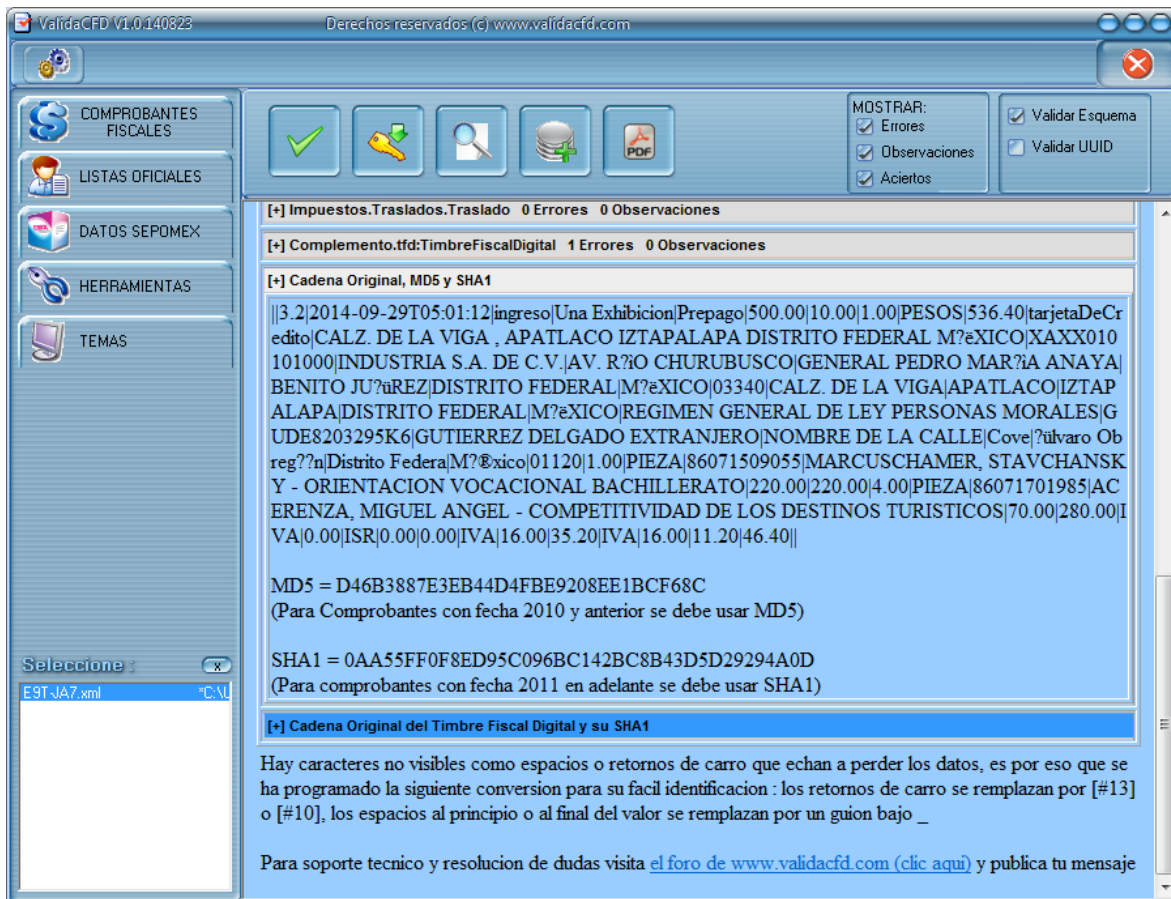


Figura 4.4.19. Utilización del freeware ValidaCFD para comprobar el proceso de generación del sello del CFDI

Por tanto el CFDI es válido y la prueba de caja blanca se superó correctamente porque el proceso interno está generando el XML en este paso correctamente.

Pruebas de volumen.

La prueba de volumen sujeta al elemento de prueba a grandes cantidades de datos, para determinar si se alcanzan los límites que hacen fallar al software. También identifica la carga máxima continua o volumen que el elemento de prueba puede manejar por un período dado.

Para generar pruebas de volumen al sistema se hará la carga y petición de generar simultáneamente 316 archivos para generar CFDI's, la programación del código permitirá tomar un archivo a la vez y el bloqueo del mismo mediante

archivos que indiquen que el archivo está en uso para que ninguna otra instancia del programa pueda tomarlo mientras este se está procesando, se mide el tiempo de respuesta para cada petición encolada y el tiempo de respuesta general para todas las peticiones, también se verifica que el programa no arroje datos erróneos dado que los archivos son correctos así como que los archivos en la prueba no sean tomados por más de una instancia del programa general. Vea figuras 4.4.20 y 4.4.21

Las salidas en la consola son como se muestran:

```
Generando CFDI archivo NJA5011200.TXT 1/316
Generada correctamente, tiempo para este comprobante: 2161 milisegundos
Generando CFDI archivo NJA5011202.TXT 2/316
Generada correctamente, tiempo para este comprobante: 444 milisegundos
Generando CFDI archivo NJA5011204.TXT 3/316
Generada correctamente, tiempo para este comprobante: 363 milisegundos
Generando CFDI archivo NJA5011205.TXT 4/316
Generada correctamente, tiempo para este comprobante: 377 milisegundos
Generando CFDI archivo NJA5011206.TXT 5/316
Generada correctamente, tiempo para este comprobante: 345 milisegundos
Generando CFDI archivo NJA5011207.TXT 6/316
Generada correctamente, tiempo para este comprobante: 322 milisegundos
Generando CFDI archivo NJA5011208.TXT 7/316
Generada correctamente, tiempo para este comprobante: 326 milisegundos
Generando CFDI archivo NJA5011213.TXT 8/316
Generada correctamente, tiempo para este comprobante: 313 milisegundos
Generando CFDI archivo NJA5011214.TXT 9/316
Generada correctamente, tiempo para este comprobante: 295 milisegundos
Generando CFDI archivo NJA5011215.TXT 10/316
Generada correctamente, tiempo para este comprobante: 300 milisegundos
Generando CFDI archivo NJA5011218.TXT 11/316
Generada correctamente, tiempo para este comprobante: 275 milisegundos
Generando CFDI archivo NJA5011219.TXT 12/316
Generada correctamente, tiempo para este comprobante: 299 milisegundos
Generando CFDI archivo NJA5011220.TXT 13/316
Generada correctamente, tiempo para este comprobante: 285 milisegundos
Generando CFDI archivo NJA5011221.TXT 14/316
Generada correctamente, tiempo para este comprobante: 276 milisegundos
Generando CFDI archivo NJA5011222.TXT 15/316
Generada correctamente, tiempo para este comprobante: 255 milisegundos
Generando CFDI archivo NJA5011223.TXT 16/316
```

Figura 4.4.20 Primer página de las pruebas de volumen

```
-----
Generada correctamente, tiempo para este comprobante: 165 milisegundos
Generando CFDI archivo NJA5011530.TXT 303/316
Generada correctamente, tiempo para este comprobante: 183 milisegundos
Generando CFDI archivo NJA5011531.TXT 304/316
Generada correctamente, tiempo para este comprobante: 174 milisegundos
Generando CFDI archivo NJA5011532.TXT 305/316
Generada correctamente, tiempo para este comprobante: 163 milisegundos
Generando CFDI archivo NJA5011533.TXT 306/316
Generada correctamente, tiempo para este comprobante: 338 milisegundos
Generando CFDI archivo NJA5011534.TXT 307/316
Generada correctamente, tiempo para este comprobante: 160 milisegundos
Generando CFDI archivo NJA5011536.TXT 308/316
Generada correctamente, tiempo para este comprobante: 161 milisegundos
Generando CFDI archivo NJA5011538.TXT 309/316
Generada correctamente, tiempo para este comprobante: 176 milisegundos
Generando CFDI archivo NJA5011539.TXT 310/316
Generada correctamente, tiempo para este comprobante: 157 milisegundos
Generando CFDI archivo NJA5011542.TXT 311/316
Generada correctamente, tiempo para este comprobante: 161 milisegundos
Generando CFDI archivo NJA5011543.TXT 312/316
Generada correctamente, tiempo para este comprobante: 160 milisegundos
Generando CFDI archivo NJA5011544.TXT 313/316
Generada correctamente, tiempo para este comprobante: 247 milisegundos
Generando CFDI archivo NJA5011545.TXT 314/316
Generada correctamente, tiempo para este comprobante: 157 milisegundos
Generando CFDI archivo NJA5011546.TXT 315/316
Generada correctamente, tiempo para este comprobante: 175 milisegundos
Generando CFDI archivo NJA5011547.TXT 316/316
Generada correctamente, tiempo para este comprobante: 207 milisegundos
Generada fin de la ejecución, tiempo total: 72 segundos
```

Figura 4.4.21. Última página de las pruebas de volumen

El tiempo promedio para generar cada comprobante es de aproximadamente 165 milisegundos, el tiempo total para generar todos los comprobantes es de 72 segundos pero este tiempo puede verse seriamente modificado hasta 6 segundos por cada archivo a procesar dependiendo de la velocidad de la conexión a Internet entre el cliente, el PAC y su tiempo de respuesta.

Pruebas de validación. Vea figura 4.4.22



Figura 4.4.22. Pruebas de validación en el SAT

Ir a la página del SAT en la liga: <https://siat.sat.gov.mx/PTSC/>

Y seleccionar en la opción de “Factura” electrónica del lado izquierdo, la sub-opción “Verificación”, dentro de la opción “Verificación”, existen tres opciones: vea figuras 4.4.23 y 4.4.24

- Factura Electrónica (CFDI)
- Factura en Papel
- Comprobante digital

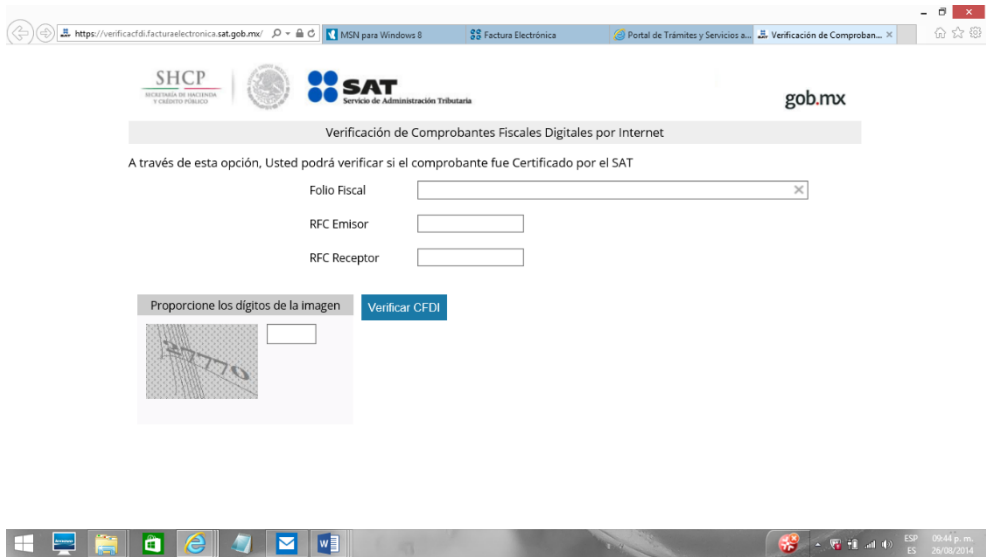


Figura 4.4.23. Pruebas de validación de comprobante.

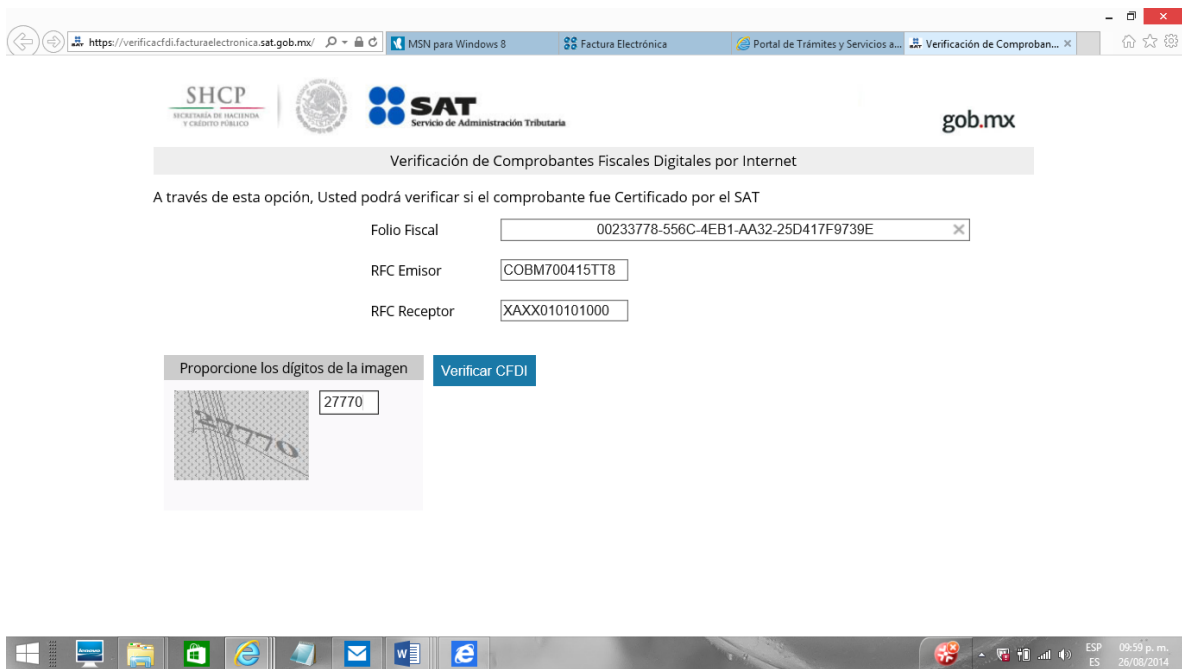


Figura 4.4.24. Pruebas de validación de comprobante con datos.

Mantenimiento

El mantenimiento del software es la modificación de un producto de software después de la entrega a producción para corregir fallas, para mejorar su rendimiento u otros atributos o para adaptar el producto a un entorno modificado.

Una solicitud de cambio a veces se refiere a los problemas del sistema que deben abordarse con mucha urgencia. Estos cambios urgentes pueden surgir por tres razones:

- La aparición de una falla del sistema que debe ser reparado para permitir el funcionamiento normal de la operación.
- Los cambios ambientales que tienen efectos inesperados en el sistema
- Cambios de negocio imprevistos que podrían deberse a la aparición de un nuevo competidor en el mercado

La vida de un sistema no termina con la entrega del "sistema final", está sujeto a continuos cambios.

Hay varios tipos de mantenimiento de software:

Correctivo

Se utiliza universalmente para referirse al mantenimiento de reparación de averías, a medida que ocurren las fallas, se pone a consideración y se busca la causa de la falla, haciéndose las correcciones necesarias. Este tipo de reparaciones es temporal, como para mantener el sistema funcionando, pero no con el arreglo óptimo, hasta que se logra arreglar la falla completamente.

Preventivo

El mantenimiento preventivo de software es el proceso por el cual se mejora y optimiza el software que se ha instalado. Este mantenimiento se realiza para la

prevención de posibles problemas que puedan llegar a surgir en el ambiente productivo.

Aumentativo

Se refiere a los cambios para incorporar, modificar o eliminar funciones de un sistema para atender ampliación o cambio en las necesidades del usuario.

Adaptativo

Algunas veces, un cambio introducido en una parte de un sistema requiere cambios en otras partes del sistema. La implementación de estos cambios secundarios se le conoce como el mantenimiento adaptativo.

Se refiere a cambios que se requieren realizar al software para adaptarse a los cambios en el entorno o ambiente, por ejemplo, si hay una actualización de hardware o nuevas versiones de Sistema operativo. El mantenimiento adaptativo no está relacionado con cambios en la funcionalidad del sistema.

Perfectivo

Principalmente se ocupa de atender las nuevas necesidades de los usuarios o nuevos requerimientos. Se refiere a mejoras funcionales al sistema. Mantenimiento perfectivo también incluye actividades para optimizar el rendimiento de sistema o para mejorar la interfaz de usuario.

Son las acciones llevadas a cabo para mejorar la calidad interna de los sistemas en cualquiera de sus aspectos: reestructuración del código de definición más clara del sistema, y optimización del rendimiento y de la eficiencia.

Horizontal

Consiste en el mantenimiento realizado a la infraestructura de hardware sobre el cual está corriendo el sistema.

Vertical

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 del Sistema Operativo
- Instalación de fixes (parches de la aplicación).
- Incremento de mejoras y nuevas capacidades del software.
- Homologación de versiones.
- Adaptación a cambios de ambiente (por ejemplo, proveedores de Internet).

4.5 GENERACIÓN DE CFDI DE PRUEBA Y EVALUACIÓN DE LOS MISMOS

La aplicación generará un Comprobante Fiscal Digital por Internet en formato PDF y XML, el resultado final se obtendrá mediante el envío de los datos al proveedor autorizado de certificación y el Servicio de Administración tributaria.

La siguiente imagen muestra el primer proceso que se lleva acabo, el cual consiste en la captura de datos del receptor, generando un archivo TXT. Figura 4.5.1

Acciones

Limpiar Campos

Concepto

Identificador:

Descripcion:

Cantidad:

Unidad:

Valor Unitario:

Modo

Receptor

Nombre:

RFC:

RFCE:

Cuenta:

Telefono:

Fax:

E-Mail:

Domicilio

Calle:

Interior:

Exterior:

Colonia:

C.P.:

Estado:

Localidad:

Pais:

Modo

Factura

Forma de pago:

Cond de pago:

Descuento:

Motivo Descuento:

Metodo de pago:

Comprobante:

Moneda

Tipo de Moneda:

Tipo de Cambio:

Notificaciones

Para iniciar debes agregar al menos un producto

Figura 4.5.1. Captura de datos

Al obtener el archivo TXT con el formato llave valor, se genera el objeto factura el cual se ingresa a la base de datos mediante el framework Hibernate, con esto se obtiene el archivo XML. Vea la Figura 4.5.2

Figura 4.5.3. Timbrado del archivo XML.


```

<?xml version="1.0" encoding="UTF-8"?>
- <cfdi:Comprobante xsi:schemaLocation="http://www.sat.gob.mx/cfd/3 http://www.sat.gob.mx/sitio_internet/cfd/3/cfdv32.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:cfdi="http://www.sat.gob.mx/cfd/3" version="3.2" total="154.50" tipoDeComprobante="egreso" subTotal="154.50" serie="JAA"
  sello="MNdVmN0jD+ySJGg9tZmLGKVOoijL7pNM5I9k5Qu4fXHGa46ey/2MLouPfrGEIz4Njx+VebtW+ZbVfgmc3pyyBoo+edImBWsBX9w5zPTW4TjvXNjHqz50+m3cYbdSgohe3G2Dv51C74/OOwaV+yG5L2mRa7T2+9QkyauJVa8j]
  noCertificado="00001000000202415670" motivoDescuento="SOLO FACT. DERECHO DEV." metodoDePago="NA" formaDePago="Pago en una sola exhibicion" folio="1" fecha="2014-05-27T10:01:01"
  certificado="MIIEhjCCA26gAwIBAgIUMDAwMEwMDAwMDAyMDIOMTU2NzAwDQYJKoZIhvcNAQEFBQAwggGVMtGwNgYVYQQDDC9BLLkMuIGRlbcBTZjZaWNPbyBkZSBBZG1pbmlzHjY2nDs24gVHJpYnV0YXJpYTEvMC0GA1UE
  TipoCambio="2510.00" Moneda="DOLARES" LugarExpedicion="Calz. de la viga 1132, Apatlaco Iztapalapa Distrito Federal México 09439">
  - <cfdi:Emisor rfc="ETR820329K89" nombre="EDITORIAL TRILLAS S.A. DE C.V.">
    <cfdi:DomicilioFiscal pais="México" noExterior="385" municipio="Benito Juárez" estado="Distrito Federal" colonia="General Pedro María Anaya" codigoPostal="03340" calle="Av. Río Churubusco"/>
    <cfdi:ExpedidoEn pais="México" noExterior="1132" municipio="Iztapalapa" estado="Distrito Federal" colonia="Apatlaco" codigoPostal="09439" calle="Calz. de la viga"/>
    <cfdi:RegimenFiscal Regimen="Régimen General de Ley Personas Morales"/>
  </cfdi:Emisor>
  - <cfdi:Receptor rfc="XEXX010101000" nombre="EDIT. TRILLAS DE COLOMBIA LTDA">
    <cfdi:Domicilio pais="COLOMBIA" colonia="APARTADO AEREO 15-151" calle="CARRERA 15 NUM. 33-A-35" localidad="BOGOTA"/>
  </cfdi:Receptor>
  - <cfdi:Conceptos>
    <cfdi:Concepto valorUnitario="10.50" unidad="PIEZA" noIdentificacion="9789682450655" importe="10.50" descripcion="COMO HACER-PINTURA AUTOMOTRIZ" cantidad="1.00"/>
    <cfdi:Concepto valorUnitario="5.50" unidad="PIEZA" noIdentificacion="9789682411038" importe="44.00" descripcion="F.A.O.-TALLER DE FRUTAS" cantidad="8.00"/>
    <cfdi:Concepto valorUnitario="6.50" unidad="PIEZA" noIdentificacion="9789682436895" importe="52.00" descripcion="F.A.O.-ELAB.PROD.CARNICOS" cantidad="8.00"/>
    <cfdi:Concepto valorUnitario="5.00" unidad="PIEZA" noIdentificacion="9789682436888" importe="40.00" descripcion="F.A.O.-SUBPRODUCTOS.ANIMALES" cantidad="8.00"/>
    <cfdi:Concepto valorUnitario="8.00" unidad="PIEZA" noIdentificacion="9789682461132" importe="8.00" descripcion="WILLIAMS-DICC.EDUC.ESPECIAL" cantidad="1.00"/>
  </cfdi:Conceptos>
  - <cfdi:Impuestos totalImpuestosTrasladados="0.00" totalImpuestosRetenidos="0.00">
    - <cfdi:Retenciones>
      <cfdi:Retencion importe="0.00" impuesto="IVA"/>
      <cfdi:Retencion importe="0.00" impuesto="ISR"/>
    </cfdi:Retenciones>
    - <cfdi:Traslados>
      <cfdi:Traslado importe="0.00" impuesto="IVA" tasa="0.00"/>
    </cfdi:Traslados>
  </cfdi:Impuestos>
  - <cfdi:Complemento>
    <cfdi:TimbreFiscalDigital xsi:schemaLocation="http://www.sat.gob.mx/TimbreFiscalDigital http://www.sat.gob.mx/TimbreFiscalDigital/TimbreFiscalDigital.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
      instance" version="1.0" xmlns:td="http://www.sat.gob.mx/TimbreFiscalDigital"
      selloSAT="QivuN2YZZP0H5/7aRnA3C0MWG8g3008HZzvi60Q+uSwHW3T0CpNciHR7TjaXoVNW3UCFh/DyFpp14Cjy4jI/4IHbC9L2AIrQboXeHncSWjGfRdDEI6CvYvAKvi0Ry1pAGtmAg78XAwqYuDf570Rq8gzMagV
      selloCFD="MNdVmN0jD+ySJGg9tZmLGKVOoijL7pNM5I9k5Qu4fXHGa46ey/2MLouPfrGEIz4Njx+VebtW+ZbVfgmc3pyyBoo+edImBWsBX9w5zPTW4TjvXNjHqz50+m3cYbdSgohe3G2Dv51C74/OOwaV+yG5L2mRa7T2
      noCertificadoSAT="00001000000202864883" UUID="3647b64e-4de3-43c0-affb-cc01793a2dbc" FechaTimbrado="2014-05-27T10:01:13"/>
    </cfdi:Complemento>
  </cfdi:Comprobante>

```


Recibiendo el archivo XML ya timbrado se actualiza el objeto factura, se realiza la actualización de la base de datos y a partir de ahí, ya se puede obtener una representación impresa de un CDFI, la cual se muestra en la figura 4.5.4


29 CADAR DE PRODUCCION
32 REPORTES VIGAS COMERCIALIZACION



Deposito en sucursal:
Caja de la viga 5332
Apartado
Distrito Federal Mexico
CP: 06439 Zetoculapala
TEL: 06339990
FAX: 06339970

RECIBO:
EDITORIAL TRILLAS S.A. DE C.V.
Av. Rio Guadalupe 280
General Pedro Maria Anaya
CP: 03240 Bello Juarez
Mexico, D.F., 067020329003
TEL: 06004233 FAX: 06041064

FECHA Y HORA DE TIMBRADO **DEVOLUCION**

27/05/2014 10:01:13	JAA 1 FOLIO INTERNO 
---------------------	--

Este documento es una representación impresa de un CDFI

Página 1 de 1

Régimen General de Ley Personas Morales Tipo de comprobante: egreso

Forma de pago: Pago en una sola exhibición Método de Pago: NA

RECEPTOR CUENTA 5842/44/29	NUMERO: 41300
EDIT. TRILLAS DE COLOMBIA LTDA CARRERA 15 NUM. 33-A-35 APARTADO AEREO 15-151 BOGOTA COLOMBIA RFC: XE0010101000	FECHA: 2014-04-23 RECIBO: COLOMBIA SOLICITO: COLOMBIA AUTORIZO: ELVIA GRANDE OBSERVACIONES: DEVS. DE CUENTES 4TO. TRIMESTRE 2013

301 TRILLAS COLOMBIA 'T'		5000912
REFERENCIA DE PEDIDO	DE FECHA	TIPO DE PAGO
SOLO FACT. DERECHO DEV.	2014-04-23	
ENVIADO POR	PLAZO DE PAGO	VENCIMIENTO
	NUM GUIA	FECHA GUIA
	BULTOS	PESO BRUTO
		PESO NETO

L.P.	ISSN	EDICION	AUTOR TITULO	EJEMPLARES		PRECIO		IMPORTE	IVA
				UNIDAD	BURTIDOS	LISTA	NETO		
00	9789682450655	1	COMO HACER-PINTURA AUTOMOTRIZ	PIEZA	1	21.00	10.50	10.50	
00	9789682411038	1	F.A.O.-TALLER DE FRUTAS	PIEZA	8	11.00	8.80	44.00	
00	9789682436895	2	F.A.O.-ELAB. PROD. CARNICOS	PIEZA	8	13.00	10.40	52.00	
00	9789682436888	2	F.A.O.-SUBPRODUCTOS ANIMALES	PIEZA	8	10.00	8.00	40.00	
00	9789682461132	1	WILLIAMS-DICC EDUC ESPECIAL	PIEZA	1	16.00	8.00	8.00	

----- APLICACION -----

NFAC. 1 JA 223	IMPORTE 154.50
-------------------	-------------------

Articulos 5 Unidades 25 Importe Libros: \$154.50	Sub Total \$ 154.50
	- Descuentos \$ 0.00
	+ Imp. Traslados \$ 0.00
	- IVA Retenido \$ 0.00
	- ISR Retenido \$ 0.00
	= Importe Neto \$ 154.50

NOTA DE REINGRESO EN DOLARES. | Tipo de cambio: 2510.00000 | Equivalencia en M.N. \$1932.79

Fecha y hora de generación: 2014-05-27T10:01:01

Número de serie del CSD del Emisor: 00001000000202415670

Número de serie del Certificado del SAT: 00001000000202864883

Folio Fiscal - UUID: 3647b64e-4de3-43c0-afb-c0c1793a20b0c

Sello del SAT:
CvUaZyZ2P0H57aRnA3C0MwG6g3008HZzW60Q+u0wHW3TOcxPnciHR7TjxwVNV3UCFvDyFp14CJg4j3I4IHBC9L2AIf00x0eHN
c0VjGfR0DEI6CvYvAKv0Ry1pAGfAg7dXAqYv0R57CRqgZMagVAGINBaf1v1g+

Sello del CDFI:
MNdVmN0D+y2JGgK2mLgKv00jL7pNmSf9k5Qu4rKHGa45ey2MLouPFRGz4Njx+VebW+ZbVfmgc3pyyBoo+edimBWsBX9w5zPTW4TjvXNjHqzqO+m3cYb0g0ne3
G2Dv51C7400aV+ygS2mRa7T2+9QkyuVv95Jc+

Cadena original del complemento de certificación:
||1|0|3647b64e-4de3-43c0-afb-c0c1793a20b0c|14-05-27T10:01:13|MNdVmN0D+y2JGgK2mLgKv00jL7pNmSf9k5Qu4rKHGa45ey2MLouPFRGz4Njx+VebW+ZbVfmgc3pyyBoo+edimBWsBX9w5zPTW4TjvXNjHqzqO+m3cYb0g0ne3|G2Dv51C7400aV+ygS2mRa7T2+9QkyuVv95Jc+|000100000002864883|

REG. CANARA DE LA IND. EDITORIAL No. 158
CEO. DE EMPADRONAMIENTO 1036/931
REG. EXP/IMP. 40623-001

COPIA

Para obtener los archivos de este CDFI visite:
trilfacturacion.com.mx

Figura 4.5.4. Representación impresa de un CDFI

CONCLUSIONES

-
- El desarrollo del Sistema para la Generación de Comprobantes Fiscales Digitales por Internet, cumple a cabalidad los requerimientos solicitados por el cliente y los requerimientos solicitados por el Sistema de Administración Tributaria.
 - La generación de los Comprobantes Fiscales Digitales por Internet, conlleva la utilización de recursos de cómputo para realizar tareas muy demandantes, que sería imposible realizar a mano como las operaciones matemáticas de encriptación.
 - Utilizar un equipo de cómputo permite al usuario tener un mayor control sobre sus operaciones comerciales y llevar de forma más transparente y ordenada la contabilidad de la empresa, contribuyendo al mejoramiento de administración de la misma.
 - La digitalización de los procesos del SAT lo encamina a ser una de las instituciones base del desarrollo económico del país, al imponer un estándar en la comprobación de ingresos y egresos de personas físicas y morales.
 - Realizar un cambio en la forma de comprobar movimientos de dinero dentro de las empresas, representa un reto, ya que las personas encargadas se rehúsan a aceptar nuevos estándares, máxime si ello les representa mayor control de parte de las instituciones de gobierno sobre ellos.
 - Utilizar herramientas de software libre nos permite abatir los costos finales del desarrollo de un proyecto, pero siempre teniendo en cuenta usar software libre plenamente probado y con una base sólida, que le permita contar con actualizaciones y desarrollo constante.
 - La utilización de Internet como medio de conexión entre los distintos sistemas involucrados (SAT – PAC – Usuario), da mayor completitud al uso de Internet, como gran red más allá de los usos más exhaustivos que se le dan como “pantalla para presentar contenido”.
 - La infraestructura prestada por los proveedores de servicio de Internet es

aún insuficiente ya que, aunque los distintos actores que intervienen en la generación de un CFDI, cuenten con la infraestructura suficiente, en ciertos momentos las conexiones proveídas por las compañías no tienen la capacidad de soportar el intercambio de datos, generando errores inesperados que es necesario controlar.

- El sistema permite eliminar el error humano en las operaciones aritméticas involucradas en la generación de un CFDI.
- La utilización del esquema cliente – servidor en la aplicación, así como una interfaz web, nos permite tener un producto que se puede utilizar en una virtual infinidad de terminales utilizando el mismo software; esto permite tener consistencia total en los productos finales generados por la aplicación.
- Una interfaz gráfica es un punto importante que le da mejor presentación a la aplicación y la hace más amigable al usuario final.
- En general, la automatización de procesos dentro del comercio y en particular en la generación de CFDI's proporciona mayor fiabilidad en los datos, producidos en las transacciones monetarias, tanto para empresas, sus clientes y la autoridad responsable de recolectar los impuestos.
- La mayoría de las empresas cuentan ya con algún equipo de cómputo para auxiliar algunos de los procesos que deben llevar a cabo en su operación diaria, por ello es altamente probable que no deban realizar mayor gasto para poder tener funcionando este sistema de generación de CFDI.
- Se ha puesto de manifiesto la importancia de contar con un equipo interdisciplinario para emprender un proyecto como este, ya que no es posible que una sola persona domine los diferentes aspectos en los que hay que involucrarse y tener conocimiento pleno.
- La formación y conocimientos que se nos impartió en las aulas fue fundamental para la realización exitosa de este proyecto, no sólo los conocimientos ofrecidos sino las experiencias y puntos claves que cada profesor nos mostró en su momento.

BIBLIOGRAFÍA

Mesografía

1. En estas páginas se obtuvo información acerca de la facturación electrónica:

http://www.sat.gob.mx/informacion_fiscal/factura_electronica/Paginas/default.aspx

http://www.sat.gob.mx/informacion_fiscal/factura_electronica/Documents/cfdi/PyRF

actElect.pdf

http://dof.gob.mx/nota_detalle.php?codigo=5332803&fecha=17/02/2014

<http://www.definicion.org/miscelanea-fiscal>

Consulta: Julio 2014

2. En estas páginas se obtuvo información acerca de los términos utilizados por el SAT:

<http://blog.vital.com.mx/?p=72>

http://www.sat.gob.mx/informacion_fiscal/factura_electronica/Documents/cfdi/ManualUsuarioPortalPrivado.pdf

Consulta: Julio 2014

3. En esta página se obtuvo información acerca de la normatividad de los PAC's:

http://www.sat.gob.mx/informacion_fiscal/factura_electronica/Paginas/requisitos_pac.aspx

Consulta: Julio 2014

4. En estas páginas se obtuvo información acerca del contenido básico de un CFDI:

http://www.sat.gob.mx/informacion_fiscal/factura_electronica/Paginas/requisitos_impresion_cfdi.aspx

<http://www.facturalight.com/elementos>

<http://www.infox.mx/datosinfxpedia.php?id=161&palabra=facturacion>

<http://factura-electronica-mexico.com/2012/01/20/representaciones-impresas-cfd-cfdi/>

<http://www.sifiscal.com.mx/sifactura/factuelect.htm>

Consulta: Julio 2014

5. En estas páginas se obtuvo información acerca de las bases de datos relacionales:

https://es.wikipedia.org/wiki/Base_de_datos_relacional

<http://www.jorgesanchez.net/bd/bdrelacional.pdf> - Autor: Jorge Sánchez

Consulta: Julio 2014

6. En estas páginas se obtuvo información cerca de POO en JAVA:

http://www.ciberaula.com/articulo/tecnologia_orientada_objetos/

<http://fcasua.contad.unam.mx/apuntes/interiores/docs/98/opt/java.pdf>

<http://sistemascanar.blogspot.mx/2011/11/programacion-orientada-objetos-poo.html>

Consulta: Julio 2014

7. En este documento se consultó información relevante al entorno JAVA:

http://www.americati.com/doc/ventajas_c.pdf

Consulta: Julio 2014

8. En estas páginas se obtuvo información acerca del SGBD MySQL:

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

<http://dev.mysql.com/doc/refman/5.0/es/choosing-version.html>

<http://es.wikipedia.org/wiki/SQL>

http://es.wikipedia.org/wiki/Hilo_de_ejecuci%C3%B3n

Consulta: Julio 2014

9. En las siguientes páginas se obtuvo información de la arquitectura Cliente-Servidor:

<http://www.ecured.cu/index.php/Cliente-Servidor>

<http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml#elemen>
http://docente.ucol.mx/sadanary/public_html/bd/cs.htm
<http://es.wikipedia.org/wiki/Cliente-servidor>
http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/marquez_a_bm/capitulo5.pdf
<http://es.scribd.com/doc/29633032/Redes-i-Convktividad-II>
Consulta: Julio 2014

10. En el siguiente documento electrónico se consultó información relevante al tema:

<http://www.cs.cinvestav.mx/tesisGraduados/2012/tesisRodrigoJurado.pdf>
Consulta: Agosto 2014

11. En las siguientes páginas se obtuvo información acerca de requerimientos:

<http://es.scribd.com/doc/37187866/Requerimientos-funcionales-y-no-funcionales>
<http://definicion.de/requerimiento/>
<http://www.alegsa.com.ar/Dic/requerimientos.php>
http://www.frsf.utn.edu.ar/matero/visitante/bajar_apunte.php?id_catedra=117&id_apunte=3591
Consulta. Agosto 2014

12. En la siguiente página se obtuvo información acerca de PHP:

<https://php.net/manual/es/index.php>
<http://php.net/manual/es/intro-whatcando.php>
Consulta. Agosto 2014

13. En la siguiente página se obtuvo información acerca de SGBD Oracle:

<http://docs.oracle.com/javase/7/docs/api/>
Consulta. Agosto 2014

14. En la siguiente página se obtuvo información del lenguaje de programación C#:

<http://msdn.microsoft.com/es-MX/library/zkxk2fwf%28v=vs.90%29.aspx>
Consulta: Agosto 2014

15. En la siguiente página se obtuvo información acerca del SGBD MySQL y Firebird:

<http://ocelot.ca/blog/blog/2014/03/09/354/15>
Consulta: Agosto 2014

16. En las siguientes páginas se obtuvo información acerca del modelo entidad-relación:

<http://www.desarrolloweb.com/articulos/modelo-entidad-relacion.html>
http://www.ecured.cu/index.php/Diagrama_Entidad_Relaci%C3%B3n
<http://www.genbetadev.com/bases-de-datos/fundamento-de-las-bases-de-datos-modelo-entidad-relacion>
Consulta: Agosto 2014

17. En las siguientes páginas se obtuvo información sobre diccionario de datos:

<http://es.slideshare.net/jorgemgr94/diccionario-de-datos-25151513>
http://www.uaeh.edu.mx/docencia/P_Presentaciones/huejutla/sistemas/base_datos2/Diccionario_datos.pdf
<http://www.desarrolloweb.com/faq/452.php>
Consulta: Agosto 2014

18. En las siguientes páginas se consultó información de la administración MySQL:

<http://dev.mysql.com/doc/refman/5.0/es/create-procedure.html>
<http://dev.mysql.com/doc/refman/5.0/es/create-trigger.html>
Consulta: Agosto 2014

19. En las siguientes páginas se consultó información acerca de tipo de pruebas y mantenimiento:

http://es.wikipedia.org/wiki/Prueba_unitaria

http://es.wikipedia.org/wiki/Pruebas_de_caja_blanca

[http://es.wikipedia.org/wiki/Caja_negra \(sistemas\)](http://es.wikipedia.org/wiki/Caja_negra_(sistemas))

http://es.wikipedia.org/wiki/Pruebas_de_integraci%C3%B3n

http://es.wikipedia.org/wiki/Pruebas_de_regresi%C3%B3n

http://es.wikipedia.org/wiki/Pruebas_de_rendimiento_del_software

<http://pruebasalfaybeta.blogspot.mx/>

http://es.wikipedia.org/wiki/Mantenimiento_correctivo

http://es.wikipedia.org/wiki/Mantenimiento_preventivo

<http://www.aesist.com/soporte/mantenimiento-perfectivo>

Bibliografía

1. Título: Programación orientada a objetos

Autor: Joyanes

Editorial: Mc Grow Hill

Año: 1998

Pag:28

2. Título: Análisis estructurado moderno

Autor: Edward Yourdon

Editorial: Prentice Hall

Año: 2007

Edición: 1ra Edición

3. Título: Análisis y diseño de sistemas de información

Autor: Jeffrey L. Whitten / Lonnie D. Benetley / Victor M. Barlow

Editorial: IRWIN

Año: 1996

Edición: 3ra Edición

4. Título: Análisis y diseño de sistemas de sistemas

Autor: Kenneth E. Kendall / Julie E. Kendall

Editorial: Prentice Hall Hispanoamericana, S. A.

Año: 1991

Edición: 1ra Edición en Español

5. Título: Structured Design

Autor: Edward Yourdon / Larry L. Constantine

Editorial: YOURDON inc.

Año: 1978

Edición: 2da Edición

6. Título: Ingeniería del Software

Un enfoque práctico

Autor: Roger S. Pressman

Editorial: McGraw-Hill

Año: 2006

Edición: 6ta Edición

7. Título: Software Engineering

Autor: Ian Sommerville

Editorial: Pearson Education Limited

Año: 2001 (6ta edición).

8. Título: Software Engineering (Principles and Practice)

Autor: Hans van Vliet

Editorial: John Wiley & Sons

Año: 1993

9. Título: Software Engineering

Autor: Shari Lawrence Pfleeger

Editorial: Macmillan Publishing Company

Año: 1991

Lugar: United States of America

10. Título: Ingeniería del Software

Un enfoque desde la guía SWEBOK

Autor: Salvador Sánchez/Miguel Ángel Sicilia/Daniel Rodríguez

Editorial: Alfaomega Grupo Editor, S. A. de C.V.

Lugar: Madrid, España

Año: 2012
