



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

Facultad de Ciencias de la Computación

---

**SISTEMA DE ADMINISTRACIÓN ELECTRÓNICA DE  
COMPROBANTES FISCALES DIGITALES POR INTERNET (SADEC)**

Presenta: Alejandro Yáñez García

TESIS PROFESIONAL PARA OBTENER EL TÍTULO DE:  
INGENIERO EN CIENCIAS DE LA COMPUTACIÓN

ASESOR:

M. C. Marco Antonio Soriano Ulloa

Puebla, Puebla. Marzo 2015



*A mis padres Silvia Neri García Rosas y Artemio Yáñez Pérez.  
Gracias por todo*



# Contenido

<b>Introducción General</b> .....	9
Organización del documento.....	10
<b>1. Marco Teórico</b> .....	13
1.1 Software .....	13
1.1.1 Características del Software .....	13
1.2 Ingeniería de Software .....	14
1.3 El proceso del software .....	16
1.4 Proceso de software ágil.....	17
1.4.1 Manifiesto por el Desarrollo Ágil de Software .....	17
1.4.2 Scrum .....	18
1.5 Ingeniería de requerimientos.....	22
1.5.1 Modelado de los requerimientos .....	24
1.5.2 Historias de usuario.....	25
1.6 Arquitectura.....	25
1.6.1 Diseño Arquitectónico.....	25
1.6.2 Patrones .....	26
1.6.3 Patrones arquitecturales.....	27
1.6.4 Patrón Arquitectural por Capas .....	27
1.6.5 Aplicaciones empresariales .....	27
1.6.6 Patrón arquitectónico por Capas en Aplicaciones Empresariales .....	27
1.7 Bases de datos .....	28
1.7.1 Modelo de datos .....	28
1.7.2 El Modelo Relacional.....	29
1.7.3 Normalización .....	29
1.8 Criptografía .....	30
1.8.1 Objetivos de la criptografía .....	31
1.8.2 Funciones Hash .....	31
1.8.3 Algoritmos de llave pública .....	31
<b>2. Análisis del SADEC</b> .....	34
2.1 Definición del Sistema .....	35
2.2 Recopilación de información .....	36

2.2.1 Cartera de Requisitos (Requirements Backlog) .....	36
2.2.2 Historias de Usuario .....	38
2.3 Especificación de Requisitos.....	40
2.3.1 Análisis de Historias de Usuario .....	40
2.3.2 Cartera del Producto (Product Backlog) .....	40
<b>3. Diseño del SADEC .....</b>	<b>44</b>
3.1 Arquitectura del Sistema .....	45
3.1.1 Patrón arquitectural por capas .....	46
3.1.2 Capa de Presentación del SADEC.....	49
3.1.3 Capa de Lógica de Negocio del SADEC.....	49
3.1.4 Subcapa de Acceso a Datos del SADEC.....	49
3.1.5 Subcapa de Fuente de Datos del SADEC.....	49
3.2 Diseño de la Base de Datos del SADEC .....	49
3.2.1 Identificación de los Elementos de Datos .....	50
3.2.2 Subdivisión de Elementos de Datos .....	52
3.2.3 Identificación de tablas y asignación de columnas.....	56
3.2.4 Identificación de Llaves Primarias y Foráneas.....	57
3.2.5 Diagrama Preliminar de la Estructura de Datos .....	60
3.2.6 Normalización de la Estructura de Datos .....	61
3.2.7 Diagrama de Clases .....	62
<b>4. Implantación del SADEC.....</b>	<b>65</b>
4.1 Equipo de Hardware y Herramientas de Software .....	66
4.1.1 Equipo de hardware.....	66
4.1.2 Herramientas de Software .....	66
4.2 Implantación de Base de Datos del Sistema.....	66
4.2.1 Tablas de la Base de Datos.....	66
4.3 Implantación de la Arquitectura del Sistema .....	68
4.3.1 Capa de Fuente de Datos .....	69
4.3.2 Capa de Lógica de Dominio.....	70
4.3.3 Capa de Presentación .....	72
4.4 Revisión y Retrospectiva de Iteración 0.....	74
4.5 Ejecución de Iteración 1 .....	74
4.5.1 Módulo de Generación de CFDI .....	74

4.5.2 Pruebas de Módulo de Generación de CFDI.....	76
4.5.3 Módulo de Sellado de CFDI.....	77
4.5.4 Pruebas del módulo de sellado .....	82
4.5.5 Módulo de Timbrado de CFDI.....	82
4.5.6 Pruebas de Timbrado de CFDI.....	86
4.5.7 Revisión y Retrospectiva de Iteración 1.....	87
4.6 Ejecución de Iteración 2.....	87
4.6.1 Implantación de Filtros para Visualización de CFDI.....	87
4.6.2 Pruebas de Visualización .....	89
4.6.3 Revisión y Retrospectiva de Iteración 2.....	93
4.7 Ejecución de Iteración 3.....	93
4.7.1 Autenticación de Usuarios Administrativos.....	93
4.7.2 Pruebas de Autenticación de Usuarios Administrativos .....	96
4.7.3 Privilegios de Usuarios Administradores .....	96
4.7.4 Pruebas de Implantación de Privilegios de Usuario.....	98
4.7.5 Revisión y Retrospectiva de Iteración 3.....	98
4.8 Ejecución de Iteración 4.....	98
4.8.1 Administración de Roles .....	99
4.8.2 Administración de Usuarios .....	100
4.8.3 Pruebas .....	101
4.8.4 Revisión y Retrospectiva de Iteración 4.....	101
4.9 Ejecución de Iteración 5.....	102
4.9.1 Autenticación de trabajadores .....	102
4.9.2 Consulta y Descarga de CFDI para Trabajadores .....	104
4.9.3 Pruebas del Portal de Consulta de CFDI.....	107
4.9.4 Revisión y Retrospectiva de Iteración 5.....	107
<b>5. Conclusiones y Perspectivas.....</b>	<b>110</b>
5.1 Conclusiones .....	110
5.2 Perspectivas.....	111
<b>Apéndice A .....</b>	<b>113</b>
<b>Bibliografía.....</b>	<b>115</b>





# Introducción General

En el año 2013 se publicaron reformas al Código Fiscal de la Federación, en donde se especifica que todo contribuyente con la facultad de emitir Comprobantes Fiscales Digitales (CFD) deberá sustituirlos por Comprobantes Fiscales Digitales por Internet (CFDI) a partir del primero de Abril de 2014 (previa prórroga). Tal es el caso de la Benemérita Universidad Autónoma de Puebla, la cual, debido a estas reformas, se encuentra obligada a proporcionar recibos bajo el esquema de Comprobante Fiscal Digital por Internet (CFDI) a todos los trabajadores que conforman la Nómina Institucional.

Un CFDI como recibo electrónico de nómina es un documento en formato XML que contiene información sobre el emisor y receptor del mismo, así como datos generales de nómina. Para la emisión de estos documentos es requerido especificar la siguiente información:

- Datos del emisor: Datos fiscales de quién emite el recibo de nómina (nombre, RFC, régimen fiscal, etc.)
- Datos del receptor: Datos fiscales de quién recibe el recibo de nómina (RFC, nombre, dirección, etc.)
- Nómina: Datos fiscales referentes a nómina (fecha de pago, percepciones, deducciones, etc.)

Todo CFDI como recibo electrónico de nómina debe contar con esta información, estructurada según estándares establecidos por el Sistema de Administración Tributaria (SAT), cuya validación se puede realizar gracias a un servicio proporcionado por este organismo.

Para que un CFDI cuente con validez fiscal ante el SAT, el comprobante debe ser sometido a un proceso de timbrado. Durante este proceso, el SAT valida la información contenida en dicho comprobante y se añade nueva información.

El proceso de emisión de CFDI requiere de conocimientos técnicos particulares. Para cubrir estos requerimientos, existen distintas alternativas, siendo la más socorrida el uso de Sistemas de Software genéricos o a la medida.

Desde la publicación de las reformas al Código Fiscal de la Federación, surgieron múltiples empresas y particulares que ofrecen Sistemas de Software de facturación electrónica, cada uno con distintas características y con el objetivo en común de cubrir los requerimientos básicos formulados en las reformas antes mencionadas. Además, el Sistema de Administración Tributaria proporciona una herramienta de Software sobre plataforma Web que cubre estos requerimientos.

La Benemérita Universidad Autónoma de Puebla es una Institución que pertenece al régimen fiscal de personas morales con fines no lucrativos, por lo cual se encuentra obligada a adoptar la ley establecida en las reformas al Código Fiscal de la Federación, y por lo tanto, debe proporcionar a sus trabajadores CFDI como recibos electrónicos de nómina.

Para cubrir esta necesidad se propone este proyecto de tesis cuyo **Objetivo General** es:

**Analizar, diseñar e implantar un sistema de Software (“Sistema de Administración Electrónica de CFDI” SADEC), para la generación, almacenamiento, gestión y consulta de Comprobantes Fiscales Digitales por Internet de la Nómina Institucional de la BUAP**

Los objetivos particulares que se plantean para alcanzar el objetivo antedicho son:

- Recopilación de información relacionada con Ingeniería de Software
- Recopilación de información del marco de trabajo Scrum
- Recopilación de información de patrones arquitecturales de software
- Recopilación de información de bases de datos
- Recopilación de información del proceso de emisión de Comprobantes Fiscales Digitales por Internet

En un futuro, los módulos principales del SADEC podrán ser utilizados para procesos de timbrado de los CFDI expedidos por las distintas dependencias que integran la Benemérita Universidad Autónoma de Puebla, sin limitarse a aquellos comprobantes de Nómina Institucional.

En lo personal, el desarrollo e implantación del SADEC me servirá para aplicar los conocimientos adquiridos en la carrera de Ingeniería en Ciencias de la Computación, así como de la experiencia laboral adquirida hasta el momento de su desarrollo, utilizando para esto el marco de trabajo Scrum, diseño de bases de datos relacionales, implantación de patrones arquitecturales, y el paradigma de programación orientado a objetos.

## **Organización del documento**

El presente documento está organizado en cinco capítulos.

El Capítulo 1 abarca la teoría que respaldará al proyecto. En este capítulo se presentan los conceptos principales de Ingeniería de Software, así como del marco de trabajo Scrum. Se exponen los conceptos de diseño arquitectónico de software y del patrón arquitectural por

capas orientado a aplicaciones empresariales, además, se proporcionan conceptos fundamentales de bases de datos relacionales y de criptografía.

En el Capítulo 2 se desarrolla el análisis del SADEC. Se establece el formato de historias de usuario para la especificación de requisitos junto con el backlog de requisitos. Además, de acuerdo al marco de trabajo Scrum, se define el backlog de producto a partir de un backlog de requisitos.

En el Capítulo 3 se presenta el diseño del SADEC; se establece su arquitectura utilizando el patrón arquitectural por capas enfocado a aplicaciones empresariales. En este capítulo se proporciona el diseño de la base de datos, donde se incluye el proceso de normalización, concluyendo con el diagrama Entidad-Relación obtenido.

El Capítulo 4 abarca la implantación del sistema, en donde se presenta el desarrollo de los módulos principales, así como de las interfaces gráficas de usuario, además, se mencionan las herramientas utilizadas durante el desarrollo.

En el Capítulo 5 se presentan las conclusiones y perspectivas, concluyendo con el desarrollo del SADEC y del documento presente.

Por último se incluye:

- El Apéndice A, donde se trata la instalación y puesta a punto del SADEC
- La bibliografía consultada



# Capítulo 1

## Marco Teórico

### 1.1 Software

Formalmente, el Software se puede definir como un conjunto de instrucciones (programas de cómputo) que al ejecutarse proporcionan las características, funciones y desempeño buscados; estructuras de datos que permiten que los programas manipulen la información en forma adecuada. Otra definición de Software es el conjunto de programas de cómputo y su documentación asociada [1].

#### 1.1.1 Características del Software

El software se desarrolla o modifica con intelecto. Al igual que en fabricación de hardware, la alta calidad se logra a través de un buen diseño. Ambas actividades requieren la construcción de un producto, pero con enfoques distintos.

El software no se desgasta. El software no es susceptible a los problemas ambientales que hacen que el hardware se desgaste (suciedad, vibraciones, abuso, temperatura, etc.), sin embargo, se presentarán tasas de fallas elevadas al comienzo de vida de un programa, las cuales se corrigen. El software sufrirá cambios durante su vida; es probable que cuando éstos se realicen se introduzcan errores. Cambios continuos traen como consecuencia que el software se deteriore.

El buen software debe entregar al usuario la funcionalidad y el desempeño requeridos, y debe ser sustentable, confiable y utilizable [2].

### 1.1.2 Dominios de aplicación

Existen distintas categorías de software. A continuación se presentan las más importantes.

**Software de sistemas:** conjunto de programas escritos para dar servicio a otros programas (compiladores, editores, herramientas de administración de archivos, etc.). Se caracteriza por gran interacción con el hardware de la computadora, uso intensivo por parte de usuarios múltiples, operación concurrente que requiere la secuenciación, recursos compartidos y administración de un proceso sofisticado, estructuras complejas de datos e interfaces externas múltiples.

**Software de aplicación:** programas aislados que resuelven una necesidad específica de negocios. Procesan datos comerciales o técnicos en una forma que facilita las operaciones de negocios o la toma de decisiones administrativas o técnicas.

**Software de ingeniería y ciencias:** caracterizado por algoritmos devoradores de números.

**Software incrustado:** reside dentro de un producto o sistema y se usa para implementar y controlar características y funciones para el usuario final y para el sistema en sí. Ejecuta funciones limitadas y particulares o provee una capacidad significativa de funcionamiento y control.

**Software de línea de productos:** diseñado para proporcionar una capacidad específica para uso de muchos consumidores diferentes. Se centra en un mercado limitado y particular o se dirige a mercados masivos de consumidores.

**Aplicaciones web:** (Ambientes sofisticados de cómputo) integradas con bases de datos corporativas y aplicaciones de negocios.

**Software de inteligencia artificial:** uso de algoritmos no numéricos para resolver problemas complejos que no son fáciles de tratar computacionalmente o con el análisis directo [1].

## 1.2 Ingeniería de Software

En los inicios de la computación, el desarrollo de software se limitaba a la búsqueda de soluciones específicas a dificultades generalmente del ámbito científico. Dichas soluciones generalmente eran abordadas por un solo individuo. Hoy en día, el software se ha incrustado en casi todo aspecto de la vida cotidiana y los requerimientos de la tecnología de la información que demandan individuos, negocios y gobiernos se hacen cada vez más complejos. Por lo anterior, se debe hacer una ingeniería (ingeniería como verbo según Scott Whitmire) con el software en todas sus formas y a través de todos sus dominios de aplicación.

Según Fritz Bauer la ingeniería de software es el establecimiento y uso de principios fundamentales de la ingeniería con el objeto de desarrollar en forma económica software que sea confiable y que trabaje con eficiencia en máquinas reales.

Según el IEEE La ingeniería de software es: 1) La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software; es decir, la aplicación de la ingeniería al software. 2) El estudio de enfoques según el punto 1.

La ingeniería de software es el proceso de diseñar, desarrollar y entregar sistemas de software que cumplan los requisitos de un cliente de manera eficiente y rentable. Para lograr este fin, los ingenieros de software no sólo se concentran en la escritura de un programa, si no en cada aspecto del proceso, desde la comunicación clara con el cliente, a través de la gestión eficaz de proyectos, hasta las prácticas de mantenimiento después de la entrega.

De lo anterior, la ingeniería de software es una tecnología con varias capas, que al igual que cualquier enfoque de ingeniería debe basarse en un compromiso organizacional con la calidad (FIGURA 1.2.1).



**Figura 1.2.1.** Capas de la ingeniería de software

El fundamento para la ingeniería de software es la capa proceso, pues, une las capas y permite el desarrollo racional y oportuno del software. Define una estructura que debe establecerse para la obtención eficaz de tecnología de ingeniería de software. El proceso de software forma la base para el control de la administración de proyectos de software, y establece el contexto en el que se aplican métodos técnicos, se generan productos del trabajo, se establecen puntos de referencia, se asegura la calidad y se administra el cambio de manera apropiada.

Los métodos de la ingeniería de software proporcionan la experiencia técnica para elaborar software. Incluyen un conjunto amplio de tareas, como comunicación, análisis de los requisitos, modelación del diseño, construcción del programa, pruebas y apoyo. Los métodos de la ingeniería de software se basan en un conjunto de principios fundamentales que gobiernan cada área de la tecnología e incluyen actividades de modelación y otras técnicas descriptivas.

Las herramientas de la ingeniería de software proporcionan un apoyo automatizado o semiautomatizado para el proceso y los métodos [1].

### 1.3 El proceso del software

Un proceso es un conjunto de actividades, acciones y tareas que se ejecutan cuando va a crearse algún producto del trabajo. Una actividad busca lograr un objetivo amplio y se desarrolla sin importar el dominio de la aplicación, tamaño del proyecto, complejidad del esfuerzo o grado de rigor con el que se usará la ingeniería de software. Una acción es un conjunto de tareas que producen un producto importante del trabajo. Una tarea se centra en un objetivo pequeño pero bien definido que produce un resultado tangible.

En el contexto de la ingeniería de software, un proceso es un enfoque adaptable que permite que las personas que hacen el trabajo busquen u elijan el conjunto apropiado de acciones y tareas para el trabajo. Se busca siempre entregar el software en forma oportuna y con calidad suficiente para satisfacer a quienes patrocinaron su creación y a aquellos que lo usarán.

La estructura del proceso establece el fundamento para el proceso completo de la ingeniería de software por medio de la identificación de un número pequeño de actividades estructurales que sean aplicables a todos los proyectos de software, sin importar su tamaño o complejidad. Una estructura de proceso general para la ingeniería de software consta de cinco actividades [1]:

- **Comunicación:** Se busca entender los objetivos de los participantes respecto del proyecto, y reunir los requisitos que ayuden a definir las características y funciones del software.
- **Planeación:** Define el trabajo de ingeniería de software al describir las tareas técnicas por realizar, los riesgos probables, los cursos que se requieren, los productos del trabajo que se obtendrán y una programación de las actividades.
- **Modelado:** Un modelo es creado con el fin de entender mejor los requisitos del software y el diseño que los satisfará.
- **Construcción:** Combina la generación de código (manual o automatizada) y las pruebas que se requieren para descubrir errores en éste.
- **Despliegue:** El software (como entidad completa o como un incremento parcialmente terminado) se entrega al consumidor que lo evalúa y que le da retroalimentación, misma que se basa en dicha evaluación.

El flujo del proceso describe la manera en que están organizadas las actividades estructurales y las acciones y tareas que ocurren dentro de cada una con respecto de la secuencia y del tiempo. Existen cuatro distintos flujos de proceso:

- **Lineal:** Ejecuta cada una de las cinco actividades estructurales en secuencia, comenzando por la comunicación y terminando con el despliegue.
- **Iterativo:** Repite una o más de las actividades antes de pasar a la siguiente.
- **Paralelo:** Ejecuta una o más actividades en paralelo con las otras.



## 1.4 Proceso de software ágil

Cualquier proceso del software ágil se caracteriza por la forma en la que aborda cierto número de suposiciones clave acerca de la mayoría de proyectos de software:

- Es difícil predecir qué requerimientos de software persistirán y cuáles cambiarán. También es difícil pronosticar cómo cambiarán las prioridades del cliente a medida que avanza el proyecto.
- Para muchos tipos de software, el diseño y la construcción están imbricados. Es decir, ambas actividades deben ejecutarse en forma simultánea, de modo que los modelos de diseño se prueben a medida que se crean. Es difícil predecir cuánto diseño se necesita antes de que se use la construcción para probar el diseño.
- El análisis, el diseño, la construcción y las pruebas no son tan predecibles como nos gustaría (desde un punto de vista de planeación).

Para que un proceso pueda manejar lo impredecible éste debe ser adaptable incrementalmente. Para lograr la adaptación incremental, un equipo ágil requiere retroalimentación con el cliente. Deben entregarse incrementos de software en periodos cortos de tiempo, de modo que la adaptación vaya a ritmo con el cambio [1].

### 1.4.1 Manifiesto por el Desarrollo Ágil de Software

En 2001 Kent Beck junto con otros 16 notables desarrolladores de software, escritores y consultores (grupo conocido como la Agile Alliance o Alianza Ágil) firmaron el “Manifiesto por el Desarrollo Ágil de Software”. En este manifiesto se destacan cuatro valores y 12 principios.

#### Valores

Los cuatro valores del Manifiesto por el Desarrollo Ágil de Software son:

- Individuos e interacciones sobre procesos y herramientas
- Software funcionando sobre documentación extensiva
- Colaboración con el cliente sobre negociación contractual
- Respuesta ante el cambio sobre seguir un plan

Cada valor consta de dos elementos y se declara que aunque se valoran los elementos de la derecha, se valoran más los de la izquierda.

#### Principios

La Alianza Ágil define 12 principios en su Manifiesto por el Desarrollo Ágil de Software, los cuales se citan a continuación [13]:

1. La mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor

2. Se acepta que los requisitos cambien, incluso en etapas tardías de desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
3. Se entrega software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
4. Los responsables de negocio y los desarrolladores trabajan juntos de forma cotidiana durante todo el proyecto.
5. Los proyectos se desarrollan en torno a individuos motivados. Se les debe proporcionar el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
6. El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
7. El software funcionando es la medida principal de progreso.
8. Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios deben ser capaces de mantener un ritmo constante de forma indefinida.
9. La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.
10. La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
12. A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

Debido a la naturaleza de los requisitos del presente proyecto, se optó por utilizar el modelo de desarrollo ágil Scrum, recurriendo al uso de historias de usuario para la recolección de dichos requisitos. En las secciones siguientes se detalla el marco de trabajo Scrum.

### **1.4.2 Scrum**

Scrum es una metodología ágil concebida por Jeff Sutherland y su equipo de desarrollo a principios de la década de 1990, la cual puede ser aplicada a casi cualquier proyecto, sin embargo, es más comúnmente utilizada en el desarrollo de software. Los proyectos a los cuales Scrum se adecúa de manera natural son aquellos que requieren cambios rápidos o que posiblemente tendrán requerimientos emergentes.

Según Ken Schwaber, Scrum no es una metodología, sino un marco de trabajo o framework.

A diferencia de los modelos de proceso prescriptivos, en Scrum no existe una serie de pasos secuenciales a seguir para conseguir un producto terminado en tiempo y forma. Por el contrario, Scrum puede ser visualizado como un marco de trabajo basado en un conjunto de valores, principios y prácticas. Cuando una organización adopta Scrum como marco de trabajo, ésta puede agregar sus propias prácticas, dando como resultado una versión de Scrum única a la organización. Todas las prácticas propias de Scrum se incorporan en roles, actividades y artefactos junto con sus reglas asociadas.

Los esfuerzos de desarrollo de Scrum se componen de uno o más equipos denominados Scrum Teams, cada uno formado por tres roles distintos: cliente (Product Owner), maestro (Scrum Master) y el equipo de desarrollo (Development Team). Dependiendo de la organización que se adopte en la implementación de Scrum, pueden existir más roles en cada equipo, sin embargo, el framework sólo requiere de los tres roles listados anteriormente [3].

#### 1.4.2.1 Roles de scrum

A continuación se describe cada uno de los roles que conforman los equipos de Scrum [4]:

- **Cliente (Product owner):** Responsable de lo que se desarrollará y el orden en que se hará. Es la única autoridad responsable de decidir qué características y funcionalidades se construirán, y el orden en que éstas serán desarrolladas. Además, se encarga de mantener y comunicar a todo participante una visión clara de lo que el equipo de Scrum debe desarrollar.  
Para asegurar que el equipo construye de manera rápida lo que el cliente requiere, éste último colabora activamente con el maestro (Scrum Master) y el equipo de desarrollo, respondiendo a cualquier duda planteada de manera rápida.
- **Maestro (Scrum master):** Ayuda a todo individuo involucrado a entender y aceptar los valores, principios y prácticas de Scrum. Actúa como coach al liderar el proyecto y ayudar al equipo de Scrum (y al resto de la organización) a desarrollar su propio enfoque.
- **Equipo de desarrollo (Development team):** Grupo de personas responsable de diseñar, construir y probar el producto deseado; se auto-organiza para determinar la mejor manera de cumplir con los objetivos establecidos.

#### 1.4.2.2 Actividades y artefactos de scrum

A continuación se presenta un resumen de las actividades y artefactos de Scrum y su interacción [3].

El cliente tiene una visión de lo que quiere crear. Debido a que el producto a crear puede ser de tamaño considerable, éste se divide en un conjunto de características, las cuales son recolectadas en una lista priorizada llamada cartera o pila del producto (product backlog), mediante una actividad denominada grooming.

Un Sprint comienza con una planeación (sprint planning), que abarca el trabajo de desarrollo durante el sprint (sprint execution), y termina con una revisión y retrospectiva. Es probable que el número de ítems en la cartera del producto sea más de lo que un equipo de desarrollo pueda completar en un sprint de corta duración, por esto, al principio de cada sprint, el equipo de desarrollo debe determinar un subconjunto de ítems de la cartera de producto que considere lograr completar. A esta actividad se le conoce como planeación del sprint (sprint planning).

Para adquirir confianza de que el equipo de desarrollo ha hecho un compromiso razonable, los miembros del equipo crean una segunda pila o cartera durante la planeación del sprint,

llamada cartera del sprint (sprint backlog). La cartera o pila del sprint describe mediante un conjunto de tareas detalladas cómo el equipo planea diseñar, construir, integrar, y probar el subconjunto seleccionado de características de la cartera del producto durante ese sprint en particular.

Lo siguiente es la ejecución del sprint (sprint execution), donde el equipo de desarrollo realiza las tareas necesarias para realizar las características seleccionadas. Cada día durante la ejecución del sprint, los miembros del equipo pueden ayudar a controlar el flujo de trabajo mediante la realización de una actividad de sincronización, inspección y planeación adaptativa conocida como scrum diario (daily scrum). Al final de la ejecución del sprint, el equipo habrá producido un incremento del producto potencialmente entregable.

El equipo de Scrum completa el sprint realizando dos actividades de inspección y adaptación. En la primera, llamada revisión del sprint (sprint review), los interesados (stakeholders) y el equipo inspeccionan el producto que se está construyendo. En la segunda, llamada retrospectiva (scrum retrospective), el equipo inspecciona el proceso de Scrum que está siendo usado para crear el producto. El resultado de estas dos actividades pueden ser adaptaciones que mejorarán su camino en el product backlog o serán incluidas como parte del proceso de desarrollo del equipo.

En este punto, se repite el ciclo del Scrum sprint, comenzando nuevamente con el equipo de desarrollo determinando el siguiente conjunto más importante de ítems que pueda completar. Después de que un número apropiado de sprints han sido realizados, la visión del cliente se habrá realizado y la solución puede ser liberada.

### **1.4.2.3 Actividades y artefactos de Scrum**

A continuación se presentan las actividades y artefactos de Scrum [3].

#### **Cartera de Producto (Product Backlog)**

En Scrum, primero se realiza el trabajo más valioso; el cliente, con entradas del resto del equipo de Scrum e interesados, es el responsable en última instancia de determinar y administrar la secuencia de este trabajo, y comunicarlo en forma de una lista priorizada u ordenada conocida como cartera del producto (product backlog). En el desarrollo de un producto nuevo, los ítems de la cartera de producto inicialmente son características requeridas para cumplir con la visión del cliente. Durante el desarrollo de un producto, la cartera de producto puede contener nuevas características, cambios a características existentes, defectos por reparar, mejoras técnicas, etc.

El cliente colabora con los interesados internos y externos para recolectar y definir los ítems de la cartera de producto. Entonces se asegura que los ítems de la cartera estén colocados en la secuencia correcta (usando factores tales como valor, costo, conocimiento y riesgos) de manera que los ítems de mayor valor aparezcan en la parte superior del product backlog, y los ítems de menor valor aparezcan hacia la parte inferior. La cartera o pila de producto es un artefacto de constante evolución. Se pueden agregar, eliminar y revisar ítems por el cliente

según como cambien las condiciones del negocio, o conforme el entendimiento del producto del equipo de Scrum crezca.

En general, la actividad de crear y refinar ítems de la cartera de producto, la estimación y priorización de los mismos es conocida como grooming.

Antes de finalizar de priorizar y ordenar el product backlog, se debe conocer el tamaño de cada ítem. El tamaño equivale a costo, y los clientes necesitan saber el costo de un ítem para determinar apropiadamente su prioridad. En Scrum no se dicta qué medida de tamaño usar con los ítems del product backlog. En la práctica se pueden utilizar medidas de tamaño relativas (tales como relative size measure o story points), las cuales expresan el tamaño total de un ítem de manera que el valor absoluto no es considerado, pero sí el tamaño relativo de un ítem comparado con otros.

### **Iteraciones (Sprints)**

En Scrum, el trabajo se realiza en iteraciones o ciclos de hasta un mes llamados sprints. El trabajo completado en cada iteración debe crear algo de valor tangible al usuario o cliente.

Las iteraciones son encasilladas en periodos de tiempo definidos, por lo que siempre tienen una fecha de inicio y fin, y generalmente todas deben ser de la misma duración. Una nueva iteración sigue inmediatamente a la completitud de la iteración anterior.

### **Planeación de Iteración (Sprint Planning)**

Una cartera de producto puede representar varias semanas o meses de trabajo, lo que puede ser más de lo que puede ser terminado en una sola iteración corta. Para determinar el subconjunto de ítems más importante de la cartera de producto a construirse en el próximo sprint, el cliente, equipo de desarrollo, y el scrum master realizan una planeación de iteración.

Durante esta planeación, el cliente y el equipo de desarrollo acuerdan un objetivo (sprint goal) que define lo que se debe lograr en la próxima iteración. Usando este objetivo, el equipo de desarrollo revisa la cartera del producto y determina los ítems de mayor prioridad que el equipo puede realmente cumplir en la próxima ejecución mientras se trabaja a un ritmo sostenible.

Para adquirir confianza de lo que se puede hacer, muchos equipos de desarrollo descomponen cada característica objetivo en un conjunto de tareas. La colección de estas tareas, junto con sus ítems de la cartera de producto asociados forman una segunda cartera llamada cartera del sprint (sprint backlog). Entonces el equipo de desarrollo provee un estimado (normalmente en horas) del esfuerzo requerido para completar cada tarea.

### **Ejecución de la Iteración (Sprint Execution)**

Una vez que el equipo de Scrum termina la planeación de la iteración y acuerda el contenido de la iteración, el equipo de desarrollo, guiado por el scrum master, realiza todo el trabajo necesario para terminar las características.

### **Scrum Diario (Daily Scrum)**

Cada día durante cada iteración, idealmente al mismo tiempo, los miembros del equipo de desarrollo mantienen un scrum diario con una duración de tiempo preestablecida.

Un enfoque común para realizar el scrum diario es que el scrum master y el equipo de desarrollo tomen turnos para responder tres preguntas para el beneficio de los otros miembros del equipo:

¿Qué se ha logrado desde el último scrum diario?

¿En qué planeo trabajar en el próximo scrum diario?

¿Cuáles son los obstáculos o impedimentos que evitan el progreso?

Al responder estas preguntas, todos comprenden lo que está ocurriendo, cómo se está progresando hacia el objetivo de la iteración, cualquier modificación que se requiera realizar a los planes para los próximos días de trabajo, y qué problemas necesitan ser abordados.

### **Término**

En Scrum, se refiere a los resultados de una iteración como incremento del producto potencialmente entregable, lo que significa que lo que el equipo de scrum acordó en hacer está realmente terminado de acuerdo a su propia definición.

Potencialmente entregable no significa que lo construido debe ser entregado. La entrega es una decisión de negocios que se puede basar en la cantidad de funcionalidades terminadas y si éstas son lo suficiente como para justificar el despliegue al cliente [3].

## **1.5 Ingeniería de requerimientos**

Dentro de la perspectiva del proceso del software, la ingeniería de requerimientos es una de las acciones más importantes de la ingeniería de software que comienza durante la actividad de comunicación y continúa en la de modelado. Debe adaptarse a las necesidades del proceso, del proyecto, del producto y de las personas que hacen el trabajo. Tiende un puente para el diseño y la construcción.

La ingeniería de requerimientos proporciona el mecanismo apropiado para entender lo que desea el cliente, analizar las necesidades, evaluar la factibilidad, negociar una solución razonable, especificar la solución sin ambigüedades, validar la especificación y administrar los requerimientos a medida de que se transforman en un sistema funcional. Incluye siete tareas diferentes: concepción, indagación, elaboración, negociación, especificación, validación y administración. Algunas de estas tareas ocurren en paralelo y todas se adaptan a las necesidades del proyecto [1].

## Concepción

En la concepción del proyecto se establece el entendimiento básico del problema, las personas que quieren una solución, la naturaleza de la solución que se desea, así como la eficacia de la comunicación y colaboración preliminares entre los otros participantes y el equipo de software.

## Indagación

Se pregunta al cliente y a los usuarios cuáles son los objetivos para el sistema o producto, qué es lo que va a lograrse, cómo se ajusta el sistema o producto a las necesidades del negocio y, finalmente, cómo va a usarse el sistema o producto en las operaciones cotidianas.

## Elaboración

La información obtenida del cliente durante la concepción e indagación se expande y refina durante la elaboración. Esta tarea se centra en desarrollar un modelo refinado de los requerimientos que identifique distintos aspectos de la función del software, su comportamiento e información.

## Negociación

Se pide a clientes, usuarios y otros participantes que ordenen sus requerimientos según su prioridad y que después analicen los conflictos. Con el empleo de un enfoque iterativo que da prioridad a los requerimientos, se evalúa su costo y riesgo, y se enfrentan los conflictos internos; algunos requerimientos se eliminan, se combinan o se modifican de modo que cada parte logre cierto grado de satisfacción.

## Especificación

Puede ser un documento escrito, un conjunto de modelos gráficos, un modelo matemático formal, un conjunto de escenarios de uso, un prototipo o cualquier combinación de estos.

## Validación

La calidad de los productos del trabajo que se generan como consecuencia de la ingeniería de los requerimientos se evalúa durante el paso de validación. La validación de los requerimientos analiza la especificación a fin de garantizar que todos ellos han sido enunciados sin ambigüedades; que se detectaron y corrigieron las inconsistencias, las omisiones y los errores, y que los productos del trabajo se presentan conforme a los estándares establecidos para el proceso, el proyecto y el producto.

## Administración de los requerimientos

Es el conjunto de actividades que ayudan al equipo del proyecto a identificar, controlar y dar seguimiento a los requerimientos y a sus cambios en cualquier momento del desarrollo del proyecto. La administración formal de los requerimientos sólo se practica para proyectos grandes que tienen cientos de requerimientos identificables. Para proyectos pequeños, esta actividad tiene considerablemente menos formalidad. [1]

### 1.5.1 Modelado de los requerimientos

El modelado de requerimientos es la primera representación técnica de un sistema. Utiliza una combinación de texto y diagramas para ilustrar los requerimientos en forma que sea relativamente fácil de entender y, más importante, de revisar para corregir, completar y hacer congruente.

#### Definición de los requerimientos

Se puede definir a los requerimientos como “una especificación de lo que debe ser implementado”. Básicamente existen dos tipos de requerimientos:

- Requerimientos funcionales: Lo que el comportamiento del sistema debe ofrecer
- Requerimientos no funcionales: Propiedades o restricciones específicas en el sistema

Los requerimientos son la base de todo sistema. Esencialmente son sentencias de lo que el sistema debe hacer. En principio, los requerimientos deben ser sentencias sólo de lo que el sistema debe hacer, y no de cómo lo debe hacer. [4]

#### Análisis de los requerimientos

El análisis de los requerimientos da como resultado la especificación de las características operativas del software, indica la interfaz de éste y otros elementos del sistema, y establece las restricciones que limitan al software. [1]

La acción de modelar los requerimientos da como resultado uno o más de los siguientes tipos de modelo:

- **Modelos basados en el escenario** de los requerimientos desde el punto de vista de los distintos “actores” del sistema.
- **Modelos de datos**, que ilustran el dominio de información del problema
- **Modelos orientados a clases**, que representan clases orientadas a objetos (atributos y operaciones) y la manera en que las clases colaboran para cumplir con los requerimientos del sistema.
- **Modelos orientados al flujo**, que representan los elementos funcionales del sistema y la manera como transforman los datos a medida que se avanza a través del sistema.
- **Modelos de comportamiento**, que ilustran el modo en el que se comparte el software como consecuencia de “eventos” externos.

Estos modelos dan al diseñador del software la información que se traduce en diseño de arquitectura, interfaz y componentes. Por último, el modelo de requerimientos brinda al desarrollador y al cliente los medios para evaluar la calidad una vez construido el software.

En resumen, el modelo de requerimientos debe lograr tres objetivos principales:

- Describir lo que requiere el cliente



- Establecer una base para la creación de un diseño de software
- Definir un conjunto de requerimientos que puedan validarse una vez construido el software

### 1.5.2 Historias de usuario

Para el desarrollo del presente proyecto se optó por la utilización del marco de trabajo Scrum como base para la metodología de desarrollo. Al tratarse de una metodología ágil, la recolección de requisitos se llevó a cabo mediante el uso de historias de usuario.

Las historias de usuario son utilizadas en lugar de extensos documentos de requisitos. Son escritas por los clientes y se describe lo que el cliente desea que el sistema realice. Son similares al uso de escenarios, con la excepción de no estar limitadas a describir únicamente interfaces de usuario. Son escritas en un formato de pocas líneas de texto por los clientes en su propia terminología excluyendo sintaxis técnica.

Las historias de usuario también dirigen la creación de pruebas (tests) de aceptación. Una o más pruebas de aceptación deben ser creadas para verificar que la historia de usuario ha sido implementada correctamente.

Una de las principales diferencias entre las historias de usuario y las especificaciones tradicionales de requerimientos es el nivel de detalle. Una historia de usuario sólo debe proveer el detalle suficiente para realizar un estimado de bajo riesgo de cuánto tiempo tomará implementar la misma. Cuando llegue la hora de implementar la historia de usuario, los desarrolladores deberán acudir al cliente y recibir una descripción detallada de los requerimientos [3].

## 1.6 Arquitectura

La arquitectura es una representación del software operativo que permite analizar la efectividad del diseño para cumplir los requerimientos establecidos, considerar alternativas arquitectónicas en una etapa en la que hacer cambios al diseño todavía es relativamente fácil y, reducir los riesgos asociados con la construcción del software [1].

### 1.6.1 Diseño Arquitectónico

El diseño arquitectónico se interesa por entender cómo debe organizarse un sistema y cómo tiene que diseñarse la estructura global de ese sistema. Durante este proceso se identifican los principales componentes estructurales en un sistema, así como la relación entre ellos, y la salida consiste en un modelo arquitectónico que describe la forma en que se organiza el sistema como un conjunto de componentes en comunicación. [2]

En los procesos ágiles se acepta que una de las primeras etapas en el proceso de desarrollo debe preocuparse por establecer una arquitectura global del sistema.

La arquitectura de software es importante porque afecta el desempeño y la potencia, así como la capacidad de distribución y mantenimiento de un sistema.

El modelo de la arquitectura define la relación entre los elementos principales de la estructura del software, los estilos y patrones de diseño de la arquitectura que pueden usarse para alcanzar los requerimientos definidos por el sistema y las restricciones que afectan la forma en la que se implanta la arquitectura [1].

### 1.6.2 Patrones

Un patrón aborda un problema recurrente que surge en situaciones específicas de diseño y presenta una solución al mismo [6].

Algunas de las características principales de los patrones arquitecturales son:

- Identifican y especifican abstracciones por encima del nivel de componentes: Regularmente, un patrón describe distintos componentes y detalla sus responsabilidades y relaciones, así como su cooperación. Todos los componentes juntos solventan el problema que el patrón aborda, usualmente de una manera más efectiva que un solo componente.
- Son un medio para documentar la arquitectura de un software.
- Apoyan la construcción de software con propiedades definidas: Proveen un esqueleto de un comportamiento funcional y ayudan a implementar funcionalidad en una aplicación. Explícitamente abordan requerimientos no funcionales de sistemas de software, tales como mutabilidad, fiabilidad, capacidad de prueba o reutilización.
- Ayudan a construir arquitecturas de software complejas y heterogéneas: Todo patrón provee un conjunto predeterminado de componentes, roles y relaciones entre ellos. Puede ser usado para especificar aspectos particulares de estructuras de software específicas. Los patrones actúan como bloques de construcción para construir diseños más complejos.
- Ayudan a manejar la complejidad de un software: Todo patrón describe una forma comprobada para manejar el problema que aborda (los tipos de componentes necesarios, sus roles, los detalles que deben ser ocultados, las abstracciones que deben ser visibles, y cómo todo trabaja).

Aunque un patrón determina la estructura básica de una solución a un problema de diseño en particular, éste no especifica una solución completamente detallada. Un patrón provee un esquema de una solución genérica a una familia de problemas. Este esquema debe ser implementado de acuerdo a las necesidades específicas del problema de diseño. Los patrones arquitecturales ayudan a solventar problemas, pero no proveen soluciones completas.

De lo anterior, un patrón arquitectónico de software describe un problema de diseño recurrente que surge en contextos específicos de diseño, y presenta un esquema genérico bien privado para su solución. El esquema es especificado describiendo los componentes que lo constituyen, sus responsabilidades y relaciones, y las formas en que colaboran [6].

Todo patrón adopta un esquema de tres partes:

- Contexto: Situación dada que da lugar a un problema
- Problema: El problema recurrente que surge en el contexto
- Solución: Solución probada del problema

### 1.6.3 Patrones arquitecturales

Un patrón arquitectural expresa un esquema de organización estructural fundamental para sistemas de software. Provee un conjunto de subsistemas predefinidos, especifica sus responsabilidades, e incluye reglas y directrices para organizar las relaciones entre ellos.

Los patrones arquitecturales ayudan a especificar la estructura fundamental de una aplicación. Toda actividad de desarrollo es gobernada por esta estructura [6].

### 1.6.4 Patrón Arquitectural por Capas

El patrón arquitectural por capas ayuda a estructurar aplicaciones que pueden ser descompuestas en grupos de sub tareas en los cuales, cada grupo de sub tareas es un nivel particular de abstracción.

### 1.6.5 Aplicaciones empresariales

Las aplicaciones empresariales son aplicaciones de negocio complejas, escalables, distribuidas, basadas en componentes y de misión crítica. Pueden ser desarrolladas en una variedad de plataformas a través de redes corporativas, intranets o la Internet. Son centradas en los datos, amigables con el usuario y deben cumplir con rigurosos requerimientos de seguridad, administración y mantenimiento.

En una aplicación empresarial se deben cumplir un gran número de requerimientos separados. Cualquier decisión en el desarrollo para satisfacer cada requerimiento afecta otros requerimientos de forma que son difíciles de entender o predecir [5].

### 1.6.6 Patrón arquitectónico por Capas en Aplicaciones Empresariales

El patrón arquitectónico por capas es ampliamente usado para el desarrollo de aplicaciones empresariales. Se definen tres capas principales:

- **Capa de presentación (Presentation layer):** Encargada de cómo manejar la interacción entre el software y el usuario. Las responsabilidades principales de esta capa son desplegar información al usuario e interpretar comandos del usuario en acciones sobre las capas de dominio y de datos.
- **Capa de fuente de datos (Data source layer):** Se encarga de la comunicación con otros sistemas que llevan a cabo tareas en nombre de la aplicación. Para la mayoría de aplicaciones empresariales la pieza central de esta capa es una base de datos responsable de almacenar información persistente.

- **Capa de lógica de dominio (Domain logic layer):** También conocida como capa de lógica de negocio (business logic). Se encarga del trabajo que la aplicación debe realizar para el dominio sobre el que se esté trabajando. Implica cálculos basados en entradas e información almacenada, validación de información provenientes de la capa de presentación, y determinar exactamente qué capa de datos despachar, dependiendo de comandos recibidos de la capa de presentación.

## 1.7 Bases de datos

Una base de datos es una colección de datos que existe durante un largo periodo de tiempo. En otras palabras, el término base de datos se refiere a una colección de datos administrados por un Sistema Administrador de Bases de Datos o DBMS (Data Management System), del cual se espera lo siguiente [7]:

- Permitir a los usuarios crear nuevas bases de datos y especificar sus esquemas (estructura lógica de datos), utilizando un lenguaje de definición de datos especializado
- Proporcionar a los usuarios la habilidad de consultar la información y modificarla usando un lenguaje apropiado, frecuentemente llamado lenguaje de consulta o lenguaje de manipulación de datos.
- Soportar el almacenamiento de una gran cantidad de datos durante un periodo largo de tiempo, permitiendo el acceso eficiente a la información para consultas y modificaciones a la base de datos.
- Permitir durabilidad, la recuperación de la base de datos de cara a fallas, errores de distintos tipos, o mal uso intencionado.
- Control de acceso a la información de múltiples usuarios al mismo tiempo, sin permitir interacciones inesperadas entre usuarios (aislamiento) y sin realizar acciones en los datos parcialmente pero no completamente (atomicidad).

### 1.7.1 Modelo de datos

Un modelo de datos es una notación para describir datos o información. La descripción consiste de tres partes:

- Estructura de los datos
- Operaciones en los datos: A diferencia de los lenguajes de programación, en los modelos de datos de bases de datos usualmente existe un conjunto limitado de operaciones que pueden realizarse. Generalmente se permite realizar un conjunto limitado de consultas y modificaciones.
- Restricciones en los datos: Regularmente los modelos de datos tienen una manera de describir limitaciones de lo que pueden ser los datos.

Existen distintos modelos de datos, en el presente proyecto se hizo uso del modelo relacional para describir la información.

### 1.7.2 El Modelo Relacional

El modelo relacional nos proporciona una manera simple para representar información como una tabla bidimensional llamada relación. A continuación se presenta la terminología más importante relacionada con este modelo:

#### Columnas

Las columnas de una relación son llamadas atributos. Usualmente un atributo describe el significado de entradas en la columna.

#### Esquema

El nombre de una relación y el conjunto de atributos de una relación es llamado esquema de la relación. En el modelo relacional, una base de datos consiste de una o más relaciones. Al conjunto de esquemas de relaciones de una base de datos se le conoce como esquema de base de datos relacional o esquema de base de datos.

#### Tuplas

Las filas de una relación son llamadas tuplas. Una tupla tiene un componente por cada atributo de la relación.

#### Dominios

El modelo relacional requiere que cada componente de cada tupla sea atómico; es decir, debe ser de algún tipo elemental como entero o cadena de caracteres. Asociado con cada atributo de una relación hay un dominio, que es un tipo elemental particular.

#### Llaves de relaciones

Existen varias restricciones en relaciones que el modelo relacional permite colocar en esquemas de bases de datos. Las restricciones de llave son fundamentales en este modelo. La llave de una relación se formada por un conjunto de atributos si no se permite que dos tuplas en una relación tengan los mismos valores en todos los atributos de la llave.

### 1.7.3 Normalización

La teoría de la Normalización es un método objetivo y riguroso que se aplica en el diseño de bases de datos relacionales.

Un esquema relacional se puede obtener de dos formas distintas:

- Directamente a partir de la observación de nuestro universo del discurso, en donde se especifican conjuntos de atributos, relaciones y restricciones que corresponden a los observados en el mundo real.
- Realizando el proceso de diseño en dos fases, primero el diseño conceptual (E/R) obteniendo el esquema conceptual y posteriormente transformar éste a un esquema relacional, siguiendo algunas reglas generales, que fueron dadas anteriormente.

Algunos problemas que se pueden presentar son:

- Incapacidad para almacenar ciertos hechos
- Redundancias
- Ambigüedades
- Pérdida de información
- Pérdida de dependencias funcionales (restricciones de integridad que dan lugar a interdependencias entre los datos)
- Aparición en la base de datos de estados no válidos (anomalías de inserción, borrado y modificación)

En conclusión el esquema relacional obtenido debe ser analizado para comprobar que no presenta los problemas anteriores.

La normalización introduce una técnica formal para diseñar bases de datos relacionales, y permite mecanizar parte del proceso al disponer de algoritmos de normalización.

La normalización tiene como objetivo obtener esquemas relacionales que cumplan determinadas condiciones, a través de las formas normales:

- **Primera Forma Normal (1FN):** Es una restricción inherente al modelo relacional por lo que su cumplimiento es obligatorio. Consiste en la prohibición de que en una relación existan grupos repetitivos, es decir, un atributo no puede tomar más de un valor del dominio subyacente.
- **Segunda Forma Normal (2FN):** Una relación está en 2FN, si además de estar en 1FN, todos los atributos que no forman parte de ninguna clave candidata suministran información acerca de la clave completa. Cabe señalar que una relación que está formada por un único elemento está en 2FN.
- **Tercera Forma Normal (3FN):** Una relación está en 3FN, si además de estar en 2FN, los atributos que no forman parte de ninguna clave candidata facilitan información sólo acerca de la(s) clave(s) y no acerca de los atributos.
- **Forma Normal de Boyce y Codd (FNBC):** Una relación está en FNBC si y solo si, el conocimiento de las claves candidatas permite averiguar todas las interrelaciones existentes entre los datos de la relación (las claves candidatas son los únicos descriptores sobre los que se facilita información por cualquier otro atributo)

## 1.8 Criptografía

La criptografía es un método automatizado (o algorítmico) en el cual se llevan a cabo objetivos de seguridad. Estos algoritmos operan en mensajes en forma de grupos de bits.

### 1.8.1 Objetivos de la criptografía

A continuación se describen los principales objetivos de la criptografía [8].

#### **Privacidad**

La privacidad es la propiedad de ocultar el significado o la intención de un mensaje.

#### **Integridad**

La integridad es la propiedad de asegurar que un mensaje sea entregado de un punto “A” a un punto “B” sin cambios en el contenido o significado del mensaje en el proceso.

Usualmente la integridad se logra utilizando funciones criptográficas hash de una sola vía. Estas funciones aceptan como entrada un mensaje de longitud arbitraria y producen una digestión del mensaje (digest) de longitud fija.

#### **Autenticación**

La autenticación es la propiedad de atribuir una identidad o representante de la integridad de un mensaje. Un algoritmo de autenticación típicamente es simétrico de manera que todas las partes puedan producir información verificable.

#### **No repudio**

Es la propiedad de acordar apegarse a una obligación. En otras palabras es la imposibilidad de refutar una responsabilidad.

### 1.8.2 Funciones Hash

Una función hash formalmente conocida como función pseudo aleatoria o PRF por sus siglas en inglés (pseudo random function) mapea una entrada de tamaño arbitrario a una salida de tamaño fijo mediante un proceso conocido como compresión. Esta forma de compresión no se trata de una compresión de datos convencional, sino un mapeo no invertible. En el ámbito de la criptografía, las funciones hash deben tener dos propiedades para ser útiles: deben ser de una sola vía (dada la entrada de una función hash, no es trivial conocer cualquier información sobre dicha entrada) y resistentes a colisiones (dada la salida de una función hash, no es trivial encontrar una entrada que produzca la misma salida) [8].

#### **Función Hash SHA-1**

Las siglas SHA significan Secure Hash Algorithm. Existen cuatro distintos

Esta función produce un valor hash de 160 bits, el cual típicamente es renderizado como un número hexadecimal de 40 dígitos de longitud [8].

### 1.8.3 Algoritmos de llave pública

Estos algoritmos también son conocidos como algoritmos de llave asimétrica, y son utilizados para solventar dos problemas que los algoritmos de llave simétrica no pueden:

distribución de llaves y no repudio. El primero ayuda a resolver problemas de privacidad, y el segundo ayuda a solventar problemas de autenticidad. Los algoritmos de llave pública logran estos objetivos operando asimétricamente; una llave es dividida en dos partes correspondientes, una llave pública y una privada. La llave pública es llamada así debido a que es seguro darla a conocer públicamente a quien la solicite. Esta llave permite encriptar mensajes y verificar firmas. La llave privada es llamada de esta manera debido a que debe permanecer privada y no se puede proporcionar. Típicamente esta llave normalmente es propiedad de una sola persona o dispositivo. Esta llave permite desencriptar mensajes y generar firmas [8].

### **1.8.3.1 RSA Public Key Cryptography**

Es un sistema criptográfico de llave pública. Es el algoritmo más utilizado tanto para cifrar como para firmar digitalmente [8]. El SADEC hará uso de este algoritmo en el penúltimo paso del proceso de timbrado de CFDI, encriptando la digestión resultante de la aplicación del algoritmo de digestión SHA-1 a la cadena original del comprobante.





# Capítulo 2

## Análisis del SADEC

### Introducción

Debido a las reformas al Código Fiscal de la Federación de 2013, existe la necesidad de emitir, consultar y administrar CFDI como recibos de nómina a todo trabajador perteneciente a la Nómina Institucional de la Benemérita Universidad Autónoma de Puebla.

En el presente capítulo se presenta el desarrollo de la fase de análisis del proyecto con la finalidad de ofrecer al lector una visión general del sistema a desarrollar, así como plantear una especificación de requisitos según los criterios establecidos por la metodología de desarrollo ágil Scrum.

Los objetivos del presente capítulo son:

- Presentar una definición clara del sistema a desarrollar
- Obtener los requisitos generales (backlog de requisitos) del SADEC apegándose a la metodología de desarrollo investigada. A partir de tal backlog se podrá obtener un conjunto de historias de usuario, las cuales serán utilizadas para generar posteriormente un backlog de producto que servirá como base para los primeros Sprints.

## 2.1 Definición del Sistema

La Nómina Institucional de la Benemérita Universidad Autónoma de Puebla está conformada por alrededor de 12,000 trabajadores. Debido a que la Institución pertenece al régimen fiscal de personas morales con fines no lucrativos, se encuentra obligada a adoptar las leyes establecidas en las reformas al Código Fiscal de la Federación de 2013, y por lo tanto, a proporcionar a sus trabajadores Comprobantes Fiscales Digitales por Internet (CFDI) como recibos de nómina electrónicos. Para cubrir esta necesidad, se plantea el desarrollo de un Sistema de Software (SADEC) sobre plataforma Web que cubra tanto los requerimientos legales como los propios de la institución.

Un CFDI como recibo electrónico de nómina es un documento en formato electrónico XML (Extensible Markup Language) que contiene información del comprobante y datos de nómina, así como información del emisor y receptor del mismo. En el documento “ANEXO 20 de la Resolución Miscelánea Fiscal para 2014” [12], publicado el 30 de Diciembre de 2013 por el Servicio de Administración Tributaria (SAT) se reglamenta la estructura que deben respetar estos comprobantes, así como la información requerida y opcional que los mismos deben y pueden contener.

Se considera que un CFDI cuenta con validez fiscal ante el SAT sólo si el comprobante es sometido de manera exitosa a un proceso de timbrado a través de un Proveedor Autorizado de Certificación (PAC). Para someter un CFDI a éste proceso, se deben seguir una serie de pasos, que, a grandes rasgos constan de lo siguiente:

- Generación del comprobante en formato electrónico XML según la estructura definida en el ANEXO 20 de la Resolución Miscelánea Fiscal para 2014 publicado por el SAT.
- Someter el comprobante generado a un proceso de sellado siguiendo los lineamientos establecidos por el SAT.
- Una vez que el comprobante se encuentre sellado, podrá someterse a un proceso de timbrado.

El SADEC llevará a cabo los procesos necesarios para obtener CFDI con validez fiscal ante el SAT (generación, sellado y timbrado). Estos procesos deberán realizarse en periodos quincenales y de manera masiva o individual para todo trabajador que forme la Nómina Institucional de la BUAP.

El control de los distintos procesos realizados por el sistema será delegado a uno o más usuarios, por lo que se contará con cuatro niveles de usuarios administradores, cada uno con un rol distinto, existiendo para cada rol un conjunto de privilegios propios. A continuación se describen los roles de usuarios que serán implantados en el SADEC:

- Usuario Administrador encargado de la generación de documentos: El usuario tendrá privilegios únicamente para generar y eliminar comprobantes de manera individual o masiva, con la restricción de que dichos comprobantes no hayan sido sometidos al proceso de sellado ni timbrado.

- Usuario Administrador encargado del sellado de documentos: El usuario tendrá privilegios para sellar de manera individual o masiva documentos previamente generados y eliminarlos en caso de ser necesarios, con la restricción de que dichos documentos no hayan sido sometidos al proceso de timbrado.
- Usuario Administrador encargado del timbrado de documentos: El usuario contará con privilegios para timbrar individual o masivamente documentos previamente sellados. En este caso el usuario únicamente podrá eliminar aquellos documentos para los cuales haya surgido algún error en el proceso de timbrado, ya que un documento correctamente timbrado no podrá ser eliminado, si no cancelado ante el SAT mediante el servicio ofrecido por un PAC para obtener de esta manera un acuse de cancelación del documento previamente timbrado.

En las reformas al Código Fiscal de la Federación antes mencionadas, se establece que toda persona moral con facultad de emitir CFDI se encuentra obligada a proporcionar los comprobantes emitidos de manera electrónica a los usuarios receptores. Dado que la BUAP pertenece al régimen fiscal de “Personas Morales con Fines no Lucrativos”, el SADEC contará con un módulo de consulta vía Web desde el cual todo trabajador que conforme la Nómina Institucional de la BUAP podrá consultar los CFDI que le hayan sido emitidos.

## 2.2 Recopilación de información

Durante la fase de análisis en el desarrollo de un sistema de software, es de crítica importancia recopilar la mayor cantidad de información posible, pues servirá como base para la especificación de requisitos del sistema. En esta sección se lleva a cabo la recopilación de información según los criterios de Scrum para la especificación de requisitos.

### 2.2.1 Cartera de Requisitos (Requirements Backlog)

El desarrollo del SADEC se llevará a cabo haciendo uso de la metodología de desarrollo ágil o marco de trabajo Scrum. Como se describió en el Capítulo 1 (sección 1.4.2) de este trabajo, el marco de trabajo Scrum puede ser adoptado por cualquier organización agregando prácticas propias de ésta. Debido a la naturaleza de la solución de software a implantar, y de acuerdo a los criterios propuestos por Scrum, se optó por implantar un ciclo de especificación de requisitos, ejecutando tantas iteraciones como sean necesarias para obtener como resultado los requisitos generales del sistema (o backlog de requisitos de acuerdo a la terminología de Scrum).

A través de reuniones realizadas con los responsables de áreas de las dependencias institucionales de la BUAP, involucradas en el desarrollo del SADEC (Tesorería General, Recursos Humanos, Sistema de Información Universitaria), y posterior al análisis de los requerimientos tanto legales como institucionales en el problema a solventar, así como de las limitantes técnicas, se obtuvo como versión definitiva de la cartera de requisitos el listado mostrado en la Figura 2.2.1.2.

Cabe señalar que dentro de la metodología de desarrollo ágil Scrum se definen tres roles de usuarios, cada uno involucrado durante todo el ciclo de desarrollo y con distintas responsabilidades. En la figura 2.2.1.1 se muestran los roles antes mencionados acordados por el área de Soluciones de Software del Sistema de Información Universitaria de la Benemérita Universidad Autónoma de Puebla.

Rol	Descripción	Responsable
Cliente (Product Owner)	Responsable de las acciones a realizar durante el desarrollo del producto a construir, así como el orden en que éstas serán ejecutadas	Coordinador de Soluciones de Software
Scrum Master	Líder de proyecto y apoyo en implantación de Scrum	Subdirector de proyectos
Desarrollador	Encargado de diseñar, construir y probar el producto a construir	Consultor de Soluciones de Software

**Figura 2.2.1.1.** Roles de Scrum definidos para el SADEC

# Ítem	Requisito	Descripción
1	Obtención de información de nómina	Una vez que Recursos Humanos valida la información de la nómina Institucional, ésta deberá poder ser accedida por el SADEC para su posterior procesamiento.
2	Autenticación de usuarios	Debido a que existirán distintos niveles de usuario, cada uno con privilegios distintos, el sistema deberá manejar la autenticación de usuarios.
3	Generación/Eliminación de CFDI	En una primera etapa del proceso de timbrado de CFDI, el SADEC deberá generar comprobantes de nómina con la información obtenida de la Nómina Institucional.
4	Sellado/Eliminación de CFDI	Como segunda etapa en el proceso de timbrado de CFDI, el sistema deberá sellar los comprobantes generados según los estándares del SAT.
5	Timbrado/Cancelación de CFDI	En la etapa final del proceso de timbrado de CFDI, el sistema deberá timbrar los documentos sellados previamente mediante el servicio de un PAC.
6	Consulta de CFDI	Todo CFDI correctamente timbrado deberá poder ser consultado por el receptor del mismo vía Web.

**Figura 2.2.1.2.** Backlog de requisitos del SADEC

A partir del backlog de requisitos es posible obtener un conjunto de historias de usuario que serán sometidas a un proceso de análisis para ser utilizadas posteriormente en la generación de la primera versión de la cartera del producto (product backlog), en donde se especificarán los requisitos específicos del sistema y la iteración en la cual serán implantados.

## 2.2.2 Historias de Usuario

A partir del backlog de requisitos es posible obtener historias de usuario proporcionadas por el cliente. En el contexto del desarrollo ágil de software, las historias de usuario son una manera simple de describir una tarea concisa, generalmente utilizando un lenguaje no técnico [3].

Las historias de usuario para el SADEC se administraron de manera electrónica, llevando un formato similar al presentado en la Figura 2.2.2.1.

Como un: <u>Trabajador</u>	# 0001
Me gustaría: <u>Acceder al sistema como trabajador de la BUAP</u>	
Y así: <u>Poder consultar y descargar los CFDI que se me hayan emitido</u>	
Prueba:	
<ul style="list-style-type: none"> <li>- <u>Acceder con usuario y contraseña proporcionados</u></li> <li>- <u>Verificar que se presentan únicamente comprobantes emitidos al dueño de la cuenta</u></li> <li>- <u>Verificar la correcta descarga de comprobantes</u></li> </ul>	

**Figura 2.2.2.1.** Formato de historia de usuario del SADEC

En el formato presentado en la Figura 2.2.2.1 se pueden identificar los elementos que componen una historia de usuario utilizada durante la fase de recopilación de información. Como encabezado, toda historia de usuario deberá contar con un identificador único situado en la parte superior derecha del formato, así como el rol de la persona encargada de realizar la historia de usuario (parte superior izquierda). El cuerpo de las historias de usuario será conformado por tres leyendas seguidas de un espacio en donde el encargado de realizar la historia de usuario podrá redactar de manera concisa el requisito del sistema. En el caso de la leyenda “Me gustaría”, el encargado deberá escribir la acción que desea que el sistema realice, mientras que para la leyenda “Y así” se deberá redactar el fin con el que se propone la acción antes escrita. En la leyenda “Prueba” se deberán especificar los criterios de prueba para la aprobación del requisito especificado. Cabe señalar que para las leyendas de usuario que sean demasiado abstractas, éstas se podrán descomponer en historias de usuario más concretas, desglosando las tareas que se engloben en ella.

Debido a que el SADEC será desarrollado para cubrir las necesidades de distintas dependencias de la BUAP, se identificaron cuatro distintos clientes:

- Usuarios administradores de Tesorería General
- Usuarios administradores de Recursos Humanos
- Usuario administrador Global
- Trabajadores de la nómina institucional de la BUAP

A cada tipo de cliente se le solicitó que proporcionara un conjunto de historias de usuario de acuerdo a lo acordado en el backlog de requisitos, obteniendo un total de 35 historias de usuario.

## 2.3 Especificación de Requisitos

En Scrum, la cartera de producto es una lista priorizada de requisitos los cuales son implantados en la iteración que se especifique. A partir del análisis de las historias de usuario obtenidas durante la fase de recopilación de información (tomando como base el backlog de requisitos), es posible generar la cartera de producto (product backlog), en la que se especifican puntualmente tanto los requisitos del sistema como la iteración en la cual serán implantados. En las secciones siguientes se describe la fase de análisis de historias de usuario y se presenta el backlog del producto resultante.

### 2.3.1 Análisis de Historias de Usuario

Una de las fases del marco de trabajo Scrum consiste en llevar a cabo una fase de análisis de las historias de usuario obtenidas durante la recopilación de información, esto con el fin de obtener una cartera de producto concisa. La fase de análisis de historias de usuario tiene como objetivos:

- Identificar aquellas historias de usuario que se encuentren en un nivel abstracto
- Descomponer historias de usuario abstractas en dos o más historias concretas
- Analizar los criterios de prueba de cada historia de usuario y en caso de ser necesario, agregar, descartar o modificar criterios de prueba de la misma

Como se mencionó en la sección 2.2.2, se obtuvieron 35 historias de usuario. Una vez finalizada la fase de análisis de las mismas, se obtuvo un total de 42 historias de usuario debido a la descomposición de aquellas que se consideró se encontraban en un nivel de descripción abstracto.

Con respecto a los criterios de prueba, los cambios fueron mínimos, pues en algunos casos se optó por realizar pruebas de mayor rigor dependiendo de la criticidad de la operación descrita.

### 2.3.2 Cartera del Producto (Product Backlog)

En la metodología Scrum se denomina cartera del producto a una lista priorizada de ítems (requisitos) la cual es determinada y administrada por el cliente (product owner). En Scrum, durante el inicio del desarrollo de un producto, los ítems del backlog de producto son características requeridas para cumplir con una visión general del producto a desarrollar. Una vez en fase de desarrollo el backlog de producto puede contener nuevas características, cambios a características existentes, defectos por reparar, mejoras técnicas, etc.

En general, en el contexto de Scrum, se conoce como grooming a la actividad de generar la cartera de producto a partir del análisis realizado a la información recabada (historias de usuario).

Para el desarrollo del SADEC se optó por indicar en la cartera de producto la iteración a la que corresponde cada ítem contenido, dando como resultado la lista priorizada mostrada en la Figura 2.3.2.1.



<i>Sprint 0</i>	0	Establecer Arquitectura del sistema
	1	Diseño de Base de Datos
	2	Implantación de Base de Datos
	3	Implantación de esqueleto según arquitectura establecida
<i>Sprint 1</i>	4	Módulo de Generación de CFDI
	5	Pruebas de generación de CFDI
	6	Módulo de Sellado de CFDI
	7	Pruebas de sellado de CFDI
	8	Timbrado de CFDI
	9	Pruebas de timbrado de CFDI
<i>Sprint 2</i>	10	Implantación de filtros para visualización de CFDI
	11	Visualización y descarga de CFDI generados para administradores
	12	Visualización y descarga de CFDI sellados para administradores
	13	Visualización y descarga de CFDI timbrados para administradores
	14	Pruebas
<i>Sprint 3</i>	15	Implantación de autenticación de usuarios administrativos
	16	Pruebas de autenticación de usuarios administrativos
	17	Implantación de privilegios de usuario
	18	Pruebas de implantación de privilegios de usuario
<i>Sprint 4</i>	19	Administración de usuarios
	20	Administración de roles
	21	Administración de privilegios
	22	Pruebas
<i>Sprint 5</i>	23	Consulta y descarga de CFDI para trabajadores
	24	Implantación de autenticación de trabajadores
	25	Pruebas de autenticación de trabajadores

**Figura 2.3.2.1.** Product Backlog del SADEC

Como se puede observar en la figura 2.3.2.1, la cartera de producto resultante consta de seis iteraciones enumeradas, cada una con distintos requerimientos priorizados de acuerdo a lo analizado en las historias de usuario recabadas.

Cabe señalar que de acuerdo a los criterios de Scrum, al principio de cada iteración el equipo de desarrollo debe determinar un subconjunto de ítems de la cartera de producto que considera lograr implantar en tiempo y forma. Esta actividad es conocida como planeación

---

de la iteración o sprint planning. Durante esta actividad, el equipo de desarrollo crea una segunda cartera denominada cartera del sprint (sprint backlog), la cual describe mediante un conjunto de tareas detalladas cómo el equipo planea diseñar, construir, integrar, y probar el subconjunto seleccionado de características de la cartera de producto durante esa iteración en particular [3].



# Capítulo 3

## Diseño del SADEC

### Introducción

El diseño de software agrupa el conjunto de principios, conceptos y prácticas que llevan al desarrollo de un sistema de alta calidad. Durante esta etapa se crea una representación o modelo del software que proporciona detalles sobre: la arquitectura del software, estructuras de datos, interfaces y componentes que se necesitan para implantar el sistema [1].

En este capítulo se abordan los aspectos referentes al diseño del SADEC, tal como se indica en los primeros ítems de la cartera de producto presentada en el Capítulo 2.

Los objetivos de este capítulo son:

- Elegir y justificar la elección de la arquitectura del SADEC
- Diseñar la estructura de datos del sistema
- Obtener un diagrama Entidad-Relación de la base de datos del sistema

### 3.1 Arquitectura del Sistema

Como se presentó en el Capítulo 2 de este trabajo (sección 2.3.2), la elección de la arquitectura del sistema corresponde al ítem de mayor prioridad en la cartera de producto obtenida a partir del análisis de requisitos del SADEC. La elección de la arquitectura de un sistema de software es un proceso crítico que depende de las características particulares del sistema a desarrollar, así como del rendimiento esperado del mismo. Según el análisis presentado en el Capítulo 2, se determinó que el SADEC deberá contar con las siguientes características:

- El sistema deberá ser escalable, debido a las posibles modificaciones de los estándares y requerimientos de las especificaciones de CFDI establecidos por el Servicio de Administración Tributaria (SAT).
- Se deberá asegurar la persistencia de la información, esto, según los lineamientos establecidos por el SAT, y los requisitos propios de la BUAP.
- Debido a la naturaleza de la aplicación, se deberá considerar el manejo de volúmenes considerables de información.
- El sistema deberá ser capaz de permitir múltiples accesos concurrentes por parte de los usuarios a la información almacenada.
- Se deberá contar con una interfaz gráfica, la cual facilite la interacción entre los usuarios y el sistema.
- Deberá ser considerada la comunicación e integración con aplicaciones externas, debido a los lineamientos del SAT y los requerimientos propios de la BUAP.
- En cuanto a la operación del sistema se refiere, deberá ser posible la implantación y modificación de reglas de negocio específicas a los requerimientos particulares de la BUAP.

Del listado anterior, se puede observar que las características listadas se asimilan a aquellas con las que debe contar una aplicación empresarial.

Otro factor a tomar en cuenta durante la etapa de elección de la arquitectura de un sistema de software es el rendimiento esperado del mismo [9]. A partir de las características presentadas con anterioridad, se puede deducir que el SADEC deberá contar con las siguientes propiedades en cuanto a rendimiento se refiere:

- Tiempo de respuesta aceptable: La cantidad de tiempo que tome el sistema para procesar una solicitud externa deberá ser relativamente aceptable dependiendo de la complejidad de la solicitud.
- Buena capacidad de respuesta: La velocidad en que el sistema reconozca una solicitud (en oposición a su procesamiento) deberá ser lo menor posible.
- Latencia: El tiempo para obtener cualquier tipo de respuesta, incluso si la acción a realizar es inexistente deberá ser mínimo.
- Rendimiento: El sistema deberá ser capaz de realizar distintas actividades en un tiempo dado.

- Escalabilidad: Se deberá tomar en cuenta la afectación del rendimiento al agregar recursos (hardware).

### 3.1.1 Patrón arquitectural por capas

Bajo el contexto de desarrollo de software existen múltiples patrones arquitecturales, gran parte de ellos orientados al desarrollo de aplicaciones empresariales. Como ya se mencionó en el marco teórico de este trabajo (sección 1.6.3), un patrón arquitectural expresa un esquema de organización estructural fundamental para sistemas de software.

Después de un análisis a las características y propiedades con las que el SADEC deberá contar, se optó por utilizar un patrón arquitectural por capas dado que éste se adapta a las necesidades de una aplicación empresarial como es el caso del SADEC.

El patrón arquitectural por capas plantea la división de un sistema en distintas entidades; esta es una de las técnicas más comúnmente utilizada en el diseño de software, y consiste en dividir un sistema en subsistemas, organizados por capas, tal como se muestra en la figura 3.1.1.1.



**Figura 3.1.1.1.** División en capas de un sistema

En la figura 3.1.1.1 se puede observar cómo según el patrón arquitectural por capas, cada una de éstas reposa sobre una capa inferior. Bajo este esquema, la capa más alta utiliza varios servicios definidos por la capa inferior, sin embargo, la capa inferior desconoce de la existencia de una capa superior.

Algunas de las ventajas de utilizar el patrón arquitectural por capas, y por ende dividir un sistema en múltiples capas son:

- Posibilidad de entender una capa como un todo, sin tener necesariamente conocimiento de las otras capas.
- Las capas pueden ser sustituidas con implantaciones alternativas de los mismos servicios.
- Minimización de dependencias entre subsistemas.
- Se da lugar a la estandarización
- Una vez construida una capa, ésta puede ser utilizada para distintos servicios de niveles superiores.

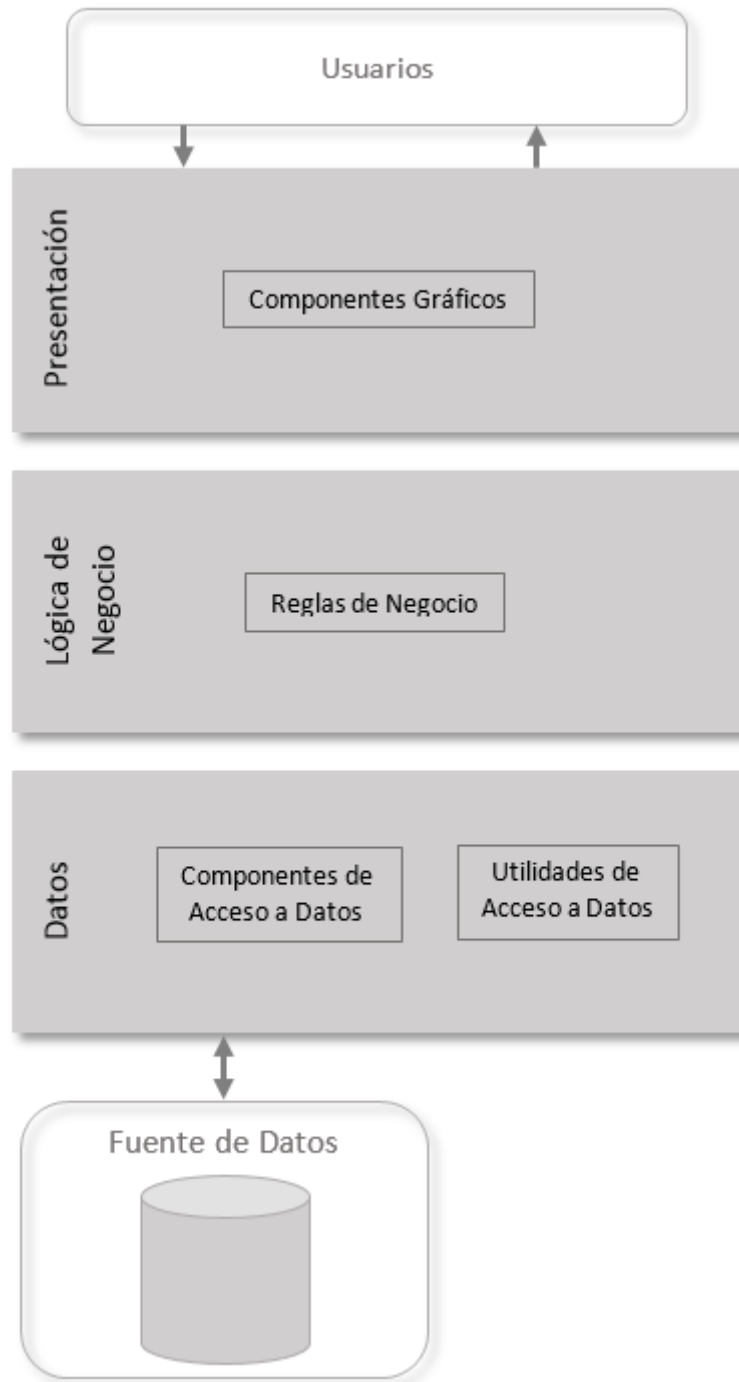
### 3.1.1.1 Patrón arquitectural por capas en Aplicaciones Empresariales

El patrón arquitectural por capas aplicado a aplicaciones empresariales consta regularmente de tres capas principales: presentación, fuente de datos y lógica de dominio. En general, bajo este contexto, las principales funciones de cada capa son:

- Capa de Presentación: Esta capa se encarga de proporcionar un medio para la interacción entre el usuario y el software. Sus principales responsabilidades son el despliegue de información al usuario y la interpretación de comandos provenientes del usuario en acciones sobre capas inferiores. Como ejemplos de implementaciones de esta capa son las consolas de línea de comandos, menús basados en texto e interfaces de usuario vía un navegador web basado en HTML.
- Capa de Fuente de datos: La principal responsabilidad de esta capa es establecer comunicación con otros sistemas que llevan a cabo tareas de parte de la aplicación; en aplicaciones empresariales usualmente se trata de sistemas de bases de datos.
- Capa de Lógica de dominio: Esta capa, también conocida como Capa de lógica de negocio, se encarga de las operaciones que una aplicación necesita realizar para el dominio sobre el cual trabaja. Se involucran cálculos basados en entradas e información almacenada, validaciones de cualquier dato proveniente de la capa de presentación, y resolución de la información que se debe proporcionar de acuerdo a comandos recibidos desde la capa de presentación.

En particular para el SADEC, se optó por dividir la capa de fuente de datos en dos subcapas (fuente de datos y acceso a datos) con el fin de proporcionar escalabilidad al sistema. La subcapa de fuente de datos será la base de datos del sistema, mientras que la subcapa de acceso a datos contendrá las reglas de negocio establecidas para acceder a la información almacenada en la base de datos.

En la figura 3.1.1.1.1 se puede apreciar el modelo final de la arquitectura del SADEC.



**Figura 3.1.1.1.1.** Arquitectura del SADEC

Como se muestra en la figura anterior, el sistema en desarrollo constará de tres capas, y dos subcapas las cuales interactuarán entre sí. En las secciones posteriores se describen las principales funciones y responsabilidades de cada capa que conforma la arquitectura del SADEC.



### 3.1.2 Capa de Presentación del SADEC

En general, como se describió en secciones anteriores, la capa de presentación es la responsable de proporcionar la interacción de los usuarios con el sistema. Como se especificó en el análisis de requerimientos, el SADEC se ejecutará sobre una plataforma web. Es por esto que la capa de presentación estará compuesta de interfaces de usuario diseñadas para ser visualizadas en un navegador Web basado en HTML.

### 3.1.3 Capa de Lógica de Negocio del SADEC

La capa de lógica de negocio es la encargada de llevar a cabo las operaciones sobre el dominio de la aplicación, por lo que esta capa deberá:

- Validar y procesar la información proveniente de la capa de presentación.
- Recuperar información necesaria mediante la subcapa de acceso a datos para realizar las operaciones que se soliciten.
- Actualizar la subcapa de fuente de datos (base de datos) mediante la subcapa de acceso a datos según se requiera.

### 3.1.4 Subcapa de Acceso a Datos del SADEC

Debido a la complejidad y la criticidad de las operaciones contenidas en esta capa, se optó por utilizar el patrón arquitectural para lógica de negocio llamado Table Module. Este patrón organiza la lógica de dominio con una clase por cada tabla en la base de datos, por lo que una sola instancia de clase contiene todos los procedimientos que actuarán sobre los datos. [9]

Una de las grandes ventajas y la razón principal de la utilización del patrón arquitectural Table Module para la capa de lógica de negocio del SADEC, es la manera en que ésta se adapta al resto de las capas de la arquitectura establecida, principalmente a la subcapa de acceso a datos, en la cual, al interactuar con una base de datos relacional, se puede implantar el patrón arquitectural de datos Record Set, patrón ampliamente utilizado por el framework Microsoft .NET, tecnología bajo la cual será desarrollado el sistema.

### 3.1.5 Subcapa de Fuente de Datos del SADEC

En la capa de fuente de datos residirá la base de datos del sistema. Según los requisitos del sistema establecidos, ésta será una base de datos relacional.

## 3.2 Diseño de la Base de Datos del SADEC

El diseño de una estructura de datos puede ser basado en un sistema existente, el cual es posible mapear a tablas, columnas y filas de una base de datos. En el diseño de una estructura de datos cada tabla representa un objeto o entidad; dentro de cada tabla, cada columna almacena un elemento de información (atributo) para la entidad, y cada fila almacena una instancia de la entidad.

En general, se pueden seguir cinco pasos para diseñar una estructura de datos; como primer paso se deberán identificar los elementos de datos que se necesitan almacenar en la base de datos, el segundo paso consiste en descomponer elementos complejos en componentes más pequeños siempre que tenga sentido. El tercer paso es identificar las tablas que constituirán el sistema y determinar qué elementos de datos serán asignados como columnas a cada tabla. Como cuarto paso se definirán las relaciones entre tablas identificando las llaves primarias y foráneas de cada una de ellas. Finalmente, como quinto paso, la estructura de datos resultante deberá ser sometida a un proceso de normalización con el fin de reducir posibles redundancias en los datos [10].

En las siguientes secciones se detalla el proceso de diseño de la estructura de datos del SADEC siguiendo los pasos antes mencionados. Como resultado final de este proceso se obtendrá un diagrama Entidad-Relación, el cual será una representación (modelo) de la base de datos del sistema.

### **3.2.1 Identificación de los Elementos de Datos**

Es posible utilizar distintas técnicas para identificar los elementos de datos que serán requeridos por un sistema. La identificación de elementos de datos del SADEC se llevó a cabo mediante un análisis de las historias de usuario recabadas durante la fase de recolección de requisitos del sistema. La tabla 3.2.1.1 presenta todos los posibles elementos de datos (en orden alfabético) identificados para el SADEC.

#	ELEMENTO
1	Acuse de Cancelación
2	Administrador
3	Año
4	Bitácora
5	CFDI
6	CFDI PDF
7	Datos personales del Empleado
8	Dependencia
9	Empleado
10	Error
11	Estado CFDI
12	Nómina
13	Privilegio
14	Quincena
15	Reporte
16	Rol
17	Sello
18	Timbre
19	Usuario

**Tabla 3.2.1.1.** Posibles elementos de datos (en orden alfabético) de la estructura de datos del SADEC

Como se puede observar en la tabla 3.2.1.1, se obtuvieron diecinueve posibles elementos de datos. Tras una depuración de esta lista, se descartaron aquellos elementos de datos que se consideraron innecesarios o que pudieran ocasionar redundancias en la información en pasos posteriores. El listado final de elementos de datos se puede observar en la figura 3.2.1.2.

#	ELEMENTO
1	Cancelación
2	Dependencia
3	Documento
4	Empleado
5	Estado de CFDI
6	Nómina
7	Privilegio
8	Quincena
9	Reporte
10	Rol
11	Sello
12	Timbre
13	Usuario

**Figura 3.2.1.2.** Listado final de elementos de datos (en orden alfabético) de la estructura de datos del SADEC

### 3.2.2 Subdivisión de Elementos de Datos

El objetivo de subdividir los elementos de datos obtenidos en la sección anterior es obtener los valores útiles más pequeños de dichos elementos para su uso en pasos posteriores en el proceso de diseño de la estructura de datos. En la presente sección se muestran las subdivisiones de los distintos elementos de datos (en orden alfabético) identificados para el SADEC presentados con anterioridad en la figura 3.2.1.2.

#### Elemento de dato Cancelación

Este elemento de dato corresponde a un acuse de cancelación proporcionado por el SAT posterior a una petición correspondiente por parte del emisor de un CFDI. Este elemento se dividió en los subelementos Acuse y Fecha, ya que son datos con los que una cancelación de CFDI debe contener. El resultado de la división del elemento Cancelación se puede observar en la tabla 3.2.2.1.

Cancelación	
Acuse	Fecha

**Tabla 3.2.2.1.** Subdivisión del elemento de dato Cancelación

#### Elemento de dato Dependencia

El elemento de dato dependencia representa la información de una dependencia de la BUAP. Toda dependencia debe contar con un nombre, mínimamente un responsable y una dirección

física, es por esto que la división del elemento dependencia dio como resultado tres subelementos, los cuales se muestran en la tabla 3.2.2.2.

Dependencia		
Nombre	Responsable	Dirección

**Tabla 3.2.2.2.** Subdivisión del elemento de dato Dependencia

### Elemento de dato Documento

Como se mencionó en el marco teórico de este trabajo (Capítulo 2, sección 2.1), un comprobante fiscal digital por Internet (CFDI) cuenta con validez fiscal únicamente cuando éste ha sido sometido satisfactoriamente a un proceso de timbrado a través de un proveedor autorizado de certificación (PAC). Es por lo anterior que se decidió utilizar el elemento Documento como representación de un documento que contiene la información necesaria para ser sometido posteriormente a los procesos de sellado y timbrado, y que por lo tanto, aún no cuenta con validez oficial ante el SAT. La subdivisión de este elemento de dato se puede observar en la tabla 3.2.2.3.

Documento				
ID de receptor	Quincena	Año	XML	Responsable de generación

**Tabla 3.2.2.3.** Subdivisión del elemento de dato Documento

Como se puede observar en la tabla anterior, la subdivisión del elemento Documento dio como resultado cinco subelementos: ID de receptor, que se refiere al identificador único del trabajador a quien le será emitido un CFDI posteriormente; Quincena, que representará la quincena del año (representación numérica entre 1 y 23) en la cual se emite el CFDI; Año, representando el año en el que el CFDI es emitido; XML, que representa un documento en formato XML cuya estructura se especifica según los lineamientos del SAT; Responsable de la generación, el cual representa la cuenta del usuario administrador responsable de la generación del documento.

### Elemento de dato Empleado

El elemento de dato Empleado es la representación de un trabajador de la BUAP, quien será el receptor de al menos un comprobante fiscal digital. Todo trabajador de la Institución cuenta con un identificador único, y, según los requisitos del sistema, un trabajador deberá poder consultar los CFDI que le hayan sido emitidos, por lo que éste contará con una cuenta de usuario en el sistema, además, en caso de existir algún inconveniente, deberá poder ser contactado vía correo electrónico o telefónica, es por esto que a partir de la subdivisión de este elemento se obtuvieron los subelementos de la tabla 3.2.2.4.

Empleado				
ID	Nombre de usuario	Contraseña	Teléfono	Correo electrónico

**Tabla 3.2.2.4.** Subdivisión del elemento de dato Empleado

### Elemento de dato Estado de CFDI

Este elemento puede ser visualizado como la representación de un catálogo que contiene los posibles estados en los que se encuentra un comprobante fiscal digital (generado, sellado, timbrado). Los datos necesarios identificados para este elemento son un identificador único del estado, un nombre para el estado y una descripción del mismo. En la tabla 3.2.2.6 se puede observar la subdivisión de este elemento de dato.

Estado del CFDI		
ID de estado	Nombre	Descripción

**Tabla 3.2.2.6.** Subdivisión del elemento de dato Estado del CFDI

### Elemento de dato Nómina

El elemento de dato Nómina representa un catálogo en el que se contienen las posibles nóminas a las que puede pertenecer un trabajador de la BUAP. Al igual que el elemento Estado del CFDI, se obtuvieron tres subelementos con el fin de identificar a cada nómina de la Institución. La subdivisión del elemento de dato Nómina se muestra en la tabla 3.2.2.7.

Nómina		
ID de nómina	Nombre	Descripción

**Tabla 3.2.2.7.** Subdivisión del elemento de dato Nómina

### Elemento de dato Privilegio

Al igual que el elemento anterior, el elemento de dato Privilegio fue considerado para fungir como catálogo dentro del sistema, por lo que la subdivisión de éste se presenta en la tabla 3.2.2.8.

Privilegio		
ID del privilegio	Nombre	Descripción

**Tabla 3.2.2.8.** Subdivisión del elemento de dato Privilegio

### Elemento de dato Quincena

En un principio, la intención del elemento Quincena era representar en forma de catálogo las 24 quincenas que conforman el año, sin embargo, se optó por identificar éstas mediante un número, por lo que se descarta este elemento.

### Elemento de dato Reporte

Este elemento representa los reportes que podrán ser generados a partir de la información almacenada en la base de datos del sistema. Debido a la existencia de distintos tipos de reportes dentro del SADEC, se descartó este elemento, utilizando otras técnicas para la implantación de reportes.

### Elemento de dato Rol

Un rol es un conjunto de privilegios, el cual puede ser asignado a uno o más usuarios. El elemento de dato Rol representa un catálogo de los distintos tipos de roles del sistema, por lo que debe contar con un identificador único dentro del catálogo, un nombre, una descripción y la lista de privilegios por los que el rol estará conformado. La subdivisión de este elemento se muestra en la tabla 3.2.2.9.

Rol			
ID del rol	Privilegios	Nombre	Descripción

**Tabla 3.2.2.9.** Subdivisión de elemento de dato Rol

### Elemento de dato Sello

Para someter un CFDI a un proceso de timbrado a través de un PAC, es menester someterle antes a un proceso de sellado. Durante el proceso de sellado se anexan distintos componentes al CFDI candidato a timbrar. El elemento de dato Sello representa un comprobante fiscal digital sellado y listo para ser timbrado, por lo que se identifican los subelementos CFDI original (CFDI sin sellar), Cadena original (requerimiento del SAT), Sello (requerimiento del SAT), XML (cfdi sellado) y Fecha de sellado. La subdivisión de este elemento de dato se puede observar en la tabla 3.2.2.10.

Sello				
CFDI original	Cadena Original	Sello	XML	Fecha de sellado

**Tabla 3.2.2.10.** Subdivisión de elemento de dato Sello

### Elemento de dato Timbre

Por razones similares al elemento de dato Sello, se optó por separar los componentes del elemento Timbre de un CFDI. Durante el proceso de timbrado se anexan datos proporcionados por el SAT a un CFDI, otorgando de esta manera validez oficial; dos de estos elementos son UUID (identificador único del CFDI) y Número de certificado (proporcionado por el SAT). Por lo anterior, la subdivisión del elemento Timbre se obtiene tal como se muestra en la tabla 3.2.2.11.

Timbre						
UUID	CFDI Original	Sello	Sello SAT	No. de certificado	Fecha de timbrado	XML

**Tabla 3.2.2.11.** Subdivisión de elemento de dato Timbre

### Elemento de dato Usuario

Como se especificó en los requisitos del sistema, existirán dos tipos de cuenta de usuario:

- Trabajadores
- Administradores

El elemento de dato Usuario representa una cuenta de usuario administrador, por lo que se identifican los subelementos ID (identificador único del usuario interno al sistema), Nombre (nombre real del responsable de la cuenta), Nombre de usuario (identificador utilizado para acceder al sistema junto con la contraseña), Contraseña (contraseña utilizada para acceder al sistema junto con el nombre de usuario) y Rol (conjunto de privilegios que le serán otorgados a la cuenta de usuario). La subdivisión del elemento de dato Usuario resultó como se observa en la tabla 3.2.2.12.

Usuario				
ID	Nombre	Nombre de usuario	Contraseña	Rol

**Tabla 3.2.2.12.** Subdivisión de elemento de dato Usuario

### 3.2.3 Identificación de tablas y asignación de columnas

El tercer paso en el proceso de diseño de la estructura de datos de un sistema consiste en determinar las entidades principales que conformarán la estructura. La selección de estas entidades se lleva a cabo tomando en cuenta los elementos de datos analizados en el segundo paso del proceso (sección 3.2.2). En el caso del SADEC, se sometieron a un proceso de análisis los elementos de datos obtenidos dando como resultado las entidades principales de la figura 3.2.3.1.

#	Cancelación
1	Cancelación
2	Documento
3	Sello
4	Timbre
5	Usuario

**Tabla 3.2.3.1.** Entidades principales de la estructura de datos

Con el fin de obtener las tablas (entidades) y columnas (elementos de datos) que conformarán la estructura de datos del sistema, es necesario elegir aquellos elementos de datos que serán asociados a cada una de las entidades principales identificadas. En la figura 3.2.3.2 se pueden observar las entidades y elementos elegidos para formar parte de la estructura de datos del SADEC. Cabe mencionar que se obviaron aquellas entidades que fungirán como catálogos en la estructura de datos.



Documento	Sello	Timbre	Cancelación	Usuario
Identificador	Identificador	Identificador	Identificador	Identificador
Responsable	Documento original	Documento original	Documento timbrado	Nombre
Receptor	Cadena Original	Sello original	UUID	Nombre de usuario
Nómina	Sello	Sello SAT	Fecha	Contraseña
Quincena	Fecha de sellado	No. Certificado	XML	Rol
Año	XML	Fecha de timbrado		
Fecha		XML		
XML		UUID		

**Tabla 3.2.3.2.** Entidades principales del sistema

### 3.2.4 Identificación de Llaves Primarias y Foráneas

A partir de las entidades (tablas) y elementos que conformarán la estructura de datos es posible determinar las relaciones entre entidades mediante la identificación de llaves primarias y foráneas [10]. En los apartados siguientes se expone el proceso de identificación de llaves primarias y foráneas de toda entidad que forma parte de la estructura de datos del SADEC.

#### Entidad Documento

Como se ha mencionado con anterioridad, y según los requisitos del sistema, un documento deberá contar con un identificador único interno al sistema. Debido a las características de este elemento de dato, éste servirá como llave primaria de la entidad.

Según los requisitos del sistema, deberá ser posible la identificación del responsable de generar un documento, por lo que se determinó que el elemento de dato que representa el identificador interno de un usuario administrador fungirá como llave primaria para la entidad. Otro de los requisitos del sistema es la plena identificación del trabajador a quien se emite el documento. En este caso, el elemento que representa el identificador institucional de cada trabajador dentro de la entidad Documento puede ser una llave foránea.

Para identificar a qué nómina de la Institución pertenece el documento generado, se optó por tomar como llave foránea el elemento de dato que representa el identificador de dicha nómina.

En la tabla 3.2.4.1 se presenta la elección de llaves primarias y foráneas para la entidad Documento.

Entidad:	Documento
Llave Primaria:	ID del documento
Llave(s) Foránea(s):	ID del responsable, ID de receptor, ID de nómina

**Tabla 3.2.4.1.** Llaves primarias y foráneas de la entidad Documento

### Entidad Sello

En el caso de la entidad Sello, se optó por establecer un identificador único interno al sistema como llave primaria.

Un sello deberá estar asociado a un documento, por lo que el elemento de dato que representa el identificador del documento al que pertenece el sello puede ser tomado como llave foránea.

En la tabla 3.2.4.2 se muestran las llaves primaria y foránea de la entidad Sello.

Entidad:	Sello
Llave Primaria:	ID del sello
Llave(s) Foránea(s):	ID del documento

**Tabla 3.2.4.2.** Llaves primaria y foránea de la entidad Sello

### Entidad Timbre

Un CFDI que ha sido sometido a un proceso de timbrado cuenta con un identificador único proporcionado por el SAT (UUID). Las propiedades de éste identificador permiten que sea utilizado como llave primaria de la entidad. Un timbre deberá estar asociado a un sello, por lo que el elemento de dato que representa el identificador de la entidad sello a la cual pertenece el timbre puede ser utilizado como llave foránea. En la tabla 3.2.4.3 se pueden visualizar las llaves de la entidad Timbre.

Entidad:	Timbre
Llave Primaria:	UUID
Llave(s) Foránea(s):	ID del sello

**Tabla 3.2.4.3.** Llaves primaria y foránea de la entidad Timbre

### Entidad Cancelación

La entidad Cancelación representará el acuse de cancelación de un comprobante fiscal validado por el SAT. Debido a que esta entidad deberá estar asociada a un Timbre, se optó por establecer el elemento UUID del timbre al que se asocia como llave foránea, y un identificador interno al sistema como llave primaria. En la tabla 3.2.4.4 se pueden observar las llaves primaria y foránea de ésta entidad.

Entidad:	Cancelación
Llave Primaria:	ID acuse de cancelación
Llave(s) Foránea(s):	UUID de CFDI

**Tabla 3.2.4.4.** Llaves primaria y foránea de la entidad Cancelación

### Entidad Usuario

Como se ha explicado con anterioridad, un usuario administrador contará con un identificador interno, el cual puede servir como llave primaria de la entidad. Debido a que se deberá conocer la dependencia a la que el usuario pertenece y los privilegios con los que éste cuenta, se optó por establecer como llaves foráneas los elementos de dato que representan el identificador de la dependencia y del rol. Las llaves primarias y foráneas de esta entidad se pueden visualizar en la tabla 3.2.4.5.

Entidad:	Usuario
Llave Primaria:	ID usuario administrador
Llave(s) Foránea(s):	ID de rol, ID dependencia

**Tabla 3.2.4.5.** Llaves primarias y foráneas de la entidad Usuario

### Entidad Nómina

Dado que esta entidad representará un catálogo de la estructura de datos el cual contendrá las distintas nóminas con las que cuenta la BUAP, se consideró como llave primaria el elemento de dato identificador de nómina, y no se identificaron llaves foráneas. La tabla 3.2.4.6 muestra la llave primaria de esta entidad.

Entidad:	Nómina
Llave Primaria:	ID Nómina
Llave(s) Foránea(s):	-

**Tabla 3.2.4.6.** Llave primaria de la entidad Nómina

### Entidad Dependencia

La entidad dependencia representará un catálogo al cual estarán asociados los usuarios administradores del sistema. Es por esto que se consideró como llave primaria el elemento de dato identificador, y no se contemplaron llaves foráneas. En la tabla 3.2.4.7 se visualiza la llave primaria de esta entidad.

Entidad:	Dependencia
Llave Primaria:	ID Dependencia
Llave(s) Foránea(s):	-

**Tabla 3.2.4.7.** Llave primaria de la entidad Dependencia

### Entidad Rol

La entidad Rol representará los distintos conjuntos de privilegios con los que un usuario administrador podrá contar. Al igual que los catálogos anteriores, se optó por establecer como llave primaria el elemento de dato identificador de esta entidad, lo cual se puede visualizar en la tabla 3.2.4.8.

Entidad:	Rol
Llave Primaria:	ID Rol
Llave(s) Foránea(s):	-

**Tabla 3.2.4.8.** Llave primara de la entidad Rol

### Entidad Privilegio

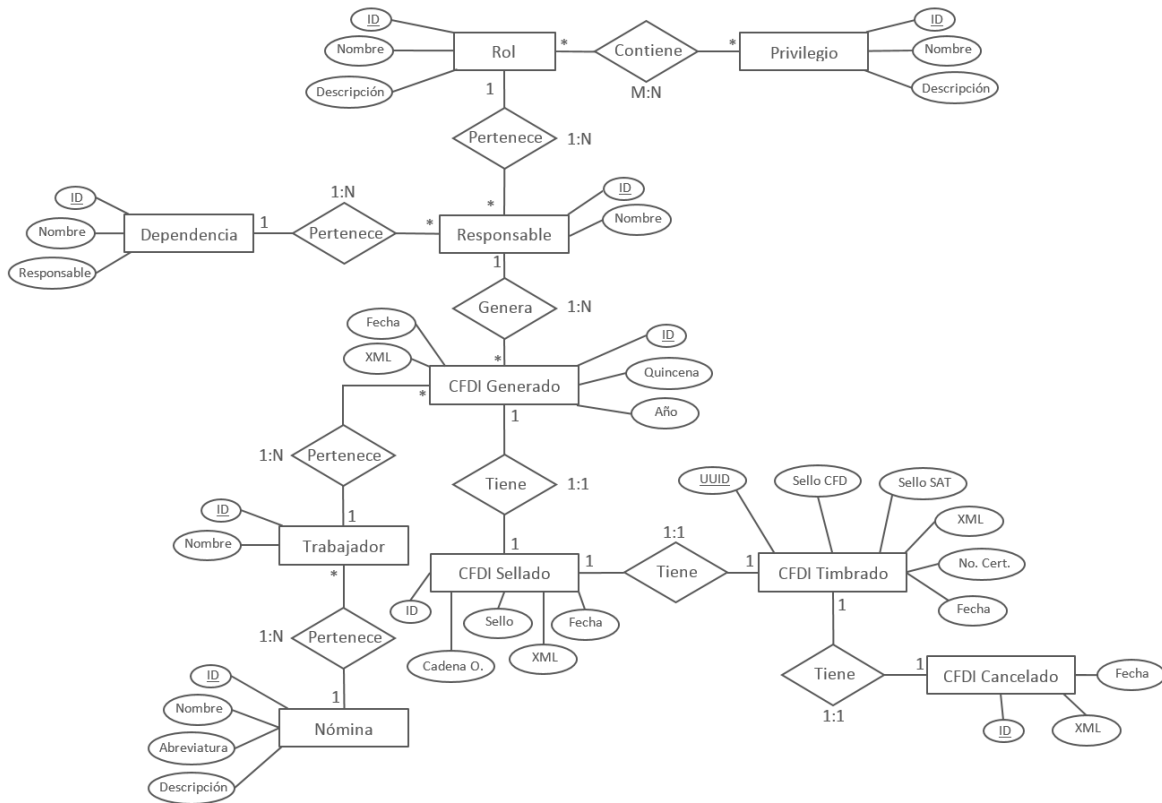
Esta entidad representará un catálogo de privilegios dentro de la estructura de datos del sistema, por lo que únicamente contará con una llave primaria la cual será el elemento de dato que representa el identificador de un privilegio. La llave primaria de esta entidad se puede visualizar en la tabla 3.2.4.9.

Entidad:	Privilegio
Llave Primaria:	ID Privilegio
Llave(s) Foránea(s):	-

**Tabla 3.2.4.9.** Llave primaria de la entidad Privilegio

## 3.2.5 Diagrama Preliminar de la Estructura de Datos

Como resultado del análisis realizado a los elementos de datos, entidades, y llaves primarias y foráneas obtenidas en las secciones anteriores, se obtuvo el diagrama Entidad-Relación preliminar de la estructura de datos del sistema. Este diagrama puede observarse en la figura 3.2.5.1.



**Figura 3.2.5.1.** Diagrama Entidad-Relación Preliminar de la estructura de datos

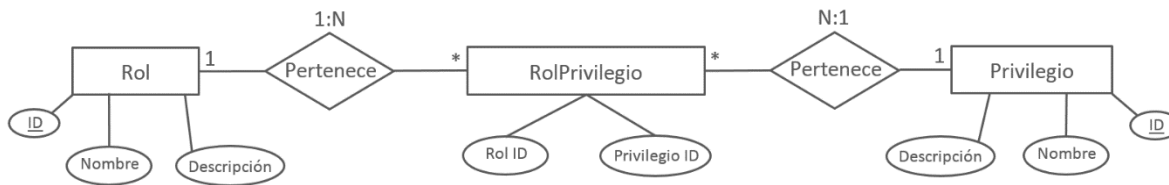
Como paso final en el proceso de diseño de la estructura de datos del sistema, y con el fin de evitar posibles redundancias en la información, el diagrama presentado en la figura 3.3.5.1 deberá ser sometido a un proceso de normalización.

### 3.2.6 Normalización de la Estructura de Datos

Como último paso en el proceso de diseño de la estructura de datos del SADEC se verificará si ésta se encuentra normalizada. En esta sección se describen los pasos involucrados en el proceso de normalización de la estructura de datos. Como resultado de este proceso se obtendrá una estructura de datos normalizada en tercera forma normal (3NF).

#### Primera Forma Normal (1NF)

La primera forma normal consiste en evitar que en una entidad existan grupos repetitivos. Realizando un análisis en el diagrama Entidad-Relación preliminar de la estructura de datos del SADEC (figura 3.2.5.1), se puede observar esta situación en la asociación de las entidades Rol y Privilegio, pues, según los requisitos del sistema, distintos roles podrán compartir un mismo privilegio, a la vez que un mismo privilegio podrá pertenecer a distintos roles. Para solventar esta situación se añadió una nueva entidad a la estructura de datos, cuyos atributos serán dos llaves foráneas, las cuales asocian la entidad con las entidades Rol y Privilegio, evitando así la repetición de grupos en la entidad Rol. El resultado de esta modificación se presenta en la figura 3.2.6.1.



**Figura 3.2.6.1.** Relación Rol-Privilegio (1NF)

Después de un análisis al diagrama de la estructura de datos con la modificación anterior, se llegó a la conclusión de que no son necesarios más cambios, por lo que ésta se encuentra en primera forma normal (1NF).

### Segunda Forma Normal (2NF)

Una entidad se considera en segunda forma normal, si además de estar en 1NF, toda columna que no sea una llave debe depender de la llave primaria. Analizando el diagrama Entidad-Relación de la figura 3.2.5.1 posterior a la modificación realizada en la sección 3.2.6, se determinó que la estructura de datos se encuentra en segunda forma normal (2NF), pues no se detectaron dependencias hacia columnas que no funjan como llave primaria.

### Tercera Forma Normal (3NF)

Una entidad se encuentra en tercera forma normal (3NF), si además de estar en segunda forma normal (2NF), los atributos que no forman parte de ninguna clave candidata facilitan información sólo acerca de la(s) clave(s) y no acerca de los atributos. Debido a que se utilizaron identificadores únicos para cada relación de la estructura de datos, se llegó a la conclusión de que una vez aplicados los cambios anteriores, la estructura de datos del sistema se encuentra en tercera forma normal.

### 3.2.7 Diagrama de Clases

Una vez realizados los pasos recomendados en [10] para el proceso de diseño de una estructura de datos, es posible obtener el diagrama Entidad-Relación definitivo de la misma.

Según los requisitos del sistema, es necesario identificar la nómina en la que se encuentra el trabajador a quien se emite un comprobante fiscal digital, sin embargo, un trabajador puede ser trasladado de una nómina a otra. Por esta situación se optó por modificar la entidad que representa un documento generado para contemplar la nómina en la que se encuentra el receptor del documento al momento de la generación del mismo. Por otra parte, como requisito del sistema, un trabajador deberá poder consultar los comprobantes fiscales digitales que se le hayan emitido. Para solventar esta situación, se consideró proporcionar a todo trabajador perteneciente a alguna Nómina Institucional de la BUAP una cuenta de usuario mediante la cual sea posible acceder al sistema, por lo que la entidad trabajador sufrió un par de modificaciones al añadir los elementos de datos que representan el nombre de usuario y contraseña de la cuenta de cada trabajador.

En la figura 3.2.7.1 se presenta el diagrama de clases final de la estructura de datos, obtenido a partir del diagrama Entidad-Relación presentado anteriormente sometido a un proceso de normalización.

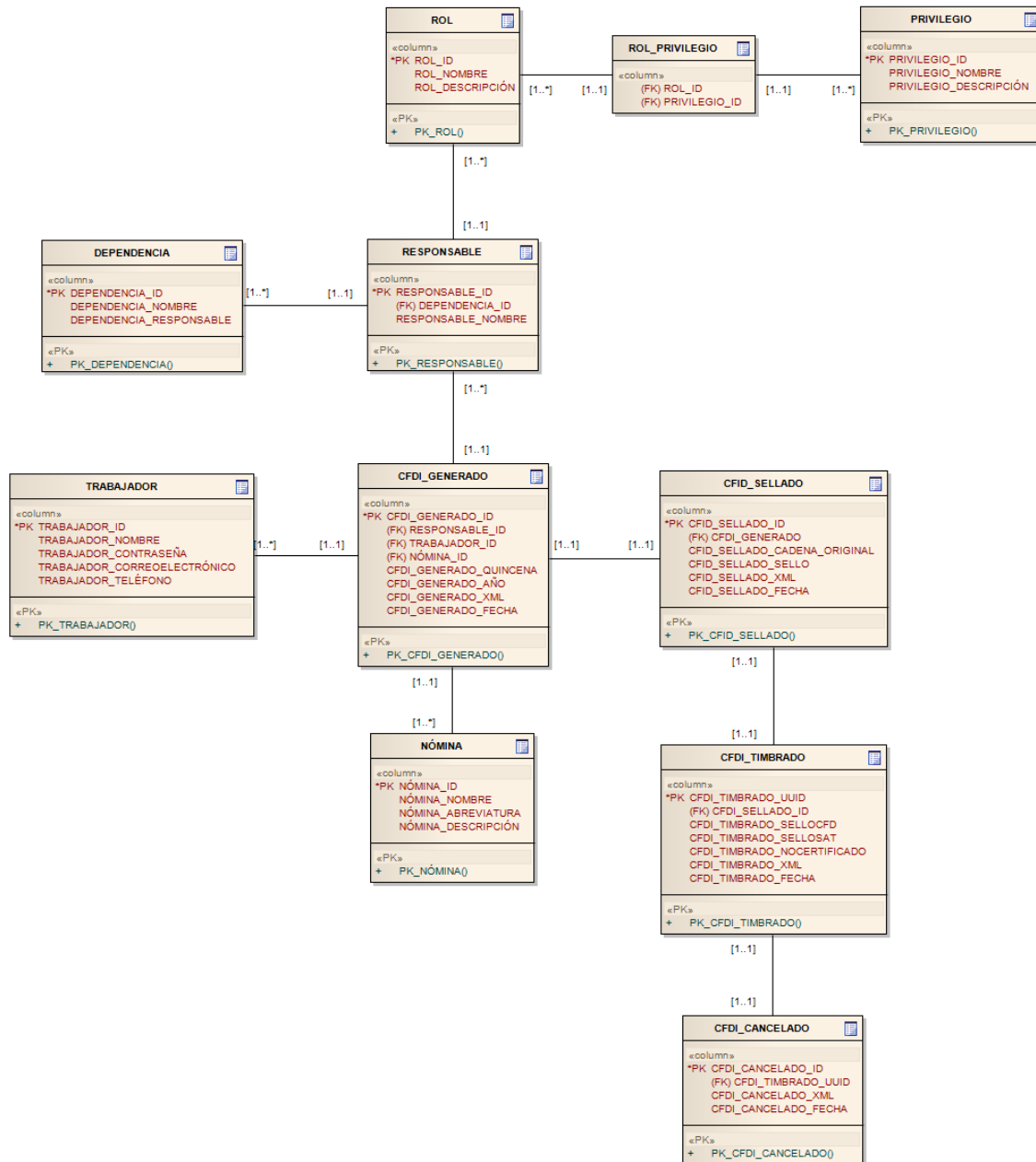


Figura 3.2.7.1. Diagrama de clases del SADEC





# CAPÍTULO 4

## Implantación del SADEC

### Introducción

En este capítulo se detallará la fase de implantación del SADEC, tomando como base el análisis y diseño del mismo realizado en el Capítulo 3, así como las herramientas de software y hardware a utilizar. La implantación del sistema se realizará siguiendo las acciones priorizadas en el backlog de producto (Capítulo 2), en el cual se establecieron seis iteraciones, cada una con tareas específicas priorizadas. Durante el presente capítulo se detallará el desarrollo de cada una de estas iteraciones junto con las tareas involucradas.

Los objetivos que se plantean en este capítulo son:

- Especificar y describir las herramientas de hardware y software a utilizar durante la fase de implantación del sistema
- Detallar la fase de implantación de la base de datos del sistema (Iteración 0) siguiendo el diseño previo (Capítulo 3)
- Describir la implantación del esqueleto de la arquitectura del sistema de acuerdo a las especificaciones presentadas en el Capítulo 3 (Iteración 0)
- Describir las etapas de desarrollo y pruebas del sistema según los puntos establecidos y priorizados en la cartera de producto presentado en el Capítulo 2 (Iteraciones 1-5)

## 4.1 Equipo de Hardware y Herramientas de Software

En esta sección se presentan y describen las herramientas de hardware y software necesarias para la implantación y correcto funcionamiento del SADEC. Cabe señalar que debido a los requisitos establecidos por la Benemérita Universidad Autónoma de Puebla, el SADEC será implantado sobre la infraestructura física y lógica propia de la Institución.

### 4.1.1 Equipo de hardware

El siguiente listado presenta el equipo de hardware utilizado durante la fase de desarrollo e implantación del SADEC junto con sus características principales:

- Computadora personal HP (desarrollo)
  - Procesador Intel Core 2 Duo a 3.00 GHz
  - 2.00 GB de memoria RAM
  - Sistema Operativo Microsoft Windows 7
- Servidor físico con Sistema Operativo AIX 5.3
- Servidor físico con Sistema Operativo Microsoft Windows Server 2003

### 4.1.2 Herramientas de Software

Las herramientas de software a utilizar durante esta fase son:

- Microsoft Visual Studio 2008/2013
- Microsoft Internet Information Services
- T.O.A.D. for Oracle IV
- Oracle Database 11g R2
- OpenSSL V1.0.1j Light

## 4.2 Implantación de Base de Datos del Sistema

Debido a la naturaleza del SADEC, la implantación de la base de datos es una fase crítica. De acuerdo al backlog de producto, esta etapa es prioritaria (iteración 0, tarea 2) en el desarrollo del sistema ya que será la capa base según el patrón arquitectural elegido para su implantación.

El diagrama Entidad-Relación presentado en la fase de diseño (Capítulo 3) será la referencia principal para la implantación de la base de datos del sistema. El proceso de implantación de la base de datos se desarrollará en las secciones posteriores.

### 4.2.1 Tablas de la Base de Datos

Tal como se detalló en el diagrama Entidad-Relación generado y analizado en la fase de diseño del sistema (Capítulo 3), la base de datos contará con once tablas, cuatro de las cuales serán contempladas como catálogos estáticos de información. De acuerdo a las características

con las que cuenta el RDBMS de Oracle, se optó por estructurar un script genérico para la creación de tablas, el cual se muestra en la figura 4.2.1.1.

```
1 create table sadec_table_name
2 (
3   sadec_table_name_col2 TYPE NULL/NOT NULL,
4   sadec_table_name_col3 TYPE NULL/NOT NULL,
5   sadec_table_name_col4 TYPE NULL/NOT NULL,
6   ...
7 )
8 tablespace sadec_table_space_name
9   pctfree 10
10  initrans 1
11  maxtrans 255
12  storage
13  (
14    initial 64K
15    next 1M
16    minextents 1
17    maxextents unlimited
18  );
19
20 alter table sadec_table_name
21 add constraint PK_constraint_name primary key (sadec_table_name_coln)
22 using index
23 tablespace sadec_table_name
24 pctfree 10
25 initrans 2
26 maxtrans 255
27 storage
28 (
29   initial 64K
30   next 1M
31   minextents 1
32   maxextents unlimited
33 );
```

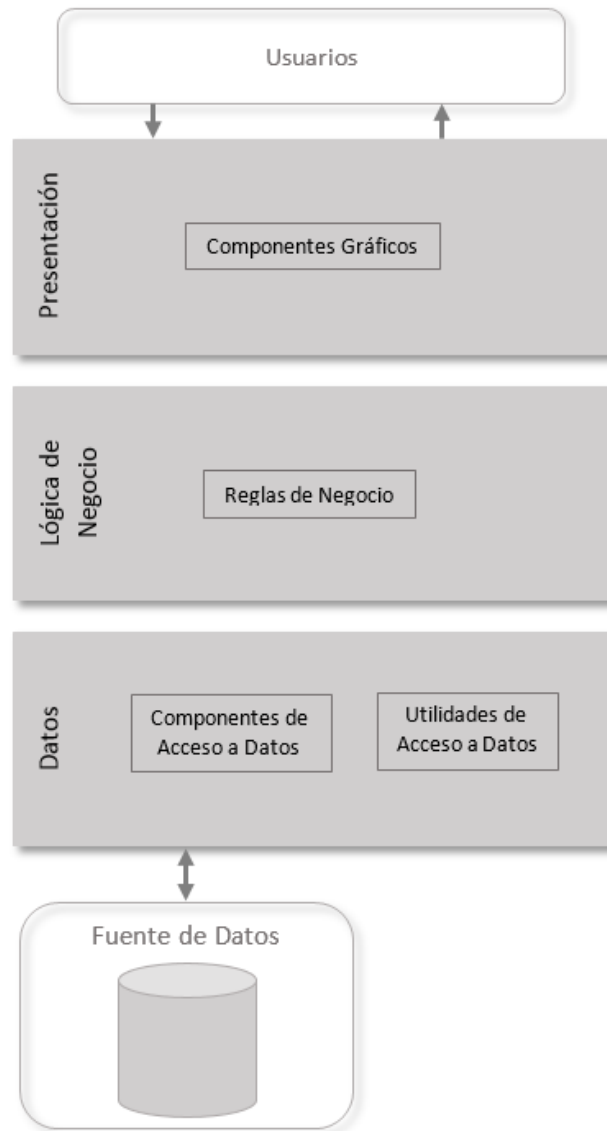
**Figura 4.2.1.1.** Estructura de script genérico para la creación de tablas de la base de datos.

Como se puede observar en el script de la figura anterior, para cada tabla se deberá especificar el nombre y las columnas de las que se compondrá, además de indicar el tipo de dato de cada columna; a su vez, se deberá indicar si una columna admitirá valores nulos. Opcionalmente, se deberá especificar a qué tablespace se asociará la tabla a crear junto con la configuración de almacenamiento adecuada (especificada por el departamento de Coordinación de Sistemas de Bases de Datos de la Institución). Posterior a este paso, se deberá agregar la restricción de llave primaria en la columna correspondiente, especificando la configuración adecuada en cuanto a almacenamiento se refiere.

Por razones de practicidad, para la generación de identificadores únicos se tomó la decisión de utilizar un objeto de base de datos que proporciona el RDBMS de Oracle denominado secuencia. Una secuencia proporciona a petición del usuario números consecutivos según los criterios establecidos en su creación. El script para la generación de la secuencia a utilizar se muestra en la figura 4.2.1.2.



distintas. Por esta razón, la organización de los distintos componentes del sistema será visualizada de manera lógica, aunque físicamente éstos residan en locaciones distintas. La figura 4.3.1 presenta la arquitectura del SADEC.



**Figura 4.3.1.** Arquitectura del SADEC

Como se describió en el capítulo 3 (sección 3.2.1.1), la arquitectura del sistema se basará en el patrón arquitectural por capas orientado a aplicaciones empresariales, por lo que dicha arquitectura será conformada por tres capas (capa de fuente de datos, capa de lógica de dominio y capa de presentación), las cuales se describen a continuación.

### 4.3.1 Capa de Fuente de Datos

En el contexto de patrones arquitecturales por capas orientados a aplicaciones empresariales, la capa de fuente de datos engloba todo lo relacionado a la comunicación con otros sistemas

que llevan a cabo tareas para la aplicación, los cuales, para la mayoría de aplicaciones empresariales se tratan de sistemas administradores de bases de datos responsables de almacenar información de manera persistente [9].

Por lo anterior, la capa de fuente de datos del SADEC se compondrá de las herramientas y marcos de trabajo proporcionados por el sistema Administrador de bases de datos relacionales utilizado (Oracle RDBMS) junto con la base de datos del sistema.

En la figura 4.3.1 se observa que la capa de fuente de datos del sistema fue dividida en dos subcapas, siendo una de éstas la sub-capas de acceso a datos. Cabe destacar que esta separación es realizada de manera implícita por el sistema administrador de bases de datos al proporcionar herramientas y marcos de trabajo necesarios para el acceso a datos almacenados.

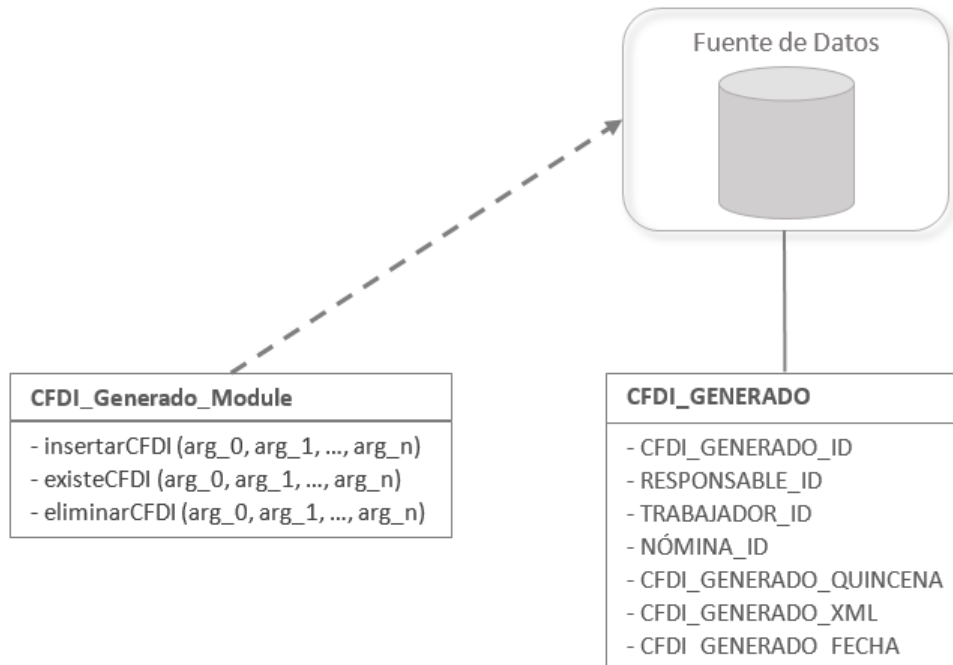
En la infraestructura tecnológica de la BUAP existen distintos servidores de bases de datos, en uno de los cuales residirá la capa de fuente de datos del sistema. Es por esto que esta capa de la arquitectura se encontrará físicamente separada de las capas de presentación y lógica de dominio.

### **4.3.2 Capa de Lógica de Dominio**

En esta capa es en donde reside la lógica de dominio o lógica de negocio. Se trata del trabajo que la aplicación necesita realizar para el dominio en el que se ejecuta. Dentro de esta lógica se involucran cálculos basados en entradas al sistema y datos almacenados, validaciones de cualquier tipo de información proveniente de la capa de presentación, y la resolución de la fuente de datos a utilizar dependiendo de los comandos recibidos desde la capa de presentación [9].

Según los requisitos del sistema, éste deberá ser implantado sobre plataforma Web, por lo que se utilizará la arquitectura cliente-servidor como modelo de aplicación, por lo que la lógica de dominio será implantada haciendo uso de las tecnologías Microsoft Internet Information Services (IIS) como servidor web y Microsoft Active Server Pages (ASP) para la generación dinámica de páginas web.

Es posible implantar distintos patrones arquitecturales para la organización de la capa de lógica de dominio. En el caso del SADEC se optó por utilizar el patrón arquitectural Table Module, cuya aplicación consiste en manejar mediante una sola instancia de clase la lógica de negocio para todos los datos de una tabla o vista de una base de datos. Este patrón arquitectural permite organizar la lógica de dominio con una clase por tabla en la base de datos, por lo que una sola clase puede contener los métodos necesarios para actuar sobre la información [9]. En la figura 4.3.2.1 se muestra un ejemplo del funcionamiento de este patrón para la tabla de base de datos CFDI\_GENERADO analizada en capítulos anteriores.



**Figura 4.3.2.1.** Ejemplo de patrón arquitectural para lógica de dominio Table Module.

Como se puede ver en el ejemplo de la figura anterior, la base de datos cuenta con una tabla llamada `CFDI_GENERADO`. Según el patrón arquitectural Table Module se deberá construir una clase en la cual se definan los métodos que llevarán a cabo operaciones sobre la tabla correspondiente. La figura 4.3.2.2 muestra la estructura de esta clase. Se puede observar que los métodos de esta clase son declarados como métodos estáticos, por lo que no es necesario crear una instancia de clase (objeto) para su invocación. La decisión de implantar esta característica fue tomada con fines de optimizar el rendimiento del sistema y para proporcionar practicidad en su programación.

```
1 Public Class CFDI_Generado_Module {
2
3     Public Shared Function insertarCFDI(arg0, arg1, ... , argn) As tipo_retorno
4
5         ' Código de inserción de CFDI en base de datos
6
7     End Function
8
9
10    Public Shared Function eliminarCFDI(arg0, arg1, ... , argn) As tipo_retorno
11
12        ' Código de eliminación de CFDI de base de datos
13
14    End Function
15
16
17    Public Shared Function obtenerCFDI(arg0, arg1, ... , argn) As tipo_retorno
18
19        ' Código de extracción y retorno de CFDI de base de datos
20
21    End Function
22
23 }
```

**Figura 4.3.2.2.** Estructura de clase CFDI\_Generado\_Module

### 4.3.3 Capa de Presentación

La capa de presentación es la encargada de manejar la interacción entre el usuario y el software. Las principales responsabilidades de esta capa son mostrar información al usuario y traducir comandos del mismo en acciones para las capas de lógica de dominio y fuente de datos [9].

Debido a que el SADEC deberá ejecutarse sobre plataforma Web, la interacción con el usuario (cliente) será implantada mediante el uso de un navegador desplegando páginas dinámicas generadas en el servidor Web. En la figura 4.3.3.1 se observa un ejemplo de programación de la página de acceso al sistema (código residente en el servidor), mientras que en la figura 4.3.3.2 se muestra la misma página renderizada por un navegador web (visualizada por el cliente).



```

1 <%@ Page Language="vb" AutoEventWireup="false" CodeBehind="sadedc_access_page.aspx.vb" %>
2
3 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4
5 <html xmlns="http://www.w3.org/1999/xhtml" >
6 <head runat="server">
7 <title>SADEC :: Acceso</title>
8 <link rel="stylesheet" href="css/login.css" />
9 <link rel="shortcut icon" href="-\img\iconn.ico" />
10 </head>
11 <body>
12 <form id="frm_login" runat="server">
13 <div id="div_wrapper">
14 <div id="div_logged"></div>
15 <div id="div_content">
16 <div id="div_module_desc">&nbsp;&nbsp;&nbsp;</div>
17 <div id="div_system">
18 <div id="div_login_wrapper">
19 <div id="div_welcome">Bienvenid@</div>
20 <div id="div_welcome_msg">
21 Para ingresar al sistema favor de proporcionar sus datos de acceso:
22 </div>
23 <div id="div_welcome_data">
24 <table>
25 <tr>
26 <td>Usuario:</td>
27 <td>
28 <asp:TextBox ID="sadedc_nombre_usuario" runat="server" MaxLength="10"></asp:TextBox>
29 </td>
30 </tr>
31 <tr>
32 <td>Contraseña:</td>
33 <td>
34 <asp:TextBox ID="sadedc_password" runat="server" TextMode="Password" MaxLength="10"></asp:TextBox>
35 </td>
36 </tr>
37 </table>
38 <asp:Label ID="sadedc_message" runat="server"></asp:Label>
39 <asp:Button ID="sadedc_button" runat="server" Text="Ingresar" />
40 </div>
41 </div>
42 </div>
43 <div id="div_footer">
44 <div id="div_siu">
45 Sistema de Administración Electrónica de CFDI (SADEC)<br />
46 Benemérita Universidad Autónoma de Puebla<br />
47 Contacto: (222) 2 29 55 00<br />
48 SIU<br />
49 </div>
50 </div>
51 </div>
52 </div>
53 </form>
54 </body>
55 </html>

```

Figura 4.3.3.1. Programación de ASP para acceso al sistema.



Figura 4.3.3.2. Página de acceso al sistema renderizada por un navegador web.

## 4.4 Revisión y Retrospectiva de Iteración 0

De acuerdo a las tareas involucradas dentro del contexto del marco de trabajo de Scrum, al término de la ejecución de una iteración se obtiene una versión funcional del producto en construcción. Por esto, una vez concluida la ejecución de una iteración es necesario llevar a cabo una revisión y retrospectiva del trabajo realizado.

En la iteración cero de la cartera de producto del SADEC se especifican las tareas de diseño e implantación de la base de datos y arquitectura del sistema. Una vez realizadas estas tareas se considera concluido el sprint en ejecución. Es en este punto donde según la metodología de Scrum, es necesario realizar una revisión (sprint review) y una retrospectiva (scrum retrospective) de lo realizado.

Durante la revisión a la ejecución de la iteración cero, se determinó que la implantación de la base de datos así como de la arquitectura del sistema se logró de manera satisfactoria dentro del tiempo estimado, por lo que no se consideraron cambios a las tareas o tiempos de iteraciones posteriores.

## 4.5 Ejecución de Iteración 1

Como se describe en el backlog de producto del SADEC, la iteración número uno consta de seis tareas a realizar organizadas de manera prioritaria, las cuales llevarán a cabo los procesos principales del sistema (generación, sellado y timbrado de comprobantes fiscales). En esta sección se abarca el desarrollo de estas tareas y se concluye con la revisión y retrospectiva de la ejecución de la iteración.

### 4.5.1 Módulo de Generación de CFDI

El proceso de generación de un comprobante fiscal digital consiste en construir un documento en formato electrónico XML con la información necesaria y la estructura establecida según los estándares definidos por el Servicio de Administración Tributaria (SAT).

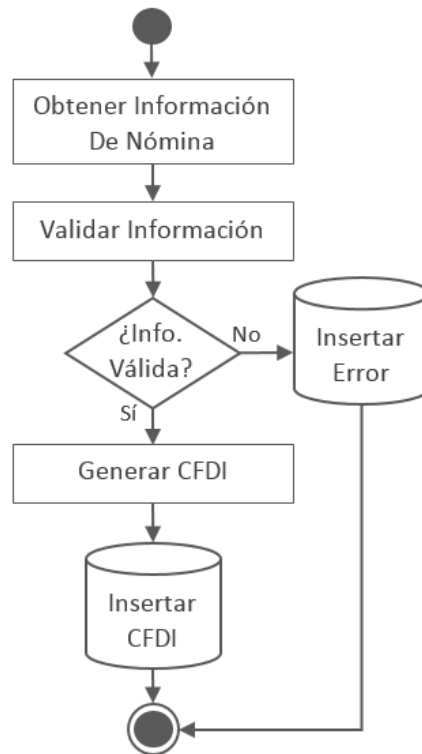
El módulo de generación de CFDI del SADEC es uno de los más importantes. Este módulo será el encargado de recolectar y procesar la información de Nómina Institucional mínimamente requerida para generar un documento en formato electrónico XML de acuerdo a los lineamientos establecidos por el SAT.

Una de las funciones del Departamento de Nóminas e Información de la Dirección de Recursos Humanos de la Benemérita Universidad Autónoma de Puebla es la gestión de la información de nómina institucional; es por esto que este departamento será el encargado de proporcionar la información de nómina necesaria al módulo de generación de comprobantes fiscales digitales del SADEC. Por razones de rendimiento y consistencia en los datos, se determinó proporcionar esta información en forma de vistas sobre tablas específicas de las bases de datos de nómina.

La información de nómina proporcionada por el Departamento de Nóminas e Información de la BUAP será procesada para su almacenamiento en una base de datos propia del SADEC, la cual será procesada para obtener un documento en formato electrónico XML. Este procesamiento se llevará a cabo mediante un procedimiento almacenado (stored procedure) residente en la base de datos del sistema. La figura 4.5.1.1 muestra el diagrama de flujo de este procedimiento.

En los requisitos del SADEC se especifica que la generación de comprobantes podrá llevarse a cabo de manera individual o masiva, por esto, el procedimiento almacenado encargado de generar CFDI tendrá como parámetros de entrada el número de quincena para la cual se generarán los documentos, el año al cual pertenece dicha quincena y, opcionalmente, el ID del trabajador para quien se genera el CFDI y/o una de las nóminas institucionales. La obtención de información de nómina se realizará mediante consultas a las vistas proporcionadas por el Departamento de Nóminas e Información en base a los parámetros de entrada del procedimiento almacenado. La validación de esta información se realizará de acuerdo a las reglas establecidas en el Anexo 20 de la Resolución Miscelánea Fiscal para 2014, así como del estándar de complemento de nómina, ambos documentos proporcionados por el SAT. En caso de que la información no cumpla con alguno de los requisitos establecidos en los documentos antes mencionados, se establecerá el estado del comprobante generado como Error de generación y se omitirá la inserción del documento XML en la base de datos; en caso contrario, el CFDI se generará estableciendo su estado como de generación correcta y se insertará el documento en formato electrónico XML generado en la base de datos. En ambos casos se regresará un mensaje en un parámetro de salida indicando las acciones realizadas.

Por cada trabajador:



**Figura 4.5.1.1.** Diagrama de Flujo de procedimiento almacenado para la generación de CFDI

En la arquitectura del sistema, el procedimiento almacenado de generación de comprobantes residirá en la capa de lógica de dominio, pues, se encarga de realizar operaciones con la información extraída de la capa de acceso a datos una vez interpretadas las entradas provenientes de la capa de presentación.

#### 4.5.2 Pruebas de Módulo de Generación de CFDI

La realización de pruebas del módulo de generación de CFDI corresponde a la quinta tarea de la iteración número uno del backlog de producto. Estas pruebas se realizaron de manera gradual, es decir, descomponiendo el procedimiento almacenado para realizar pruebas individuales a cada tarea que éste realice.

Para comprobar la correcta extracción de información de las vistas de la base de datos de la Nómina Institucional se realizaron comparaciones automatizadas entre esta información y la almacenada por el procedimiento.

En un principio se contempló realizar pruebas de validación a los resultados del proceso de generación de documentos en formato electrónico XML haciendo uso de una herramienta en línea proporcionada por el SAT, sin embargo, dadas las dificultades presentadas en el uso de esta herramienta con comprobantes que no cuentan con sello digital del emisor, se optó por delegar estas pruebas al módulo de sellado de CFDI.

### 4.5.3 Módulo de Sellado de CFDI

El proceso de sellado de un comprobante fiscal digital consiste en generar un sello digital para un CFDI utilizando criptografía de clave pública aplicada a una cadena original. El propósito del sello digital es emitir comprobantes fiscales con autenticidad, integridad, verificables y no repudiables por el emisor. Los elementos necesarios en la generación de sellos digitales son:

- Cadena Original del elemento a sellar
- Certificado de sello digital y su correspondiente clave privada
- Algoritmos de criptografía de clave pública
- Especificaciones de conversión a Base 64

Los algoritmos utilizados en la generación de un sello digital son:

- **SHA-1:** Función de digestión de un solo sentido tal que para cualquier entrada produce una salida compleja de 160 bits denominada digestión.
- **RSAPrivateEncrypt:** Utiliza la clave privada del emisor para encriptar la digestión del mensaje.
- **RSAPublicDecrypt:** utiliza la clave pública del emisor para desencriptar la digestión del mensaje.

Se entiende como cadena original, a la secuencia de datos formada con la información contenida dentro del comprobante fiscal digital a través de Internet, siguiendo para ello las siguientes reglas:

- Ninguno de los atributos que conforman al CFDI deberá contener el carácter “pipe” (|) debido a que éste será utilizado como carácter de control en la generación de la cadena original.
- El inicio de la cadena original se deberá indicar con dos caracteres “pipe” (||).
- Se expresará únicamente la información del dato sin expresar el atributo al que hace referencia.
- Cada dato individual se encontrará separado de su dato subsiguiente, en caso de existir, mediante un carácter “pipe” sencillo (|).
- Los espacios en blanco que se presenten dentro de la cadena original serán tratados de la siguiente manera:
  - Se deberán reemplazar todos los tabuladores, retornos de carro y saltos de línea por espacios en blanco.
  - Acto seguido se elimina cualquier carácter en blanco al principio y al final de cada separador “pipe” sencillo (|)
  - Finalmente, toda secuencia de caracteres en blanco intermedias se sustituyen por un único carácter en blanco.
- Los datos opcionales no expresados, no aparecerán en la cadena original y no tendrán delimitador alguno.

- El final de la cadena original será expresado mediante la cadena de caracteres doble “pipe” (||).
- Toda la cadena original deberá expresarse en formato de codificación UTF-8.
- El nodo Timbre Fiscal del SAT será integrado posterior a la validación realizada por un proveedor autorizado por el SAT que forma parte de la Certificación Digital del SAT. Dicho nodo no se integrará a la formación de la cadena original del CFDI, las reglas de conformación de la cadena original del nodo se describen en el rubro II.C del anexo 20 [12].

La secuencia de formación de la cadena original de un CFDI será en el orden que se expone en la sección Secuencia de Formación del rubro II.B del anexo 20. Cabe señalar que la cadena original de un CFDI puede formarse de manera manual o mediante el uso de un transformador XSLT proporcionado por el SAT.

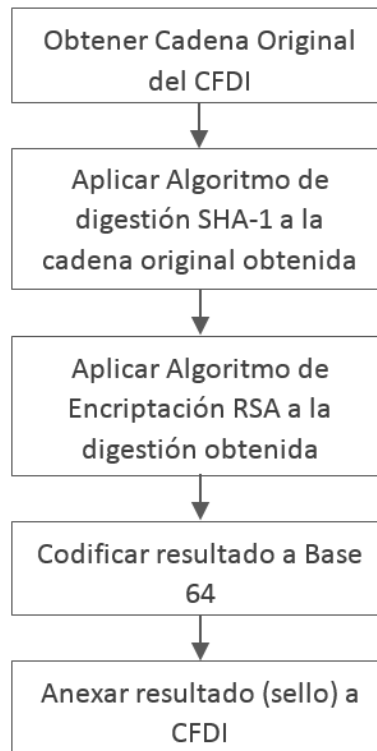
Para toda cadena original a ser sellada digitalmente, la secuencia de algoritmos a aplicar es la siguiente [12]:

- I. Aplicar el método de digestión SHA-1 a la cadena original a sellar incluyendo los nodos Complementarios. Este procedimiento genera una salida de 160 bits (20 bytes) para todo mensaje. La posibilidad de encontrar dos mensajes distintos que produzcan una misma salida es de 1 en  $2^{160}$ , y por lo tanto en esta posibilidad se basa la inalterabilidad del sello, así como su no reutilización. Es de hecho una medida de la integridad del mensaje sellado, pues toda alteración del mismo provocará una digestión totalmente diferente, por lo que no se podrá autenticar el mensaje.
- II. Con la clave privada correspondiente al certificado digital del emisor del mensaje y del sello digital, encriptar la digestión del mensaje obtenida en el paso I utilizando para ello el algoritmo de encriptación RSA.
- III. El resultado será una cadena binaria que no necesariamente consta de caracteres imprimibles, por lo que deberá traducirse a una cadena que sí conste solamente de tales caracteres. Para ello se utilizará el modo de expresión de secuencias de bytes denominado “Base 64”, que consiste en la asociación de cada 6 bits de la secuencia a un elemento de un “alfabeto” que consta de 64 caracteres imprimibles. Puesto que con 6 bits se pueden expresar los números del 0 al 63, si a cada uno de estos valores se le asocia un elemento del alfabeto se garantiza que todo byte de la secuencia original puede ser mapeado a un elemento del alfabeto Base 64, y los dos bits restantes formarán parte del siguiente elemento a mapear. Este mecanismo de expresión de cadenas binarias produce un incremento del 25% en el tamaño de las cadenas imprimibles respecto de la original.

La codificación en base 64, así como su decodificación, se hará tomando los bloques a procesar en el sentido de su lectura, es decir, de izquierda a derecha.

El alfabeto a utilizar se puede consultar en el catálogo de la sección “Generación del Sello Digital” del rubro II.B del anexo 20 [12].

El diagrama de la figura 4.5.3.1 muestra en resumen los pasos involucrados en el proceso de sellado de la cadena original de un CFDI.



**Figura 4.5.3.1.** Proceso de sellado de CFDI

El desarrollo del módulo de sellado de CFDI del SADEC se llevó a cabo utilizando distintas librerías del framework Microsoft .NET, así como de la herramienta OpenSSL.

La generación manual de la cadena original de un CFDI es propensa a errores, por lo que en el desarrollo de este módulo se optó por utilizar el transformador XSLT que proporciona el SAT para este propósito.

En la figura 4.5.3.2 se presenta el código de la función encargada de obtener la cadena original de un documento XML a partir de un transformador XSLT y haciendo uso de las librerías System.Xml del framework Microsoft .NET. Esta función recibe como parámetro el comprobante fiscal como texto en una variable de tipo String. El valor de retorno será un objeto de tipo StringWriter que contendrá la cadena original obtenida.

```

1 Public Shared Function obtenerCadenaOriginal(ByVal xmldoc As String) As StringWriter
2
3     Dim stream As StringReader = New StringReader(xmldoc)
4     Dim xpathdoc As XPathDocument = New XPathDocument(stream)
5     Dim xsltTrans As XslCompiledTransform = New XslCompiledTransform()
6     Dim str As StringWriter = New StringWriter()
7     Dim writer As XmlTextWriter = New XmlTextWriter(str)
8
9     ' Se carga transformador XSLT proporcionado por el SAT
10    xsltTrans.Load("Z:\cadenaoriginal_3_2.xslt")
11
12    ' Se aplica transformación para obtener cadena original
13    xsltTrans.Transform(xpathdoc, Nothing, writer)
14
15    Return str
16
17 End Function

```

**Figura 4.5.3.2.** Función de obtención de cadena original de CFDI

Los siguientes pasos en el proceso de sellado, son aplicar el algoritmo de digestión SHA-1 a la cadena original obtenida, cuya digestión se encriptará utilizando el algoritmo RSA y la llave privada del emisor. Este resultado se codificará en Base 64 y se anexará al comprobante fiscal a sellar. Estos pasos se implantaron en una sola función mediante llamadas a métodos de la librería System.Security.Cryptography del framework Microsoft .NET (Figura 4.5.3.3).

```

1 Public Shared Function obtenerselloDigital(ByVal path_pfx As String &
2     , ByVal pwd_pfx As String &
3     , ByVal cadena_original As String) As String
4
5     Dim cer As X509Certificate2 = New X509Certificate2(path_pfx, pwd_pfx)
6     Dim rsa As RSACryptoServiceProvider = CType(cer.PrivateKey, RSACryptoServiceProvider)
7
8     If rsa Is Nothing Then
9         Throw New Exception("El certificado " & path_pfx & " no es válido.")
10    End If
11
12    rsa.PersistKeyInCsp = False
13
14    Dim data As Byte() = System.Text.Encoding.UTF8.GetBytes(cadena_original)
15
16    Dim hash_name As String
17
18    hash_name = CryptoConfig.MapNameToOID("SHA1")
19
20    ' Aplica algoritmo SHA-1
21    data = hash_shal_byte(data)
22
23    ' Firma hash
24    Dim encriptado As Byte() = rsa.SignHash(data, hash_name)
25
26    rsa.Clear()
27
28    ' Regresa resultado codificado en base64
29    Return Convert.ToBase64String(encriptado)
30
31 End Function

```

**Figura 4.5.3.3.** Función encargada de obtener el sello digital de CFDI a partir de la cadena original.

La aplicación del algoritmo de encriptación RSA requirió de un archivo con extensión *.pfx*, el cual se puede obtener a partir del CSD (Certificado de Sello Digital) del emisor. A continuación se describen los pasos necesarios para la obtención de este archivo.



## Generación de archivo *.pfx* para el sellado de comprobantes

Para la generación de este archivo es necesario contar con el certificado de sello digital y el archivo con extensión *.key* asociado, así como de su contraseña privada. Durante el desarrollo de este módulo, estos elementos fueron proporcionados por la Tesorería General de la BUAP.

Una vez localizados los archivos y clave necesarios, se deberá acceder a la herramienta OpenSSL y convertir el archivo de certificado de sello digital (archivo con extensión *.cer*) a un archivo de certificado con extensión *.pem* con el comando:

```
x509 -inform DER -in certificado.cer -out certificado.pem
```

donde *certificado.cer* es el archivo de certificado de sello digital, y *certificado.pem* será el archivo resultante.

Como siguiente paso se deberá convertir el archivo con extensión *.key* a un archivo con extensión *.pem* mediante el comando:

```
pkcs8 -inform DER -in llave.key -passin pass:cve_privada -out llave.pem
```

donde *llave.key* es el archivo con extensión *.key* asociado al certificado de sello digital, *cve\_privada* es la contraseña privada de este último y *llave.pem* es el nombre del archivo de salida.

Como último paso se deberá obtener el archivo con extensión *.pfx* a partir de los archivos con extensión *.pem* obtenidos anteriormente, lo cual se lleva a cabo mediante el comando:

```
pkcs12 -export -out resultado.pfx -inkey llave.pem -in certificado.pem -passout  
pass:clave_salida
```

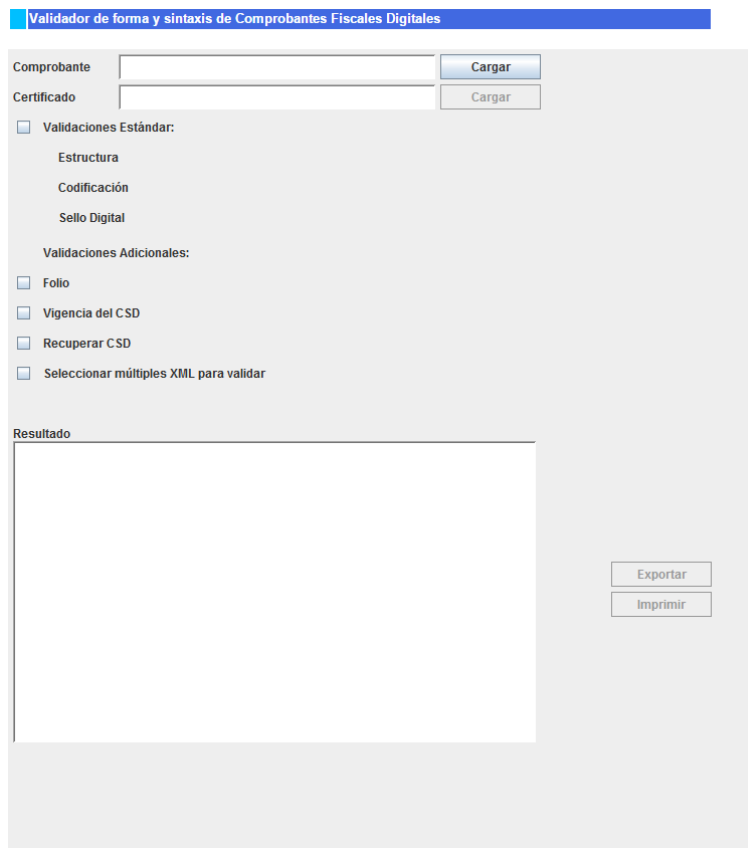
donde *resultado.pfx* es el nombre del archivo resultante, *llave.pem* y *certificado.pem* son los archivos obtenidos en pasos anteriores y *clave\_salida* será la contraseña asociada al archivo resultante con extensión *.pfx*. Esta contraseña será la utilizada para aplicar el archivo de encriptación RSA, así como para futuras cancelaciones de CFDI en caso de ser necesario.

En la figura 4.5.3.3 se muestra la función encargada de aplicar el algoritmo de digestión SHA-1, el algoritmo de encriptación RSA y la codificación en base64 a la cadena original de un comprobante a sellar. Como se puede observar en la figura, esta función recibe como parámetros la ruta del archivo *.pfx* generado a partir del certificado de sello digital y la contraseña especificada durante su creación, así como la cadena original a codificar. El resultado de esta función será el sello digital, el cual deberá ser anexado al comprobante como atributo de nombre sello perteneciente al elemento *cfdi:Comprobante*.

#### 4.5.4 Pruebas del módulo de sellado

Como se mencionó en la sección 4.5.3, las pruebas de revisión de estructura y validación de CFDI se delegaron al módulo de sellado. Estas pruebas se realizaron mediante el uso del transformador XSLT proporcionado por el SAT, el cual impide la obtención de la cadena original de un comprobante fiscal con errores en su estructura.

Las pruebas al módulo de sellado se realizaron validando los comprobantes fiscales sellados a través de la herramienta de validación de CFDI proporcionada por el SAT (Figura 4.5.4.1). Mediante el uso de esta herramienta fue posible identificar la mayoría de los errores presentados en el proceso de sellado de comprobantes, siendo los más frecuentes aquellos relacionados con la generación del sello digital.



**Figura 4.5.4.1.** Herramienta en línea de validación de comprobantes fiscales

([https://www.consulta.sat.gob.mx/sicofi\\_web/moduloECFD\\_plus/ValidadorCFDI/Validador%20cfdi.html](https://www.consulta.sat.gob.mx/sicofi_web/moduloECFD_plus/ValidadorCFDI/Validador%20cfdi.html))

#### 4.5.5 Módulo de Timbrado de CFDI

El proceso de timbrado es la certificación de un comprobante fiscal digital; consiste en la verificación de la información, validación del comprobante y en agregar un sello digital que compruebe y corrobore que dicho comprobante ha sido certificado por el SAT, y por lo tanto cuenta con validez fiscal. El timbrado fiscal únicamente puede ser emitido por un Proveedor

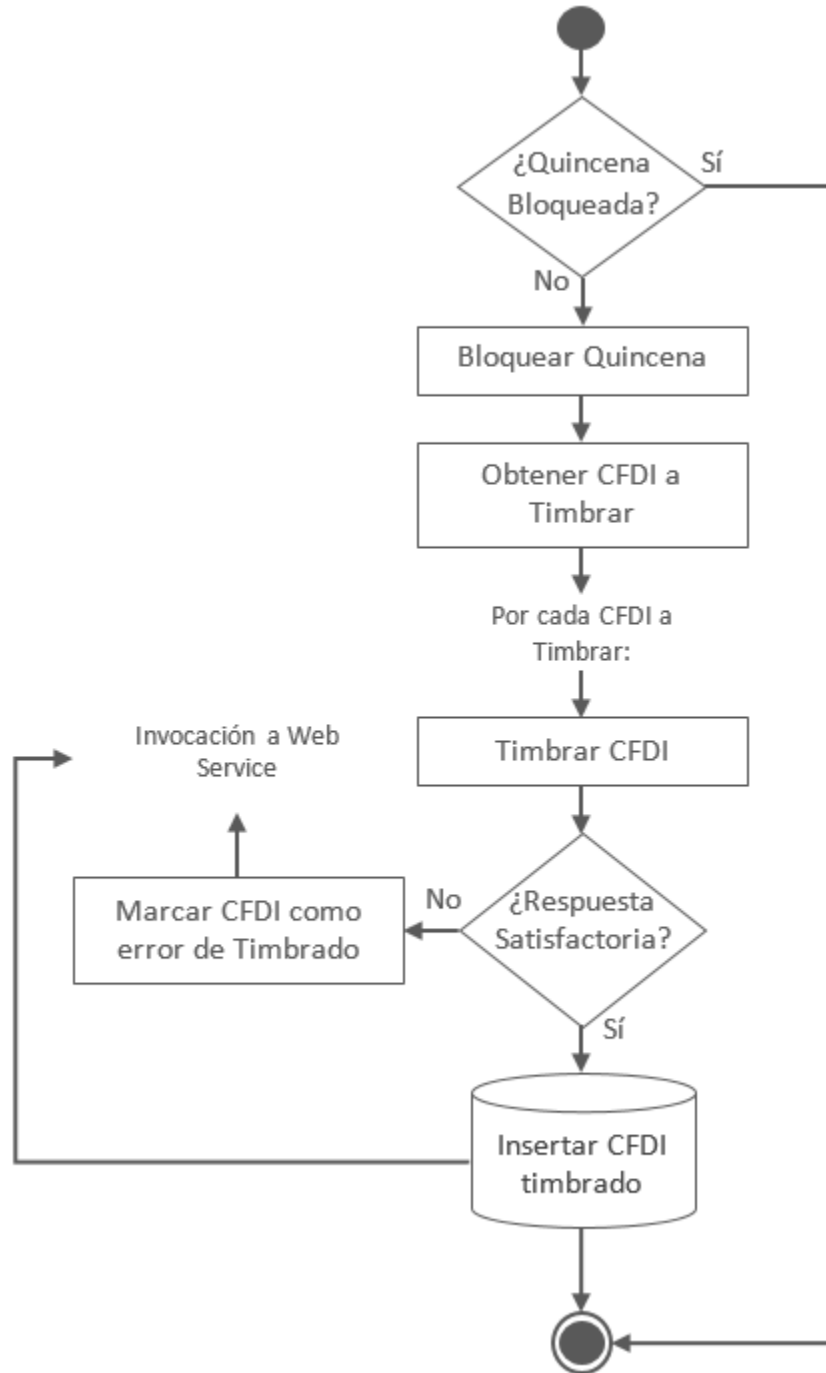
Autorizado de Certificación (PAC), cuya principal función es proporcionar de forma directa al SAT la información de facturas electrónicas emitidas.

Tanto el SAT como los distintos PAC del mercado actual brindan un servicio de timbrado público y gratuito, ya que por ley, todo Proveedor Autorizado de Certificación se encuentra obligado a proporcionar este servicio, sin embargo, se trata de un servicio básico que no proporciona la flexibilidad requerida en procesos que requieren el timbrado de CFDI de manera masiva. Para cubrir esta necesidad en el timbrado de Nómina Institucional, se determinó contratar los servicios de timbrado mediante web services ofrecidos por un PAC, los cuales ofrecen distintas funciones, entre las cuales se encuentran: timbrado, consulta y validación de CFDI de forma individual.

Para cubrir los requisitos especificados en el módulo de timbrado del SADEC se implantó un método que permite timbrar de manera masiva o individual comprobantes fiscales haciendo uso del servicio web proporcionado por el PAC contratado. El método de timbrado de comprobantes fiscales recibirá como parámetros la quincena a timbrar junto con el año al que ésta pertenece, y, opcionalmente, el identificador de un trabajador y/o una de las Nóminas Institucionales. Este método invocará el web service proporcionado por el PAC, llamando la función correspondiente al timbrado y pasando como parámetros las credenciales de acceso al servicio, el comprobante fiscal en formato de texto y, opcionalmente, una referencia única. Como respuesta se obtendrá un objeto mediante el cual será posible determinar si el CFDI se timbró de manera correcta y obtener el comprobante timbrado con validez fiscal. En caso de que el proceso de timbrado del comprobante fiscal no se lleve a cabo de manera satisfactoria, el objeto respuesta contendrá un código de error y un mensaje asociado. En la figura 4.5.5.3 se presentan los códigos y mensajes del objeto respuesta del servicio web de timbrado.

Con el fin de evitar duplicidades en el envío de peticiones de timbrado, se implantó un método de bloqueo para asegurar no someter una quincena al proceso de timbrado en más de una ocasión.

En la figura 4.5.5.1 se muestra el diagrama de flujo correspondiente al método implantado para el timbrado de CFDI mediante la invocación del web service proporcionado por el PAC.



**Figura 4.5.5.1.** Diagrama de flujo de timbrado de CFDI

La figura 4.5.5.2 muestra un ejemplo de la función encargada de la invocación del servicio web de timbrado.

```

1 Public Function timbrar As Boolean
2
3     ' Se crea instancia para invocar el servicio web
4     Dim cliente As WSCFDI = New WSCFDI()
5
6     ' Se invoca el servicio web y se almacena el objeto de respuesta
7     Dim obj_respuesta As RespuestaCFDI = cliente.TimbrarCFDI("USUARIO", "PASSWORD", documento_xml, referencia)
8
9     return obj_respuesta
10
11 End Function
    
```

**Figura 4.5.5.2.** Ejemplo de invocación de servicio web del PAC

Código	Mensaje	Descripción
301	XML mal formado	El XML recibido no cumple con los estándares del SAT
302	Sello mal formado o inválido	El sello que contiene el XML se generó de manera incorrecta
303	Sello no corresponde a emisor o caduco	El XML se selló con un CSD que no corresponde al RFC del emisor
304	Certificado revocado o caduco	El XML se selló con un CSD que se encuentra con un estatus revocado en la LCO o ya caducó su vigencia
305	La fecha de emisión no está dentro de la vigencia del CSD del Emisor	El XML se generó en una fecha fuera del rango de vigencia del CSD según la LCO
306	El certificado no es de tipo CSD	El XML se selló con la FIEL
307	El CFDI contiene un timbre previo	El XML contiene ya contiene el complemento Timbre Fiscal Digital
308	Certificado no expedido por el SAT	El XML se selló con un Certificado no emitido por el SAT
401	Fecha y hora de generación fuera de rango	El XML se generó antes de 72 horas o en una fecha/hora posterior a la actual
402	RFC del emisor no se encuentra en el régimen de contribuyentes	El RFC del emisor no se encuentra en la LCO
201	UUID Cancelado	Cancelación exitosa ante el SAT
202	UUID Previamente Cancelado	El UUID ya está cancelado en los registros del SAT
203	UUID no corresponde al Emisor	El UUID enviado para cancelar no corresponde al RFC del emisor enviado
204	UUID no aplicable para cancelación	El UUID no se registró correctamente ante el SAT (caso poco probable)
205	UUID no existe	El UUID no existe en los registros del SAT
801	El comprobante ya fue timbrado por FEL	XML timbrado previamente por FEL

805	El comprobante contiene el nodo Addenda	El XML contiene este nodo. No se debe timbrar un XML con Addenda, pero se puede agregar a su XML después del timbrado
806	Error genérico de invocación del servicio	Alguno de los datos enviados para acceder al servicio es incorrecto
807	Error de autenticación de usuario	El usuario no existe o está mal la contraseña
808	El usuario no cuenta con permiso de acceso	El usuario existe, pero se le revocó el permiso de acceso por uso indebido del servicio
809	El paquete de timbres ha expirado	El paquete de timbres adquirido ya caducó
811	El RFC de usuario no corresponde al del emisor de CFDI	Se está tratando de timbrar un XML de otro contribuyente
815	Ha alcanzado el límite de intentos de autenticación. Intente después de 30 minutos	Después de 3 intentos fallidos, se bloqueará al usuario por 30 minutos
816	816 No se pudo realizar el envío al SAT	Ocurrió un error al tratar de acceder al servicio del SAT para la entrega del CFDI

**Figura 4.5.5.3.** Códigos y mensajes recibidos en objeto de respuesta del servicio web proporcionado por el PAC

#### 4.5.6 Pruebas de Timbrado de CFDI

El proceso de timbrado de un CFDI dota a éste de validez fiscal ante el SAT, por lo que las pruebas del módulo de timbrado de comprobantes fiscales se realizaron con accesos a un ambiente de pruebas proporcionado por el PAC.

Las pruebas de invocación del servicio web de timbrado se llevaron a cabo en primera instancia de manera individual, obteniendo en la mayoría de los casos un resultado satisfactorio. Posteriormente se realizaron pruebas de manera masiva. En estas pruebas se obtuvieron errores con código 301, 302, 401, 801, 806, 808 y 809. A continuación se expone el análisis de estos errores y la solución planteada.

##### Error 301. XML mal formado

Este error se presentó en documentos cuya estructura y/o información contenida no se validó de manera correcta. Los comprobantes fiscales (documentos con formato electrónico XML) se validan durante la obtención de la cadena original, mediante el transformador XSLT que proporciona el SAT. Fue necesario realizar un análisis detallado de aquellos documentos que presentaron errores durante las pruebas de timbrado, encontrando que durante el proceso de sellado, al añadir el sello digital al documento se insertaban caracteres especiales no imprimibles. Para solucionar este problema se implantó una función de limpieza de texto, la

cual recibe como parámetro un documento XML en una variable de tipo *String* y se regresa este mismo omitiendo caracteres especiales no imprimibles.

Error 302. Sello mal formado o inválido

Este error se presentó con gran frecuencia durante las pruebas realizadas al módulo de timbrado. Debido a que se trata de un problema directamente relacionado con la generación del sello digital, se determinó que el problema residía en el módulo de sellado, causado por una diferencia en los conjuntos de caracteres utilizados entre el servidor de bases de datos y el ambiente de desarrollo, lo que ocasionaba la inserción de caracteres especiales.

Error 401. Fecha y hora de generación fuera de rango

El error con código 401 se presenta al intentar timbrar un comprobante fiscal después de 72 horas a partir de su creación. La única manera de solventar este error es repitiendo los procesos de generación y sellado del mismo.

Error 806. Error genérico de invocación de servicio

Este error se presentó de manera esporádica y sin motivo aparente. El área de soporte técnico del PAC determinó que se trata de errores ocasionales presentes en la comunicación con el servicio web de timbrado. Para solventar este error bastó con repetir el proceso de timbrado para el comprobante fiscal.

### **4.5.7 Revisión y Retrospectiva de Iteración 1**

La iteración número uno abarcó las tareas de desarrollo de los módulos principales del SADEC. El desarrollo de los módulos de generación y sellado junto con sus pruebas se realizaron satisfactoriamente dentro de los tiempos estimados, sin embargo, la fase de prueba del módulo de timbrado demoró más tiempo del considerado, debido a los errores no presentados durante las fases de prueba de los módulos de generación y sellado, por lo que fue necesario realizar revisiones exhaustivas de estos módulos.

## **4.6 Ejecución de Iteración 2**

La segunda iteración de la cartera de producto del SADEC se compone por tareas relacionadas con la implantación de interfaces gráficas del sistema. Esta sección aborda el desarrollo de estas tareas y sus respectivas pruebas.

### **4.6.1 Implantación de Filtros para Visualización de CFDI**

Según los requisitos del SADEC obtenidos a partir de las historias de usuario obtenidas en la fase de análisis, es necesario contar con un filtro que permita la visualización de CFDI por quincena, año, y, opcionalmente, por identificador de trabajador, tipo de nómina y estado del comprobante.

Como se puede observar en el backlog de producto (Capítulo 2, sección 2.4.2), las tareas que conciernen a la visualización de CFDI de la iteración número dos son:

- Implantación de Filtros para Visualización de CFDI
- Visualización y Descarga de CFDI Generados para Administradores
- Visualización y Descarga de CFDI Sellados para Administradores
- Visualización y Descarga de CFDI Timbrados para Administradores

En un principio se consideró tratar estas tareas como secciones separadas, sin embargo, por razones de usabilidad se tomó la decisión de unificarlas en una sola sección. Es por esto que las cuatro tareas listadas anteriormente se unificaron dando como resultado el desarrollo de una tarea que consiste en la visualización y descarga de CFDI independientemente de su estado, proporcionando un filtro que contemple todas las posibles opciones de visualización. Además, se optó por implantar las acciones a realizar sobre un comprobante fiscal (generar, sellar, timbrar).

La visualización de comprobantes fiscales digitales se realizará mediante un filtro, en el cual se especifica la quincena y el año al que pertenece. Opcionalmente se podrá especificar el identificador de un trabajador, la nómina y el estado del CFDI, pudiendo especificar cualquier combinación de estas opciones. La figura 4.6.1.1 muestra la interfaz gráfica de este filtro.

La imagen muestra una interfaz de usuario para un filtro de búsqueda. Está organizada en una lista vertical de campos de entrada:

- ID:** Un campo de texto vacío.
- Quincena:** Un menú desplegable con el valor '1' seleccionado.
- Nómina:** Un menú desplegable con el valor 'Todos' seleccionado.
- Año:** Un menú desplegable con el valor '2015' seleccionado.
- Estado del documento:** Un menú desplegable con el valor 'Todos' seleccionado.

Debajo de estos campos hay un botón rectangular con el texto 'Filtro'.

**Figura 4.6.1.1.** Filtro para la visualización y descarga de CFDI.

Como se puede observar en la figura, este filtro se compone de cinco elementos: el elemento ID corresponde al identificador único de un trabajador. El elemento Quincena indica la quincena de la cual se desea visualizar los CFDI, seleccionándola en la lista desplegable correspondiente que se compone de las 24 quincenas que conforman un año. El elemento Nómina se refiere al tipo de nómina al que pertenecen los CFDI. El elemento año indica el año del cual se desea elegir la quincena y es una lista desplegable con los años para los cuales se genera información de CFDI (de 2014 en adelante). Finalmente, el elemento Estado del documento hace referencia al estado en el que deberán estar los CFDI que se requiere visualizar (no generado, generado, sellado, timbrado, error de generación, error de sellado, error de timbrado).



Una vez establecidos los criterios de filtrado, al presionar el botón Filtro se ejecutará un método que recuperará los CFDI filtrados y se desplegarán con la opción de visualizar/descargar cada comprobante.

Se proporcionará un menú de opciones, el cual, dependiendo del estado de los CFDI filtrados permitirá realizar ciertas operaciones sobre éstos. A continuación se describen las opciones presentes en el menú:

- **Generar:** Esta opción generará los CFDI a partir de la información de nómina residente en las bases de datos de la Institución. Estará disponible siempre que en el filtro se especifique el estado del documento como No generado.
- **Sellar:** Esta opción permitirá ejecutar el proceso de sellado sobre los CFDI filtrados. Esta opción estará disponible únicamente para CFDI cuyo estado sea Generado.
- **Timbrar:** Permitirá el timbrado de los CFDI filtrados. Estará disponible para CFDI con estado Sellado.
- **Eliminar:** Esta opción permitirá eliminar los CFDI filtrados. Estará disponible para aquellos CFDI con estado Generado, Sellado, Error de generación, Error de sellado o Error de timbrado (un CFDI con estado Timbrado cuenta con validez fiscal ante el SAT, por lo que no se permitirá su eliminación).
- **Regresar:** Permite regresar a un estado anterior los CFDI filtrados. Estará disponible para los CFDI con estado Generado, Sellado, Error de generación, Error de sellado o Error de timbrado. Un CFDI timbrado no podrá regresar a su estado Sellado debido a su validez fiscal ante el SAT.

#### 4.6.2 Pruebas de Visualización

Las pruebas realizadas a las distintas secciones de visualización se llevaron a cabo listando y manipulando un conjunto de CFDI desde su generación hasta su timbrado.

Como primera prueba se consultaron comprobantes sin generar pertenecientes a la primera quincena del año 2014. El resultado se puede observar en la figura 4.6.2.1.

The screenshot shows a web interface for filtering CFDI records. At the top, there is a 'Filtrar por:' section with several dropdown menus: 'ID:' (empty), 'Quincena:' (set to '1'), 'Nómina:' (set to 'Todos'), 'Año:' (set to '2014'), and 'Estado del documento:' (set to 'NO GENERADO'). A 'Filtro' button is located below these menus. Below the filter section is a 'MENÚ DE OPCIONES' with a camera icon and a list of actions: 'Generar', 'Sellar', 'Timbrar', 'Cancelar', 'Eliminar', and 'Regresar'. Below the menu, the current filter settings are displayed: 'ID: Todos', 'Quincena: 1', 'Nómina: Todos', 'Año: 2014', 'Estado: NO GENERADO', and 'Registros Totales: 0'. At the bottom, a table displays the filtered records.

CVE. DOC.	EDO. DOC.	ID TRAB.	NÓMINA	QUINCENA	AÑO	GENERADO	SELLADO	TIMBRADO	DET. TIM.	CANCELADO	PDF
1	NO GENERADO	██████████	██████████	1	2014	---	---	---	--	---	--
2	NO GENERADO	██████████	██████████	1	2014	---	---	---	--	---	--
3	NO GENERADO	██████████	██████████	1	2014	---	---	---	--	---	--
4	NO GENERADO	██████████	██████████	1	2014	---	---	---	--	---	--

**Figura 4.6.2.1.** Visualización de CFDI sin generar.

Como se puede observar, en el menú de opciones únicamente se muestra la opción Generar. Los registros visualizados se obtienen a partir de una cláusula de combinación JOIN interno entre tablas de la base de datos del Departamento de Nóminas e Información y la tabla de la base de datos del SADEC que almacena los comprobantes fiscales digitales. Cabe mencionar que a petición de los usuarios administradores finales se añadió un apartado a la interfaz gráfica en donde se muestran los parámetros de filtrado, así como el número de CFDI listados.

Una vez visualizados los comprobantes a generar y presionando el botón Generar del menú de opciones, se invocará al módulo de generación de CFDI. Como se puede ver en la figura 4.6.2.2, esta acción desplegará los comprobantes fiscales generados correctamente. En la sección informativa se mostrará el número total de comprobantes fiscales generados correctamente y con errores. Para visualizar los errores de generación de CFDI se deberán filtrar los comprobantes con estado Error de Generación en el filtro (Figura 4.6.2.3).

MENÚ DE OPCIONES

Generar Sellar Timbrar Cancelar Eliminar Regresar

ID: Todos Quincena: 1 Nómina: ES - BONO ESPECIAL Año: 2014 Estado: GENERADO Registros Totales: 4 Exportar a Excel

CVE. DOC.	EDO. DOC.	ID TRAB.	NÓMINA	QUINCENA	AÑO	GENERADO	SELLADO	TIMBRADO	DET. TIM.	CANCELADO	PDF
1	643810	GENERADO		1	2014	✓	---	---	..	---	--
2	643811	GENERADO		1	2014	✓	---	---	..	---	--
3	643812	GENERADO		1	2014	✓	---	---	..	---	--
4	643813	GENERADO		1	2014	✓	---	---	..	---	--

**Figura 4.6.2.2.** CFDI generados

MENÚ DE OPCIONES

Sellar Timbrar Cancelar Eliminar Regresar

ID: Todos Quincena: 1 Nómina: Todos Año: 2015 Estado: ERROR DE GENERACIÓN Registros Totales: 47

CVE. DOC.	EDO. DOC.	ID TRAB.	NÓMINA	QUINCENA	AÑO	GENERADO	SELLADO	TIMBRADO	DET. TIM.	CANCELADO	PDF
1	650313	ERROR DE GENERACIÓN	100282755	QU	1	2015	!	---	---	..	---
2	650836	ERROR DE GENERACIÓN	100320855	QU	1	2015	!	---	---	..	---
3	651151	ERROR DE GENERACIÓN	100006824	QU	1	2015	!	---	---	..	---
4	652536	ERROR DE GENERACIÓN	100427733	QU	1	2015	!	---	---	..	---

**Figura 4.6.2.3.** Ejemplo de error en generación de CFDI

Como se observa en la figura 4.6.2.3, en caso de existir errores en el proceso de generación, se indicará con un ícono, que al ser presionado se desplegarán los mensajes asociados a los errores generados. En este caso sólo se muestra la opción Eliminar en el menú, pues, la corrección de un CFDI generado implica la modificación de la información que sirve de fuente para su creación.

A continuación se muestra la información del CFDI con clave **650313**, además de los errores registrados del mismo.

Cve. Doc.: 650313      Fecha. Doc.: 23:08      Estado: ERROR DE GENERACIÓN      ID:      Quincena: 1      Nómina:      Año: 2015      Errores Totales: 1

ERRORES											
Error de Generación: 301 XML mal formado The element "Deducciones" in namespace "http://www.sat.gob.mx/nomina" has incomplete content. List of possible elements expected: "Deduccion" in namespace "http://www.sat.gob.mx/nomina".											

**Figura 4.6.2.4.** Despliegue de errores de generación

Al presionar sobre el ícono que indica un error en la generación de CFDI, se mostrará la lista de errores asociados a dicho CFDI. La figura 4.6.2.4 muestra un ejemplo de esta situación.

Para realizar las pruebas de visualización de sellado de CFDI, se deberán visualizar los CFDI a sellar, es decir, los comprobantes fiscales cuyo estado es Generado, y a continuación presionar el botón de sellado en el menú de opciones. Esto invocará al módulo de sellado descrito en la sección 4.5.3. Al igual que en la generación de CFDI, se mostrará un listado con los CFDI sellados correcta y erróneamente, mostrando un ícono según corresponda. En la figura 4.6.2.5 se muestra un ejemplo donde se despliegan los CFDI sellados correctamente, mientras que en la figura 4.6.2.6 se presenta un CFDI para el cual ocurrió un error en el proceso de sellado.

MENÚ DE OPCIONES

Generar Sellar Timbrar Cancelar Eliminar Regresar

ID: Todos      Quincena: 1      Nómina:      Año: 2015      Estado: SELLADO      Registros Totales: 2

CVE. DOC.	EDO. DOC.	ID TRAB.	NÓMINA	QUINCENA	AÑO	GENERADO	SELLADO	TIMBRADO	DET. TIM.	CANCELADO	PDF
1	655710	SELLADO		1	2015	✓	✓	---	--	---	--
2	655709	SELLADO		1	2015	✓	✓	---	--	---	--

**Figura 4.6.2.5.** Visualización de CFDI sellados

Como se observa en la figura anterior, para CFDI sellados correctamente se muestran las opciones de Timbrar y Eliminar en el menú de opciones, las cuales se encargarán del timbrado o eliminación de los CFDI visualizados respectivamente.

MENÚ DE OPCIONES

Generar Sellar Timbrar Cancelar Eliminar Regresar

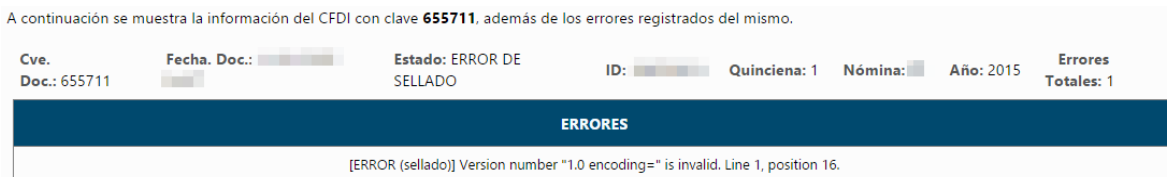
ID: Todos      Quincena: 1      Nómina:      Año: 2015      Estado: ERROR DE SELLADO      Registros Totales: 1

CVE. DOC.	EDO. DOC.	ID TRAB.	NÓMINA	QUINCENA	AÑO	GENERADO	SELLADO	TIMBRADO	DET. TIM.	CANCELADO	PDF
1	655711	ERROR DE SELLADO		1	2015	✓	!	---	--	---	--

**Figura 4.6.2.6.** Visualización de CFDI con error de sellado

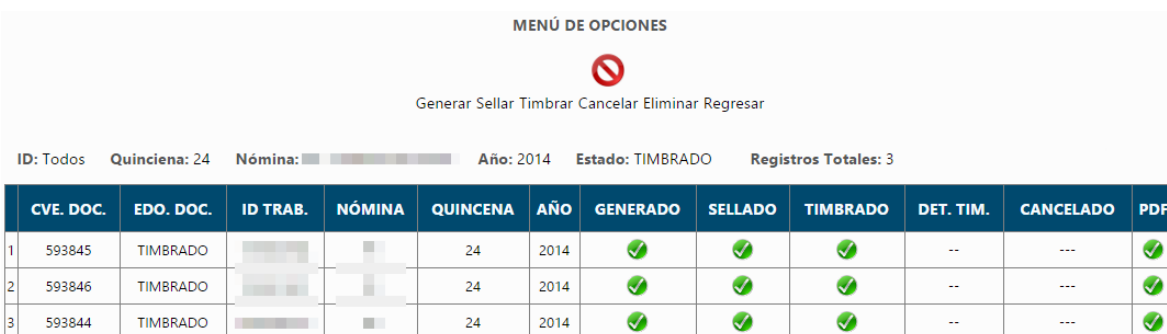
Para visualizar el error que se produjo durante la invocación al módulo de sellado se deberá presionar sobre el ícono de error del CFDI erróneo. La figura 4.6.2.7 muestra un ejemplo de esta operación. En el caso de los CFDI con error de sellado el menú de opciones mostrará

dos posibles operaciones (eliminar y regresar). La operación Eliminar eliminará los CFDI visualizados, mientras que la opción Regresar regresará el CFDI al estado Generado. Esta última operación se implantó debido a que se presentaron pruebas en las que ocurría un error con la carga en memoria del archivo transformador XSLT, lo que ocasionaba errores de sellado, para lo cual bastaba con repetir la operación para un sellado exitoso. En la figura 4.6.2.7 se presenta un ejemplo de despliegue de errores de sellado para un comprobante fiscal.



**Figura 4.6.2.7.** Despliegue de errores de sellado de CFDI



Con respecto a las pruebas de visualización de timbrado de CFDI, se deberán filtrar los CFDI a timbrar, es decir, aquellos comprobantes cuyo estado sea Sellado, y a continuación presionar sobre el botón Timbrar del menú de opciones, lo que invocará al módulo de timbrado de CFDI. Análogo a la operación de sellado, al finalizar la ejecución de este proceso se mostrarán los comprobantes correctamente timbrados, y se indicará el número de errores ocurridos. La figura 4.6.2.8 muestra un listado de CFDI timbrados correctamente, en tanto que la figura 4.6.2.9 muestra los comprobantes con los que se obtuvieron errores en el proceso de timbrado.



**Figura 4.6.2.8.** Listado de CFDI correctamente timbrados




Como se observa en la figura anterior, para descargar el archivo en formato XML de comprobantes timbrados se deberá presionar sobre el ícono que indica un correcto proceso de timbrado. Si se desea visualizar la representación impresa de éste CFDI se deberá presionar sobre el ícono que corresponde a la columna PDF, cuya implantación se dejó pendiente hasta definir el formato de este documento. En el menú de opciones se muestra la opción de cancelar a CFDI timbrados, la cual, por cuestiones externas a la institución su implantación se encuentra pendiente.

MENÚ DE OPCIONES

Generar Sellar Timbrar Cancelar Eliminar Regresar

ID: [redacted]    Quincena: 1    Nómina: [redacted]    Año: 2015    Estado: ERROR DE TIMBRADO    Registros Totales: 1

CVE. DOC.	EDO. DOC.	ID TRAB.	NÓMINA	QUINCENA	AÑO	GENERADO	SELLADO	TIMBRADO	DET. TIM.	CANCELADO	PDF	
1	655710	ERROR DE TIMBRADO	[redacted]	[redacted]	1	2015				--	---	--

**Figura 4.6.2.9.** Listado de CFDI con errores de timbrado

La figura anterior muestra el listado de CFDI con error de timbrado. Para desplegar los errores generados se deberá presionar sobre el ícono que indica este error. Un ejemplo de esta operación se muestra en la figura 4.6.2.10, en donde se despliegan los errores producidos durante la operación de timbrado del comprobante fiscal. En este tipo de errores se muestra el código de error regresado por el servicio web del PAC invocado desde el módulo de timbrado, así como el mensaje de error asociado.

A continuación se muestra la información del CFDI con clave **655710**, además de los errores registrados del mismo.

Cve. Doc.: 655710    Fecha. Doc.: [redacted]    Estado: ERROR DE TIMBRADO    ID: [redacted]    Quincena: 1    Nómina: [redacted]    Año: 2015    Errores Totales: 1

ERRORES	
Error de timbrado: COD:301 MSG:XML mal formado MSGDETAIL:The element "Percepciones" in namespace "http://www.sat.gob.mx/nomina" has incomplete content. List of possible elements expected: "Percepcion" in namespace "http://www.sat.gob.mx/nomina".	

**Figura 4.6.2.10.** Despliegue de errores de timbrado de CFDI

### 4.6.3 Revisión y Retrospectiva de Iteración 2

La iteración número dos correspondió al desarrollo y pruebas de la visualización de los distintos procesos que involucra el timbrado de comprobantes fiscales. Durante la revisión de esta iteración se encontró que el hecho de conjuntar estos procesos representó un aumento no considerable en la complejidad de la implantación, sin embargo, se disminuyó el tiempo de desarrollo y pruebas por parte del usuario, además, se determinó que la interfaz gráfica implantada es amigable e intuitiva según comentarios de los usuarios del sistema.

## 4.7 Ejecución de Iteración 3

Las tareas involucradas en la tercera iteración de la cartera de producto del SADEC se relacionan con el acceso y administración de los usuarios que se encargarán de realizar las operaciones de generación, sellado y timbrado de comprobantes fiscales digitales.

### 4.7.1 Autenticación de Usuarios Administrativos

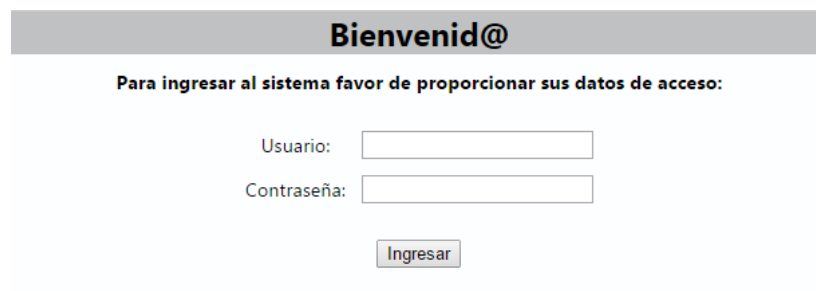
Con el fin de proveer seguridad de acceso a los procesos de generación, sellado y timbrado de comprobantes fiscales, el SADEC tiene como requisito implantar un método de

autenticación para usuarios administradores; debido a que el sistema se ejecutará sobre plataforma Web el método seleccionado para esta autenticación fue mediante la validación de un nombre de usuario y contraseña.

En el diagrama Entidad-Relación de la base de datos del SADEC, existe la entidad Responsable, en la cual se almacenarán los datos de acceso de los usuarios encargados de gestionar los distintos procesos del sistema, entre estos datos se tienen los campos nombre de usuario y contraseña del responsable, los cuales servirán como credenciales de acceso a las secciones administrativas del sistema. Para mayor seguridad las contraseñas de los usuarios responsables serán almacenadas aplicando el algoritmo de reducción criptográfico MD5.

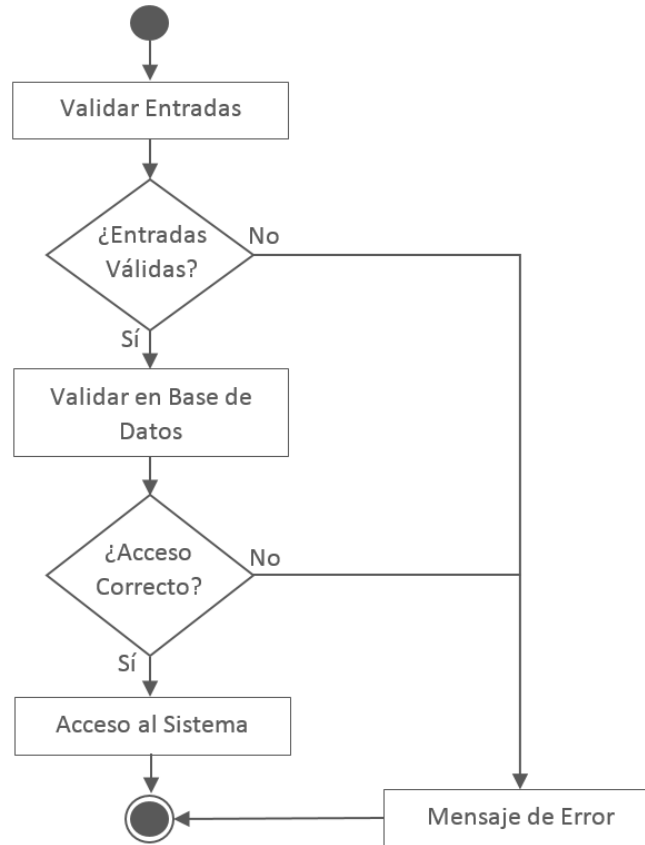
La figura 4.7.1.1 muestra el formulario de acceso a secciones administrativas del sistema. Como se puede observar, este formulario se compone de dos campos de texto para el usuario y contraseña, así como de un botón para su envío.

El módulo encargado de procesar los datos introducidos desde el formulario se compone de un método que tiene como parámetros el nombre de usuario y contraseña a validar. La figura 4.7.1.2 muestra el diagrama de flujo de este método.



El formulario de acceso al sistema tiene un encabezado gris con el texto "Bienvenid@" en blanco. Debajo del encabezado, hay un texto que dice "Para ingresar al sistema favor de proporcionar sus datos de acceso:". A continuación, hay dos campos de texto: "Usuario:" y "Contraseña:". Debajo de los campos de texto, hay un botón que dice "Ingresar".

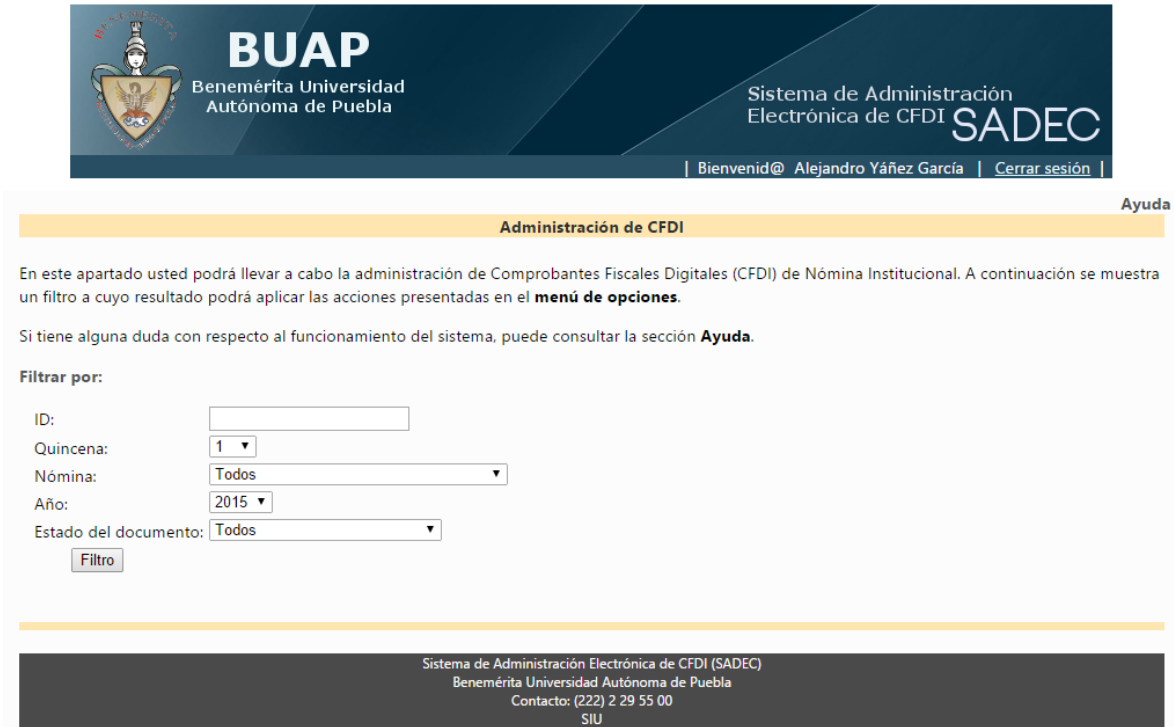
**Figura 4.7.1.1.** Formulario de acceso al sistema



**Figura 4.7.1.2.** Diagrama de flujo del módulo de validación de usuario

Con el fin de evitar posibles inyecciones de código malicioso, en el módulo de autenticación de usuarios se realiza una validación de la información recibida para detectar caracteres especiales. Si esta prueba se realiza con éxito, es decir, las entradas no contienen caracteres especiales, se procede a validar la existencia del usuario en la base de datos del sistema, para posteriormente permitir o denegar el acceso según sea el caso. Si el nombre de usuario y/o contraseña proporcionados contienen caracteres especiales se desplegará un mensaje de error al usuario. En caso de que la validación del nombre de usuario y contraseña en la base de datos no sea satisfactoria se mostrará un mensaje de error indicando este incidente.

Al realizarse una autenticación exitosa en el sistema, internamente se creará una instancia de una clase que contará con la información más relevante del usuario autenticado. Esta instancia existirá como variable de sesión mientras la sesión del usuario se encuentre activa. La figura 4.7.1.3 muestra un ejemplo de acceso exitoso al sistema, en donde se puede observar que aparece el nombre del usuario a quien pertenece la cuenta, así como una liga con la leyenda "Cerrar sesión", la cual, como su nombre lo indica, terminará la sesión del usuario actualmente autenticado.



The screenshot displays the SADEC (Sistema de Administración Electrónica de CFDI) interface. At the top left is the BUAP logo (Benemérita Universidad Autónoma de Puebla). The main header area contains the text 'Sistema de Administración Electrónica de CFDI SADEC'. Below this, a navigation bar includes 'Bienvenid@ Alejandro Yáñez García' and a 'Cerrar sesión' link. A yellow banner reads 'Administración de CFDI' with an 'Ayuda' link on the right. The main content area explains the system's purpose and provides a 'Filtrar por:' section with input fields for ID, Quincena (set to 1), Nómina (set to Todos), Año (set to 2015), and Estado del documento (set to Todos), along with a 'Filtro' button. The footer contains contact information for the SIU (Sistema de Información de la Universidad).

**Figura 4.7.1.3.** Acceso exitoso al SADEC

## 4.7.2 Pruebas de Autenticación de Usuarios Administrativos

Las pruebas del módulo de autenticación de usuarios se realizaron haciendo uso de un diccionario de datos e intentos de inyección de código. De las pruebas se pudo concluir que el módulo realiza correctamente la autenticación de usuarios y evita satisfactoriamente la inyección de código.

## 4.7.3 Privilegios de Usuarios Administradores

De acuerdo a los requisitos especificados del sistema, se deberá contar con un sistema de privilegios con el fin de asignar tareas a distintas dependencias según sea su función. Para cubrir esta necesidad, como se describió durante la fase de análisis del sistema (Capítulo 2, sección 2.2), es necesario contar con roles, los cuales contarán con un conjunto de privilegios asociados.

Los privilegios de los que se puede conformar un rol se presentan en el listado 4.7.3.1, el cual contiene el nombre del privilegio y una descripción del mismo. Cabe señalar que estos privilegios fueron especificados por la institución.



Privilegio	Descripción
LOGIN	Indica si el rol tendrá acceso al sistema
GENERAR	Permite/Deniega la generación de comprobantes fiscales digitales
SELLAR	Permite/Deniega el sellado de CFDI
TIMBRAR	Permite/Deniega el timbrado de CFDI
STATUS GENERADO	Permite/Deniega filtrar comprobantes en estado Generado
STATUS SELLADO	Permite/Deniega filtrar comprobantes en estado Sellado
STATUS TIMBRADO	Permite/Deniega filtrar comprobantes en estado timbrado
STATUS NO GENERADO	Permite/Deniega filtrar comprobantes sin generar
STATUS ERROR SELLO	Permite/Deniega filtrar comprobantes con estado Error de Sellado
STATUS ERROR TIMBRADO	Permite/Deniega filtrar comprobantes con estado Error de Timbrado
STATUS ERROR GENERADO	Permite/Deniega filtrar comprobantes con estado Error de Generado
STATUS TODOS	Permite/Deniega filtrar comprobantes sin importar su estado
ELIMINAR GENERADOS	Permite/Deniega la eliminación de comprobantes cuyo estado sea Generado
ELIMINAR SELLADOS	Permite/Deniega la eliminación de comprobantes cuyo estado sea Sellado
ELIMINAR ERROR SELLADO	Permite/Deniega la eliminación de comprobantes cuyo estado sea Error de Sellado
ELIMINAR ERROR GENERADO	Permite/Deniega la eliminación de comprobantes cuyo estado sea Error de Generación
REGRESAR A GENERADO	Permite/Deniega cambiar de estado Sellado/Error de Sellado a estado Generado
REGRESAR A SELLADO	Permite/Deniega cambiar de estado Error de Timbrado a Sellado

#### 4.7.3.1. Privilegios del sistema

Los roles definidos para el SADEC se muestran en el listado 4.7.3.2 junto con el conjunto de privilegios asociados.

Rol	Privilegios
Administrador	*Todos
Recursos Humanos	LOGIN GENERAR STATUS NO GENERADO STATUS GENERADO STATUS ERROR GENERADO ELIMINAR GENERADOS ELIMINAR ERROR GENERADO
Tesorería	LOGIN SELLAR TIMBRAR STATUS SELLADO STATUS TIMBRADO STATUS ERROR SELLO STATUS ERROR TIMBRADO ELIMINAR SELLADOS ELIMINAR ERROR SELLADO REGRESAR A GENERADO

**Figura 4.7.3.2.** Conjunto de Roles del SADEC

Como se observa en la figura anterior, existen tres roles, cada uno con privilegios para realizar tareas específicas, con excepción del privilegio de Administrador, el cual puede realizar cualquier actividad dentro del sistema.

#### 4.7.4 Pruebas de Implantación de Privilegios de Usuario

Las pruebas realizadas para la implantación de privilegios de usuarios se llevaron a cabo accediendo al sistema con el usuario propietario del rol e identificando las tareas a realizar permitidas.

#### 4.7.5 Revisión y Retrospectiva de Iteración 3

Las tareas involucradas en la iteración 3 del backlog de producto del SADEC tratan lo relacionado con la autenticación y privilegios de usuarios administradores. Durante la revisión de la ejecución de este sprint se concluyó que la implantación de los módulos encargados de la autenticación de usuarios se llevó a cabo de manera satisfactoria. La implantación de privilegios y roles no sufrió demora alguna, por lo que la ejecución de esta tercer iteración se logró en tiempo y forma, por ende, no se consideraron cambios a las tareas restantes priorizadas en la cartera de producto ni a los tiempos establecidos.

### 4.8 Ejecución de Iteración 4

Las tareas correspondientes al cuarto sprint de la cartera de producto del SADEC corresponden a la administración de usuarios administradores y privilegios. Esta sección

describe el desarrollo e implantación de estas tareas. Cabe señalar que el acceso a esta sección súper administrativa del sistema se realizará mediante una liga especial para aquellos usuarios con los privilegios suficientes.

#### 4.8.1 Administración de Roles

La sección de administración de roles consiste en la gestión de conjuntos de privilegios, los cuales podrán ser asignados a usuarios administradores del sistema. En la figura 4.8.1.1 se muestra una captura de la sección de administración de roles del SADEC.

ROL CVE.	NOMBRE	DESCRIPCIÓN	PRIVILEGIOS	ELIMINAR
196915	ADMINSITRADOR	ADMINISTRADOR DEL SISTEMA	<a href="#">Ver</a>	
196919	DISABLED	Usuario deshabilitado	<a href="#">Ver</a>	<a href="#">Eliminar</a>
367699	TESORERÍA	Acciones correspondientes a TG	<a href="#">Ver</a>	

**Figura 4.8.1.1.** Administración de roles

Como se observa en la figura 4.8.1.1, en esta sección se muestra una tabla con los roles existentes en el sistema. Para crear un nuevo rol bastará con especificar el nombre del rol junto con una descripción y a continuación presionar el botón Agregar Nuevo Rol. Si se desea modificar el conjunto de privilegios que conforman un rol, se deberá presionar sobre el enlace Ver de la columna PRIVILEGIOS, lo cual dará paso a una sección de edición de los privilegios con los que cuenta el rol seleccionado (Figura 4.8.1.2). La eliminación de un rol se puede llevar a cabo presionando sobre el enlace Eliminar. Esta última acción sólo se permitirá para aquellos roles que no se encuentren asignados a ningún usuario y que no cuenten con un conjunto de privilegios establecido.

Privilegios				
PRIV. CVE.	NOMBRE	DESCRIPCIÓN	AGREGAR	REMOVE
5	CANCELAR	CANCELACIÓN DE DOCUMENTOS		<a href="#">Remove</a>
39	DESCARGA ARCHIVOS (DESK)	DESCARGA MASIVA DE ARCHIVOS	<a href="#">Agregar</a>	
31	ELIMINAR ERROR GENERADO	ELIMINAR DOCUMENTOS CON STATUS: ERROR DE GENERADO	<a href="#">Agregar</a>	
28	ELIMINAR GENERADOS	ELIMINAR DOCUMENTOS GENERADOS	<a href="#">Agregar</a>	
29	ELIMINAR SELLADOS	ELIMINAR DOCUMENTOS SELLADOS		<a href="#">Remove</a>
2	GENERAR	GENERACIÓN DE DOCUMENTOS	<a href="#">Agregar</a>	
30	GENERAR ERROR GENERADO	VOLVER A GENERAR DOCUMENTOS CON STATUS: ERROR DE GENERADO	<a href="#">Agregar</a>	
1	LOGIN	*ACCESO AL SISTEMA		<a href="#">Remove</a>
38	LOGIN (DESK)	LOGIN EN APLICACIÓN DE ESCRITORIO	<a href="#">Agregar</a>	
35	REGRESAR A GENERADO	CAMBIAR ESTATUS DE DOCUMENTO A GENERADO		<a href="#">Remove</a>
36	REGRESAR A SELLADO	CAMBIAR ESTATUS DE DOCUMENTO A SELLADO		<a href="#">Remove</a>
37	REGRESAR A TIMBRADO	CAMBIAR ESTATUS DE DOCUMENTO A TIMBRADO		<a href="#">Remove</a>

**Figura 4.8.1.2.** Edición de privilegios de un rol

En la figura anterior se puede observar que en la edición de privilegios de un rol éstos se pueden agregar o remover presionando sobre el enlace correspondiente, lo que permite la creación y modificación del conjunto de privilegios que componen un rol.

Por razones de seguridad y consistencia de la información los distintos privilegios que pueden conformar un rol son considerados constantes y no pueden ser modificados desde el sistema.

### 4.8.2 Administración de Usuarios

En la sección administrativa de usuarios del SADEC se podrán crear nuevos usuarios, así como editar o eliminar usuarios existentes. En la figura 4.8.2.1 se puede ver la pantalla principal de la sección administrativa de usuarios.

Usuarios						
CVE. USUARIO	NOMBRE	NOMBRE DE USUARIO	DEPENDENCIA	ROL	EDITAR	ELIMINAR
			RECURSOS HUMANOS	ADMINISTRADOR	<a href="#">Editar</a>	
			TESORERÍA GENERAL	ADMINISTRADOR	<a href="#">Editar</a>	
			TESORERÍA GENERAL	TESORERÍA	<a href="#">Editar</a>	<a href="#">Eliminar</a>

**Figura 4.8.2.1.** Sección de administración de usuarios

La creación de un nuevo usuario podrá realizarse presionando sobre el enlace “Nuevo Usuario”. Esta acción mostrará un formulario donde se pedirá la información necesaria para agregar el nuevo usuario administrador al sistema (Figura 4.8.2.2).

The screenshot shows a web form titled "Nuevo Usuario". It contains the following fields and controls:

- Nombre:** Text input field.
- Apellido:** Text input field.
- Nombre de Usuario:** Text input field.
- Password:** Text input field.
- Dependencia:** Dropdown menu with "RECURSOS HUMANOS" selected.
- Rol:** Dropdown menu with "ADMINISTRADOR" selected.
- Actualizar:** Button.

At the bottom of the form, there is a footer with the following text:

Sistema de Administración Electrónica de CFDI (SADEC)  
Benemérita Universidad Autónoma de Puebla  
Contacto: (222) 2 29 55 00  
SIU

**Figura 4.8.2.2.** Agregar nuevo usuario

Como se observa en la figura anterior, la creación de un nuevo usuario administrador requiere del nombre y apellido del mismo para su identificación en las acciones realizadas dentro del sistema, nombre de usuario y contraseña para su autenticación, dependencia a la que pertenece y un conjunto de privilegios (rol).

La eliminación de usuarios administradores puede realizarse presionando sobre el enlace Eliminar de la sección administrativa de usuarios. Esta acción podrá realizarse siempre y cuando el usuario a eliminar no haya realizado operaciones administrativas críticas dentro del sistema.

### 4.8.3 Pruebas

Como pruebas de la implantación de las secciones administrativas del sistema, se creó un rol de prueba y se asignó a un usuario administrador. Al variar el conjunto de privilegios de este rol se pudo comprobar la eficacia de este método.

### 4.8.4 Revisión y Retrospectiva de Iteración 4

La iteración número cuatro de la cartera de producto del SADEC corresponde a la gestión de usuarios administradores del sistema. Por la naturaleza de estas tareas (altas, bajas, consultas, actualizaciones) esta iteración se realizó en tiempo y forma de acuerdo a lo priorizado en la cartera de producto, por lo que no fueron necesarios cambios en la programación de tareas ni en tiempos de entrega.

## 4.9 Ejecución de Iteración 5

En cuanto a la entrega de CFDI por concepto de nómina al usuario receptor, en la Resolución Miscelánea Fiscal para 2015 se establece:

*Para los efectos de los artículos 29, fracción V del Código Fiscal de la Federación y 99, fracción III de la Ley del ISR, los contribuyentes entregarán o enviarán a sus trabajadores el CFDI en un archivo con el formato electrónico XML de las remuneraciones cubiertas.*

*Los contribuyentes que se encuentren imposibilitados para cumplir con lo establecido en el párrafo anterior, podrán entregar una representación impresa del CFDI. Dicha representación deberá contener al menos los siguientes datos:*

- I. El folio fiscal.
- II. La clave en el RFC del empleador.
- III. La clave en el RFC de empleado

*Los contribuyentes que pongan a disposición de sus trabajadores una página o dirección electrónica que les permita obtener la representación impresa del CFDI, tendrán por cumplida la entrega de los mismos.*

Por lo anterior, se estableció como requisito poner a disposición del empleado todo CFDI expedido en representación impresa y formato electrónico XML.

En las tareas involucradas en la quinta y última iteración de la cartera de producto del SADEC se considera el desarrollo e implantación de la consulta de comprobantes fiscales digitales de nómina por parte del receptor.

### 4.9.1 Autenticación de trabajadores

El acceso a la consulta de comprobantes fiscales de nómina emitidos a trabajadores se restringió de manera que un trabajador pudiera consultar sus comprobantes fiscales mediante credenciales de acceso propias. Para cubrir este requisito, se desarrolló un módulo de autenticación especial para trabajadores, el cual, valida la información proporcionada en tablas de una base de datos existente propia de la BUAP, las cuales contienen la información de contacto necesaria del trabajador. Por esto, y por instrucciones del responsable del producto (product owner), se determinó utilizar el número de identificación del trabajador junto con su fecha de nacimiento como usuario y contraseña respectivamente para el primer acceso al portal de consulta de CFDI. Una vez autenticado el usuario, se ofrecerá la posibilidad de cambiar la contraseña actual, así como su información de contacto. Se determinó mostrar un mensaje de aviso de privacidad cada vez que el trabajador acceda al portal de consulta de CFDI (Figura 4.9.1.1).



Figura 4.9.1.1. Aviso de Privacidad BUAP

Para prever situaciones de olvido de contraseñas, se desarrolló una sección de reinicio de contraseña. Para acceder a esta sección se deberá presionar sobre el enlace ¿Olvidó su contraseña? del formulario de acceso (Figura 4.9.1.2), lo cual presentará al usuario un formulario de reinicio, en donde se solicitará el ID de trabajador junto con su fecha de nacimiento.



Figura 4.9.1.2. Formulario de acceso a portal de consulta de CFDI


### 4.9.2 Consulta y Descarga de CFDI para Trabajadores

El módulo de consulta y descarga de CFDI consta de dos secciones. En la sección principal se podrán consultar los comprobantes fiscales emitidos al trabajador, pudiendo seleccionar el año del cual desee realizar la consulta. En la figura 4.9.2.1 se presenta un ejemplo de esta sección. Como se puede observar, se ofrece una lista desplegable que contendrá los distintos años en los que se han emitido comprobantes fiscales al trabajador. Una vez seleccionado el año de consulta, se mostrará una lista con los CFDI emitidos indicando la quincena para la cual se expedieron, el año al que pertenecen y la fecha de generación del comprobante, además, se ofrecerá la opción de visualizar y descargar el CFDI como archivo en formato electrónico XML o su representación impresa en formato PDF. La figura 4.9.2.2 muestra un ejemplo de la representación impresa de un CFDI, cuyo formato no se encuentra establecido por el SAT, por lo que se implantó un diseño propio de la universidad.

QUINCENA	AÑO	FECHA DE CREACIÓN	XML	PDF
	2014			
	2014			
	2014			
	2014			
	2014			
	2014			
	2014			
	2014			

Figura 4.9.2.1. Sección de consulta de CFDI para trabajadores





**Benemérita Universidad Autónoma de Puebla**  
 4 sur 104 Centro Histórico C.P. 72000  
 Puebla, Puebla  
 RFC: UAP370423PP3

Puebla, Pue. a 19 de Enero de 2015 15:42:50

<b>Nombre:</b> [REDACTED]	<b>RFC:</b> [REDACTED]
<b>CURP:</b> [REDACTED]	<b>ID:</b> [REDACTED]
<b>Periodo de Pago:</b> 2014-05-16 al 2014-05-31	<b>Fecha de Pago:</b> 2014-05-31
<b>Método de Pago:</b> Depósito en cuenta	

**PERCEPCIONES**

Clave SAT	Descripción	Gravado	Exento
008	AYUDA	[REDACTED]	[REDACTED]
016	OTROS	[REDACTED]	[REDACTED]
001	SUELDOS, SALARIOS RAYAS Y JORNALES	[REDACTED]	[REDACTED]
022	PRIMA POR ANTIGÜEDAD	[REDACTED]	[REDACTED]
<b>Totales:</b>		[REDACTED]	[REDACTED]

**DEDUCCIONES**

Clave SAT	Descripción	Gravado	Exento
001	SEGURIDAD SOCIAL	[REDACTED]	[REDACTED]
004	OTROS	[REDACTED]	[REDACTED]
002	ISR	[REDACTED]	[REDACTED]
019	CUOTAS SINDICALES	[REDACTED]	[REDACTED]
<b>Totales:</b>		[REDACTED]	[REDACTED]

( [REDACTED] PESOS 34/100 M.N.)

**Importe Neto:** [REDACTED]

**Sello Digital del Emisor**  
[REDACTED]

**Sello digital del SAT**  
[REDACTED]


**UUID**  
[REDACTED]

**Fecha y hora de Certificación**  
[REDACTED]

**No. de Serie del Certificado del SAT**  
[REDACTED]

Este documento es una representación impresa de un CFDI.  
 La reproducción apócrifa de este comprobante constituye un delito en los términos de las disposiciones fiscales.

**Pago en una sola exhibición**



#### 4.9.2.2. Representación impresa de CFDI

La segunda sección del portal de consulta de CFDI está dedicada a la actualización de los datos personales del trabajador, tales como teléfono, correo electrónico y contraseña de acceso. Se puede acceder a esta sección mediante el botón Mis Datos del menú de opciones en la sección principal (Figura 4.9.2.3).

Benemérita Universidad Autónoma de Puebla

SADEC  
Sistema de Administración Electrónica de CFDI

Bienvenid@ | Cerrar sesión | Ayuda


**Actualización de datos**

← Regresar

A continuación se presentan los datos de contacto registrados en el sistema asociados a su cuenta.  
Para actualizar esta información, proporcione los datos requeridos y presione el botón **Guardar**

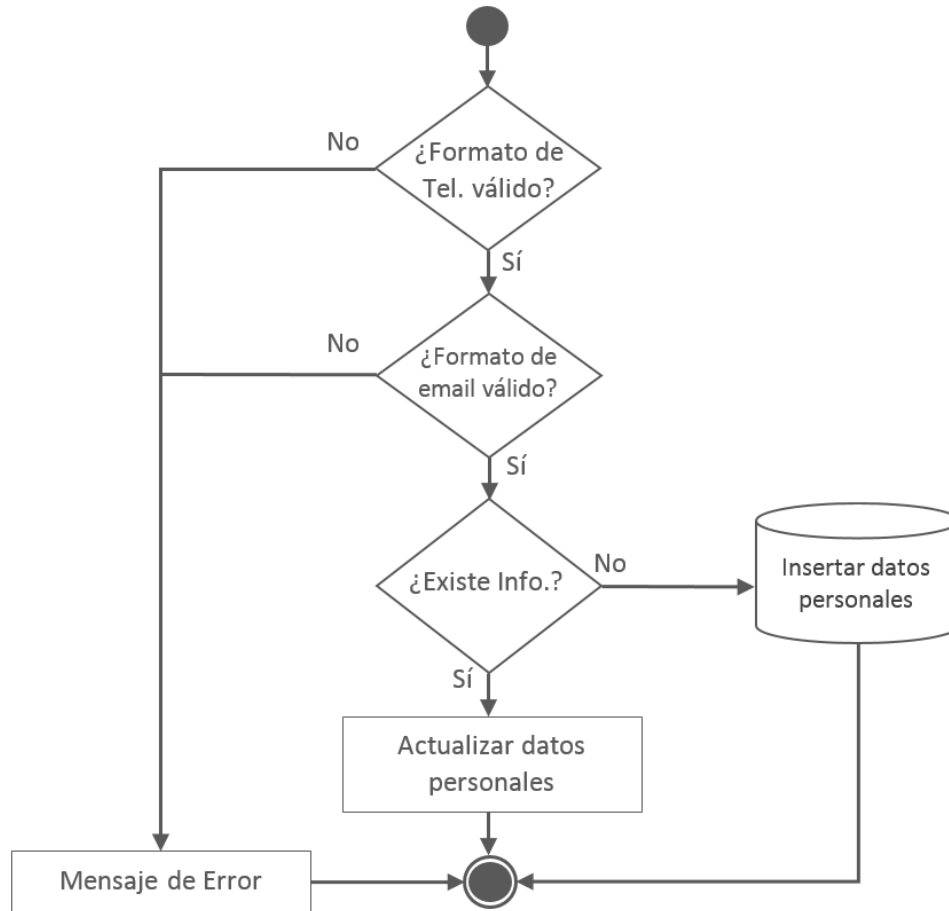
Teléfono:

Correo Electrónico:

  
**Editar mi  
Contraseña**

**Figura 4.9.2.3.** Sección de actualización de datos personales del portal de consulta de CFDI

Tomando en cuenta que el módulo responsable de la actualización de datos personales de trabajadores accede a tablas de una base de datos ajena al SADEC, se muestra el diagrama de flujo del módulo en la figura 4.9.2.4.



**Figura 4.9.2.4.** Diagrama de flujo de actualización de datos personales

### 4.9.3 Pruebas del Portal de Consulta de CFDI

Las pruebas realizadas al portal de consulta de CFDI constan de dos fases, la primera consiste en probar la consulta y descarga de CFDI, así como la actualización de datos personales de manera interna al área de Soluciones de Software. Para esta prueba se utilizaron identificadores únicos controlados. La segunda fase de pruebas consiste en dar acceso al personal de una dependencia de tamaño moderado de la BUAP, de esta manera, se podrá generar una retroalimentación con los usuarios involucrados para lograr mejorar la usabilidad y rendimiento de este portal.

### 4.9.4 Revisión y Retrospectiva de Iteración 5

Durante el desarrollo de las tareas priorizadas en la quinta iteración de la cartera de producto del SADEC se anunció por parte del área de Contabilidad de Tesorería General que por ley, se debía ofrecer al trabajador el CFDI de recibo de nómina en su representación impresa. Debido a que esta tarea no se contempló en la especificación de requisitos del SADEC, no se encontraba priorizada en el backlog de producto, lo que ocasionó un retraso en los tiempos

de entrega, así como esfuerzo extra debido a los cambios necesarios y la implantación de librerías involucradas no contempladas en un principio, sin embargo, debido al patrón arquitectural implantado para el desarrollo del sistema, no se presentó mayor inconveniente al aplicar los cambios antes mencionados. A pesar del retraso sufrido durante la ejecución de esta iteración, la implantación de las tareas se logró de manera exitosa.



# Capítulo 5

## Conclusiones y Perspectivas

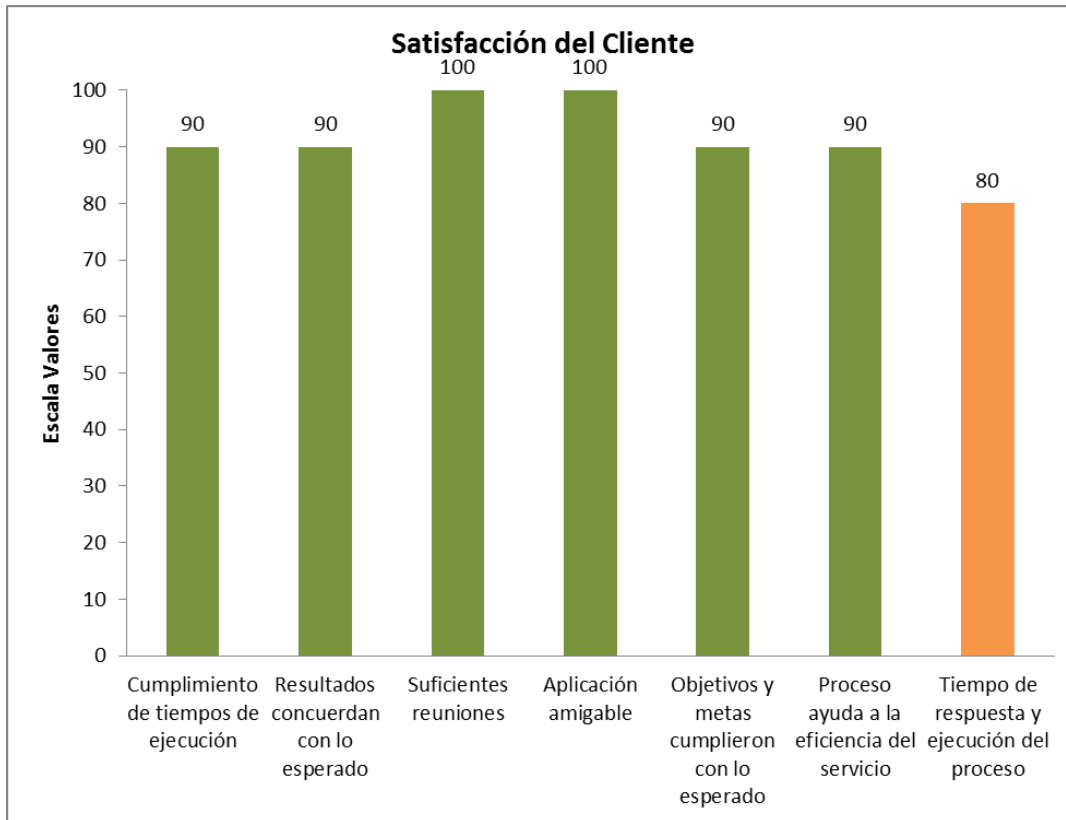
### 5.1 Conclusiones

De acuerdo al **Objetivo general** planteado al inicio de este documento de tesis, se puede concluir que se logró satisfactoriamente su cumplimiento, pues se realizó el análisis, diseño e implantación de un sistema (SADEC), el cual permite la generación, administración, almacenamiento y consulta de Comprobantes Fiscales Digitales por Internet de la Nómina Institucional de la Benemérita Universidad Autónoma de Puebla.

En cuanto a los **Objetivos particulares** expuestos al inicio de la tesis, se puede confirmar su ejecución satisfactoria, pues, se recopiló la información necesaria de Ingeniería de Software, del marco de trabajo Scrum, del patrón arquitectural por capas y del funcionamiento del proceso de emisión de comprobantes fiscales (CFDI). Esta información fue utilizada durante las fases de análisis, diseño e implantación tanto del sistema como de la base de datos, cumpliendo en ambos casos con los requisitos especificados, además, se elaboró en su totalidad el documento de tesis en el tiempo contemplado.

Durante el desarrollo de este trabajo de tesis se pudo observar que haciendo uso del modelo de desarrollo ágil Scrum es posible diseñar, desarrollar, implantar y probar sistemas de software de tamaño considerable en un periodo de tiempo relativamente corto. Además, se comprobó que utilizando el patrón arquitectural por capas enfocado a aplicaciones empresariales es posible realizar pruebas de manera rápida y confiable, así como de implantar cambios durante el desarrollo de un sistema y después de su implantación gracias a la flexibilidad que el patrón arquitectural otorga.

Como parte del proceso de certificación en la Norma ISO 9001:2008 del Sistema de Gestión de Calidad, el Área de Soluciones de Software del Sistema de Información Universitaria (SIU) de la BUAP realizó distintas encuestas de satisfacción del cliente a usuarios administradores del SADEC. Los resultados de estas encuestas se presentan en la siguiente gráfica, validando las conclusiones expuestas anteriormente.



**Figura 5.1.1.** Gráfica de satisfacción del cliente del SADEC

Como se puede apreciar en la gráfica anterior, la utilización de las metodologías de desarrollo seleccionadas así como los patrones arquitecturales implantados dan como resultado un grado de satisfacción del cliente aceptable.

## 5.2 Perspectivas

Las distintas dependencias que integran la Benemérita Universidad Autónoma de Puebla emiten comprobantes fiscales digitales de los distintos servicios que proporcionan a través de distintos proveedores autorizados de certificación. En un futuro, el SADEC podrá extender su funcionalidad para la gestión y emisión de CFDI emitidos por estas dependencias haciendo las adaptaciones necesarias a los módulos de generación, sellado, timbrado y almacenado de comprobantes fiscales. Además, el SADEC podrá servir como base para la solución de dificultades similares a las detectadas en la BUAP con respecto a la emisión de CFDI de nómina.

Un aspecto importante a tomar en cuenta es la constante evolución del Código Fiscal de la Federación y de la Resolución de Miscelánea Fiscal y sus Anexos. Esta situación se refleja en posibles cambios a distintos módulos del SADEC, los cuales podrán ser solventados de manera rápida y confiable gracias a la separación por capas del patrón arquitectural utilizado en el desarrollo e implantación del sistema.





# Apéndice A

## Instalación y Puesta a Punto del SADEC

La instalación del SADEC requiere de un servidor Web Microsoft Information Services 7.0 y un Sistema Administrador de Bases de Datos Oracle 11G, pudiendo utilizar en ambos casos versiones superiores, sin embargo, las pruebas de implantación del sistema se llevaron a cabo utilizando las versiones mencionadas.

A continuación se listan los pasos necesarios para la instalación del SADEC y su puesta a punto:

- 1.- Crear un usuario en el Sistema Administrador de Bases de Datos Oracle para el SADEC, preferentemente de uso exclusivo.
- 2.- Importar la base de datos del sistema (archivo con extensión .dmp) al Sistema Administrador de Bases de Datos Oracle. Esta operación se puede llevar a cabo utilizando la herramienta Oracle Data Pump Import (impdp).
- 3.- Crear un nuevo sitio en el servidor Web, indicando la ubicación del directorio que contendrá los archivos necesarios del sistema
- 4.- Copiar en el directorio designado para alojar el SADEC:
  - Los archivos obtenidos a partir de la compilación y construcción del sistema
  - Transformador XSLT proporcionado por el SAT para la obtención de cadena original de comprobantes
  - Archivo con extensión .pfx utilizado para el sellado de CFDI
- 5.- Acceder al sistema mediante la URL del sitio creado para alojar el SADEC



## Bibliografía

- [1] **PRESSMAN, Roger**. Ingeniería de Software. Un enfoque práctico. 7ª. ed. México: MCGRAW-HILL INTERAMERICANA, 2010, 900 p. ISBN: 9786071503145.
- [2] **SOMMERVILLE, Ian**. Ingeniería de Software. 9ª. ed. México: Pearson, 2011, 792 p. ISBN: 9786073206037.
- [3] **RUBIN, Kenneth**. Essential Scrum. A practical guide to the most popular agile process. 1ª ed. USA: PEARSON EDUCATION, 2012, 504 p. ISBN: 0137043295.
- [4] **ARLOW, Jim**. UML and the Unified Process. Practical Object-Oriented Analysis & Design. 1ª ed. USA: ADDISON-WESLEY PROFESSIONAL, 2002, 416 p. ISBN: 0201770601.
- [5] **En línea**. Enterprise Design and Architecture. USA: Microsoft Development Network, 2014. [Consulta: 26 Junio 2014]. Disponible en: [http://msdn.microsoft.com/en-us/library/aa244216\(v=vs.60\).aspx](http://msdn.microsoft.com/en-us/library/aa244216(v=vs.60).aspx)
- [6] **BUSCHMANN, Frank; MEUNIER, Regine; ROHNERT, Hans, SOMMERLAD, Peter, STAL, Michael**. Pattern-Oriented Software Architecture. A system of patterns. 1ª. ed. Inglaterra: JOHN WILEY & SONS, 1996, 487 p. ISBN: 0471958697.
- [7] **GARCÍA-MOLINA, Hector; ULLMAN, Jeffrey; WIDOM, Jennifer**. Database Systems, The Complete Book. 2ª ed. USA: Prentice Hall, 2009, 1248 p. ISBN: 0131873253.
- [8] **ST DENIS, Tom; JOHNSON, Simon**. Cryptography for Developers. 1a ed. Syngress Publishing, Incorporated, 2007, 423 p. ISBN: 1597491047.
- [9] **FOWLER, Martin, Patterns of enterprise application architecture**. USA: Addison Wesley, 2002, 560 p. ISBN: 0321127420.
- [10] **MURACH, Joel; SYVERSON, Bryan**. Murach's SQL Server 2008 for developers. 3ª ed. USA: Mike Murach & Associates, Inc., 2008, 778 p. ISBN:978-1-890774-51-6.
- [11] **En línea**. World Wide Web Consortium. W3C, 2015. [Consulta: 06 de Enero de 2015]. Disponible en: <http://www.w3.org/TR/ws-arch/#whatis>

---

[12] **En línea.** ANEXO 20 de la Resolución Miscelánea Fiscal para 2014; México, Secretaría de Hacienda y Crédito Público, 2013. Disponible en [http://www.dof.gob.mx/nota\\_detalle.php?codigo=5332803&fecha=17/02/2014](http://www.dof.gob.mx/nota_detalle.php?codigo=5332803&fecha=17/02/2014)

[13] **En línea.** Twelve Principles of Agile Software. 2001. [Consulta: 01 de Noviembre de 2014]. Disponible en: <http://www.agilemanifesto.org>

