

# Quetzalcoatl: Una Herramienta para Generar Contratos de Desarrollo de Software en Entornos de Outsourcing

Jezreel Mejía <sup>1</sup>, Sergio D. Ixmattlahua <sup>2</sup>, Alma I. Sánchez <sup>2</sup>

jez.mejia@gmail.com, sergio.ixmatlahua@outlook.com, almixanchez@gmail.com

<sup>1</sup> Centro de Investigación en Matemáticas, Unidad Zacatecas. Avenida Universidad Num. 222, Colonia Hidrahulica, Zacatecas, Zacatecas.

<sup>2</sup> Instituto tecnológico de Orizaba, Departamento de Posgrado e Investigación. Avenida Instituto Tecnológico, No. 852, Colonia Emiliano Zapata, Orizaba, Veracruz, México.

DOI: 10.4304/risti.e1.47-60

**Resumen:** Actualmente el outsourcing es una de las actividades principales de trabajo. Sin embargo, las relaciones que se dan entre un cliente y un proveedor de servicios no son lo suficientemente fuertes para lograr las expectativas de los acuerdos. El contrato de outsourcing para proyectos en desarrollo de software es una alternativa a este tipo de relaciones. En este artículo se presenta la arquitectura de la herramienta *Quetzalcoatl*, así como las funciones principales que ofrece la herramienta, todo esto con el objetivo de generar y evaluar contratos para proyectos de desarrollo de software en entornos de outsourcing como apoyo a las PYMES.

**Palabras-clave:** Outsourcing, desarrollo de software, Web, modelo de contrato, arquitectura. Internet, Quetzalcoatl.

## *Quetzalcoatl: A Tool for Generating Software Development Contracts in Outsourcing Environments*

**Abstract:** Nowadays, outsourcing is one of the most important work activities for the software development companies. However, the relationships between a client and a service provider are not strong enough to meet the expectations of the agreements. The outsourcing contract for software development projects is an alternative to this type of relationship. This paper presents the architecture of the tool named *Quetzalcoatl*, also the main functions that this tool offers in order to generate and evaluate contracts for software development projects in outsourcing environments to support SMEs.

**Keywords:** Outsourcing, software development, Web, contract model, architecture, Internet, Quetzalcoatl.

## 1. Introducción

La evolución del outsourcing de Tecnologías de la Información (TI) ha crecido a pasos agigantados desde los años 60's hasta el día de hoy, convirtiéndose así, en una estrategia empresarial rentable, desde los grandes corporativos mundiales hasta las Pequeñas y Medianas Empresas (PYMEs) (Jae-Nam, 2003). Sin duda alguna, esto ha sido el resultado de la evolución de las estrategias de negocio como un factor determinante para el desarrollo de las empresas.

Por un lado, algunos estudios realizados por el Instituto de Ingeniería de Software (SEI) (Gallagher, 2007), muestran que entre el 20 y 25% de los grandes proyectos de outsourcing fracasan dentro de los primeros años y un 50% en los primeros 5 años. Dentro de algunas razones de estos fracasos se encuentran: una mala gestión del proyecto, la definición equivocada de los requerimientos, la inexistencia de procesos para la selección de proveedores, la falta de un contrato específico de TI fundamentado en la Ingeniería de Software y también la falta de herramientas especializadas en la generación de contratos en entornos de outsourcing.

Por otro lado, en México, la *International Data Corporation* (International Data Corporation, 2011) indicó que para el año 2011 el mercado del outsourcing de TI crecería unos 10 billones de dólares que representan un crecimiento del 7% con respecto al año 2010. Sin embargo pese a que este mercado cada vez es más demandado por todo tipo de clientes, ya sean empresas transnacionales, PYMEs o particulares, sufren problemas de cumplimiento de las actividades parciales o totales del proyecto de desarrollo de software en este tipo de entornos. (Mattehew K. O, 1996) indica que la definición de un contrato, sin duda, es la clave para que una relación exista entre clientes y proveedores de TI, con las expectativas que se esperan de ambas partes. Por lo tanto, en este contexto, el contrato es un mecanismo para asegurar que las expectativas sean cumplidas, ya que los contratos definen los derechos, obligaciones, responsabilidades y las expectativas que corresponden a ambas partes. Además, la falta de herramientas que apoyen la generación y evaluación de contratos como apoyo a la toma de decisiones de las PYMEs es, sin duda alguna, una alternativa para mejorar aun más las relaciones entre las partes.

Con base en lo anterior, el objetivo de este artículo es presentar la arquitectura de la herramienta *Quetzalcoatl*, que integra el modelo de contrato definido en (Mejía, 2010), así como las principales funcionalidades que ofrecerá la herramienta tanto a las grandes organizaciones como a PYMEs y particulares que ofrezcan o contraten servicios para el desarrollo de software.

Este artículo está estructurado de la siguiente manera: en la sección 2 se presenta el modelo de contrato y método de evaluación que sustenta la herramienta, su arquitectura, principales funcionalidades. En la sección 3 se presentan las conclusiones del trabajo realizado.

## 2. Quetzalcoatl

Es una herramienta que tiene como objetivo proveer de elementos que ayuden a las organizaciones, tanto grandes, PYMEs como a particulares, a conocer los componentes primordiales que un contrato de desarrollo de software en entornos de outsourcing

debe tener. Para establecer la arquitectura de la herramienta, se analizó el modelo de contrato y el método de evaluación. En las siguientes secciones se describen el modelo de contrato y el método de evaluación definido en (Mejía, 2010) el cual la herramienta *Quetzalcoatl* toma como referencia para la definición de la funcionalidad que ofrece,

En ese sentido, *Quetzalcoatl* es una herramienta que integra un modelo de contrato de Ingeniería de Software basado en estándares internacionales como *CMMI-ACQ*, *ISO-12207:2008* y *PMBOOK*. Dicho modelo se describe brevemente en la sección 2.1.

## 2.1. Modelo de Contrato de Desarrollo de Software

Cabe mencionar que la herramienta se basa en un modelo de contrato y método de evaluación definido en (Mejía, 2010). Por lo que, en esta sección se describe de manera breve el modelo de contrato y método de evaluación. Cada una de estas categorías está formada por cláusulas, que se interrelacionan con el mismo tópico, de esta manera, se define a una cláusula como un conjunto de componentes relacionados entre sí, y a un componente como una actividad o información básica que debe ser incluida dentro de un contrato.

La tabla 1 muestra la descripción breve de cada una de las categorías que componen al modelo de contrato.

Tabla 1 - Categorías del modelo de contrato

Categorías del modelo de contrato	Descripción
Servicios	Las cláusulas dentro de esta categoría establecen el tipo de servicio, el alcance y la naturaleza de todos los servicios requeridos, el tiempo en que los servicios deberían estar listos y los niveles de rendimiento necesario.
Gestión de Servicios	Las cláusulas agrupadas en esta categoría establecen y evalúan la relación entre las distintas partes implicadas para garantizar el mantenimiento de la relación.
Financiera	En esta categoría se abordan los aspectos referentes a los acuerdos sobre precios y pagos, incluyendo incentivos y penalizaciones
Duración	Las cláusulas de esta categoría ayudan a especificar las fechas acordadas durante la negociación definida al inicio del contrato, la finalización, las fechas de entrega de cada servicio y su duración.
Transición	Las cláusulas dentro de esta categoría hacen referencia a la elaboración de procedimientos que deben cumplirse con objeto de garantizar las condiciones actuales para la finalización o la transferencia de los activos relacionados con el servicio o producto.
Seguridad	El objetivo de las cláusulas agrupadas en esta categoría es dar garantía a los intereses del cliente en dado caso que se presente una finalización anticipada del contrato, proporcionándole al cliente una posición más fuerte ante una futura renegociación del contrato.
Comunicación	En esta categoría las cláusulas establecen las normas y procedimientos para futuras contingencias, las cuales permitirán mejores resultados.

## 2.2. Método de Evaluación

Otra característica que la arquitectura de la herramienta *Quetzalcoatl* debe considerar, es la capacidad de realizar evaluaciones de los contratos de acuerdo al método de evaluación que se definió en (Mejía Miranda, 2010). Esta evaluación puede realizarse

sobre contratos creados directamente en la herramienta o contratos propiedad del usuario. El Método de Evaluación definido permite la evaluación de los contratos, y tener en consideración los aspectos que deberá cumplir el contrato del proveedor para satisfacer las necesidades del cliente.

Este método se basa en dos criterios:

1. Tomar como base el modelo de contrato: Tomando como base el Modelo de Contrato. Este método de evaluación tiene como objetivo identificar el porcentaje de cobertura de la cláusula, de la categoría y del contrato, tomando como referencia la suma de los porcentajes alcanzados por cada componente en el Modelo de Contrato.
2. Tomar como base los objetivos del negocio: Tomando como base los objetivos del negocio. Este método de evaluación tiene como objetivo identificar el porcentaje de cobertura de la cláusula o de la categoría o del contrato tomando como base la ponderación que se asigne a cada uno de los componentes, cláusulas y categorías del modelo, de acuerdo a los objetivos del negocio. En otras palabras, se asigna el porcentaje que refleje el grado de importancia que tiene para el negocio dicho componente, cláusula o categoría.

### **2.3. Arquitectura para la Herramienta Quetzalcoatl**

A lo largo de esta sección, se describe la arquitectura, tomando en consideración el análisis del modelo de contrato y método de evaluación que integrará la herramienta *Quetzalcoatl*.

De acuerdo con (I. Jacobson et. at., 2000), la arquitectura de un sistema de información define no solo la estructura y comportamiento, si no también, el uso, la funcionalidad, el rendimiento, la flexibilidad, la reutilización, la facilidad de comprensión, las restricciones y compromisos económicos y tecnológicos, y por supuesto la vista. Algunos investigadores definen a la arquitectura de un sistema de software como el conjunto de decisiones de diseño tomadas para un sistema (Tahuítón Mora, 2011). Si se hace una analogía con la construcción de un edificio, se puede asumir que la arquitectura representa a los planos de construcción del software.

La arquitectura de la herramienta está definida bajo el patrón arquitectónico Modelo-Vista-Controlador, y de acuerdo a las recomendaciones realizadas en (Tahuítón Mora, 2011). , ya que en este patrón es notable la separación de los componentes relacionados con los datos de la aplicación de los componentes de la interfaz de usuario. Permitiendo tener, a nivel de desarrollo, un código más claro, flexible y reusable.

Cuando la aplicación sufre un cambio en su modelo de objetos, se envía un aviso del cambio a la vista de manera que los elementos afectados puedan ser modificados o actualizados. Estos avisos se emiten al controlador típicamente, de manera que el controlador puede modificar la vista si se diera el caso. La vista se comunica con el modelo para obtener los datos necesarios para mostrarlos en la interfaz. Al momento de realizar una entrada por parte del usuario, el controlador está pendiente de la información de la vista para verificar qué medidas tomar de acuerdo a las solicitudes hechas. Después, en caso de que el modelo de objetos cambia de valores, esto debe ser comunicado a la vista y al controlador con el objetivo de que la interfaz sea actualizada.

Como se mencionó anteriormente la arquitectura de *Quetzalcoatl* está dividida en dos capas: aplicación y datos. En la capa de aplicación se encapsula los elementos de la arquitectura basados en el patrón MVC, en la cual se han definido los siguientes módulos:

- Vista.

En este nivel se contiene la interfaz gráfica de usuario, que dan soporte a las pantallas gráficas de la herramienta *Quetzalcoatl*, estas interfaces (formularios, gráficos, imágenes, etc.) se componen de cuatro secciones correspondientes a: *vista\_contratos*, *vista\_metodos*, *vista\_usuarios*, *vista\_plantillas*.

- La sección de *vista\_contratos* encapsula la interfaz referente a la administración de los contratos.
- La sección *vista\_metodos* contiene las interfaces correspondiente a los métodos de evaluación de los contratos.
- La sección *vista\_usuarios* contempla las interfaces de la administración de cuentas de usuario, login, registro y perfiles de usuario.
- La sección *vista\_plantillas* concentra al conjunto de interfaces que corresponden al proceso de creación de los patrones de contratos.

- Control.

En este nivel se define los módulos correspondientes a la capa de control, esta capa se encarga de las operaciones diversas que se realicen. Como se sabe de acuerdo al patrón MVC, la capa de control encapsula las clases correspondientes a cada una de las acciones o peticiones realizadas a través de la interfaz gráfica de usuario, por lo que se encuentran separadas de acuerdo a la funcionalidad que representan: *gestion\_contratos*, *gestion\_metodos*, *control\_perfiles*.

- La sección de *gestion\_contratos* corresponde al conjunto de clases escritas en java, la cuales representan a la capa de control para establecer la comunicación con la capa de vistas en la sección *vista\_contratos* y la capa de modelo en la sección *gestion\_contratos*, además de contener los diferentes métodos que permiten tomar las decisiones sobre que modelos se utilizan respecto a las peticiones que se envían desde la capa de vistas. Estas operaciones permiten la manipulación de los datos de todos los contratos que los usuarios generan durante el proceso de creación de los contratos.
- La sección de *gestión\_métodos* corresponde a las clases que definen las operaciones que permiten decidir que clases de la capa de control utilizar de acuerdo a las peticiones realizadas por la capa de vista. Esta sección incluye los métodos que permiten generar los cálculos para realizar la evaluación de cada uno de los contratos de los usuarios. Dado el caso de que la herramienta proporciona dos métodos de evaluación, se realizan diversas operaciones sobre los datos de los contratos.
- La sección *control\_perfiles* encapsula las clases que permite tomar las decisiones sobre el acceso a los diferentes módulos de la herramienta, tomando en cuenta el nivel y tipo de usuario.

- **Modelo.**

En este nivel se define la parte correspondiente a la lógica de negocio, es decir en términos coloquiales “el motor de nuestra herramienta”. Esta es la capa más importante de la herramienta, puesto que encapsula todas las clases que mantienen la relación entre los datos y la información que se requiere mostrar en la interfaz gráfica. Como se puede apreciar en la figura 1, esta capa está compuesta de cuatro paquetes encapsulando a las clases correspondientes de acuerdo a la funcionalidad que representan: *gestion\_contratos*, *gestion\_metodos\_evaluacion*, *gestion\_usuarios*.

- La sección *gestion\_contratos* de la capa de modelo, encapsula las clases más importantes que realizan las operaciones directamente sobre la base de datos que alberga la información de todos los contratos que los usuarios de la herramienta generan. El correcto funcionamiento de este módulo permite que las secciones de *gestion\_usuarios* y *gestion\_metodos\_evaluacion* trabajen de manera sincronizada. También en esta sección se encapsulan las clases que permiten la manipulación de la información que corresponde a la creación de nuevos patrones de contratos. Estas clases contienen método que permiten el ensamblado de un patrón de forma jerárquica: *categorías*, *cláusulas* y *componentes*.
- La sección *gestion\_metodos\_evaluacion* encapsula las clases con métodos que corresponden a los procesos que se realizan con los datos que se obtienen de la *vista\_metodos*. Estos datos se procesan para realizar la evaluación de acuerdo a los métodos de evaluación definidos en el apartado 2.2. Así mismo los datos generados son utilizados para generar los gráficos de rendimiento, por lo que también este módulo contiene los procedimientos que permiten generar los diversos gráficos de rendimiento que se ofrecen en la capa de vistas al usuario.
- La sección *gestion\_usuarios* engloban el conjunto de clases que permiten realizar la manipulación sobre los datos de las cuentas de los usuarios que tienen cuentas activas en la herramienta, así como el proceso de creación de nuevas cuentas. Todas estas operaciones se realizan directamente hacia el módulo de *Acceso\_Datos*.

En la capa de datos, se encapsulan los elementos que sirven de conexión con los elementos de persistencia de la información.

- *Acceso\_Datos*. En este módulo se encapsulan las clases de conexión a la base de datos que servirá para la persistencia de la información referente a los proyectos, cuentas de usuario y la gestión de los contratos y patrones.

## 2.4. Funciones de la herramienta

*Quetzalcoatl* busca ofrecer un ambiente de trabajo en entorno Web, con la finalidad de que cualquier persona con acceso a Internet pueda utilizarla, y así generar de manera fácil y rápida contratos a través de plantillas predefinidas en la herramienta, así como

también, realizar la evaluación de sus contratos. De manera que tanto los contratos creados en la herramienta como los contratos externos a la herramienta, sean evaluados.

En las siguientes subsecciones se presentan la descripción breve de las principales funciones que la herramienta *Quetzalcoatl* ofrece, comenzando por la creación de un patrón de contrato, seguido de la creación de un contrato para el desarrollo de un sitio Web, para una farmacia y para concluir la evaluación del contrato creado.

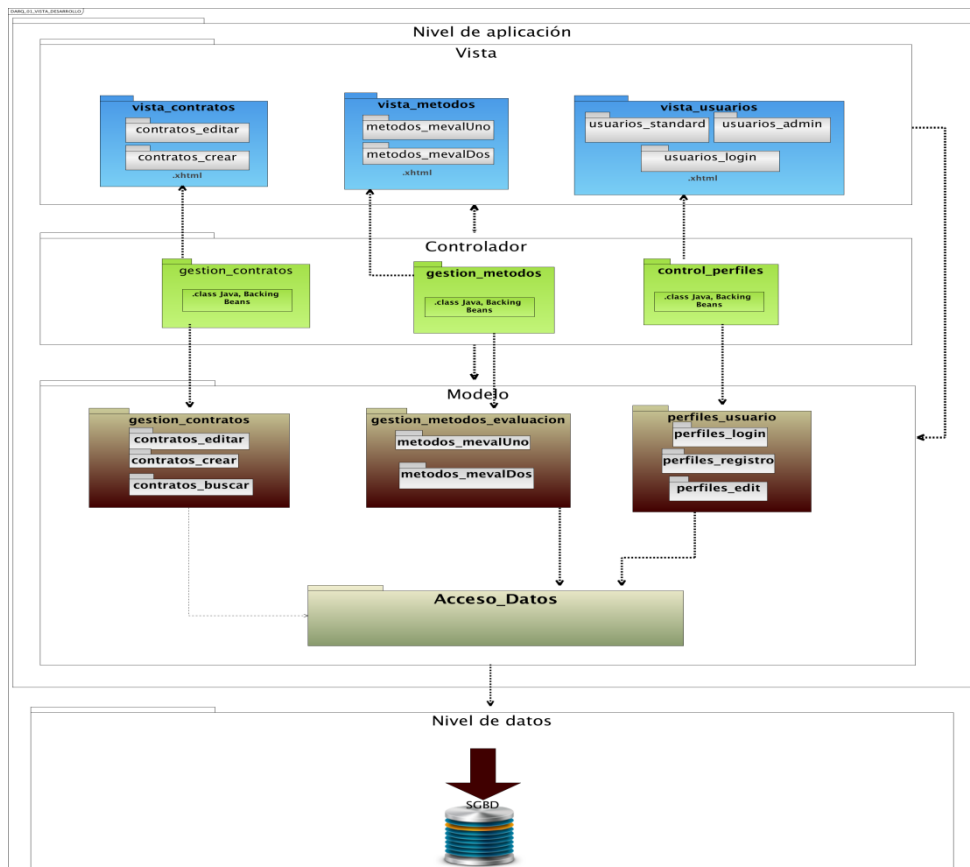


Figura 1 Arquitectura *Quetzalcoatl*

### 2.4.1. Creación de un patrón de contrato

En primer lugar, *Quetzalcoatl* cuenta con dos tipos de usuarios (Administrador y Estándar) con diferentes privilegios para acceder a la funcionalidad. En ese sentido, el usuario Administrador es el que tiene el privilegio de acceder a la gestión de los patrones de contrato, es decir, crear, actualizar y consultar todos los patrones que existan en la herramienta. Un patrón de contrato se puede definir como una plantilla

de un contrato, compuesto por un conjunto de elementos seleccionados del modelo de contrato de Ingeniería de Software, es decir, se compone de categorías, cláusulas y componentes según sea el tipo de contrato que represente cada patrón. En este caso se ejemplifica con la creación de un patrón para un contrato para el desarrollo de un sitio Web.

*Quetzalcoatl* define un sencillo proceso de creación de patrones de contrato, estableciendo cuatro sencillos pasos.

Paso 1.- Creación del patrón de contrato: en este paso se hace la captura del nombre del patrón, una breve descripción y el área al que pertenecerá dicho patrón.

Paso 2.- Agregar categorías al patrón: en este paso se agregan las categorías necesarias al patrón. Estas categorías se seleccionan de entre las siete categorías que el modelo de contrato ofrece.

Paso 3.- Agregar cláusulas al patrón: en este paso se agregan las cláusulas que están clasificadas dentro de las categorías agregadas en el paso anterior.

Paso 4.- Agregar componentes al patrón: en este paso se agregan los componentes encapsulados dentro de cada cláusula. Posteriormente se debe editar componente por componente el texto que cada elemento agregado tendrá a la hora de presentarse como un contrato al usuario.

En este último paso es dónde se agrega cada parte del texto que corresponde a cada componente como parte del patrón. En la Figura 2 se observa un resumen con todos los componentes que han sido agregados al patrón que se ha creado.

En la Figura 3 se observa cómo se realiza la edición de cada uno de los componentes para agregar el texto correspondiente. En este caso para editar el texto de un componente, es necesario seleccionar el componente deseado y a continuación dar click en editar. *Quetzalcoatl* carga un editor para agregar el texto, tablas, y elementos que se consideren necesarios para definir el contenido del componente.

Tal como se muestra en la figura 3, se agrega el texto *“Los requisitos funcionales definen cada una de las funciones que el CLIENTE solicita al PROVEEDOR, qué el sistema solicitado, cumpla al término del contrato”*, la cual representa al texto del componente seleccionado para editarse. El proceso de edición es similar para cada uno de los componentes, ya que *Quetzalcoatl* mantiene un mecanismo de interfaces con mucha facilidad de uso, lo que ayuda a que cualquier usuario pueda realizar este proceso sin problema alguno.



<input type="checkbox"/>	3	Roles	Se definen los roles identificados durante el análisis de los requisitos del sistema
<input checked="" type="checkbox"/>	4	Asignación de las tareas a los roles	Se especifican cada una de las tareas que cada rol debe realizar en el sistema
<input checked="" type="checkbox"/>	5	Información técnica sobre la funcionalidad	Se define la especificación técnica sobre la funcionalidad solicitada
<input checked="" type="checkbox"/>	6	Información técnica sobre la capacidad	Se definen las capacidades que el sistema tendrá al momento de ser terminado
<input type="checkbox"/>	7	Información técnica sobre la calidad	Se define información sobre los métodos de calidad que dan soporte al buen servicio del sistema
<input type="checkbox"/>	8	Información técnica sobre las Interfaces	Se especifica la información acerca de la usabilidad que tendrán las interfaces del sistema
<input type="checkbox"/>	9	Criterios de capacidad	Se definen cuales son los criterios de calidad que debe cubrir el sistema a la hora de ser entregado al cliente
<input type="checkbox"/>	10	Criterios de calidad	Se definen los criterios de calidad que el sistema debe cubrir al momento de ser entregado

[Agregar](#)

**Componentes agregados**

Id	Nombre
254	Requerimientos funcionales
255	Asignación de las tareas a los roles
256	Información técnica sobre la funcionalidad
257	Información técnica sobre la capacidad

Figura 2 - Edición de un patrón de contrato

**Paso 5: Editar componentes seleccionados**

Componentes Checkbox Based Select		
Id	Nombre	
254	Requerimientos funcionales	Define los requerimientos func
255	Asignación de las tareas a los roles	Se especifican cada una de las tareas
256	Información técnica sobre la funcionalidad	Se define la especificación téc
257	Información técnica sobre la capacidad	Se definen las capacidades q

[Editar componente](#)

**Editar elementos del componente**

Fuente HTML

Estilo Formato Fuente

Los requerimientos funcionales definen cada una de las funcionalidades que el CLIENTE solicita al PROVEEDOR que el sistema cumpla al final del término del contrato. En la siguiente tabla se deben listar cada uno de los requerimientos a detalle.

**Lista de requerimientos funcionales**


body table tbody tr td

Figura 3 - Edición de componentes del patrón de contrato

Los patrones que se van creando en la herramienta son parte de los patrones que Quetzalcoatl ofrece a los usuarios finales, es decir, el usuario Estándar tiene acceso a consultar los patrones que están disponibles, y a partir de un patrón que seleccione puede crear su contrato, según sean sus necesidades

#### 2.4.2. Creación de un contrato a partir de un patrón

Quetzalcoatl ofrece un mecanismo de creación automatizada de contratos a partir de patrones disponibles. Esta funcionalidad está disponible para los usuarios Estándar.

Para ejemplificar el proceso de creación de un contrato se realizó la creación de un contrato tomando como base el patrón creado en la sección 2.4.1. En ese sentido, en esta sección se creó un contrato para el desarrollo de un sitio Web informativo para una farmacia llamada LouSan.

Antes del proceso de desarrollo del contrato, es necesario conocer la definición de un contrato en el marco del uso de la herramienta. En este contexto, un contrato se define como un conjunto de cláusulas y componentes clasificados en categorías, con el objetivo de ayudar a que las Pymes y/o particulares conozcan cuales son los elementos que un contrato, dentro del área de las TI, debe incluir de manera mínima. Cada contrato creado en la herramienta se basa en el patrón que el usuario seleccione de acuerdo a sus necesidades particulares.

El proceso de creación de un contrato se centra en 4 sencillos pasos:

**Paso 1.-** Buscar y seleccionar el contrato indicado: En este paso el usuario Estándar realiza la búsqueda de contratos de acuerdo a palabras relacionadas con sus necesidades particulares.

**Paso 2.-** Capturar los datos generales del contrato: En este paso el usuario Estándar captura información relacionada con los representantes por parte del cliente y proveedor.

**Paso 3.-** Editar elementos de contrato: Una vez capturados los datos generales, se procede a editar cada uno de los componentes proporcionados por el contrato que se ha seleccionado en el paso 1, es decir, se llenan las plantillas que están definidas en cada uno de los componentes del contrato.

**Paso 4.-** Generar contrato final: Como paso final, se realiza la generación del contrato en formato PDF y se guarda una copia completa del contrato que se ha creado.

En la figura 4 se muestra como se realiza la edición de los componentes del contrato, en la parte superior de la imagen se muestra el componente tal como aparece en el contrato de manera general y en la parte inferior se muestra el componente cargado en el editor para llenar la plantilla sugerida y/o agregar elementos según las necesidades del usuario.

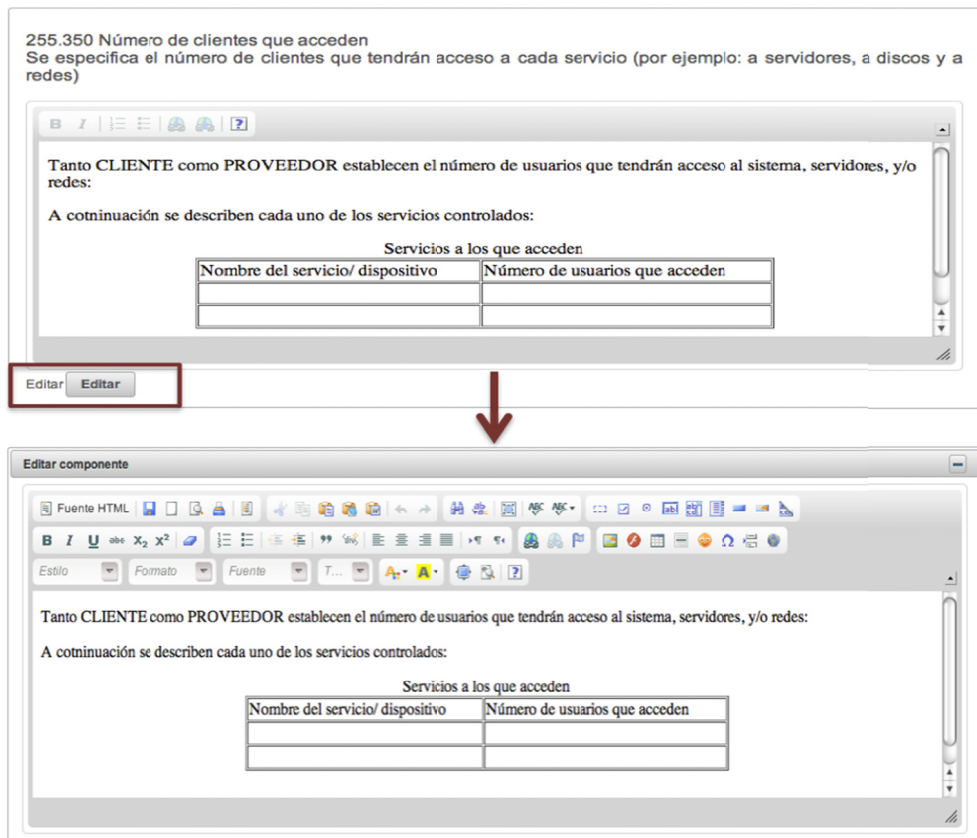


Figura 4 - Edición de los componentes del contrato

Después de editar cada uno de los componentes del contrato, el último paso es generar el documento final del contrato en PDF como se observa en la Figura 5. Este documento es el producto final que el usuario obtiene al concluir el proceso de creación del contrato.

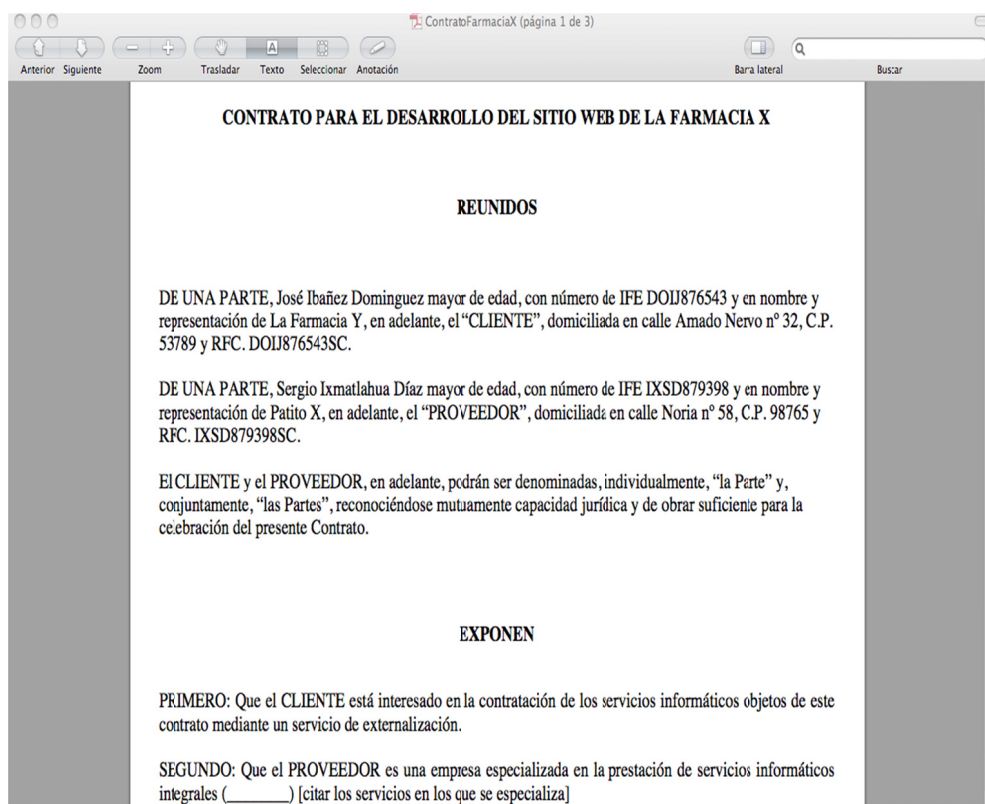


Figura 5 - Contrato generado en formato PDF

### 2.4.3. Evaluación de contratos

Por último, *Quetzalcoatl* también ofrece un mecanismo de evaluación de contratos. Este mecanismo ayuda a los usuarios a medir los porcentajes de rendimiento de sus contratos, de manera que puedan evaluar los contratos creados desde la herramienta misma (internos) y/o los contratos físicos (externos) que ellos deseen evaluar, con el objetivo de tomar la mejor decisión a la hora de contratar u ofrecer los servicios de TI.

Para realizar la evaluación tanto de los contratos internos como externos, *Quetzalcoatl* proporciona un panel para administrar los contratos del usuario. Los contratos internos se consideran como los contratos que fueron creados dentro de la herramienta y por lo tanto se pueden evaluar directamente. Los contratos externos, para ser evaluados, primeramente se deben registrar en la herramienta, posteriormente se realiza un análisis manual del contrato físico para identificar los componentes de acuerdo al modelo de contrato de la herramienta y por último se registran en la herramienta los componentes encontrados durante el análisis.

En la figura 6 se muestra el panel de administración de los contratos internos y externos, es aquí de dónde se parte para realizar la evaluación de los contratos. De lado izquierdo, se muestran los contratos Internos y de lado derecho, los contratos externos.

Para evaluar un contrato basta con seleccionar y elegir el botón “Eval. contrato” para los contratos internos; para evaluar un contrato externo se selecciona el contrato y se da click en el botón “Reg. componentes”.



Figura 6 – Panel de administración de contratos

Para el caso de la evaluación de contratos internos, se evaluó el contrato creado en la sección 2.4.3, para ello se selecciona el contrato a evaluar, el sistema carga los datos de contrato y muestra un resumen de los porcentajes del contrato. Dicho resumen, se hace con base al modelo de contrato integrado en *Quetzalcoatl*, es decir se realiza la comparación entre qué componentes tiene el contrato elegido y el modelo de contrato, de esta manera se obtienen las gráficas de rendimiento después de realizar la evaluación con el objetivo de ayudar al usuario a tomar la mejor decisión para la adquisición o venta de servicios de TI.

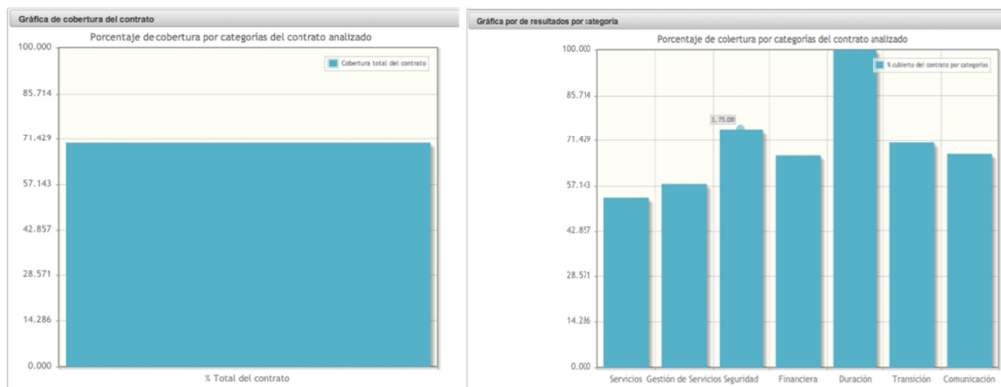


Figura 7 – Gráficas de rendimiento por contrato y por categorías

*Quetzalcoatl* ofrece varios tipos de gráficas por porcentaje del contrato total, por categorías del contrato, y por cláusulas del contrato. En la Figura 7 se observa la gráfica por contrato y categorías, resultante de la evaluación del contrato creado en la sección 2.4.2. Que en este caso el contrato obtuvo un porcentaje del 70% aproximadamente, y por categoría obtuvo: Servicios – 53.34%; Gestión de servicios – 57.66%; Seguridad – 75%; Financiera – 66.65%; Duración – 100%; Transición – 70%; Comunicación –

67.02%. El resultado es considerable puesto que el porcentaje del contrato cubre un 70% de las cláusulas y componentes que el modelo de contrato ofrece, por lo que la decisión final de adquirir o vender un servicio con este contrato queda a criterio del usuario.

### 3. Conclusiones

Hoy en día la práctica del *Outsourcing* de recursos de TI, sin duda alguna es una de las actividades con mayor demanda en el mercado local, nacional e internacional, siendo esta una de las tantas razones para conocer qué elementos (entre cláusulas y componentes) se deben considerar dentro de un contrato a la hora de establecer una relación entre clientes y proveedores. En este sentido, el uso de herramientas Software que apoyen a las Pymes y particulares que adquieren u ofrecen servicios de TI, es una buena alternativa para ayudar a establecer las relaciones entre las partes involucradas. Dicho lo anterior, *Quetzalcoatl* se presenta como una herramienta innovadora que ayuda a las Pymes y particulares, que adquieren u ofrecen servicios de TI, a poder crear, gestionar y evaluar sus contratos, tanto internos como externos, con e objetivo de apoyar a la toma de decisiones a la hora de adquirir o vender servicios de TI.

### Referencias

- Gallagher, B. P., Phillips M., et at. (2007) .CMMI for Acquisition, Version 1.2.
- International Data Corporation. (2010) “Visión de la Industria en el Segundo Semestre”.
- Jacobson, I., Booch, G., Rumbaugh, J. (2000). El proceso unificado de desarrollo de software. Rational Software Corporation. Pearson Addison Wesley. Madrid.
- Jae-Nam, L., Minh Q. H. (2007). IT outsourcing evolution: past, present, and future. Communications ACM 46.
- Matthew K. O. L. (1996). IT Outsourcing Contracts: Practical Issues for Management. Information Systems Department City University of Hong KongKowloon. Hong Kong.
- Mejía Miranda, J. (2010). Marco para el gobierno de la externalización del proceso de desarrollo del software a través de la Ingeniería del Contrato. Cap. 4., Universidad Politécnica de Madrid, Madrid, España.
- Tahuítón Mora, J. (2011). Arquitectura de software para aplicaciones Web. Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional. México, D.F.