



ESCUELA DE INGENIERÍA EN SISTEMAS

Tema:

Desarrollo del Módulo de Anexo Transaccional Simplificado
y su integración al ERP Adempiere.

**Proyecto de Investigación previo a la obtención del
título de Ingeniera en Sistemas y Computación**

Líneas de Investigación:

Sistemas de información y/o nuevas tecnologías de la
información y comunicación y sus aplicaciones.

Autora:

Mayra Alejandra Coronel Balseca

Director:

Ing. Mg. Ricardo Patricio Medina Chicaiza

Ambato - Ecuador

Mayo 2016

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

SEDE AMBATO

HOJA DE APROBACIÓN

Tema:

“Desarrollo del Módulo de Anexo Transaccional
Simplificado y su integración al ERP Adempiere”

Línea de Investigación:

Sistemas de información y/o nuevas tecnologías de la
información y comunicación y sus aplicaciones.

Autora:

MAYRA ALEJANDRA CORONEL BALSECA

Ricardo Patricio Medina Chicaiza, Mg. f. _____

CALIFICADOR

Darío Javier Robayo Jácome, Mg. f. _____

CALIFICADOR

Galo Mauricio López Sevilla, Mg. f. _____

CALIFICADOR

Teresa Milena Freire Aillón, Mg. f. _____

DIRECTORA DE LA ESCUELA DE INGENIERIA EN SISTEMAS

Hugo Rogelio Altamirano Villarroel, Dr. f. _____

SECRETARIO GENERAL PUCESA

Ambato - Ecuador

Mayo 2016

DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD

Yo, Mayra Alejandra Coronel Balseca portadora de la cedula de ciudadanía N° 092174943-8 declaro que los resultados obtenidos en la investigación que presento como informe final, previa a la obtención del título de Ingeniera en Sistemas y Computación son absolutamente originales, auténticos y personales.

En tal virtud, declaro que el contenido, las conclusiones y los efectos legales y académicos que se desprenden del trabajo propuesto de investigación y luego de la redacción de este documento son y serán de mi sola y exclusiva responsabilidad académica.

Mayra Alejandra Coronel Balseca

C.I. 092174943-8

AGRADECIMIENTO

A Dios por llevarme de su mano,
A mí amada Familia por todo su amor,
A Sofy y Adri por su apoyo incondicional,
A mis jefes por la apertura y apoyo en lo laboral,
A los docentes de la universidad por sus conocimientos.

Y en especial a ella mi segunda madre y abuela Fanny M.

Mayra Coronel.

DEDICATORIA

A Dios,
Mi Esposo,
Mis Papis,
Mi Abuela,
Mis Niños,
Mi Familia.

Mayra Coronel.

RESUMEN

El presente proyecto de investigación tiene como finalidad desarrollar el módulo de Anexo Transaccional Simplificado y su integración en el manejador de recursos empresariales en código abierto Adempiere, cuyo objetivo principal es apoyar y agilizar la labor que el usuario responsable del área dentro de la empresa realiza durante las declaraciones mensuales, dicho proceso se pretende llevar a cabo mediante la extracción y consolidación de datos recaudados de los distintos módulos que se manejan dentro del ERP, los cuales podrán ser revisados y modificados en el módulo previo procesamiento final, el cual dará lugar al archivo XML deseado. El desarrollo se llevó a cabo bajo las siguientes características: Sistema Operativo Windows, Base de Datos Oracle XE versión 11g de uso y distribución gratuita, Lenguaje de programación Java y Pl/Sql, asimismo se encuentra sustentado bajo la metodología de desarrollo ágil "Extreme Programming" la cual posee como característica fundamental su interacción directa con el cliente así como también una rápida respuesta de desarrollo ante cambios solicitados en sistemas que se encuentran en funcionamiento.

ABSTRACT

The aim of this study is to develop the module of simplified transaction spreadsheet and its integration into the corporate resource handler in Adempiere open code whose main objective is to support and set into motion the work that the user in charge of the department of the company carries out during monthly declarations. This process is intended to be carried out through the extraction and consolidation of data collected from the different modules that are handled within the ERP, which will be able to be revised and modified in the module prior to final processing and generate the desired XML file. The development was accomplished under the following characteristics: Windows Operating System, Oracle Database XE version 11g of use and free distribution as well as the programming language of Java and Pl/Sql. In addition, it is based on Extreme Programming, the methodology for flexible development, whose fundamental feature is its direct interaction with the client as well as a quick development response in light of requested changes in systems which are in operation.

ÍNDICE DE CONTENIDOS

PRELIMINARES

DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD.....	iii
AGRADECIMIENTO.....	iv
DEDICATORIA.....	v
RESUMEN.....	vi
ABSTRACT.....	vii
ÍNDICE DE CONTENIDOS.....	viii
ÍNDICE DE GRÁFICOS.....	xii
ILUSTRACIONES.....	xii
INTRODUCCIÓN.....	2
CAPÍTULO I.....	3
1.1 Antecedentes.....	3
1.2 Problema.....	5
1.2.1 Descripción del Problema.....	5
1.2.2 Preguntas Básicas.....	5
1.3 Justificación.....	6
1.4 Objetivos.....	7
1.4.1 Objetivo General.....	7
1.4.2 Objetivos Específicos.....	7
1.5 Meta.....	8

1.6	Fundamentos Teóricos.....	8
1.6.1	Clasificación de Contribuyente en Ecuador.....	8
1.6.2	Deberes Tributarios en Ecuador.....	11
1.6.3	Anexo Transaccional Simplificado en Empresas.	13
1.6.4	Planeación de Recursos Empresariales (ERP)...	16
1.6.5	Software Libre.....	18
1.6.6	Base de Datos.....	22
1.6.7	Oracle.....	25
1.6.8	Java.....	26
1.6.9	Metodologías de Desarrollo e Implementación..	28
1.6.10	Metodología de Desarrollo XP.....	30
	CAPÍTULO II.....	32
2.1	Metodología de Investigación.....	32
2.1.1	Método General.....	32
2.1.2	Método Específico.....	32
2.2	Metodología de Desarrollo.....	33
2.2.1	Fase de Planeación.....	34
2.2.2	Fase de Diseño.....	37
2.2.3	Fase de Codificación.....	38
2.2.4	Fase de Adaptación y Prueba.....	40
	CAPÍTULO III.....	43

3.1	Planificación de Requisitos.....	43
3.1.1	Historias de Usuario.....	43
3.1.2	Plan de Entregas.....	46
3.1.3	Velocidad del Proyecto.....	47
3.1.4	Plan de Iteraciones.....	47
3.2	Diseño.....	49
3.2.1	Metáfora del Sistema.....	49
3.2.2	Tarjetas C.R.C.....	50
3.2.3	Diseño Simple Módulo ATS.....	51
3.2.4	Diagrama de Clases.....	56
3.3	Codificación.....	57
3.3.1	Disponibilidad del Cliente.....	57
3.3.2	Estándares de Programación.....	57
3.3.3	Programación por Parejas.....	58
3.3.4	Integración Continua.....	58
3.4	Pruebas.....	67
3.4.1	Pruebas Unitarias.....	67
3.4.2	Pruebas de Aceptación.....	74
CAPÍTULO IV.....		78
4.1	Análisis de Resultados.....	78
4.1.1	Resultado de la Implementación.....	78

4.2 Validación de Resultados.....	86
CONCLUSIONES Y RECOMENDACIONES	87
Conclusiones	87
Recomendaciones	88
BIBLIOGRAFIA	89

ÍNDICE DE GRÁFICOS

ILUSTRACIONES

Ilustración 1.1 Niveles de la Base de Datos.....	24
Ilustración 1.2 Java.....	27
Ilustración 1.3 Proceso de Edición y Compilación.....	28
Ilustración 2.1 Fases Metodología XP.....	33
Ilustración 3.1 Pestaña de Datos de empresa.....	53
Ilustración 3.2 Pestaña de Compras.....	53
Ilustración 3.3 Pestaña Forma de Pago.....	54
Ilustración 3.4 Pestaña de Retenciones en Compra.....	55
Ilustración 3.5 Pestaña Ventas.....	55
Ilustración 3.6 Documentos Anulados.....	56
Ilustración 3.7 Diagrama de Clase.....	56
Ilustración 3.8 Prueba de Caja Blanca - Compras.....	68
Ilustración 3.9 Prueba de Caja Blanca - Anuladas.....	69
Ilustración 3.10 Prueba de Caja Blanca - Ventas.....	70
Ilustración 3.11 Prueba de Caja Blanca - Pagos.....	72
Ilustración 3.12 Prueba de Caja Blanca - Retenciones..	73
Ilustración 4.1 Ingreso Valido al ERP.....	78
Ilustración 4.2 Ingreso Inválido al ERP.....	79
Ilustración 4.3 Asignación de Acceso al módulo por Rol	79
Ilustración 4.4 Menú con Acceso Módulo.....	80
Ilustración 4.5 Datos Empresa Guardado Validado.....	80
Ilustración 4.6 Datos Maestros Ingreso Inválido.....	81

Ilustración 4.7 Generar Datos para ATS.....	81
Ilustración 4.8 Datos Compras- Rejilla.....	82
Ilustración 4.9 Datos Forma de Pago.....	82
Ilustración 4.10 Datos de Retenciones.....	83
Ilustración 4.11 Datos de Ventas.....	83
Ilustración 4.12 Datos de Documentos Anulados.....	83
Ilustración 4.13 Configuración Campo Genera XML.....	84
Ilustración 4.14 Generar XML.....	84
Ilustración 4.15 Configuración de Directorio XML.....	85
Ilustración 4.16 Verificación de XML generado.....	85

TABLAS

Tabla 3.1 Esquema historia de usuario.....	43
Tabla 3.2 Historia de usuario I.....	44
Tabla 3.3 Historia de usuario II.....	45
Tabla 3.4 Historia de usuario III.....	45
Tabla 3.5 Historia de usuario IV.....	46
Tabla 3.6 Plan de Entrega.....	46
Tabla 3.7 Plan de Iteración.....	47
Tabla 3.8 Esquema Tarjetas C.R.C.....	50
Tabla 3.9 CRC - Vista Compras.....	50
Tabla 3.10 CRC - Vista Formas de Pago.....	50
Tabla 3.11 CRC - Vista Retenciones.....	51
Tabla 3.12 CRC - Vista Anuladas.....	51
Tabla 3.13 CRC - Vista Ventas.....	51

Tabla 3.14 Esquema de Pruebas de Aceptación.....	74
Tabla 3.15 Pruebas de Aceptación 1.....	75
Tabla 3.16 Pruebas de Aceptación 2.....	76
Tabla 3.17 Pruebas de Aceptación 3.....	76
Tabla 3.18 Pruebas de Aceptación 4.....	77

INTRODUCCIÓN

El presente proyecto de investigación sobre el "Desarrollo del Módulo de Anexo Transaccional Simplificado y su integración al ERP Adempiere".

Dentro del primer capítulo las empresas que manejan el sistema Adempiere detallan la necesidad que poseen en contar con el modulo anexo transaccional simplificado integrado al sistema, para de esta manera simplificar la revisión y declaración de datos a entres gubernamentales evitándose manejar herramientas externas que puedan vulnerar la calidad de información, de esta manera se justifica los motivos por el cual se propone llevar a cabo la solución al presente problema, como punto final se fundamenta teórica y metodológicamente los temas a abordar dentro del desarrollo.

Acto seguido en el segundo capítulo se estudia la metodología Extreme Programming mediante sus fases de planificación, diseño, implementación y pruebas.

Por otro lado el tercer capítulo detalla el desarrollo llevado a cabo dentro de la metodología estudiada con anterioridad, debido a la importancia de la mencionada

documentación se crea respaldo del proceso mismo que va queda plasmada a medida que avanza el capítulo, una vez culminada esta etapa se pasa al análisis de resultados.

La validación y análisis de resultados lo encontramos dentro del capítulo IV, en el cual queda demostrado el resultado final, las funcionalidades y beneficios que presenta al usuario final, mismas que fueron verificadas acorde avanzaba el desarrollo y validadas al finalizar el mismo.

Como punto final se encuentra las respectivas conclusiones y recomendaciones que deja el proceso de este proyecto.

CAPÍTULO I

1. Fundamentos Teóricos

1.1 Antecedentes

En esta sección se analizará los distintos aspectos que intervienen en las empresas se considera la importancia de un Planeador de Recursos Empresariales (ERP), Contabilidad, Anexos transaccionales solicitados por entes gubernamentales, debido a esto se detalla lo siguiente.

En el artículo encontrado en Ramos, (2015) menciona que la actual evolución y crecimiento dentro del área operacional de las empresas, además de la desmedida demanda de este tipo de actividades y por sus consecuencias en la sociedad, últimamente ha sido creciente los requerimientos en cuanto a información; y como efecto las demandas a nivel internacional. Para cubrir esos intereses y, poder facilitar el análisis de los estados financieros emitidos en países que poseen diferentes normativas, se ha empezado desde hace varios años atrás un proceso de concordancia contable de carácter irreversible.

Por lo referido en las disposiciones que se mencionan en la web Servicio de Rentas Internas, (2015) o más conocido como SRI las personas que llevan contabilidad están obligadas a declarar un detalle de movimientos realizados

referentes a las adquisiciones, retenciones de impuesto al valor agregado y renta, forma de pago, ventas y su total dentro de dicho mes.

Por lo encontrado en la web mencionada se debe considerar que, si es un contribuyente obligado a llevar contabilidad y no posee ningún tipo de transacción dentro del mes, no debe presentar la declaración.

El manejo de una óptima contabilidad en las empresas del entorno es lo que define la posterior toma de decisiones sobre los correctivos o mejoras que se llevarán a cabo para mantenerlas en la cima o llevarlas a la quiebra.

Adempiere ERP es considerada una completa solución de negocios la cual cuenta con módulos como compras, ventas, inventario, bancos, contabilidad, manufactura desarrollada en Java en base a software libre y compatible con base de datos OpenSource y propietario que día a día toma poder en el mercado ERP a nivel mundial.

A través de la experiencia laboral obtenida en las distintas empresas que manejan el ERP Adempiere, se ha observado que no cuentan con el módulo Anexo Transaccional Simplificado (ATS) requerido en Ecuador, lo cual ocasiona que el personal encargado de la empresa tengan una gran carga laboral manual cada vez que tienen que declarar al

ente gubernamental sus transacciones, por otra parte de se debe tomar en cuenta que en el Ecuador no se explota de manera adecuada todo lo referente a desarrollo en software libre, se apoyará el análisis financiero económico de las empresas, además se pretende brindar información actualizada para mejorar la competitividad, por este motivo se procederá a Desarrollar del Módulo de Anexo Transaccional Simplificado y su integración al ERP Adempiere.

1.2 Problema

1.2.1 Descripción del Problema

✓ Espacial

El módulo propuesto será desarrollado de tal forma que pueda ser integrado en el ERP Adempiere para su posterior implementación en la empresa piloto Corpocalza.

✓ Temporal

El proyecto propuesto se desarrollará durante el transcurso de los meses de Octubre 2015 y Abril del 2016.

1.2.2 Preguntas Básicas

✓ ¿Por qué se origina?

Lo genera la alta carga operativa manual que genera el ingreso de todas las transacciones realizadas dentro del mes a reportar.

✓ **¿Qué lo origina?**

La no existencia de un módulo Anexo Transaccional Simplificado en el ERP Adempiere.

✓ **¿Dónde se origina?**

En las empresas que manejan el ERP Adempiere.

1.3 Justificación

El manejo de información a tiempo y de forma adecuada es un tema primordial en el diario vivir de las empresas, las malas prácticas o retrasos en el flujo de procesos puede acarrear repercusiones o decisiones negativas por parte de las gerencias, por lo que con el desarrollo del módulo de Anexo Transaccional Simplificado y la implementación y adaptación del ERP en software libre bajo normas contables vigentes en Ecuador se automatizará y apoyará el manejo de información en las empresas con un mejor desempeño y crecimiento de las mismas.

El proyecto proporciona control, eficiencia y calidad además de brindar apoyo en cuanto a la carga operativa manual al personal encargado, y como resultado beneficios, puntualidad y mayor rapidez de análisis al área contable al declarar sus transacciones a las respectivas entidades

gubernamentales, manteniéndolas así al nivel competitivo del mercado Ecuatoriano.

Por lo cual cabe destacar que la importancia del presente proyecto radica en la finalidad de aportar en una considerable reducción de tiempo y facilidad en el análisis de datos por parte del área contable que es la encargada de declarar dicha información a entes gubernamentales de manera mensual ya que de momento las empresas que manejan Adempiere hacen doble trabajo al ingresar forma manual al sistema de anexos transaccionales facilitado por el Servicio de Rentas Internas (SRI).

1.4 Objetivos

1.4.1 Objetivo General

- ✓ Desarrollar el módulo Anexo Transaccional Simplificado e integrar en el ERP Adempiere.

1.4.2 Objetivos Específicos

- ✓ Cimentar metodológica y teóricamente sobre el Anexo Transaccional Simplificado del SRI y el ERP Adempiere.
- ✓ Diseño del módulo Anexo transaccional Simplificado.
- ✓ Integrar el módulo Anexo Transaccional Simplificado en el ERP Adempiere.

1.5 Meta

Con el desarrollo del módulo Anexo Transaccional Simplificado y su integración en el ERP Adempiere, se apoyará el trabajo operativo manual que en la actualidad maneja las empresas que poseen el ERP.

1.6 Fundamentos Teóricos

1.6.1 Clasificación de Contribuyente en Ecuador

Según se detalla en Servicio de Rentas Internas, (2015) la clasificación acorde a los contribuyentes es:

✓ **Personas naturales régimen general**, son aquellos individuos sean ellos ecuatorianos o foráneos cuyos movimientos económicos son legítimos.

Se debe notar también que dichas entidades acorde al Servicio de Rentas Internas, (2015) deben:

- ❖ Obtener el Registro Único de Contribuyente.
- ❖ Formular recibos de venta previamente acreditados.

Además de mencionarse que las personas naturales se dividen según la web del Servicio de Rentas Internas, (2015) en:

- ❖ **Con obligatoriedad de contabilidad** todas aquellas personas que al iniciar sus diligencias de negocio

cuenten con sustento monetario propio, o que a su vez al primer día del inicio de año las transacciones generadas superen nueve fragmentos básicos unificados tomado en consideración que los mismos son libres de impuestos o a su vez sus totales netos anuales generadas de sus transacciones no hayan sido mayores a quince fragmentos básicos y por último que los gastos de la entidad afectados por el giro del negocio no hayan sido mayores a doce fragmentos básicos, como detalle final se debe contar con la firma de un contador autorizado o CPA con lo cual queda un sello de responsabilidad.

❖ **No obligadas a llevar contabilidad** son todos aquellos que no entran en el grupo anterior, esto no significa que deban obviar llevar un control sobre toda transacción es decir de todo lo que sale e ingresa al negocio.

✓ **Personas naturales con régimen simplificado (RISE)**

El régimen impositivo simplificado cuya adhesión al mismo es voluntario, tiene como objetivo reemplazar la cancelación del IVA y de la renta mediante cuotas mensuales proponiendo optimizar la forma de tributación en el país.

Entre los beneficios que desataca el Servicio de Rentas Internas, (2015) encontramos:

- ❖ No necesita hacer declaraciones.
- ❖ No es sujeto de retención de impuestos.
- ❖ Emite comprobante de comercio con valor monetario.
- ❖ No posee obligatoriedad de contabilidad.
- ❖ Por cada nuevo personal, se podrá pedir un reducción del cinco por ciento de su pago.

✓ **Sociedades**

Conocidas también como entes jurídicas cuyas transacciones monetarias son legalmente refugiadas bajo una representación legal autónoma. Dichas entidades se encuentran divididas en privadas y públicas, acorde lo menciona el escrito de creación.

Acorde la información encontrada en Servicio de Rentas Internas, (2015) de forma obligatoria deben:

- ❖ Sacar el Registro Único de Contribuyente.
- ❖ Generar documentos de venta acreditados.
- ❖ Presentar las declaraciones respectivas acorde su categoría de la clasificación nacional de Actividad Económica (CIIU) y bajo firma de un contador.

- ❖ Dichas declaraciones deben ser entregadas según la tabla que se encuentra expuesta en el sitio web del Servicio de Rentas Internas (SRI).

1.6.2 Deberes Tributarios en Ecuador

Entre las obligaciones tributarias que se deben considerar dentro del país encontramos que el Servicio de Rentas Internas, (2015) destaca las siguientes:

✓ Adquirir el registro único de contribuyentes (RUC)

Es el número de identidad de los entes cuya gestión financiera es legal. El registro único de contribuyente posee una estructura que es certificada por el SRI y por otros entes para su utilización en diferentes áreas.

✓ Las declaraciones a manejar son:

- ❖ **Declaración de impuesto al valor agregado (IVA):** Se realiza mes a mes, inclusive cuando no se haya generado ventas dentro del mes, no hayan compras o no existan retenciones en la renta.
- ❖ **Declaración del impuesto a la renta:** llevada a cabo cada año, mediante la consignación de valores monetarios que corresponden al balance general, estado de ganancias y pérdidas y la concordancia tributaria.
- ❖ **Pago del anticipo por impuesto a la renta:** dentro de este proceso se recomienda la lectura de la ley orgánica de régimen tributario interno.

❖ **Declaración del impuesto a los consumos especiales:** tan solo las sociedades que traspasen patrimonios o brinden servicios con IVA están obligadas a realizar esta declaración.

✓ **Presentar anexos**

Corresponde a los datos presentados de forma minuciosa de las transacciones que efectúa el contribuyente las cuales están forzadas a ser presentadas a través del sistema de declaraciones, en la fecha indicada acorde la tabla desplegada en la página web del SRI.

Los anexos a considerar son:

❖ **Gastos Personales** los entes cuyos gastos del año anterior a la declaración superen un cincuenta por ciento referente al sueldo básico fuera de impuestos. Se podrá hacer deducción los gastos generados por sus familiares inmediatos que dependan económicamente de la persona que se encuentra declarando.

❖ **Retenciones en la fuente del impuesto a la renta por otros conceptos también conocido como (REOC)** Listado del mes sobre la información referente a adquisiciones ligadas a los documentos de ventas y retenciones de renta, en cuyo mes no exista transacciones no es necesario declararlo.

❖ **Transaccional simplificado (ATS)** es el detalle de los datos relacionados a adquisiciones, ventas, exportaciones, documentos anulados y retenciones en general con respecto a un mes determinado, el mismo que debe reportado a mes próximo. Los entes obligados a presentar este anexo son las sociedades catalogadas como especiales o que a su vez posean una autorización de auto impresión de documentos de venta, retención y complementarios.

Se debe mencionar que la declaración del anexo transaccional simplificado reemplaza la obligatoriedad del REOC.

❖ **De impuesto a la renta por relación de dependencia (RDEP)** concerniente a los datos relacionados con las retenciones en renta llevadas a cabo a los empleados que se encuentran en figura de relación de dependencia debido a su remuneración en las fechas concernientes entre el 1 de enero y el 31 de diciembre.

❖ **De ICE** referente al reporte de cuantía en egresos por artículo.

1.6.3 Anexo Transaccional Simplificado en Empresas.

En la web Servicio de Rentas Internas, (2015) indica que los sujetos tributarios deben declarar un informe

minucioso de los movimientos correspondientes a adquisiciones, ventas, retenciones IVA / Renta, forma de pago y total de ventas dentro de dicho mes.

Por lo encontrado en la web mencionada se debe considerar que, en caso de no crear alguna transacción dentro del mes, no se debe presentar dicha declaración de anexo.

Los datos solicitados por el SRI deberán ser entregados mediante un documento de extensión XML y vía web con la estructura acorde los detalles de la tarjeta técnica valedera en la página web del SRI.

✓ **Contabilidad y las empresas**

El autor Wandeem-Bergher Lozano, (2012) cataloga a la contabilidad como ciencia económica que mediante procesos puede identificar, procesar y transmitir información económica apta para emitir juicios y poder tomar las respectivas decisiones en las distintas áreas de donde proviene la información.

En lo que concierne a Rodriguez Rocha, (2009) sustenta la contabilidad como ciencia en la cual a través de técnicas, normas y procedimientos aplicadas a los datos e información de la empresa permite pasar a la etapa de análisis, orden y registro de las transacciones realizadas por las empresas.

Se puede así definir a la contabilidad como ciencia económica que agrupa técnicas para elaborar procesos que permitiendo identificar de forma ordenada las operaciones realizadas en las empresas, permite así la toma de decisiones ya sea a nivel interno o externo para mantener la empresa a la vanguardia.

Acorde a lo mencionado por Horgren, (2010) divide a la contabilidad en Financiera y Administrativa, Además acota que la contabilidad financiera es aquella que proporciona los datos ideales para los agentes externos como los inversionistas o prestamistas. En cambio comenta sobre la contabilidad administrativa que esta se enfoca en brindar información para los jefes o gerencias internas de la empresa.

Por otro lado Wandeen-Bergher Lozano, (2012) considera a la contabilidad financiera como objeto de clasificaciones según sea la perspectiva de estudio. Las decisiones de mayor trascendencia dentro de las empresas están relacionadas con la gestión a la que están orientadas y su largo y difícil camino hacia el éxito.

Se concluye que las principales diferencias entre contabilidad financiera y administrativa es su enfoque a nivel interno y externo, cabe tomar en cuenta también que la contabilidad financiera entrega como resultado final

estados financieros de uso general enfocándose en el pasado o datos históricos de las empresas, en cambio que, la contabilidad administrativa entrega informes según como los requiera la administración y al contrario de la financiera se enfoca en el futuro de la empresas.

✓ **Estados Financieros**

Tennent, (2010) señala que todas las compañías necesitan un sistema para registrar las transacciones y plasmarlas de forma coherente, lo que es adecuado en términos globales; debido a sus necesidades el sistema contable debe contar con los siguientes informes financieros que permita el análisis y toma eficaz de decisiones.

- ❖ **Balance General:** Información breve que muestra los activos, pasivos y el patrimonio.

- ❖ **Estado de Resultado:** Informe que resume entradas y salidas incurridos en un determinado tiempo.

1.6.4 Planeación de Recursos Empresariales (ERP)

Por medio del documento electrónico recopilado del Centro de Ingeniería del Software e Ingeniería del Conocimiento, (2004) se define ERP como una aplicación informática que facilita la gestión de todos los procesos involucrados en el negocio de forma integrada. Por lo regular esta clase de software informático facilita datos relacionados en las

diferentes áreas del negocio. Este sistema debe ser configurado y adaptado de forma tal que pueda responder a los requerimientos puntuales en cada organización. Luego de que el planeador de recursos empresariales (ERP) se halle implementado permitirá representar los diferentes contextos y adquirir datos afianzados en fechas precisas.

Por su parte García Blanco & Morales Ramos, (2003, pág. 3) menciona que los elementos principales de un software dentro del manejo de datos son tres:

- ❖ **Recursos Físicos:** Son los documentos, equipos informáticos, de telecomunicaciones, que lo forman.
- ❖ **Recursos Humanos:** Son las personas y grupos que trabajan en él.
- ❖ **Reglas:** Son las normas y métodos que establecen como deben realizarse los distintos procesos.

✓ **Gestión Financiera y el Sistema Contable**

Para Tennent, (2010) la gestión financiera y el sistema contable debe contar con:

- ❖ **Funciones del directorio:** Personas contratadas o delegadas por los accionistas para ser quienes conduzcan sus inversiones.
- ❖ **El Rol financiero:** Los directores tienen la responsabilidad legal de garantizar que la firma mantenga los registros contables adecuados.

✓ **Software Ecuatoriano relacionado con las PYME**

El Ministerio de Comercio Exterior s.f., señala al software ecuatoriano con relación a las PYME como un caso especial dentro del área del software ya que lo analiza como un rápido crecimiento en el mercado ecuatoriano, da como ejemplo los distintos tipos de sistema que se manejan en el sector público que de alguna u otra manera facilitan o mejoran el manejo de información.

1.6.5 Software Libre

En el portal HispaLinux s.f., se define que en software libre se contempla que los beneficiarios de compilar, reproducir, cambiar, comercializar, aprender y optimizar el sistema.

Entre los alcances incluidos en las libertades que posee el usuario se encuentra la de poder distribuir copias que incluyen los ejecutables del software así como también su código fuente, sean estas en su estado original o a su vez se trate de versiones adaptadas al medio.

Cabe resaltar que software libre no tiene una connotación de software no comercial. Un programa libre puede encontrarse para uso, desarrollo y distribución comercial.

Por otra parte es importante comentar que también existe el software libre sin libre distribución de copias. En

HispaLinux, s.f., se considera que existen importantes razones por las que es mejor usar la libre distribución de copias.

✓ **Características de Software Libre**

Para Open Source Initiative, s.f., es primordial acotar que software libre no es tan solo poseer acceso al código fuente, sino que además:

- ❖ **Libre Redistribución:** La autorización no debe limitar a un usuario, además por la licencia no debe requerir un valor monetario así como comisiones para esta venta.
- ❖ **Código Fuente:** los distintos mecanismos por los cuales un desarrollador o programador puede modificar el software.
- ❖ **Trabajos derivados:** para contar con trabajos derivados dicha licencia debe otorgar permisos para modificarlos, además pueden ser distribuidos bajo términos de igualdad al primario.
- ❖ **Integridad del código fuente del autor:** este tipo de licencia podría solicitar que todo trabajo llevado a cabo a partir del original deba poseer su propio número de versión o nombre.
- ❖ **No discriminación contra personas o grupos:** dicha autorización no debe excluir a una o varias personas.

- ❖ **No discriminación en función a la finalidad perseguida:** no debe limitar en ningún área de utilización. Sea el caso de ejemplo, que pueda ser llevado a cabo en una escuela pero no pueda ser llevado a cabo en una tienda de abarrotes o una librería.
- ❖ **Distribución de la licencia:** todas las libertades relacionadas con el software en cuestión deberán ser derivados en cada distribución, si requerir ningún tipo de licencia anexa por las partes adaptadas o mejoradas.
- ❖ **Licencia no debe ser específica de un producto:** la licencia debe conservar en esencia todos los lineamientos del software original incluyendo las adaptaciones en el campo que haya sido requerido.
- ❖ **Licencia no debe restringir otro software:** no se debe limitar el licenciamiento ni funcionamiento de programas externos que hayan sido acoplados al software original.
- ❖ **Licencia debe ser tecnología neutral:** no debe existir disposición algún sobre la cual la licencia puede fundamentar en una tecnología individual o estilo de interfaz.

✓ **Importancia del Software Libre para un País**

En la consulta realizada en Secretaria Nacional de la Administración Pública, s.f., se dice que el uso de software libre facilita la soberanía tecnológica, impulsa la innovación y desarrollo en el país, reduce considerablemente el gasto del estado además de fortalecer el desarrollo local y facilitar la inclusión digital.

A nivel del sector público, el uso de software libre permite:

❖ **Autonomía Tecnológica:** Secretaria Nacional de la Administración Pública, s.f. indica que si se adopta software libre y la facilidad que éste ofrece en cuanto a acceso al código fuente, la gran mayoría de usuarios pasarán del consumismo al desarrollo de software. Esto se reduce a que se podrán acoplar los distintos programas a las necesidades específicas de las distintas áreas, y todas esas modificaciones deberán realizarse mediante los requisitos exigidos por el modelo del Software Libre.

❖ **Estandarización e Integración** acorde Secretaria Nacional de la Administración Pública, s.f. también denominados estándares abiertos. Benefician el acople de sistemas e intercambio o comparación de

información, de forma que se pueda garantizar la accesibilidad sin restricción por la ciudadanía.

❖ **Seguridad:** por lo encontrado en la Secretaria Nacional de la Administración Pública, s.f. el hacer públicos los códigos fuente del software ayuda de gran manera a la seguridad de los mismos. Si se utiliza software libre se puede saber qué se realiza realmente un programa, qué tipo de información y transacciones maneja y cómo lo realiza. Una buena seguridad debe basarse en la transparencia.

❖ **Independencia de proveedores:** acorde la Secretaria Nacional de la Administración Pública, s.f. dice que mediante la compra de software privado estará implícita una relación de dependencia directa con el fabricante, así como para obtener actualizaciones. En muchos casos, se depende del proveedor para actualizar las versiones aun cuando la empresa no lo desee o el sistema no lo requiera.

1.6.6 Base de Datos

De acuerdo a La Biblia MySQL, (2015, pág. 2), consiste en un instrumento que permite recabar, organizar y procesar datos de un grupo determinado de registros relacionados.

Por su parte en la tesis sustentada por Enriquez Villavicencio & Silva Corrales, (2015) se resalta la

importancia de la organización de una base de datos ya que la información almacenada se encuentra estructurada independientemente de la utilización de los mismos en tiempo real, también detallan que una base de datos puede ser local o distribuida, es decir que puede poseer acceso mono usuario o a su vez multiusuario a través de un red respectivamente.

✓ **Sistema de gestión de base de datos (SGBD)**

Para Sistemas Gestores de Bases de Datos, (2015), a los SGBD se los puede conceptualizar como un conjunto de datos relacionados, estructurados y organizados entre sí. Las comentadas aplicaciones permiten al desarrollador precisar, instaurar y conservar la base de datos y habilitar libertades a la misma. Una base de datos cuenta con lo siguiente:

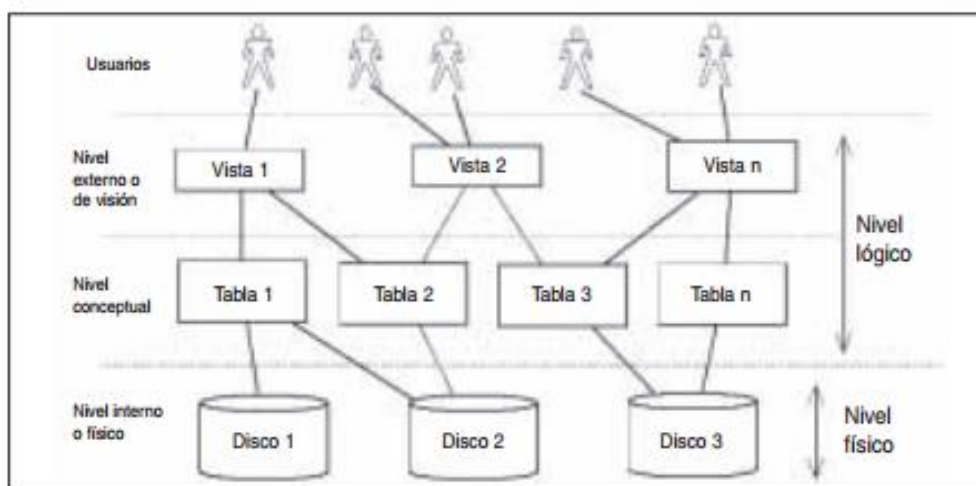
- ❖ **Creación y definición de la base de datos:** detalles de la organización, detalles sobre la información, las limitaciones e interacción entre los mismos a través de los lenguajes de esclarecimiento de datos.
- ❖ **Manipulación de los datos:** permite gestionar informes, inclusiones y reajuste de los mismos mediante un respectivo lenguaje de programación.
- ❖ **Acceso controlado a la base de datos:** mediante mecanismos de seguridad de acceso.

- ❖ **Mantener Integridad y consistencia:** de esta manera se logra prevenir que la información sea perjudicada por modificaciones no contempladas o programados.
- ❖ **Acceso compartido a la Base de Datos:** permite el control entre la interacción de beneficiarios oyentes.
- ❖ **Mecanismos de respaldo y recuperación:** logra restituir los datos a partir de un respaldo previo en caso de fallos en el sistema.

✓ **Diseño de los sistemas de base de datos**

De acuerdo a Sistemas Gestores de Bases de Datos, (2015) existen 3 niveles de BD:

Ilustración 1.1 Niveles de la Base de Datos



Fuente: Sistemas Gestores de Bases de Datos, 2015.

- ❖ **Nivel interno o físico:** permite almacenar datos físicos, se hace hincapié en el almacenamiento de datos tal cual se encuentran en el computador.

Representa la forma física de la BD a través de un bosquejo interno.

- ❖ **Nivel conceptual:** trata sobre los datos que van a ser utilizados, detalla también la organización para el acceso de los usuarios a la base de datos utilizando una representación conceptual. Detalla las entidades, propiedades, relaciones, procedimientos y limitaciones para los usuarios, mientras oculta los pormenores de las distribuciones físicas de almacenamiento.
- ❖ **Nivel externo o de visión:** detalla los distintos bosquejos externos o percepciones del usuario final. Cada representación contiene la zona de la base de datos que necesita un determinado conjunto de usuarios.

1.6.7 Oracle

Acorde con el autor Matsukawa Maeda, (2006), Oracle es un instrumento que nos permite optimizar el manejo de procesos no contemplados por los ordenadores, mediante la sincronización de almacenamiento sin tener que crear el acceso a la red por parte del usuario. Se destaca además que es conocido como el primer servidor de base de datos diseñado para un entorno de desarrollo *grid computing*, en la actualidad se cuentan con millones de equipos

informáticos conectados a internet. Tomado en cuenta esto, muchos de estos equipos no son utilizados de forma continua sino tan solo por cortos intervalos de tiempo, y como resultado se obtiene tiempos muertos durante la conexión.

1.6.8 Java

En lo dicho por Guevara, (2015):

✓ Lenguaje de Programación

El lenguaje de programación es una forma artificial en la cual los programadores pueden dar instrucciones entendibles para el computador por medio de las cuales se expresan cálculos y procedimientos para conseguir un determinado resultado, se resalta además que por cada lenguaje de programación existe grupo de palabras reservadas, así como también símbolos, normas sintácticas y semánticas que permiten especificar estructuras.

El proceso por el cual pasa un lenguaje de programación es primero la escritura del mismo, luego la compilación y por último la verificación del correcto funcionamiento del sistema.

✓ Historia

Java es un lenguaje desarrollado por Sun Microsystems, fue presentado a mediados de 1995 y desde aquel tiempo se ha

vuelto un lenguaje de uso mundial debido que es gratuito además de poder ser ejecutado en distintas plataformas se detalla en la Ilustración 1.2

La portabilidad de Java ha permitido que gran cantidad de empresas haya desarrollado sus sistemas basados en este lenguaje, ya que es menos costoso además de beneficiarse que el lenguaje es multiplataforma.

Ilustración 1.2 Java

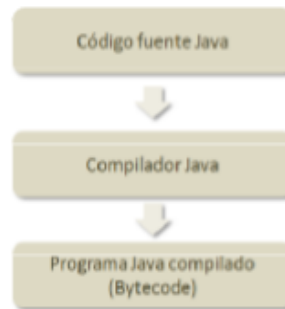


Fuente: Ladrón de Guevara Jorge Martínez, 2015.

✓ **Proceso de edición y compilación**

La estructura que Java sigue para compilar es la mostrada a continuación.

Ilustración 1.3 Proceso de Edición y Compilación



Fuente: Ladrón de Guevara Jorge Martínez, 2015.

Además se consideran los siguientes puntos en el proceso de desarrollo de software:

- ❖ Especificaciones
- ❖ Diseño
- ❖ Codificación
- ❖ Pruebas
- ❖ Mantenimiento

1.6.9 Metodologías de Desarrollo e Implementación.

Para Piattini & García, (2003, pág. 10) un proceso de desarrollo de software determina las tareas a realizar por cada quién así como también detalla cuándo y cómo. Para iniciar el proceso de desarrollo se espera como datos de entrada las nuevas funcionalidades o requisitos que debe generar el sistema una vez desarrollado. En esencia dicho procedimiento está ligado a un modelo de proceso junto con la definición de los objetos actividades y roles, acorde

con Cendejas Valdéz, (2015) este se basa en una de las modalidades a continuación:

- ❖ Codificar y reparar
- ❖ Modelo en cascada,
- ❖ Desarrollo evolutivo
- ❖ Desarrollo formal
- ❖ Desarrollo basado en componentes
- ❖ Desarrollo ágil
- ❖ Desarrollo adaptativo de software (DAS)

Entre las metodologías más conocidas en el ámbito de desarrollo de software podemos destacar las siguientes acorde a García Blanco & Morales Ramos, (2003, pág. 9), Wells, (2009) y Beck, (1999):

✓ **Espiral**

Se distribuye en cuatro grandes fases: planeación, estudios de peligro, codificación y pruebas. Recibe el nombre por la forma circular y creciente en que se va de cada etapa a la siguiente.

✓ **Cascada**

Se llama ciclo de vida en cascada porque se va de una fase a la siguiente de manera lineal. Esto significa que hasta que una fase no esté completamente terminada no se pasa a la siguiente, sin existir la posibilidad del volver atrás.

✓ **Prototipo**

Un prototipo es un modelo evolutivo de la solución software final, no es el producto que finalmente se obtendrá.

✓ **Extreme Programming (XP)**

Acorde a Wells, (2009) XP es exitoso porque hace hincapié en la satisfacción del cliente. En vez de entregar todo lo que pueda desear en una fecha lejana en el futuro este proceso proporciona el software que necesita cuando lo necesite.

XP faculta a sus desarrolladores para responder con seguridad a las cambiantes necesidades de los clientes, incluso tarde en el ciclo de la vida. Enfatiza el trabajo en equipo.

1.6.10 Metodología de Desarrollo XP

Según la Revista Digital Universitaria, (2012) XP no es un método de desarrollo del todo sencillo, requiere de orden y disciplina, además de la capacidad para consolidar un equipo de trabajo con todas las personas y áreas que intervienen en el proyecto, incluido el cliente.

Por otra parte encontraremos que Piattini & García, (2003, pág. 10) indica que la metodología XP es básicamente, un método que dice que hacer en cada proceso y con qué

contenido; este tipo de organización se lo conoce como metodología de desarrollo, los autores coinciden en que parte fundamental en el desarrollo del proyecto es el cliente, además hacen hincapié en la importancia de la comunicación del equipo de trabajo y el cliente.

Según lo detalla Fernández Escribano, (2002) dentro de la introducción a la metodología Extreme Programming se encuentran las siguientes fases:

- ❖ **Etapa I:** Planeación
- ❖ **Etapa II:** Diseño Simple
- ❖ **Etapa III:** Codificación o Desarrollo
- ❖ **Etapa IV:** Adaptación y Pruebas

CAPÍTULO II

2. Metodología

2.1 Metodología de Investigación

2.1.1 Método General

El método general que se empleará es el deductivo debido a que permite identificar desde datos generales del ERP Adempiere en software libre hasta llegar a pequeñas referencias del módulo Anexo dentro del área contable.

2.1.2 Método Específico

✓ Investigación Bibliográfica

Será de apoyo en la búsqueda de conceptos, de esta manera se fundamentará el ERP, software libre y ATS, que permiten aclarar la escenografía y correcta funcionalidad que debe tener el módulo, dicha investigación se realizará en la infraestructura de la biblioteca de la Universidad Católica así como también de bibliografía web.

✓ Investigación de Campo

Se realizó en el área contable de la empresa piloto Corpocalza ubicada en la Av. Panamericana Km 2½ sector Catiglata de la ciudad de Ambato, para así comprender el correcto funcionamiento del ATS.

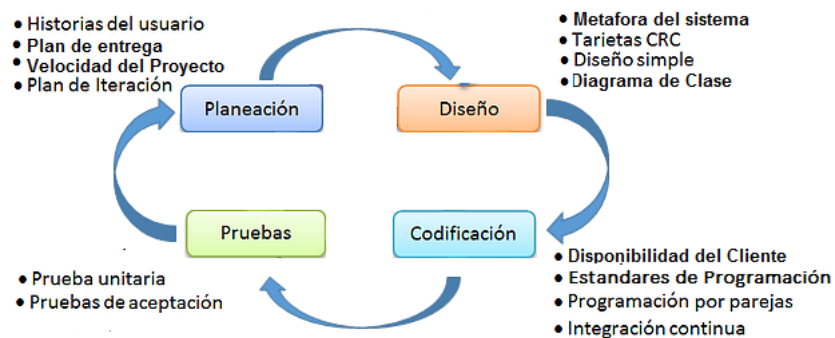
2.2 Metodología de Desarrollo

De acuerdo a Beck, Extreme Programming Explained. Embrace Change, (1999); Wake, (2002); Piattini & García, (2003), la metodología XP destaca dentro de las metodologías de desarrollo ágil por los siguientes puntos:

- ❖ **Comunicación:** Entre los desarrolladores y el cliente.
- ❖ **Simplicidad:** Diseño puntual y de fácil manejo.
- ❖ **Retroalimentación:** El usuario final debe estar presente durante todo el proyecto.
- ❖ **Pruebas Unitarias:** Etapa de testeo tras el desarrollo.
- ❖ **Historias de usuario:** Documentación de las pruebas realizadas.

Las Fases que se contemplan en la metodología XP acorde a Beck, Extreme Programming Explained. Embrace Change, (1999), son:

Ilustración 2.1 Fases Metodología XP



Fuente: Beck, Extreme Programming Explained. Embrace Change, (1999)

2.2.1 Fase de Planeación

Acorde a Kendall & Kendall, (2011), esta etapa define las funciones, procesos y las áreas de transacción de datos dentro de la empresa que se apoyará así como también se define el alcance del sistema.

Para alcanzar un correcto levantamiento de información la fase se apoya en:

✓ Historias de Usuario

Poseen el mismo propósito de generar estimaciones de tiempo para las reuniones de planificación. Conocida como la técnica para levantar las necesidades del sistema, consiste en tarjetas donde el usuario final menciona con brevedad las funciones con las que el software debe contar, sean estos detalles funcionales o no.

Una de las interrogantes que suele estar presente en las historias de usuario es ¿Qué nivel de detalle es el más adecuado? No existe una respuesta concreta como lo indican Jeffries & Anderson, (2001), quienes dicen que dichas historias van acorde a la dificultad del proyecto, por lo cual debería existir como mínimo una historia de usuario por proceso relevante.

✓ **Plan de Entrega**

Etapa en la cual el usuario procede a establecer la preferencia de las historias de usuario, por lo que, los desarrolladores llevan a cabo un estimado de recursos en base a cada una de las historias definidas.

Se establecen parámetros sobre el avance de la primera entrega y se levanta un cronograma de trabajo no mayor a tres meses junto al usuario final.

El cronograma se puede llevar a cabo acorde el tiempo especificado o el alcance del proyecto, por otra parte la velocidad de desarrollo permitirá establecer la cantidad de historias que pueden llegar a ser implementadas en un determinado tiempo o a una fecha previamente programada.

✓ **Velocidad del Proyecto**

Como su nombre lo indica y Wells, (2009) lo corrobora, permite medir la velocidad de un proyecto, se toma en cuenta las historias realizadas e implementadas en el cumplimiento de la primer etapa del proyecto, esta manera facilita la estimación de fechas y permite delegar actividades respecto a las funcionalidades del software.

✓ **Plan de Iteración**

En todas las iteraciones tanto el programador así como el cliente aprenden. Es recomendable no sobrecargar al programador con más trabajo del que pueda realizar, ya que no se cumplirán los plazos, como lo señala Wells, (2009) en su documento en línea; por otra parte, dice que los datos destacados a considerarse en la programación del plan son: las historias no analizadas, la celeridad de desarrollo, los intentos realizados que no han pasado la revisión y los pendientes de la iteración previa.

Al iterar por primera vez es recomendable establecer una estructura de sistema que pueda ser llevada a cabo durante todo el proyecto por lo encontrado en la publicación de Wells, (2009), sin embargo, cabe mencionar que no siempre puede ser ejecutada debido a que es el usuario final quien tiene la última palabra sobre qué historia de usuario se considera en cada iteración.

Todo el trabajo del plan de iteración es equivalente en tareas de desarrollo, cada tarea es entregada a un desarrollador. Podemos revisar también lo que menciona Wake, (2002) donde da notar varias guías de apoyo para generar la planeación de entrega dentro de cada iteración.

2.2.2 Fase de Diseño

En esta etapa se debe tomar en consideración lo dicho por Beck, *Extreme Programming Explained. Embrace Change*, (1999), quien comenta que el correcto diseño del software es el cual ha pasado sin inconveniente las pruebas a las que ha sido sometido, además de manifestar de manera clara la finalidad de los programadores en la menor cantidad de código posible.

✓ Metáfora del Sistema

Por lo encontrado en Letelier & Penadés, (2006) se concreta que la metáfora del sistema concibe la visión progresiva del proyecto a desarrollar y define los objetivos del sistema mediante las historia de usuario, se mantiene así relación entre nombres de aquello que se va a realizar.

✓ Tarjetas *Class Responsibility Collaborator* (CRC)

El objetivo de dichas tarjetas, según Wells, (2009), es realizar un inventario de las clases que vamos a requerir para llevar a cabo la implementación del sistema y la forma en que van a interactuar, de esta forma se puede facilitar el análisis de las mismas por parte del equipo de trabajo con la finalidad de conseguir el diseño más simple acorde las especificaciones y requerimientos del sistema.

✓ **Diseño Simple**

Según Beck, (Extreme Programming Explained. Embrace Change, 1999), se sugiere diseñar la solución más simple y funcional la cual puede ser puesta en funcionamiento en una determinada etapa del desarrollo. La dificultad redundante y el código excedente o innecesario deben ser removidos inmediatamente.

2.2.3 Fase de Codificación

✓ **Disponibilidad del Cliente (*in-situ*)**

Para Letelier & Penadés, (2006), el usuario debe encontrarse accesible en todo momento que el equipo de desarrollo lo requiera. El triunfo del desarrollo del proyecto mediante la metodología programación extrema se basa en que el usuario final es quien dirige asiduamente las labores a realizar. La comunicación cara a cara es más efectiva que la escrita, afianzando la relación programador - cliente además de fortalecer el desempeño del trabajo en equipo.

✓ **Estándares de Programación**

La metodología a utilizar hace énfasis en la intercomunicación entre los desarrolladores por medio de la fuente, motivo por el cual es primordial se sujeten a

ciertos esquemas de desarrollo. Los estándares permiten mantener el código comprensible para el equipo de trabajo, y esto a su vez facilita los cambios que puedan suscitarse.

✓ **Programación en parejas**

Referente la programación es recomendable realizarla como una labor en parejas de desarrolladores. Cockbun & Williams, (2000), señalan que en un análisis ejecutado con la finalidad de dar a conocer resultados costos/beneficio de llevar a cabo la programación en parejas, se destaca que las ventajas de esta modalidad de programación son:

- Gran cantidad de errores son detectados conforme avanza el desarrollo, como consecuencia el porcentaje de error es menor, despliega mejores diseños y su codificación es reducida.
- Los contratiempos dentro del desarrollo se solventan con mayor rapidez, debido a que existe transferencia de conocimientos entre el equipo de trabajo.
- Las ventajas mencionadas se consiguen tras meses de práctica sobre este tipo de programación.

- En los estudios realizados el tiempo es susceptible al primer trimestre.

✓ **Integración Continua**

Cada parte o elemento del código va acoplándose en el software cuando esté lista. De esta manera, el software puede ser unificado el número de instancias requeridas en un determinado tiempo. Este tipo de integración comúnmente disminuye la desintegración de lo realizado por parte de los programadores en caso de existir poca intercomunicación acerca de todo aquel código que permita ser reusable dentro de otras áreas de desarrollo del sistema.

Por su parte Fowler & Foemmel, (2001) afirman que es esencial coordinar la programación de un proceso disciplinado dentro del proyecto controlado, el equipo de trabajo se encuentra de esta manera preparado para modificar la fuente cuando se requiera, ya que cuenta con la seguridad en detección y manejo de fallos suscitados en la unificación.

2.2.4 Fase de Adaptación y Prueba

Kendall & Kendall, (2011) contempla la etapa de pruebas e inducción al usuario final, además de la implementación

del sistema se cumple con la capacitación sobre el manejo y funcionalidad del módulo del ATS.

✓ **Pruebas Unitarias**

Son establecidas previo a la escritura del código y llevadas a cabo de manera constante frente a la modificación del sistema.

Para Wells, (2009) el desarrollo guiado por pruebas formalizado dentro de la programación extrema se conoce como *Test Driven Development* (TDD), cuyo objetivo es el de mejorar la distribución de código interior sin afectar el desempeño superficial o visible al usuario.

Dentro de las pruebas unitarias encontramos las pruebas de caja blanca Pressman, (2006) menciona que es una de las pruebas más importantes dentro del software, las también llamadas pruebas de caja de cristal o estructurales se centran en detalles procedimentales del software, por lo que su diseño está directamente ligado al código fuente, de esta manera se logra una significativa disminución de errores junto a una mayor confiabilidad y calidad del sistema.

✓ **Pruebas de Aceptación**

Las pruebas de aceptación derivan de las historias de usuario levantadas y verifican que el sistema que se pone

en producción funcione acorde a los requisitos y necesidades levantadas con anterioridad. Para ser participe en un proceso de prueba de aceptación, se debe tener conocimiento más profundo sobre cómo se requiere funcione el sistema.

Para Wells, (2009) las también llamadas pruebas del cliente son realizadas por el usuario y se enfoca en las particularidades y funcionalidades generales del sistema que son visibles y revisables por parte del cliente además menciona que un sistema no puede ser considerado finalizado mientras no haya pasado por sus debidas pruebas de aceptación.

CAPÍTULO III

3. Resultados

3.1 Planificación de Requisitos

La planificación se llevó a cabo mediante reuniones entre el asistente contable encargado del manejo de la información requerida de la empresa Corpocalza y el equipo de desarrollo.

3.1.1 Historias de Usuario

A través de reuniones entre la persona encargada del manejo de ATS y el equipo de desarrollo se levantaron las respectivas historias de usuario de forma tal que se documenten correctamente las especificaciones de los requerimientos.

Para la realización de las historias de usuario se toma como referencia la tabla a continuación.

Tabla 3.1 Esquema historia de usuario

Historia de Usuario			
Número	Nro. Identificador	Usuario	Responsable del Proceso
Nombre de Historia	Proceso que se identifica		
Fecha	Día de realización		
Descripción	Pormenor de la historia en frases simples con lenguaje técnico.		
Anotaciones	Detalles importantes a analizar con el equipo de desarrollo.		

Fuente: Fernández Escribano, Introducción a Extreme

Programming, (2002)

A continuación se detallan las historias de usuario para su posterior análisis.

Tabla 3.2 Historia de usuario I

Historia de Usuario			
Número	1	Usuario	Asistente Contable
Nombre de Historia	Acceso al Módulo Anexo		
Fecha	20 de Noviembre del 2015		
Descripción: Se requiere tener el acceso al módulo de ATS para su posterior trabajo en el mismo.			
Anotaciones: Debido a que es un módulo del área contable se debe asignar los permisos solo al personal indicado por la empresa.			

Fuente: Elaboración Propia.

Tabla 3.3 Historia de usuario II

Historia de Usuario			
Número	2	Usuario	Asistente Contable
Nombre de Historia	Extraer datos desde otros módulos de Adempiere		
Fecha	22 de Noviembre del 2015		
Descripción: Se requiere que el módulo Anexo traiga los datos ingresados previamente en los distintos módulos de Adempiere, para evitar el reingreso de los mismos.			
Anotaciones: Tomar en consideración la configuración del sistema para la extracción correcta de datos.			

Fuente: Elaboración Propia.

Tabla 3.4 Historia de usuario III

Historia de Usuario			
Número	3	Usuario	Asistente Contable
Nombre de Historia	Generar datos para Anexo Transaccional Simplificado.		
Fecha	24 de Noviembre del 2015		
Descripción:			
Debe contener la manera de generar los datos a declarar en el ATS para su posterior consulta, revisión, aprobación y generación del respectivo archivo XML.			
Anotaciones:			
Tomar en cuenta los campos necesarios acorde la estructura del archivo XML del ATS que de momento no posee el sistema Adempiere.			

Fuente: Elaboración Propia.

Tabla 3.5 Historia de usuario IV

Historia de Usuario			
Número	4	Usuario	Asistente Contable
Nombre de Historia	Consultar datos para la generación del Anexo Transaccional Simplificado.		
Fecha	27 de Noviembre del 2015		
Descripción:			
El módulo debe desplegar la información que contendrá el XML del ATS para su revisión y confirmación de datos.			
Anotaciones:			
Tomar en cuenta que el usuario desea revisar la información que el módulo ATS trae de los módulos involucrados en todo el proceso.			

Fuente: Elaboración Propia.

Tabla 3.6 Historia de usuario V

Historia de Usuario			
Número	5	Usuario	Asistente Contable
Nombre de Historia	Generar Archivo XML		
Fecha	3 de Diciembre del 2015		
Descripción:			
El módulo debe contar con la opción para generar el XML que luego se subirá al DIMM (programa del SRI).			
Anotaciones:			
Como sugerencia debe permitir guardar el XML generado y declarado dentro de la ventana principal del módulo, por otro lado en caso de error en dato el sistema debe permitir volver a generar el XML.			

Fuente: Elaboración Propia.

3.1.2 Plan de Entregas

Acorde la relevancia de la historia de usuario se procedió a elaborar el plan de entregas, mismo en el que se consideró el tiempo calendario en detalle de horas, días y semanas a trabajar en el desarrollo.

Tabla 3.7 Plan de Entrega

Módulo	Nro.	Historia de Usuario	Calendario Estimado		
			Semanas	Días	Horas
Anexo	1	Acceso al módulo Anexo			2
	2	Extraer datos desde otros módulos de Adempiere	4		
	3	Generar datos para Anexo Transaccional Simplificado	2		
	4	Consultar datos para Anexo Transaccional Simplificado		3	
	5	Generar XML	4		

Fuente: Elaboración Propia.

3.1.3 Velocidad del Proyecto

Durante el desarrollo del módulo Anexo no se tuvo en consideración la estimación de la velocidad del proyecto en cuanto a duración de las iteraciones de forma detallada, sin embargo se consideró lo dicho por Wells, (2009) quien recomienda levantar un estimado y proceder al plan detallado de iteraciones ya que puede tomarse más tiempo en la determinación de la velocidad que el mismo desarrollo de las historias de usuario.

3.1.4 Plan de Iteraciones

En el plan de iteraciones los detalles importantes a tener en cuenta son:

Tabla 3.8 Plan de Iteración

Nombre de la Tarea	Inicio	Fin
Módulo de Anexo Transaccional Simplificado	15/10/2015	12/03/2016
ITERACIÓN 1 - PLAN	15/10/2015	24/11/2015
Planificación	15/10/2015	15/10/2015
Planificación del trabajo de la Iteración	15/10/2015	15/10/2015
Análisis	16/10/2015	09/11/2015
Estudio de estructura del ERP Adempiere	16/10/2015	16/10/2015
Análisis del requerimiento del sistema	17/10/2015	01/11/2015
Especificación de nuevas historias de usuario	02/11/2015	09/11/2015
Desarrollo	10/11/2015	23/11/2015
Definir la Metodología de Trabajo	10/11/2015	15/11/2015
Planificación de las actividades en el proyecto	16/11/2015	23/11/2015
Aceptación	24/11/2015	24/11/2015

Documento de respaldo del Plan del Proyecto	24/11/2015	24/11/2015
ITERACIÓN 2 - Diseño	25/11/2015	04/01/2016
Planificación	25/11/2015	25/11/2015
Planificación del trabajo de la iteración	25/11/2015	25/11/2015
Análisis	26/11/2015	09/12/2015
Estudio y análisis del documento generado en la iteración 2 para lograr un ambiente amigable al usuario	26/11/2015	01/12/2015
Identificar los componentes necesarios de la arquitectura	02/12/2015	09/12/2015
Desarrollo	10/12/2015	02/01/2016
Preparar documento de arquitectura	10/12/2015	13/12/2015
Diseño de base de datos	14/12/2015	19/12/2015
Diseño de interfaz de las historias de usuario	20/12/2015	25/12/2015
Determinar si se puede pasar a la fase de Construcción	26/12/2015	02/01/2016
Aceptación	03/01/2016	04/01/2016
Revisión del prototipo y funcionalidades junto al cliente	03/01/2016	04/01/2016
ITERACIÓN 3 - Desarrollo	04/01/2016	23/02/2016
Planificación	04/01/2016	04/01/2016
Planificación del trabajo de la Iteración	04/01/2016	04/01/2016
Análisis	05/01/2016	12/01/2016
Estudio de datos a conseguir, para posterior exportación de archivo requerido.	05/01/2016	10/01/2016
Toma en cuenta los estándares de programación planteados	11/01/2016	12/01/2016
Desarrollo	13/01/2016	19/02/2016
Desarrollo de vistas requeridas	13/01/2016	28/01/2016
Desarrollo de los triggers y procedimientos a utilizar	29/01/2016	19/02/2016
Aceptación	20/02/2016	23/02/2016
Pruebas de que lo desarrollado se encuentra en funcionamiento	20/02/2016	23/02/2016
ITERACIÓN 4 - Pruebas	24/02/2016	12/03/2016
Planificación	24/02/2016	24/02/2016
Planificación del trabajo de la Iteración	24/02/2016	24/02/2016

Análisis	25/02/2016	29/02/2016
Revisar logros de la iteración contra la planificación	25/02/2016	27/02/2016
Elaboración del plan de pruebas conjunto con el usuario.	28/02/2016	29/02/2016
Desarrollo	01/03/2016	08/03/2016
Desarrollo de pruebas en conjunto con el usuario	01/03/2016	04/03/2016
Evaluación de la fase	05/03/2016	08/03/2016
Aceptación	09/03/2016	09/03/2016
Documento de aceptación de diseño y funcionalidad por parte del usuario final.	09/03/2016	12/03/2016

Fuente: Elaboración Propia.

3.2 Diseño

Para una implementación más clara y dinámica de la aplicación, se realizaron, diseños simples y amigables que detallan las funcionalidades mediante un vocabulario de fácil entendimiento para el usuario final.

3.2.1 Metáfora del Sistema

El sistema funciona como una colmena de abejas, las cuales buscan polen en las flores y las traen a la colmena para hacerla más grande. Los procesos de recuperación de datos cumplen la función de las abejas, otros módulos como compras y ventas de donde se extrae la información se comparan con las flores y el XML a generar es la colmena robusta de información provista por los procesos. Para Beck, Extreme Programming Explained. Embrace Change, (1999) y Wells, (2009), es recomendable establecer una

metáfora dentro del desarrollo para no perder el foco del proyecto y que tanto el cliente como el desarrollador tengan un idea clara de lo que se busca construir.

3.2.2 Tarjetas C.R.C.

Dichas tarjetas permiten analizar con mayor facilidad cada una de las clases que intervienen dentro del desarrollo del proyecto y su posterior implementación.

Tabla 3.9 Esquema Tarjetas C.R.C.

Clase	
Responsabilidades	Colaboradores
Las responsabilidades o funciones que cumple la clase	Peticiones de un cliente al servidor o clase.

Fuente: Beck, (A Laboratory For Teaching Object-Oriented Thinking, 1989)

Tabla 3.10 CRC - Vista Compras

vats_compras	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> • Extrae los datos de las compras realizadas en el periodo requerido. • Revisa y verifica los datos con reportes del ERP para su validación. 	<ul style="list-style-type: none"> • Ninguno.

Fuente: Elaboración Propia.

Tabla 3.11 CRC - Vista Formas de Pago

vats_pagos	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> • Revisa y verifica la forma de pago con la que fueron canceladas las facturas de compra superiores a 1000 dólares. 	<ul style="list-style-type: none"> • vats_compras

Fuente: Elaboración Propia.

Tabla 3.12 CRC - Vista Retenciones

vats retenciones	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> • Extrae las retenciones emitidas al proveedor, detalla también los campos requeridos para el ATS. • Revisa y verifica los datos con los mayores contables de cuentas tributarias del ERP para su validación. 	<ul style="list-style-type: none"> • vats_compras

Fuente: Elaboración Propia.

Tabla 3.13 CRC - Vista Anuladas

vats anuladas	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> • Revisa y verifica los datos devueltos en la consulta de las facturas / NC emitidas por el cliente que han sido anuladas dentro del periodo a reportar. 	<ul style="list-style-type: none"> • Ninguno.

Fuente: Elaboración Propia.

Tabla 3.14 CRC - Vista Ventas

vats ventas	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> • Extrae las ventas de la empresa, así como las retenciones que han sido recibidas en el periodo requerido, todo esto agrupado por cliente. • Revisa los datos con reportes del ERP para su validación. 	<ul style="list-style-type: none"> • Ninguno.

Fuente: Elaboración Propia.

3.2.3 Diseño Simple Módulo ATS

El diseño simple referente al "MODULO DE ANEXO TRANSACCIONAL APLICADO AL ERP ADEMPIERE" constará de las ventanas de datos de empresa, compras, ventas,

retenciones, documentos anulados, formas de pago, Es importante destacar que dentro del diseño simple el software cubra las necesidades del cliente.

Dentro del diseño simple se encuentran también los campos que han sido agregados y/o adaptados en otros módulos del ERP Adempiere para el correcto funcionamiento del módulo ATS.

Estas adaptaciones han sido desplegadas en la ventanas de proceso respectivos acorde la ficha técnica del ATS, misma que se halla alojada en la web Servicio de Rentas Internas, (2015), dichos cambios pretenden facilitar el manejo para el usuario final durante el desarrollo del proceso.

Un mejor detalle de lo mencionado anteriormente es lo detallado a continuación.

✓ **Datos de Empresa**

Permite gestionar los datos relevantes de la empresa, así como el periodo contable que se pretende reportar a las entidades gubernamentales respectivas a través del ATS.

Ilustración 3.1 Pestaña de Datos de empresa

Fuente: Elaboración Propia.

✓ Compras

Reporta de forma detallada las compras realizadas durante el periodo requerido, además de desplegar las respectivas retenciones que han sido aplicadas, se obtiene así los datos requeridos dentro de la estructura del XML para su posterior carga.

Ilustración 3.2 Pestaña de Compras

Fuente: Elaboración Propia.

✓ Formas de Pago

Se encuentran detalladas las formas de pago de las facturas de proveedor que superen un valor monetario de 1000 dólares americanos, según lo indica el manual del ATS.

Ilustración 3.3 Pestaña Forma de Pago

The screenshot displays the 'Forma de Pago' (Payment Method) configuration screen. The interface includes a menu bar with 'Archivo', 'Editar', 'Ver', 'Ir', 'Herramientas', 'Ventana', and 'Ayuda'. Below the menu is a toolbar with various icons. The main content area contains the following fields and controls:

- Organization**: Text input field.
- Client**: Text input field.
- Active**: Checkable checkbox.
- Invoice**: Text input field.
- Codigo Forma de Pago**: Text input field.

The left sidebar contains the following menu items: 'Datos Empresa', 'Compras', 'Forma de Pago', 'Retenciones', 'Ventas', and 'Doc. Anulados'. The status bar at the bottom indicates 'Navegar o actualizar registro' and '-1/0'.

Fuente: Elaboración Propia.

✓ Retenciones Compra

En esta sección del módulo se detallan las retenciones emitidas por la empresa hacia los proveedores, para esto se debe configurar de forma correcta la matriz de retenciones que se usa durante el proceso de ingreso de compras.

Ilustración 3.4 Pestaña de Retenciones en Compra

The screenshot shows a web application window titled 'Anexo Transaccional prd System (Administrador) @ Pucesa.Pucesa [localhost[localhost-xe-pucesa]]'. The interface includes a menu bar (Archivo, Editar, Ver, Ir, Herramientas, Ventana, Ayuda) and a toolbar with various icons. On the left, there is a navigation pane with tabs: Datos Empresa, Compras, Forma de Pago, Retenciones, Ventas, and Doc. Anulados. The 'Retenciones' tab is active, displaying a form with the following fields: Organization, Client, Active, Invoice, Codigo Retención, Base Imponible, Porcentaje, and Valor Retenido. At the bottom of the window, there is a status bar that reads 'Navegar o actualizar registro' and a page number '-10'.

Fuente: Elaboración Propia.

✓ Ventas

Permite visualizar de forma agrupada por cliente las ventas realizadas y las retenciones que han emitido los clientes en el periodo a declarar.

Ilustración 3.5 Pestaña Ventas

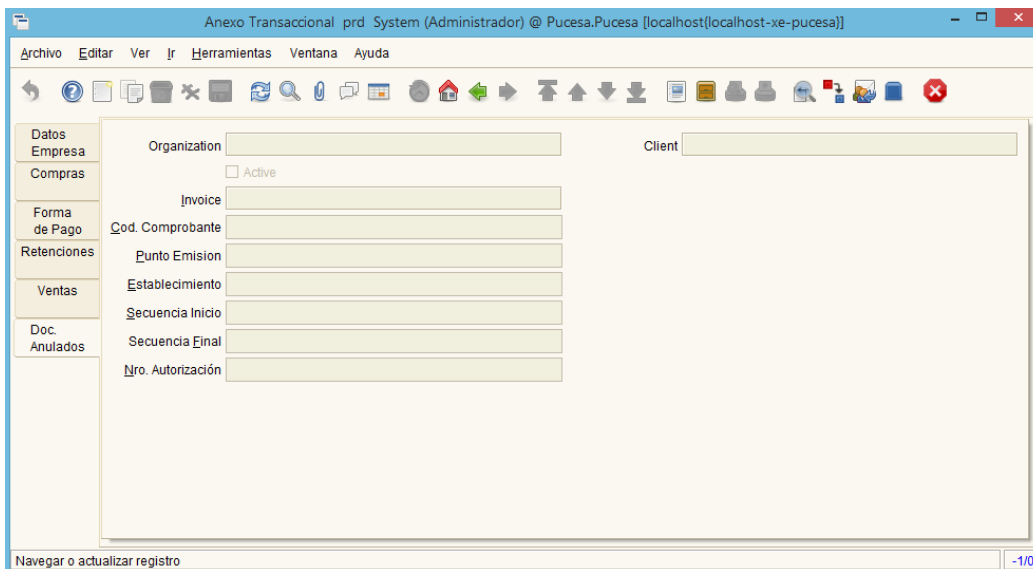
The screenshot shows the same application window as in Illustration 3.4, but with the 'Retenciones' tab selected for sales. The form fields are: Organization, Client, Active, Business Partner, Tipo ID Cliente, RUC / Cédula, Parte Relacionada, Tipo de Comprobante, Nro. Comprobantes, Base no Gravada, Base 0, Base Grava, Monto Iva, Monto Ice, Valor Ret. Iva, and Valor Ret. Fuente. The status bar at the bottom still shows 'Navegar o actualizar registro' and '-10'.

Fuente: Elaboración Propia.

✓ **Documentos Anulados**

Reporta los documentos fiscales anulados que han sido emitidos por la empresa durante el periodo requerido.

Ilustración 3.6 Documentos Anulados

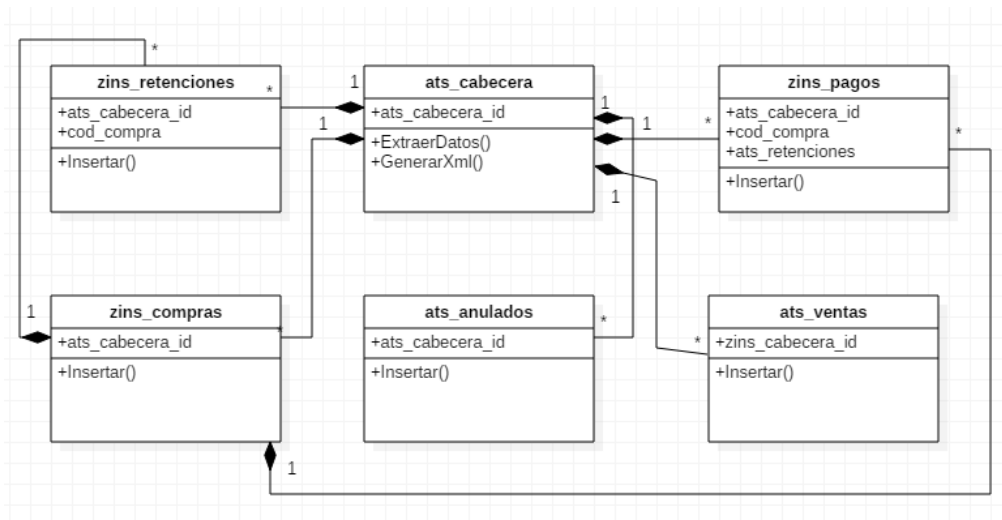


Fuente: Elaboración Propia.

3.2.4 Diagrama de Clases

La ilustración 3.7 detalla la estructura de la aplicación.

Ilustración 3.7 Diagrama de Clases



Fuente: Elaboración Propia.

3.3 Codificación

3.3.1 Disponibilidad del Cliente

La metodología XP permitió que el usuario de la empresa Corpocalza afine detalles en el proyecto conforme avanzaba su desarrollo, de tal manera que fue de gran aporte y generó un valor agregado al mismo, que el cliente se encontrara disponible durante todo el proceso de desarrollo, pruebas e implementación.

3.3.2 Estándares de Programación

Con el objetivo de que el código sea entendible para otro programador, se detallan las medidas y reglas usadas para escribir y documentar código.

- ❖ Documentar/Comentar el código de forma organizada y puntual acorde el lenguaje de programación, de tal manera que sea entendible para que otros programadores.
- ❖ Reutilizar código siempre que sea factible.
- ❖ En la creación de procesos u objetos las 3 o 4 primeras letras deben hacer referencia al tipo de proceso a crear, seguido de un guion bajo y el nombre que se le otorgará a dicho proceso.
- ❖ Manejar una correcta indentación en el código.
- ❖ Las variables deben empezar por "v".

3.3.3 Programación por Parejas

La metodología XP recomienda programar por parejas sin embargo esta fase sólo merece ser realizada en código complejo o extenso, en el cual quepan diferentes puntos de vista, sugerencias o análisis, por cuanto en código sencillo no da ventajas realizar este punto dentro de la metodología XP ya que los procesos son considerados simples y de fácil entendimiento.

3.3.4 Integración Continua

Se implementaron las historias de usuario definidas en base al diseño, hasta culminar con la aprobación de ensayos, acto seguido se observan fragmentos de código utilizado:

```
-- Acceso a UTL_FILE desde Sys para PUCESA
Grant Execute On SYS.UTL_FILE To PUCESA;

-- Crear directorio
create or replace directory DIR_PUCESA as 'C:\Temp';
```

Se detallan a continuación dos de las vistas utilizadas para la extracción de datos de Adempiere. Se debe considerar que la misma lógica de extracción se usó en las tres vistas restantes.

Vats_Compras.sql

```

create or replace view vats_compras as

select  fac.c_invoice_id,
        'Compra' || fac.c_invoice_id as cod_compra,
        ...
        //campos restantes
        ...

from    c_bpartner prv,
        c_invoice fac,
        c_doctype doct

where  fac.c_bpartner_id =prv.c_bpartner_id
and    fac.docstatus in('CO','CL')
and    fac.ad_client_id=1000002
and    fac.ad_org_id=1000002
and    fac.c_doctype_id=doct.c_doctype_id
and    doct.c_doctype_id in (1000005,1000006,1000050);

```

Vats_Retenciones.sql

```

create or replace view vats_retenciones as

select  zcd.cod_compra,
        ...
        // Campos Restantes
        ...
        zcd.c_invoice_id

from    lco_invoicewithholding ret,

```

```

        c_tax re,
        vats_compras zcd
where ret.c_tax_id= re.c_tax_id
...
        // condiciones
...

```

A continuación se describen uno de los cinco procedimientos creados para inserción en las nuevas tablas, de datos generados en las vistas mencionadas con anterioridad.

zins_ventas.prc

```

create or replace procedure zins_ventas(PC_Period_ID in
number,Pats_cabecera_id in number) is
vatsventasid number;
cursor cis is
    Select ...
        // Campos Restantes
    ...
From VATS_VENTAS t
    where t.c_period_id = PC_Period_ID;
ris cis%ROWTYPE;
BEGIN
    for ris in cis loop
        ad_sequence_next('ats_ventas', 1000002, atsventasid);
        insert into ats_ventas

```

```

        ( ...
            // Detalle de campos de table
        ... )
values
    (...
        // Datos a Insertar
    ...);

end loop;

EXCEPTION WHEN OTHERS THEN

    DBMS_OUTPUT.PUT_LINE('Error fallo en la inserción de
ventas');

END;
```

Acorde el funcionamiento del módulo desarrollado se menciona el disparador de la inserción de datos.

Inserta_datos.trg

```

CREATE OR REPLACE TRIGGER inserta_datos
after update ON ats_cabecera
For Each Row
when (new.generar_datos= 'Y' and old.generar_datos= 'N')
DECLARE
BEGIN

    zins_compras(:new.c_period_id,:new.ats_cabecera_id);
    zins_ventas(:new.c_period_id,:new.ats_cabecera_id);
    zins_pagos(:new.c_period_id,:new.ats_cabecera_id);
```

```

zins_anuladas (:new.c_period_id, :new.ats_cabecera_id);
ins_retenciones (:new.c_period_id, :new.ats_cabecera_id);
END inserta_datos;

```

Una vez se encuentran insertados los datos en las tablas respectivas se un procedimiento generará el XML mediante un disparador.

Z_generar_xml.prc

```

create or replace procedure
z_generar_xml (Pats_cabecera_id number, name_ats
varchar, num_establecimiento_ats varchar
, cod_operativo_ats varchar, c_period_id_ats number, ruc_ats
varchar, tipo_informante_ats varchar) is
vArchivoxml utl_file.file_type;
-- Info Tributaria Empresa
vperiod varchar(10);
...
// Variables
...
vc_invoicer_id number (10);
-- Info Factura Compras
cursor cif is
select distinct
...
// Campos

```

```
...  
  
from ats_compras f  
where f.ats_cabecera_id = Pats_cabecera_id;  
rif cif%ROWTYPE;  
  
-- info retenciones iva  
  
cursor cifr is  
  
    select distinct  
        ...  
        // Campos  
        ...  
    from ats_compras f  
    where f.ats_cabecera_id = Pats_cabecera_id  
    and f.c_invoice_id = vc_invoicer_id;  
    rifr cifr%ROWTYPE;  
  
-- Info Pagos  
  
cursor cim is  
  
    select distinct  
        ...  
        // Campos  
        ...  
    from ats_formapago f  
    where f.ats_cabecera_id = Pats_cabecera_id  
    and f.c_invoice_id = vc_invoice_id;  
    rim cim%ROWTYPE;  
  
-- Info Retenciones
```

```
cursor cdet is
    select ...
        // Campos
        ...
    from ats_retenciones f
    where f.ats_cabecera_id = Pats_cabecera_id
    and f.c_invoice_id = vc_invoiceret_id;
    rdet cdet%ROWTYPE;

-- Info Facturas de Venta
cursor cdim is
    select ...
        // Campos
        ...
    from ats_ventas f
    where f.ats_cabecera_id = Pats_cabecera_id;
    rdim cdim%ROWTYPE;

--Info Factura Anuladas
cursor cdim a is
    select ...
        // Campos
        ...
    from ats_anulados f
    where f.ats_cabecera_id = Pats_cabecera_id;
    rdim a cdim a%ROWTYPE;

begin
```

```
-- sacar numero periodo
select lpad(cp.periodno, 2,'0')
      into vperiod
from c_period cp
where cp.c_period_id=c_period_id_ats;

-- saca año
select cy.fiscalyear
      into vanio
from c_year cy, c_period ce
where ce.c_period_id=c_period_id_ats
and cy.c_year_id=ce.c_year_id;

-- sacar total ventas
select nvl((sum (v.basenograiva)+ sum(v.baseimponible)+
sum(v.baseimpgrav)),0)
      into vtotal
from ats_ventas v
where v.ats_cabecera_id=Pats_cabecera_id;

execute immediate 'alter session set
nls_numeric_characters = ','.';

-- Asignación de directorio y nombre para XML
vArchivoxml := utl_file.fopen('DIR_PUCESA',
'tesis'||to_char(sysdate,'ddmmyyyy')||'.xml', 'w');

-- Cabecera
utl_file.put_line(vArchivoxml, '<?xml version="1.0"
encoding="UTF-8"?>');
```

```

-- Informacion cabecera empresa
utl_file.put_line(vArchivoxml, ' <iva>');
...
// Código de inserción Cabecera XML
...
-- Info compras
utl_file.put_line(vArchivoxml, ' <compras>');
  for rif in cif loop
utl_file.put_line(vArchivoxml, ' <detalleCompras>');
...
// Código de inserción XML
...
utl_file.put_line(vArchivoxml, ' </detalleAnulados>');
end loop;
utl_file.put_line(vArchivoxml, ' </anulados>');
utl_file.put_line(vArchivoxml, ' <rendFinancieros/>');
utl_file.put_line(vArchivoxml, '</iva>');
utl_file.fclose(vArchivoxml);

EXCEPTION WHEN OTHERS THEN
  DBMS_OUTPUT.PUT_LINE('Error al generar el archivo
XML');
end z_generar_xml;

```

En el disparador detallado a continuación se hace referencia al procedimiento que genera el archivo XML.

Genera_xml.trg

```
CREATE OR REPLACE TRIGGER genera_xml
after update ON ats_cabecera
FOR EACH ROW
when (new.generar_xml = 'Y' and old.generar_xml = 'N')
DECLARE
BEGIN
    z_generar_xml(:new.ats_cabecera_id,:new.name,
                 :new.num_establecimiento,
                 :new.cod_operativo,
                 :new.c_period_id,:new.ruc,
                 :new.tipo_informante);
END genera_xml;
```

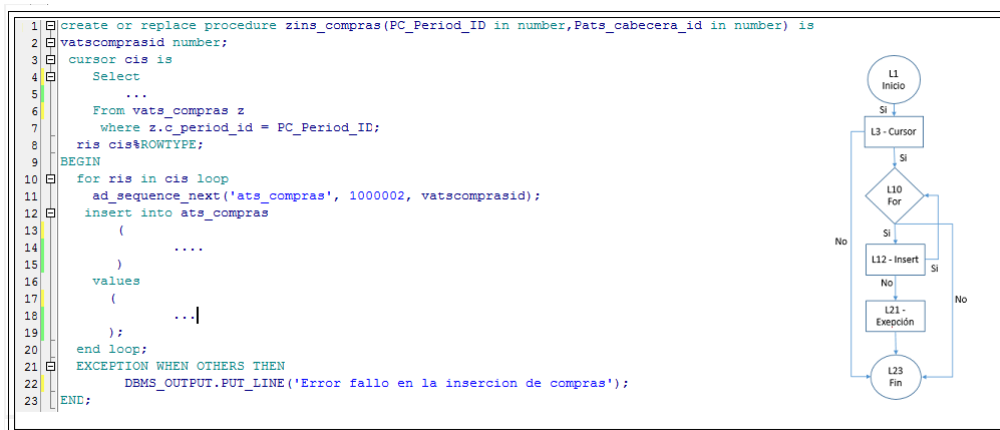
3.4 Pruebas

3.4.1 Pruebas Unitarias

Ya que se conoce el código y a través del seguimiento de la estructura lógica del mismo, se procedió a diseñar pruebas de caja blanca destinadas a comprobar que el código realice correctamente su función.

En la ilustración 3.8 se puede observar las pruebas unitarias realizadas en el código de inserción de compras, tras la consulta generada en la vista ats_compras.

Ilustración 3.8 Prueba de Caja Blanca - Compras



Fuente: Elaboración Propia.

Caja Blanca Pruebas 1ª Compras.

- Inicio Nodo L3 Cursor
 - o Se ejecutó el procedimiento sin datos en compras.
- Como resultado salta al Nodo L23 dando por finalizando el proceso ya que no existen datos.

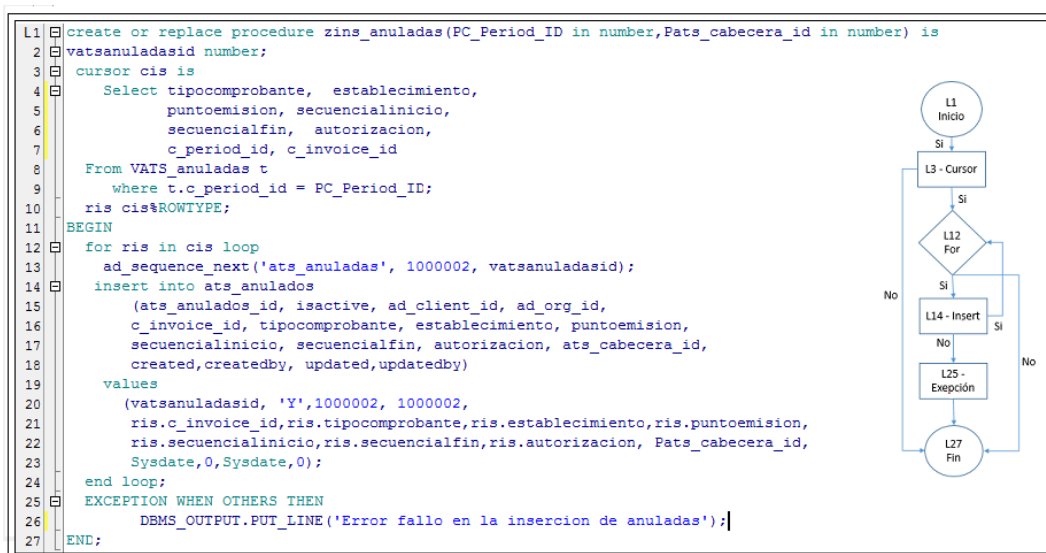
Caja Blanca Pruebas 2da Compras.

- Inicio Nodo L3 Cursor
 - o Se ejecuta el procedimiento con datos en compras.
- Ingresa al ciclo de inserción Nodo L10
 - o Al momento de insertar en caso de haber algún dato de tipo incorrecto salta al Nodo L21 mostrando una excepción, caso contrario inserta datos y comprueba que existen más datos.

- Regresa al ciclo inserción Nodo L10
- De no haber más datos pasa al Nodo L23 finalizando al proceso.

Por consiguiente se detallan las pruebas unitarias mediante caja blanca en el proceso de extracción de documentos anulados por la empresa, como se puede observar se utilizó la misma lógica de inserción que en compras para llevar una codificación uniforme.

Ilustración 3.9 Prueba de Caja Blanca - Anuladas



Fuente: Elaboración Propia.

Caja Blanca Pruebas 1ª Documentos Anulados.

- Inicio Nodo L3 Cursor
 - o Se ejecutó el procedimiento sin datos en documentos anulados.

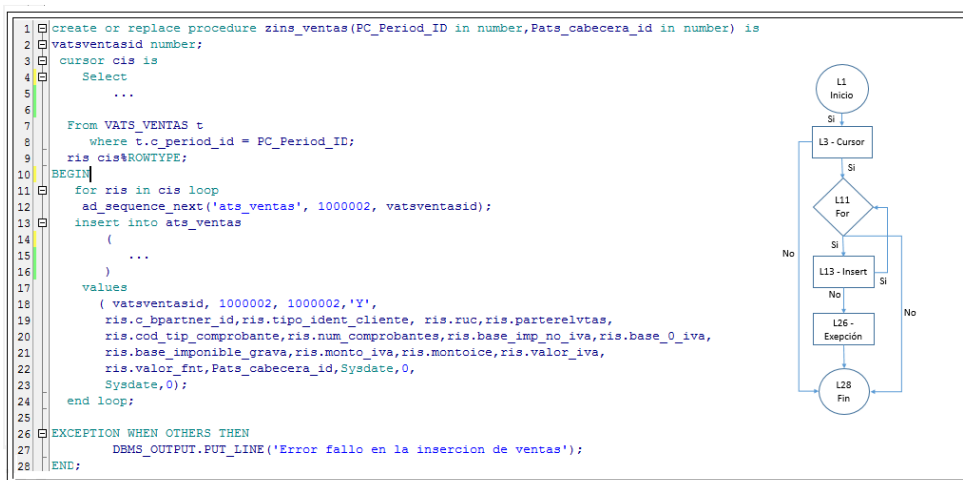
- Como resultado salta al Nodo L27 dando por finalizando el proceso ya que no existen datos.

Caja Blanca Pruebas 2da Documentos Anulados.

- Inicio Nodo L3 Cursor
 - o Se ejecuta el procedimiento con datos en pagos
- Ingresa al ciclo de inserción Nodo L12
 - o Al momento de insertar en caso de haber algún dato de tipo incorrecto salta al Nodo L25 mostrando una excepción, caso contrario inserta datos y comprueba que existen más datos.
- Regresa al ciclo inserción Nodo L12
- De no haber más datos pasa al Nodo L27 finalizando al proceso.

A continuación se observa las pruebas de caja blanca realizados sobre las ventas.

Ilustración 3.10 Prueba de Caja Blanca - Ventas



Fuente: Elaboración Propia.

Caja Blanca Pruebas 1ª Ventas.

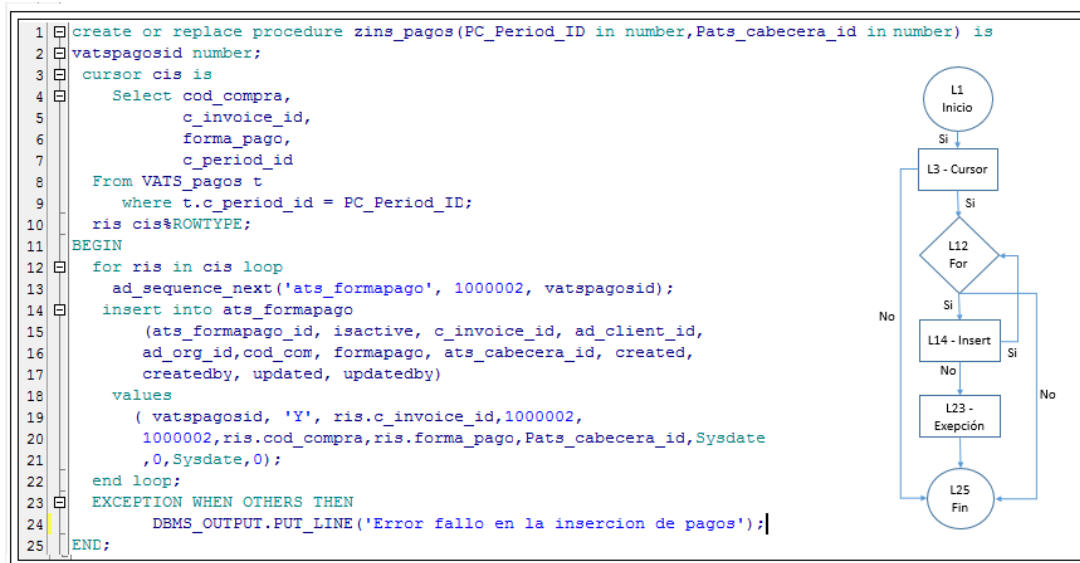
- Inicio Nodo L3 Cursor
 - o Se ejecutó el procedimiento sin datos en ventas.
- Como resultado salta al Nodo L28 dando por finalizando el proceso ya que no existen datos.

Caja Blanca Pruebas 2da Ventas.

- Inicio Nodo L3 Cursor
 - o Se ejecuta el procedimiento con datos en ventas.
- Ingresa al ciclo de inserción Nodo L11
 - o Al momento de insertar en caso de haber algún dato de tipo incorrecto salta al Nodo L26 mostrando una excepción, caso contrario inserta datos y comprueba que existen más datos.
- Regresa al ciclo inserción Nodo L11
- De no haber más datos pasa al Nodo L28 finalizando al proceso.

Siguiendo las pruebas de caja blanca se encuentra las realizadas sobre la inserción de pagos.

Ilustración 3.11 Prueba de Caja Blanca - Pagos



Fuente: Elaboración Propia.

Caja Blanca Pruebas 1ª Pagos.

- Inicio Nodo L3 Cursor
 - o Se ejecutó el procedimiento sin datos en ventas.
- Como resultado salta al Nodo L25 dando por finalizando el proceso ya que no existen datos.

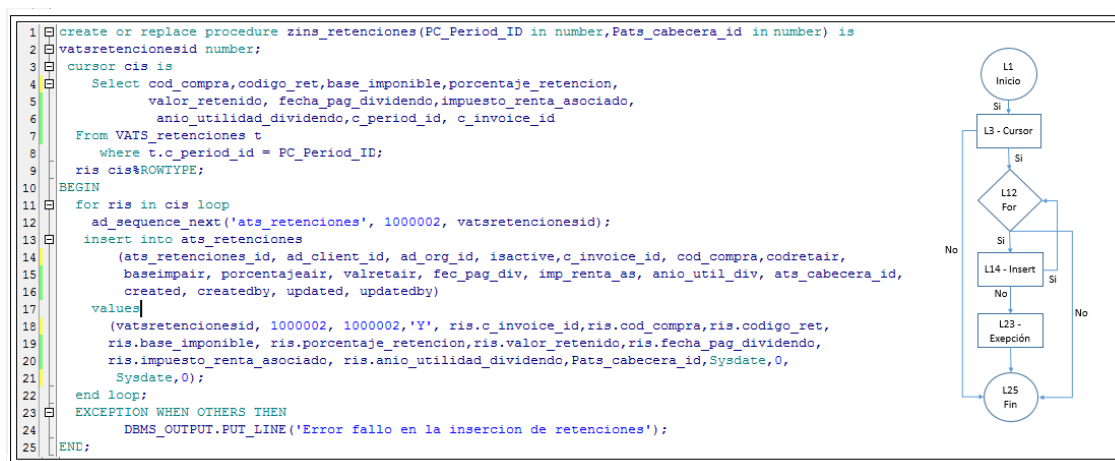
Caja Blanca Pruebas 2da Pagos.

- Inicio Nodo L3 Cursor
 - o Se ejecuta el procedimiento con datos en pagos.
- Ingresa al ciclo de inserción Nodo L12
 - o Al momento de insertar en caso de haber algún dato de tipo incorrecto salta al Nodo L23 mostrando una excepción, caso contrario inserta datos y comprueba que existen más datos.

- Regresa al ciclo inserción Nodo L12
- De no haber más datos pasa al Nodo L25 finalizando al proceso.

En la ilustración 3.12 encontramos las pruebas de caja blanca en la inserción de datos desde la vista `vats_retenciones` a la tabla de retenciones.

Ilustración 3.12 Prueba de Caja Blanca - Retenciones



Fuente: Elaboración Propia.

Caja Blanca Pruebas 1ª Retenciones.

- Inicio Nodo L3 Cursor
 - o Se ejecutó el procedimiento sin datos en ventas.
- Como resultado salta al Nodo L25 dando por finalizando el proceso ya que no existen datos.

Caja Blanca Pruebas 2da Retenciones.

- Inicio Nodo L3 Cursor
 - o Se ejecuta el procedimiento con datos en pagos.

- Ingresa al ciclo de inserción Nodo L12
 - o Al momento de insertar en caso de haber algún dato de tipo incorrecto salta al Nodo L23 mostrando una excepción, caso contrario inserta datos y comprueba que existen más datos.
- Regresa al ciclo inserción Nodo L12
- De no haber más datos pasa al Nodo L25 finalizando al proceso.

3.4.2 Pruebas de Aceptación

La finalidad del desarrollo es llegar a un diseño simple que cumpla con las expectativas del usuario y las funcionalidades definidas en las historias de usuarios. Acto seguido se procede a presentar las pruebas de aceptación realizadas sobre el presente proyecto.

Tabla 3.15 Esquema de Pruebas de Aceptación

Número de Prueba:	Nro. de Prueba d Aceptación
Historia de Usuario:	Acceso al Módulo
Requisitos:	<ul style="list-style-type: none"> • Requisitos para el funcionamiento de la historia de usuario
Entrada:	<ul style="list-style-type: none"> • Datos ingresados
Resultado 1:	<ul style="list-style-type: none"> • Resultados Obtenidos 1
Resultado 2:	<ul style="list-style-type: none"> • Resultados Obtenidos 2

Fuente: Junta de Andalucía, (2016)

El cliente quien en este caso y para las posteriores pruebas será la asistente contable, accede al sistema mediante el usuario y contraseña que le ha sido asignado, visualiza el módulo de Anexo asignado y da por aceptada la primera prueba como lo indica la Tabla 3.16.

Tabla 3.16 Pruebas de Aceptación 1

Número de Prueba:	1
Historia de Usuario:	Acceso al Módulo
Requisitos:	<ul style="list-style-type: none"> • Usuario creado en el ERP
	<ul style="list-style-type: none"> • Tener asignado un rol predeterminado en el ERP
Entrada:	<ul style="list-style-type: none"> • Ingresa usuario / Contraseña
	<ul style="list-style-type: none"> • Elige el rol predeterminado o asignado
	<ul style="list-style-type: none"> • Visualiza el módulo ATS dentro de su menú
Resultado 1:	<ul style="list-style-type: none"> • Permite el acceso a la aplicación
Resultado 2:	<ul style="list-style-type: none"> • Permite la visualización y manejo del módulo ATS

Fuente: Elaboración Propia.

En la siguiente prueba de aceptación la asistente contable ingresa los datos de cabecera de la empresa y además gestiona la extracción de datos mediante el modulo Anexo de los demás módulos del ERP Adempiere, los cuales posteriormente se verán reflejados en el archivo XML.

Tabla 3.17 Pruebas de Aceptación 2

Número de Prueba:	2
Historia de Usuario:	<ul style="list-style-type: none"> • Generar Datos para Anexo Transaccional Simplificado
Requisitos:	<ul style="list-style-type: none"> • Datos en Compras / Ventas / Retenciones
Entrada:	<ul style="list-style-type: none"> • El usuario ingresa datos en cabecera de ventana Anexo
	<ul style="list-style-type: none"> • Generar Datos para el Anexo Transaccional Simplificado
Resultado 1:	<ul style="list-style-type: none"> • Datos para su posterior revisión
Resultado 2:	<ul style="list-style-type: none"> • No Aplica

Fuente: Elaboración Propia.

Continuando con las pruebas de aceptación como lo visualizaremos en la Tabla 3.18 la asistente contable revisa los datos previamente generados en el módulo de Anexo para su posterior prueba de aceptación.

Tabla 3.18 Pruebas de Aceptación 3

Número de Prueba:	3
Historia de Usuario:	Consultar Datos Anexo
Requisitos:	Haber generado datos en el Anexo
Entrada:	Revisión de Datos
Resultado 1:	El usuario procede al siguiente proceso de generar XML.
Resultado 2:	No Aplica

Fuente: Elaboración Propia.

En la prueba final es generado el archivo XML por parte de la asistente contable, dando por finalizado el proceso y la aceptación de la total funcionalidad del módulo.

Tabla 3.19 Pruebas de Aceptación 4

Número de Prueba:	4
Historia de Usuario:	Genera archivo XML
Requisitos:	Haber verificado y cortejado que los datos sean correctos
Entrada:	El usuario genera el XML
Resultado 1:	Obtiene el archivo XML
Resultado 2:	El Usuario sube el archivo generado al DIMM

Fuente: Elaboración Propia.

CAPÍTULO IV

4. Análisis y validación de los resultados

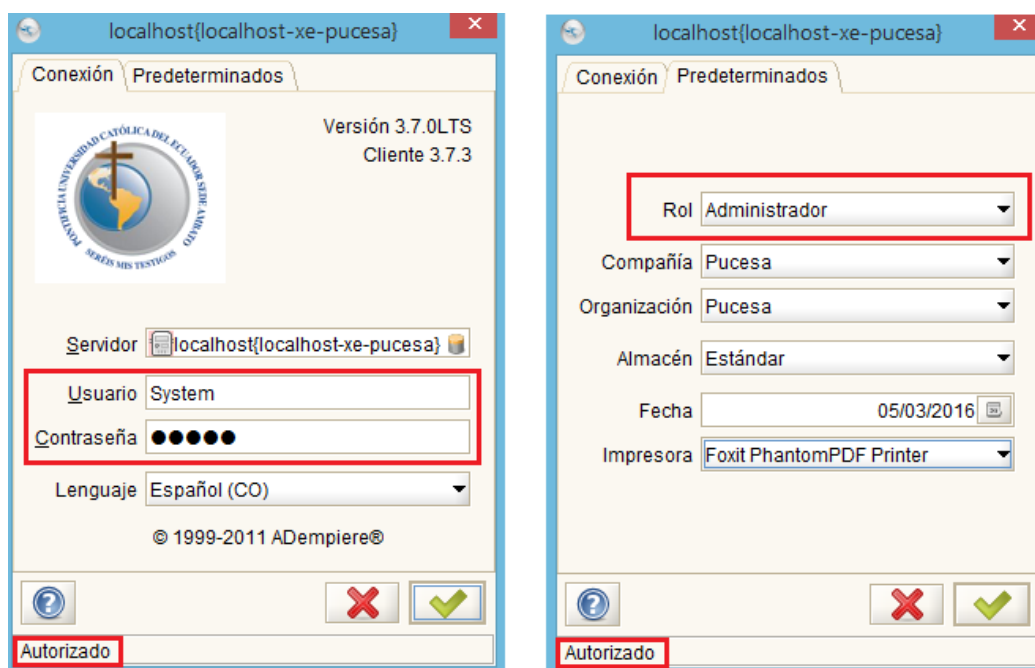
4.1 Análisis de Resultados

4.1.1 Resultado de la Implementación

✓ Ingreso

La ilustración indica el ingreso al sistema, el validado de usuario y contraseña, además de permitir visualizar el rol a cual se ha otorgado el respectivo acceso.

Ilustración 4.1 Ingreso Valido al ERP



Fuente: Elaboración Propia.

Ilustración 4.2 Ingreso Inválido al ERP

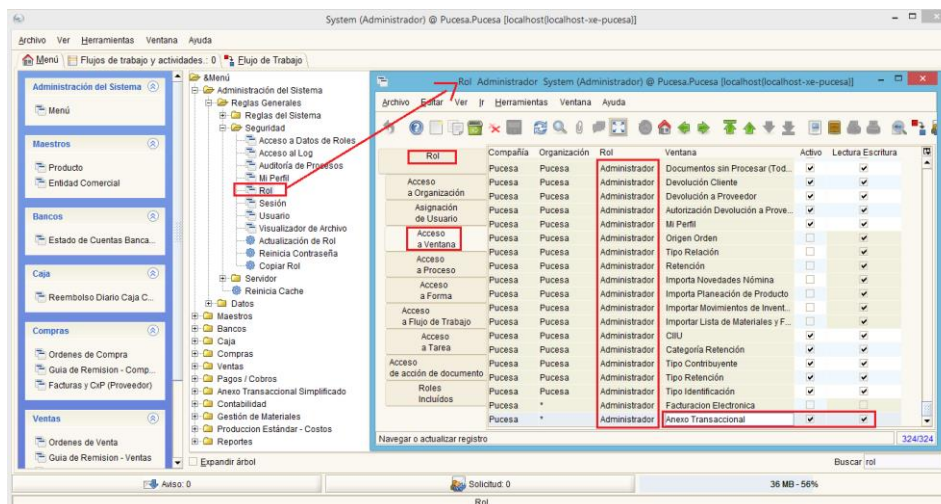


Fuente: Elaboración Propia.

✓ Acceso al Módulo

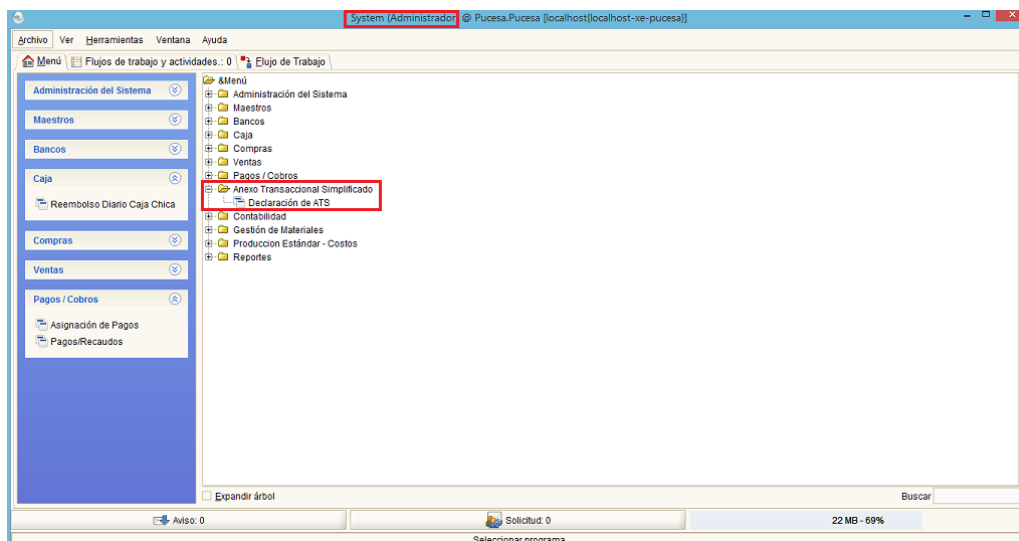
Con los resultados se puede observar que el menú ya cuenta con el módulo de Anexo Transaccional Simplificado asignado al rol respectivo, además se observa la ventana donde se realizan las respectivas transacciones para generar el XML.

Ilustración 4.3 Asignación de Acceso al módulo por Rol



Fuente: Elaboración Propia.

Ilustración 4.4 Menú con Acceso Módulo



Fuente: Elaboración Propia.

✓ Detalle de Datos Maestros

Dentro de la ventana que se despliega se observa que la pestaña principal contiene los datos relevantes de la empresa así como también el periodo del cual se desea generar el XML para su posterior declaración.

Ilustración 4.5 Datos Empresa Guardado Validado

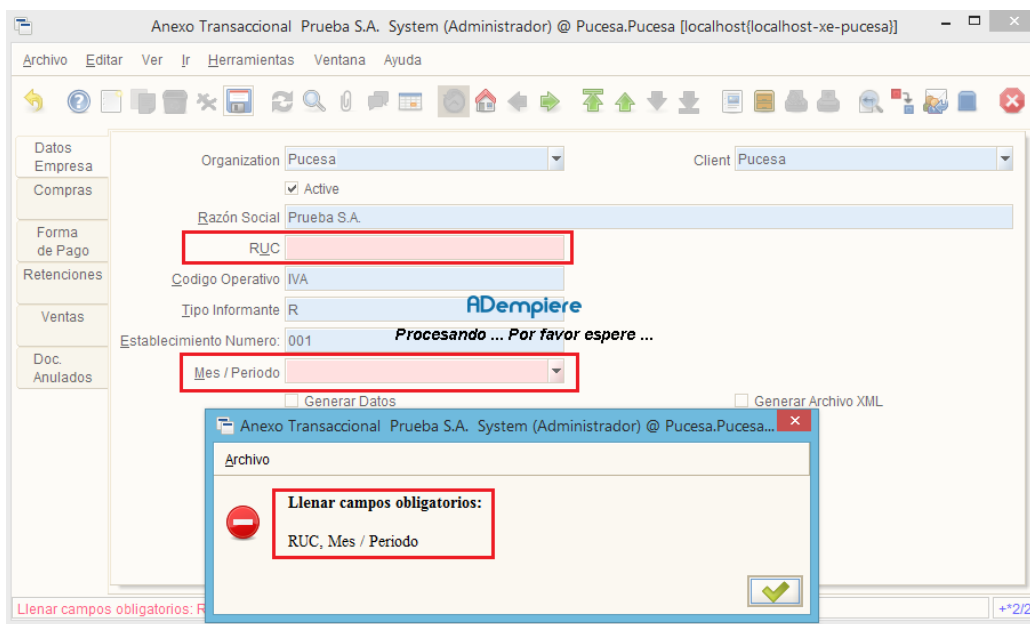
 A screenshot of a software application window titled 'Anexo Transaccional Corporcalza S.A. System (Administrador) @ Pucesa.Pucesa [localhost[localhost-xe-pucesa]]'. The window displays a form for 'Datos Empresa'. The form contains the following fields:

Datos Empresa	Organization	Pucesa	Client	Pucesa
Compras		<input checked="" type="checkbox"/> Active		
Forma de Pago	Razón Social	Corporcalza S.A.		
Retenciones	RUC	0921799999		
Ventas	Código Operativo	IVA		
Doc. Anulados	Tipo Informante	R		
	Establecimiento Numero:	001		
	Mes / Periodo	feb-16		
		<input type="checkbox"/> Generar Datos		

 The status bar at the bottom indicates 'Datos refrescados' and '2/2'.

Fuente: Elaboración Propia.

Ilustración 4.6 Datos Maestros Ingreso Inválido

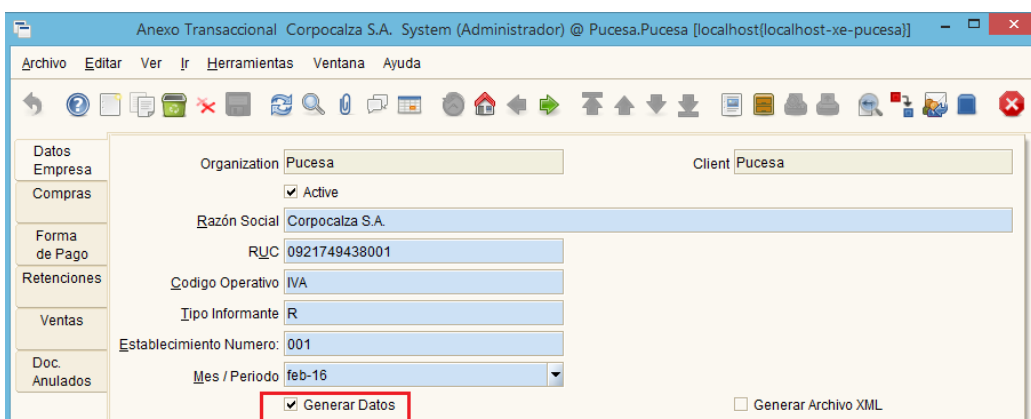


Fuente: Elaboración Propia.

✓ Carga y Revisión de Datos

En las pestañas restantes se puede visualizar la información generada que posteriormente pasa a formar el XML.

Ilustración 4.7 Generar Datos para ATS



Fuente: Elaboración Propia.

Los datos se pueden revisar individualmente o en modo rejilla como se muestra a continuación.

Ilustración 4.8 Datos Compras- Rejilla

Invoice	001-001-897654567_29/02/2016_1.611,00
Código Sustento	01
Cod. ID Proveedor	02
Buc / Cédula	0929999999
Cgd. Comprobante	01
Parte Relacionada	NO
Fecha Registro	29/02/2016
Establecimiento	001
Punto de Emisión	001
Secuencial	897654567
Fecha Emisión	29/02/2016
Nro. Autorización	8888888888
Base No Grava	0,00
Base Imponible	0,00
Base Grava	1.500,00
Base Eyecta	0,00
Monto Ica	0,00
Monto Iya	180,00
Valor Ret. Bienes 10%	0,00
Valor Ret. Servicios 20%	0,00

Fuente: Elaboración Propia.

Ilustración 4.9 Datos Forma de Pago

Invoice	001-001-897654567_29/02/2016_1.611,00
Código Forma de Pago	02

Fuente: Elaboración Propia.

Ilustración 4.10 Datos de Retenciones

Organization: Pucesa Client: Pucesa

Compras: Active

Invoice: 001-001-269876567_29/02/2016_107,40

Código Retención: 312

Base Imponible: 100,00

Porcentaje: 1

Valor Retenido: 1,00

Fuente: Elaboración Propia.

Ilustración 4.11 Datos de Ventas

Organization: Pucesa Client: Pucesa

Compras: Active

Business Partner: Estándar

Tipo ID Cliente: 05

RUC / Cédula: 0929999999

Parte Relacionada: NO

Tipo de Comprobante: 18

Nro. Comprobantes: 1

Base no Gravada: 0,00

Base 0: 0,00

Base Grava: 34.587,00

Monto Iva: 4.150,44

Monto Ice: 0,00

Valor Ret. Iva: 0,00

Valor Ret. Fuente: 0,00

Fuente: Elaboración Propia.

Ilustración 4.12 Datos de Documentos Anulados

Organization: Pucesa Client: Pucesa

Compras: Active

Invoice: 001-002-909009851_06/02/2016_2,24

Cod. Comprobante: 18

Punto Emision: 002

Establecimiento: 001

Secuencia Inicio: 909009851

Secuencia Final: 909009851

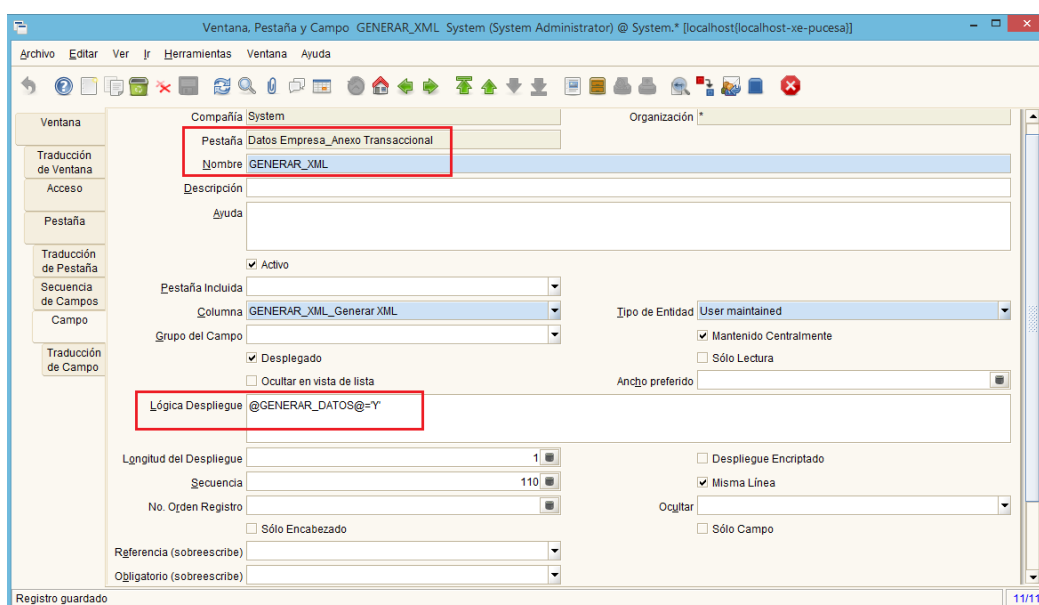
Nro. Autorización: 8909876567

Fuente: Elaboración Propia.

✓ Generar XML

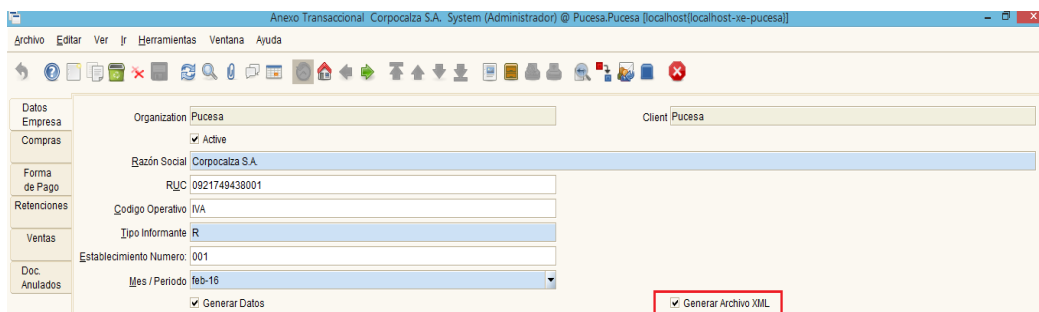
Dentro de la pestaña cabecera se encuentra la opción que permite generar el archivo XML, en la ubicación predefinida, para luego ser cargada al DIMM por el usuario final.

Ilustración 4.13 Configuración Campo Genera XML



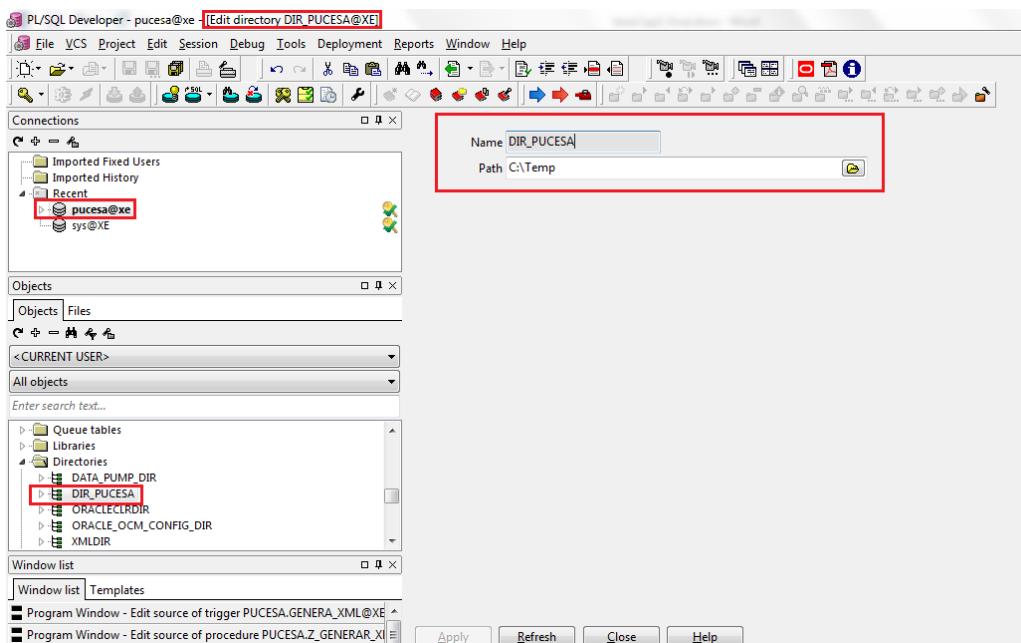
Fuente: Elaboración Propia.

Ilustración 4.14 Generar XML



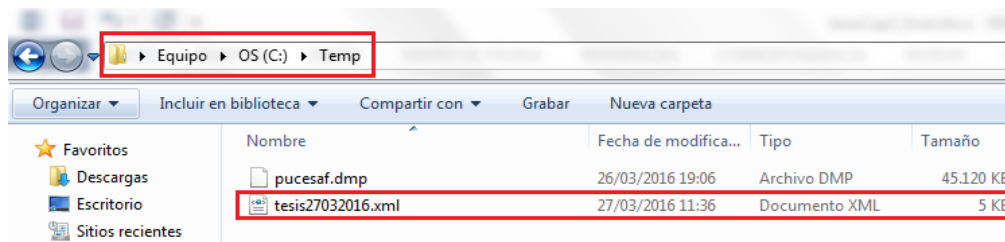
Fuente: Elaboración Propia.

Ilustración 4.15 Configuración de Directorio XML



Fuente: Elaboración Propia.

Ilustración 4.16 Verificación de XML generado



Fuente: Elaboración Propia.

4.2 Validación de Resultados



Calle Toronto Ne001-226 y Av. Indoamerica,
Sector Catiglata Edificio Plasticaucho Industrial
Telf. 032 998500
Ambato - Ecuador

Ambato, 24 Marzo 2016.

Msc. Teresa Freire Aillón
Directora de la Escuela de Sistemas
Pontificia Universidad Católica del Ecuador sede Ambato
Presente.-

De mi consideración.

Yo, Sandra Carolina Andrade López, analista contable del área de tributación de la empresa **CORPOCALZA S.A** RUC 1891730213001, certifico que se realizaron las respectivas revisiones y verificaciones, validando correctamente el módulo de "*Anexo Transaccional Simplificado integrado al ERP Adempiere*" realizado por la Sra. Mayra Alejandra Coronel Balseca.

Por la atención quedo de usted muy agradecida.

Atentamente,

CORPOCALZA S.A.

CPA. Sandra Andrade
C.C 180414317-8

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- ✓ El desarrollo del módulo Anexo transaccional simplificado, se llevó a cabo luego de indagar, estudiar y profundizar sobre el manejo y funcionamiento del ERP Adempiere, además de aprender y estudiar el proceso que genera el XML para su posterior declaración a entidades gubernamentales.

- ✓ El diseño fue realizado con la mayor simplicidad posible, se consideraron las sugerencias del usuario final, así como también la fase de la metodología la cual propone simplicidad de diseño.

- ✓ La integración del módulo en la empresa Corpocalza, luego de haber analizado y estudiado el ERP Adempiere fue un éxito, cumpliendo con las expectativas del usuario final.

Recomendaciones

- El personal responsable de ingreso de la información debe tener claro el panorama respecto a las normativas legales vigentes al momento de ingresar la información para que los resultados del desarrollo sean los correctos.

- El administrador del sistema será el responsable de mantener actualizado el ERP, en caso de surgir cambios del Servicio de Rentas Internas que afecten la estructura del XML.

- Se aconseja respaldar la información que maneja el ERP de forma periódica, debido a que si se presenta algún inconveniente tras una actualización, la última base respaldada podrá ser migrada a un nuevo servidor sin mayor contratiempo.

BIBLIOGRAFIA

- Beck, K. (1989). *A Laboratory For Teaching Object-Oriented Thinking*. Obtenido de <http://c2.com/doc/oopsla89/paper.html>
- Beck, K. (1999). *Extreme Programming Explained. Embrace Change*. Pearson Education.
- Centro de Ingeniería del Software e Ingeniería del Conocimiento. (2004). *METODOLOGÍA PARA SELECCIÓN DE SISTEMAS ERP*. Obtenido de <http://www.ucla.edu.ve/dac/departamentos/informatica-II/metodologia-para-seleccion-de-sistemas-erp.PDF>
- Cockbun, & Williams. (2000). *The Costs and Benefits of Pair Programming*.
- Enriquez Villavicencio, E. G., & Silva Corrales, D. F. (20 de Mayo de 2015). *Universidad Politecnica Nacional*. Obtenido de <http://bibdigital.epn.edu.ec/handle/15000/10588>
- Fernández Escribano, G. (09 de Diciembre de 2002). *Introducción a Extreme Programming*. Obtenido de <http://eblogingenieria.blogspot.com/2012/11/introduccion-la-programacion-extrema-xp.html>
- Fernández Escribano, G. (9 de Diciembre de 2002). *Introducción a Extreme Programming*. Obtenido de https://www.dsi.uclm.es/documentos/memoaca/0708/Memo_0708.pdf

- Fowler, M., & Foemmel, M. (2001). *Continuous Integration*.
- García Blanco, S., & Morales Ramos, E. (2003). *Análisis y Diseño Detallado de Aplicaciones Informaticas de Gestión*.
- Guevara, J. M. (12 de Junio de 2015). *Fundamentos de Programacion en Java*. Obtenido de <http://pendientedemigracion.ucm.es/info/tecnomovil/documentos/fjava.pdf>
- HispaLinux. (s.f.). *HispaLinux*. Obtenido de <http://hispalinux.es/SoftwareLibre>
- Horgren, C. T. (2010). *Contabilidad*.
- Jeffries, R., & Anderson, A. (2001). *Extreme Programming Installed*. Addison-Wesley. Obtenido de Jeffries, R., Anderson, A., Hendrickson, C. "Extreme Programming Installed". Addison-Wesley. 2001
- Junta de Andalucía. (2016). *Junta de Andalucía*. Obtenido de Plantilla Plan de Pruebas de Aceptación: <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/463>
- Kendall, K., & Kendall, J. (2011). Analisis y diseño de sistemas. Mexico: Pearson Educación.
- La Biblia MySQL*. (12 de Mayo de 2015). Obtenido de <https://osmell.files.wordpress.com/2008/08/la-biblia-de-mysqylanaya-multimedia.pdf>

Letelier, P., & Penadés, M. (15 de Enero de 2006).

Métodologías ágiles para el desarrollo de software:

eXtreme Programming (XP). Obtenido de

<http://www.cyta.com.ar/ta0502/v5n2a1.htm>

Matsukawa Maeda, S. (2006). Oracle: Implementación y

operación de una base de datos para version 8i, 9i y

10g. Lima: Macro.

Ministerio de Comercio Exterior. (s.f.). *PRO ECUADOR*.

Obtenido de

[www.proecuador.gob.ec/2012/07/23/microempresas-](http://www.proecuador.gob.ec/2012/07/23/microempresas-crecen-un-41-en-este-ano)

[crecen-un-41-en-este-ano](http://www.proecuador.gob.ec/2012/07/23/microempresas-crecen-un-41-en-este-ano)

Open Source Initiative. (s.f.). Recuperado el 12 de Mayo

de 2015, de <http://opensource.org/>

Piattini, M., & García, F. (2003). *Calidad en el*

desarrollo y mantenimiento del software.

Pressman, R. J. (2006). *Ingeniería de Software Un enfoque*

práctico. México: McGraw Hill.

Ramos, S. T. (2015). *SISBIB*. Recuperado el 12 de 06 de

2015, de Sistema de Bibliotecas:

<http://sisbib.unmsm.edu.pe/bibvirtual/publicaciones/>

[quipukamayoc/1999/primer/global.htm](http://sisbib.unmsm.edu.pe/bibvirtual/publicaciones/quipukamayoc/1999/primer/global.htm)

Revista Digital Universitaria. (01 de Enero de 2012).

Obtenido de

<http://www.revista.unam.mx/vol.13/num1/art16/>

- Rodriguez Rocha, F. X. (2009). *Manual de Contabilidad Práctica*.
- Secretaria Nacional de la Administración Pública. (s.f.).
Secretaria Nacional de la Administración Pública.
Obtenido de
<http://www.administracionpublica.gob.ec/importancia-del-software-libre-para-un-pais/>
- Servicio de Rentas Internas. (10 de 06 de 2015). Obtenido de <http://www.sri.gob.ec/web/10138/32@public>
- Servicio de Rentas Internas. (08 de 10 de 2015). *Servicio de Rentas Internas*. Obtenido de
<http://www.sri.gob.ec/de/anexo-transaccional-simplificado-ats>
- Servicio de Rentas Internas. (2015). *SRI - Régimen Impositivo Simplificado*. Obtenido de
<http://www.sri.gob.ec/web/guest/230>
- Sistemas Gestores de Bases de Datos*. (12 de Junio de 2015). Obtenido de <http://www.mcgraw-hill.es/bcv/guide/capitulo/8448148797.pdf>
- Tennent, J. (2010). *Gestión financiera: principios básicos paso a paso*. Quito: Luppá.
- Wake, W. (2002). *Extreme Programming Explored*. Addison-Wesley.
- Wandeen-Bergher Lozano, J. L. (2012). *Introducción a la Contabilidad*.

Wells, D. (2009). *Extreme Programming: A gentle introduction*. Obtenido de <http://www.extremeprogramming.org/rules.html>