



Universidad de las Ciencias Informáticas
Facultad 3

**Título: Análisis y Diseño del proceso Verificación Fiscal para
el Sistema de Informatización de la Gestión de las Fiscalías**

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autor(es):

Leanet Broche Ceballo.

Julio Adalberto Leyva.

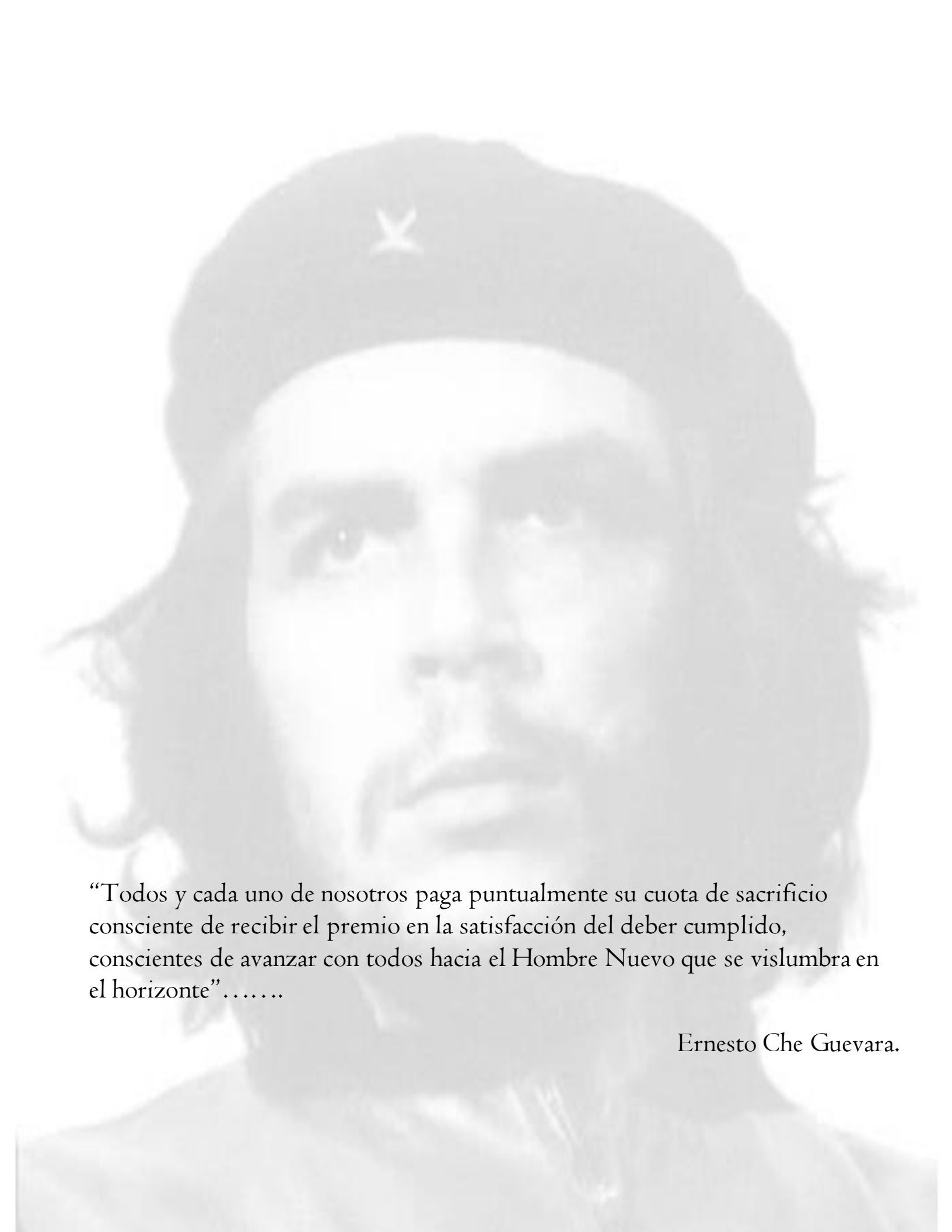
Tutor:

Yenier Figueroa Machado

Co-tutor:

Annaliet Parra Pérez

La Habana, junio de 2011



“Todos y cada uno de nosotros paga puntualmente su cuota de sacrificio consciente de recibir el premio en la satisfacción del deber cumplido, conscientes de avanzar con todos hacia el Hombre Nuevo que se vislumbra en el horizonte”

Ernesto Che Guevara.

DECLARACIÓN DE AUTORÍA

Declaramos ser los autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Leanet Broche Ceballo

Firma del Autor

Julio A Leyva González

Firma del Autor

Yenier Figueroa Machado

Firma del Tutor

Annaliet Pérez Parra

Firma de la Cotutora

DATOS DE CONTACTO

Datos de tutor: Nombre: Ing. Yenier Figueroa Machado

Ingeniera en Ciencias Informáticas. 4 años de experiencia laboral. Jefe del proyecto Sistema de Informatización de Gestión de las Fiscalías.

Correo electrónico: yfigueroa@uci.cu

Datos de co-tutor: Nombre: Ing. Annaliet Parra Pérez

Ingeniera en Ciencias Informáticas. Labora en el proyecto SGEF con el rol de Analista Principal. Instructora Recién Graduado del curso 2009-2010.

Correo electrónico: apperez@uci.cu

Datos del autor:

Nombre: Leanet Broche Ceballo

Correo electrónico: lbroche@estudiantes.uci.cu

Datos del autor:

Nombre: Julio Adalberto Leyva.

Correo electrónico: jleyvag@estudiantes.uci.cu

AGRADECIMIENTOS

A mi abuela porque sé que desde donde está, disfruta este logro mío con orgullo y afán.

A mi viejito lindo, por ser una de las cosas más bella que me ha dado la vida. Te quiero mucho mi abue.

A mi madre bella, por hacer papel de dos en todos estos 23 años, por sentir mi dolor y mis alegrías en esas llamadas de desesperación, por ser la guía que me ha llevado hacer lo que soy.

A mi hermanita chula, a ti te agradezco por esas carticas tan bellas que me enviabas que tanto me hacían llorar de alegría, por la alegría que me causaban tus llamadas cuando mis ánimos eran de no hablar con nadie, eres mi más preciado tesoro.

A mi padrastro Yeyo por hacer el papel de un padre y cuidar de la casa cuando más lo necesitábamos.

A Alberto por querer mucho a la persona que me ha dado la vida, por llegar cuando más lo necesitó.

A toda mi familia porque sé que siempre han estado pendiente de mí y orgullosos.

Le agradezco mucho a Yailen y a sus tíos Margo y Nando, por ser una familia más, por acogerme en su casa como una sobrina.

A Nany porque si no hubiera sido por ella quizás hoy no estuviera aquí presentando este trabajo, por apoyarme en mis momentos de desesperación que no sabía qué hacer cuando me quede sin tesis, gracias por tu apoyo.

A todas las amistades con la que he compartido todos estos años, Yailen, Yami, Ligia, Nany, Titi, Liliana, Mary y Maye las compañeras de 1 año.

A mi antigua compañera de tesis Dasiely porque aprendí muchas cosas buenas de ellas y a mi nuevo compañero de tesis por acogerme en su trabajo, por darme su apoyo y por entenderse también conmigo.

A mis compañeros Yordanis y Dasiely por todo el ánimo del mundo que me dieron cuando nos encontrábamos lejos de la familia, muchas gracias a ustedes.

A Armando por su apoyo incondicional, por quererme y ayudarme, por llegar a mi vida con su alegría cuando más lo necesité.

A todos los profes que me han ayudado, en especial a Yari, Dailin y Yiset.

AGRADECIMIENTOS

Gracias...

A mi mamá y a mi abuela, por ser lo más grande que tengo, por esforzarse tanto para que esté donde estoy.

A mi otro papá, Alvarito, por formarme desde que era un niño, por ayudarme y apoyarme siempre en todo.

A mi abuelo por sus consejos, su cariño y por ser un ejemplo para mí.

A mis hermanitos por contagiarme con su alegría.

A mi novia Daysi, casi esposa, por ayudarme en todo y apoyarme en los malos momentos, por tenerme siempre presente.

A mis cuñadas, Alys y Nelly por confiar en mí y darme tanto apoyo.

A mis amigos Lary y Enma, por estar a mi lado y preocuparse siempre por mí.

A todos mis compañeros y ponchadores de estos cinco años.

A Yenier y Annaliet por ayudarme tanto.

DEDICATORIA

A mis abuelos, mis padres Mabel y Eudelio, a mi hermana bella, porque han sido, son y serán mi guía e inspiración.....

A toda mi familia porque sé que un triunfo mío es orgullo y victoria de ellos también.....

Leanet

Le dedico este trabajo a mi mamá, a mis abuelos, mis hermanitos, a mi novia y a la memoria de mi papá.

Julio

RESUMEN

El gobierno cubano realiza un gran esfuerzo para informatizar gradualmente todos los sectores de la sociedad y uno de los que han sido favorecidos con esta política es el Sector Jurídico. Como parte de esta decisión se encuentra la Universidad de las Ciencias Informáticas encomendada a desarrollar un Sistema para la Informatización de la Gestión de las Fiscalías, dicho sistema tiene como objetivos agilizar el proceso de la toma de decisiones, ayudar a reducir en tiempo el trabajo que se lleva a cabo en las oficinas de las fiscalías y elevar el grado de conformidad por parte de los ciudadanos cubanos.

El propósito fundamental de este trabajo es realizar el análisis y diseño del proceso Verificación Fiscal que controla como se lleva a cabo cada etapa en las que se encuentra enmarcado.

Este documento consta de 4 capítulos. El primero aborda lo relacionado con la fundamentación teórica realizada para el presente trabajo. El capítulo 2 todo lo relacionado con la fase de Modelado del Negocio. En el capítulo 3 se desarrolló la fase de Requisitos y en el capítulo 4 se presenta el Análisis y Diseño. Aclarar que en cada capítulo se incluye su validación.

Con la presente propuesta de solución se da paso a la implementación para lograr que el trabajo en las fiscalías se haga de una manera ágil y que la realización de los procesos se haya reducido al mínimo en tiempo y ejecución, que tenga gran aceptación por parte de los fiscales y que el objetivo del mismo se haya logrado al optimizar el proceso Verificación Fiscal.

PALABRAS CLAVES

Negocio, Caso de Uso, diseño, métrica, patrones, requerimientos, Verificación Fiscal.

ÍNDICE

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA	14
Introducción.....	14
1.1. Conceptos.....	14
1.2. Soluciones informáticas para el Sector Judicial.....	15
1.3. Proceso de Desarrollo de Software	16
1.3.1. Metodología de desarrollo	16
1.3.2. Lenguajes de Modelado	20
1.3.3. Herramientas CASE	23
1.3.4. Herramienta para la elaboración de los prototipos no funcionales del sistema.....	26
1.3.5. Métricas.....	27
Conclusiones parciales.....	28
CAPÍTULO 2: MODELACIÓN DEL NEGOCIO	29
Introducción.....	29
2.1. Modelado del Negocio	29
2.2. Técnicas utilizadas para comprender el negocio	30
2.3. Proceso del Negocio.....	30
2.4. Patrones para el modelado del negocio.....	35
2.5. Validación del negocio.....	37
CAPÍTULO 3: ESPECIFICACIÓN DE REQUISITOS	38
Introducción.....	38
3.1. Requisitos	38

3.2.	Aplicación de las técnicas para la captura de requisitos	38
3.3.	Requisitos Funcionales	39
3.4.	Validación de los Requisitos	44
3.5.	Diagrama de Caso de Uso del Sistema.....	48
3.6.	Requisitos no Funcionales	49
	Conclusiones Parciales	53
	CAPÍTULO 4: ANÁLISIS Y DISEÑO	54
	Introducción.....	54
4.1.	Modelo de Diseño.....	54
4.2.	Principios del diseño.....	54
4.3.	Modelo Conceptual	55
4.4.	Arquitectura definida.....	57
4.5.	Diagramas de clases de diseño	59
4.6.	Patrones de diseño.....	61
4.7.	Métricas para la validación del diseño	63
	Conclusiones Parciales	69
	CONCLUSIONES GENERALES	70
	RECOMENDACIONES	71
	BIBLIOGRAFÍA	72
	ANEXOS	¡ERROR! MARCADOR NO DEFINIDO.
	GLOSARIO DE TÉRMINO	76

ÍNDICE DE TABLAS

Tabla 1: Involucrados, Artefactos y Subproceso.....	35
Tabla 2: Patrones para el modelado del negocio.....	36
Tabla 3: Grupo de revisores	46
Tabla 4: Estereotipos y Relaciones	60
Tabla 5: Utilización de los patrones GRASP	62
Tabla 6: Utilización de los patrones GoF	63
Tabla 7: Resultado de la métrica tamaño de clase.....	67
Tabla 8: Tamaño del umbral.....	68

ÍNDICE DE FIGURAS

Figura 1: Ciclo de Vida de RUP	17
Figura 2: Fases de XP.	18
Figura 3: Diagrama del Negocio del Subproceso Ejecutar la Verificación Fiscal	31
Figura 4: Grado de especificidad de los requisitos	47
Figura 5: Diagrama de Caso de Uso del Sistema.....	48
Figura 6: Modelo Conceptual.	56
Figura 7: El flujo de trabajo de Symphony	58
Figura 8: Diagrama de clase del diseño Gestionar Expediente de la Verificación Fiscal.....	61

INTRODUCCIÓN

Hoy en día la sociedad está regida por normas jurídicas, procedimientos y leyes que surgieron con el objetivo de garantizar la estabilidad de cualquier estado. Este conjunto de normas y procedimientos al que se le denomina ordenamiento jurídico, está en correspondencia con muchos factores, fundamentalmente las costumbres, tradiciones y normas. En Cuba es la Fiscalía General de la República órgano del Estado que tiene como objetivos fundamentales llevar a cabo el control y la preservación de la legalidad sobre la base de la vigilancia del estricto cumplimiento de la Constitución, las leyes y demás disposiciones legales, por los organismos del Estado, entidades económicas - sociales y por los ciudadanos, además la promoción y el ejercicio de la acción penal pública en representación del Estado. (Cuba, 1999)

La FGRC¹ en virtud de aprovechar el desarrollo vertiginoso de la informática y las comunicaciones ha identificado la necesidad de informatizar sus principales procesos. Esto le permitirá un mayor control en las informaciones obtenidas y generadas, facilidades en su almacenamiento, mejoramiento del recurso tiempo, seguridad de la información que se maneja, así como otras ventajas inherentes al proceso de informatización, las cuales se revierten en mejorar la gestión interna del trabajo.

Con el objetivo de que se pueda llevar a cabo lo antes expuesto surge el Sistema de Informatización de la Gestión de las Fiscalías para informatizar todo el Sector Fiscal. Este sistema ha sido desarrollado en la Universidad de las Ciencias Informáticas para obtener resultados tangibles. Fue dividido en dos fases debido a la magnitud de sus procesos, dentro de los procesos existentes en la fase II se encuentra el proceso Verificación Fiscal.

Este proceso, se ejecuta de forma manual implicando gran esfuerzo por parte del personal que trabaja en las fiscalías, provocando un desgaste de la fuerza de trabajo, pues existe carencia de especialistas. Además, el procedimiento desarrollado es complejo por lo que se hace difícil llevar un control y seguimiento real de todos los casos tramitados diariamente, influyendo negativamente en la prontitud y actualización de la información brindada.

Actualmente la obtención de los datos relacionados con la Verificación Fiscal se realiza desde el nivel de las Fiscalías Municipales y Provinciales hasta el nivel de la FGRC, para completar el flujo de este proceso

¹ Fiscalía General de la República de Cuba.

se debe enviar la información vía telefónica y en algunos de los casos mediante ficheros Excel vía correo electrónico. El manejo de todos los datos recibidos es engorroso y genera gran cantidad de tiempo su procesamiento y disponibilidad. Esto dificulta obtener los resultados finales.

A pesar de que el almacenamiento de la documentación se realiza de forma correcta, con el paso del tiempo, ésta se deteriora y provoca dificultades con su posterior uso al no contar los fiscales con una base de datos que permita conocer determinada información en un momento dado.

La incorporación de este proceso al Sistema de Informatización de la Gestión de las Fiscalías implica que la misma se acople al sistema ya implementado, por lo que se deben transformar del lenguaje natural todos los aspectos correspondientes al proceso Verificación Fiscal en artefactos de software para facilitarle la implementación a los futuros desarrolladores.

Surge entonces como **Problema de investigación:** ¿Cómo transformar las necesidades del cliente en elementos que contribuyan a facilitar la implementación del proceso Verificación Fiscal para el Sistema de Informatización de la Gestión de las Fiscalías?

Donde se tiene como **Objeto de Estudio:** El proceso de desarrollo de software.

Definiendo como **Objetivo General:** Generar los artefactos necesarios del Análisis y Diseño del proceso Verificación Fiscal que faciliten su implementación para SIGEF².

Teniendo como **Campo de Acción:** El análisis y diseño del proceso Verificación Fiscal.

Entonces se **defiende la Idea** de que: Generando los artefactos necesarios correspondientes al Análisis y Diseño del proceso Verificación Fiscal, se podrá transformar las necesidades de los clientes en elementos que faciliten su implementación para el Sistema de Informatización de la Gestión de las Fiscalías.

Obteniendo los siguientes **Objetivos Específicos:**

- Elaborar el marco teórico de la investigación.
- Desarrollar el modelado del negocio generando los artefactos necesarios.

²Sistema de Informatización de la Gestión de las Fiscalías.

- Realizar la identificación y especificación de los requisitos, generando los artefactos necesarios.
- Desarrollar el diseño generando los artefactos necesarios.
- Validar los artefactos generados.

Para dar solución a los objetivos planteados anteriormente se trazan las siguientes **Tareas de la Investigación:**

- Estudio y análisis de las principales metodologías de desarrollo del software.
- Estudio y análisis de Herramientas CASE³.
- Estudio y análisis de lenguajes de modelado, patrones y métricas para la validación.
- Realización de entrevistas a los especialistas funcionales de la institución con el objetivo de comprender el negocio.
- Estudio y análisis de la documentación de los procesos de la institución.
- Modelación del proceso del negocio.
- Descripción del proceso del negocio.
- Elaboración del mapa de procesos.
- Elaboración del modelo conceptual.
- Validación de los artefactos en la modelación del negocio aplicando métricas y técnicas como formas de validación.
- Identificación de requisitos funcionales a partir del proceso del negocio.
- Elaboración de la descripción de los requisitos del software.
- Realización del diagrama de caso de uso del sistema.
- Realización de la descripción textual de los casos de uso.
- Realización de prototipos de interfaz de usuario para validar los requisitos.

³ Ingeniería de Software Asistida por Ordenador (Computer Aided Software Engineering).

- Elaboración del modelo del diseño.
- Validación de los artefactos de análisis y diseño, aplicando métricas.

Métodos y técnicas de investigación utilizados

Para validar metodológicamente la investigación se utilizaron los siguientes métodos:

Teóricos

- Histórico - Lógico: Se utilizó para analizar la trayectoria y evolución de la metodología de desarrollo de software y las herramientas que se utilizaron durante el trabajo.
- Método hipotético-deductivo: Se utilizó para la definición de la idea a defender de la investigación.
- Método de la modelación: Se utilizó para el diseño ya que es necesaria la elaboración de diagramas, figuras y otros artefactos importantes, mediante este método se pueden crear abstracciones con el propósito de explicar la realidad.

Empíricos

- Entrevista: Se utilizó para obtener información detallada acerca de cómo ocurre el proceso Verificación Fiscal que transita por la FGRC.

Estructura de la Tesis

El presente trabajo de diploma consta de 4 capítulos:

Capítulo 1. Fundamentación Teórica: Este capítulo aborda el estudio realizado para la construcción de la investigación, como lo es el estado de arte sobre los sistemas existentes en el mundo para la gestión fiscal. Se realizó un estudio sobre el proceso de desarrollo entiéndase Ingeniería de Requisitos, metodologías de desarrollo, herramientas CASE, lenguajes de modelado, herramientas para el desarrollo de prototipos no funcionales, así como técnicas y métricas utilizadas para medir la calidad de un sistema.

Capítulo 2. Modelado de negocio: En este capítulo se realiza una caracterización del negocio propuesto. Se exponen los artefactos generados producto del análisis del proceso Verificación Fiscal y las descripciones de éstos, haciendo uso de patrones para el modelado del negocio. Se emplean técnicas y métricas para la validación de la propuesta de solución.

Capítulo 3. Especificación de requisitos: En este capítulo a partir de los resultados obtenidos en el capítulo anterior y la aplicación de la Ingeniería de Requisitos, se realizan las actividades relacionadas con la especificación de requisitos, técnicas utilizadas para la identificación de ellos, artefactos y descripciones de los mismos, así como técnicas y métricas utilizadas para su validación.

Capítulo 4. Diseño: En este capítulo se realiza el modelado del diseño, donde se exponen los artefactos que lo componen, como diagramas de clases, Modelo del diseño. Se aplican, además, los patrones de diseño y las métricas que se utilizan para evaluar la calidad del diseño realizado.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

Introducción

En el presente capítulo se abordarán temas sobre cómo realizar un correcto análisis y diseño, las bases necesarias para el desarrollo de éste y conceptos importantes para el dominio del problema, así como un estudio detallado sobre los sistemas que existen en el mundo para la informatización del proceso Verificación Fiscal. Se realiza un estudio sobre el proceso de desarrollo de software, Ingeniería de Software, herramientas CASE, lenguajes de modelado, además de las herramientas para el desarrollo de los prototipos de interfaz de usuario no funcionales como base para la construcción de la presente investigación. Se abordará la Ingeniería de Requisitos, patrones de casos de uso, patrones de diseño, técnicas y métricas utilizadas para la validación de la propuesta de solución presentada.

Es bueno destacar que se hará alusión a términos como Fiscalía General de la República de Cuba (FGRC), Verificación Fiscal, Expediente de la Verificación Fiscal, los cuales se hacen necesarios definir dentro del marco de la misma para facilitar su entendimiento y comprensión.

1.1. Conceptos

Fiscalía General de la República de Cuba

La Fiscalía General de la República de Cuba es el órgano del estado que tiene como objetivo fundamental llevar a cabo el control y la preservación de la legalidad, la Constitución, las leyes y demás disposiciones legales vigentes en los organismos del estado, entidades económicas-sociales y ciudadanos, además la promoción y el ejercicio de la acción penal pública en representación del estado. Constituye una unidad orgánica subordinada únicamente a la Asamblea Nacional del Poder Popular y al Consejo de Estado. (Justicia, 2008)

Verificación Fiscal

La Verificación Fiscal constituye el método principal que emplea la Fiscalía para comprobar el cumplimiento de la Constitución, las leyes y demás disposiciones legales en las instituciones y sus entidades independientes, formulando los pronunciamientos que resulten procedentes para que se restablezca la legalidad quebrantada, accionar contra los infractores, actuando como medio de educación jurídica y de prevención de nuevas infracciones. La Verificación Fiscal se sustancia mediante las distintas

acciones y diligencias que realizan los fiscales, con el auxilio de especialistas y peritos en la materia objeto de inspección, organizándose en orden cronológico en el expediente. (Cuba, 1999)

Expediente de Verificación Fiscal

Conjunto de datos o conjunto de incidencias recopilados por el fiscal durante el trámite de un proceso Verificación Fiscal. (Cuba, 1997)

1.2. Soluciones informáticas para el Sector Judicial

A continuación se hace un estudio de algunos sistemas informáticos utilizados en el sector judicial a nivel mundial con el objetivo de determinar si estas aplicaciones se pueden utilizar para la informatización de la Fiscalía General de la República de Cuba y así no incurrir en gastos de desarrollo de un nuevo sistema. Además se pretende con este análisis adquirir conocimientos sobre las tendencias actuales para el desarrollo de sistemas informáticos del sector judicial.

SICC: En la actualidad, la Fiscalía General del Estado de España dispone de un sistema informático denominado Sistema de Inspección Control y Consulta, herramienta que permitirá acceder a información procesal de asuntos tramitados en las distintas fiscalías puesto que cuenta con una base de datos centralizada e incorporará un cuadro de mandos para examinar, desde diversas perspectivas estratégicas, la actuación tanto de las fiscalías como de los órganos judiciales con los que éstas se relacionan. (estado, 2009)

SAF: El Sistema de Apoyo a los Fiscales permite la tramitación y gestión de causas criminales, posibilita la entrega de la información en línea y a nivel nacional, de cualquier antecedente, procedimiento o instancia. Es la herramienta clave en la Reforma Procesal Penal en Chile. Este sistema está organizado en módulos como el de ingreso de denuncias, asignación de casos, gestión de causas, administración de documentos, custodia, consultas e informes, indicadores de gestión y estadísticas que funcionan integradamente. (Sonda, 2010)

SOFTLEX: En Cuba se desarrolló una solución informática denominada SOFTLEX, en el seno de la Sociedad Cubana de Derecho e Informática, que da tratamiento a normativas y tiene un módulo de seguimiento de asuntos, hasta cierto punto configurable por el usuario, que recoge una serie de datos útiles de los procesos.

Hasta la fecha no se tiene conocimiento de la existencia de ninguna solución informática de envergadura y profundidad desarrollada sobre software libre con un grado de madurez elevado en la gestión de información y ayuda en la toma de decisiones a nivel nacional para los fiscales, por lo que se aprecia la necesidad latente de contar con un sistema informático para la gestión de las fiscalías que dé respuesta a esta problemática.

Luego de haber analizado los diferentes sistemas descritos anteriormente se llegó a la conclusión de que no constituyen una solución robusta para el proceso Verificación Fiscal por la singularidad de las leyes judiciales cubanas, además de haber aportado varios aspectos a tener en cuenta para el desarrollo del mismo. Se decide incorporar este proceso como una nueva funcionalidad dentro del Sistema de Informatización de la Gestión de las Fiscalías.

1.3. Proceso de Desarrollo de Software

El Proceso de Desarrollo de Software forma la base para el control de la gestión de los proyectos de software. El mismo establece el contexto en el cual se aplican los métodos técnicos, se generan los productos del trabajo (modelos, documentos, métodos, datos, reportes, formatos), se establecen los fundamentos, se asegura la calidad y el cambio se maneja de manera apropiada. (Pressman, 2005)

1.3.1. Metodología de desarrollo

RUP

Rational Unified Process o Proceso Unificado de Desarrollo es un proceso de ingeniería de software que suministra un enfoque para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que satisfaga la necesidad del usuario final dentro de un tiempo y presupuesto previsible. Es una metodología de desarrollo iterativo enfocada hacia los casos de uso, manejo de riesgos y el manejo de la arquitectura. RUP mejora la productividad del equipo al permitir que cada miembro del grupo, sin importar su responsabilidad específica, acceda a la misma base de datos de conocimientos. Esto hace que todos compartan el mismo lenguaje, la misma visión y el mismo proceso acerca de cómo desarrollar software. Es un proceso de desarrollo de software cuyo ciclo de vida se caracteriza por ser: (GALLEGO, 2007)

- Dirigido por CU⁴.
- Centrado en arquitectura.
- Iterativo e incremental.

Ciclo de Vida

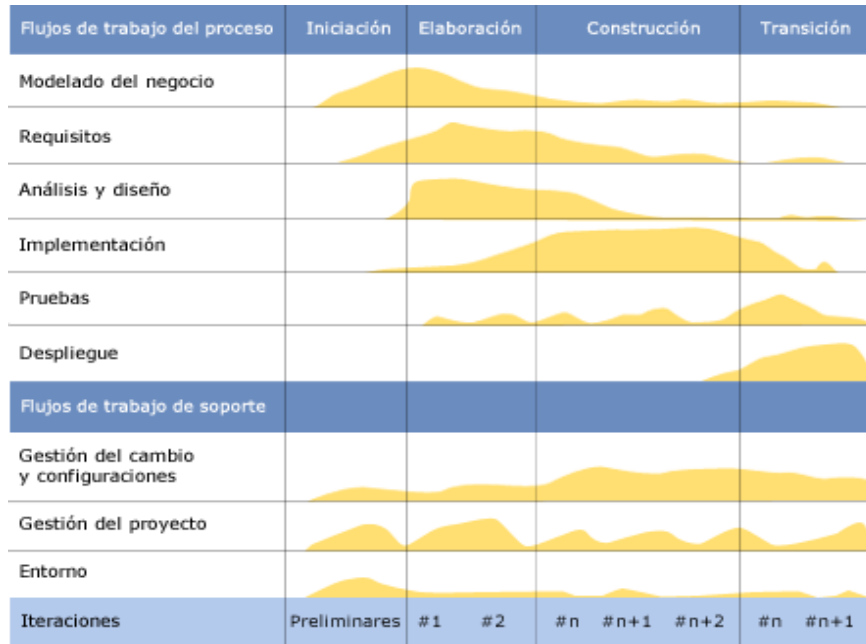


Figura 1: Ciclo de Vida de RUP

Programación Extrema⁵

XP es una metodología ágil, centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios.

⁴ Caso de Uso: “Un caso de uso es una secuencia de interacciones entre un sistema y alguien o algo que usa alguno de sus servicios.” (Ceria, 2001)

⁵ En lo adelante XP por sus siglas en inglés (Extreme Programming)

Surgió como respuesta y posible solución a los problemas derivados del cambio en los requerimientos y se plantea como una metodología a emplear en proyectos de riesgo, siendo útil en el aumento de la productividad de un proyecto. Esta metodología tiene como ventaja la comunicación que existe entre sus desarrolladores, además de la programación en pareja que permite que constantemente se esté revisando el código y como principal desventaja, no brinda documentación para ser usada en otro momento. (Letelier, 2008)

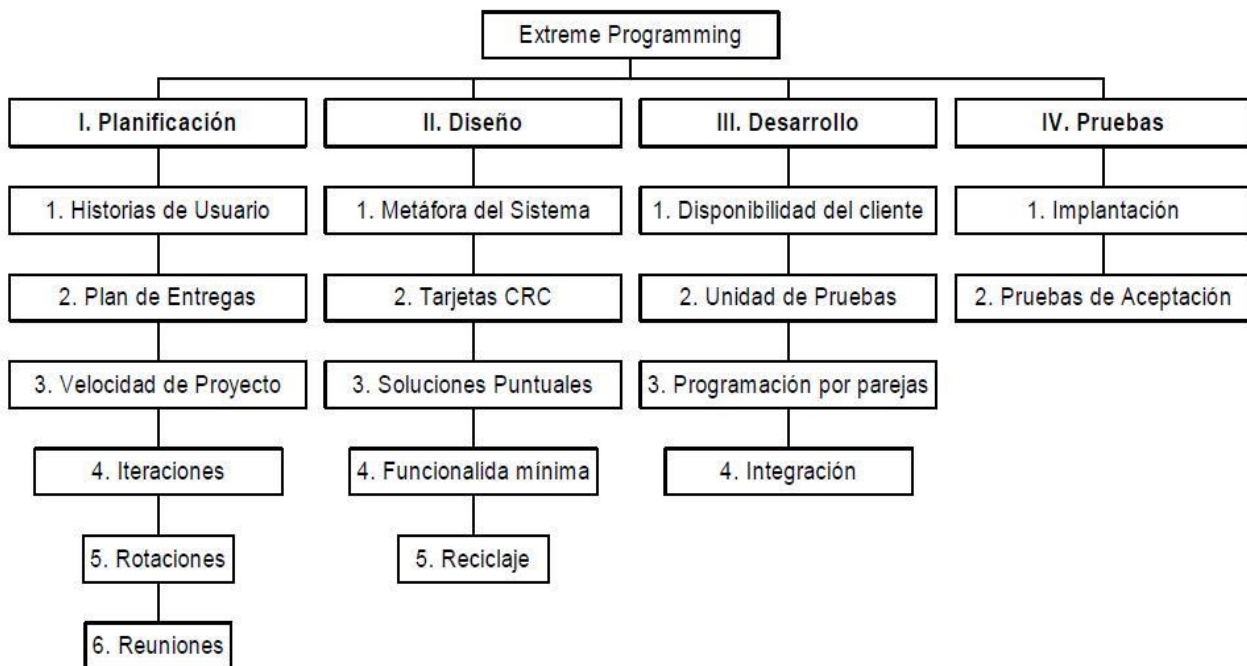


Figura 2: Fases de XP.

Programa de Mejora

La Universidad de las Ciencias Informáticas, ha tenido desde sus comienzos la meta de alcanzar altos índices de producción e ingresos en la esfera de la informática, por lo que se involucró desde finales del

año 2008 en un Proceso de Mejora de Software SPI⁶ basado en el modelo CMMI⁷, cuyos objetivos han estado enfocados en:

- Mejorar los procesos, métodos, tecnología y la calidad de los proyectos a partir de la incorporación de buenas prácticas propuestas por el modelo CMMI.
- Obtener una evaluación del nivel 2 de CMMI. (Calisoft, 2009)

CMMI: Es un modelo de referencia para el crecimiento de capacidades y madurez, que se enfoca tanto en procesos de Administración como de Ingeniería de Sistemas y Software. Con su instauración se espera alcanzar beneficios como:

- Calendarios y presupuesto predecibles en los proyectos.
- Mejora de ciclo de vida dentro del desarrollo de software.
- Mayor productividad.
- Mayor calidad de los productos y servicios que ofrece la universidad a sus clientes y por ende, la satisfacción de los mismos.
- Mejorar la moral del personal que labora en el centro. (Calisoft, 2009)

Para el desarrollo del sistema se ha determinado utilizar el Programa de Mejora basado en el modelo CMMI, éste en su nivel 2 explica que aspectos deben cumplirse en determinadas áreas de procesos y el Programa de Mejora establece como cumplir con estos aspectos. Para ello propone un ciclo de vida compuesto por varias fases. En el desarrollo del proceso Verificación Fiscal se hará uso de las siguientes fases:

Modelado del Negocio: Es la fase destinada a comprender los procesos de negocio de una organización. En ella se comprende cómo funciona el negocio que se desea informatizar para tener garantías de que el software desarrollado va a cumplir su propósito. Para la descripción y modelado de negocio pueden ser utilizadas diferentes técnicas como el Modelado de Casos de Uso del Negocio y Business Process

⁶Proceso de Mejora de Software (Software Process Improvement).

⁷ Integración de Modelos de Madurez de Capacidades (Capability Maturity Model Integration).

Modeling Notation (BPMN⁸). En esta fase se generan los artefactos: Mapa de Procesos de Negocio y Descripción de los Procesos del Negocio.

Requisitos: El esfuerzo principal de esta fase es desarrollar un modelo del sistema que se va a construir. Incluye un conjunto de casos de uso, servicios que describen todas las interacciones que tendrán los usuarios con el software, éstos responden a los requisitos funcionales del sistema. Además la especificación de requisitos incluye requisitos no funcionales. En esta fase se generan los artefactos: Especificación de Requisitos de Software, Descripción de Requisitos de Software, Diagrama de Casos de Uso, Especificación de Casos de Uso, Modelo Conceptual.

Análisis y Diseño: Durante ésta fase es modelado el sistema y su forma (incluida su arquitectura) para que soporte todos los requisitos, incluyendo los requisitos no funcionales. Esto contribuye a una arquitectura sólida y estable que se convierte en un plano para la implementación. Los modelos desarrollados en ésta fase son más formales y específicos de una implementación. (Acosta, 2008). En ésta fase se genera el artefacto Diagrama de Clases de Diseño y el Modelo de Diseño.

Además se decide utilizar este Programa de Mejora en el proyecto SIGEF ya que éste se rige por alcanzar un alto grado de capacidades y madurez es sus procesos y quedaría certificado dentro de los proyectos que obtienen finalmente ese valor de madurez.

1.3.2. Lenguajes de Modelado

El proceso usado para capturar, documentar y rediseñar procesos de negocio es llamado modelado de procesos. Cuando un proceso es modelado, con ayuda de una representación gráfica, pueden apreciarse con facilidad las interrelaciones existentes entre distintas actividades, el análisis para cada una de ellas y la definición de los puntos de contacto con otros procesos. A continuación se muestran algunos de los lenguajes más usados en la actualidad:

IDEF⁹

⁸ Notación para el Modelado del Negocio (BPMN).

⁹ Conocido por sus siglas en inglés, que significan Integrated Definition Methods, que se traducen al español como: Métodos Integrados de Información.

IDEF es una técnica de modelado de sistemas que comprende desde la modelación de la información de un sistema u organización hasta el análisis y diseño orientado a objetos. La familia de técnicas de modelado IDEF fue desarrollada como un conjunto de notaciones formales para representar y modelar procesos y estructuras de datos en una forma integrada, es decir, proveer técnicas de modelado simples y formales que permitan describir, analizar y evaluar distintos puntos de vista de un sistema. Un modelo IDEF 0, permite representar las actividades que conforman un sistema de forma jerárquica. Está compuesto por un conjunto de diagramas que facilitan la descripción de las funciones especificadas en el nivel superior. En las vistas superiores del modelo la interacción entre las actividades representadas permite visualizar los procesos fundamentales que sustentan la organización. Los elementos gráficos utilizados para la construcción de los diagramas IDEF 0 son cuadros y flechas.

De manera general IDEF0 tiene las siguientes características:

- Es una técnica genérica que permite modelar gráficamente procesos de sistemas de diferentes propósitos y a cualquier nivel de detalle.
- Es comprensivo y expresivo, capaz de representar gráficamente reglas y una amplia variedad de negocio.
- Es un lenguaje simple y coherente.
- Realiza la comunicación entre los analistas de sistemas, los desarrolladores y los usuarios por la facilidad de aprender sus distintas representaciones y por su énfasis en la exposición jerárquica del detalle.
- Puede ser generado por una variedad de herramientas de modelado gráfico de computadora.

Entre las desventajas que posee están:

- Es limitado en la simbología, el único símbolo utilizado es una caja rectangular que representa una actividad o función.
- No permite definir responsabilidades fácilmente.
- Requiere de una amplia formación y experiencia tanto de la persona que lo elabora como del que interpreta. (Rolando Rodríguez Andres, 2008)

Lenguaje Unificado de Modelado (UML)

Es un sistema de notación que se ha estandarizado en el mundo dentro del proceso de desarrollo de software. Permite el desarrollo de distintos tipos de diagramas, cada uno de los cuales representa el sistema a especificar, analizar o diseñar desde distintas perspectivas. Es utilizado para modelar la información del sistema basado en concepto de objetos. Los objetivos de UML son muchos pero se pueden sintetizar sus funciones: (Orallo, Enrique Hernández, 2007)

- Visualizar: Permite expresar de una forma gráfica un sistema de forma que otro lo pueda entender.
- Especificar: Permite especificar cuáles son las características de un sistema antes de su construcción.
- Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.
- Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

Para el desarrollo del presente trabajo se utilizará UML como lenguaje de modelado para modelar los artefactos generados durante el proceso Verificación Fiscal, por ser un lenguaje del que se tiene gran dominio por parte del equipo de desarrollo, reduciendo con ello los costos de tiempo que deben emplearse en capacitación para el empleo de otro lenguaje. Es importante resaltar que UML es un lenguaje para especificar y no para describir métodos o procesos, debido a que en las fiscalías se trabaja por áreas y éstas a su vez contienen procesos, resulta factible realizar el modelado del negocio por procesos, y se estaría de una manera en pleno entendimiento con el cliente, al realizar una similitud entre la actividad que realizan los fiscales a la hora de realizar un proceso y lo modelado por los analistas. Esta facilidad no lo brinda UML por ello se decide solo utilizarlo en el sistema.

Notación para el Modelado del Negocio (BPMN)

BPMN es la notación gráfica para modelar los procesos del negocio. La notación ha sido diseñada específicamente para coordinar la secuencia de procesos y los mensajes que fluyen entre los diferentes procesos participantes, incluyendo la unión con el diseño y la implementación. Su principal objetivo es:

- Resolver las dificultades de comunicación que tiene el lenguaje común.
- Proporciona un método normalizado para representar procesos de negocio.
- Facilita su entendimiento debido a la poca complejidad de su notación.

- Proporciona un lenguaje común entre los usuarios de negocio y los técnicos.
- Facilita la diagramación de los procesos de negocio. (White, 2008)

Se decide utilizar BPMN como notación para el modelado del negocio, por ser fácil, legible y entendible. Permite modelar el proceso Verificación Fiscal de un forma organizada y centralizada, proporcionando un entendimiento común entre los clientes y el equipo de trabajo, al brindar mayor flexibilidad y agilidad para adaptarse a los cambios. A través del modelado de procesos puede lograrse un mejor entendimiento de los negocios, creando la oportunidad de mejorarlos y estableciendo un puente estandarizado para disminuir la brecha entre el proceso de negocio y su implementación. Se decidió utilizarlo ya que ésta notación permitió modelar el proceso Verificación Fiscal como mismo se llevaba a cabo en la vida real, es decir, existía una gran similitud entre lo realizado por los analistas utilizando un medio informático, al trabajo realizado por un fiscal. Esto permitió que existiera una mayor flexibilidad a la hora de realizar los cambios y a que los fiscales se sintieran identificados fácilmente en lo realizado por los analistas. Además esta notación está incluida como propuesta del Programa de Mejora.

1.3.3. Herramientas CASE

Las herramientas CASE, son aplicaciones que facilitan el desarrollo de software, reduciendo el esfuerzo, el costo y el tiempo, además de estructurar la documentación asociada a los artefactos generados. A continuación se exponen las principales características de tres de las herramientas CASE más utilizadas en el mundo actualmente. (CEDS, 2009)

Visual Paradigm (VP)

Es una herramienta CASE que utiliza UML como lenguaje de modelado. Soporta el ciclo de vida completo de desarrollo de software, ayuda a una rápida construcción de aplicaciones de calidad y a un menor costo. Permite modelar todos los tipos de diagramas de clases, código inverso, generar documentación y código desde diagramas.

Dentro de las funcionalidades que soporta el Visual Paradigm están:

- Administración de requisitos.
- Modelado de procesos del negocio.
- Modelado de base de datos.

- Generación de código.
- Ingeniería inversa.

Se integra con herramientas Java¹⁰ como:

- Eclipse¹¹.
- JBuilder¹².
- NetBeans/sunONE¹³.

Visual Paradigm da la posibilidad de lograr una mejor integración entre todos los involucrados en el desarrollo del software, brindando la posibilidad de organizar los diagramas y documentación asociada al desarrollo del proyecto. (Visual Paradigm International, 2005)

Rational Rose Enterprise Edition

Rational Rose soporta UML, proporciona un lenguaje común de modelado que facilita la creación de software con mayor calidad. Entre las características que posee se encuentran: (ECURED, 2008)

- Soporte de Ingeniería hacia adelante y/o Ingeniería Inversa para algunos de los conceptos más comunes de Java 1.5.
- Permite la generación de códigos, con capacidad de generación de sincronización de modelo – código configurable.
- Capacidad de análisis de calidad de código.
- Capacidad de crear definiciones de tipo de documento XML¹⁴ (DTD¹⁵).
- Soporte Enterprise Java Beans¹⁶™ 2.0.

¹⁰Java: Lenguaje de programación orientado a objetos.

¹¹Eclipse: Es un entorno integrado de desarrollo para Java muy potente.

¹²JBuilder: En un entorno integrado de desarrollo para el lenguaje de programación Java de Borland.

¹³NetBeans/sunONE: Entorno integrado de desarrollo más popular, permite desarrollar software de forma ordenada y rápida.

¹⁴XML: Lenguaje de anotación extensible.

¹⁵DTD: Definición de Tipo de Documento.

¹⁶Enterprise Java Beans: Componentes del contexto de servidor.

- Software propietario.

Esta herramienta propone la utilización de cuatro tipos de modelo para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software.

Enterprise Architect

Es una herramienta CASE orientada a objetos que provee su alcance para el desarrollo de sistemas, administración de proyectos y análisis de negocios. Maneja totalmente el ciclo de vida de desarrollo de software, utilizando el UML como lenguaje de modelado. Facilita y soporta el levantamiento de requerimientos, el análisis y diseño, las pruebas y mantenimiento del software en desarrollo.

Dentro de las funcionalidades que soporta el Enterprise Architect están:

- Diagramas UML.
- CU, Modelos Lógico, Dinámico y Físico.
- Extensiones personalizadas para modelado de procesos.
- Documentación de alta calidad compatible con MS Word.
- Modelado de Datos.
- Ingeniería directa de base de datos e ingeniería inversa de base de datos.
- Soporte de pruebas.
- Multi-usuario, con sistema de seguridad.
- Se integra a través de plugins con herramientas como: Eclipse, Visual Estudio.Net. (SYSTEMS, 2000-2007)

Como herramienta CASE se decide utilizar Visual Paradigm ya que puede integrarse con UML, BPMN y PHP, lenguajes y notaciones que serán utilizados para el desarrollo del sistema a realizar, UML para el sistema, BPMN para modelar el proceso y PHP porque es el lenguaje de programación que soporta el framework de desarrollo. La herramienta es multiplataforma, robusta y de fácil uso. Además permite crear diferentes tipos de diagramas que son realizados de una manera precisa y clara para su entendimiento,

como es el caso de los modelos de proceso del negocio. Esta herramienta a pesar de ser propietaria la UCI cuenta con una licencia para su utilización legal y soporta todo el ciclo de vida de un software. Además está incluida como propuesta del Programa de Mejora.

1.3.4. Herramienta para la elaboración de los prototipos no funcionales del sistema

Un prototipo es una pequeña muestra, de funcionalidad limitada, de cómo sería el producto final una vez terminado. La técnica del prototipado aplicado al desarrollo de productos web tiene muchas utilidades y aplicaciones, puesto que permite que se puedan verificar los requerimientos del producto antes de que se inicie su desarrollo. Permite fijar presentaciones en las que se puede solicitar la aprobación del prototipo del software, así como: especificar el contenido o detectar errores de diseño antes de iniciar el proceso de desarrollo.

Axure RP

Axure RP es una herramienta que permite diseñar borradores y prototipos de páginas web de forma eficiente. Cuenta con las herramientas necesarias para diseñar una página web previamente a ser programada. Además, permite distribuir las tareas y realizar anotaciones en el proyecto, aunque los proyectos deben ser finalizados en una herramienta que permita programar páginas web, ya que Axure RP solo creará los borradores. Axure RP es una aplicación totalmente sencilla de manejar pero que sin duda arrojará grandes resultados si se maneja bien. Con Axure se puede ir editando, añadiendo y quitando elementos con suma facilidad para ir viendo el trabajo previamente y así tener una página web bien maquetada y con grandes aspectos. (Help, 2010).

Irise Professional Edition

Es una herramienta de alto alcance, ya que no solo está especializada en prototipos, sino para detallar requisitos, escenarios, workflows, y definición funcional. Los prototipos que genera están orientados a maquetas navegables y a presentaciones. Irise es sumamente completo y si se trabaja en proyectos grandes con un alto número de perfiles distintos puede resultar muy útil emplearse como marco de trabajo. Tiene como desventaja que obliga a un estilo gráfico limitado, no se pueden usar librerías de elementos y los elementos de interfaz no se pueden personalizar. Tiene una lógica de publicación que permite lanzar los prototipos como documento HTML, lo que puede ser muy útil para generar maquetas

sin tener que tirar una sola línea de código. Otra desventaja que posee es que la licencia de la herramienta es muy costosa.

Microsoft Office Visio 2007

Microsoft Office Visio 2007 permite documentar, diseñar y comprender de forma visual el estado de los sistemas y procesos empresariales con una gran variedad de diagramas, de flujo de proceso empresarial, y diagramas de software, entre otros. Estos diagramas facilitan la visualización, el análisis y la comunicación de información compleja. (Visio, 2007)

Beneficios que brinda Microsoft Office Visio:

- Aumentar la productividad integrando diagramas con información de varias fuentes.
- Visualizar y actuar sobre información compleja mostrando datos en diagramas.
- Analizar datos y realizar de forma sencilla el seguimiento de tendencias, la identificación de problemas y la señalización de excepciones con diagramas dinámicos.
- Comunicar información compleja con nuevas plantillas y formas.
- Transmitir información de manera eficiente con diagramas de aspecto profesional.
- Comunicar el uso de diagramas y compartirlos con un amplio número de destinatarios.

El estudio anteriormente realizado permitió definir que se utilizará para la creación de los prototipos no funcionales del sistema el Axure RP en su versión 5.5, por sus facilidades de permitir la creación rápida de estos para páginas web, basándose en conceptos conocidos desde Visio hasta herramientas de diseño web, lo que significa que toma de cada una los aspectos esenciales para la confección de los prototipos. Permite que varias personas trabajen en un mismo proyecto a la misma vez. Esta herramienta es dinámica y permitió que existiera una navegación entre los prototipos, dando la posibilidad al cliente de tener una visión del sistema antes de su implementación.

1.3.5. Métricas

Las métricas constituyen una “medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo determinado” (Pressman, 2005).

Conclusiones parciales

En este capítulo se realizó un estudio de diferentes sistemas para la Informatización de la Gestión Fiscal existentes en el mundo, permitiendo identificar características comunes que pudieron ser de gran utilidad para el sistema a desarrollar. El estudio de lenguajes de modelado y herramientas tanto CASE como para el desarrollo de prototipos no funcionales permitió identificar cual de estos sería el utilizado para el desarrollo del sistema. Así como términos en cuanto al negocio que eran de suma importancia conocer.

CAPÍTULO 2: MODELACIÓN DEL NEGOCIO

Introducción

En el presente capítulo se explicarán aspectos que ayudarán a una mejor comprensión de las funcionalidades del proceso Verificación Fiscal para luego facilitar la implementación en sistema. Se centra la atención en el entendimiento del negocio, con el fin de obtener la información que brinda el cliente y los artefactos correspondientes a esta fase.

A partir de lo investigado en el capítulo anterior, en el presente se realiza el Modelo del Negocio correspondiente al proceso Verificación Fiscal, que se lleva a cabo en la Fiscalía General de la República de Cuba, lo que ayudará a comprender este proceso describiéndolo, identificando quiénes participan y las actividades que requieren de informatización.

2.1. Modelado del Negocio

El Modelado de Negocios se define como un proceso de representación de uno o más aspectos o elementos de una empresa, tales como su propósito, su estructura, su funcionalidad, su dinámica, su lógica de negocios y sus componentes. Es una actividad previa y complementaria al proceso de desarrollo de software que permite:

- Entender los problemas actuales en la organización o empresa, para identificar los aspectos a mejorar.
- Comprender la estructura y el dinamismo de la organización o empresa para la cual se va a desarrollar el sistema de software.
- Estudiar el impacto que pueden producir los cambios a nivel organizativo.
- Asegurar que los clientes, usuarios finales, desarrolladores y otros involucrados tengan una visión común de la organización.
- Obtener los requisitos del sistema de software.
- Entender como el sistema de software se ajusta a la organización. (Montilva, 2007)

2.2. Técnicas utilizadas para comprender el negocio

Para comprender el negocio se aplicaron las siguientes técnicas:

Entrevistas: Se realizaron entrevistas a los clientes (Fiscales del Departamento de Verificación Fiscal) Esp¹⁷. Teidy Colomar Zaldívar, por parte del equipo de analistas, con el fin de entender el negocio, las mismas fueron registradas en el documento Minuta de Reunión.

Análisis de documentación: Se realizó un estudio de las leyes y normas de Verificación Fiscal para determinar los procedimientos y las condiciones que se deben cumplir para llevar a cabo el proceso Verificación Fiscal. El estudio de estas documentaciones permitió comprender términos y aspectos esenciales que se manejaban en las fiscalías.

Tormenta de Ideas: Se realizaron talleres con especialistas de la FGR y analistas del proyecto para obtener una visión general de la recepción de documentos en el proceso Verificación Fiscal.

2.3. Proceso del Negocio

Las técnicas aplicadas para comprender el negocio permitieron que el equipo de analistas comprendiera como se llevaba a cabo el proceso Verificación Fiscal en la FGR, identificando que el mismo para su realización está comprendido en tres etapas fundamentales, Etapa Previa, Etapa de Ejecución y Etapa de Conclusión las cuales están constituidas por varios subprocesos.

Etapa Previa: En esta etapa se crean las condiciones necesarias para el desarrollo adecuado de la VF, realizando un conjunto de tareas fundamentales como son la confección del plan de trabajo, selección de los especialistas que se consideran necesarios para enfrentar la tarea, el estudio y análisis de las leyes y demás disposiciones.

Etapa de Ejecución: Es el período que comprende la llegada sorpresiva o coordinada del fiscal con los especialistas a la entidad, hasta que termina el trabajo de comprobación de la observancia de las leyes y demás disposiciones que se verifican. Durante esta etapa los fiscales pueden tomar comparencia, ocupar documentos u otros bienes. Excepcionalmente se otorgará una prórroga de hasta 30 días para el término de la tramitación de 60 días de la Verificación Fiscal.

¹⁷ Especialista (Esp.)

Etapa de Conclusión: Es la que culmina el proceso de ejecución de la Verificación Fiscal y comprende el análisis de los resultados del trabajo. La reunión de conclusión de la Verificación Fiscal ejecutada, la elaboración de la Resolución y la notificación de la Resolución emitida por el fiscal a la dirección administrativa de la Entidad Verificada y al nivel superior de ésta.

En las descripciones de los subprocesos correspondientes a la Verificación Fiscal se encuentran cómo es que se lleva a cabo cada una de estas etapas y se explican cada uno de los subprocesos correspondientes. A continuación se muestra el modelo del negocio del subproceso Ejecutar la Verificación Fiscal correspondiente a la Etapa de Ejecución. La descripción de este subproceso se muestra en el Anexo 2.

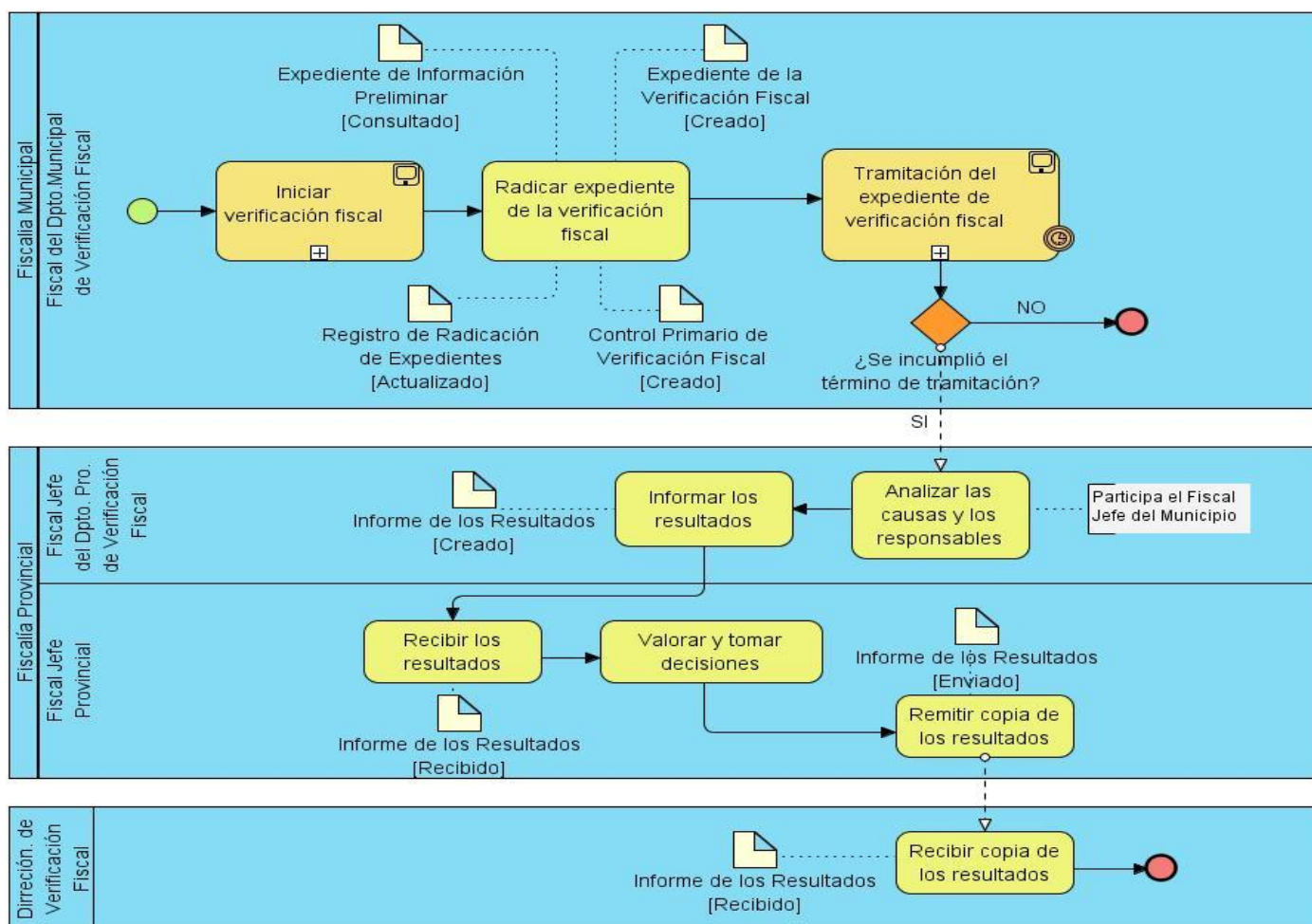


Figura 3: Diagrama del Negocio del Subproceso Ejecutar la Verificación Fiscal

A continuación se muestra la relación involucrados del negocio, artefactos y subprocesos del negocio.

Involucrados	Artefactos	Subproceso
Fiscal del Dpto. Municipal de VF.	Listado Objetivos Económicos. Expediente Información Preliminar.	Crear Expediente Información Preliminar.
Fiscal del Dpto. Municipal de VF.	Plan Trimestral VF Municipio.	Confeccionar Propuesta Plan Trimestral.
Fiscal del Dpto. Provincial de VF.	Plan Trimestral VF Provincia. Escrito devolviendo Plan VF del Municipio.	
Dirección de VF.	Plan Trimestral VF Nacional. Escrito devolviendo Plan VF Prov.	
Fiscal del Dpto. Municipal de VF.	Informe cumplimiento Plan Municipio.	Verificar Cumplimiento Plan.
Fiscal del Dpto. Provincia VF.	Informe cumplimiento Plan Provincia.	
Dirección de VF.	Informe cumplimiento Plan Nacional.	
Fiscal del Dpto. de VF.	Plan de Trabajo VF. Acta Reunión de Inicio VF. Acta Comparecencia.	Preparar la VF

	Acta de Requerimiento. Acta Ocupación.	
Fiscal del Dpto. Municipio VF. Fiscal del Dpto. Provincia VF.	Escrito solicitud de aprobación de VF extra plan o cambio plan.	Verificación Extra Plan.
Fiscal Jefe Municipal. Fiscal Jefe Provincial.	Escrito de denegación de la VF extra plan o cambio plan.	
Dirección de VF.	Escrito comunicando la denegación o aprobación de la VF extra plan o cambio.	
Fiscal del Dpto. Municipal de VF.	Notificación de Inicio de la VF a la entidad.	Iniciar la VF
Fiscal del Dpto. Municipal VF. Fiscal del Dpto. Provincial VF.	Expediente de la VF. Informe de los resultados	Ejecutar VF
Fiscal del Dpto. Municipal de VF.	Registrar hechos delictivos Municipal.	Presunto de hechos delictivos
Fiscal del Dpto. Provincial de VF.	Registro hechos delictivos Prov. Informe sobre hechos delictivos.	
Fiscal del Dpto. Municipal de VF.	Comunicación de indicios de enriquecimiento.	Detectar Hechos Proceso Confiscatorio.
Fiscal del Dpto. Municipal de VF.	Solicitud de prórroga.	Prórroga.


Fiscal Jefe Municipal.	Denegar solicitud de prórroga.	
Fiscal Jefe Provincial.	Denegar o autorizar solicitud de prórroga.	
Fiscal del Dpto. Municipal de VF.	Acta Reunión Previa VF. Informe resumen de los resultados de la VF. Acta Reunión con los Trabajadores.	Concluir VF
Fiscal del Dpto. Municipal VF.	Acta Reunión de Conclusiones. Acta Reunión de Re comprobación. Escrito solicitud prórroga.	Reunión de Conclusiones
Fiscal Jefe Municipal.	Escrito denegando solicitud prórroga.	
Fiscal Jefe Provincial	Escrito denegando o autorizando solicitud prórroga	
Fiscal del Dpto. Municipal de VF.	Resolución de la VF. Notificación de Resolución Informe resumen VF sectorializadas Registro Medidas Disciplinarias Escrito mostrando inconformidad medida disciplinaria aplicada y no aplicada	Resolución de la VF.

Fiscal del Dpto. Provincial de VF.	Informe resumen VF sectorializadas.	
Fiscal del Dpto. Municipal de VF.	Resolución por violaciones detectadas durante la Reinspección. Acta visita de la Reinspección	Reinspección
Fiscal Dpto. Provincia VF	Informe resumen Reinspección	

Tabla 1: Involucrados, Artefactos y Subproceso

2.4. Patrones para el modelado del negocio

Un patrón es la abstracción de una forma concreta el cual mantiene a repetirse en un contexto específico no arbitrario. Para modelar adecuadamente un proceso bajo los estándares, es necesario conocer el estándar BPMN y los patrones de workflows¹⁸ identificados por la investigación del profesor Van Der Aalst. El objetivo del desarrollo de los patrones fue describir la capacidad potencial que un workflows podría tener durante el rendimiento del proceso de negocio. El rango de patrones va desde los más simples a los más complejos y comprende los comportamientos esperados en la mayoría de los modelos de procesos. (bizagi, 2009). A continuación se mostrarán los patrones de workflows bajo el estándar de BPMN para modelar el proceso Verificación Fiscal:

Patrones Básicos de Control de Flujo	
 <pre> graph LR A[A] --> B[B] B --> C[C] </pre>	<p>Secuencia: Es el patrón básico de todo flujo de trabajo. Se requiere cuando hay una dependencia entre dos actividades, de tal forma que una actividad no pueda iniciarse antes de que otra haya terminado.</p>

¹⁸ Flujos de Trabajos (workflows).

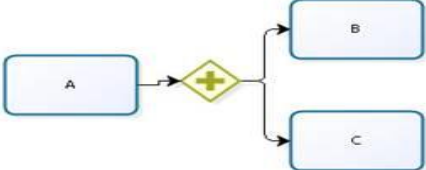
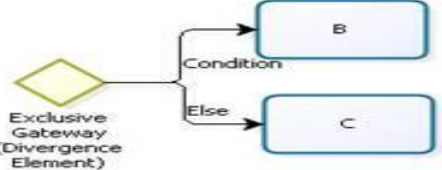
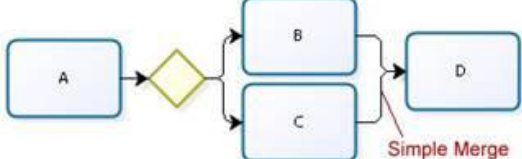
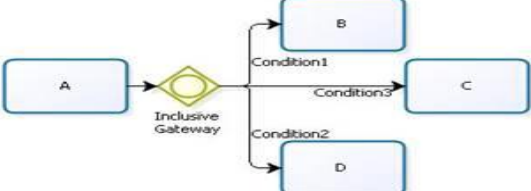
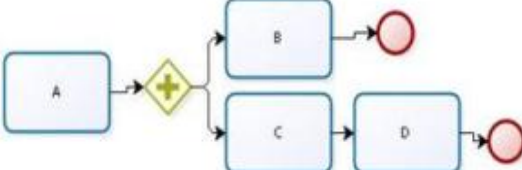
	<p>Distribución en Paralelo: La distribución en paralelo es necesaria cuando dos o más actividades deben ejecutarse de forma concurrente o en paralelo.</p>
	<p>Selección exclusiva: Ocurre cuando en un punto del flujo de trabajo se escoge sólo una de varias ramas del proceso, generalmente esta decisión se toma basándose en datos de control del flujo de proceso.</p>
<p>Sincronización y Enrutamiento Avanzada</p>	
	<p>Mezcla Múltiple o Serialización: Es un patrón del flujo de trabajo en donde dos o más ramas se unen sin sincronización.</p>
	<p>Selección Múltiple: El patrón selección múltiple asume que una de las alternativas es seleccionada y ejecutada. Sin embargo, algunas veces es importante usar una opción en la cual se permita escoger un subconjunto de alternativas de un conjunto de opciones posibles. Por ello, se encuentra el patrón de selección múltiple (inclusiva).</p>
<p>Estructurales</p>	
	<p>Terminación Implícita: Un subprocesso debería terminar cuando no hay nada más por hacer. En otras palabras, no hay actividades activas y ninguna actividad puede pasar a activa.</p>

Tabla 2: Patrones para el modelado del negocio

2.5. Validación del negocio

Al concluir toda la fase de Modelación del Negocio se realizó la revisión de cada uno de los Modelos de Procesos de Negocio por parte del cliente, en dicho caso la Fiscal responsable por la Dirección Verificación Fiscal, Esp. Teidy Colomar Zaldívar. Mediante esta revisión fueron detectados errores en algunos de estos modelos y posteriormente validados por el cliente dichos Modelos de Procesos de Negocio. Siendo entregada una carta de aceptación por parte de estos. (Ver Anexo 1)

Conjuntamente también se aplicó la siguiente técnica:

Revisiones Técnica Formales (RTF): A cada artefacto generado en esta fase se le fue aplicado esta técnica, un grupo de estudiantes de calidad fueron los seleccionados para que llevaran a cabo esta tarea, con el objetivo de que detectaran deficiencias, ambigüedades, omisiones, errores tanto de formato como de contenido. Una vez que estos fueron detectados son corregidos por el grupo de analistas teniendo finalmente la aceptación esperada, al ser eliminadas todas las no conformidades detectadas por el equipo de calidad.

Conclusiones Parciales

El análisis realizado en este capítulo permitió comprender el funcionamiento del proceso Verificación Fiscal, identificar los subprocesos asociados a cada etapa, asegurarse de que existiera un entendimiento común entre clientes y analistas. Se generaron artefactos que serán la entrada para la siguiente fase, los cuales fueron validados con el objetivo de que éstos cumplan un alto grado de calidad en su confección. Al finalizar el modelado del negocio el cliente quedó satisfecho entregando una carta de aceptación, concluyendo que el modelado de cada uno de los subproceso asociado a las Etapas de la Verificación Fiscal tuvo 100% de aceptación para el cliente.

CAPÍTULO 3: ESPECIFICACIÓN DE REQUISITOS

Introducción

En el presente capítulo se explicarán las técnicas utilizadas para el desarrollo de la fase de requisitos para el proceso Verificación Fiscal, técnicas que ayudaron a los analistas a la identificación de las funcionalidades y cualidades que el sistema debe cumplir. Se centra la atención en cada requisito asociado a cada Etapa de la Verificación Fiscal.

A partir de lo investigado en el capítulo anterior, se realiza la identificación de los requisitos funcionales correspondientes al proceso Verificación Fiscal que se lleva a cabo en la Fiscalía General de la República de Cuba, lo que ayudará a comprender cada requisito y siendo éstos descritos en el documento de especificación de requisitos, se presenta el diagrama de caso de uso del sistema para un entendimiento del mismo. Aplicando métricas para la validación en esta fase.

3.1. Requisitos

El diseño y la construcción de un elegante programa de computadora que resuelva el problema incorrecto no satisface las necesidades de nadie por lo tanto es muy importante entender que es lo que el cliente desea es por eso que una de las fases fundamentales es la de requisitos.

Estos son clasificados en dos tipos: (SOMMERVILLE, 2005)

Requisitos Funcionales: Característica requerida del sistema que expresa una capacidad de acción del mismo, una funcionalidad; generalmente expresada en una declaración en forma verbal.

Requisitos No Funcionales: Característica requerida del sistema, del proceso de desarrollo, del servicio prestado o de cualquier otro aspecto del desarrollo, que señala una restricción del mismo.

3.2. Aplicación de las técnicas para la captura de requisitos

Las entrevistas realizadas a la Fiscal responsable por la Dirección de Verificación Fiscal, Esp. Teidy Colomar Zaldívar, permitió al equipo de analistas detectar las dificultades existentes actualmente en el proceso Verificación Fiscal de la FGR. Básicamente se pudo obtener una amplia visión del trabajado que se lleva a cabo en dicho Órgano. Además de las tormentas de ideas realizadas por parte de un grupo de estudiantes y profesores, así como fiscales, con el objetivo de que estos últimos mostrarán sus ideas de forma libre, se pudo acumular una amplia gama de información e ideas. La aplicación de estas técnicas

permitió definir los requisitos funcionales del sistema, siendo estos un total de 124 requisitos distribuidos de la siguiente manera 53 en la Etapa Previa, 29 Etapa Ejecución y 41 en la Etapa Conclusión.

3.3. Requisitos Funcionales

A continuación se muestran los requerimientos funcionales del sistema, para acceder a la descripción íntegra de cada uno ver documento Especificación de requisitos funcionales del sistema.

Etapa Previa

RF_VF_VF_1: Mostrar informe de la situación del territorio.

RF_VF_VF_2: Modificar informe de la situación del territorio.

RF_VF_VF_3: Adicionar informe de la situación del territorio.

RF_VF_VF_4: Adicionar listado de los objetivos económicos-sociales del territorio.

RF_VF_VF_5: Modificar listado de los objetivos económicos-sociales del territorio.

RF_VF_VF_6: Mostrar listado de los objetivos económicos-sociales del territorio.

RF_VF_VF_7: Mostrar listado de sectores priorizados.

RF_VF_VF_8: Modificar listado de sectores priorizados.

RF_VF_VF_9: Adicionar listado de sectores priorizados.

RF_VF_VF_10: Crear ficha del objetivo económico-social del territorio.

RF_VF_VF_11: Mostrar ficha del objetivo económico-social del territorio.

RF_VF_VF_12: Crear ficha del objetivo económico-social del territorio

RF_VF_VF_13: Crear Expediente de Información Preliminar.

RF_VF_VF_14: Registrar plan trimestral de verificación fiscal del municipio.

RF_VF_VF_15: Mostrar plan trimestral de verificación fiscal del municipio.

RF_VF_VF_16: Modificar plan trimestral de verificación fiscal del municipio.

RF_VF_VF_17: Registrar plan trimestral de verificación fiscal de la provincia.

RF_VF_VF_18: Mostrar plan trimestral de verificación fiscal de la provincia.

RF_VF_VF_19: Modificar plan trimestral de verificación fiscal de la provincia.

RF_VF_VF_20: Registrar escrito de solicitud de aprobación de una verificación extra plan.

RF_VF_VF_21: Mostrar escrito de solicitud de aprobación de una verificación extra plan.

RF_VF_VF_22: Registrar escrito comunicando la denegación de una verificación extra plan.

RF_VF_VF_23: Mostrar escrito comunicando la denegación de una verificación extra plan.

RF_VF_VF_24: Registrar escrito comunicando la aprobación de una verificación extra plan.

RF_VF_VF_25: Mostrar escrito comunicando la aprobación de una verificación extra plan.

RF_VF_VF_26: Registrar escrito de solicitud de cambio de plan de una verificación fiscal.

RF_VF_VF_27: Mostrar escrito de solicitud de cambio de plan de una verificación fiscal.

RF_VF_VF_28: Registrar escrito comunicando la aprobación de un cambio de plan de verificación fiscal.

RF_VF_VF_29: Mostrar escrito comunicando la aprobación de un cambio de plan de verificación fiscal.

RF_VF_VF_30: Registrar escrito comunicando la denegación de un cambio de plan de verificación fiscal.

RF_VF_VF_31: Mostrar escrito comunicando la denegación de un cambio de plan de verificación fiscal.

RF_VF_VF_32: Registrar Informe del cumplimiento del plan del municipio.

RF_VF_VF_33: Mostrar Informe del cumplimiento del plan del municipio.

RF_VF_VF_34: Registrar Informe del cumplimiento del plan de la provincia.

RF_VF_VF_35: Mostrar Informe del cumplimiento del plan de la provincia.

RF_VF_VF_36: Registrar Informe del cumplimiento del plan del trimestre concluido.

RF_VF_VF_37: Mostrar Informe del cumplimiento del plan del trimestre concluido.

RF_VF_VF_38: Registrar la verificación fiscal.

RF_VF_VF_39: Registrar plan de trabajo de verificación fiscal.

RF_VF_VF_40: Mostrar plan de trabajo de verificación fiscal.

RF_VF_VF_41: Modificar plan de trabajo de verificación fiscal.
RF_VF_VF_42: Registrar los requerimientos o despachos.
RF_VF_VF_43: Mostar los requerimientos o despachos.
RF_VF_VF_44: Modificar los requerimientos o despachos.
RF_VF_VF_45: Registrar el Acta de reunión de inicio de la verificación fiscal.
RF_VF_VF_46: Mostrar el Acta de reunión de inicio de la verificación fiscal.
RF_VF_VF_47: Modificar el Acta de reunión de inicio de la verificación fiscal.
RF_VF_VF_48: Registrar el Acta de comparecencia.
RF_VF_VF_49: Mostrar el Acta de comparecencia.
RF_VF_VF_50: Modificar el Acta de comparecencia.
RF_VF_VF_51: Registrar el Acta de ocupación.
RF_VF_VF_52: Mostrar el Acta de ocupación.
RF_VF_VF_53: Modificar el Acta de ocupación.

Etapas Ejecución

RF_VF_VF_54: Mostrar Notificación Inicio de VF.
RF_VF_VF_55: Modificar Notificación Inicio de VF.
RF_VF_VF_56: Registrar Notificación Inicio de VF.
RF_VF_VF_57: Registrar Expediente de VF.
RF_VF_VF_58: Mostrar Expediente de VF.
RF_VF_VF_59: Registrar Control Primario.
RF_VF_VF_60: Mostrar Control Primario.
RF_VF_VF_61: Modificar Control Primario.
RF_VF_VF_62: Registrar Informe de los Resultados.
RF_VF_VF_63: Mostrar Informe de los Resultados.
RF_VF_VF_64: Registrar Informe de los especialistas.

RF_VF_VF_65: Mostrar Informe de los especialistas.

RF_VF_VF_66: Registrar Hechos Delictivos.

RF_VF_VF_67: Modificar Hechos Delictivos.

RF_VF_VF_68: Mostrar Hechos Delictivos.

RF_VF_VF_69: Registrar informe resumen sobre hechos delictivos derivados de VF.

RF_VF_VF_70: Mostrar informe resumen sobre hechos delictivos derivados de VF.

RF_VF_VF_71: Registrar Solicitud de prórroga a la tramitación de la VF.

RF_VF_VF_72: Mostrar Solicitud de prórroga a la tramitación de la VF.

RF_VF_VF_73: Registrar escrito denegando prórroga a la tramitación de la VF.

RF_VF_VF_74: Mostrar escrito denegando prórroga a la tramitación de la VF.

RF_VF_VF_75: Registrar escrito autorizando prórroga a la tramitación de la VF.

RF_VF_VF_76: Mostrar escrito denegando prórroga a la tramitación de la VF.

RF_VF_VF_77: Registrar el acta comunicación de indicios de enriquecimiento indebido.

RF_VF_VF_78: Mostrar el acta comunicación de indicios de enriquecimiento indebido.

RF_VF_VF_79: Registrar acta comunicación de violaciones graves.

RF_VF_VF_80: Mostrar acta comunicación de violaciones graves.

RF_VF_VF_81: Registrar notificación violaciones graves.

RF_VF_VF_82: Mostrar Notificación de violaciones graves.

Etapa de Conclusión

RF_VF_VF_83: Registrar acta de reunión previa de la VF.

RF_VF_VF_84: Mostrar acta de reunión previa de la VF.

RF_VF_VF_85: Registrar informe resumen de los resultados de la VF.

RF_VF_VF_86: Mostrar informe resumen de los resultados de la VF.

RF_VF_VF_87: Registrar acta de la reunión con los trabajadores.

RF_VF_VF_88: Modificar acta de la reunión con los trabajadores.

RF_VF_VF_89: Mostrar acta de la reunión con los trabajadores.

RF_VF_VF_90: Registrar resolución por violaciones de la legalidad detectada durante la re-inspección.

RF_VF_VF_91: Mostrar resolución por violaciones de la legalidad detectada durante la re-inspección.

RF_VF_VF_92: Registrar acta de visita de re-inspección.

RF_VF_VF_93: Mostrar acta de visita de re-inspección.

RF_VF_VF_94: Registrar informe resumen de las re-inspecciones en el trimestre.

RF_VF_VF_95: Mostrar informe resumen de las re-inspecciones en el trimestre.

RF_VF_VF_96: Registrar acta de visita de re-inspección.

RF_VF_VF_97: Mostrar acta de visita de re-inspección

RF_VF_VF_98: Registrar el informe resumen de las re-inspecciones en el trimestre.

RF_VF_VF_99: Mostrar el informe resumen de las re-inspecciones en el trimestre.

RF_VF_VF_100: Registrar acta de la reunión de conclusiones de la VF.

RF_VF_VF_101: Mostrar acta de la reunión de conclusiones de la VF.

RF_VF_VF_102: Registrar el escrito de solicitud de prórroga.

RF_VF_VF_103: Mostrar el escrito de solicitud de prórroga.

RF_VF_VF_104: Registrar el escrito denegando la solicitud de la prórroga.

RF_VF_VF_105: Mostrar el escrito denegando la solicitud de prórroga.

RF_VF_VF_106: Registrar el escrito autorizando la solicitud de prórroga.

RF_VF_VF_107: Mostrar el escrito autorizando la solicitud de prórroga.

RF_VF_VF_108: Registrar el acta de la reunión de re-comprobación.

RF_VF_VF_109: Mostrar el acta de la reunión de re-comprobación.

RF_VF_VF_110: Registrar la resolución de la VF

RF_VF_VF_111: Mostrar la resolución de la VF

RF_VF_VF_112: Registrar la notificación de la resolución de la VF

RF_VF_VF_113: Mostrar la notificación de la resolución de la VF

RF_VF_VF_114: Registrar el informe resumen sobre la VF sectorializadas municipal.

RF_VF_VF_115: Mostrar el Informe resumen sobre la verificación fiscal sectorializadas municipal

RF_VF_VF_116: Registrar el Informe resumen sobre la verificación fiscal sectorializadas provincial.

RF_VF_VF_117: Mostrar el Informe resumen sobre la verificación fiscal sectorializadas provincial

RF_VF_VF_118: Registrar el registro de medidas disciplinarias.

RF_VF_VF_119: Mostrar el registro de medidas disciplinarias.

RF_VF_VF_120: Modificar el registro de medidas disciplinarias.

RF_VF_VF_121: Registrar escrito mostrando inconformidad con la medida disciplinaria aplicada.

RF_VF_VF_122: Mostrar escrito mostrando inconformidad con la medida disciplinaria aplicada.

RF_VF_VF_123: Registrar reclamación con la medida disciplinaria no aplicada.

RF_VF_VF_124: Mostrar reclamación con la medida disciplinaria no aplicada.

3.4. Validación de los Requisitos

Una vez definidos los requisitos se hace necesario su validación con el fin de comprobar su calidad a la hora de ser especificados y verificar si no existen ambigüedades en los mismos, para esto se han de tener en cuenta y llevar a cabo un conjunto de técnicas y métricas (Monzón, 2005), a continuación se reflejan las utilizadas durante esta fase para el proceso Verificación Fiscal:

Durante el desarrollo de la fase de requisitos se realizaron revisiones de los mismos con la Fiscal responsable por la Dirección de Verificación Fiscal, Esp. Teidy Colomar Zaldívar, estas revisiones

permitieron detectar errores a tiempo, los cuales fueron corrigiéndose por el equipo de analistas y finalmente validados por parte de la fiscal.

Revisiones Técnicas Formales: Una vez especificados los requisitos estos fueron revisados por un grupo de calidad, estas revisiones permitieron que se detectarían errores en el artefacto de especificación de requisitos en cuanto ambigüedades, deficiencia y omisiones, siendo corregidos por el equipo de analistas que llevaba a cabo esta tarea.

Conjuntamente se aplicaron las siguientes métricas:

Métrica para el cálculo de estabilidad de los requisitos con el fin de determinar si estos fueron variables o estables desde el inicio, y su cálculo es de la siguiente manera:

$$\text{ETR} = [(\text{RT} - \text{RM}) / \text{RT}] * 100$$

Donde:

ETR: Valor de la estabilidad de los requisitos.

RT: Total de requisitos definidos.

RM: Número de requisitos modificados, que se obtienen como la sumatoria de los requisitos insertados, modificados y eliminados.

Fueron identificados un total de 124 requisitos funcionales y de ellos 8 no se comportaron estables.

Aplicando la fórmula quedaría de la siguiente manera:

$$\text{ETR} = [(124 - 8) / 124] * 100 = 93.5$$

Arrojando un valor de ETR = 93.5, ya que para un total de 124 requisitos funcionales, 8 de ellos se mantuvieron cambiantes, demostrando que éstos no se comportaron suficientemente estables durante el proceso, se realizó una segunda iteración con el objetivo de eliminar esa inestabilidad en los requisitos y finalmente fueron eliminados.

Otra métrica aplicada fue el cálculo de la especificidad de los requisitos para evaluar la misma. En la cual se ha de tener en cuenta lo siguiente:

- Número total de requerimientos: Nt.

- Número de RF: Nf.
- Número de RNF: Nnf.
- Calidad de la especificación: Q1.
 - Número de requerimientos interpretados de la misma manera: Nui.
 - Nt

$$Q1 = Nui / Nt.$$

El valor óptimo de Q1 es 1, lo que significa que mientras más cercano sea el valor a éste, mayor calidad tendrá la especificación de los requerimientos.

Aplicando la métrica al documento de especificación de requisitos dado como propuesta de solución.

1- Número Total de requerimiento:

$$Nt = Nf + Nnf$$

$$Nt = 124 + 50$$

$$Nt = 174$$

2- Revisores de la documentación:

REVISOR	ROL
Ing. Annaliet Perez Parra	Analista Principal
Ing. Yenier Figueroa Machado	Jefe del Proyecto
Ing. Anamarys García Rodríguez	Analista

Tabla 3: Grupo de revisores

3- Resultado de la Interpretación:

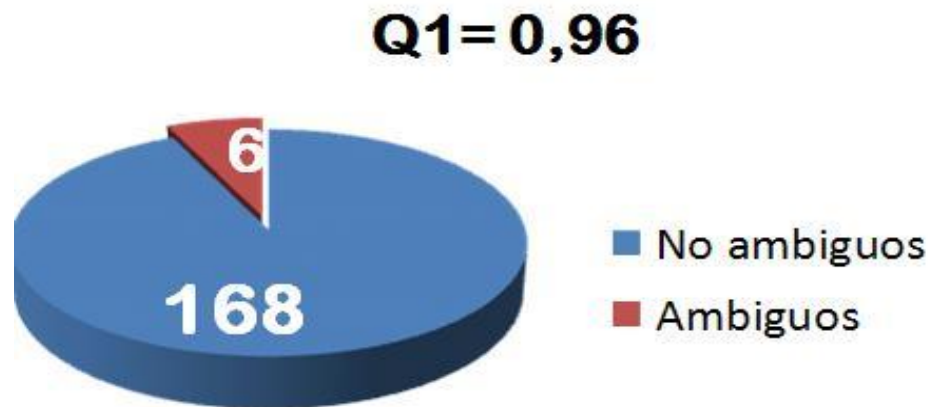


Figura 4: Grado de especificidad de los requisitos

4- Resultado de la aplicación de la métrica:

$$Nui / Nt = Q1 \rightarrow 168 / 174 = 0.96.$$

5- Interpretación del resultado una vez aplicada la métrica:

El valor de Q1 resultó ser de 0.96 demostrando que de un total de 174 requisitos, 168 resultaron interpretaciones idénticas y 6 mostraron diferencias en cuanto al entendimiento e interpretación por parte del equipo de revisores. Concluyendo que la revisión por parte de los revisores tiene una adecuada consistencia y que la especificación de éstos es poco ambigua, por lo que se procedió a un refinamiento de éstos para la eliminación de la ambigüedad en una segunda iteración.

Los prototipos también fue una de las formas de validación utilizadas que contribuyeron a garantizar la calidad del desarrollo de la fase:

Prototipos: Los prototipos fueron una de las técnicas más importantes en el desarrollo del presente trabajo, pues demostraron al cliente una visión del sistema, permitiéndole a los clientes probarlo en situaciones reales y explorar su uso, al revelar errores como atributos, que resultaron omitidos por el cliente una vez capturado los requisitos y que al ver las funcionalidades, descubrieron que habían sido omitidos, así como ambigüedades por parte de los analistas. Siendo éstos corregidos y permitiendo una aprobación por parte del cliente. En el Anexo 3 se muestra el ejemplo de un prototipo y su descripción.

3.5. Diagrama de Caso de Uso del Sistema

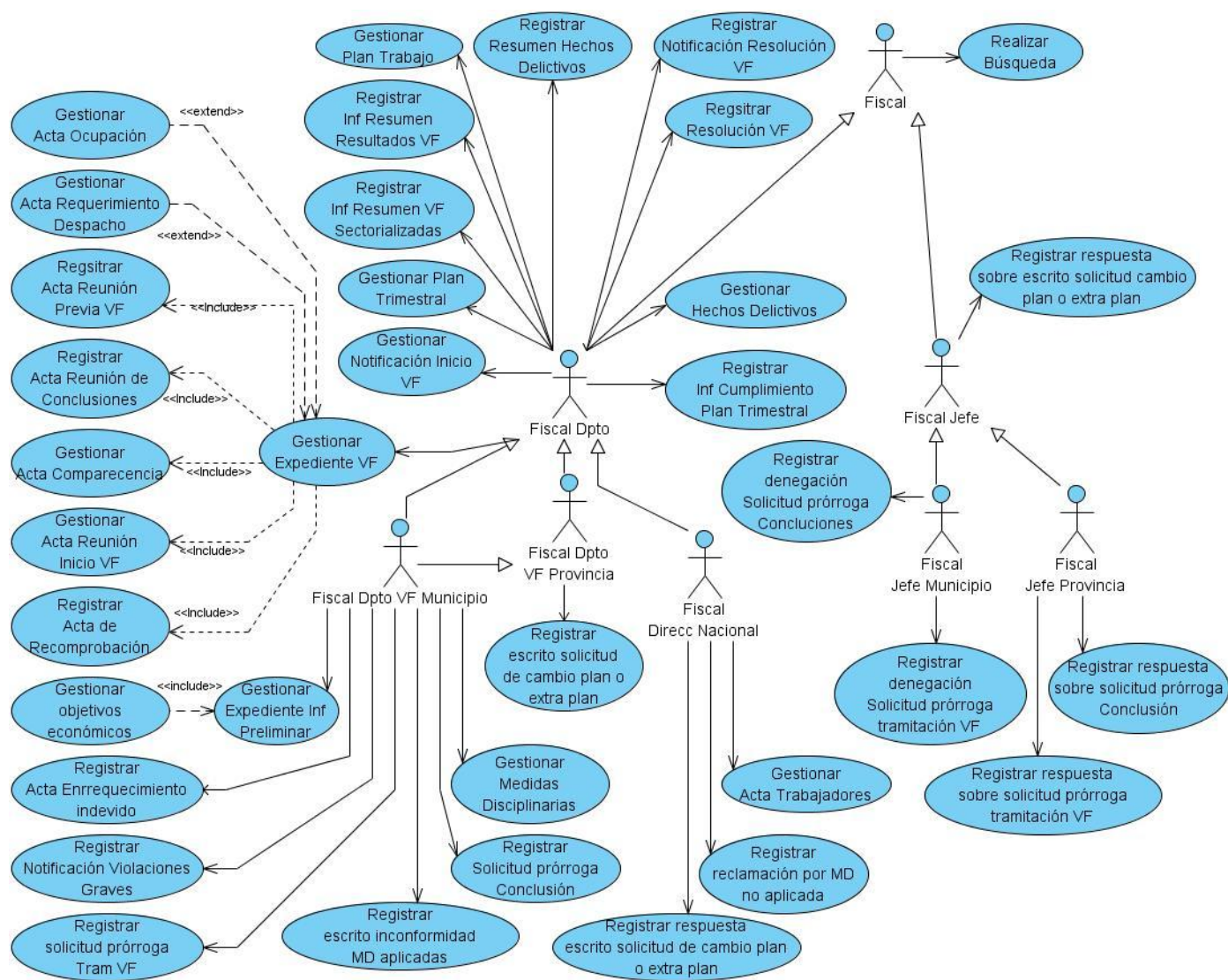


Figura 5: Diagrama de Caso de Uso del Sistema

Patrones utilizados en el presente diagrama:

En el diagrama se han utilizados algunos patrones con el fin de que el trabajo se realice de una manera más precisa y que actividades que en ocasiones resultan engorrosas se hagan de la manera correcta. La aplicación de los patrones de caso de uso traen consigo los siguientes beneficios: (ECURED, 2008)

- Aumentar la productividad.
- Reutilizar elementos existentes (en este caso fragmentos de modelos)
- Evitar el re trabajo por errores.
- No invertir tiempo en resolver problemas ya resueltos.
- Aplicar la teoría al trabajo práctico.
- Habilitar las herramientas de soporte para modelar el desarrollo.

Múltiples actores en su variante rol común: En el diagrama se destacan actividades que son comunes para varios actores, por lo que se define un actor del cual van a heredar los actores con las funcionalidades comunes.

Extensión Concreta: Es un patrón de estructura que consiste en dos casos de uso y una relación de extensión. En el caso del presente diagrama de caso de uso de sistema, existe un caso de uso llamado “Gestionar Expediente VF” y como extendido tiene un caso de uso, por ejemplo “Gestionar Acta Ocupación”, lo cual significa que este último es concreto, es decir puede ser instanciado por su cuenta como por el caso de uso base.

Inclusión Concreta: Es un patrón de estructura. Consiste en dos casos de uso y una relación de inclusión entre el caso de uso base y el caso de uso incluido. En el caso del presente diagrama de caso de uso, existe un caso de uso llamado “Gestionar Expediente VF” y como incluido tiene un caso de uso por ejemplo “Gestionar Acta Reunión de Inicio VF”, este último puede ser instanciado en el mismo, mientras que para que el caso de uso base cumpla toda su funcionalidad tiene que haberse iniciado el caso de uso incluido.

CRUD (Crear, Leer, Actualizar, Eliminar): Es un patrón de estructura que se base en la función de casos de usos simples para formar una unidad conceptual. En este caso se utiliza el CRUD en su variante CRUD Parcial pues no se cumplen para un caso de uso todas las acciones que incluye el CRUD.

3.6. Requisitos no Funcionales

A continuación se muestran los identificados para el proceso Verificación Fiscal.

Usabilidad

- El software tendrá siempre la posibilidad de ayuda disponible para cualquier tipo de usuario, lo que le permitirá un avance considerable en la explotación de la aplicación en todas sus funcionalidades.
- Existirán servidores locales con capacidad necesaria para el procesamiento de las solicitudes del conjunto de aplicaciones de las diferentes oficinas.
- Las aplicaciones siempre solicitarán los datos a través del servidor local.
- Desde cada servidor local se establecerá la conexión con servidores centrales para mantener la actualización de los datos en ambos sentidos.
- El tiempo de entrenamiento requerido para que usuarios normales y avanzados sean productivos operando el sistema es de 15 días.
- Debe poseer una interfaz agradable para el cliente.

Fiabilidad

- El sistema estará disponible 24 horas al día, 7 días a la semana.
- La disponibilidad de los casos asignados desde cualquier parte del país.
- El tiempo medio entre fallos es de 1 mes.
- El tiempo permitido para que el sistema quede fuera de operación luego de haber fallado es de 2 días.
- La precisión y exactitud requerida en las salidas del sistema o sea el máximo de errores, es de 5 errores/MLC.
- La herramienta de implementación a utilizar tiene soporte para recuperación ante fallos y errores.

Eficiencia

- Tiempo de respuesta promedio de las peticiones que se realizan al servidor no deberá ser mayor de 3 segundos.
- El número de clientes o transacciones que el sistema puede alojar es de 2000.

Soporte

- Soporte para grandes volúmenes de datos y velocidad de procesamiento.
- Tiempo de respuesta rápido en accesos concurrentes.
- El sistema debe ser multiplataforma.

Restricciones de diseño

- El lenguaje de programación es PHP 5.2.0.
- El framework¹⁹ de desarrollo es Symfony²⁰ 1.3.6.
- La herramienta IDE de desarrollo utilizada será Eclipse 3.5+PDT.
- La herramienta case utilizada es el Visual Paradigm 6.0.
- La herramienta gestor de base de datos es el PostgreSQL 8.4.

Interfaz

Interfaces de usuario

- Diseño sencillo, con pocas entradas, permitiendo que no sea necesario mucho entrenamiento para que los usuarios puedan utilizar el sistema.
- El sistema tiene que ofrecer una interfaz amigable, fácil de operar.
- El sistema tiene que mantener la línea de diseño establecida para la institución que mantiene la uniformidad y representatividad de la misma.

Requerimientos de Seguridad

- Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.
- El sistema debe mantener en todo momento la seguridad de la información asegurando la autenticidad de la misma.

¹⁹ La palabra inglesa framework define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular.

²⁰Framework de desarrollo para aplicaciones web.

- La seguridad se establecerá por roles que se le asignarán a los usuarios que interactúen con el sistema.
- El software brindará solamente aquellas funcionalidades que competen a la Unidad Ejecutora donde esté implantado.
- El sistema mantendrá en todo momento las trazas que se corresponden con las diferentes situaciones críticas que se puedan ocurrir.

Hardware

Cliente

- Memoria RAM 256 MB o superior.
- Disco duro de 20 GB o superior.
- Procesador Intel a 1 GHz de velocidad de procesamiento o superior.
- Tarjeta de red Ethernet.
- Debe tener acceso a una impresora.
- Debe tener acceso a un scanner.

Servidores

- Se debe contar con un servidor que brinde las siguientes funciones integradas: DNS, DHCP, Firewall.
- Se debe contar con un servidor de correo independiente.
- Se debe contar con un servidor independiente LDAP.
- Se debe contar con un servidor PostgreSQL y FTP.
- Se debe contar con un servidor independiente de Salva.
- Quad-Core Intel Xeon E5430 (2.66 GHz, 80 Watts, 1333 FSB) o superior.
- Memoria RAM de 32 GB o superior.

Software

Cliente

- Navegador Firefox 3 o superior.

Servidor

- Sistema Operativo Debian GNU/Linux 5.
- PostgreSQL 8.4.
- PHP 5.3 con las librerías y configuraciones necesarias desplegar Symfony 1.3.8.

Requisitos Legales, de Derecho de Autor y otros.

El Sistema de Informatización de la Gestión Fiscal debe garantizar el cumplimiento de lo dispuesto en las normas jurídicas, fundamentalmente las que se emitieron con rango de Ley, permitiendo adecuar el sistema de forma fácil a cambios en dichas normas. Debe igualmente hacer referencia al Registro y depósito legal de la aplicación.

Estándares Aplicables

El estándar utilizado en el proyecto para la especificación de requisitos es el IEEE 830. Los estándares de codificación utilizados en el proyecto están basados en normas seguidas por el código Symfony, específicamente el estándar "UpperCamelCase" para nombre de clases y variables.

Conclusiones Parciales

El análisis realizado en este capítulo permitió identificar los requisitos asociados a cada etapa de la Verificación Fiscal debido a las técnicas utilizadas para la captura de los mismos. El diagrama de casos de uso del sistema permitió un mejor entendimiento entre actores del sistema y casos de usos, así como los patrones aplicados para su construcción proporcionó la correcta organización. Las métricas aplicadas lograron medir la calidad con la que fueron identificados los requisitos durante esta fase.

CAPÍTULO 4: ANÁLISIS Y DISEÑO

Introducción

En este capítulo se realiza el diseño de software del módulo Verificación Fiscal, se dan a conocer los artefactos generados, la arquitectura definida y se explican los patrones utilizados, además se realiza una evaluación de los resultados del trabajo realizado, aplicando métricas para determinar la calidad.

4.1. Modelo de Diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso, centrándose en como los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. El modelo de diseño sirve de abstracción a la implementación y se utiliza como una entrada fundamental de las actividades de implementación (ROJAS, 2002)

4.2. Principios del diseño

El diseño es uno de los pilares fundamentales en la Ingeniería del Software, es una de las actividades técnicas necesarias para la elaboración del software. Entre las tareas fundamentales del diseño están producir el diseño de datos, diseño arquitectónico, diseño de interfaz y diseño de componentes.

Cuando se diseñan sistemas informáticos es indispensable hacerlo de forma correcta. El diseño propuesto tiene que cumplir a cabalidad con los requerimientos del sistema. Es la única forma de convertir exactamente los requisitos de un cliente en un producto o sistema de software finalizado; debe ser capaz de facilitar las mejoras del software, tiene que ser entendible por otros profesionales de la especialidad, servir como guía para los demás pasos de la ingeniería de software, permitiendo la comprobación del sistema fácilmente.

Pressman plantea una serie de principios básicos que se tienen que tener en cuenta en el momento de diseñar, ellos garantizan el correcto funcionamiento del sistema. Dichos principios son: (Pressman, 2005)

- En el proceso deben tomarse enfoques alternativos.
- El diseño deberá poderse rastrear hasta el modelo de análisis.
- El diseño no deberá inventar nada que ya esté inventado.

- El diseño deberá minimizar la distancia intelectual entre el software y el problema como si de la misma vida real se tratara.
- El diseño deberá presentar uniformidad e integración.
- El diseño deberá estructurarse para admitir cambios.
- El diseño deberá estructurarse para degradarse poco a poco.
- El diseño no es escribir código y escribir código no es diseñar.
- El diseño deberá evaluarse en función de la calidad mientras se va creando, no después de terminado.
- El diseño deberá revisarse para minimizar los errores conceptuales.

Cuando estos principios son aplicados correctamente, entonces el diseño creado muestra los factores de calidad internos y externos. Los factores externos son aquellas propiedades del software que se observan fácilmente: velocidad, fiabilidad, grado de corrección, usabilidad. Los internos son importantes para los ingenieros y personal encargo de realizar el proyecto, conducen a un diseño de alta calidad. Para lograrlos es necesario comprender correctamente los conceptos básicos del diseño.

4.3. Modelo Conceptual

Un modelo conceptual se realiza con el objetivo de captar las abstracciones esenciales y la información indispensable para comprender el dominio dentro del contexto de los requerimientos actuales. Explica los conceptos más significativos en un dominio del problema, es la herramienta más importante del análisis orientado a objetos, se representa en UML mediante un grupo de diagramas de estructura estática (Diagrama de clases conceptuales), donde no se define ninguna operación, en estos diagramas se muestran conceptos (objetos), asociaciones entre conceptos (relaciones) y atributos de conceptos. (Informática, 2000). A continuación se muestra el Modelo Conceptual correspondiente al proceso Verificación Fiscal.

4.4. Arquitectura definida

La arquitectura representa la clave para comprender, organizar y comunicar un sistema, además permite implantar conceptos para volver a usarlo y facilitar la evolución de la solución. En el proyecto SIGEF fase II se define la utilización de la implementación del Modelo Vista Controlador que realiza Symfony pues éste es el framework de desarrollo que utiliza SIGEF en su fase I. Este patrón MVC se ve frecuentemente en aplicaciones web como es el caso de la aplicación que se va a desarrollar, separando los datos de la aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos: (Fabien Potencier, 2008).

La capa del Modelo

- Abstracción de la base de datos.
- Acceso a los datos.

La capa de la Vista

- Vista.
- Plantilla.
- Layout.

La capa del Controlador

- Controlador Frontal.
- Acción.

A continuación se muestra una figura con el flujo de trabajo de Symfony y la representación del patrón arquitectura Modelo Vista Controlador:

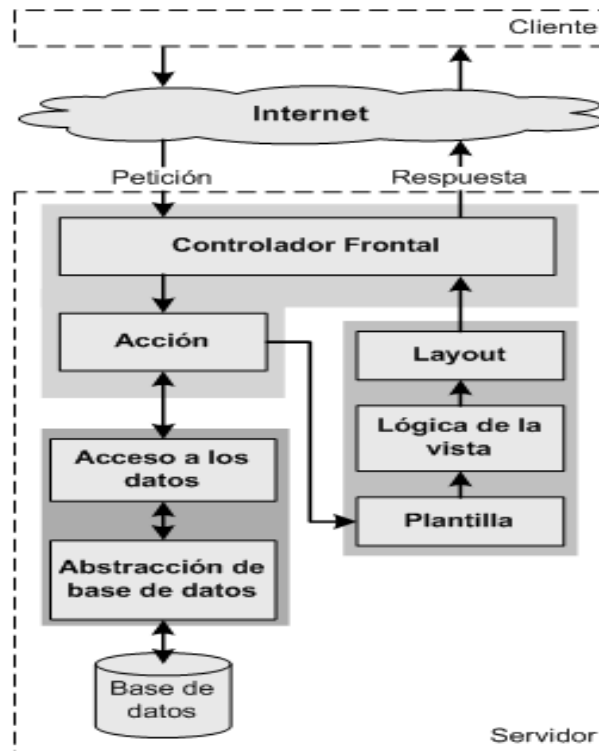


Figura 7: El flujo de trabajo de Symfony





El patrón MVC obliga a dividir y organizar el código de acuerdo a las convenciones establecidas por el framework. El código de la presentación se guarda en la Vista, el código de manipulación de los datos en el Modelo y la lógica de procesamiento de las peticiones constituyen el Controlador. (Fabien Potencier, 2008)

Quedando según el flujo de trabajo de Symfony distribuido el MVC de la siguiente manera:

- ✓ El Modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio. Es aquí donde se encuentra la Abstracción de la base de datos y el Acceso a los datos.
- ✓ La Vista transforma el modelo en una página web que permite al usuario interactuar con ella. Es aquí donde se encuentra la parte de la Plantilla de la Vista, la parte de la lógica de la Vista y la parte del Layout de la Vista.
- ✓ El Controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista. Este nivel tiene contenido el Controlador frontal y la Acción.

4.5. Diagramas de clases del diseño

Teniendo en cuenta la decisión del proyecto SIGEF II de realizar una aplicación web, se orienta utilizar para la realización de los diagramas de clases del diseño, los estereotipos web. A continuación se muestra una tabla con los utilizados:

Estereotipos web para las clases	
Estereotipo	Descripción
 Página Servidora	Representa una página Web que tiene scripts ejecutados por el servidor. Estos scripts interactúan con los recursos que se encuentran al alcance del servidor. Sólo puede mantener relaciones con objetos que se encuentren en el servidor.
 Página Cliente	Representan páginas que son dibujadas por el navegador web y pueden ser una combinación de algún o algunos lenguajes de marcado, scripts del lado del cliente, islas de datos.
 Formulario	Representa una colección de campos de entrada que forman parte con una página del lado cliente (Client Page). Tiene una correspondencia directa con la etiqueta <FORM> de XHTML.
 Java Script	Es una colección de scripts del lado del cliente que existe como un archivo separado y que son incluidos mediante una petición independiente por parte del navegador.
Relaciones entre las Clases	
<<include>>	Refleja el comportamiento de la clase incluida colocada dentro del comportamiento de la clase base.

<<submit>>	Esta relación siempre se da entre una “form” y una “server page”, por supuesto, la “server page” procesa los datos que la “form” le envía (submits)
<<build>>	Sirve para identificar cuales “server page” son responsables de la creación de una “client page”. Una “server page” puede crear varias “client page”, pero una “client page” sólo puede ser creada por una sola “server page”. Esta relación siempre es unidireccional.

Tabla 4: Estereotipos y Relaciones

Con la utilización de los diferentes estereotipos mostrados anteriormente fueron realizados los diagramas de clases del diseño, a continuación se muestra un ejemplo de ello:

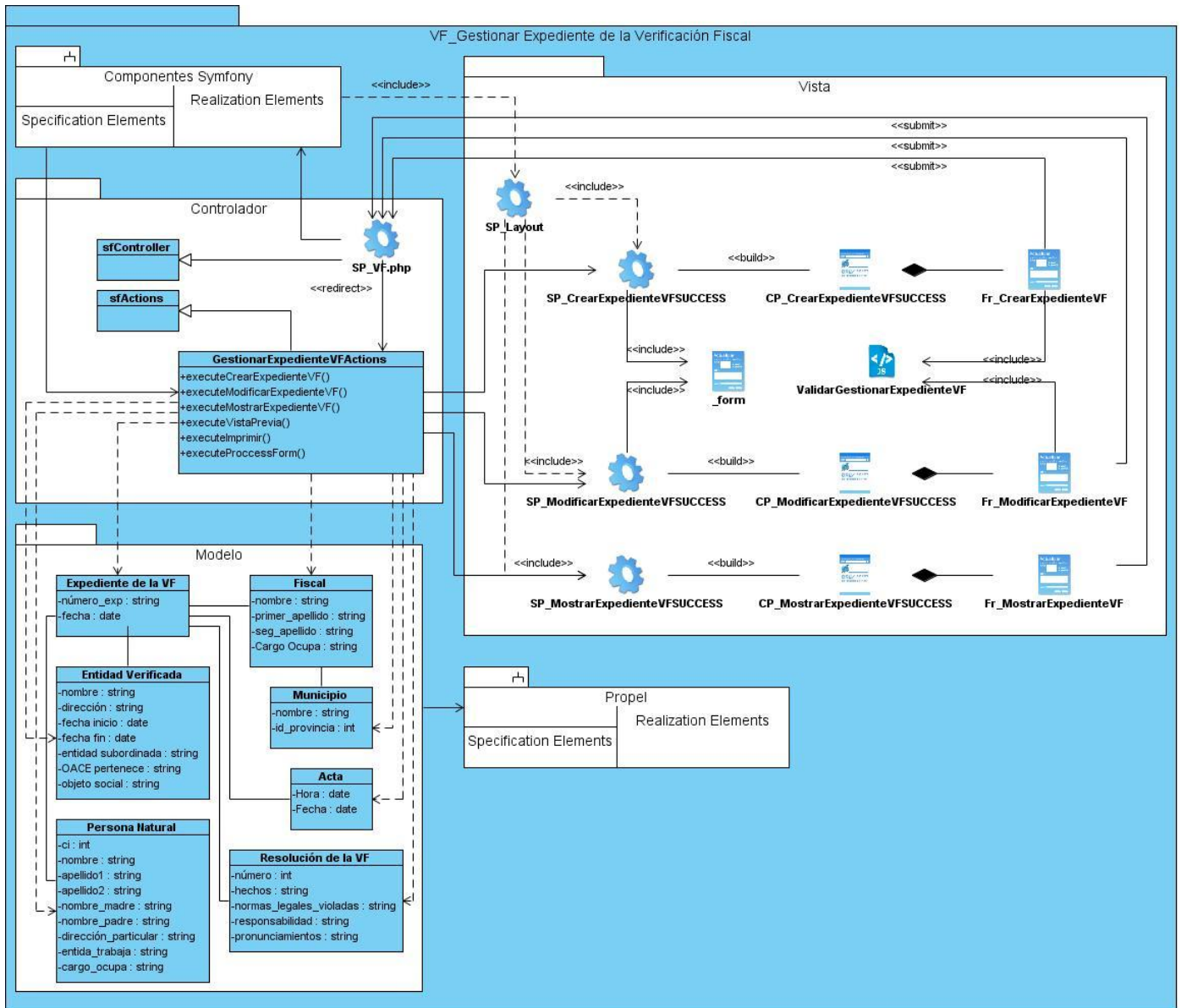


Figura 8: Diagrama de clase del diseño Gestionar Expediente de la Verificación Fiscal.

Nota: Dirigirse al Anexo 4 para ver los demás diagramas de clases del diseño.

4.6. Patrones de diseño

Con el objetivo de ayudar a identificar los errores y obstáculos comunes que ocurren al crear el sistema, así como establecer un vocabulario de diseño común entre los desarrolladores, se define la utilización de

los patrones GoF y los GRASP (Rojas, 2005). Como se ha utilizado el framework de desarrollo Symfony para facilitar el trabajo y la realización del software SIGEF II, éste promueve las buenas prácticas de diseño y programación ya que maneja patrones de diseño. A continuación se muestra como se reflejan los patrones de diseño dado determinadas circunstancias a la hora de tener un problema de diseño:

Patrones GRASP son patrones generales de software para asignación de responsabilidades:

Patrones GRASP: Son patrones generales de software para la asignación de responsabilidades

Experto: Este patrón se refleja con la inclusión de la librería Propel para el mapeo de la base de datos. Es uno de los patrones que más se utiliza cuando se trabaja con Symfony, utiliza esta librería para realizar su capa de abstracción en el modelo.

Creador: Se evidencia en la clase Actions donde se encuentran las acciones definidas para el sistema. En dichas acciones se crean los objetos de las clases que representan las entidades, lo que demuestra que la clase Actions es creador de dichas clases entidades.

Alta Cohesión: Esto se evidencia al utilizar el framework Symfony ya que éste permite la organización del trabajo en cuanto a estructura del proyecto y la asignación de responsabilidades con una alta cohesión. Ejemplo de ello es la clase Actions la cual está formada por varias funcionalidades que están estrechamente relacionadas, siendo la misma la responsable de definir las acciones para las plantillas y colaborar con otras para la realización de operaciones, instanciar objetos y acceder a las properties.

Bajo Acoplamiento: Este patrón se refleja también en la clase Actions que hereda únicamente de sfActions para alcanzar un bajo acoplamiento de clases. Las clases que implementa la lógica del negocio y de acceso a los datos se encuentran en el modelo, las cuales no tienen asociaciones con la de la vista o el controlador, proporcionando de esta manera que la dependencia en este caso sea baja.

Controlador: Todas las peticiones Web son manipuladas por un solo controlador frontal, que es el punto de entrada de toda la aplicación. Este patrón se evidencia por ejemplo en la clase sfController, que es la clase del controlador que se encarga de decodificar la petición y transferirla a la acción correspondiente.

Tabla 5: Utilización de los patrones GRASP

Patrones GoF

Patrón Creacional Instancia Única: Este patrón se aplica en la clase `sfRouting`, ya que es muy utilizada porque es la encargada de enrutar todas las peticiones que se hagan al registro. El Singleton `sfRouting` también define otros métodos que pueden llegar a ser muy útiles para el manejo manual de las rutas como son el: `hasRouter ()` y el `GetRoutesByName ()`.

Patrón de Comportamiento Command: Este patrón se ve de manifiesto en el método `dispatch ()` de la clase `sfWebFrontController`, que es la encargada de determinar cuál módulo y acción usar en dependencia de la petición del usuario.

Patrón Estructural Decorador: Este método pertenece a la clase abstracta `sfView`, padre de todas las vistas, que contienen un decorador para permitir agregar funcionalidades dinámicamente. Este patrón se observa en el archivo denominado `layout.php` que contiene el layout (plantilla global) de todas las páginas. El mismo almacena el código HTML que es común en todas las páginas, para no tener que repetirlo en cada página, por lo que el layout decora la plantilla.



Tabla 6: Utilización de los patrones GoF

4.7. Métricas para la validación del diseño

La IEEE²¹ define como métrica a la medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado (Sanz, 1990). Varios investigadores han intentado desarrollar una métrica específica que proporcione la medida completa de la complejidad del software. A pesar de que existen numerosas propuestas en cuanto a métricas y formas de medir la calidad de un software, es cada vez

²¹ Instituto de Ingenieros Eléctricos y Electrónicos (Institute of Electrical and Electronics Engineers)

más frecuente la utilización de éstas, debido a que las mediciones son la esencia para obtener un producto con un alto grado de calidez. Entre los investigadores que han definido métricas de este tipo se encuentran las propuestas por Chidamber y Kemerer (CK) y las propuestas por Lorenz y Kidd. Las mismas son definidas como métricas orientadas a objetos, las mismas son: (Olmedilla Arregui, 2005)

Métricas propuestas por Lorenz y Kidd

Estos autores las proponen basadas en las clases. Las separan en cuatro categorías: tamaño, herencia, valores internos y valores externos. También proponen otras orientadas al tamaño y la complejidad de las clases.

Las métricas orientadas al tamaño se centran en contar los atributos y operaciones de cada clase y los valores para sistema como un todo son:

- Número de Métodos de Instancia Públicos (PIM).
- Número de Métodos de Instancia (NIM).
- Número de Variables de Instancia (NIV).
- Tamaño de clase (TC).
- Tamaño medio de operación (TMO).
- Número de escenarios (NE).
- Número de clases claves (NCC).
- Número de subsistemas (NSUB).

Las que se basan en la herencia para ver en la forma que las operaciones se reutilizan en la jerarquía de clases son:

- Número de operaciones redefinidas para una subclase (NOR).
- Número de operaciones añadidas por una subclase (NOA).
- Índice de especialización (IES).
- Número de Métodos Heredados (NMI).

Métricas propuestas por Chidamber y Kemerer:

Es uno de los conjuntos de métricas más difundidos y conocidas como las CK, también se les llama MOOSE. Adoptan tres criterios a la hora de definir las: capacidad de satisfacer propiedades analíticas, aspecto intuitivo a los profesionales y facilidad para su recogida automática. Definen los siguientes conceptos:

- Métodos ponderados por clase (MPC).
- Profundidad del árbol de herencia (PAH).
- Número de hijos (NDH).
- Acoplamiento entre clases objeto (AEC).
- Respuesta para una clase (RPC).
- Carencia de cohesión en los métodos (CCM).

Para la validación del diseño realizado se aplicó la métrica Tamaño de clase y se obtuvo la siguiente tabla como resultado de su aplicación:

NO.	CLASES	CANTIDAD DE ATRIBUTOS	CANTIDAD DE OPERACIONES	UMBRALES	TAMAÑO
1.	Gestionar Acta Inicio VF	0	4	4	Pequeño
2.	Gestionar Acta Previa VF	0	4	4	Pequeño
3.	Gestionar Acta Recomprobación	0	4	4	Pequeño
4.	Gestionar Acta Comparecencia	0	4	4	Pequeño
5.	Gestionar Acta Ocupación	0	4	4	Pequeño
6.	Gestionar Acta Requerimiento	0	4	4	Pequeño

7.	Gestionar Acta trabajadores	0	4	4	Pequeño
8.	Gestionar Acta Entrega	0	4	4	Pequeño
9.	Gestionar Acta Conclusión	0	4	4	Pequeño
10.	Gestionar Expediente Información Preliminar	0	5	5	Pequeño
11.	Gestionar Plan trimestral municipio	0	4	4	Pequeño
12.	Gestionar Plan trimestral Provincia	0	4	4	Pequeño
13.	Gestionar Plan trimestral Nacional	0	4	4	Pequeño
14.	Gestionar Expediente VF	0	4	4	Pequeño
15.	Crear Solicitud Cambio Plan	0	4	4	Pequeño
16.	Disponer sobre Solicitud Cambio Plan	0	4	4	Pequeño
17.	Crear Resolución	0	3	3	Pequeño
18.	Acta	2	5	7	Pequeño
19.	Entidad Verificada	7	15	22	Medio
20.	Fiscal	4	9	13	Pequeño

21.	Acta Requerimiento	6	13	19	Pequeño
22.	Acta Reunión Inicio	3	7	10	Pequeño
23.	Acta Comparecencia	5	11	16	Pequeño
24.	Acta Ocupación	5	11	16	Pequeño
25.	Acta trabajadores	6	13	19	Pequeño
26.	Acta Conclusión	7	15	22	Medio
27.	Acta Recomprobación	6	13	19	Pequeño
28.	Acta Reunión Previa	4	9	13	Pequeño
29.	Fiscalía	1	3	4	Pequeño
30.	Resolución VF	5	11	16	Pequeño
31.	Plan VF Trimestral	6	13	19	Pequeño
32.	Cambio Plan	0	3	3	Pequeño
33.	Expediente	4	9	13	Pequeño
34.	Listado Objetivos	4	9	13	Pequeño
35.	Municipio	2	5	7	Pequeño
36.	Provincia	2	5	7	Pequeño
37.	Expediente VF	1	3	4	Pequeño

Tabla 7: Resultado de la métrica tamaño de clase.

Resultado después de ser aplicada la métrica

Se presentó un **total de 37 clases** para un **promedio de atributos de 2.16** y un **promedio de operaciones de 8,91**.

De esta forma el umbral queda con los datos mostrados a continuación:

Umbral	Tamaño	Cantidad de clases
≤ 20	Pequeño	35
$20 < \leq 30$	Medio	2
> 30	Grande	0

Tabla 8: Tamaño del umbral.

Con esta métrica se pudo comprobar que atendiendo a la cantidad de atributos y operaciones, el 94.7 % de las clases diseñadas están consideradas como pequeñas, lo que facilitará el proceso de construcción del proceso Verificación Fiscal.

La métrica Árbol de Profundidad de Herencia fue propuesta por Chidamber y Kemerer (CK). Esta métrica está definida por la máxima longitud que existe entre el nodo y la raíz de árbol. Donde el nodo es una clase hija que hereda de una clase y así respectivamente hasta llegar a la raíz. A medida que esa longitud va creciendo, entonces se van heredando más operaciones y atributos por las clases hijas. Se hace difícil predecir el comportamiento de las clases que se encuentran en los niveles más bajos del árbol. Esta tiene sus ventajas y desventajas. Si los valores de APH son grandes, entonces se garantiza que se reutilice gran cantidad de código, pero al mismo tiempo hace que el diseño sea más complejo. Esto provoca un mayor acoplamiento entre las clases. En este sistema no se presenta un alto nivel de jerarquía de herencia, ya que una vez siendo aplicada esta métrica al diseño propuesto, se obtiene que el resultado que APH tiene valor 1, lo que demuestra la existencia de un bajo acoplamiento en el diseño y es de fácil reparación.

Conclusiones Parciales

En el presente capítulo se realizó el diseño del proceso Verificación Fiscal, generando los diagramas de clases del diseño, facilitándolo al hacer uso de las métricas un resultado factible para el desarrollo del sistema, obteniendo el diseño de las clases pequeñas un valor de 94.7 % demostrando la reutilización de las clases y facilidad a la hora de implementar. Además el uso de patrones que utiliza el Symfony, proporcionaron una forma ágil y rápida en el desarrollo del diseño.

CONCLUSIONES GENERALES

Al concluir el presente trabajo se arriban a las siguientes conclusiones:

El análisis de los diferentes sistemas de gestión fiscal existente en el mundo, las diferentes herramientas CASE, los lenguajes de modelado, las herramientas para la creación de los prototipos, los patrones para facilitar el desarrollo del trabajo y las métricas con el fin de medir la calidad de lo que se desarrollaba, sustentan la fundamentación teórica de este trabajo.

Las diferentes técnicas aplicadas en la elicitación, junto a la interacción con los clientes del sistema, proporcionó identificar con exactitud el proceso Verificación Fiscal y obtener resultados satisfactorios en la identificación de los requerimientos que debe cumplir el sistema.

La elaboración de los artefactos del modelo del sistema posibilitó un entendimiento común entre desarrolladores y clientes en cuanto a las funcionalidades que el sistema debe brindar.

La construcción de los artefactos del modelo de diseño permitió obtener los elementos que deberán ser implementados para obtener un sistema que cumpla con los requerimientos especificados por el equipo de análisis y aprobados por los clientes.

Las validaciones realizadas a los artefactos permitieron medir el grado de factibilidad de estos en cuanto a la calidad, ambigüedad, deficiencias y errores.

RECOMENDACIONES

Tomando en cuentas los resultados obtenidos:

- Se recomienda realizar la gestión de requisitos para tener un control sobre los riesgos que puedan atender con el buen desempeño del software, pues en este negocio se encuentran involucradas leyes que en un futuro pueden cambiar.

BIBLIOGRAFÍA

1996-2010. [En línea] 1996-2010. <http://www.gedex.net> .

2010 . Sistemas Jurídicos S.R.L. [En línea] 2010 . <http://www.lex-doctor.com/about.php>.

A. Smith and P. Spencer. OSRMT . [En línea] <http://sourceforge.net/projects/osrmt>.

Acosta, karine Ramos. 2008. *Ciclo de Vida de proyectos*. 2008.

bizagi. 2009. BIZAGI. [En línea] 2009. [Citado el: 24 de 03 de 2011.] <http://wiki.bizagi.com>.

BOOCH G, Addison-Wesley. 2007. *Object-Oriented Analysis and Design with Applications*. 2007.

Boost Productivity with Innovative and Intuitive Technologies. 2007. [En línea] 2007. <http://www.visual-paradigm.com/aboutus/10reasons.jsp>.

Calisoft. 2009. Centro de Calidad para Soluciones Informáticas. [En línea] 2009. <http://calisoft.uci.cu>.

CEDS. 2009. Centro de Estudios y Diseño de Sistemas. [En línea] Madrid España, 2009. <http://ceds.nauta.es/informes/case01.htm>.

Constitución de la República de Cuba.

Cuba, Fiscalía General de la República de. 1999. *Instrucciones de Verificaciones Fiscales*. Habana : s.n., 1999.

Cuba, Fiscalía General República de. 1997. *Ley No 83, Artículo 8 ley de la Fiscalía General República*. 1997.

Derecho y Justicia - Gobierno de Cuba. [En línea] [Citado el: 3 de febrero de 2011.] www.cubagob.cu/des_soc/derecho.htm.

Durocher, E. 2007. *Business process Managements* . 2007.

Enrique, Galia y Manuel. 2010. *Módulo para la gestión de la entrada y salida de la correspondencia y documentación oficial de la Fiscalía General de la República de Cuba*. Habana : s.n., 2010. 1.

ESCALONA, M. J., KOCH N. 2002. *Ingeniería de Requisitos en Aplicaciones para la Web - Un estudio comparativo*. 2002.

estado, Fiscalía General del. 2009. Fiscal. es. [En línea] 2009. [Citado el: 20 de 04 de 2011.] <http://www.fiscal.es>.

2005 . Estudios y Proyectos Informáticos. [En línea] 2005 . <http://www.abogest.com/enlaces.htm>.

Informática, Instituto Nacional de Estadística e. 2000. Organización y Diseño de la Base de Datos. [En línea] 2000. [Citado el: 2 de 05 de 2011.] <http://www.inei.gob.pe>.

1997- 2010. *INTERCOM.* 1997- 2010.

Jacoboson, I., Booch, G., Rumbaugh J., Addison Wesley. 2000. *El Proceso Unificado de Desarrollo de Software.* 2000.

Justicia, República de Cuba. Ministerio de. 2008. Gaceta Oficial. [En línea] 2008. [Citado el: 22 de 02 de 2011.] <http://www.gacetaoficial.cu/html/fiscaliageneralrepublica.html#FGR>.

Kendall, K. 1997. *Análisis y Diseño de Sistemas.* 1997.

LARMAN C. 1999. *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* México: PRENTICE HALL : s.n., 1999.

Ley no. 83 de la FGR. Cap. 1. Art. 1.

MJ Escalona, N Koch-IDEAS. 2003. sistema. [En línea] 2003. <http://www.sistemas.edu.cu>.

Montilva, Jonás A. 2007. *Modelado de Negocio.* 2007.

Monzón, Antonio. 2000. *CALIDAD DE LA ESPECIFICACIÓN: ¿SE PUEDEN MEDIR LOS REQUISITOS?* 2000.

Olmedilla Arregui, Juan José. 2005. *Revisión Sistemática de Métricas de Diseño Orientada a Objetos.* Universidad Politécnica de Madrid : s.n., 2005.

Orallo, Enrique Hernández. 2007. *El lenguaje Unificado de Modelado (UML).* 2007.

Pressman, Roger S. 2005. *Enfoque Práctico 6ta Edición.* 2005.

Rafael, Juan. 2011. Neleste. *Neleste.* [En línea] 2011. [Citado el: 26 de mayo de 2011.] <http://www.neleste.com/modelo-vista-controlador/>.

Robertson, Suzanne y Robertson, James. 2006. *Mastering the Requirements Process*. s.l. : Second Edition, 2006.

Sanz, L. Fernández. 1990. Institute of Electrical and Electronics Engineers, Institute of Electrical and Electronics Engineers, IEEE Standard 610. Computer dictionary. Compilation of IEEE Standard Computer Glossaries, IEEE, 1990. [En línea] 1990. [Citado el: 26 de 05 de 2011.] <http://www.sc.ehu.es/jiwdocoj/remis/docs/terminos.html>.

Solcre Technology Solutions - www.solcre.com. [En línea] [Citado el: 22 de 03 de 2011.] http://www.solcre.com/files/ventajas_de_las_aplicaciones_web.pdf.

SOMMERVILLE, IAN. 2005. *Ingeniería de Software*. 2005.

Sonda. 2010. Sonda. [En línea] 2010. [Citado el: 23 de 04 de 2011.] <http://www.sonda.com/casos/12>.

2007. Spark Systems. [En línea] 2007. [Citado el: 26 de 03 de 2011.] <http://www.sparxsystems.com.ar/products/ea.html>.

Stephen A. Process Modeling Notation and workflow Patterns. [En línea]

SYSTEMS, SPARX. 2000-2007. SPARX SYSTEMS. [En línea] 2000-2007. <http://www.sparxsystems.com.ar/products/ea.html>.

Visio, Microsoft Office. 2007. 2007.

Visual Paradigm International. 2005. Visual Paradigm. *Visual Paradigm*. [En línea] 2005. <http://www.visual-paradigm.com/product/vpuml/>.

White Stephen A, PHD Dereck Miers. *Guía de Referencia y Modelado*.

ECURED. 2008. ECURED. [En línea] 2008. http://www.ecured.cu/index.php/Rational_Rose_Enterprise_Edition.

Fabien Potencier, François Zaninotto. 2008. *Symfony Guía Definitiva*. 2008.

GALLEGO, JUAN PABLO GOMEZ. 2007. FUNDAMENTOS DE LA METODOLOGIA RUP. [En línea] 2007. <http://es.scribd.com/doc>.

Letelier, Patricio y Penadés, María Carmen. 2008. Metodologías Agiles para el desarrollo de software Extreme Programming. [En línea] 2008. [Citado el: 29 de 03 de 2011.] <http://www.willydev.net/descargas/masyxp.pdf>..

ROJAS, ENRIQUE JOFRÉ. 2002. *MODELO DE DISEÑO Y EJECUCIÓN DE.* 2002.

Rojas, M.C. Juan Carlos Olivares. 2005. *Patrones del Diseño.* México : s.n., 2005.

Rolando Rodríguez Andres. 2008. GestioPolis. [En línea] 2008. [Citado el: 20 de 03 de 2011.] <http://www.gestiopolis.com>.

White, Stephen A. 2008. WE SET THE STANDARD. [En línea] 2008. [Citado el: 24 de 03 de 2011.] <http://www.bpmn.org/>.

GLOSARIO DE TÉRMINO

Proceso de Software: Es un conjunto de actividades y resultados asociados, que generan un producto de software, las cuales son llevadas a cabo por los ingenieros de software.

Sistema: Colección de componentes interrelacionados que trabajan conjuntamente para cumplir algún objetivos.

CASE: (Computer Aided Software Engineering), Ingeniería de Software Asistida por Ordenador).

Ingeniería de Requisitos: Es el uso sistemático de procedimientos, técnicas, lenguajes y herramientas para obtener con un coste reducido el análisis, documentación, evolución continua de las necesidades del usuario y la especificación del comportamiento externo de un sistema que satisfaga las necesidades del usuario.

Metodologías: Es una palabra compuesta por tres vocablos griegos: metà (“**más allá**”), odòs (“**camino**”) y logos (“**estudio**”). Son los métodos de investigación que permiten lograr ciertos objetivos en una ciencia. En otras palabras, la metodología es una etapa específica que procede de una posición teórica y epistemológica, para la selección de técnicas concretas de investigación.

Métrica: Medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado.

Patrón de diseño: Es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

Requisitos: Condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo.

Script: También se lo conoce como un archivo de órdenes o archivo de procesamiento por lotes, es un programa simple, que se almacena en un archivo de texto plano y cuyo uso fundamental resulta a la hora de tener que realizar diversas tareas como ser la combinación de componentes, la interacción con el usuario o con el sistema operativo en cuestión. Facilita la automatización de tareas a través de la creación de pequeñas utilidades.