

Universidad de las Ciencias Informáticas

Facultad 3



Sistema Integrado de Gestión Estadística. Analista
del Módulo Entrada de Datos.

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

Autora: Elizabeth García García

Tutor: Ridosbey Milián Iglesias.

Consultante: Elena Fernández García

Ciudad de la Habana, junio 2007.

“Año 49 de la Revolución.”

Declaración de la Autoría

Yo, Elizabeth García García, declaro que soy la única autora de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los 16 días del mes de junio del año 2007.

Firma del Autor

Firma del Tutor

Dedicatoria

A las personas que más amo en este mundo:

A mi mamá linda que me dio fuerza para seguir hasta el final,

A mi papá que confió en mí con los ojos cerrados,

*A mi hermanita que me aconsejó siempre y a mi mami Isabel y mi papi José
Ángel por formar parte de mi vida.*

Gracias

Agradecimientos

Agradezco a todos aquellos que me ayudaron en la realización de este trabajo:

A mi tutor Ridosbey, por ser un magnífico profesor y amigo.

A la compañera Elena de la Oficina Nacional de Estadísticas, por ayudarme en todo momento.

A los compañeros Cecilia y Ambrustio, especialistas de la Oficina Territorial de Estadísticas de Santiago de Cuba, gracias por toda su cooperación.

A Omar Ahmed y Armando Lobo, integrantes del proyecto ONE, por paciencia.

A mi novio Pedro Carlos, por ser amigo y compañero en los momentos más difíciles.

A todos mis amigos que no me han abandonado nunca, en especial a Hanillilian por ser una hermanita para mí.

Y por último y más especial a mis padres y mi hermanita linda, por creer en mí.

A este país y esta Revolución que me ha dado la posibilidad de formarme como profesional.

Resumen

En Cuba existe un proceso para la gestión estadística, con el objetivo de analizar y procesar información útil para la toma de decisiones de la dirección del país que permita mejorar las condiciones tanto económicas como sociales. Este proceso es dirigido por la Oficina Nacional de Estadísticas (ONE), la cual se encarga de registrar, almacenar y controlar todos los indicadores de producción de las empresas en Cuba, así como los indicadores en diversas áreas de la economía y la sociedad a través del sistema MicroSet NT.

Este sistema está presentando problemas de ejecución, por lo que la ONE necesita un software nuevo que sustituya al obsoleto y perfeccione el trabajo. Se firma así un contrato con la Universidad de las Ciencias Informáticas (UCI) para la implementación del Sistema Integrado de Gestión Estadística (SIGE). Una parte importante de este sistema es el Módulo de Entrada de Datos que tiene como objetivo facilitar la entrada y validación de la información estadística.

Para lograr la implementación exitosa del Módulo de Entrada de Datos del SIGE, se necesita realizar un análisis del sistema que permita el buen entendimiento entre los clientes/usuarios y los desarrolladores. Con un correcto análisis se obtendrá un software que cumpla con las necesidades de la ONE.

La realización satisfactoria del Modelo de Casos de Uso del Negocio, la especificación de requisitos, el Modelo de Casos de Uso del Sistema y la elaboración de un prototipo no funcional permitió que se realizara un buen diseño del sistema y una correcta implementación.

Índice

Declaración de la Autoría	I
Dedicatoria	II
Agradecimientos	III
Resumen.....	IV
Índice.....	V
Introducción.....	1
Capítulo1.....	7
Fundamentación Teórica.....	7
1.1. Introducción.....	7
1.2. Estado actual del negocio estadístico respecto a los programas especializados.....	7
1.3. Estudio de sistemas de gestión de información estadística.....	8
1.4. Metodologías de desarrollo	10
1.5. Herramientas de Modelado	19
1.6. Patrones de Casos de Uso	20
1.7. Técnicas para la captura de requisitos	23
1.8. Conclusiones.....	25
Capítulo 2.....	26
Análisis del Sistema	26
2.1. Introducción.....	26
2.2. Metodología seleccionada	26
2.3. Herramienta de modelado visual utilizada	27
2.4. Modelo del Negocio	27
2.5. Estrategia para la captura de requisitos	37
2.6. Modelo de Casos de Uso del Sistema.....	44
2.7. Conclusiones.....	63
Capítulo 3.....	64

Análisis de los Resultados Obtenidos	64
3.1. Introducción.....	64
3.2. Modelo del Negocio	65
3.3. Especificación de Requisitos.....	66
3.4. Diagrama de Casos de Uso del Sistema	67
3.5. Prototipo no funcional (Interfaz)	69
3.6. Conclusiones.....	70
Conclusiones Generales	71
Recomendaciones	72
Bibliografía.....	73

Introducción

Antecedentes del tema

En Cuba existe un proceso de gestión de información estadística el cual consiste en el procesamiento de datos y su fluidez a través de las Oficinas de Estadísticas (nivel municipal, provincial y nacional) con el objetivo de difundir la información actualizada y fidedigna que permita realizar estudios y tomar decisiones de importancia para la nación. Este proceso en sus inicios se hacía manualmente, haciendo una programación específica para cada modelo de información estadística que se iba a procesar. Posteriormente se implementa el SUPRE (Sistema Uniforme para el Procesamiento Estadístico), un generador de programas que permitía con cierta facilidad generar el código de los programas a utilizar. En 1981 a partir de estudios realizados se llegó a la conclusión de la necesidad que existía de estandarizar los modelos, de tal manera que permitieran a través de un sistema general realizar su captación, procesamiento y emisión de resultados primarios. Una vez logrado este objetivo se desarrolló el sistema SETAMOT, el primer sistema de gestión estadística que facilitara este procesamiento, el cual a los finales de 1987, con la introducción de las microcomputadoras en el país, se programa con esta concepción para ser ejecutada en las nuevas computadoras, con las limitantes de memoria y espacio en disco que tenían en esos momentos. Surge así la primera versión del MicroSet y con el transcurso del tiempo aparecen nuevas versiones, hasta lograr un sistema más completo y eficiente (MicroSet NT versión 2.0) que brindaba además mayores posibilidades, tanto en la validación como en las ediciones, según las particularidades de cada modelo. Este sistema es el más completo de todos los anteriores y es el que han utilizado las oficinas de estadísticas en todo el país para la entrada de datos para los modelos del SIEN excepto en Villa Clara que utilizan el VALTV, sistema reprogramado por uno de los especialistas en informática en las nuevas computadoras. Para los modelos del SIEN que tienen formato de encuestas se realizan programas particulares. Actualmente se está probando, en los centros informantes, para la entrada de datos y el envío por correo en formato MicroSet.txt, un sistema llamado EMODELO.

Con la evolución de las tecnologías y los avances en la informática el sistema MicroSet NT ha estado presentado fallos, en específico problemas de compatibilidad con el nuevo sistema operativo (Windows XP) instalado en las computadoras de todos los territorios utilizadas para la gestión de información (se

tiene que tener en cuenta que este sistema fue programado para MS-DOS) inutilizándose muchas de las funcionalidades del mismo, además han aparecido, con el desarrollo y evolución de la estadística nuevos requerimientos este no están considerados en MicroSet y que son de gran prioridad para el trabajo de los estadísticos, quedando así este sistema ineficiente ante las necesidades actuales del proceso de captación de información estadística.

Sumado a estos problemas está la variedad de programas con los que tienen que trabajar el personal estadístico para el manejo de la información, por causa de estos mismos problemas mencionados con anterioridad, causando a su vez otros problemas para la red estadística nacional (los archivos almacenados con la información estadística no tienen el mismo formato).

Una de las causas fundamentales de estos problemas existentes con el sistema es la pobre o la no realización de un análisis y una especificación de requisitos, además del modelado del sistema durante el proceso de desarrollo de MicroSet (programado por dos estadísticos “tirando líneas de código”), provocando que sea un sistema rígido que no permite la adición de nuevas funcionalidades surgidas con el tiempo y la ampliación o modificación de otras ya contenidas en este sistema.

Con la existencia de estos problemas y sin haber ningún sistema mundialmente reconocido que cumpla con las condiciones del proceso estadístico cubano, se establecen relaciones entre la Oficina Nacional de Estadística (ONE) y la Universidad de las Ciencias Informáticas (UCI) donde se firma un contrato para la implementación de un software (SIGE-Sistema Integrado de Gestión Estadística) que cumpla con todas las funciones del sistema anterior y presente otras nuevas. Para facilitar y optimizar el trabajo se ha dividido el proyecto en módulos, de los cuales uno de los principales es el de Entrada de Datos, que se encarga de toda la gestión de los datos estadísticos (llenar modelos estadísticos, validar modelos estadísticos, etc.) a través de los diferentes niveles (municipal, provincial y nacional), en el que se debe hacer un profundo análisis de software, imprescindible en todo proceso de desarrollo por su gran importancia para el logro de un producto software con calidad.

Para desarrollar una solución software para la entrada de datos estadísticos en toda la red estadística es necesario analizar convenientemente el funcionamiento interno de las oficinas a los distintos niveles, su forma de trabajar, sus procesos y procedimientos. Mediante una buena especificación y análisis de requisitos se podrá llegar a obtener un producto software con calidad que optimizará el rendimiento de

estas oficinas estadísticas en todo el país. Este proceso constituirá una pieza fundamental del desarrollo de software ya que es la base de todo el trabajo.

Para desarrollar bien una solución hay que analizar convenientemente el entorno, todos los pasos del proceso y obtener los datos necesarios de los futuros usuarios, es decir, en esta primera fase, según Roger S. Pressman, se intenta identificar qué información ha de ser procesada, qué función y rendimiento se desea, qué comportamiento del sistema, qué interfaces van a ser establecidas, qué restricciones del diseño existen, y qué criterios de validación se necesitan para definir un sistema correcto. Por tanto, han de identificarse los requisitos clave del sistema y del software (Pressman, 2005).

Problema

No aplicar una Ingeniería de Requisitos que permita el entendimiento común entre los clientes y los desarrolladores para el desarrollo del Módulo de Entrada de Datos del SIGE provoca la implementación de un producto software sin calidad y no deseado por el usuario final.

Objeto de Estudio

Estudio y aplicación de la Ingeniería de Software para proyectos productivos.

Campo de Acción

Estudio y aplicación de la Ingeniería de Requisitos para el Módulo de Entrada de Datos.

Objetivo General

Generar todos los artefactos correspondientes al analista que garanticen un entendimiento común entre los clientes y los desarrolladores del Módulo de Entrada de Datos del SIGE.

Objetivos específicos

- Modelar el negocio.
- Capturar los requisitos del sistema.
- Realizar la modelación de los casos de uso del sistema.
- Obtener el prototipo no funcional.

Tareas

- Recopilar y analizar información necesaria acerca del funcionamiento de las oficinas estadísticas en el país.
- Estudiar el Sistema de Gestión de Información (MicroSet NT) utilizado por las oficinas estadísticas en Cuba.
- Realizar un estudio del arte de sistemas de gestión estadística existentes a nivel mundial.
- Realizar un estudio del arte de metodologías de desarrollo de software y herramientas de modelado visual reconocidas a nivel mundial.
- Estudiar técnicas y métodos utilizados para la captura de requisitos.
- Estudiar los diferentes patrones que existen para los casos de uso (CU).

Hipótesis

Si se logra generar exitosamente los artefactos correspondientes al analista, entonces se llegará a un entendimiento común entre los clientes y los desarrolladores, facilitando así el desarrollo del Módulo de Entrada de Datos del SIGE.

Métodos de investigación utilizados

La utilización de distintos métodos científicos de investigación en este trabajo permite tener una visión clara del problema y una solución mental de la situación. A través de los cuales se pueden realizar acciones en pos de llegar a resultados satisfactorios.

Métodos teóricos: Histórico-lógico, Analítico-Sintético, Método de la Modelación

Se trata la situación problemática y la importancia del rol de analista del sistema desde un enfoque histórico-lógico, haciendo una breve introducción del negocio y un estudio del arte de sistemas anteriores al que se desea implementar y de metodologías de desarrollo que se puede utilizar para el desarrollo del software, utilizando referencias bibliográficas.

A través del método analítico-sintético se dividió el problema planteado para un mejor entendimiento, para luego sintetizar todo lo analizado buscando las relaciones entre cada parte y llegar así a la solución previa.

En este trabajo se modela la realidad del negocio, representando los distintos procesos del negocio y los posibles a automatizar.

Método empírico: Entrevista

En la etapa inicial del trabajo se le realizaron varias entrevistas al usuario interesado en el producto software con el objetivo de obtener información necesaria y suficiente para comenzar a realizar el proyecto, donde el analista del sistema tuvo conocimiento del funcionamiento de las oficinas de estadísticas, las dificultades que presenta el sistema con que están trabajando actualmente estas y las nuevas funcionalidades que desean que tengan el nuevo sistema a implementar.

Estructura del trabajo

Para una mejor organización se ha dividido el trabajo en 3 capítulos:

Capítulo 1:

En este capítulo se muestra todo un estudio del arte de algunas de las metodologías reconocidas mundialmente, de herramientas de modelado visual, así como el estudio de técnicas para la captura de requisitos y patrones para los casos de uso.

Capítulo 2:

En el capítulo 2 se plasma todo lo relacionado con el negocio, la especificación de requisitos y el modelado del sistema con una descripción extendida del mismo.

Capítulo 3:

Se hace un análisis y valoración de los resultados obtenidos en el capítulo 2 a través de métricas aplicadas para medir la calidad de los distintos artefactos desarrollados.

Capítulo 1

Fundamentación Teórica

1.1. Introducción

En la actualidad, el mundo informático ha avanzado de manera tal que le permite al personal técnico escoger las tecnologías y metodologías justas para el desarrollo de software en dependencia del tipo de sistema que se quiera automatizar. En este caso es un Sistema de Gestión de Información Estadística para las Oficinas Nacionales de Estadística de Cuba, donde el analista del sistema debe realizar un profundo estudio y análisis del estado del arte del tema con el objetivo de estar preparado para realizar una debida ingeniería de requisitos.

En este capítulo se abordan temas como el estado actual del negocio y sistemas similares utilizados en estos momentos para la gestión estadística, tanto en Cuba como en otros países, el estudio del estado del arte de las metodologías más utilizadas actualmente para el desarrollo de software, la descripción de herramientas de modelado visual utilizadas para elaborar diagramas y generar artefactos (documentos, código fuente, etc.), así como también el resultado de un estudio realizado sobre las técnicas y métodos utilizados para la captura de los requisitos y el estudio de los diferentes patrones que existen para los casos de uso.

1.2. Estado actual del negocio estadístico respecto a los programas especializados

El análisis estadístico permite realizar pronósticos en varios sectores (económicos y sociales), emitir información de relevancia que puede ser determinante para la toma de decisiones de la directiva del país, acumular datos estadísticos que pueden ayudar a elegir mercados factibles para el desarrollo de alguna rama determinada. En esta tarea los expertos (estadísticos) manejan una gran cantidad de datos y operaciones analíticas complejas, por lo que necesitan software necesario que facilite y agilice su trabajo. En la mayoría de los casos se ven limitados por falta de presupuesto necesario para la compra de estos programas. En esta situación se ven principalmente los países de Latinoamérica, un ejemplo de esto es

Cuba, que debido a los pocos recursos económicos se han visto obligados a la utilización de programas no profesionales, ejemplos de estos programas son: SUPRE(Sistema Uniforme para el Procesamiento Estadístico), TIPIFICADO, el EVA, el SETAMOT, el VALTV, entre otros. A pesar de esto el sistema estadístico cubano es uno de los más confiables del mundo debido a la condición socialista del país ya que se mantiene una política de cooperación mutua y fidedigna entre las oficinas estadísticas y los centros informantes.

1.3. Estudio de sistemas de gestión de información estadística

MicroSet NT

Actualmente en Cuba se utiliza MicroSet NT versión 2.0, diseñado y programado hace aproximadamente 10 años en el lenguaje de programación Borland Pascal versión 7.0, para MS-DOS, producto de varias versiones realizadas con anterioridad debido a la necesidad de los expertos estadísticos en Cuba. Constituye un sistema multiusuario bastante completo para el procesamiento de los modelos estadísticos desde la base (municipio) hasta el nivel de la nación. El MicroSet NT brinda la posibilidad de realizar el diseño de un modelo estadístico, la captación de datos y la realización de consultas (Oficina Nacional de Estadísticas, 1997).

En la actualidad no existe un sistema similar al utilizado en Cuba. Se han implementado otros dedicado solamente al análisis estadístico como son:

SPSS

El SPSS es una herramienta para el análisis de datos; este programa permite realizar diversos tipos de análisis estadísticos de acuerdo a las características de la información que se utilice. A través de SPSS se puede generar diferentes estudios, ya sean descriptivos o de inferencia, permitiendo realizar desde una mera descripción de datos hasta modelos estadísticos (Martínez, 2005).

SAS

El Sistema Estadístico SAS trata de dar una visión extensa del sistema estadístico automatizado más potente que existe en la actualidad, que está introduciéndose progresivamente en las empresas y en todos los niveles de la enseñanza universitaria.

SAS es utilizado actualmente por profesionales de múltiples sectores como la economía, las ciencias sociales, la biología, la medicina, la psicología, el análisis de mercados, la minería de datos y sobre todo en la investigación.

En un mercado tan competitivo como el actual, las organizaciones deben focalizar sus escasos recursos en las estrategias más adecuadas para conducir a la compañía al éxito. El software de SAS ayuda a conseguir el curso para que una compañía llegue al éxito, completando la inversión ya realizada en sistemas operacionales.

El SAS proporciona una plataforma completa, comprensiva e integrada para el análisis de datos. Las capacidades estadísticas extensas del SAS pueden resolver las necesidades de una organización entera. Con el SAS, se puede tener acceso fácilmente a datos de cualquier fuente, realizar gerencia de datos, realizar análisis estadístico y después presentar los resultados en una variedad de informes y de gráficos (todos dentro de un ambiente de software solo, fácilmente manejable). El SAS permite evaluar datos de una variedad de fuentes, incluyendo los ensayos clínicos, bases de datos de la comercialización, encuestas sobre la salud, estudios de la preferencia del cliente, investigación de la bolsa y así sucesivamente.

Los procedimientos listos para utilizar manejan una amplia gama de análisis estadísticos, incluyendo el análisis de la variación, de la regresión, del análisis de datos categórico, del análisis multivariante, del análisis de la supervivencia, del análisis sicométrico, del análisis del racimo, y del análisis no paramétrico (sas, 2007).

CSPro

También se puede mencionar el CSPro que es un software estadounidense, pero es específico para el análisis demográfico con la realización de encuestas a la población pero ya a partir de un modelo estadístico hecho, no así como pasa con el sistema cubano que hace el diseño del modelo.

Statistica

Statistica es un sistema completo para el análisis de datos con miles de pantallas personalizables y gráficos de alta calidad totalmente integrados con todos los procedimientos. Posee soporte para Windows y barra de herramientas con menús desplegables. Amplia selección de procedimientos estadísticos y gráficos. Interpreta los resultados y facilita su comprensión y análisis. Presenta amplias facilidades en la gestión de datos. Con lenguaje de programación STATISTICA BASIC. Administra ficheros de hasta 32.000 variables por registro. Ilimitado tamaño de ficheros, precisión extendida y alta velocidad Intercambio de datos y gráficos con otras aplicaciones, una amplia selección de tipos de ficheros para importar y exportar. Facilidad para diseñar nuevos tipos de gráficos. El sistema STATISTICA ofrece alta velocidad en operaciones de gestión de base de datos, gráficos y cálculos numéricos. Incluye cientos de procedimientos de optimización de la velocidad y también de gestión de la memoria. Guarda los datos con la máxima precisión y puede manejar incluso casos extremos de varianza relativa pequeña (Statistica, 2004).

Haciendo una comparación del proceso estadístico en Cuba con el de otros países, se llega a la conclusión de que no puede existir un producto software para la gestión de información estadística similar al cubano, debido a que en los otros países se capta la información estadística a través de censos y encuestas y son mayormente datos económicos, no siendo así en Cuba, donde se capta mediante un proceso de estadística continua, lo que influye en que la información que llegue sea la más real posible. En otros países no pasa lo mismo pues son propiedades privatizadas, donde los expertos tienen que realizar sus análisis estadísticos con los datos que le entregan las empresas, ya sean falsos o verdaderos. Además estos sistemas son propietarios (se debe pagar una licencia para poder utilizarlos) y son programados en la mayoría de las ocasiones para cumplir un objetivo específico (ejemplo: son demográficos, educacionales, de salud, etc.), pocos pueden ser utilizados para varios trabajos; sin embargo el sistema que necesita el país debe ser un sistema integrado de gestión estadística que homogenice el trabajo estadístico en todo el país.

1.4. Metodologías de desarrollo

En este trabajo se ha realizado un estudio del estado del arte de las metodologías de desarrollo de software más nombradas a nivel mundial en estos momentos.

El objetivo fundamental de las metodologías de desarrollo de software es elevar la calidad del producto. Una correcta elección de la metodología, en correspondencia con las características del proyecto, garantiza la terminación del software en el tiempo planificado y asegura que no existan insatisfacciones por parte del cliente y de los desarrolladores, ya que pueden aparecer requisitos adicionales en el transcurso del proceso y si no existe una buena definición de la metodología puede implementarse un sistema rígido al cual no se le pueden adicionar nuevas funcionalidades, apareciendo entonces las discrepancias entre clientes y desarrolladores.

Antes de empezar un proyecto siempre el analista debe analizar qué metodología va a utilizar para un adecuado desarrollo del software. La definición de la metodología de desarrollo a utilizar depende de las características del proyecto, por ejemplo: su duración y volumen de complejidad y el presupuesto con que se cuenta.

Con el objetivo de ajustar más las metodologías de desarrollo de software al proyecto en el que se va a trabajar, se han dividido estas en dos grupos: las metodologías ágiles o ligeras, que tratan de mejorar la calidad del software por medio de una comunicación directa e inmediata entre las personas que intervienen en el proceso, y las metodologías pesadas o robustas, que intentan conseguir el objetivo común por medio de orden y documentación (Molpeceres, 2002).

Extreme Programming (XP)

Se dice que XP es una metodología ágil que intenta reducir la complejidad del software por medio de un trabajo orientado directamente al objetivo, basado en las relaciones interpersonales y la velocidad de reacción (Molpeceres, 2002).

Es una de las metodologías más exitosas en la actualidad utilizada para proyectos de corto plazo, con poco personal y cuyo plazo de entrega es inminente. Tiene como requisito particular para el éxito del proyecto que el usuario final sea parte del equipo de desarrollo, existiendo así una estrecha relación desarrollador-cliente.

XP define UserStories (historias del usuario) como base del software a desarrollar. Se utiliza para especificar los requisitos del software a través de tarjetas que hace el cliente donde describe las características que debe tener el sistema.

Según Mendoza (Sanchez, 2004) XP se basa en:

Pruebas unitarias: pruebas que se le realizan a los principales procesos de tal manera que se puedan anticipar los posibles errores que puede tener el software.

Refabricación: reutilización de código, creándose patrones o modelos estándares, siendo más flexible a futuros cambios.

Programación en pares: dos desarrolladores participan en un mismo flujo de trabajo, de manera que cada uno lleva la acción que el otro no está haciendo en ese momento (chofer-copiloto). Por lo que se espera que la calidad del software suba en el mismo momento de escribirlo

Además, XP sigue un diseño evolutivo, que consiste en conseguir la funcionalidad deseada de la forma más sencilla posible, lo que hace que no se le de apenas importancia al análisis como fase independiente, puesto que se trabaja exclusivamente en función de las necesidades del momento.

XP no define muchas tareas organizativas sobre los desarrolladores (como modelado o el desarrollo de documentación), lo que se minimiza por medio de la presencia de un representante del cliente).

XP es la metodología ideal para proyectos en los cuales el equipo de desarrollo es pequeño, con plazos reducidos, requisitos volátil y nuevas tecnologías, donde el contexto es muy cambiante, y en donde se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad.

Según Beck (Beck, 1999), los roles de XP son:

Programador. El programador escribe las pruebas unitarias y produce el código del sistema.

Cliente. Escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio.

Encargado de pruebas (Tester). Ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.

Encargado de seguimiento (Tracker). Proporciona realimentación al equipo. Verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, para mejorar futuras estimaciones. Realiza el seguimiento del progreso de cada iteración.

Entrenador (Coach). Es responsable del proceso global. Debe proveer guías al equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.

Consultor. Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto, en el que puedan surgir problemas.

Gestor (Big boss). Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

Microsoft Solution Framework (MSF)

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas (Sanchez, 2004).

Entre las características que presenta MSF se puede resaltar que es: adaptable, escalable (sirve tanto para equipos pequeños como para grandes), flexible (se utiliza en el ambiente de desarrollo de cualquier cliente); puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología por lo que se caracteriza por tecnología agnóstica.

MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto (Sanchez, 2004):

Modelo de Arquitectura del Proyecto: Diseñado para acortar la planificación del ciclo de vida. Este modelo define las pautas para construir proyectos empresariales a través del lanzamiento de versiones.

Modelo de Equipo: Este modelo ha sido diseñado para mejorar el rendimiento del equipo de desarrollo. Proporciona una estructura flexible para organizar los equipos de un proyecto. Puede ser escalado dependiendo del tamaño del proyecto y del equipo de personas disponibles.

Modelo de Proceso: Diseñado para mejorar el control del proyecto, minimizando el riesgo, y aumentar la calidad acortando el tiempo de entrega. Proporciona una estructura de pautas a seguir en el ciclo de vida del proyecto, describiendo las fases, las actividades, la liberación de versiones y explicando su relación con el Modelo de equipo.

Modelo de Gestión del Riesgo: Diseñado para ayudar al equipo a identificar las prioridades, tomar las decisiones estratégicas correctas y controlar las emergencias que puedan surgir. Este modelo proporciona un entorno estructurado para la toma de decisiones y acciones valorando los riesgos que puedan provocar.

Modelo de Diseño del Proceso: Diseñado para distinguir entre los objetivos empresariales y las necesidades del usuario. Proporciona un modelo centrado en el usuario para obtener un diseño eficiente y flexible a través de un enfoque iterativo. Las fases de diseño conceptual, lógico y físico proveen tres perspectivas diferentes para los tres tipos de roles: los usuarios, el equipo y los desarrolladores.

Modelo de Aplicación: Diseñado para mejorar el desarrollo, el mantenimiento y el soporte, proporciona un modelo de tres niveles para diseñar y desarrollar aplicaciones software. Los servicios utilizados en este modelo son escalables, y pueden ser usados en un solo ordenador o incluso en varios servidores.

Object Modeling Technique (OMT)

OMT es una de las metodologías de análisis y diseño orientadas a objetos más madura y eficiente que existe en la actualidad. Fue creada por James Rumbaugh y Michael Blaha en 1991. La gran virtud que aporta esta metodología es su carácter de abierta (no propietaria), que le permite ser de dominio público y, en consecuencia, sobrevivir con enorme vitalidad. Esto facilita su evolución para acoplarse a todas las necesidades actuales y futuras de la ingeniería de software (Chávez Gaona, et al., 1997).

Presenta varias fases:

Análisis: El analista construye un modelo del dominio del problema. El modelo de análisis es una abstracción resumida y precisa de lo que debe de hacer el sistema deseado y no de la forma en que se hará. Los elementos del modelo deben ser conceptos del dominio de aplicación y no conceptos informáticos tales como estructuras de datos. Un buen modelo debe poder ser entendido y criticado por expertos en el dominio del problema que no tengan conocimientos informáticos.

Diseño del sistema: El diseñador del sistema toma decisiones de alto nivel sobre la arquitectura del mismo. Durante esta fase el sistema se organiza en subsistemas basándose tanto en la estructura del análisis como en la arquitectura propuesta. Se selecciona una estrategia para afrontar el problema.

Diseño de objetos: El diseñador de objetos construye un modelo de diseño basándose en el modelo de análisis, pero incorporando detalles de implementación. El diseño de objetos se centra en las estructuras de datos y algoritmos que son necesarios para implementar cada clase. OMT describe la forma en que el diseño puede ser implementado en distintos lenguajes (orientados y no orientados a objetos, bases de datos, etc.).

Implementación: Las clases de objetos y relaciones desarrolladas durante el análisis de objetos se traducen finalmente a una implementación concreta. Durante la fase de implementación es importante tener en cuenta los principios de la ingeniería del software de forma que la correspondencia con el diseño sea directa y el sistema implementado sea flexible y extensible.

La metodología OMT emplea tres clases de modelos para describir el sistema (Chávez Gaona, et al., 1997):

Modelo de objetos. Describe la estructura estática de los objetos del sistema (identidad, relaciones con otros objetos, atributos y operaciones). El modelo de objetos proporciona el entorno esencial en el cual se pueden situar el modelo dinámico y el modelo funcional. El objetivo es capturar aquellos conceptos del mundo real que sean importantes para la aplicación. Se representa mediante diagramas de objetos.

Modelo dinámico. Describe los aspectos de un sistema que tratan de la temporización y secuencia de operaciones (sucesos que marcan los cambios, secuencias de sucesos, estados que definen el contexto para los sucesos) y la organización de sucesos y estados. Captura el control, aquel aspecto de un sistema que describe las secuencias de operaciones que se producen sin tener en cuenta lo que hagan las operaciones, aquello a lo que afecten o la forma en que están implementadas. Se representa gráficamente mediante diagramas de estado.

Modelo funcional. Describe las transformaciones de valores de datos (funciones, correspondencias, restricciones y dependencias funcionales) que ocurren dentro del sistema. Captura lo que hace el sistema,

independientemente de cuando se haga o de la forma en que se haga. Se representa mediante diagramas de flujo de datos.

Entre las herramientas CASE que soportan OMT se pueden mencionar: ObjectMarker, BOCS, ObjectTeam, OMTool, Paradigm Plus, Software Through Pictures, System Architect, entre otros.

De manera general se puede decir que OMT es una tecnología fuerte en el análisis, por lo que puede ser utilizada para cualquier actividad de la Ingeniería de Software que necesite de un análisis para resolver un problema. Puede ser utilizada en aspectos de implementación incluyendo multimedia, web, cliente/servidor, base de datos orientada a objetos, etc.

OMT es un método simple que se puede utilizar principalmente en pequeños proyectos. Sirve de base para otras metodologías más modernas y complejas como es RUP (Rational Unified Process).

Rational Unified Process (RUP)

RUP es uno de los procesos más generales que existe, ya que está dirigido a adaptarse al proyecto más que al propio software. Es una metodología preparada para desarrollar grandes y complejos proyectos. Utiliza UML como lenguaje de representación visual.

El ciclo de vida de RUP está *dirigido por casos de uso* (CU) que reflejan lo que los usuarios futuros necesitan y desean (los CU guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los CU, es decir, cómo se llevan a cabo); *centrado en la arquitectura* (la arquitectura muestra la visión común del sistema completo, describe los elementos del modelo que son más importantes para su construcción, los cimientos que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente); e *iterativo e incremental* (propone que cada fase se desarrolle en iteraciones. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto).

RUP es un proceso que en su modelación define como principales elementos:

Trabajadores: define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo.

Actividades: son tareas que tienen un propósito claro, son realizadas por los trabajadores y manipulan elementos.

Artefactos: productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser: un documento, un modelo, un elemento de modelo, etc.

Flujo de actividades: secuencia de actividades realizadas por trabajadores y que producen un resultado de valor observable.

RUP se divide en 4 fases el desarrollo del software (Molpeceres, 2002):

Inicio: en esta etapa se determina la visión del proyecto.

Elaboración: se determina la arquitectura óptima.

Construcción: se obtiene la capacidad operacional inicial.

Transmisión: se obtiene el release del proyecto.

En cada una de estas fases se ejecutarán una o varias iteraciones, y dentro de cada una de ellas seguirá un modelo de cascada para los distintos flujos de trabajos.

RUP define nueve flujos de trabajo (los seis primeros son de ingeniería y los tres últimos de apoyo) (Molpeceres, 2002). Estos son:

Modelamiento del Negocio: Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.

Requerimientos: Define qué es lo que el sistema debe hacer

Análisis y diseño: Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos).

Implementación: Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.

Prueba (Testeo): Busca los defectos a lo largo del ciclo de vida.

Instalación: Produce release del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.

Administración del proyecto: Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.

Administración de configuración y cambios: Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.

Ambiente: Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

Además RUP define un grupo de roles que son repartidos entre los integrantes del proyecto según sus competencias y define que tareas y resultados (artefactos) deben desarrollar en el transcurso el ciclo de vida del proyecto. Entre los roles que define RUP se encuentran:

Analista: el rol de analista agrupa un conjunto de roles primarios implicados en la captura y elicitación de requisitos.

Analista del negocio: es responsable de definir la arquitectura del negocio, y de definir a los casos uso y los actores del negocio, y cómo estos interactúan.

Diseñador del negocio: detalla la especificación de una parte de la organización.

Analista de sistema: conduce y coordina la especificación de requisitos y el modelado de los casos de uso, contorneando la funcionalidad del sistema y delimitando el sistema; por ejemplo, identifica qué actores existen y qué casos de uso requerirán al interactuar con el sistema.

Especificador de requisitos: especifica los detalles de una o más de una de las partes de la funcionalidad del sistema a través de la descripción de los requisitos.

Desarrollador: el rol de desarrollador organiza los roles implicados sobre todo en el diseño y la implementación del software.

Arquitecto del software: es responsable de la arquitectura del software, que incluye las decisiones técnicas claves que rigen todo el diseño y la implementación del proyecto.

Diseñador: es el responsable de diseñar una parte del sistema.

1.5. Herramientas de Modelado

Las herramientas CASE son aplicaciones informáticas que sirven de gran apoyo al equipo de desarrollo en todos los aspectos del ciclo de vida de desarrollo del software, ya que reducen esfuerzo, tiempo y dinero. Con ellas se puede diseñar diagramas que ayudan a un claro entendimiento del negocio y las posibles aristas de soluciones, documentar todo el proyecto o detectar errores en fases iniciales que pueden ser arreglados a tiempo sin provocar retrasos del proyecto ni incumplimientos, entre otras facilidades que brindan. Entre las más conocidas están:

Rational Rose

Es una herramienta de modelamiento visual de gran alcance. Sirve de ayuda en el análisis y el diseño de sistemas de software orientados a objetos. Se utiliza para modelar el sistema antes de escribir cualquier código, por lo que garantiza que el sistema esté arquitectónicamente bien desde el principio. Usando el modelo, se pueden coger defectos de diseño.

Rational Rose sirve de soporte para el modelo del negocio, ayudando a entender el negocio a lrededor del sistema. Apoya el análisis de los sistemas permitiendo diseñar casos del uso y utilizar diagramas del caso para demostrar la funcionalidad del sistema. Permite diseñar diagramas de interacción para demostrar cómo los objetos trabajan juntos para proporcionar la funcionalidad necesaria. Los diagramas de clase se pueden crear para mostrar las clases en un sistema y cómo se relacionan la una con la otra. Los diagramas de componentes se pueden desarrollar para ilustrar cómo existe una traza entre las clases y la implementación de los componentes. Finalmente, el diagrama del despliegue que se puede desarrollar para el diseño de red para el sistema.

Visual Paradigm (VP)

Es una poderosa herramienta CASE para la representación visual de UML. Está diseñada para usuarios como: los ingenieros de software, analistas del sistema, analistas del negocio y los arquitectos, que están interesados en la construcción de sistemas de software confiables. El ambiente de VP-UML proporciona medios para realizar el análisis y el diseño del sistema orientado a objeto, donde se puede crear diagramas de UML con operaciones simples de la fricción y de la gota. Permite crear diversos tipos de diagramas en un ambiente totalmente visual.

Enterprise Architect (EA)

Es una herramienta automatizada que se utiliza para diseñar y construir sistemas de software. El EA soporta la especificación de UML 2.0, que describe un lenguaje visual para definir mapas o modelos de un proyecto. El EA es una herramienta progresiva que soporta todos los aspectos del ciclo de desarrollo, proporcionando completa trazabilidad de la fase de diseño inicial al despliegue y mantenimiento. También soporta control de pruebas, de mantenimiento y de cambios.

1.6. Patrones de Casos de Uso

Para la modelación del negocio, el analista se puede apoyar en los patrones de casos de uso que existen, para un fácil manejo del problema.

Los patrones de casos de uso son diseños que están probados en modelos de casos de uso, que tienen una descripción donde se especifica en que situación debe ser utilizado y las consecuencias que tiene su aplicación en un modelo de caso de uso.

Existen varios tipos de patrones, de los que se pueden mencionar algunos de los más importantes y utilizados por los analistas en la Universidad: Patrón de Reglas del Negocio (ver figura 1), Patrón de Concordancia (ver figura 2), Patrón de Múltiples Actores (ver figura 3), Patrón CRUD. (Övergaard, et al., 2004)

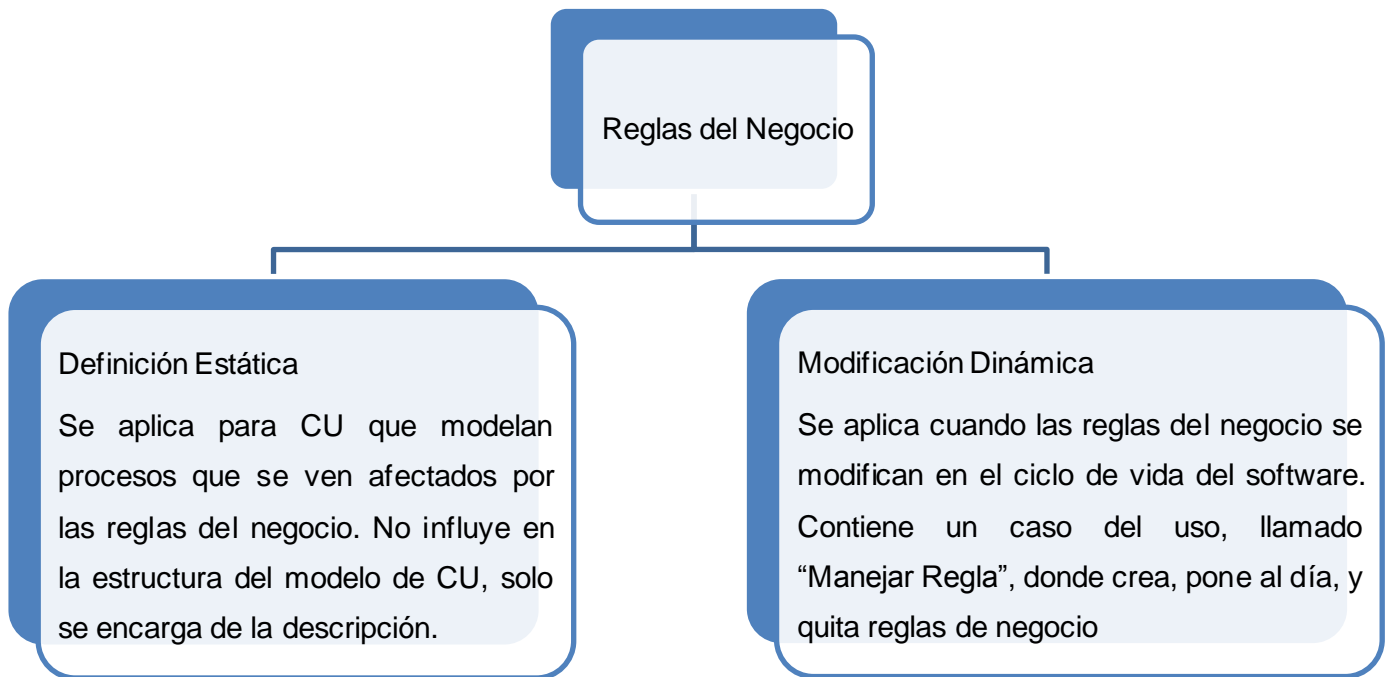


Figura1. Patrón de Reglas del Negocio

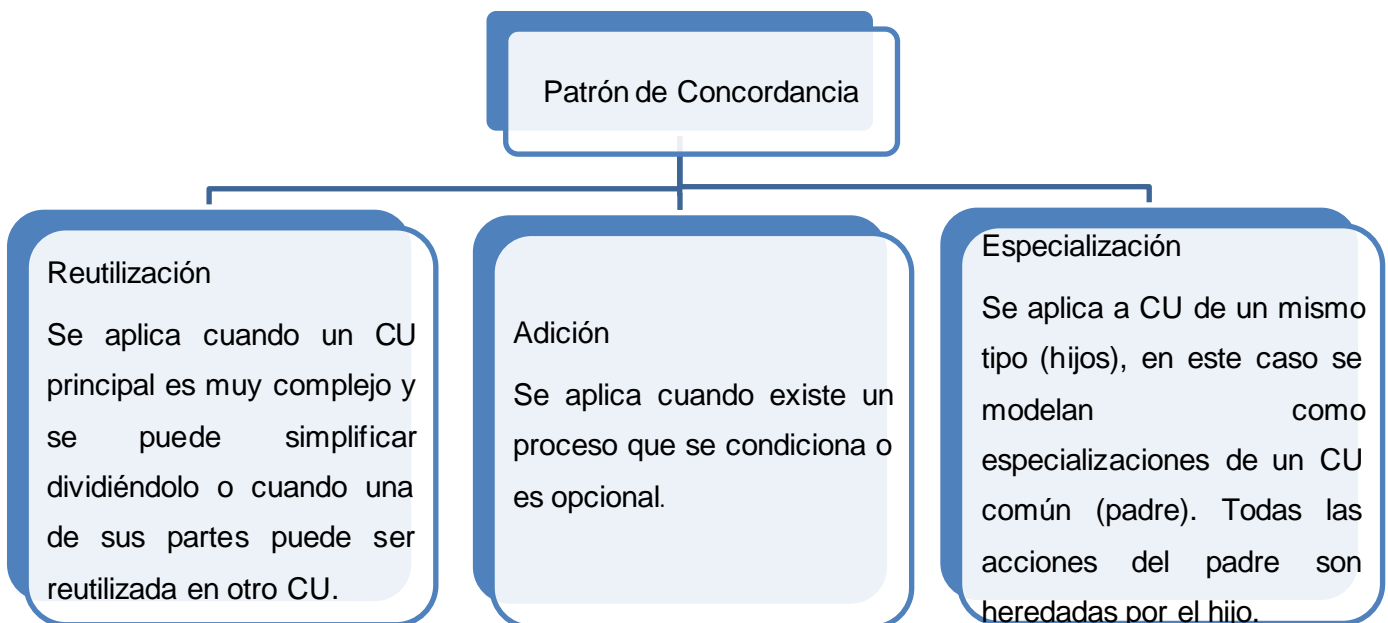


Figura2. Patrón de Concordancia

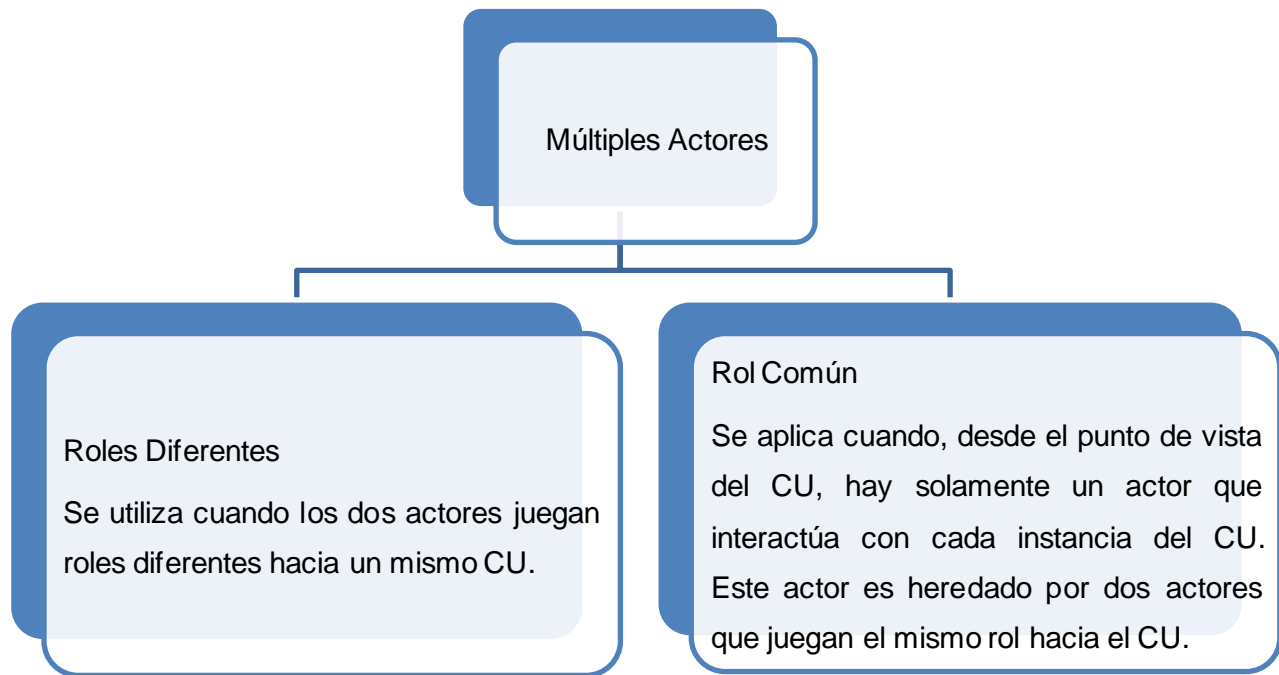


Figura3. Patrón de Múltiples Actores

- CRUD: crear (creating), leer (reading), actualizar (updating), borrar (deleting) información dentro de un CU formando una unidad conceptual. Se divide en completo (consiste en un caso de uso que modela todas las diferentes operaciones que se pueden realizar a la información contenida en este caso de uso, como crear, actualizar, borrar, etc.) y parcial (en este caso se separan los caso de uso, esta alternativa se usa cuando el caso de uso es más significativo, más largo o mucho más complejo que la otra alternativa).

Existen otros patrones no menos importantes pero sí más complejos y más específicos, no siendo estos de interés de estudio, como son:

- Jerarquía de Componentes, que encapsula: Caja Negra con CU y Caja Blanca.
- Extensión o Inclusión Concreta.
- Caso de Uso Grande. Se divide en: Secuencia Larga y Trayectorias Múltiples.
- Sistema en Capas: la estructura del modelo de CU de cada CU esta definida dentro de una capa.

- Servicio Opcional: partes separadas obligatoriamente de los casos del uso de las partes opcionales que se pueden pedir y entregar por separado.
- Vistas Ortogonales: proporciona diversas vistas de los flujos de un sistema que son percibidos distintamente por diversos stakeholders.
- Secuencia de Caso de Uso: expresa el orden temporal entre una colección de CU que se deben invocar solamente en un orden específico, aun cuando los casos del uso están funcionalmente sin relación el uno con el otro.

1.7. Técnicas para la captura de requisitos

La captura de requisitos se realiza a través de encuentros con los clientes de los cuales se extrae todos los datos necesarios. Para la realización segura y eficiente de esta obtención de información, la Ingeniería de Requisitos ha desarrollado técnicas que son utilizadas actualmente por los equipos de desarrollo, especialmente los analistas. Entre las técnicas que se han desarrollado están:

Introspección

Esta técnica consiste en que el analista se pone en el lugar del cliente y empieza a suponer que funcionalidades debe tener el sistema y a recomendarle al cliente los requisitos. El problema de esta técnica es que el analista puede ser un experto en la ingeniería de software pero no está especializado en el negocio y puede que el sistema tenga funcionalidades innecesarias o que no cumpla con las necesidades del cliente.

Entrevistas

La entrevista consiste en una serie de encuentros y reuniones donde el analista realiza un conjunto de preguntas a los clientes con el objetivo de conocer lo que desean estos que el sistema tenga, es decir, las funcionalidades del sistema.

En las entrevistas se puede identificar tres fases: preparación, realización y análisis.

Para una óptima captura de requisitos existen varios tipos de entrevistas, entre las que están:

Entrevistas de Cuestionarios: este tipo de entrevista consiste en preparar previamente un cuestionario de todas las posibles preguntas a realizar y aplicarlo a los clientes para comenzar la captura de requisitos.

Open ended Interview (Entrevista Ampliable): este tipo de entrevista es más abierta ya que se deja que el cliente hable sobre su negocio y la problemática existente, durante todo el encuentro el analista va a ir guiando la conversación con el objetivo de ir determinando los requisitos.

Discusiones: con este tipo de entrevista se entabla un debate o discusión entre los clientes y analistas sobre el problema del negocio para en conjunto sacar los requisitos.

JAD (Joint Application Development, Desarrollo Conjunto de Aplicaciones)

Se desarrolla a lo largo de un conjunto de reuniones en grupo durante un período de 2 a 4 días. En estas reuniones se ayuda a los clientes y usuarios a formular problemas y explorar posible soluciones, involucrándolos y haciéndolos sentirse partícipes del desarrollo (Software, 2006).

Esta técnica se basa en cuatro principios: dinámica de grupo, el uso de técnicas audiovisuales para mejorar la comunicación (diagramas, multimedia, herramientas CASE, etc.), mantener un proceso organizado y racional, y una filosofía de documentación WYSIWYG (What You See Is What You Get- Lo Que Ves Es Lo que Obtienes) (Software, 2006)

Análisis de Protocolo

Esta técnica se pone en práctica cuando el cliente ya va al encuentro con una idea o un modelo mental del sistema que desean y a partir de este esquema se realiza un análisis y se determinan los requisitos. Esta técnica casi no se utiliza porque es muy poca las veces que los clientes llevan un modelo predeterminado del sistema que desean que se implemente.

Brainstorming (Tormenta de Ideas)

Es una técnica de reuniones en grupo cuyo objetivo es generar ideas en un ambiente libre de críticas o juicios. Puede ayudar a generar una gran variedad de vistas del problema y a formularlo de diferentes formas, aunque no con una gran calidad de detalles. Esta técnica requiere poca organización y es fácil de aprender.

Casos de Uso

Esta técnica es una de las que más se utiliza, consiste en la captura de todas las funciones del sistema y en base a cada una de ellas se realiza la especificación de los requisitos de ese sistema.

Un caso de uso es la descripción de una secuencia de interacciones entre el sistema y uno o más actores, en la que los actores obtienen resultados observables.

Los casos de uso facilitan la adquisición de requisitos y son fácilmente comprensibles por los clientes y los usuarios

VORD

Esta técnica es utilizada para capturar requisitos en base a puntos de vista. Es utilizado en sistemas que van a ser desarrollados con el paradigma de programación orientados a objetos (scholar, 2007).

Todas estas técnicas no son eficientes si se aplican de forma independiente, se deben combinar unas con otras para una correcta captura de requisitos.

1.8. Conclusiones

El estudio de algunas de las metodologías de desarrollo de software y de las herramientas de modelado visual ha permitido ampliar el conocimiento del analista, siendo capaz de seleccionar de esta manera las tecnologías adecuadas, es decir, las que más se acoplan al proyecto, garantizando un trabajo eficiente. El estudio de distintas técnicas para la captura de requisitos posibilitó que se obtuvieran los requisitos deseados por el sistema, fundamental para obtener un alto de calidad de la funcionalidad del sistema. El estudio de los patrones par los casos de uso ayudaron a modelar los diagramas de casos de uso adecuadamente.

Capítulo 2

Análisis del Sistema

2.1. Introducción

Para hacer una propuesta de la solución del Módulo de Entrada de Datos como analista se ha seleccionado la metodología de desarrollo de software a utilizar (RUP), teniendo en cuenta las tendencias actuales, a partir de la cual se ha seleccionado la herramienta de modelado visual (Rational Rose) que facilite el desarrollo de los artefactos correspondientes.

En este capítulo se define la estrategia de captura de requisitos para la obtención de los requerimientos pedidos por los clientes y su adecuada traducción a los desarrolladores.

Se han seleccionado además, los patrones de casos de uso que se aplicarán para conformar el diagrama de casos de uso del sistema.

2.2. Metodología seleccionada

Cada metodología tiene su particularidad, cuando se habla de RUP se puede decir que se adapta a proyectos de largo plazo; MSF se adapta a proyectos de cualquier dimensión y de cualquier tecnología; mientras que XP es una metodología ágil, útil para proyectos de corto plazo y OMT es una metodología no propietaria expuesta a adaptarse a cualquier futuro cambio en la Ingeniería de Software, lo que da ventajas a países como Cuba a insertarse en el mercado del software libre.

Sin embargo en estos momentos la metodología que se escoge para el análisis del Módulo de Entrada de Datos es RUP, por ser una metodología robusta que se acopla fácilmente a proyectos de gestión de largo plazo.

Además RUP define un conjunto de actividades que son realizadas por un conjunto individuos que se agrupan en el rol de analista, el cual genera artefactos como: el modelo de casos de uso del negocio, la especificación de requisitos, el diagrama de casos de uso del sistema y el prototipo de interfaz de usuario

no funcional, que permiten desarrollar una propuesta de solución para el futuro diseño y la implementación del módulo.

A parte de todo lo analizado, se debe tener en cuenta que RUP es la metodología más conocida entre los equipos de desarrollo de la Universidad de las Ciencias Informáticas (UCI) y en otros centros de producción de software en Cuba. Por lo que se hace más factible la utilización de la misma en el proceso de desarrollo del sistema a implementar.

2.3. Herramienta de modelado visual utilizada

Para facilitar el trabajo se ha seleccionado Rational Rose 2003, herramienta CASE que permite realizar el modelamiento del negocio, entre otras funcionalidades. Presenta varias vistas que permite la organización del trabajo en las distintas iteraciones. Además cuenta con paquetes como el Rational Pro que permite el desarrollo de artefactos como la especificación de los requisitos. Rational es la herramienta de modelado visual que más se acopla a la metodología utilizada (RUP).

2.4. Modelo del Negocio

Según I. Jacobson, para capturar los requisitos correctos y para construir el sistema correcto el ingeniero de software (analista) debe tener un firme conocimiento del entorno en el que se mueve el negocio.

El modelado del dominio y modelado del negocio son las dos vías de traducir el entorno del negocio de una manera entendible para los desarrolladores de software.

El objetivo del modelado del negocio es describir los procesos -existentes u observados- con el objetivo de comprenderlos. El modelado del negocio especifica qué procesos de negocio soportará el sistema (Rumbaugh, et al., 2000).

Para comprender la estructura y dinámica del proceso de gestión de información en las oficinas de estadística, comprender los problemas que presentan actualmente las mismas, asegurar un entendimiento

común entre los clientes, usuarios y desarrolladores, y obtener los requisitos del sistema se realizó el modelamiento del negocio, obteniendo una visión de la organización que permitió definir los procesos, roles y responsabilidades de la organización a través de los diagramas de casos de uso, la realización de los casos de uso y el modelo de clase de objetos.

En nuestro país se almacena información estadística a todos los niveles, con el objetivo de conformar un sistema estadístico nacional potente, que contenga información actualizada y fidedigna que permita realizar estudios y tomar decisiones de importancia para el territorio nacional. Para esto se cuenta con un conjunto de Oficinas de Estadísticas distribuidas por todo el país que son las encargadas de recopilar, almacenar y procesar la información que se obtiene de las diferentes empresas e instituciones cubanas.

La red de oficinas de estadísticas está conformada por una ONE que constituye la institución rectora de las estadísticas en Cuba, un conjunto de OTE situadas una en cada provincia que se subordinan a la ONE y finalmente un conjunto de OME, una por cada municipio que se subordinan a la OTE de su provincia. Cada una de estas instituciones tiene como tarea principal obtener la información estadística de su dominio y informarla a la oficina de estadísticas a la que se subordina, de este modo la información fluye desde los municipios hasta las provincias y luego a la nación que es la que contiene realmente la información de todo el país.

Para la obtención de toda la información estadística, cada oficina se apoya en un conjunto de CI que son los que rinden la información que esta debe almacenar, procesar y informar a instancias superiores.

Actores del Negocio

Actor	Descripción
Oficina Estadística	Representa al personal de las oficinas de estadísticas que solicitan los partes a las oficinas subordinadas, intercambian información con los centros informantes y solicitan los chequeos de información.

Eventos del negocio

Evento	Descripción
Llegada de Información Estadística	Representa el envío de la de información estadística por el CI y la llegada de esta información a las oficinas de estadísticas. Este evento ocurre de manera personal, es decir, la información estadística se entrega en la oficina correspondiente en formato palpable y de persona a persona.

Trabajadores del Negocio

Trabajador	Descripción
Responsable de Estadística CI	Este trabajador del negocio es el encargado de enviar los datos estadísticos desde los centros informantes hasta la oficina de estadísticas.
Digitador	Es la persona encargada de llenar los modelos con los datos estadísticos.
Validador	Es la persona encargada de validar la información estadística que se digita en las oficinas estadísticas.
Responsable del Departamento de Informática	Este es el trabajador que recibe la información validada y la comprime, encripta y envía a la Oficina inmediata superior.

Diagramas de Casos de Uso del Negocio

Con el propósito general de organizar el trabajo de manera que fuera más entendible el negocio por los desarrolladores, se agruparon los casos de uso del negocio en dos paquetes (subsistemas), dividiendo así el diagrama en subdiagramas más pequeños que se pueden relacionar entre sí (ver figura4).

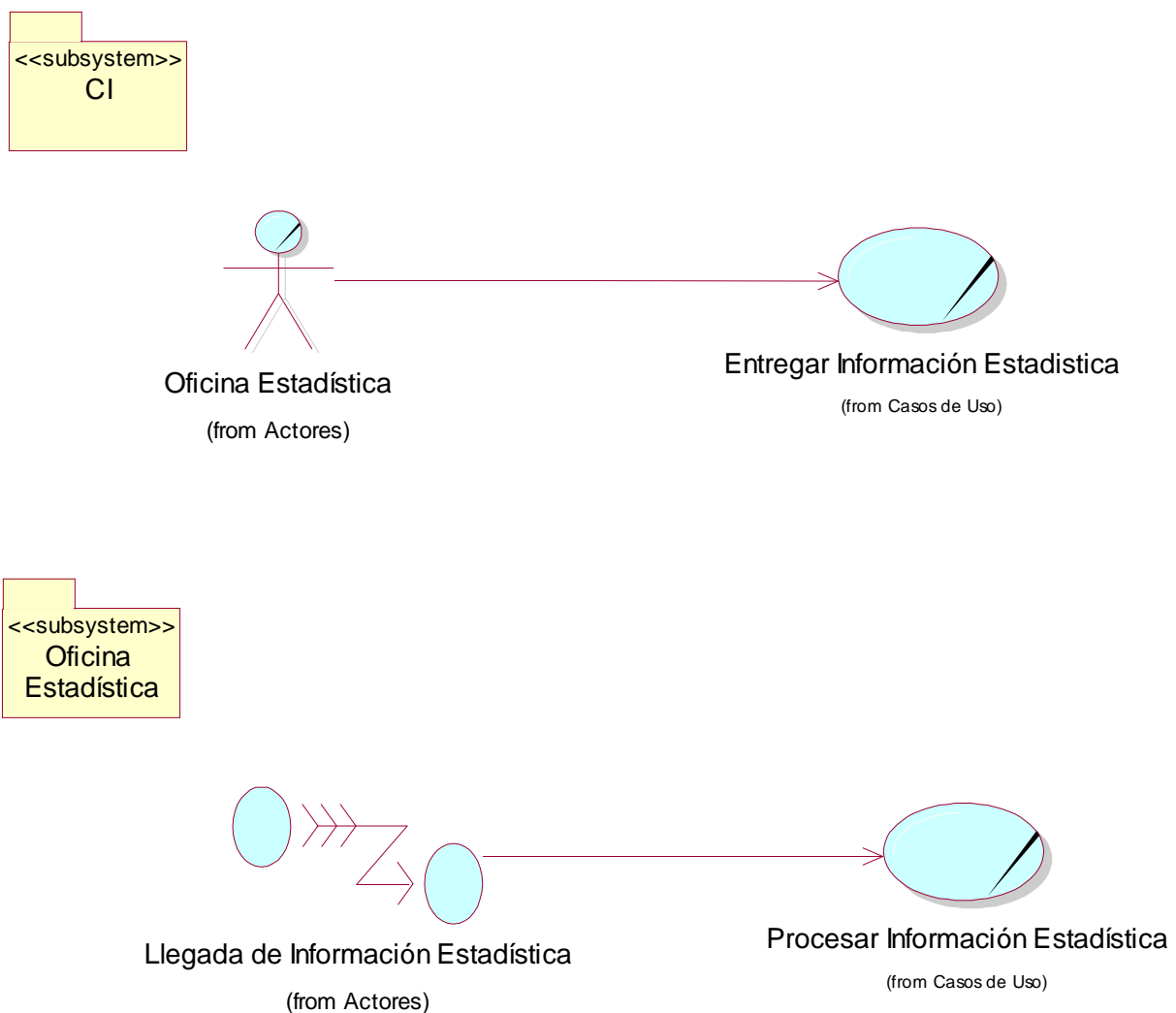


Figura4. Diagramas de Casos de Uso del Negocio.

Realización de los Casos de Uso del Negocio

Con la realización de los casos de uso se logra una descripción más detallada del negocio, donde los casos de uso pueden ser especificados lo mismo textual, mediante un documento llamado “Especificación del caso de uso de negocio” que gráficamente mediante un diagrama en UML (Diagrama de Actividades).

Descripción textual CU: *Entregar Información Estadística.*

Caso de Uso	Entregar Información Estadística	
Actores	Oficina Estadística	
Trabajadores	Responsable de Estadística CI	
Resumen	El caso de uso comienza cuando la oficina de estadística solicita la información estadística, el responsable de estadística del centro informante llena el modelo estadístico y se lo entrega a la oficina para que sea procesada la información.	
Precondiciones	<ul style="list-style-type: none"> •El CI debe conocer los periodos de entrega de los diferentes modelos de información, establecidos en el convenio de trabajo. •El CI debe tener la metodología para el llenado de los modelos. Establecido en la Base Metodológica. 	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1. La oficina estadística solicita al CI la entrega de la información estadística.	1.1. El responsable de estadística del CI recopila o capta los datos estadísticos correspondientes al modelo a llenar. 1.2. El responsable de estadísticas del CI llena el modelo acordado por el convenio de trabajo según la Base Metodología.	

	1.3. El responsable de estadística del CI entrega la información estadística a la oficina de estadística.
2. La oficina estadística recibe el modelo con la información estadística. 3. Se procesa la información estadística satisfactoriamente.	
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
[3]. Se devuelve el modelo con errores.	[3.1]. Se rectifican los errores. [3.2]. Ir a paso 1.3.
Poscondiciones	<ul style="list-style-type: none"> • La información queda guardada en la oficina de estadística. • Se entrega una constancia de la entrega de la información al CI.

Diagrama de Actividades CU: *Entregar Información Estadística*

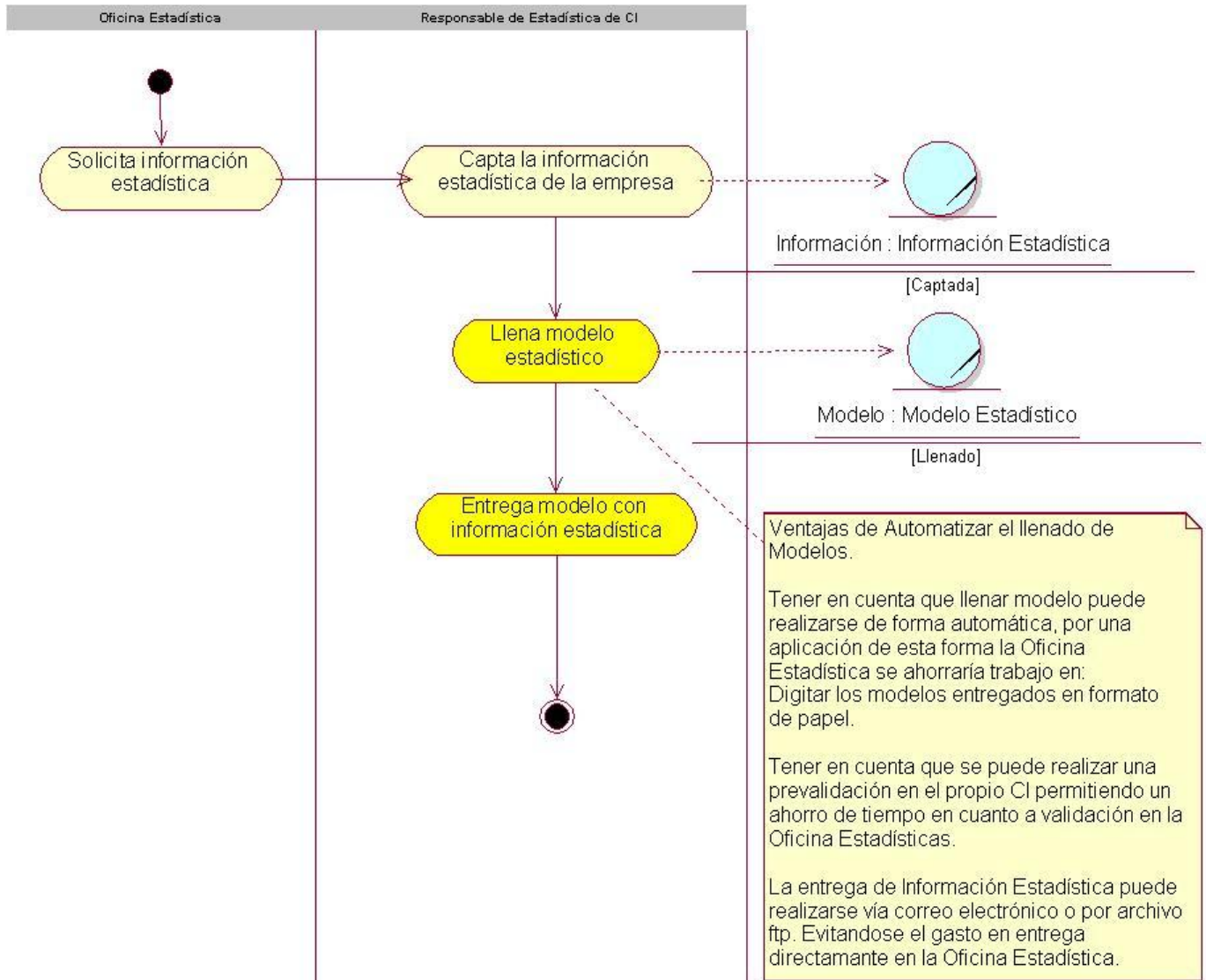


Figura5. Diagrama de Actividades del Caso de Uso: *Entregar Información Estadística*

Descripción textual del CU: *Procesar Información Estadística*

Caso de Uso	Procesar Información Estadística	
Actores	Llegada de Información Estadística (evento del negocio)	
Trabajadores	Digitador, Validador, Responsable del Departamento de Informática	
Resumen	Este caso de uso se inicia cuando llega la información estadística a la oficina y es procesada.	
Precondiciones	El modelo con la información estadística debe estar certificado por la autoridad de la empresa correspondiente al CI.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1. Llega a la Oficina Estadística (ONE, OTE, OME) la información estadística.	1.1. El digitador mediante el MSET NT digita la información recibida. 1.2. El validador revisa la información estadística digitada por el digitador. 1.3. La información está validada correctamente. 1.4. El responsable del departamento de estadística encripta y comprime la información validada. 1.5. El responsable del departamento de estadística envía el modelo estadístico al organismo inmediato superior.	
Flujo Alternativo		
Acción del Actor	Respuesta del Negocio	

	<p>[1.3]. Existen errores.</p> <p>[1.4]. El validador verifica y corrige los errores del modelo estadístico.</p> <p>[1.5]. Ir a paso 1.3.</p>
<p>Poscondiciones</p>	<p>La información queda almacenada en la oficina de estadística.</p>

Diagrama de Actividades CU: Procesar Información Estadística

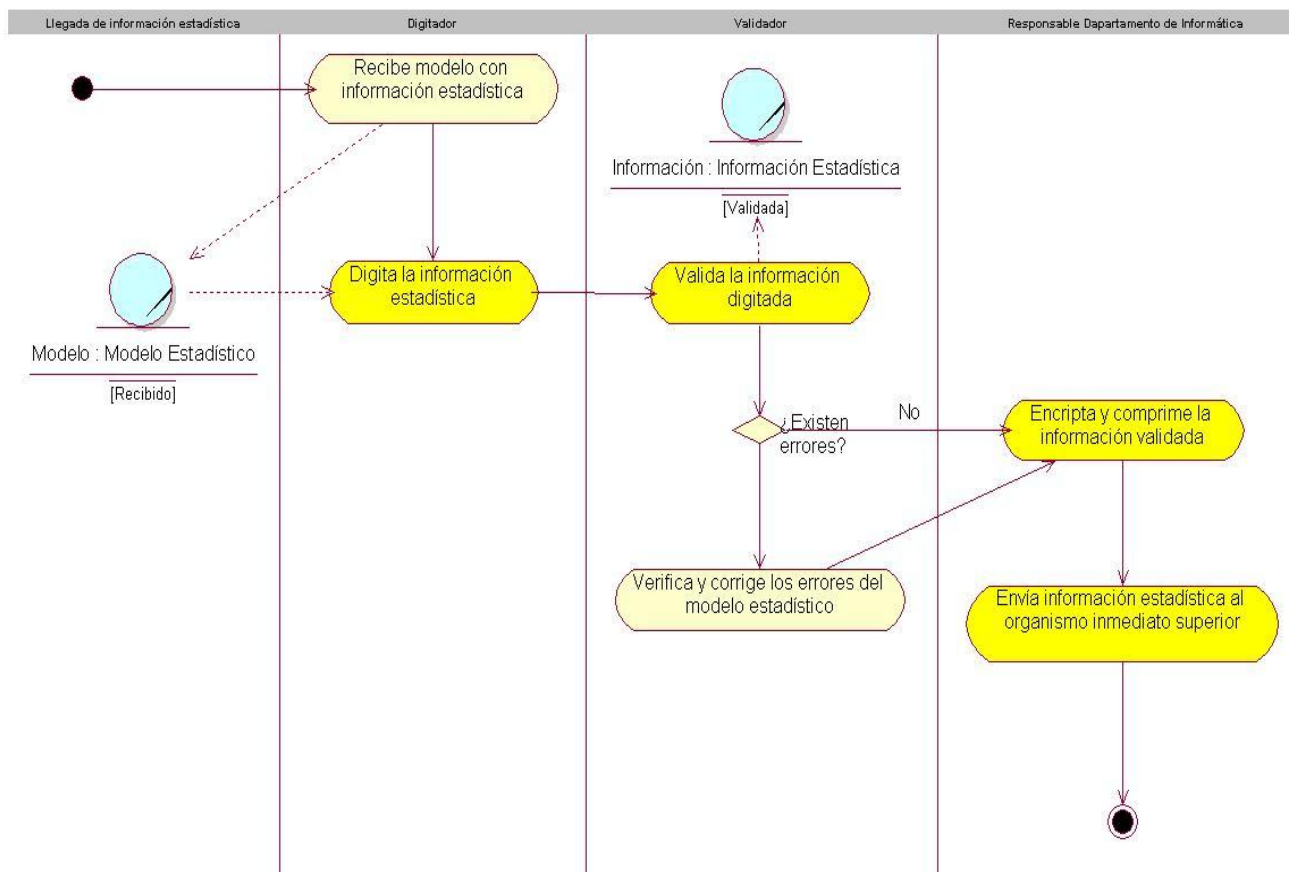


Figura6. Diagrama de Actividades del CU: Proce sar Información Estadística.

Modelo de Objetos

El modelo de objetos del negocio se describe a través de un diagrama de clases, en el cual se representa cómo colaboran los trabajadores y las entidades del negocio (objetos que los trabajadores del negocio toman, inspeccionan, manipulan, producen o utilizan durante la realización de los casos de uso de negocio), es decir, su participación dentro del negocio y la relación entre ellos.

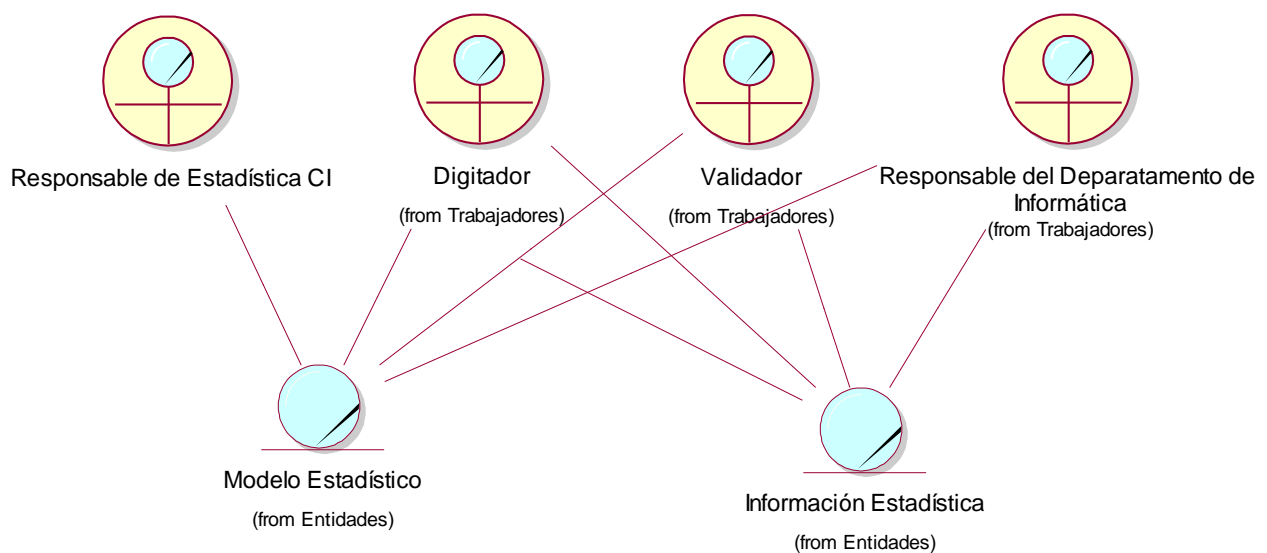


Figura7 Modelo de clase de objetos

Reglas Generales del Negocio

- EL CI debe dejar una copia de la información estadística entregada para futuras referencias en caso de que se detecten errores.
- El CI cumple lo establecido en el Convenio de Trabajo con la Oficina Estadística, y con la Base Metodológica para el llenado de los modelos estadísticos.

- El CI está obligado por el Convenio de Trabajo a la entrega de la información estadística.
- El CI tiene la responsabilidad de corregir la información estadística en caso de esta tener errores.

2.5. Estrategia para la captura de requisitos

La captura de requisitos es una parte importante en el desarrollo del software, ya que permite conocer el dominio del problema, es decir, conocer el sistema actual ya sea manual o informatizado y sus aspectos positivos y negativos, esto posibilita la comunicación con los clientes y los usuarios de manera que se puedan identificar sus necesidades y las expectativas que tienen sobre el sistema a desarrollar.

Para realizar el levantamiento de requisitos del Módulo de Entrada de Datos se utilizaron técnicas como la Tormenta de Ideas, donde el analista, en conjunto con el equipo de desarrollo y los clientes se reunió de forma informal y emitieron ideas, de esta manera se tuvieron varias vistas del problema y algunos requisitos un poco difusos.

Para continuar la identificación y captura de requisitos más detallados se prepararon Entrevistas, con esta técnica se contrastaron las opiniones de los involucrados directamente, en estas reuniones el cliente era el protagonista y el analista guiaba la conversación entorno al problema, este método generó una mayor implicación de los clientes y usuarios al proyecto.

Aplicando el Desarrollo Conjunto de Aplicaciones (JAD), se organizaron reuniones en las cuales, utilizando herramientas del Microsoft Office como el Visio 2003 y el Power Point que ayudaron a transformar conceptos técnicos y empresariales en diagramas más claros desde el punto de vista visual, se les mostró a los clientes y usuarios, de forma dinámica y en un lenguaje natural entendible por todos, un resume con ideas de la estructura del sistema a desarrollar, esta técnica permitió detallar los requisitos e identificar otros que no se tuvieron en cuenta en encuentros anteriores, llegando a un acuerdo común de las funcionalidades que debe tener el sistema, quedando un documento aprobado por los clientes como constancia. Esta técnica ahorró tiempo de trabajo en la etapa inicial.

La técnica de los Casos de Uso se utilizó tanto para la obtención como para especificación de los requisitos funcionales del sistema. Se realizó una descripción textual de los casos de uso, apoyándose de plantillas que propone RUP (anexo3), describiendo las interacciones en un lenguaje natural, fácilmente comprensibles por los clientes y usuarios.

Especificación de requisitos de software

Estos requisitos reflejan todas las funcionalidades (respecto a la Entrada de Datos Estadísticos) que brindaba el sistema en uso (MicroSet NT), extiende las mismas porque ha tomado en consideración la experiencia alcanzada a lo largo de todo este tiempo de explotación así como las recomendaciones realizadas por los clientes, se adaptan a las nuevas condiciones en cuanto a tecnología, que han de introducirse próximamente en las oficinas estadísticas a todos los niveles e incluye requisitos totalmente nuevos solicitados por los clientes a fin de satisfacer necesidades que han surgido o que no fueron consideradas en la implementación de la actual aplicación. En lo que respecta a los Centros Informantes y teniendo en cuenta la paulatina incorporación de la tecnología suficiente para realizar la digitalización de los datos estadísticos en los mismos, ahorrando trabajo a las oficinas de estadísticas, se ha tenido en cuenta la solicitud de implantar una variante de la aplicación prevista a desarrollar pero con menos funcionalidades.

Descripción general

Funciones Generales:

Para las oficinas estadísticas el módulo garantiza la entrada de los datos estadísticos y la validación de los mismos como funcionalidades fundamentales, adicionalmente se incluye la función de auditar el sistema que consiste en permitir hacer un seguimiento cronológico para cada persona de las acciones que ha realizado en el mismo, así como las funcionalidades de administración y configuración todo esto en integración a la base de datos. Se han considerado las restricciones de seguridad para los usuarios de forma que a la ampliación no es posible acceder sin el debido permiso.

Para los centros informantes, las funcionalidades de entrada de datos y la validación asociada al modelo exclusivamente. Respecto al envío de información desde los CI a la oficina estadística correspondiente, el

sistema garantiza la comunicación vía correo electrónico y la conversión de los datos digitados en un formato de archivo para su transportación en algún tipo de soporte.

Características de los Usuarios:

- **Administrador:** es el encargado de la administración general del modulo, tanto de los usuarios con acceso a este, como de mantener actualizados los modelos en uso aprobados por la ONE o la OTE y es también quien garantiza la firma digital de los modelos. Configura las diferentes funcionalidades a fin de adaptarlas de acuerdo a las características donde este instalada la aplicación (Oficina Estadística o CI).
- **Digitador:** Este actor trabaja directamente en el llenado de los modelos y guarda los mismos.
- **Validador:** Es el usuario encargado de validar los datos una ves llenada las plantillas.
- **Auditor del Sistema:** dispone de los privilegios adecuados para consultar la actividad en el sistema.

Requisitos Funcionales

Los requisitos funcionales descritos poseen un nombre genérico que los identifica, una explicación que detalla objetivamente en que consiste y el actor que el que se relaciona.

R1. Autenticar.

La autenticación tiene alta importancia para el sistema, brinda una mayor seguridad y permite que cada usuario acceda solo a la parte del sistema que lo permita su rol y da los derechos para realizar las acciones que requiera su rol. Un mismo usuario puede interpretar varios roles. Solo el administrador otorga los permisos necesarios a cada usuario.

R2. Adicionar Usuario.

El administrador es el autorizado de adicionar un usuario. Cuando el administrador adiciona un usuario nuevo el sistema debe permitir que especifique el nombre de usuario, apellidos, el nick de usuario y la contraseña.

R3. Eliminar Usuario.

El administrador es el autorizado de eliminar un usuario. Cuando el administrador elimina un usuario el sistema debe permitir que seleccione el usuario de la lista y comprobar si lo desea eliminar.

R4. Asignar rol.

El administrador asigna el rol al usuario en correspondencia al nivel de acceso que debe tener el usuario dentro del sistema. La aplicación debe dar los permisos al sistema en función del rol que desempeñe.

R5. Instalar Modelo.

Cuando el digitador desea instalar un modelo para digitarlo el sistema debe cargar el modelo de la base de datos.

R6. Eliminar Modelo.

Cuando el digitador desea eliminar un modelo seleccionado el sistema debe comprobar que se desea realizar esa operación y advertir las consecuencias de la eliminación de un modelo.

R7. Actualizar Modelo.

Al digitador solicitar actualización de un modelo el sistema debe cargar de la base de datos el modelo.

R8. Administrar Configuración de Módulo.

El sistema debe permitir la personalización del módulo, en función del entorno donde corre el mismo de forma que este varíe en dependencia de que esté instalado en un CI o en alguna Oficina Estadística.

R9. Digitar Datos Estadísticos.

El sistema debe de permitir cargar los modelos (entiéndase el formulario) desde un archivo y llenarlos según los datos, tanto desde una vista amigable diseñada para los CI, como desde una interfaz de formularios dedicada a las oficinas estadísticas y adecuada a las características del Digitador (similares a la de MicroSet). Este proceso será realizado por el Digitador.

R10. Guardar los Datos digitados en archivo.

El sistema debe permitir al digitador guardar el modelo llenado en un archivo xml.

R11. Ofrecer ayuda de acuerdo a la base metodológica.

Para el proceso de llenado de los modelos, el sistema ofrecerá la ayuda necesaria, según el modelo a digitar y el centro en el que se trabaje.

R12. Validar Modelo.

El sistema debe permitir la validación de los datos una vez realizado cada modelo, debe verificar que los datos entrados sean correctos, para ello es necesario que se haya ofrecido la ayuda correspondiente en el llenado de los modelos. Esta acción será realizada por el validador.

R13. Auditar el Sistema.

El sistema debe permitir tener un control detallado del manejo de información sobre las modificaciones realizadas. Debe brindar un reporte con la fecha, hora y modificación realizada, así como por la persona que fue realizada dicha transformación.

R14. Exportar datos del modelo al formato de MicroSet.

El sistema debe exportar datos del modelo al formato de MicroSet con el fin de lograr compatibilidad con este sistema, por lo que luego de realizado el llenado de modelos y guardados se exportarán a dicho formato, esto no se realizará en el caso de los CI.

R15. Introducir datos del Modelo en la Base de Datos una vez que estos estén validados.

Cuando se culmina con las validaciones correspondientes el sistema debe permitir que se guarden los datos del modelo (la información estadística digitada) en la base de datos.

Requisitos No Funcionales

Usabilidad

Debido a que los usuarios no son profesionales en la rama informática, debe ser un sistema con una interfaz manipulable, de fácil acceso y debe ser intuitiva a los usuarios, con el objetivo de facilitar su uso. Se debe ofrecer la interfaz necesaria en cada caso y además exportar los datos al formato con el que actualmente trabajan. Se debe ofrecer un manual de ayuda para el uso de la aplicación, donde se detallen cada aspecto sobre esta.

Confiabilidad

La información manejada por el sistema esta protegida de acceso no autorizado y divulgación. Además permite una vez que los usuarios, hagan cualquier transformación, quede registrado el cambio realizado, por quien fue hecho y a que hora. La Aplicación permite la creación de copias de seguridad a fin de mantener una Política de Seguridad respecto a la información.

Rendimiento

El sistema permite el trabajo de varios usuarios sobre mismo documento, en diferentes estaciones de trabajo, realizando las actualizaciones necesarias. Permite el envío de información mediante: correo electrónico, soporte magnético, vía ftp, o por conexión directa a la base de datos de forma rápida y eficiente.

Soporte

Se le brindara soporte y mantenimiento a la aplicación por un tiempo requerido de 6 meses, con el fin de detectar fallas en el sistema y corregir errores de funcionamiento. Además para brindar asesoría técnica con el eso del sistema

Restricciones de diseño e implementación

El sistema será una aplicación de escritorio con menús principales. Su diseño tiene como principal objetivo, ser del agrado del usuario. Además debe trabajar con teclas de fácil acceso, ente las que se encuentran: Esc., y las teclas del bloque de acceso rápido de calculadora (Num. Lock). Se desarrollará en

la plataforma Visual Studio .NET 2005 con programación orientada a objetos (POO), se usará como herramienta de modelación Rational Rose Enterprise Edition.

Software

En todas las computadoras sólo se requiere de la instalación de cualquier sistema operativo de la familia Windows NT y del Framework .NET 2.0.

Hardware

Se necesita de 128 MB de RAM como mínimo y 10 MB de espacio libre en disco duro, por ser estos los requisitos del framework. Todas las computadoras implicadas deben estar conectadas a una red.

Requerimiento de ayuda y documentación

Se ofrecerá la documentación necesaria, con la correspondientes especificaciones, además de un manual de al usuario que indique de forma detallada y clara como utilizar el software, sus facilidades y funcionalidades. Estos documentos serían:

Manual de Usuario de la Aplicación.

Ayuda Integrada a la Aplicación.

Documentos de Ingeniería de Software realizados en el proyecto.

Adquisición de Componentes

Se utilizarán componentes disponibles en el framework de .NET y otros gratis que pone a disposición Microsoft. Es importante destacar que todos estos componentes son revisados y aprobados por el equipo de Arquitectura del proyecto.

Interfaz

La interface de usuario tiene que cumplir con los requerimientos de la identidad del diseño establecidos por la Oficina Nacional de Estadística: con color predominante azul, debe tener los tamaños de letra (8 pts) y tipo de letra (Arial) de manera estándar, proveniente del IDE utilizado (Visual Studio 2005).

2.6. Modelo de Casos de Uso del Sistema

Patrones de casos de uso utilizados

Los patrones de los casos de uso son parte importante en el Proceso de Ingeniería de Software, con ellos se puede agilizar el trabajo del analista, contribuyendo a un modelado del sistema más eficiente.

En este módulo se utilizó el patrón de Múltiples Actores-Rol Común, ya que se tiene al actor “usuario del sistema” que interactúa con el caso de uso “Autenticar”, este actor, desde el punto de vista del caso de uso, es el único actor que interactúa con cada instancia de este caso de uso, sin embargo este actor es heredado por los actores: “administrador”, “digitador”, “validador” y “auditor del sistema” que juegan el mismo rol hacia el caso de uso.

Se utilizó también el patrón CRUD-Completo para los casos de uso “Administrar Usuarios” y “Administrar Modelos”, en los cuales se modelan operaciones como insertar usuario, eliminar usuario, modificar rol para el caso de los usuarios y las operaciones insertar modelo, modificar modelo y eliminar modelo para el caso de los modelos.

Actores del Sistema

El sistema cuenta con cinco actores que realizarán diferentes funciones dentro del sistema según sus privilegios: Usuario del sistema, Administrador, Digitador, Validador y el Auditor del Sistema.

Actor	Descripción
Usuario del sistema	Es la persona que accede al sistema, el que se autentica y se le da por tanto un determinado nivel de acceso según su rol. Puede ser Administrador, Digitador, Validador o Auditor del Sistema.
Administrador	Es el usuario con mayor nivel de acceso, el encargado de administrar o controlar los usuarios que accedan al sistema y los roles que desempeñan. Se encarga

	además de controlar lo referente a los modelos, la inserción, eliminación y actualización de los mismos. También es el responsable de la configuración del modulo, en dependencia del centro donde esté instalado el sistema (ONE, OTE, OME, CI).
Digitador	Este usuario tiene la tarea de digitar la información estadística. El debe cargar los modelos y llenarlos; y luego guardarlos listos para validar.
Validador	El Validador es quien se encarga de verificar que los datos entrados por el digitador en los modelos estén correctos.
Auditor del Sistema	Este actor es quien se encarga de que exista constancia de los cambios. El auditor solicita al sistema un reporte detallado con fecha, hora y persona que realizó cada evento en la aplicación.

Diagrama de Casos de Uso del Sistema

A través del diagrama de los casos de uso del sistema se representa gráficamente la interacción entre los procesos (estos proceso asocian las funcionalidades que debe tener el sistema) y los actores del sistemas (terceros que va a interactuar con la aplicación). Con este diagrama los clientes y los desarrolladores deben entender el sistema.

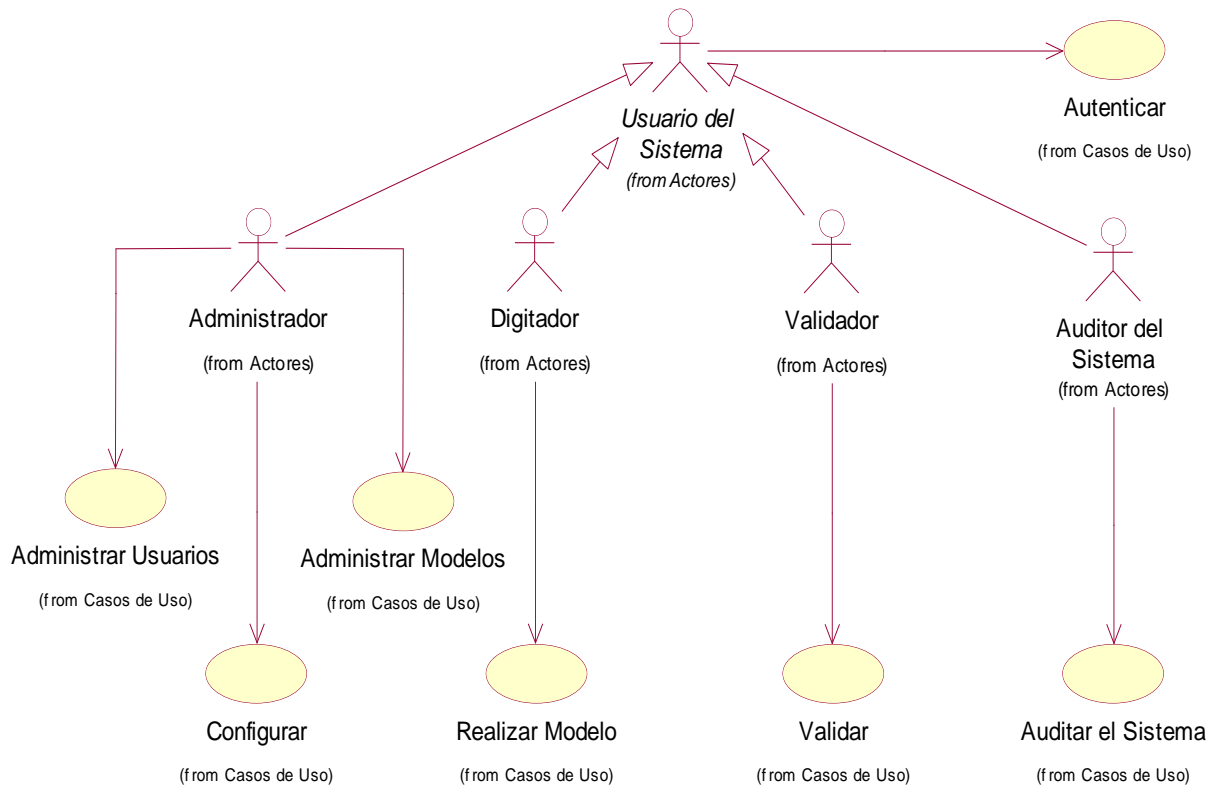


Figura8. Diagrama de Casos de Uso del Sistema.


Especificación de los Casos de Uso del Sistema



La descripción textual de forma extendida se realiza con el objetivo de entender más claramente la funcionalidad asociada a cada caso de uso del sistema. A esta descripción se le añade el prototipo no funcional por cada caso de uso, este prototipo muestra, a través de diseños, el comportamiento del sistema de forma estática, que se presenta al usuario para verificar los requerimientos funcionales.

Autenticar

Caso de Uso	Autenticar.
--------------------	-------------

Actor	Usuario del Sistema.	
Resumen	El caso de uso inicia cuando el usuario del sistema se autentica, el sistema verifica si existe ese usuario y si la contraseña es correcta, el sistema da permiso de acceso.	
Precondiciones		
Referencias	R1.	
Prioridad	Crítico.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El usuario introduce los datos Usuario y Contraseña.	1.1. El sistema comprueba que el usuario exista en la base de datos. 1.2. Si lo encuentra verifica que su contraseña es correcta. 1.3. Si es correcta la contraseña, sistema le da permiso de acceso al usuario.	
Prototipo de Interfaz		

	
Flujo Alterno	
Acción del Actor	Respuesta del Sistema
	[1.2.] Si no lo encuentra el sistema muestra un mensaje "Usted no está registrado en el sistema".
	[1.3.] Si la contraseña no es correcta, el sistema muestra un mensaje "Clave incorrecta"
Prototipo de Interfaz	

	
	
Poscondiciones	

Administrar Usuarios

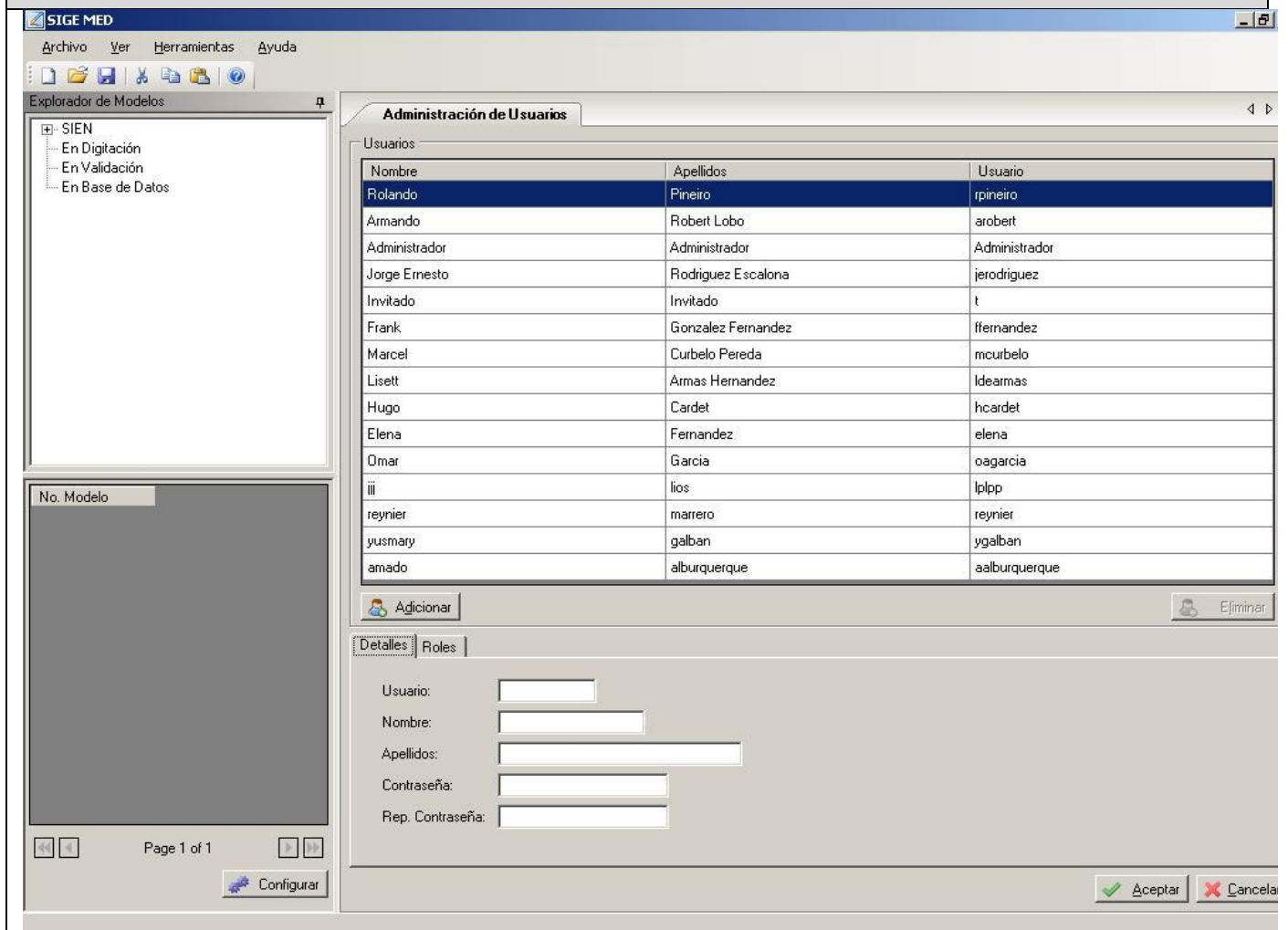
Caso de Uso	Administrar Usuarios.
Actor	Administrador.
Resumen	El caso de uso se inicia cuando el administrador accede al sistema y adiciona usuarios al sistema, elimina un usuario, o le asigna un rol a algún usuario.
Precondiciones	El administrador se debe autenticar para realizar estas acciones.
Referencias	R2, R3 y R4.
Prioridad	Crítico.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El administrador selecciona la opción que desea ejecutar.	1.1. a) Si desea insertar un nuevo usuario al sistema ir a Sección "Insertar Usuario" b) Si desea modificar el rol de un usuario ir a Sección "Modificar Rol" c) Si desea eliminar un determinado usuario ir a Sección "Eliminar usuario"
Sección "Insertar Usuario"	
Acción del Actor	Respuesta del Sistema

2. El administrador introduce los datos del nuevo usuario.	2.1. El sistema registra los datos del usuario en la base de datos.
	2.2. El sistema muestra el listado de los usuarios previamente insertados más los del nuevo ingreso.
Sección “Modificar Rol”	
Acción del Actor	Respuesta del Sistema
3. El administrador indica el usuario a modificar su rol.	3.1. El sistema busca al usuario y muestra una interfaz al administrador con los datos del mismo.
4. Asigna el rol	4.1. Si lo encuentra el sistema modifica su rol en la base de datos.
	4.2. Muestra el listado de usuarios actualizado.
Flujos Alternos	
	[4.1.] Si no lo encontró muestra un mensaje que indica que el usuario no se encuentra en la base de datos.
Sección “Eliminar Usuario”	
Acción del Actor	Respuesta del Sistema
5. El administrador indica el usuario que desea eliminar.	5.1. El sistema busca el usuario en la base de datos.
	5.2. Si lo encuentra lo elimina de la base de datos.
	5.3. Muestra el listado actualizado de usuarios.

Flujos Alternos

[5.2.] Si no lo encontró muestra un mensaje que indica que el usuario no se encuentra en la Base de Datos.

Prototipo de Interfaz



Poscondiciones

Administrar Modelos

Caso de Uso	Administrar Modelos.	
Actor	Administrador.	
Resumen	El caso de uso se inicia cuando el administrador accede al sistema e inserta, elimina o actualiza modelos estadísticos.	
Precondiciones	El administrador se debe autenticar para realizar estas acciones.	
Referencias	R5, R6 y R7.	
Prioridad	Crítico.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El administrador selecciona la opción que desea ejecutar.	1.1. a) Si desea insertar un nuevo modelo al sistema ir a Sección "Insertar Modelo" b) Si desea modificar el modelo ir a Sección "Actualizar Modelo" c) Si desea eliminar un determinado modelo ir a Sección "Eliminar modelo"	
Sección "Insertar Modelo"		
Acción del Actor	Respuesta del Sistema	

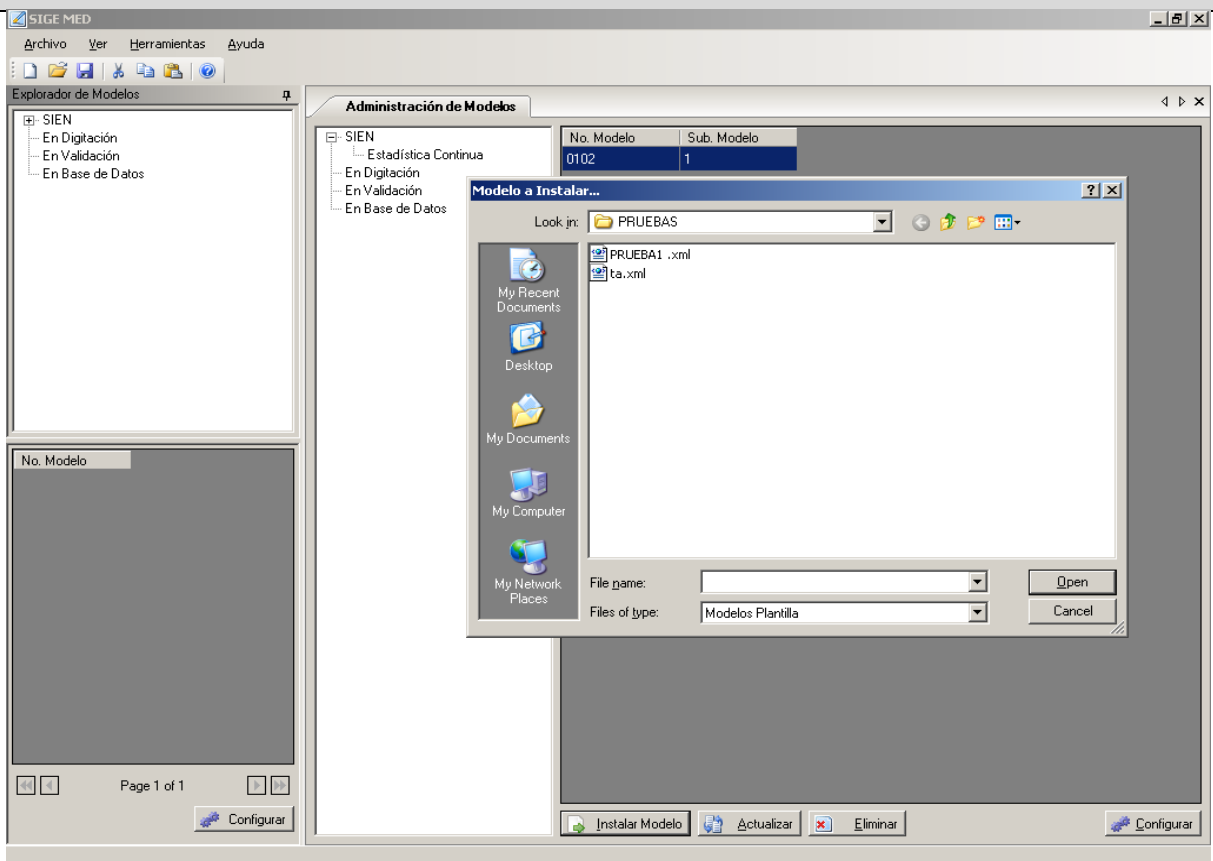
2. El administrador introduce el nuevo modelo.	2.1. El sistema registra el modelo en la base de datos.
	2.2. El sistema muestra el listado de los modelos previamente insertados más los del nuevo ingreso.
Sección “Actualizar Modelo”	
Acción del Actor	Respuesta del Sistema
3. El administrador indica el modelo a actualizar.	3.1. El sistema busca el modelo en la base de datos.
	3.2. Si lo encuentra el sistema muestra el modelo.
4. El administrador actualiza la información en el modelo.	4.1. Actualiza el modelo en la base de datos.
	4.2. Muestra el modelo actualizado.
Flujo Alternativo	
	[3.2.] Si no lo encontró muestra un mensaje que indica que el modelo no se encuentra en la base de datos.
Sección “Eliminar Modelo”	
Acción del Actor	Respuesta del Sistema
5. El administrador indica el modelo que desea eliminar.	5.1. El sistema busca el modelo en la base de datos.

	5.2. Si lo encuentra lo elimina de la base de datos.
	5.3. Muestra el listado actualizado de los modelos.

Flujo Alternativo

	[5.2.] Si no lo encontró muestra un mensaje que indica que el modelo no se encuentra en la base de datos.
--	-----------------------------------------------------------------------------------------------------------

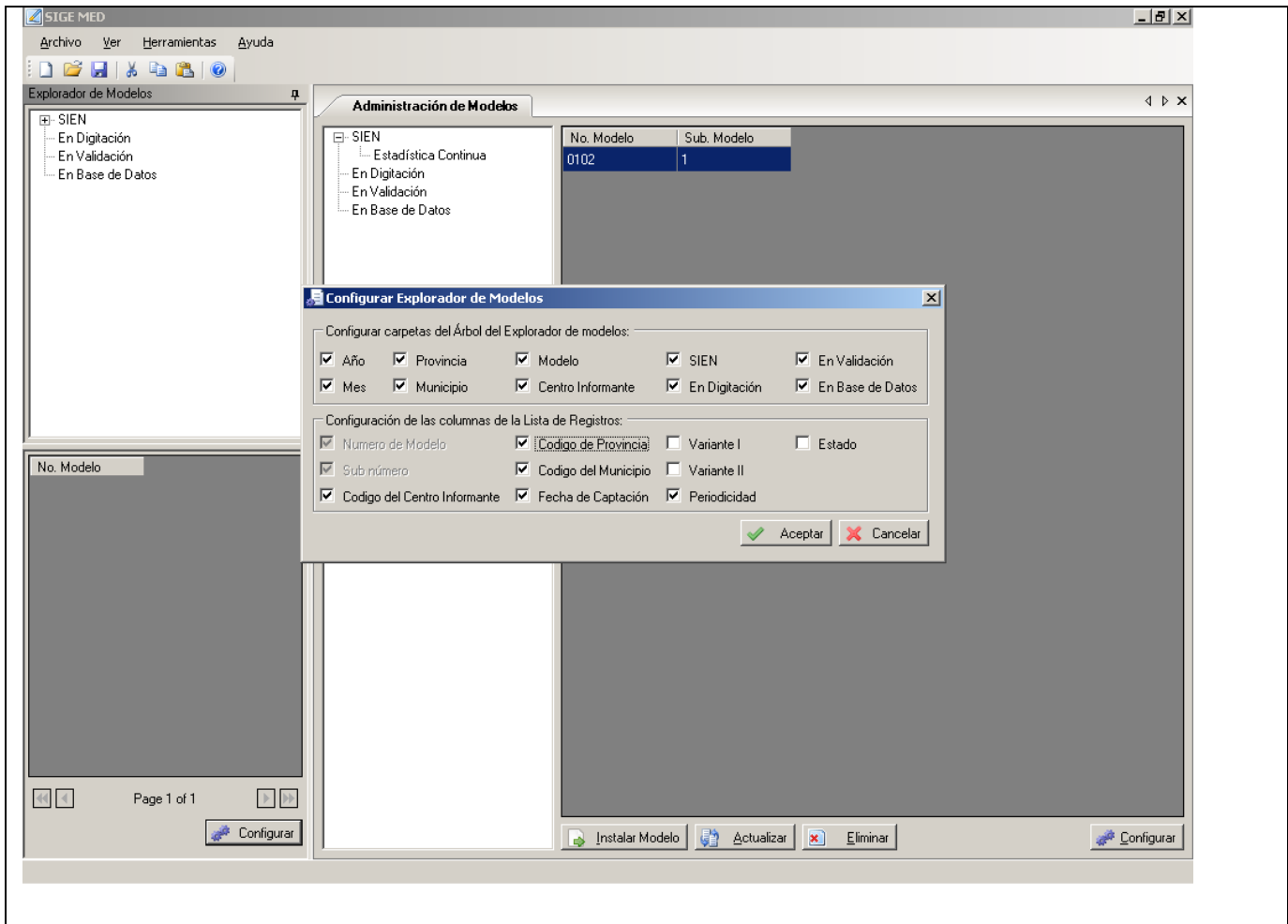
Prototipo de Interfaz



Poscondiciones

Configurar

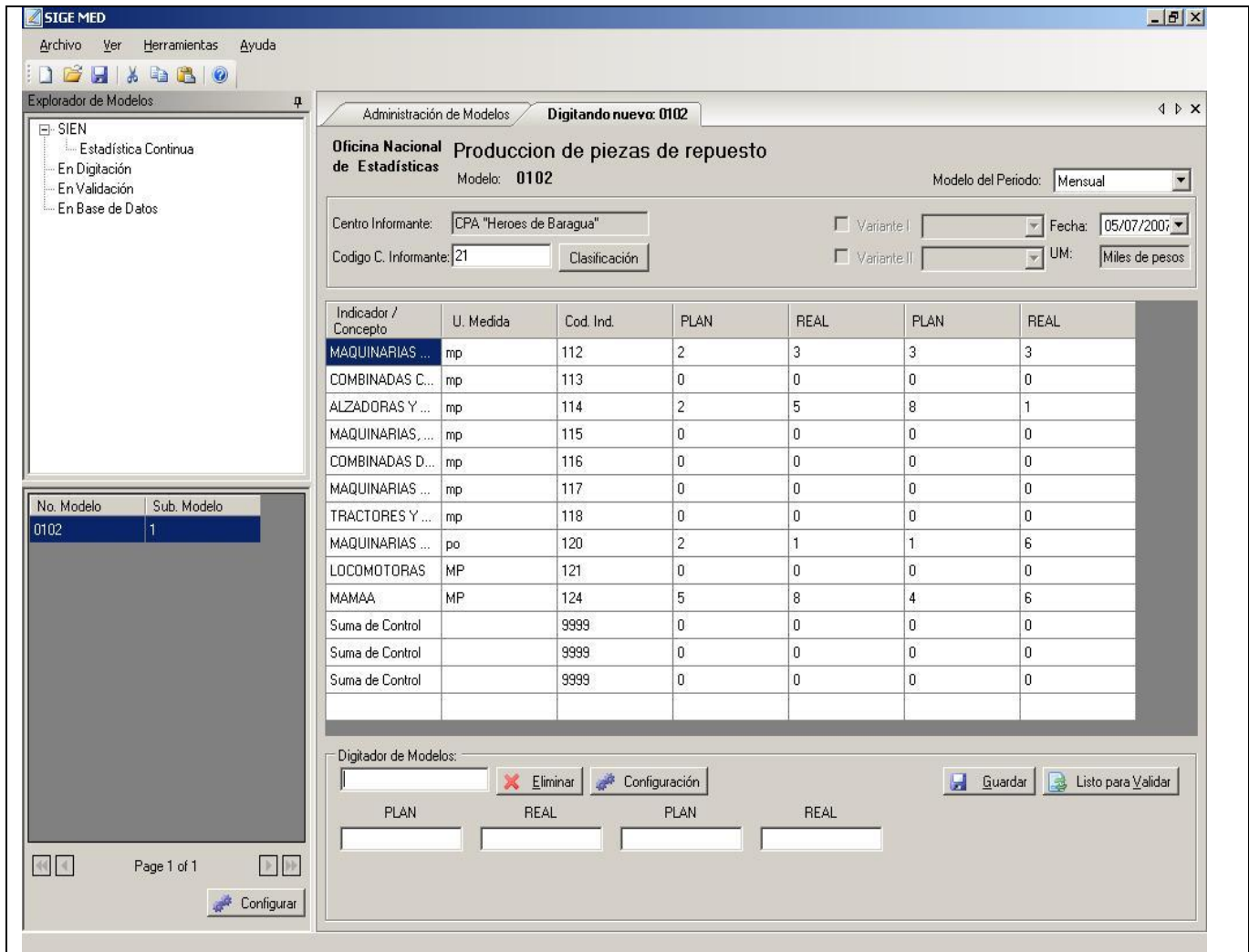
Caso de Uso	Configurar.	
Actor	Administrador.	
Resumen	El caso de uso inicia cuando el administrador accede al sistema y configura el módulo según la entidad donde esté instalada la aplicación (ONE, OTE, OME, CI).	
Precondiciones	El administrador se debe autenticar para realizar estas acciones.	
Referencias	R8.	
Prioridad	Secundario.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El administrador solicita la configuración del módulo.	1.1. El sistema muestra las opciones de configuración requeridas.	
2. El usuario selecciona la configuración a implantar.	2.1. El sistema aplica la configuración asignada.	
Prototipo de Interfaz		



Realizar Modelo

Caso de Uso	Realizar Modelos
Actor	Digitador
Resumen	Este caso de uso se inicia cuando el digitador selecciona un modelo a realizar y abarca todo el llenado de este con información estadística.
Precondiciones	El digitador se debe autenticar para realizar esta acción.

Referencias	R9, R10, R11, R14 y R15.	
Prioridad	Crítico.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El digitador selecciona el modelo a realizar.	1.1. El sistema abre el archivo del modelo seleccionado. 1.2. Muestra el formulario.	
2. El digitador digita los datos estadísticos.	2.1 Guarda los datos estadísticos introducidos por el digitador. 2.2 Termina con la entrada de datos.	
3. Activa el estado listo para validar.		
Prototipo de Interfaz		

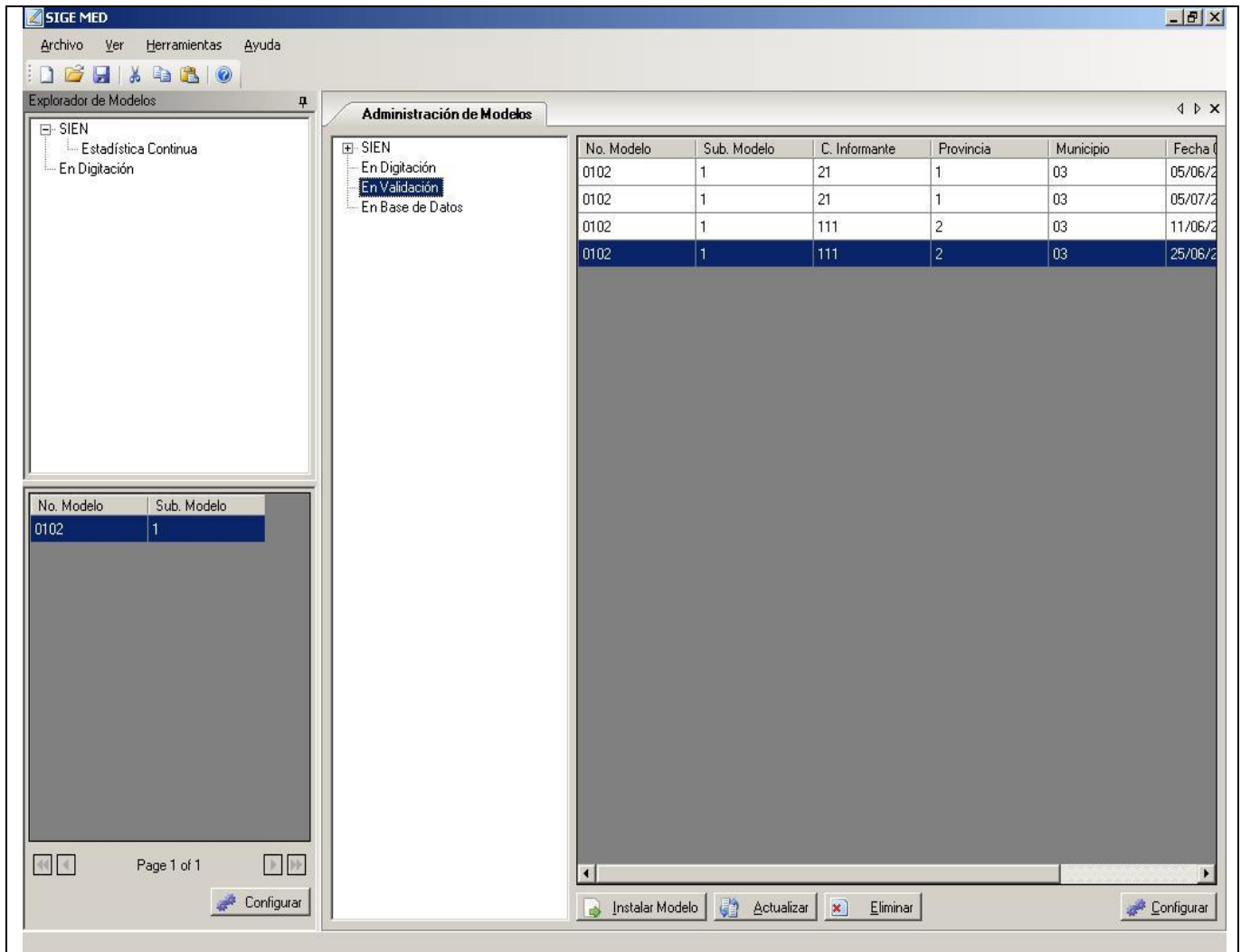


Flujo Alterno

Acción del Actor	Respuesta del Sistema
	[2.2.] Si no termina con la entrada de datos da la opción de guardar modelo.

Validar

Caso de Uso	Validar	
Actor	Validador	
Resumen	El caso de uso inicia cuando el validador accede a los modelos que se encuentran listos para validar y realiza las acciones de validación correspondientes.	
Precondiciones	El validador se debe autenticar para acceder a la validación de los modelos.	
Referencias	R12.	
Prioridad	Secundario.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El validador solicita un modelo a validar.	1.1. El sistema muestra el modelo con los datos correspondientes.	
2- El validador realiza la revisión.	2.1. El sistema muestra y corrige los errores. 2.2 Guarda los cambios	
Prototipo de Interfaz		

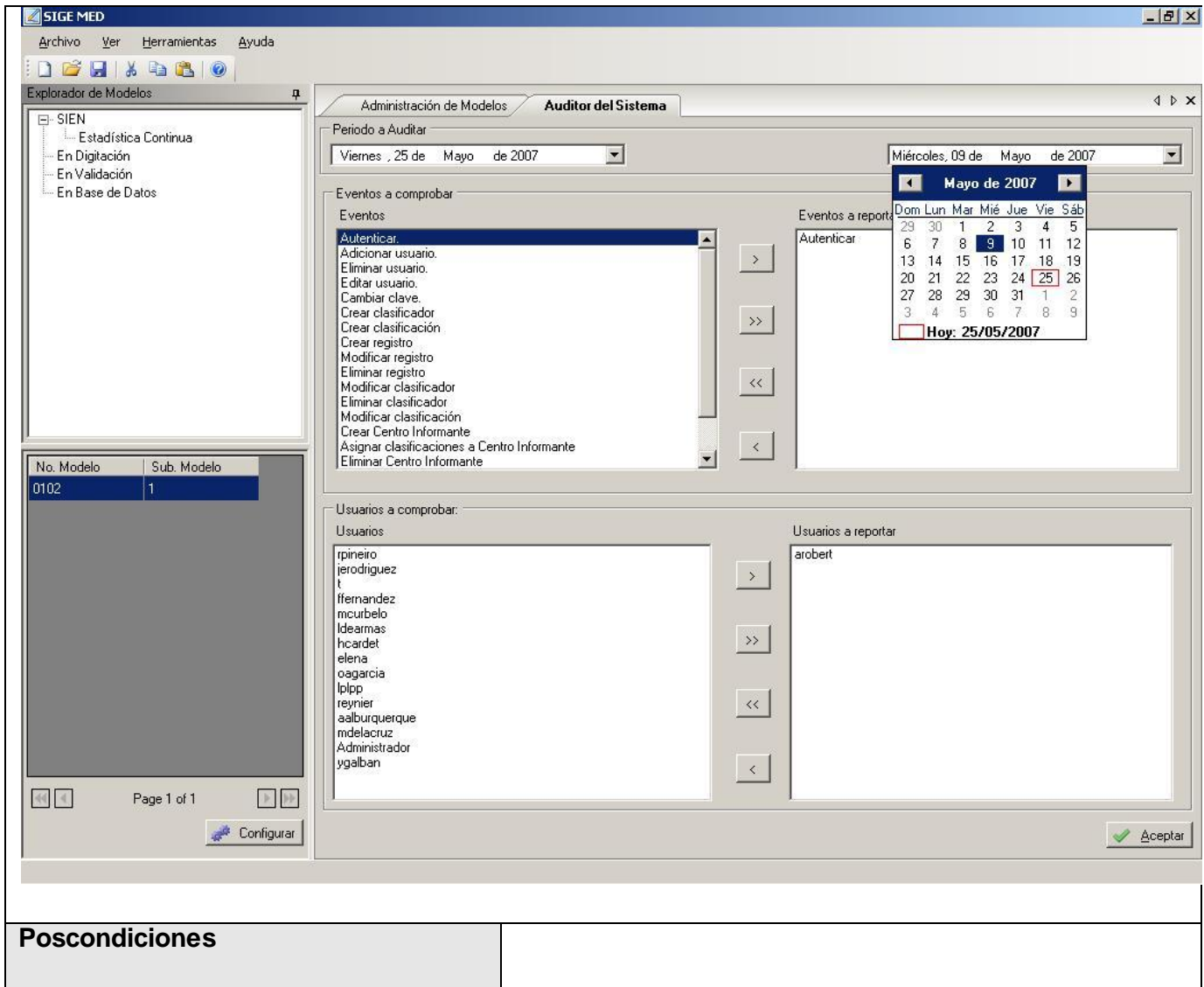


Flujo Alterno

Flujo Alterno	
Acción del Actor	Respuesta del Sistema
	[2.1.] El modelo no presenta errores.
Poscondiciones	

Auditar el Sistema

Caso de Uso	Auditar el Sistema	
Actor	Auditor	
Resumen	El caso de uso inicia cuando el auditor accede al sistema y solicita un reporte de los eventos ocurridos en el sistema, en el período que ocurrieron y los usuarios que los ejecutaron.	
Precondiciones	El auditor debe autenticarse para solicitar un reporte.	
Referencias	R13.	
Prioridad	Secundario.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El auditor solicita un reporte.	1.1. El sistema muestra una interfaz de búsqueda avanzada.	
2. El auditor selecciona la información que necesita.	2.1. Genera el reporte y lo muestra al auditor.	
Prototipo Interfaz		



2.7. Conclusiones

En este capítulo se presentó el desarrollo de los distintos artefactos generados por el analista mostrando la estructura del negocio en términos del diagrama de casos de uso del negocio y el modelo de objetos del negocio, la funcionalidad a partir del diagrama de casos de uso del sistema y el comportamiento en los diagramas de interacción, es decir, los diagramas de actividades.

Capítulo 3

Análisis de los Resultados Obtenidos

3.1. Introducción

Según Pressman (Pressman, 2005), las métricas del software se refieren a un amplio elenco de mediciones para el software de computadora. La medición se puede aplicar en el proyecto del software con el intento de mejorarlo sobre una base continua. (...) Finalmente el ingeniero de software puede utilizar la medición para ayudar a evaluar la calidad de los resultados de trabajos técnicos y para ayudar en la toma de decisiones táctica a medida que el proyecto evoluciona.

Existen cuatro razones para medir los procesos del software, los productos y los recursos (Pressman, 2005):

- Caracterizar
- Evaluar
- Predecir
- Mejorar

Las métricas no son absolutas, sin embargo ayudan en gran medida al ingeniero de software a descubrir a tiempo muchos problemas y corregirlos antes de que ocurra un desastre mayor. Ayudan a realizar un análisis de los resultados obtenidos en determinada etapa del proceso del software, evaluando los artefactos desarrollados, midiendo el nivel de calidad y veracidad de los modelos y los requisitos capturados, punto principal para que el producto software cumpla con las funcionalidades que necesita el cliente y que tenga la calidad requerida.

En este capítulo se hace un análisis de los resultados obtenidos, valorando así el trabajo realizado por el analista en el proceso de desarrollo del Módulo de Entrada de Datos, para esto se aplicaron métricas a los artefactos generados por el analista que dan una medida tanto cuantitativa como cualitativa del grado de calidad.

3.2. Modelo del Negocio

Más Opciones del Menú que Casos de Uso del Negocio

Con el modelado del negocio se logró tener una visión de la ONE, comprendiendo su estructura y dinámica, entendiendo el problema actual e identificando las mejoras potenciales presentes en el proceso de captación de información estadística.

En el modelo del negocio el principal artefacto que se generó fue el diagrama de casos de uso del negocio, en él se representaron los procesos fundamentales en la captación de información (representados por los casos de uso) y los actores del negocio involucrados en este proceso.

En la realización de los casos de uso se hace una descripción más detallada de los procesos, sacando de ahí las posibles actividades a automatizar, de donde se derivan las funcionalidades que la aplicación debe soportar, estas funcionalidades están representadas en el prototipo no funcional básicamente en la Opciones del Menú.

$$\#Opciones\ del\ Menú / \#Casos\ de\ Uso \geq 1$$

$$5/2 > 1$$

$$2.5 > 1$$

Esta métrica garantiza que el Modelo de Casos de Uso del Negocio sea los más óptimo posible, o sea, que describa lo que pasa en el negocio, pero los procesos más generales, es decir, que no se representen procesos innecesarios que no son relevantes o que no brindan un resultado observable. Es por ello que debe haber menos o igual cantidad de Casos de Usos del Negocio que Opciones del Menú, lo que queda garantizado en la fórmula anterior.

Cubrimiento de los Requisitos del Sistema por Caso de Uso del Negocio

Cada caso de uso del negocio describe un proceso del negocio y a la vez agrupan un conjunto de actividades que se convierten en funcionalidades del sistema, por lo que es lógico validar que cada uno de los requisitos del sistema se encuentre enmarcado en alguno de los casos de uso del negocio.

$$\# \text{Requisitos} \geq 2 * \# \text{CUN}$$

$$25 \geq 2 * 2$$

$$25 > 4$$

De esta manera se garantizó que los requisitos del sistema no se fueran del margen del negocio (proceso de captación de información estadística) para el que se va a desarrollar el módulo.

3.3. Especificación de Requisitos

Todos los requisitos deben estar contenidos en los CUS

Cada CUS tiene asociada una funcionalidad del sistema, los requisitos funcionales representan las capacidades o condiciones que el sistema debe cumplir (funcionalidad), por lo que todos estos requisitos deben ser mostrados en al menos un caso de uso del sistema.

En la descripción de los CUS se referencia los requisitos que se encuentran asociados a cada CUS, de manera que los 15 requisitos funcionales identificados están recogidos en estos CUS.

Compleción de los requisitos

Según Pressman (Pressman, 2005), la completión (característica que mide hasta que grado está completa la especificación de requisitos) puede emplearse para medir la calidad de la especificación de requisitos, y aunque se puede decir que es una característica cualitativa se puede calcular evaluando el grado de validación de los requisitos a través de la siguiente relación:

$$Q_3 = n_c / (n_c + n_{nv}); \text{ donde } \begin{cases} n_c: \text{ número de requisitos que se han validado como correctos} \\ n_{nv}: \text{ número de requisitos que no se han validado todavía} \end{cases}$$

$$Q_3 = 22 / (22+3)$$

$$Q_3 = 22/25$$

$$Q_3 = 0.88$$

Mediante el cálculo de esta relación se puede decir que un 88 % de los requisitos han sido validados correctamente. Con este resultado comprueba que se cumplió a un alto grado con los requisitos especificados, logrando que los usuarios queden altamente satisfechos.

3.4. Diagrama de Casos de Uso del Sistema

Para evaluar la calidad de la funcionalidad a partir del diagrama de casos de uso del sistema se aplicó un modelo de métricas Orientado a Objetos (OO) de la universidad EAFIT de Medellín en Colombia. Este modelo tiene en cuenta cuatro atributos fundamentales para medir la calidad: consistencia, correctitud, completitud y complejidad.

Completitud: Grado en que se ha logrado detallar todos los casos de uso relevantes.

Consistencia: Grado en que los casos de uso del sistema describen las interacciones adecuadas entre el usuario y el sistema

Correctitud: Grado en que las interacciones actor / sistema soportan adecuadamente el proceso del negocio

Complejidad: Grado de claridad en la presentación de los elementos que describen el contexto y funcionalidad del sistema.

Para cada atributo se analiza un conjunto de factores, cada uno de los cuales está asociado a una métrica (ver tabla1).

Atributos	Factores
Compleitud	<ul style="list-style-type: none"> • ¿Se presenta una descripción detallada (descripción extendida esencial) de todos los casos de uso? • ¿Están definidos todos los requisitos que justifican la funcionalidad del caso de uso? • ¿Existen requisitos que no han sido considerados en algún caso de uso? • ¿Están todas las acciones del flujo de eventos redactadas en función del responsable? • ¿Se describen las condiciones de excepción que debe contemplar cada flujo de eventos?
Consistencia	<ul style="list-style-type: none"> • ¿El nombre dado a los casos de uso es una expresión verbal que describe alguna funcionalidad relevante en el contexto del usuario? • ¿Representa el caso de uso una interacción observable por un actor? • ¿Existe una adecuada separación entre el flujo básico de eventos y los flujos alternos y/o flujos subordinados?
Correctitud	<ul style="list-style-type: none"> • ¿Existe para cada caso de uso por lo menos un usuario responsable? • ¿Las interacciones definidas describen la funcionalidad requerida del sistema? • ¿Representa el caso de uso requisitos comprensibles por el usuario? • ¿Las interacciones definidas introducen mejoras al proceso actual?
Complejidad	<ul style="list-style-type: none"> • ¿Los elementos dentro del diagrama de CUS

	están adecuadamente ubicados de manera que facilitan su interpretación?
--	-------------------------------------------------------------------------

Tabla1 Atributos y factores a sociados

Estos factores fueron analizados por ingenieros especialistas en le tema y los usuarios, los que calificaron la elaboración del diagrama y la descripción detalla de los casos de uso del sistema de alta calidad.

Aplicando las métricas asociadas a cada factor para evaluar los atributos correspondientes (ver anexo 3) se constató que se presenta una descripción detallada de todos los casos de uso y que todos los requisitos funcionales están considerados en algún caso de uso, logrando así la completitud al detallar todos los casos de uso del sistema relevantes. Todos los nombres de los casos de uso expresan alguna funcionalidad relevante del sistema y todas las descripciones detalladas presentan una adecuada separación entre el flujo básico de eventos y los flujos alternos, garantizando una alta consistencia al describir, todos los casos de uso, interacciones adecuadas entre el usuario y el sistema. Para cada caso de uso existe al menos un usuario responsable y las interacciones definidas describen la funcionalidad requerida del sistema, verificando así la correctitud del diagrama. Los elementos dentro del sistema de casos de uso del sistema están adecuadamente ubicados, garantizando un alto grado de claridad (complejidad), de manera que facilitan su interpretación.

3.5. Prototipo no funcional (Interfaz)

Hoy día con el auge del software en el mundo existen normas para la realización de prototipos no funcionales. Estas normas garantizan que la parte del diseño se realice con buen gusto y además que sea de agrado a la vista del usuario. Es por ello que en la realización del prototipo del Módulo de Entrada de Datos además de atender las peticiones del cliente como por ejemplo que brinde la posibilidad de manejar las opciones del menú por el teclado, teniendo en cuenta que la mayoría de los usuarios que van a utilizar la aplicación manejan muchos datos, cumple con las siguientes características.

No muy cargadas

Los iconos utilizados son de extensión png y jpg, entre ellos se destacan los utilizados universalmente en las aplicaciones profesionales como son: Nuevo, Abrir, Guardar, Eliminar, etc.

La imagen que aparece en el fondo posee el nombre de la aplicación, vinculado con el color azul y blanco (colores predominantes en la aplicación además del gris). Tratando siempre de no cargar las interfaces con muchas opciones.

Patrón único por interfaz

Para que haya uniformidad se decidió que cada uno de los prototipos diseñados tenga una apariencia única y uniforme.

Tamaño de letra

De forma general se puede plantear que todos los controles de usuario e interfaces gráficas poseen los tamaños de letra (8 pts) y tipo de letra (Arial) de manera estándar, proveniente del IDE utilizado (Visual Studio 2005).

Colores adecuados

El color prevaleciente en el menú es el estándar de Windows (azul y blanco).

3.6. Conclusiones

Con la aplicación de métricas a los distintos artefactos desarrollados se midió la calidad de los mismos, demostrando que se realizó una especificación de requisitos adecuada, logrando la representación correcta de la estructura y la dinámica del negocio, así como también garantizando la funcionalidad del sistema, cumpliendo con lo especificado por los clientes.

Conclusiones Generales

Al terminar este trabajo se puede llegar a las conclusiones siguientes:

- La recopilación de información necesaria acerca del funcionamiento de las Oficinas Estadísticas en el país a través de los continuos encuentros con los clientes y el estudio profundo del Sistema de Gestión de Información (MicroSet NT) utilizado por estas oficinas permitió la modelación del negocio de una manera aceptable para la comprensión y entendimiento de los desarrolladores.
- El estudio de las técnicas y métodos utilizados para la captura de requisitos posibilitó que se realizara una elicitación de requisitos satisfactoria, eso se comprueba con el alto grado de requisitos validados correctamente por el cliente a través del sistema ya implementado.
- El estudio de patrones para los casos de uso y los actores agilizó la modelación del sistema y posibilitó la ubicación adecuada de los elementos del diagrama de casos de uso del sistema de manera que los desarrolladores interpretaran el sistema tal y como el cliente lo desea.
- Mediante la aplicación de métricas que miden la calidad de los artefactos desarrollados y evalúan los resultados obtenidos por el analista, se constató el alto nivel de organización y aceptación de trabajo realizado que contribuyó a la implementación de un producto software con calidad y con los requerimientos pedidos por el cliente.

Recomendaciones

En el transcurso de la elaboración de este trabajo han surgido nuevas ideas producto de la experiencia adquirida en el proyecto y recomendaciones por parte del cliente:

- Incorporar el procesamiento de la validación de los modelos mediante los cuadros. Este proceso se hace actualmente con el MicroSet. Con esta funcionalidad se lograría la gestión de los datos informáticos más confiables en el SIGE.
- Automatizar el trabajo con los modelos de encuestas. Con el auge que ha existido en las estadísticas el proceso de captación de información ha evolucionado, apareciendo los modelos de encuestas que es muy utilizado en el país en estos momentos. Este trabajo se hace en estos momentos manualmente ya que el MicroSet solo procesa modelos compuestos por filas y columnas.
- Exportar los datos del modelo al formato MicroSet. Es necesario automatizar este requerimiento con el fin de lograr compatibilidad con este sistema.
- Introducir los datos del modelo estadístico en la base de datos una vez que estén validados. Hasta el momento se guarda el modelo digitado en un archivo xml.

Bibliografía

Beck, K. 1999. Extreme Programming Explained. Embrace Change. s.l.: Pearson Education, 1999.

Chávez Gaona, Víctor Manuel and Olivares Rojas, Juan Carlos. 1997. Monografias.com. Monografias.com. [Online] Lucas Morea / Sinexi S.A., 1997. [Cited: diciembre 6, 2006.] <http://Monografias.com>.

dlsi. dlsi. [Online] [Cited: marzo 20, 2007.] <http://www.dlsi.ua.es/webe02/articulos/5.pdf>.

Escalona, M.J. Koch, N. 2002. Ingeniería de Requisitos en Aplicaciones para la Web:Un estudio comparativo. Sevilla: s.n., 2002.

Escobar, Yanvary, et al. monografias.com. monografias.com. [Online] [Cited: noviembre 28, 2006.] <http://monografias.com/Desarrollo del software>.

Espinoza, Javier Alberto Moya. 2004. ilustrados.com. ilustrados.com. [Online] enero 31, 2004. [Cited: diciembre 6, 2006.] <http://ilustrados.com>.

flazx. flazx. [Online] IPS, Inc. [Cited: marzo 25, 2007.] <http://www.flazx.com>.

Hernández León, Rolando Alfredo and Coello González, Sayda. 2002. EL PARADIGMA CUANTITATIVO DE LA INVESTIGACIÓN CIENTIFICA. Ciudad de la Habana: EDUNIV, 2002.

Ledesma, Rubén. 2004. SIPIE. SIPIE. [Online] 2004. [Cited: febrero 6, 2007.] <http://www.sipie.net>.

Ltd., Sparx Systems Pty. Sparx Systems Enterprise Architect User Guide. Australia: s.n.

Martínez, Andrés G. 2005. spssparatodos. spssparatodos. [Online] 2005. [Cited: diciembre 5, 2006.] <http://www.spssparatodos.com>.

Metodologías Ágiles en el Desarrollo de Software. Letelier Torres, Patricio and Sánchez López, Emilio A. 2003. Alicante: ISSI (Ingeniería del Software y Sistemas de Información), 2003.

Molpeceres, Alberto. 2002. javaHispano. javaHispano. [Online] diciembre 15, 2002. [Cited: diciembre 5, 2006.] <http://www.javaHispano.org>.

O'Decherty, Mike. 2005. Object-Oriented. Analysis & Desing. England : s.n., 2005.

Oficina Nacional de Estadísticas, Departamento de Informática Delegación Territorial. 1997. Manual de usuario de MicroSet NT versión 2.0. Camaguey : s.n., 1997.

Övergaard, Gunnar and Palmkvist, Karin. 2004. Use Cases Patterns and Blueprints. s.l. : Addison Wesley Professional, 2004. 0-13-145134-0.

P., José Luis Torres. IEEE. IEEE. [Online] [Cited: abril 24, 2007.] <http://www.ewh.ieee.org/r9/guadalajara/boletin/sep01/requerimientos.htm>.

Pérez, Pedro Yobanis Piñero. 2005. Un modelo para el aprendizaje y la clasificación. 2005.

Pressman, Roger S. 2005. Ingeniería del Software. Un enfoque práctico. La Habana: Félix Varela, 2005.

pst. pst. [Online] [Cited: marzo 20, 2007.] <http://www.pst.informatik.uni-muenchen.de/personen/kochn/ideas03-escalona-koch.pdf>.

Roles en el desarrollo de software (Taller de Ingeniería de Software) . Padilla, David Fuller. 2003. 2003.

Rumbaugh, J, Jacobson, I. and Booch, G. 2000. El Lenguaje Unificado de Modelado. s.l. : Addison Wesley, 2000.

Rumbaugh, James, et al. 1996. Modelado y diseño orientados a objetos: Metodología OMT. s.l. : Prentice Hall, 1996.

Sanchez, María A. Mendoza. 2004. informatizate. informatizate. [Online] Copyright, Junio 7, 2004. [Cited: diciembre 6, 2006.] <http://www.informatizate.net>.

sas. 2007. sas. sas. [Online] 2007. <http://www.sas.com/technologies/analytics/statistics/>.

scholar. 2007. scholar. scholar. [Online] 2007. [Cited: marzo 20, 2007.] <http://scholar.google.com/scholar?q=tecnicas+de+captura+de+requisitos&hl=es&um=1&oi=scholar>.

Software, Grupo de Ingeniería del. 2006. Elicitación de Requisitos. s.l. : Departamento de, 2006.

spss. 2007. spss. spss. [Online] 2007. [Cited: febrero 13, 2007.] www.spss.com/worldwide..

Statistica. 2004. softwarecientifico. softwarecientifico. [Online] StatSoft Inc., 2004. [Cited: febrero 3, 2007.] <http://www.softwarecientifico.com/paginas/cualsta.htm>.

T., José Camilo Daccach. 1997. Delta. Delta. [Online] 1997. [Cited: mayo 5, 2007.] <http://www.deltaasesores.com>.

Torres, Soulberto Lorenzo. monografías. monografías. [Online] <http://www.monografias.com/trabajos15/analista-sistem>.

Vera, Karol Loyola. 2002. mmug. mmug. [Online] 2002. [Cited: enero 30, 2007.] <http://www.mmug.cl>.

1999. Visual Paradigm for the Unified Modeling Language: VP-UML 5.0 User's Guide (part 1). s.l.: Visual Paradigm, 1999.

Wieggers, Karl E. 2006. More about Software Requirements: Thorny Issues and Practical Advice. s.l.: Microsoft Press, 2006.

Young, Ralph R. 2004. The Requirements Engineering Handbook. Boston: Artech House, 2004.