

# Universidad de las Ciencias Informáticas

## Facultad 1



**Título:** Módulo para la gestión de trazas en el servidor de despliegue del Sistema Único de Identificación Nacional de la República de Cuba.

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

**Autor(es):** Liliett Machado Hernández.

Yoan Palacio Machado.

**Tutor(es):** Ing. Adrian Torres Graña.

Ing. Jorge Eduardo Porrata Torres.

Ciudad de La Habana, julio de 2013.

"Año 54 de la Revolución"

## DECLARACIÓN DE AUTORÍA

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

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_.

\_\_\_\_\_  
Liliett Machado Hernández

Autor

\_\_\_\_\_  
Ing. Adrian Torres Graña

Tutor

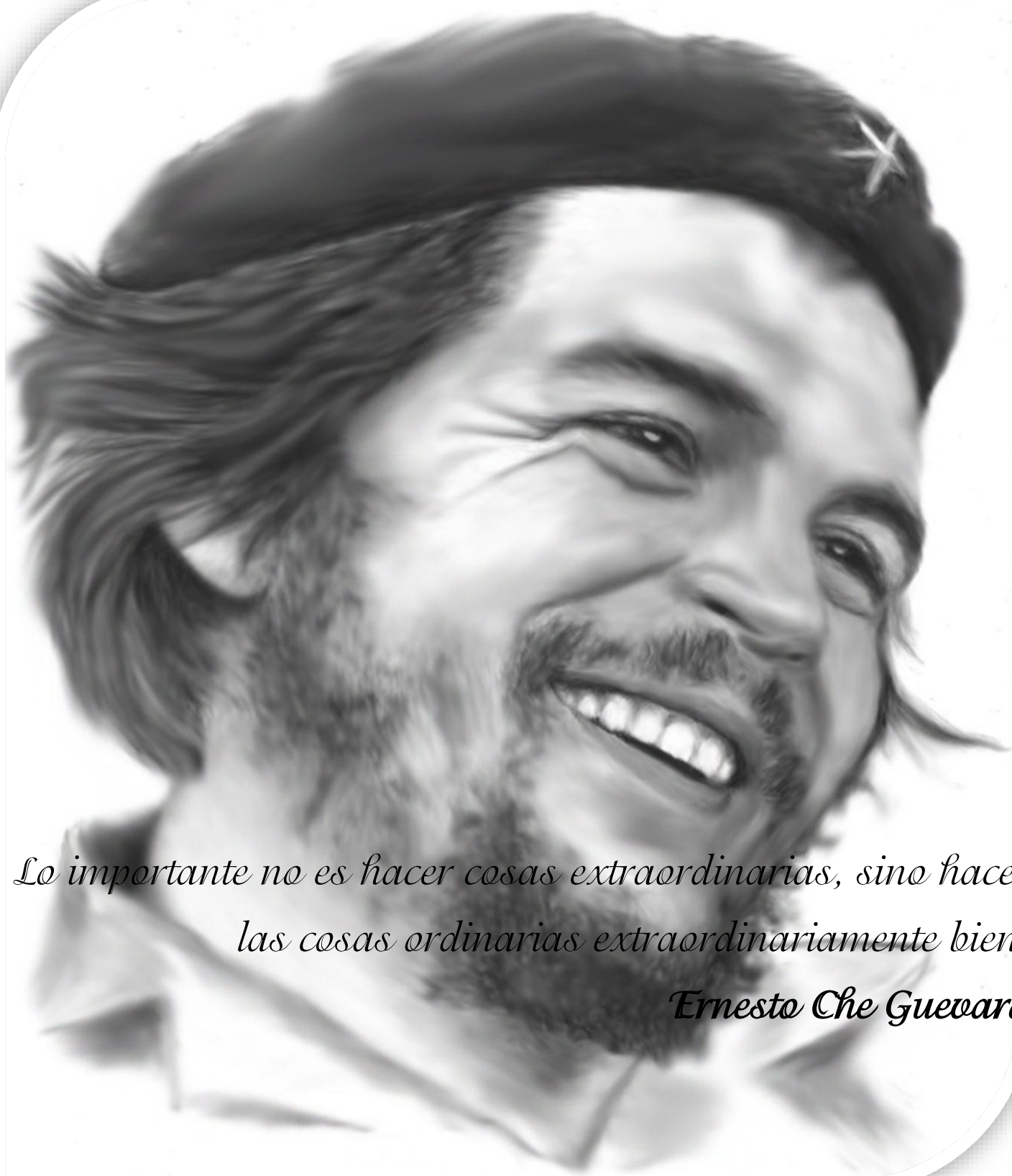
\_\_\_\_\_  
Yoan Palacios Machado

Autor

\_\_\_\_\_  
Ing. Jorge Eduardo Porrata Torres

Tutor

\_\_\_\_\_  
( )



*Lo importante no es hacer cosas extraordinarias, sino hacer  
las cosas ordinarias extraordinariamente bien.*

*Ernesto Che Guevara*

## DATOS DE CONTACTO

**Ing. Adrian Torres Graña:** Graduado de Ingeniero en Ciencias Informáticas, graduado en el 2010. Cuenta con varios años de experiencia como desarrollador en el proyecto Identidad Cuba, así como en el área relacionada a la minería de procesos. Es el Jefe del Sistema de Administración del Sistema Único de Identificación Nacional de la República de Cuba, actualmente se desempeña como desarrollador del Sistema Único de Identificación Nacional de la República de Cuba (SUIN) y como desarrollador del componente de trazas ETRACE para el Ministerio del Interior de la República de Cuba. Correo electrónico: [agrana@uci.cu](mailto:agrana@uci.cu)

**Ing. Jorge Eduardo Porrata Torres:** Graduado de Ingeniero en Ciencias Informáticas, graduado en el 2010. Cuenta con varios años de experiencia como desarrollador en el proyecto Identidad Cuba. Es el Jefe del sub-proyecto Sistema de Personalización de Documentos de Identidad, perteneciente al Sistema Único de Identificación Nacional de la República de Cuba (SUIN), y actualmente se desempeña como desarrollador del proyecto Sistema de Personalización de Documentos de Identidad (SPDI). Correo electrónico: [jeporrata@uci.cu](mailto:jeporrata@uci.cu)

## AGRADECIMIENTOS

### De Liliett:

*Yo le quiero agradecer por todo el apoyo que me han dado desde la primera vez que abrí los ojos a: Mis papitos lindos, que los quiero con la vida.*

*A todos mis abuelos en especial a mi abuelito Miguel que aunque no pudo estar conmigo en este momento de mi vida, fue una de mis fuentes de inspiración para llegar hasta el final.*

*A una de las personas más importantes de mi vida en el transcurso mi carrera a: Julio por siempre apoyarme en todo, cumplir todos mis caprichos y por quererme como nunca nadie me ha querido.*

*A mi hermana por ser mi segunda mamá y apoyarme en todo momento.*

*A mis tías preciosas por estar siempre a mi lado, cuando me hace falta.*

*A mis primas por preocuparse por mí y por quererme mucho.*

*A mis tutores y demás compañeros del proyecto por apoyarnos en el desarrollo de este trabajo, sin ustedes no se hubieran realizado mis sueños.*

*A todos mis súper friends por estar siempre a mi lado cuando más falta me ha hecho.*

### De Yoan:

*Quiero agradecer infinita y eternamente a mis padres por haber estado ahí siempre, dándome todo su apoyo incondicional cada segundo de mi vida, los amo con todo mi corazón.*

*A mi hermano por ayudarme a llegar hasta este maravilloso momento.*

*A mi familia, tías y tíos, primas y primos, a mi abuelo, en fin, a todos por confiar en mí siempre y darme ánimos para seguir adelante, son muy especial.*

*A mis amigos, los mejores amigos del mundo, los que siempre han estado en las buenas, en las malas, los de toda la vida y los que llegaron después.*

*A mi compañera de momentos duros y difíciles, de momentos felices y tristes, a: Osneidis por haberme soportado durante estos últimos tres años de carrera, por su apoyo, su temple, su lealtad, su amor sincero y por la gran persona que es. Gracias por tu amor.*

*A mis tutores Jorge y Adrian por su preocupación, su esfuerzo y apoyo en las situaciones de aprietos. Gracias por todo.*

*A todos mis compañeros del proyecto que de una forma u otra me brindaron su ayuda en los momentos que les solicité. Nunca lo olvidaré.*

*A todas las personas que hicieron posible cumplir este sueño. Gracias a todos.*

*A Dios por permitirme llegar hasta aquí y creer en él a mi manera.*

## DEDICATORIA

### De Liliett:

*Quiero dedicarle este trabajo que es el fruto de 5 años de sacrificio y estudio a mis padres. A mi mamá por soportar todos los cambios de humor, los días de estrés y sobre todo por darme mucho amor y cariño que es lo que me ayuda a seguir adelante, Te Quiero Mucho!! A mi papito querido por siempre apoyarme y guiarme cuando más perdida me he encontrado, por todos los malos viajes que soportó para que yo llegara al final y sobre todo por quererme mucho, Te Quiero Mucho!!*

*A Julio César por pasar estos 5 años a mi lado, apoyándome desde el principio hasta el final. Por ser lo más lindo que me ha pasado en la vida. Sin ti no hubiera llegado a graduarme. Nunca te olvidaré!!*

*A toda mi familia por brindarme su amor, cariño y apoyo, en especial a mi hermana y mis sobrinos preciosos que son mis hijos adoptados, que ocupan un espacio muy grande y especial en mi corazón, Los Amo!!*

### De Yoan:

*Este trabajo va dedicado especialmente a las personas más importantes de mi vida, las que me inculcaron todos los valores que llevo conmigo, a mis padres, los mejores del mundo sin duda alguna. Soy lo que soy por ustedes, y he llegado hasta aquí por ustedes. De no tenerlos a mi lado todo este tiempo nada de esto hubiera sucedido.*

*A mi hermano y a mi familia que me adoran como yo a ellos.*

*A mi compañera de todos los momentos en estos últimos tres años de carrera: Osneidis.*

*A mis amigos que siempre han estado ahí cuando los he necesitado y me han apoyado sin flaqueza alguna.*

*A todos los que desearon que este trabajo fuera posible y lo hicieron con toda la sinceridad del mundo.*

*A todas estas personas va dedicado este trabajo con el sentimiento más puro y genuino desde lo profundo de mi corazón.*

## **RESUMEN**

Día a día las tecnologías de la información y las comunicaciones tienen mayor presencia en todos los aspectos de la vida laboral. Ofreciendo a la vez nuevos espacios de innovación.

El Ministerio del Interior (MININT) como parte del desarrollo tecnológico que se está llevando a cabo en nuestro país. Se dió la tarea de automatizar los procesos que se desarrollan en las instituciones de identificación de la población que forma parte de la Dirección de Identificación y Registro (DIR). Permitiendo que las actividades que se realicen tengan mayor seguridad, una mejor calidad la documentación y un registro único de la población.

En la actualidad se necesita controlar las trazas técnicas que son generadas en el servidor de despliegue del Sistema Único de Identificación Nacional de la República de Cuba (SUIN). Estas son generadas por diferentes aplicaciones las cuales crean sus propias trazas, provocando que se torne complejo y engorroso su análisis. Se hace necesario para los desarrolladores de este sistema conocer lo que ocurre en las aplicaciones, ya que es almacenada información de gran importancia para el correcto funcionamiento del mismo.

En el presente trabajo de diploma se presenta un sistema para la gestión de trazas en el servidor de despliegue del SUIN. Cuyo objetivo es desarrollar un software que realice el procesamiento del gran volumen de información que se guarda diariamente en el servidor del SUIN. Permitiendo el estudio y tratamiento de los datos para extraer conclusiones e información relevante de ellos.

**Palabras Claves:** Trazas, Análisis, Software.

## TABLA DE CONTENIDOS

INTRODUCCIÓN.....	10
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	14
1.1 Principales conceptos asociados al dominio del problema .....	14
1.2 Sistemas gestores de trazas en el mercado.....	15
1.2.1 Servicios Catrian .....	15
1.2.2 Sawmill .....	16
1.2.3 Webalizer .....	17
1.2.4 WebLog Expert Lite .....	17
1.2.5 WebSpy Vantage.....	18
1.3 Metodología, Tecnología y Herramientas utilizadas en la solución. ....	19
1.3.1 Metodología de desarrollo. ....	19
1.3.2 Lenguaje de modelado .....	22
1.3.3 Herramientas de modelado .....	22
1.3.4 Plataforma de desarrollo. ....	24
1.3.5 Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés).....	26
1.3.6 Lenguaje de programación.....	27
1.3.7 Gestor de base de datos .....	29
Conclusiones Parciales.....	30
CAPÍTULO 2: PROPUESTA DE SOLUCIÓN .....	31
2.1 Causas que originan la situación problemática .....	31
2.2 Propuesta de sistema .....	31
2.3 Definición de requisitos funcionales .....	34
2.3.1 Especificación de requisitos funcionales.....	34
2.3.2 Descripción de requisitos funcionales.....	34
2.4 Requisitos no funcionales .....	35



2.5 Arquitectura de software .....	37
2.6 Patrones de diseño .....	39
2.7 Diagrama de clases del diseño .....	43
2.8 Modelo de datos .....	44
2.9 Descripción del modelo de datos .....	45
Conclusiones Parciales.....	45
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS.....	47
3.1 Estándares de codificación .....	47
3.2 Tratamiento de errores.....	48
3.3 Implementación.....	48
3.4 Diagrama de componentes .....	49
3.5 Diagrama de despliegue .....	50
3.6 Prototipo de interfaz del módulo.....	50
3.7 Pruebas .....	51
3.7.1 Pruebas de caja blanca .....	51
3.7.2 Pruebas de caja negra.....	52
3.8 Beneficios del sistema .....	54
Conclusiones Parciales.....	54
CONCLUSIONES GENERALES .....	56
RECOMENDACIONES.....	57
REFERENCIA BIBLIOGRÁFICA .....	58
BIBLIOGRAFÍA CONSULTADA .....	61
GLOSARIO DE TÉRMINOS .....	62
ANEXOS .....	64

## Índice de Figuras

Figura 1.1: Fases de MSF para CMMI. ....	20
Figura 2.2: Modelo de dominio.....	33
Figura 2.3: Arquitectura del Módulo de Gestión de Trazas en el Servidor de Despliegue del SUIN. ....	38
Figura 2.4: Ejemplo del patrón encapsulación.....	40
Figura 2.5: Ejemplo del patrón subclase. ....	40
Figura 3.10: Diagrama de componentes de la aplicación.....	49
Figura 3.11: Diagrama de despliegue de la solución.....	50
Figura 9.12: Interfaz del módulo para la gestión de trazas del servidor de despliegue del SUIN.....	51
Figura 3.13: Resultado de la prueba de caja blanca realizada a la funcionalidad InsertarVariable.....	52
Figura 3.14: Resultados de las iteraciones. ....	54
Figura 15: Interfaz para insertar la fecha de inicio de la carga. ....	68
Figura 16: Interfaz para insertar la fecha de fin de la carga.....	69
Figura 17: Interfaz después de llevado a cabo la carga de las trazas. ....	69
Figura 18: Interfaz que muestra las opciones de filtrado. ....	69
Figura 19: Diagrama de componentes del servicio.....	74
Figura 20: Diagrama de componentes del servicio en el cliente.....	74

## Índice de Tablas

Tabla 2.1: Descripción de roles.....	34
Tabla 2.2: Descripción RF “Cargar Trazas”.....	35
Tabla 2.3: Descripción de la tabla Variables. ....	45
Tabla 2.4: Descripción de la tabla Logs. ....	45
Tabla 3.5: Descripción del CPRF1 “Cargar Trazas”. ....	53
Tabla 6: Especificación del RF "Mostrar Resultado del Análisis".....	64
Tabla 7: Especificación del RF "Filtrar Información". ....	65
Tabla 8: Especificación del RF "Eliminar Basura". ....	66
Tabla 9: Especificación del RF "Establecer Etiquetado".....	67
Tabla 10: Especificación del RF "Establecer Configuración del Servicio".....	67
Tabla 11: Especificación del RF "Cargar Fichero". ....	68
Tabla 12: Especificación del RF "Generar Log". ....	68
Tabla 13: Descripción del CPRF2 “Mostrar Resultados”. ....	70
Tabla 14: Descripción del CPRF1 “Filtrar Información”. ....	71
Tabla 15: Descripción de Variables del CPRF3 “Filtrar Información”.....	71
Tabla 16: Descripción del CPRF4 “Eliminar Basura”.....	72
Tabla 17: Descripción del CPRF5 “Establecer Etiquetado”. ....	72
Tabla 18: Descripción de Variables del CPRF5 “Establecer Etiquetado”.....	73
Tabla 19: Descripción del CPRF6 "Establecer Configuración del Servicio". ....	73
Tabla 20: Descripción del CPRF6 "Cargar Fichero". ....	74
Tabla 21: Descripción del CPRF6 "Generar Log". ....	74

## Introducción

A finales del año 2008 se realizó un estudio a los sistemas de identificación de la población nacional, para conocer el estado actual en que se encontraban, y proponer la informatización de los mismos. Esto garantizaría la identificación e inscripción de personas con procesos y documentos seguros, obteniendo un sistema único. Después de concluir este período se realizó un intercambio de ideas entre el Ministerio del Interior (MININT) y la Universidad de las Ciencias Informáticas (UCI), con el fin de desarrollar un Sistema Único de Identificación Nacional (SUIN).

Dicho sistema daría solución a todos los problemas planteados, dando paso a la creación del proyecto Identidad de Cuba. Cuyo objetivo principal es la automatización de los procesos realizados, en las oficinas de Carné de Identidad y Registro de la Población (CIRP). Para la creación de este sistema se llevó a cabo un estudio profundo que permitió, determinar los principales problemas que afectan a dicha entidad. Algunos de estos se mencionan a continuación:

- Demora en los trámites.
- Mala atención a la población.
- Falta de interés del personal en la realización de su trabajo.

El SUIN actualmente se encuentra desplegado por todo el país. Garantizando la inscripción e identificación de personas con procesos y documentos seguros. Contiene el uso de la biometría, a partir del cual se genera una base de datos única que permite la identificación de la población.

Este sistema genera un gran volumen de trazas en el servidor de despliegue. Careciendo de información visual, lo que provoca que se torne complejo y engorroso su análisis. Al ser generadas diariamente por diferentes módulos, crea que esta información se torne incoherente y eventualmente inconsistente.

La carencia de estos elementos influye grandemente en los resultados de la organización. El Sistema Único de Identificación Nacional de la República de Cuba presenta dificultades para supervisar, controlar y analizar el sistema de trazas generadas en el servidor de despliegue.

Teniendo en cuenta la **Situación Problemática** antes planteada se presenta el siguiente **Problema Científico**:

¿Cómo facilitar el proceso de gestión de las trazas generadas en el servidor de despliegue del Sistema Único de Identificación Nacional de la República de Cuba?

La investigación tiene como **Objeto de Estudio** la gestión de trazas en servidores. Delimitando como **Campo de Acción** la gestión de trazas en el servidor de despliegue del Sistema Único de Identificación Nacional.

El presente trabajo propone como **Objetivo General** desarrollar un módulo que permita gestionar las trazas generadas en el servidor de despliegue del Sistema Único de Identificación Nacional de la República de Cuba.

**Posibles Resultados:**

- Un módulo que permita gestionar las trazas generadas en el servidor de despliegue del Sistema Único de Identificación Nacional de la República de Cuba.
- Documentos de especificación de requisitos.
- Manual de usuario.

**Posibles Resultados Socio - Económicos:**

- Ambiente amigable para los usuarios.
- Mejorará los tiempos de respuestas ante situaciones poco tangibles al análisis manual.
- Reduce en gran medida el gasto monetario, debido a que todos los sistemas homólogos referentes a este tema son propietarios.

Para dar cumplimiento al objetivo planteado se definieron los siguientes **Objetivos Específicos:**

- Valorar los fundamentos teóricos y tendencias en el desarrollo de un analizador de trazas para el servidor de despliegue.
- Levantar requerimientos funcionales y no funcionales del módulo a desarrollar.
- Diseñar e implementar el módulo de análisis de trazas del servidor de despliegue del SUIN.
- Realizar pruebas de caja negra y caja blanca a la solución implementada.

Se proponen las siguientes **Tareas** a cumplir:

- Análisis de sistemas nacionales e internacionales relacionados con la temática tratada; definición de ventajas y deficiencias. (Liliett Machado Hernández y Yoan Palacios Machado)
- Definición de todas las trazas que pueden ser generadas en el servidor de despliegue del Sistema Único de Identificación Nacional. (Yoan Palacios Machado)
- Análisis de formas de representación de la información obtenida a partir del análisis a las trazas del servidor para contribuir a la asimilación de la misma por el usuario final. (Liliett Machado Hernández)
- Análisis de la conveniencia de uso de las herramientas definidas por el proyecto. (Liliett Machado Hernández y Yoan Palacios Machado)
- Realización entrevistas con los clientes para tener una primera aproximación de los requerimientos funcionales del módulo a desarrollar. (Liliett Machado Hernández)
- Realización del artefacto catálogo de requisitos. (Liliett Machado Hernández)

- Realización del artefacto especificación de requerimientos. (Liliett Machado Hernández)
- Realización prototipos de interfaz de usuarios. (Liliett Machado Hernández)
- Análisis de las tecnologías Asp.Net, JavaScript para el diseño e implementación de la propuesta.(Yoan Palacios Machado)
- Realización manual de usuario del sistema. (Yoan Palacios Machado)
- Realización de pruebas de caja blanca para comprobar legibilidad y claridad del código.(Yoan Palacios Machado)
- Realización de pruebas de caja negra para verificar funcionalidades del módulo implementado. (Liliett Machado Hernández)
- Realización de pruebas de aceptación por parte de los clientes implicados. (Liliett Machado Hernández y Yoan Palacios Machado)

Para la elaboración de este trabajo se tiene en cuenta algunos métodos tradicionales investigativos. A continuación se mencionarán cada uno de ellos y de qué forma se ponen de manifiesto en la investigación. Los **Métodos Teóricos** aplicados en la investigación son: **método histórico-lógico** este método permitió adquirir nuevos conocimiento sobre la evolución y desarrollo de diferentes elementos asociados a la gestión de trazas en los servidores, el **método de análisis y síntesis**, pues se realiza un análisis de la bibliografía que se utilizó para el estudio del tema y el **método de modelación**, ya que este método es utilizado en una de las fases de la metodología para el modelado de los artefactos que se generan.

El **Método Empírico** que se utiliza es la **entrevista**, debido a que posibilitó obtener información luego de conversaciones planificadas con personas que tienen experiencia del tema.

### **Justificación de la Investigación:**

El Sistema Único de Identificación Nacional de la República de Cuba tiene un gran valor para la sociedad, ya que contribuye a la organización y control de la población. Dicho sistema es de un alto valor monetario a nivel mundial. Al ser de producción nacional conlleva a un gran ahorro, obteniendo un máximo beneficio para la evolución e informatización de la misma.

El módulo de gestión de trazas como parte indisoluble de este proyecto. Constituye un apoyo al desarrollo del Sistema Único de Identificación Nacional de la República de Cuba. Brindando la posibilidad de realizar un análisis esquemático de forma organizada y precisa a las trazas generadas en el servidor de despliegue del SUIN, contribuyendo con el funcionamiento del mismo.

El documento quedará estructurado de la siguiente manera:

### **Capítulo 1: Fundamentación Teórica.**

Se definen los conceptos fundamentales sobre las tecnologías que se abordaron en la concepción del sistema. Se describen las herramientas y metodologías empleadas en la construcción del software. Así como un análisis detallado de los sistemas homólogos a la solución propuesta.

## **Capítulo 2:** Propuesta de Solución.

Se realiza una descripción detallada de las características del sistema, se definen los requisitos funcionales y no funcionales que guiarán el desarrollo del sistema de análisis de trazas. Describiéndose los principales elementos que componen la arquitectura y que guían el proceso de construcción del software. Además de definir las clases que intervienen en la solución. Así como los servicios que dan soporte a las funcionalidades, se obtienen los artefactos generados a partir de la aplicación de la metodología de desarrollo utilizada. Se realiza además una descripción detallada de las entidades que tienen persistencia en la base de datos.

## **Capítulo 3:** Implementación y Pruebas.

Se describen los aspectos relacionados con el flujo de implementación y la validación del producto. Se presentan los estándares de codificación empleados, así como el tratamiento a errores dado. Se detallan los diagramas correspondientes al flujo de implementación y se describen los prototipos de interfaz de usuario. Para comprobar el correcto funcionamiento del módulo, se le realizan pruebas de caja blanca y de caja negra.

## Capítulo 1: Fundamentación Teórica.

Con el crecimiento explosivo de las fuentes de información disponibles en Internet, se hace necesario que los desarrolladores utilicen herramientas automatizadas. Para el descubrimiento de los recursos deseados de la información y poder conocer y analizar sus patrones de uso. En el presente capítulo se describen los principales conceptos relacionados con el dominio del problema y el objeto de estudio. Haciendo un estudio de las herramientas utilizadas para el análisis de trazas. Se presenta una fundamentación teórica de la situación actual, en cuanto a la utilización e importancia de las trazas en los servidores. Además de realizar un análisis de las herramientas existentes, para el diseño del sistema y la propuesta de implementación y desarrollo.

### 1.1 Principales conceptos asociados al dominio del problema

En el siguiente epígrafe se realizará una explicación de los conceptos necesarios, para el desarrollo de este trabajo de diploma. Permitiendo una mejor comprensión del tema a elaborar.

#### Servidor web

Un concepto que es necesario conocer para dar comienzo al estudio de este tema es el de servidor web. Según (García 2012) un *servidor web* o *servidor HTTP* es un programa que realiza el procesamiento de una aplicación del lado del servidor. Este genera o cede respuestas en cualquier lenguaje o aplicación del lado del cliente. Se utiliza el protocolo HTTP para llevar a cabo la transmisión de los datos. El código que suele ser recibido por el cliente es interpretado por los navegadores web.

Los servidores web son de gran importancia en el ámbito de las aplicaciones web. Estos son los encargados de prestar servicios, los cuales son consumidos por el cliente. Los servidores web, disponen de un intérprete de otros lenguajes de programación que ejecutan código embebido dentro del código HTML.

#### Trazas en el servidor

Según el gran diccionario de Oxford y (Norberto López Mazón 2011) se define como un **log**: Un registro oficial de eventos durante un periodo de tiempo en particular. Generalmente la información es almacenada en un fichero .log. Al que se le va incrementando información a medida que se realizan acciones en el sistema. Estos ficheros son utilizados para guardar información sobre actividades de sistemas informáticos. Además permiten detectar problemas que ocurren en el caso de las aplicaciones con respecto a su funcionamiento o incidencias de seguridad.



En el servidor del SUIN los log se utilizan para almacenar la información referente a los sucesos que ocurren en la aplicación y en la base de datos. Mediante los log se puede descubrir información que permita detectar posibles problemas en caso de que el sistema tuviera una anomalía.

En el libro (Carmen de Pablos Heredero 2012) se expresa que las **trazas** son las vías que recorren los datos a través de los diferentes programas informáticos. Permitiendo la verificación del correcto funcionamiento de los sistemas. Las trazas son utilizadas para comprobar la ejecución de la validación de datos y no deben modificar el sistema en ninguna circunstancia.

En el servidor del SUIN las trazas contienen la información referente a la secuencia de acciones e instrucciones que ocurren durante la ejecución del sistema. Con la gestión de las trazas almacenadas, se puede detectar posibles anomalías que ocurren en el sistema. Mediante las trazas se puede conocer sobre cada una de las acciones que se realizan en la ejecución del sistema.

## 1.2 Sistemas gestores de trazas en el mercado

En la actualidad los sistemas informáticos generan información de gran valor, que por su formato y su volumen, no son muy utilizados estos son conocidos como logs. Los cuales recogen cada una de las aplicaciones y sistemas que configuran la infraestructura de TI<sup>1</sup>. Existen diversos sistemas en el mercado para realizar los distintos análisis, permitiendo una mejor comprensión de los datos guardados en dichos ficheros. Entre estos sistemas se encuentran:

### 1.2.1 Servicios Catrion

Según (S.L. 2010) Catrion es una empresa española especializada en la provisión de servicios gestionados de infraestructura y seguridad informática, para administración y empresas. Dentro de los múltiples servicios que brinda dicha empresa se encuentra el gestionado de análisis y correlación de logs. Este servicio permite explotar esta información de manera sistemática generando informes o alarmas de fácil comprensión y uso. Los informes pueden incluir hasta 70 indicadores diferentes del comportamiento de los usuarios en la web. El servicio recupera los ficheros del servidor web y los transfiere al servidor de análisis donde son tratados por una herramienta de análisis de logs. La información se almacena en una base de datos que facilita la manipulación de un gran volumen de información. Una base de datos se actualiza incrementalmente cuando llega un nuevo fichero de log, normalmente esto ocurre una vez al día.

Este servicio contiene una herramienta de análisis la cual se configura para cada tipo de fichero log y tipo de información a analizar. Además de incluir las relaciones existentes dentro del contenido de los logs. Permite crear perfiles donde se pueden definir filtros para realizar búsquedas de una información específica

---

<sup>1</sup> Tecnología de Información.

y configurar informes deseados. Estos informes son generados a partir de un acceso directo a la base de datos.

Catrian propone que a la vez que realice la recogida de los logs, muestre un primer informe con los logs de 1 mes. En vista del resultado, permita optimizar el proceso en función del tipo de informes de interés para el cliente. Mediante este servicio se gestionan las trazas de los servidores que se ejecutan sobre los sistemas operativos Windows, Unix y Linux.

El código fuente, los diseños gráficos y demás funcionalidades, están protegidos por la legislación española sobre los derechos de propiedad intelectual e industrial a favor de Informática Catrian, S.L y no se permite la reproducción y/o publicación, total o parcial de los servicios y herramientas antes mencionadas.

### 1.2.2 Sawmill

Según (2013) Sawmill es una potente herramienta de análisis de log. Está especialmente diseñado para analizar cualquier tipo de logs de acceso a servidores web. Se ejecuta como un programa que regula el intercambio de información entre clientes web y las aplicaciones que se ejecutan en el servidor, o sea, lo que se conoce como un programa CGI<sup>2</sup>. Este puede utilizarse desde cualquier navegador para configurar y ejecutar Sawmill o para ver estadísticas de páginas, funcionalidad que podría abrir agujeros de seguridad en el servidor, ya que una aplicación en CGI mal diseñada podría permitir acceso total o parcial al servidor. Las estadísticas son jerárquicas, atractivas y poseen enlaces que facilitan la navegación. El programa incluye una completa documentación.

Este software ofrece una gran cantidad de opciones, incluida una base de datos persistente, el control sobre la apariencia de las páginas de estadísticas y diversas opciones de filtrado sobre el log. Muestra, tras su instalación, una interfaz amigable en Windows Internet Explorer y presenta, en un cuadro de selección de opciones ubicado a la izquierda, una serie de estadísticas posibles:

- Cantidad de visitas por hora, por día, por mes, etcétera.
- Horas pico y horas de baja audiencia.
- Páginas más visitadas.
- Páginas de entrada y salida más frecuentes del sitio.
- Utilización de buscadores, clasificación de palabras clave empleadas para buscar.

El producto con licencia está registrado y protegido por las leyes de copyright de Estados Unidos y las disposiciones de tratados internacionales.

---

<sup>2</sup> Del inglés Common Gateway Interface (Interfaz de Entrada Común).

### 1.2.3 Webalizer

Según (Barrett 2012) el Webalizer es un rápido servidor web gratuito de archivos de registro en un programa de análisis. Produce de manera detallada y fácilmente configurable, informes de uso en formato HTML para su visualización con un navegador web estándar. Está escrito en C para ser extremadamente rápido y portátil. Puede procesar cerca de 70.000 registros por segundo, lo que significa que un archivo de registro con unos 2 millones de visitas puede ser analizado en unos 30 segundos.

Los informes generados se pueden configurar desde la línea de comandos, o más comúnmente, mediante el uso de uno o más archivos de configuración. La información detallada sobre las opciones de configuración se encuentra en el archivo README, que se suministra con todas las distribuciones. Soporta múltiples idiomas, presenta ilimitado tamaño de los archivos de registro y los registros parciales son compatibles. Eliminando la necesidad de mantener grandes archivos mensuales en el sistema.

Este programa fue ideado para correr sobre la plataforma libre (Linux, Squid), puede ser redistribuido y/o modificado bajo los términos de la Licencia Pública General de GNU, ya sea la versión 2 de la Licencia, o cualquier versión posterior, y siempre que el aviso de copyright anterior y el permiso se incluya con todas las copias distribuidas de este software o derivados.

### 1.2.4 WebLog Expert Lite

Según (2012-2013) este software analiza de forma rápida y precisa, los ficheros log generados, por un servidor web, ya sea Apache o IIS<sup>3</sup>. Generando un informe completo, con diversos datos relacionados con visitas de un sitio web, estadísticas de actividades, los ficheros más descargados entre otros datos.

WebLog Expert Lite muestra los resultados de su análisis en formato HTML los cuales pueden ser visualizados en los navegadores web. Muestra estadísticas de acceso, datos relevantes sobre visitas. Además de generar gráficas que ayudan a percibir cualquier dato relevante que pueda interesar.

Esta herramienta brinda funcionalidades como estadísticas de la actividad de todos los días, de las horas del día, por días de la semana, por semana y por mes. También estadística del acceso, estadísticas de las páginas, archivos, imágenes, directorios, consultas, tiempo de vista, páginas de entrada, páginas de salida, rebotes, senderos a través del sitio, tipos de archivos, dominios virtuales y cargar servidores balanceados. Además información acerca de los errores así como los tipos de error, información detallada de errores entre otras funcionalidades que permiten dar seguimiento a la actividad en los servidores.

Este software es de carácter privativo y está protegido por la compañía israelí Alentum Software Ltd. Una licencia de WebLog Expert (cualquier edición) permite usar el programa en un equipo y analizar los datos de registro de un número ilimitado de sitios. Cualquier usuario que compre el programa tendrá derecho a 12

---

<sup>3</sup> Del inglés Internet Information Server (Servicio Información de Internet)

meses de actualizaciones gratuitas, a las nuevas versiones de la edición de compra y al precio especial de actualización después de este período.

### **1.2.5 WebSpy Vantage**

Según (Ltd. 2012) WebSpy Vantage emplea un módulo web que permite a los administradores distribuir de forma segura los informes de una organización. Con este software las organizaciones pueden monitorear y dar a conocer, no sólo el uso de internet y el comportamiento de navegación web. También el uso del correo electrónico, la mensajería instantánea, los registros de eventos, los routers, el tráfico de visitantes en el sitio web, firewalls, antivirus y aplicaciones anti-spam.

Además puede identificar el mal uso de internet y correo electrónico, resolver problemas de la red y los problemas de descarga. El fortalecimiento de los controles de seguridad, preserva el ancho de banda, reduciendo al mínimo los riesgos de litigios, y la identificación del origen de la pérdida de datos o daños. La funcionalidad principal consiste en importar texto o logs desde una amplia variedad de productos de otros proveedores, o de sucesos de Windows.

Registra directamente desde los servidores o estaciones de trabajo de la red. Vantage soporta más de 200 formatos de archivo de registro de diferentes proveedores. Por otra parte se utiliza para asignar permisos a usuarios y grupos que definen lo que pueden y no pueden tener acceso en el módulo web. Elige entre una amplia lista de informes web, correo electrónico, redes y seguridad o crear sus propias plantillas de informes personalizados para su entorno. Los informes pueden ser generados en HTML, Microsoft Word, CSV y formatos de texto. Crea documentos separados de informes para cada usuario, departamento, fecha o cualquier otro elemento registrado.

Vantage es un software de carácter privativo diseñado para correr sobre los sistemas operativos Microsoft Windows, Server 2003 SP1, Vista, Server 2008 y Windows 7. Protegido por la compañía de servicios WebSpy, establecida desde 1996 con oficinas estratégicamente ubicadas en los Estados Unidos, el Reino Unido y Australia.

### **Conclusiones del estudio de los sistemas homólogos**

El Sistema Único de Identificación Nacional de la República de Cuba presenta ciertas particularidades en cuanto a la generación de las trazas. Estas particularidades se basan fundamentalmente en que no presentan un formato estándar. Este puede sufrir cambios por lo que se debe tener en cuenta la variabilidad y versatilidad de la estructura de las trazas que componen los logs.

Algunas de las causas que imposibilitan el uso de los analizadores de logs antes mencionados son:

Estos sistemas fueron desarrollados para el trabajo con las trazas rigiéndose por un estándar definido para las estructuras de las mismas. No realizan un procesamiento de un conjunto de trazas sino que las analiza

una por una. Lo que es una desventaja debido a que en el caso del SUIN genera un gran volumen de trazas y llevar a cabo el procesamiento uno por uno sería un trabajo engorroso. Además de que no brindan la posibilidad de ejecutar una carga sistemática y automatizada, o sea que la carga se debe realizar de forma manual. Igualmente se debe tener en cuenta de que estos sistemas son propietarios, por lo que se debe realizar la compra del producto para poder utilizar todas las funcionalidades que estos sistemas presentan. Después de llevar a cabo el presente estudio a los sistemas similares. Se llega a la necesidad de crear un sistema propio que permita llevar a cabo la gestión de trazas en el servidor de despliegue del SUIN. Que se adapte a la variabilidad del formato que presentan las trazas y que permita realizar la carga de todas las trazas que se encuentren en los servidores de manera automática y sistemática.

### **1.3 Metodología, Tecnología y Herramientas utilizadas en la solución.**

En el siguiente epígrafe se caracterizan la metodología, tecnologías y herramientas que permiten el desarrollo del módulo propuesto. Estos son definidos por políticas del proyecto.

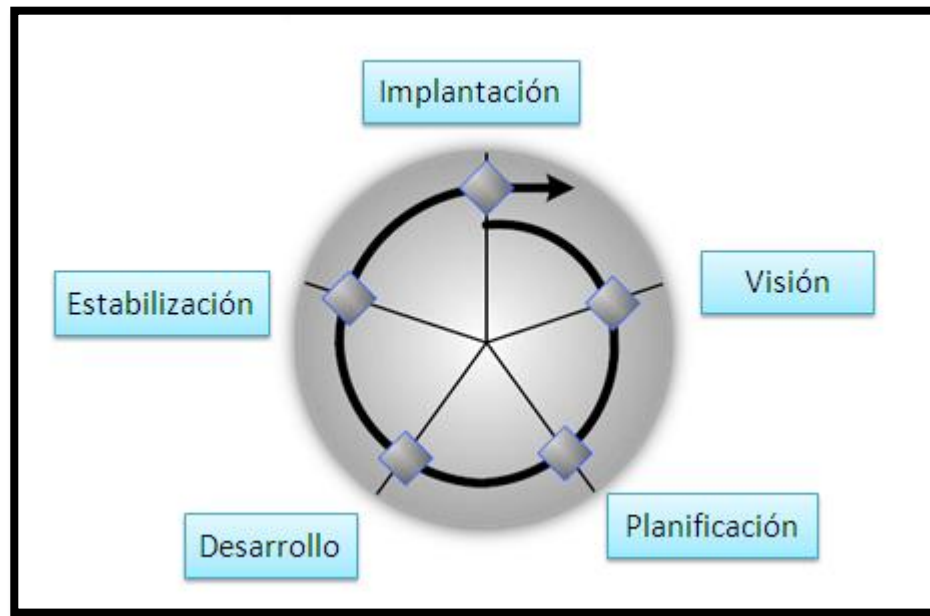
#### **1.3.1 Metodología de desarrollo.**

La metodología de desarrollo (Colombia 2000) no es más que un marco de trabajo. Es utilizado para estructurar, planificar y controlar los procesos de desarrollo en sistemas de información. En el mundo actual se utilizan diferentes metodologías para el desarrollo de software.

Para guiar el proceso de desarrollo de software se utilizará MSF para CMMI. MSF es una metodología utilizada para la construcción de sistemas basados en herramientas y tecnologías de Microsoft. CMMI permite la búsqueda de un lineamiento con las buenas prácticas, establecidas por la Universidad de las Ciencias Informáticas (UCI) para el desarrollo del software.

#### **MSF para CMMI**

Según (Fraile), (Luis Ramiro Villarroel Gonzalez 2008), (Microsoft 2012) es un proceso ágil de desarrollo de software. Cumple con los requerimientos para el nivel tres de CMMI. En este proceso se define cinco fases durante el desarrollo de un proyecto, que encapsula flujo de actividades y actividades. A continuación en la Figura 1.1 se muestran estas fases.



**Figura 1.1:** Fases de MSF para CMMI.

Cada fase concluye con un punto de control. Cada punto de control proporciona una oportunidad para autorizar el trabajo, continuar en el proyecto, cancelar o suspender el proyecto. (MSF 2002)

Fases de la metodología de desarrollo MSF para CMMI:

**Visión:** Esta fase aborda uno de los requisitos fundamentales para el éxito del proyecto. Se debe obtener la visión y el alcance del proyecto, el cual debe estar compartido, comunicado y entendido, y alineado con los objetivos del negocio. Con el desarrollo de esta fase el equipo y el cliente integran el proyecto definiendo los requisitos funcionales, sus alcances y restricciones. Esta fase culmina con el hito Visión y Alcance aprobados. Dentro de los documentos entregables se encuentran: el documento de Visión y Alcance, además la Matriz de Identificación de Riesgos.

**Planificación:** Durante esta fase, el equipo prepara la especificación funcional, a través del proceso de diseño, y elabora planes de trabajo, presupuestos y horarios para los distintos entregables. Se creará un borrador del plan maestro del proyecto y un cronograma que identifica puntos de control específico. Esta fase concluye con el hito Plan de Proyecto (Especificaciones y Cronogramas) aprobado. En esta fase los documentos entregables son: Documento de Especificación Funcional, Plan de Gestión de Riesgos, Master Plan del Proyecto y Cronograma del Proyecto Maestro.

**Desarrollo:** Durante la fase de desarrollo el equipo lleva a cabo la mayor parte de la construcción de la solución (documentación, código). Se concluye con el hito Alcance Completo. Dentro de los documentos entregables se encuentra: Código Fuente y Ejecutables, Actas de Aceptación de Entregas Parciales, Plan de Pruebas, Manual de Instalación y Operación, Documento de Arquitectura, Actas de Control de Cambios Aprobadas (que justifican el ajuste al alcance, tiempo y/o costo del proyecto, de existir).

**Estabilización:** La fase de estabilización lleva a cabo pruebas en una solución completa. Las pruebas durante esta fase hacen hincapié en el uso y el funcionamiento en el medio ambiente realista. Una vez que una generación ha sido considerada lo suficientemente estable como para ser un candidato de liberación, la solución se instala en un grupo piloto. La fase de estabilización culmina en el hito de: Aceptación de Pruebas. Dentro de los entregables de esta fase se encuentran: Acta de Aceptación de Pruebas, Acta de Capacitación a Usuarios y Mesa de Servicios, Acta de Entrega (comité de proyectos).

**Implantación:** Durante el desarrollo de esta fase, el equipo implementa la tecnología y componentes utilizados en el desarrollo de la solución. Apoya el funcionamiento y transición del proyecto, obtiene la aprobación final del cliente. Esta fase concluye con el hito Cierre de la Entrega. Dentro de la documentación entregable se encuentran: Acta de Implantación, Encuesta de Satisfacción del Cliente y el Acta de Cierre del Proyecto.

En esta metodología se definen diferentes roles como:

- Líder.
- Jefe de Desarrollo.
- Jefe de Producto.
- Arquitecto de Software.
- Desarrollador.
- Analista.
- Diseñador.
- Probador.
- Integrador.
- Documentador – Capacitador.
- Administrador de la Calidad.

Los diferentes miembros del equipo de desarrollo pueden desempeñar diferentes roles en las distintas fases del ciclo de vida del proyecto. Siendo responsables del cumplimiento de las actividades y de generar la documentación. En esta metodología se define que ningún rol es más importante que otro.

*Para los autores de este trabajo: Se utiliza esta metodología como guía del proceso de desarrollo de software, ya que permite la adaptación de los diferentes puntos de control que guían el proceso a nuestro entorno de desarrollo. Además de brindar la posibilidad de una mejor comunicación e interacción entre el cliente y el equipo de desarrollo disminuyendo en gran medida la cantidad de documentación a generar. Otra de las características por la que se selecciona a MSF para CMMI como metodología, es que admite combinar los roles, lo que permite que los miembros del equipo puedan desempeñar diversos roles influyendo en la formación profesional de estos.*



### 1.3.2 Lenguaje de modelado

Después de haber definido la metodología que guiará el proceso de desarrollo del sistema el equipo se enfocó en la selección del lenguaje para modelar dicho sistema. Durante el desarrollo del módulo de gestión de trazas se hace necesario modelar la realidad existente. Brindando la posibilidad de que se pueda tener una visión general de los procesos que se quieren informatizar. A continuación se especifica dicho lenguaje.

#### UML

UML (Pressman 2010), (Schmuller), (*Unified Modeling Language*<sup>4</sup>) es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema. Proporciona una forma estándar de escribir los planos de un sistema, tales como procesos del negocio, funciones del sistema, clases escritas en un lenguaje de programación específico, esquemas de bases de datos y componentes software reutilizables. Es la sucesión de una serie de métodos de análisis y diseño orientados a objetos. Está consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Mediante UML es posible establecer la serie de requisitos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código.

*Para los autores de este trabajo: Se usa este lenguaje para el desarrollo de los diferentes diagramas, ya que es uno de los lenguajes más utilizados para la modelación en un proyecto. Permite capturar la idea de un sistema y desarrollarla mediante un conjunto de símbolos y diagramas. Posibilita establecer los requisitos del sistema antes de llevarlo a cabo mediante la implementación. Este lenguaje presenta diferentes ventajas las cuales permitieron seleccionarlo para realizar el modelado dentro de estas se encuentran: presenta una notación estándar para el análisis y diseño de negocio del sistema, este lenguaje es independiente a la arquitectura y lenguaje de programación que se utilizará para desarrollar el sistema, además de incorporar las mejores prácticas a nivel internacional.*

### 1.3.3 Herramientas de modelado

Las herramientas de modelado en el entorno de desarrollo forman un componente importante en la actualidad. A continuación se presentan las herramientas de modelado que se utilizaron en el desarrollo del sistema.

---

<sup>4</sup> Lenguaje Unificado de Modelado.



## Altova UModel 2010 Enterprise Edition

Según el libro (Altova 2005-2013) y (2008) se define Altova UModel como una herramienta completa de desarrollo UML. Soporta todo tipo de diagramas, además de la generación de código y documentación, ingeniería inversa, y características avanzadas de usabilidad. Es compatible con la última especificación de UML 2.2. Es una herramienta de modelado que posee capacidad de compatibilidad con sistemas de control de versiones además de una estrecha integración con Microsoft Visual Studio.

Esta herramienta combina una amigable interfaz visual con funciones de usabilidad que favorecen a los usuarios con las más completas ventajas en el modelado de software. Las características de UModel 2010 para el desarrollo de software basado en las capacidades de modelado avanzado son:

- Soporte para los 14 tipos de diagramas UML.
- Generación de código fuente en lenguajes Java, C#, y VB.NET.
- Crea diagramas de secuencia desde el código fuente de la ingeniería inversa.
- Generación de documentación personalizable de proyecto.
- Integración con sistemas de control de versiones.
- Estrecha integración con *Microsoft Visual Studio* y *Eclipse*<sup>5</sup>.

*Para los autores de este trabajo: Se utiliza Altova UModel como herramienta de modelado por las diversas ventajas que ofrece. Siendo esta una herramienta totalmente compatible con el lenguaje de modelado que se utiliza en este trabajo (UML). Presenta una estrecha vinculación con Visual Studio lo que es de gran importancia ya que el modulo se implementa en dicho entorno de desarrollo. Además de ser una de las herramientas que compone las políticas del proyecto.*

## ER/Studio

Según (Altova) ER / Studio es una aplicación de modelado visual utilizado para la plataforma independiente de análisis lógico, arquitectura de datos y el diseño. Además de la plataforma de diseño específico de cada base de datos física y su construcción. Su potente, multi-nivel de entorno de diseño se ocupa de las necesidades diarias de los administradores de bases de datos, desarrolladores y arquitectos de datos que construyen y mantienen aplicaciones de bases de datos grandes y complejas.

Tiene una interfaz progresiva y simple. Ha sido diseñado para abordar eficazmente las cuestiones de facilidad de uso que han plagado a las herramientas de modelado de datos y CASE<sup>6</sup> durante la última década y más. ER / Studio ofrece:

- Las fuertes capacidades de diseño lógico.
- La capacidad de generar muchos diseños físicos a partir de un diseño lógico corporativo.

---

<sup>5</sup> Es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar proyectos.

<sup>6</sup> Del inglés Computer Aided Software Engineering (Ingeniería de Software Asistida por Computadora)

- Visual-Basic para Aplicaciones API para personalización de productos.
- Importación y exportación de metadatos de capacidades.
- Sofisticado XML, HTML, RTF y documentación basada en informes y las instalaciones.
- Base de datos completa del ciclo de vida de soporte.
- Enterprise modelo de gestión.
- Enterprise capacidades de comunicación.
- Almacén de datos y apoyo a la integración.
- Calidad en los diseños de base de datos.

*Para los autores de este trabajo: Se utiliza ER/Studio para hacer el modelado de la base de datos, lo que es de gran importancia ya que permite definir la estructura que tendrán las tablas y las relaciones entre ellas. Esta herramienta presenta una interfaz amigable. Presenta gran calidad en el desarrollo de los diseños de la base de datos permitiendo generar un diseño físico a partir de uno lógico, una de las grandes ventajas que posee.*

### **1.3.4 Plataforma de desarrollo.**

La plataforma de desarrollo .Net Framework pertenece a Microsoft. Está conformada por herramientas y bibliotecas de clases con las que se pueden crear aplicaciones tanto web como de escritorio.

#### **Microsoft .NET Framework 4.0**

En el sitio (Microsoft 2013) se expresa que el Framework .NET es una plataforma para desarrollar código administrado de Microsoft. Está compuesto por una serie de bibliotecas de clases y herramientas que permiten la creación de todo tipo de software.

Este framework es un componente de Windows que acepta la compilación y la ejecución de la generación de aplicaciones y servicios web XML.NET Framework, contiene dos componentes principales: Common Language Runtime y la biblioteca de clases de .NET Framework. El motor en tiempo de ejecución se puede considerar como un agente que administra el código en tiempo de ejecución y proporciona servicios centrales. Como la administración de memoria, la administración de subprocesos y la comunicación remota, al tiempo que aplica una seguridad estricta a los tipos y otras formas de especificación del código que promueven su seguridad y solidez.

Un componente principal de este framework es la biblioteca de clases ya que presenta una completa colección orientada a objetos de tipos reutilizables. Estos se pueden emplear para desarrollar aplicaciones que abarcan desde las tradicionales herramientas de interfaz gráfica de usuario (GUI) o de línea de comandos, hasta las aplicaciones basadas en las innovaciones más recientes proporcionadas por ASP.NET. Como los formularios Web Forms y los servicios Web XML.

## Entity Framework

Entity Framework (Corporation 2013) permite a los desarrolladores crear aplicaciones de acceso a datos, programando con un modelo de aplicaciones conceptuales en lugar de programar directamente con un esquema de almacenamiento relacional. El objetivo es reducir la cantidad de código y el mantenimiento necesarios para las aplicaciones orientadas a datos. A continuación se muestran algunas de las ventajas que ofrecen las aplicaciones de Entity Framework.

Estas pueden funcionar en términos de un modelo conceptual más centrado en la aplicación, que incluye tipos con herencia, miembros complejos y relaciones. Se encuentran libres de dependencias de codificación rígida. Las asignaciones entre el modelo conceptual y el esquema específico de almacenamiento pueden cambiar sin tener que cambiar el código de la aplicación.

Los desarrolladores pueden trabajar con un modelo de objeto de aplicación coherente que se puede asignar a diversos esquemas de almacenamiento, posiblemente implementados en sistemas de administración de base de datos diferentes. Se pueden asignar varios modelos conceptuales a un único esquema de almacenamiento. La compatibilidad con Language Integrated Query (LINQ) proporciona validación de la sintaxis en el momento de la compilación para consultas en un modelo conceptual. Entity Framework se presentó primero como parte de .NET Framework y de Microsoft Visual Studio. A partir del lanzamiento de la versión 4.1 de Entity Framework, Entity Framework no forma parte de .NET sino que está integrado en .NET.

## LINQ

Según (Microsoft) Language-Integrated Query (LINQ) es un conjunto de características presentado en Visual Studio que agrega capacidades de consulta eficaces a la sintaxis de los lenguajes C# y Visual Basic. LINQ incluye patrones estándar y de fácil aprendizaje para consultar y actualizar datos, y su tecnología se puede extender para utilizar potencialmente cualquier tipo de almacén de datos. Visual Studio incluye ensamblados de proveedores para LINQ que habilitan el uso de LINQ con colecciones de .NET Framework, bases de datos SQL Server, conjuntos de datos de ADO.NET y documentos XML.

*Para los autores de este trabajo: Una de las partes más importantes de este trabajo referente a tecnología y herramientas a utilizar; es la plataforma de desarrollo ya que esta es la que permite la confección del sistema que dará solución a los problemas existentes. En este caso se utiliza Microsoft .NET Framework 4.0 ya que es una plataforma que cumple con todos los requerimientos necesarios para el desarrollo del módulo ya que es la plataforma donde se ha desarrollado todos los módulos correspondientes al SUIN. Contiene además Entity Framework que en este trabajo es de gran importancia ya que permite crear*

aplicaciones de acceso a datos. Este es compatible con LINQ proporcionando la validación de la sintaxis en el momento de la compilación para consultas en un modelo conceptual.

### 1.3.5 Entorno de Desarrollo Integrado (IDE<sup>7</sup>, por sus siglas en inglés)

Un entorno de desarrollo integrado (Jesús M. González Barahona 2003-2007) conocido también por sus siglas en inglés como IDE es un programa informático el cual está compuesto por un conjunto de herramientas de programación. Facilitando a los desarrolladores de software el trabajo para la creación de los mismos. Integra la edición orientada al lenguaje, la compilación o interpretación, las medidas de rendimiento, la incorporación de las fuentes a sistemas de control de fuentes.

#### Microsoft Visual Studio 2010

Microsoft Visual Studio 2010 (Corporation 2013) es un conjunto de herramientas que simplifica el desarrollo de aplicaciones para desarrolladores individuales o equipos, haciendo posible la creación de aplicaciones escalables y de alta calidad. Proporciona la capacidad de optimizar su entorno de desarrollo con compatibilidad para varios monitores y permite soportar varias versiones de .NET Framework con una sola herramienta.

Presenta novedosas herramientas de pruebas que automatizan la realización de pruebas de la interfaz de usuario en aplicaciones basadas en web, así como pruebas manuales, *Pruebas Profesionales*, pruebas de rendimiento de web, pruebas de carga, cobertura de código, generación de datos de pruebas; además incluye nuevas características de depuración y diagnóstico.

Incluye una versión renovada del servidor *Team Foundation Server*, el cual ofrece un gestor de pruebas mejorado y nuevas herramientas para el servidor; además de elementos nuevos como *Lab Management*, que permite crear y gestionar entornos virtuales.

*Para los autores de este trabajo: Se utiliza Visual Studio 2010 por ser uno de los IDE más completos que existe para el trabajo con el lenguaje C#. Es una herramienta que soporta varias versiones de .NET en ella misma. Además ofrece diversas funcionalidades importantes para el desarrollo de software dentro de ellas se encuentran: conjunción con XAML (Extensible Applications Markup Language por sus siglas en inglés); un diseñador para Windows Presentation Foundation y Workflow Foundation que son parte del Framework .NET 4.0; IntelliSense para JavaScript; el nuevo Lenguaje LINQ siendo este un agregado a los lenguajes Visual Basic y Visual C#.*

---

<sup>7</sup> Integrated Development Environment

## Team Foundation Server 2010

*Microsoft Visual Studio Team Foundation Server 2010* (Microsoft 2013) es la plataforma de colaboración central en la solución de administración del ciclo de vida de las aplicaciones de *Microsoft*. Automatiza el proceso de entrega de software y permite a las organizaciones administrar proyectos de desarrollo de software de forma eficiente durante su ciclo de vida. *Team Foundation Server 2010* permite al equipo ser más ágil, colaborar de un modo más eficaz y ofrecer software con mayor calidad, así como adquirir y compartir conocimiento institucional.

*Para los autores de este trabajo: Esta herramienta mejora la calidad y la confiabilidad del proceso de desarrollo para el equipo de desarrollo de software. Permite al equipo unir el ciclo de vida del software con compatibilidad integrada para el control de código fuente y elementos de trabajo. Además posibilita la administración de compilaciones, informes detallados que muestran tendencia de errores, velocidad del proyecto e indicadores de calidad.*

### 1.3.6 Lenguaje de programación

Un lenguaje de programación no es más que un lenguaje que permite expresar acciones que se pueden llevar a cabo por las máquinas. Posibilitando la creación de sistemas de cómputo que puedan controlar el comportamiento de las mismas. En la actualidad existen diversos lenguajes cada uno con características diferentes que permiten identificarse del resto. Estos permiten especificar de una manera precisa con que datos debe operar una computadora. Además de poder especificar como serán almacenados los datos y que acciones debe realizar ante diversas circunstancias.

#### C Sharp (C#)

C# (2000-2013) es un lenguaje orientado a objetos,<sup>8</sup> diseñado por Microsoft para ser utilizado en la plataforma .NET. Permitiendo la creación de aplicaciones clientes de windows tradicionales, servicios web XML, componentes distribuidos, aplicaciones cliente-servidor y aplicaciones de base de datos.

Como lenguaje orientado a objetos admite los conceptos de encapsulación, herencia y polimorfismo. Todas las variables y métodos, incluido el método main que es el punto de entrada de la aplicación, se encapsulan dentro de definiciones de clase. La sintaxis de C# basada en signos de llave es muy expresiva, pero también es sencilla y fácil de aprender.

C# ofrece diversas ventajas frente a otros lenguajes dentro de las que se encuentran:

---

<sup>8</sup> Se basa en la idea de un objeto, que es una combinación de variables locales y procedimientos llamados métodos que juntos conforman una entidad de programación.

- Declaraciones en el espacio de nombres: Al empezar a programar algo, se puede definir una o más clases dentro de un mismo espacio de nombres.
- Tipos de datos: Existe un rango más amplio y definido de tipos de datos que los que se encuentran en C, C++ o Java.
- Atributos: Cada miembro de una clase tiene un atributo de acceso del tipo público, protegido, interno, interno protegido y privado.
- Pase de parámetros: Aquí se puede declarar a los métodos para que acepten un número variable de parámetros. De forma predeterminada, el pase de parámetros es por valor, a menos que se use la palabra reservada “ref”, la cual indica que se pasa por referencia.
- Métodos virtuales y redefiniciones: Antes de que un método pueda ser redefinido en una clase base, debe declararse como virtual.
- Propiedades: Un objeto tiene intrínsecamente propiedades, y debido a que las clases en C# pueden ser utilizadas como objetos, se permite la declaración de propiedades dentro de cualquier clase.
- Inicializador: Un inicializador es como una propiedad, con la diferencia de que en lugar de un nombre de propiedad, un valor de índice entre corchetes se utiliza en forma anónima para hacer referencia al miembro de una clase.
- Control de versiones: Permite mantener múltiples versiones de clases en forma binaria, colocándolas en diferentes espacios de nombres. Esto permite que versiones nuevas y anteriores de software puedan ejecutarse en forma simultánea.

*Para los autores de este trabajo: Este lenguaje es utilizado para el desarrollo del módulo debido a que permite una mejor integración con el SUIN debido a que se utilizan .Net. C# es un lenguaje ideal para desarrollar aplicaciones en .Net debido a la buena integración con el entorno de desarrollo integrado Visual Studio.*

## **ASP. NET**

Según los libros (MacDonald) y (Narváez) ASP.NET es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es utilizado para la construcción de sitios web dinámicos, aplicaciones web y servicios web XML con el código mínimo. ASP.NET pertenece a .NET Framework y al codificar las aplicaciones ASP.NET se tiene acceso a las clases en .NET Framework.

El código de las aplicaciones puede escribirse en cualquier lenguaje compatible con el Common Language Runtime (CLR), entre ellos Microsoft Visual Basic, C#, JScript.NET y J#. Estos lenguajes permiten desarrollar aplicaciones ASP.NET que se benefician del Common Language Runtime, seguridad de tipos, herencia, entre otros.

ASP.NET incluye:

- Marco de trabajo de página y controles.

- Compilador de ASP.NET.
- Infraestructura de seguridad.
- Funciones de administración de estado.
- Configuración de la aplicación.
- Supervisión de estado y características de rendimiento.
- Capacidad de depuración.
- Marco de trabajo de servicios Web XML.
- Entorno de host extensible y administración del ciclo de vida de las aplicaciones.
- Entorno de diseñador extensible.

*Para los autores de este trabajo: ASP.NET reduce la cantidad de código para la creación de aplicaciones de gran complejidad. Mejora el rendimiento mediante el aprovechamiento de enlace en tiempo, justo a tiempo de compilación, optimización nativa y los servicios de almacenamiento en caché a la derecha de la caja. Permite una fácil implementación debido a que no se necesita el registro de componentes ya que incorpora la información de configuración.*

### **1.3.7 Gestor de base de datos**

En la actualidad los sistemas informáticos manejan un gran volumen de datos (Cuéllar 2009), los cuales se necesitan almacenarlos para luego realizar diferentes operaciones con ellos. Estos pueden ser almacenados en diferentes contenedores que brindan la posibilidad de insertar, actualizar y eliminar datos en los mismos. Realizando estas acciones al menor costo y rapidez posible. Una base de datos no es más que una colección de información organizada de forma tal que un software pueda seleccionar las fracciones de datos que necesite. Un sistema gestor de base de datos es un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

#### **Oracle Data Base 11g**

Según (Oracle), (Alapati 2009), (Alapati 2008) oracle permite a las empresas tener un control de su información, garantizando una mejor vista interna del negocio y una adaptación de manera rápida y confiable al entorno competitivo en constante cambio en la actualidad. En esta versión se amplía las capacidades de oracle de agrupación de base de datos, automatización de centro de datos y gestión de carga de trabajo. Los clientes de oracle pueden hacer frente al procesamiento de transacciones, almacenamiento de datos y aplicaciones de gestión de contenido, más exigentes.

Oracle *Database* 11g es el primer gestor de base de datos que incluye funcionalidades que permiten hacer pruebas de cambios en aplicaciones. Simulando las cargas reales generadas por los usuarios en los



entornos de producción. Brinda a los desarrolladores la posibilidad de elegir herramientas de desarrollo, y un mejor proceso de desarrollo de aplicaciones que aprovecha al máximo las funciones clave de este. Esta versión incluye nuevas funcionalidades como: almacenamiento en caché del lado del cliente, XML binario para un desempeño de aplicaciones más rápido, procesamiento XML y el almacenamiento y recuperación de archivos. Incluye integración nativa con Visual Studio 2005 para desarrollar aplicaciones .NET sobre oracle; herramientas de migración de acceso con Oracle Application Express<sup>9</sup>; y creación de consulta fácil SQL Developer para una codificación veloz de las rutinas SQL y PL/SQL. Se ha incorporado soporte para XML binario, lo cual hace posible que los clientes puedan elegir opciones de almacenamiento XML para cumplir con los requerimientos específicos de desempeño y de aplicaciones.

*Para los autores de este trabajo: En el desarrollo del módulo se utilizó como gestor de base de datos Oracle 11g debido a la gran cantidad de funcionalidades que presenta. Esto tiene como ventaja facilidad de uso para los desarrolladores. Brinda una gran variedad de herramientas y la posibilidad de la elección de las mismas, lo que permite que se aproveche al máximo las funcionalidades claves que este posee.*

## Conclusiones Parciales

Después de realizado el estudio a los principales conceptos relacionados con la gestión de trazas en el servidor, así como el análisis a las soluciones existentes tanto en el ámbito nacional como internacional. Se detectaron diferentes características que dificultan el empleo de estas en el desarrollo de la solución a la situación problemática planteada. Por lo que se hace necesario el desarrollo del módulo de gestión de trazas en el servidor.

Seguido se realizó un estudio a la metodología, herramientas y tecnologías, lo que permitió profundizar en los principales elementos y aspectos novedosos a tener en cuenta para la implementación de la solución. Permitiendo el diseño de prototipos de interfaz de usuarios agradable y de fácil interacción. La selección de estas herramientas posibilitó una estructuración del módulo en correspondencia con lo especificado en el proyecto. Logrando una fácil integración con el módulo de administración del SUIN.

---

<sup>9</sup> Oracle Application Express o APEX (anteriormente llamado HTML DB) es una herramienta de desarrollo rápido de aplicaciones que se ejecuta con una base de datos Oracle.



## Capítulo 2: Propuesta de Solución

En el presente capítulo se realizará un estudio detallado de las causas que originan la situación problemática; así como una descripción detallada de las características del sistema. Se especifican los requisitos funcionales y no funcionales que guiarán el desarrollo del módulo de configuración. Especificando la arquitectura que se utilizará para la implementación del mismo. Se definen los patrones de diseño que se utilizarán en la implementación del sistema. Además de realizar una descripción del diagrama de clases y del modelos de datos.

### 2.1 Causas que originan la situación problemática

La problemática de este trabajo surge a raíz de la necesidad de tener un control de las trazas técnicas que se generan en el servidor de despliegue del SUIN. Una traza técnica no es más que un fichero físico que se encuentra en los servidores a los cuales, actualmente después de implementado dicho sistema no se tiene acceso. Teniendo en cuenta que son generados por n aplicaciones creando cada una sus propias trazas, lo que provoca que se torne complejo y engorroso su análisis. Al ser generadas diariamente por diferentes módulos, crea que esta información se torne incoherente y eventualmente inconsistente.

El formato de estos ficheros varía según la aplicación que lo genera. Para los desarrolladores del SUIN es de vital importancia conocer lo que sucede en las aplicaciones, ya que en ellas se almacena información de gran valor para el correcto funcionamiento del sistema. Alguna de la información que se almacena es referente al acceso a datos, errores que se producen en el código, errores de permiso para acceder a ficheros y/o errores que pueden ocurrir al tratar de acceder al sistema.

En la actualidad los sistemas para la gestión de trazas en el servidor son diversos, pero ninguno satisface las necesidades del SUIN (ver capítulo anterior). Entre los software existentes en el MININT no existe ninguno que cumpla con lo requerido para la gestión de trazas en el servidor. Por lo que genera la situación problemática: El Sistema Único de Identificación Nacional de la República de Cuba presenta dificultades para supervisar, controlar y analizar el sistema de trazas generadas en el servidor de despliegue. A raíz de esta problemática se realiza la siguiente propuesta de solución.

### 2.2 Propuesta de sistema

Después de realizado el análisis anterior, se realiza la propuesta de un sistema que resuelva la situación existente en la actualidad. Este sistema permite realizar la gestión de trazas del servidor de despliegue del SUIN necesitando tres elementos fundamentales:

- Una base de datos que permita el almacenamiento de todas las trazas con que se trabajará.
- Un servicio de Windows que permita cargar las trazas y almacenarla en la base de datos.

- Una aplicación web que permita mostrar toda la información analizada.

Después de haber especificado los elementos que componen el futuro sistema que permita la gestión de trazas. Se presenta el funcionamiento del mismo, donde se realiza una descripción detallada de todas las actividades que realizará el sistema.

## **Propuesta de Solución**

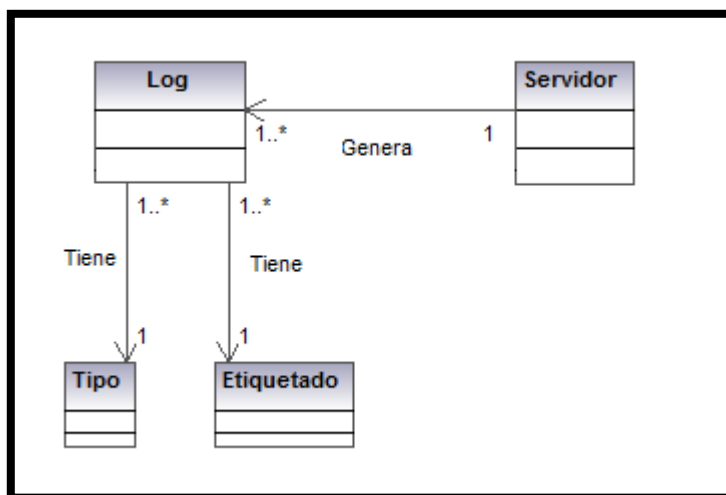
Los logs se encuentran como ficheros físicos almacenados en el servidor. Para ello se necesita un servicio de Windows que los capture y los procese, de forma tal que permita identificar cada una de sus trazas. Este a su vez realiza el procesamiento y transformación de ficheros, envía los datos a un servicio de Windows communication foundation. El cuál es el responsable del procesamiento y etiquetado de la información, garantizando que esta llegue en el formato correcto al usuario. Una vez procesada la información se procede a realizar la salva en la base de datos. Teniéndose en cuenta que se va a realizar la salva de un gran volumen de información. Después de almacenada la información el propio servicio de Windows realiza consultas a la base de datos, brindándole la información a la aplicación web. Donde se expone las mismas trazas capturadas de los ficheros que se encuentran en el servidor pero esta vez de una forma más cómoda. A través del uso de filtros, gráficas, paginado y otros elementos más prácticos para el consumo del usuario. Señalando que la aplicación web va a manejar un gran volumen de información.

## **Funcionamiento del Sistema**

El sistema permitirá al administrador cargar las trazas a partir de un intervalo de fecha definido. El mismo brinda la opción de realizar un filtrado flexible, tanto por columnas como por cualquier elemento que el usuario introduzca en el campo de filtro. Otro método de filtrado que brinda la aplicación web es limpiar los datos, esta opción permite ocultar toda la información que no sea necesaria a partir de los datos definidos por el usuario. Cuenta también con un paginado que permite desplazarse por toda la tabla y con un selector de filas cuya función es mostrar la cantidad de filas según con la que se desea trabajar.

La tabla está compuesta por una columna de operaciones en la cual se encuentra las funciones etiquetar y eliminar. La operación etiquetar permite al usuario definir una etiqueta a los logs. Este especifica el valor de la etiqueta y un conjunto de palabras a partir de las cuales se va a buscar en los mensajes aquellos que contengan estas palabras y se les aplicará el etiquetado de manera automática. Se tendrá en cuenta también estas etiquetas definidas para cuando se vuelva a realizar la carga de los nuevos logs, verificando si alguno cumple con aquellas etiquetas que ya existen y se les asocia de manera automática. La operación eliminar brinda la posibilidad de eliminar todas aquellas trazas que se consideren innecesarias en el análisis que se está realizando.

A continuación se realiza una representación de los conceptos más significativos asociados al dominio del problema y las relaciones existentes entre los mismos. Esta representación es conocida como **Modelo de Dominio**. En la figura 2.2 se muestra una representación gráfica del modelo, y luego se explican cada uno de sus conceptos.



**Figura 2.2:** Modelo de dominio.

### Descripción de las entidades del modelo conceptual

**Servidor:** Un servidor abastece a varios clientes, evitando que cada uno de estos, tenga la información almacenada e instalada localmente en su computadora.

**Log:** Fichero físico que se encuentra almacenado en los servidores.

**Tipo:** Clasificación de las trazas según son generadas.

**Etiquetado:** Descripción del log.

### Descripción de rol

En la tabla a continuación se define los roles que trabajarán con el módulo implementado. Además de la función que desempeñan cada uno de ellos.

Rol	Descripción
Administrador Nacional del Sistema	<ul style="list-style-type: none"> <li>Garantiza el control de los procesos de negocios ejecutados en el SUIN a nivel nacional.</li> <li>Ejecuta diferentes acciones en el sistema tales como: monitorear los procesos ejecutados en las oficinas de identificación y registro a nivel nacional, monitorización general de los procesos así como la representación de gráficos.</li> </ul>

Administrador de las Trazas	<ul style="list-style-type: none"> <li>• Establece el etiquetado a las trazas que lo necesiten.</li> <li>• Ejecuta diferentes acciones en el sistema tales como: monitorear las trazas que son generada por las diferentes aplicaciones en el servidor.</li> </ul>
-----------------------------	--

Tabla 2.1: Descripción de roles.

## 2.3 Definición de requisitos funcionales

En el siguiente epígrafe se realiza la definición y especificación de cada uno de los requisitos necesarios para el desarrollo de la aplicación. Permitiendo obtener una clara comprensión del problema a resolver.

### 2.3.1 Especificación de requisitos funcionales

Los requisitos funcionales constituyen las capacidades o condiciones que el sistema debe tener, o sea, las utilidades expuestas por la solución para los usuarios, los cuales deben redactarse en el lenguaje del cliente. A continuación se enumeran las funcionalidades resumidas:

RF1. Cargar Trazas.

RF2. Graficar Errores por Día.

RF3 Filtrar Información.

RF4. Eliminar Basura.

RF5. Establecer Etiquetado.

Además de las funcionalidades que componen a la aplicación web, se encuentran las funcionalidades que contiene el servicio de Windows. El cual permite cargar los logs y almacenarlos en la base de datos. A continuación se especifican dichos requerimientos.

RF6. Establecer Configuración del Servicio.

RF7. Cargar Ficheros.

RF8. Generar Logs.

### 2.3.2 Descripción de requisitos funcionales

A continuación se muestra una descripción del requisito funcional “Cargar Trazas”.

<b>Propósito:</b>	Permite cargar las trazas generadas en el servidor.
<b>Roles:</b>	Administrador Nacional del Sistema.
<b>Precondiciones:</b>	1. El usuario debe estar autenticado en el sistema.
<b>Entidades tratadas:</b>	No Procede.


<b>Descripción:</b>	<p>1.1. Seleccionar opción “Trazas del Servidor”.</p> <p>1.2. Mostrar interfaz para seleccionar intervalo de fecha de carga.</p> <p>a) Fecha de Inicio.</p> <p>b) Fecha Fin.</p> <p>1.3. Mostrar la opción “Cargar”</p> <p>1.3.1. Realizar la carga de las trazas si se selecciona la opción “Cargar”.</p> <p>1.3.2. Mostrar listado de las trazas en formato tabular y las funcionalidades para el trabajo con estas.</p>
<b>Validaciones:</b>	<p>Para realizar las operaciones:</p> <p>1. Ver las validaciones de los conceptos tratados en el Diccionario de datos.</p>
<b>Poscondiciones:</b>	<p>1. Se obtiene el resultado de la carga.</p>
<b>Prototipo:</b>	 <p>The screenshot shows a web interface for loading logs. At the top, there are two date pickers labeled 'Fecha inicio:' and 'Fecha fin:' with dates '10/24/2012' and '10/28/2012' respectively. Below them is a button labeled 'Cargar'. Underneath is a table titled 'Tabla 1: Trazas del servidor'. The table has columns: 'DateTime', 'Index', 'Type', 'Category', 'Instance id', 'Message', 'Etiqueta', and 'Operaciones'. The first row of data shows a timestamp '2012-10-27 10:00:45', index '[160]', type 'Warn', category 'Pres', and a message 'El recurso suin_proceso_estado_error se encuentra repetido'. At the bottom of the table, it says 'Entradas del (0) - (1) de 9'.</p> <p><b>Figura 2 Cargar Trazas.</b></p>

Tabla 2.2:

Descripción RF “Cargar Trazas”.

Para visualizar las descripciones de los restantes requisitos funcionales debe remitirse al Anexo 1.

## 2.4 Requisitos no funcionales

Los requisitos no funcionales especifican las propiedades o cualidades que debe tener la solución a desarrollar. Representan las características que hacen al producto atractivo, usable, rápido o confiable. Estos requisitos pueden marcar la diferencia entre un producto bien aceptado y otro con poca aceptación. A continuación se listan estas características:

### Requerimientos de Usabilidad

RNF1. Podrá ser usado por personas con las siguientes características:

- Conocimientos básicos en el manejo de computadoras.
- Conocimiento básico en el sistema operativo Windows.

RNF2. Tendrá siempre visible la opción de Ayuda, lo que posibilitará un mejor aprovechamiento por parte de los usuarios de sus funcionalidades.

RNF3. Será distribuido en idioma español.

RNF4. Contará con menús laterales y desplegables que permitan un acceso rápido a la información.

### **Requerimientos de Confiabilidad**

RNF5. La información manejada estará protegida de acceso no autorizado y de divulgación.

RNF6. Deben montarse sistemas de respaldo eléctrico para mantener la vitalidad de los servicios.

### **Requerimientos de Apariencia o Interfaz Externa**

RNF7. Interfaz consistente con las pautas de diseño definidas para los sistemas del proyecto.

### **Restricciones de Diseño**

RNF8. El sistema debe implementarse usando el lenguaje C#, sobre la plataforma ASP.NET.

RNF9. El sistema gestor de bases de datos, será Oracle 11g.

RNF10. El sistema debe desarrollarse usando el IDE Visual Studio 2010.

RNF11. Se utilizará el Team Explorer para el control de código fuente.

RNF12. Requisitos para la documentación de usuarios en línea y ayuda del sistema.

- Debe poseer documentos técnicos.
- Debe poseer el manual de usuario.
- Debe poseer el manual de administración y configuración del sistema.

### **Requerimientos Políticos- Culturales**

RNF13. Debe respetar los términos empleados normalmente, por los especialistas en el tema de la esfera que se automatiza.

### **Requerimientos Legales**

RNF14. Se garantizará la protección de los datos, privacidad y derecho a la información.

### **Requerimientos de Seguridad**

RNF15. Los usuarios deberán autenticarse para entrar al sistema.

RNF16. Cuando se intente realizar cualquier acción irreversible, existirá una opción de advertencia antes realizar dicha acción.

RNF17. El sistema registrará todas las acciones que se realizan.

RNF18. La conexión a los servidores de bases de datos y de aplicaciones es requerida para el correcto funcionamiento.

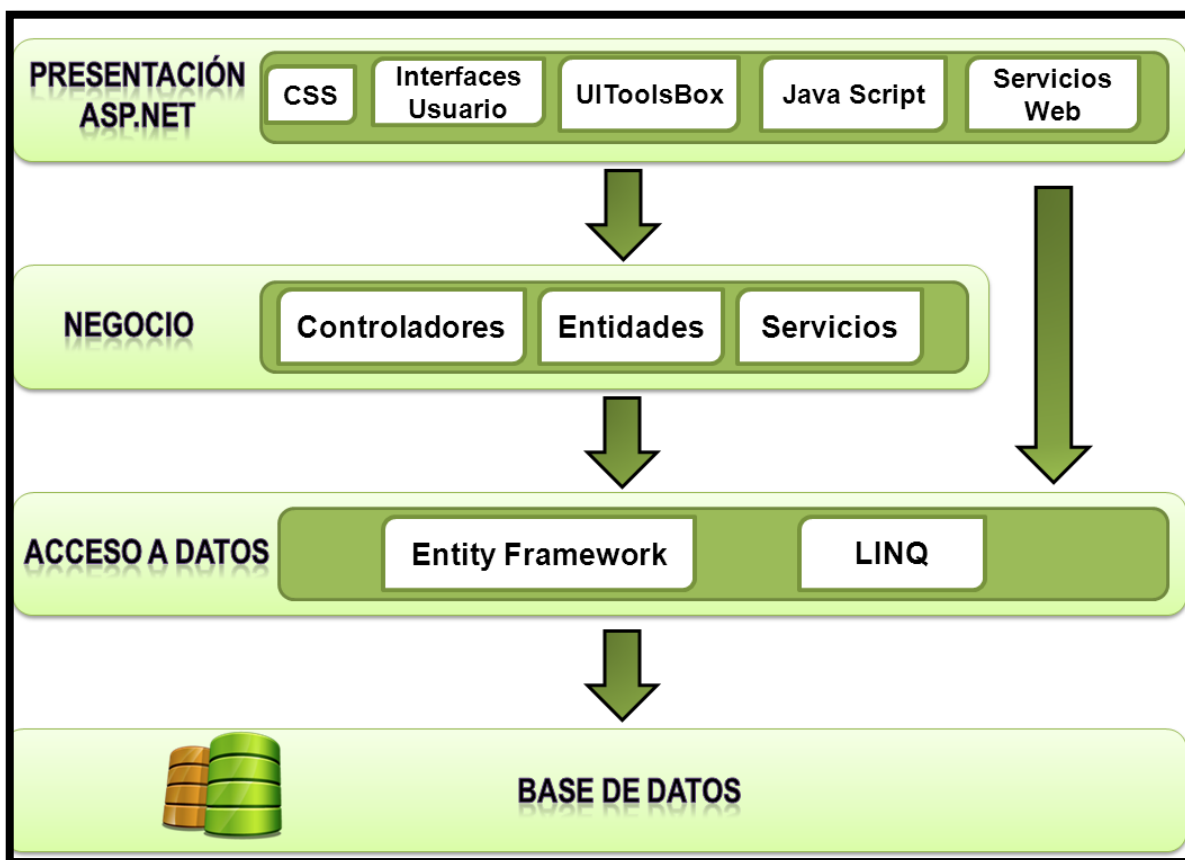
### **Requerimientos de Software**

RNF19. Sistema Operativo: Windows.

## **2.5 Arquitectura de software**

Este módulo se desarrollará a partir de la arquitectura n capas. La cual contará con 4 capas lógicas que proporcionan un alto nivel de encapsulamiento de las responsabilidades. Permitiendo reducir al máximo el acoplamiento y aumentar la reutilización de las mismas (César de la Torre Llorente 2010).

La distribución de estas capas permite realizar grandes cambios en cada una individualmente, sin tener que afectar a las restantes. Una vez que estén bien definidas, la comunicación entre ellas se realizaría solo a nivel de interfaz lo cual permite trabajar de manera transparente a las instancias reales. En la siguiente figura se muestra la arquitectura definida para el módulo de gestión de trazas en el servidor de despliegue del SUIN.



**Figura 2.3:** Arquitectura del Módulo de Gestión de Trazas en el Servidor de Despliegue del SUIN.

**Presentación:** Está compuesta por todas las interfaces de usuario y los componentes necesarios para su correcto funcionamiento. Estos elementos pueden ser ficheros JavaScript, CSS, servicio web, interfaces de usuario, etc. Esta capa se encuentra representada por el proyecto web de la aplicación y tiene interacción directa con la capa de negocio.

**UIToolsBox:** El UIToolsBox es una biblioteca de componentes de interfaz de usuario que rige el modo en que se visualizan los controles en el navegador basado en las siguientes características:

- Define los controles básicos basándose en los controles de ASP.NET para mantener el comportamiento de las aplicaciones ASP.NET.
- Rige el comportamiento de las interfaces de usuario a partir de una capa basada en JQuery que le da un comportamiento más dinámico a los controles utilizados.
- Garantiza peticiones asíncronas al servidor a través de servicios Web que son consumidos desde JavaScript.

**Negocio:** En esta capa se recogen todos los controladores necesarios para darle solución a los requerimientos de negocio. Los controladores se encuentran definidos según el contexto en el que se



desenvuelven y tienen la responsabilidad de manejar todas las operaciones sobre una entidad de negocio en específico, así como todas las entidades que por conceptos de composición se encuentran relacionadas con esta. Por cada entidad de negocio se crea un controlador y una interfaz que debe ser implementada por el acceso a datos que le dará soporte. Además en esta capa se encuentran definidos los servicios que utiliza la aplicación para llevar a cabo el procesamiento, transformación y almacenamiento de los datos.

**Acceso a Datos:** La capa de acceso a datos está directamente relacionada con los controladores definidos en el negocio. Para establecer esta relación se hace uso de la interfaces de conectores y de la fábrica de conectores que define la capa de negocio. De esta manera es posible realizar cambios en esta capa sin que se vean afectadas las demás. Su principal función es realizar una implementación de las interfaces definidas en la capa de negocio y al mismo tiempo trabajar directamente con las fuentes de datos establecidas.

**Base de Datos:** Por la importancia que posee el desarrollo de procedimientos y vistas dentro del negocio, se ha decidido separar la base de datos en una nueva capa donde se concentra una pequeña parte de la lógica de las funcionalidades dentro de la aplicación. La única capa encargada de interactuar con esta es la de Acceso a Datos.

## 2.6 Patrones de diseño

Un patrón de diseño (Pressman 2006) describe una estructura de diseño que resuelve un problema de diseño particular dentro de un contexto específico y en medio de fuerzas que pueden tener un impacto en la manera en que se aplica y utiliza el patrón. La finalidad de cada patrón de diseño es proporcionar una descripción que le permita al diseñador determinar si el patrón es aplicable al trabajo actual, si se puede reutilizar (por ende, ahorrar tiempo del diseño) y si puede servir como guía para desarrollar un patrón similar, pero diferente en cuanto a la funcionalidad o estructura

Los patrones de diseño pueden incrementar o disminuir la capacidad de comprensión de un diseño o de una implementación. Además de esto, es necesario tener en cuenta que hacer uso de patrones significa en muchos casos disminución en el rendimiento de la aplicación, por ello es necesario lograr un equilibrio entre la flexibilidad y el rendimiento dentro del sistema. Esto implica que no es razonable utilizar patrones hasta que no se tenga totalmente claro el dominio y el problema a tratar.

Para el desarrollo del módulo en cuestión se han tenido en cuenta un conjunto de patrones que permiten darle flexibilidad, además se identificaron un conjunto de patrones específicos para el desarrollo que brindan claridad y fortaleza a los diseños.

Durante el desarrollo del trabajo se utilizaron los siguientes patrones:

**Encapsulación:** Permite solo accesos estilizados al objeto. Se hace uso de este patrón en casi todas las clases que componen al sistema, permitiendo que estas solo posean como elementos públicos aquellos que son exclusivamente necesarios. Un ejemplo de esto se presenta en el siguiente fragmento de código.

```
public double Timerxml1
{
    get { return Timerxml; }
    set { Timerxml = value; }
}
```

Figura 2.4: Ejemplo del patrón encapsulación.

**Excepciones:** Propone introducir estructuras de lenguaje para arrojar e interceptar excepciones. Se identificaron diferentes tipos de errores a tratar dentro del sistema. Creando clases que permitan identificar cada tipo de error en el momento de ejecución.

**Subclase:** Propone heredar miembros por defecto de una superclase. Seleccionando la implementación correcta a través de resoluciones sobre qué implementación debe ser ejecutada. Se puede encontrar este patrón con más fuerza en las entidades de negocio, que por su conceptualización las funciones y la información que almacenan pueden estar diferenciadas en cierta medida.

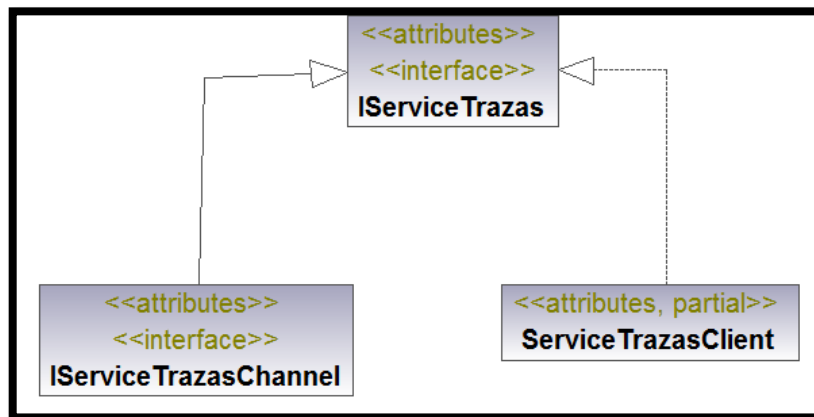


Figura 2.5: Ejemplo del patrón subclase.

Dentro de los patrones a utilizar se encuentran los patrones **Gang of Four** más conocidos como patrones **GoF**. En la arquitectura n capas hay cerca de cien patrones que son variantes del patrón básico de capas. Patrones de uso común relativos al estilo son Facade, Adapter, Bridge y Strategy. A continuación se realiza una explicación de cada patrón.

**Fachada** (Software): Proporciona una interfaz unificada de alto nivel. Representando a un subsistema, facilitando su uso. La fachada satisface a la mayoría de los clientes sin ocultar las funciones de menos nivel a aquellos que necesiten acceder a ellas. Se utiliza cuando se quiere proporcionar una interfaz simple para

un subsistema complejo, cuando se quiere estructurar en capas el sistema (donde cada capa tiene su fachada) y cuando existen muchas dependencias entre los clientes y las clases que implementan una abstracción. Una “fachada” proporciona al subsistema independencia y portabilidad. La siguiente figura muestra el uso de este patrón en la solución.

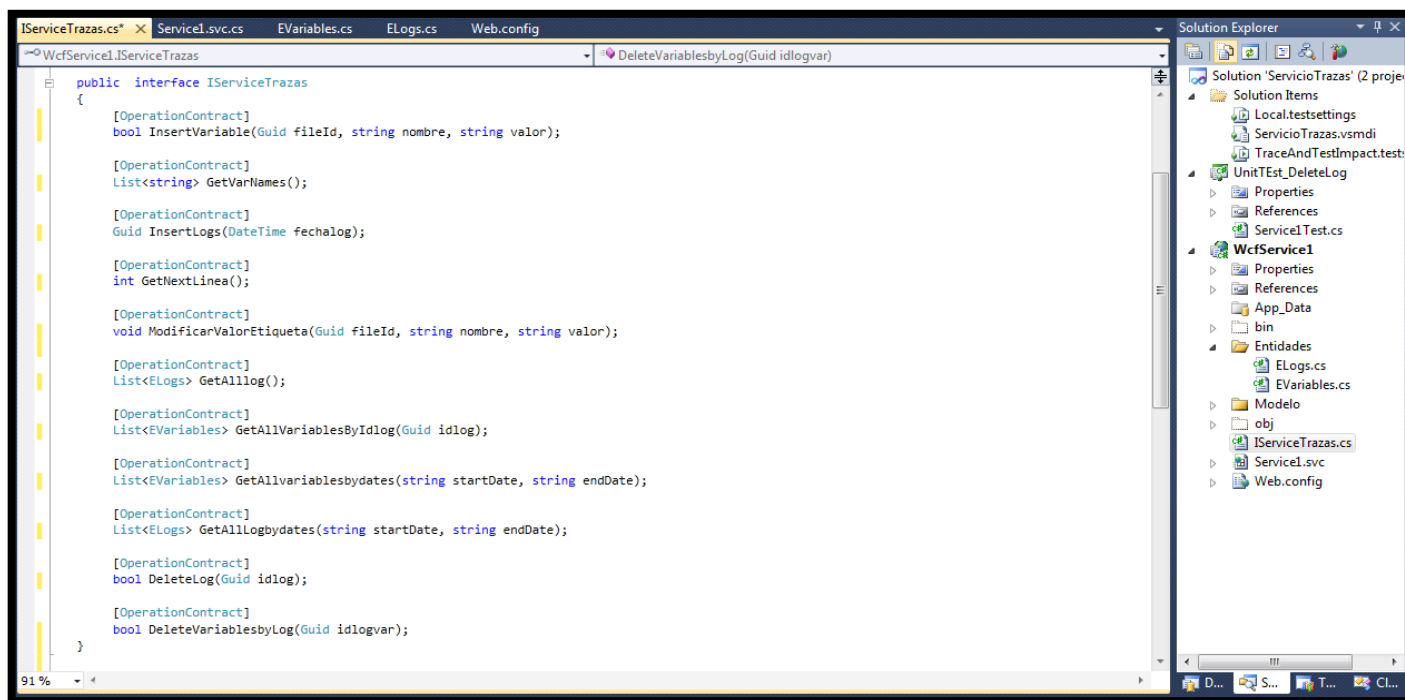


Figura 2.6: Ejemplo del uso del patrón fachada.

**Singleton (Software):** Este patrón es uno de los más conocidos y sencillos dentro del conjunto de patrones GoF. Tiene como propósito garantizar que una clase sólo tiene una única instancia, facilitando un punto de acceso global a la misma. Su uso trae consigo una serie ventajas tales como:

- Acceso más controlado a la única instancia. Puede tener un control estricto sobre cómo y cuándo acceden los clientes a la instancia.
- Se reduce el espacio de nombres. Este es una mejora sobre las variables globales.
- Permite refinamientos en las operaciones y en la representación, facilitando una subclase de singleton.
- Es fácilmente modificable para permitir más de una instancia y, en general, para controlar el número de las mismas.

A continuación se muestra un ejemplo del uso del patrón singleton en modulo desarrollado.

```

public class Servicio : IServicioTrazas
{
    private static Modelo_Entidades contex;
    public static Modelo_Entidades Context
    {
        get
        {
            if (contex == null) { contex = new Modelo_Entidades(); }

            return contex;
        }
    }
}

```

**Figura 2.7:** Ejemplo del uso del patrón singleton.

**GRASP<sup>10</sup>:** El uso de este conjunto de patrones está totalmente unido a cada componente desarrollado en el módulo, donde cada uno de ellos posee sólo las funcionalidades acorde a las particularidades que lo caracterizan. Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Se aplican durante la elaboración de los diagramas de interacción, al asignar las responsabilidades a los objetos y al diseñar la colaboración entre ellos. Algunos de los patrones que describe GRASP son: *Experto*, *Creador*, *Alta Cohesión*, *Bajo Acoplamiento* y el patrón *Controlador*.(Larman 1999) A continuación se describen ejemplos del uso de algunos de estos patrones en el desarrollo de la aplicación.

El patrón *Creador* guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que debe conectar con el objeto producido en cualquier evento. Al escogerlo como creador, se da soporte al bajo acoplamiento. A continuación se muestra un ejemplo del empleo del patrón Creador en el módulo.

<sup>10</sup> Acrónimo que significa **General Responsibility Assignment Software Patterns** (patrones generales de software para asignar responsabilidades)

```

namespace LogParser.Controlador
{
    public class Controladora
    {
        public void Ejecutar_Proceso_Trazas()
        {
            ParseoXML();
            objestr.ProcesarEstructura();
            IList<Log> logs = objfuentlog.ObtenerLogs();
            var cliente = new ServiceTrazasClient();

            foreach (var log in logs)
            {
                log.ProcesarLog(objestr, Rutaxml);

                foreach (var item in log.trazas)
                {
                    try
                    {
                        var fecha = DateTime.Parse(item[fechalog].ToString());
                        var idlog = cliente.InsertLogs(fecha);
                        foreach (var variable in item)
                        {
                            cliente.InsertarVariable(idlog, variable.Key, variable.Value.ToString());
                        }
                    }
                    catch (Exception)
                    {
                        throw new Exception("Error insertando los logs en la base de datos");
                    }
                }
            }
        }
    }
}

```

Figura 2.7: Ejemplo del uso del patrón creador.

Un *Controlador* es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema. Define además el método de su operación. Un ejemplo de la aplicación del patrón controlador se muestra en la siguiente figura.

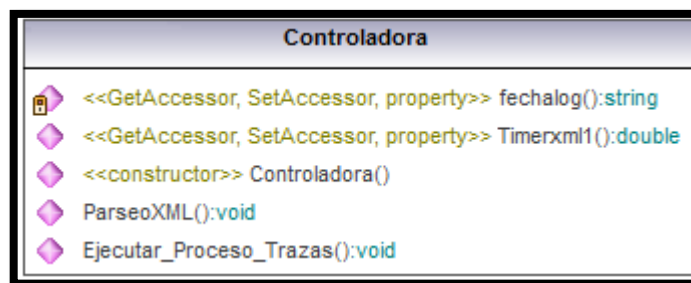
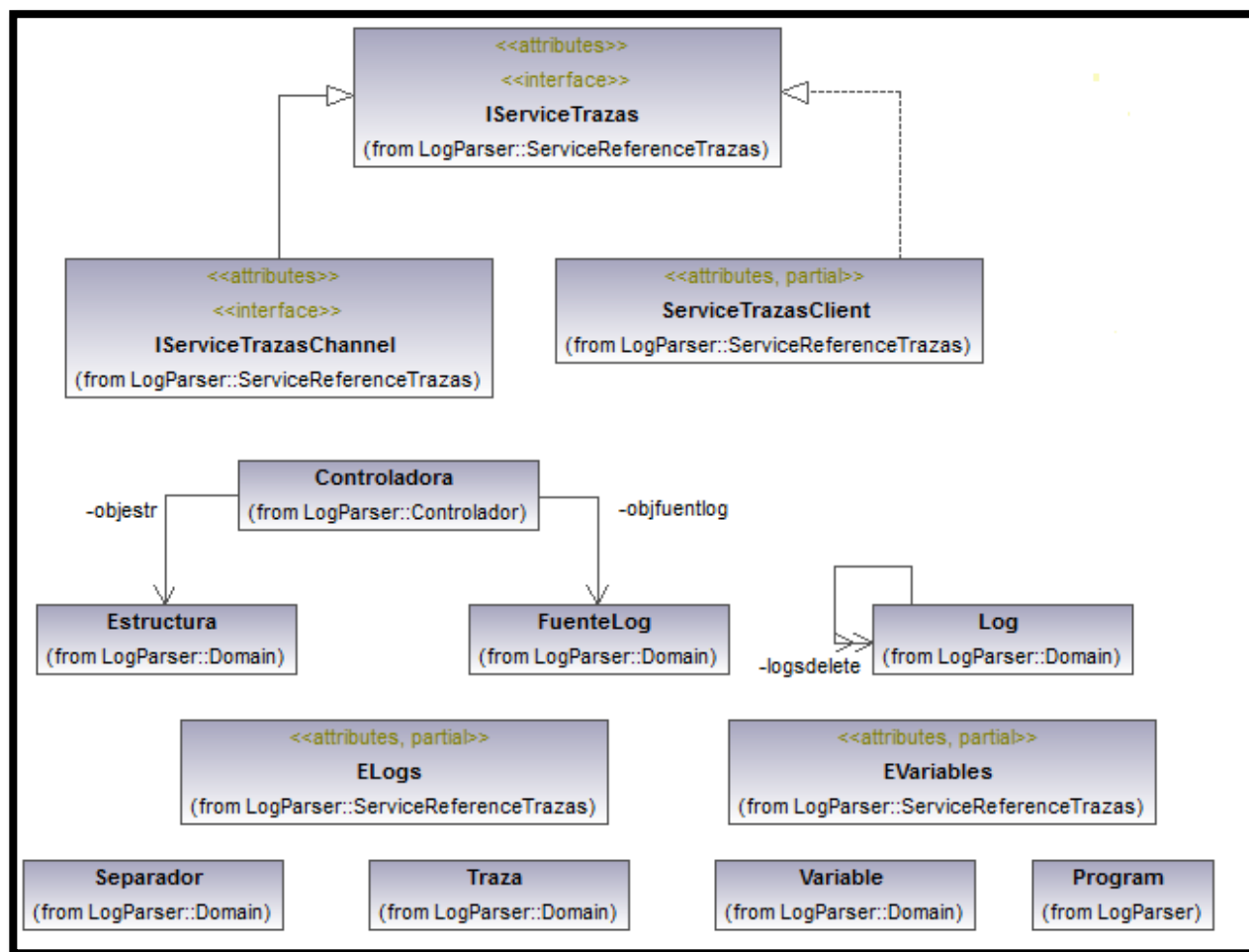


Figura 2.8: Ejemplo del uso del patrón controlador.

## 2.7 Diagrama de clases del diseño

El diagrama de clases del diseño es uno de los principales diagramas para el análisis y diseño. Este diagrama presenta las clases del sistema con sus relaciones estructurales y de herencia. La definición de una clase incluye definir sus atributos y operaciones (Roly Aruquipa Huaycho). A continuación en la figura

2.9 se muestra el diagrama de clases del servicio que utilizará el módulo para la carga de los logs y el almacenamiento en la base de datos.



**Figura 2.9:** Diagrama de clases del servicio de Windows.

## 2.8 Modelo de datos

Un modelo de datos es un lenguaje utilizado para la descripción de una base de datos. Por lo general, permite describir los elementos que intervienen en una realidad o problema dado y la forma en que se relacionan dichos elementos entre sí (Microsoft 2013). A continuación en la figura 2.10 se muestra el modelo de datos del módulo.

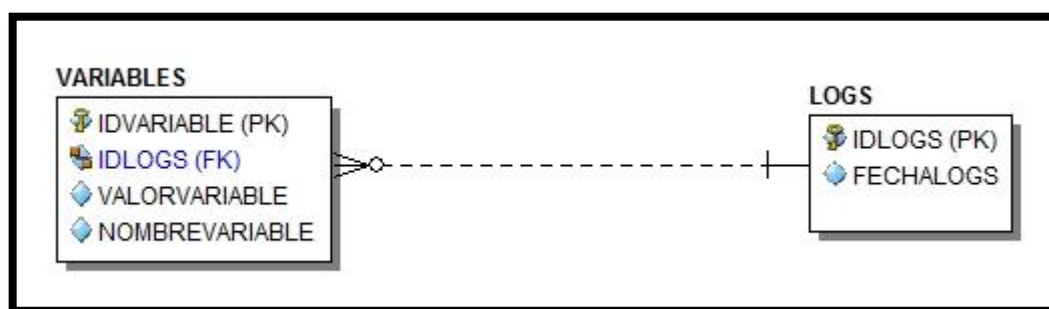


Figura 2.10: Modelo de datos.

## 2.9 Descripción del modelo de datos

En las tablas a continuación se muestra una descripción de cada uno de los atributos que componen el modelo de datos propuesto.

Nombre	Variables	
Descripción	Se encuentran almacenados los atributos referentes a las variables.	
Atributo	Tipo	Descripción
IDVARIABLE	RAW	Identificador de la variable.
IDLOGS	RAW	Identificador de un log.
VALORVARIABLE	VARCHAR	Valor que le corresponde a cada variable.
NOMBREVARIABLE	VARCHAR	Nombre de la variable

Tabla 2.3: Descripción de la tabla Variables.

Nombre	Logs	
Descripción	Se encuentran almacenados los atributo referente a los logs	
Atributo	Tipo	Descripción
IDLOGS	RAW	Identificador del log.
FECHALOGS	DATE	Fecha que toma el log cuando se genera en el servidor.

Tabla 2.4: Descripción de la tabla Logs.

## Conclusiones Parciales

Después de realizado un estudio detallado a los factores que dan origen a la situación problemática. Se demostró que en la actualidad no existe un procedimiento que permita realizar un análisis a las trazas que son generadas en el servidor de despliegue. Lo que hace necesario plantear un modelo de dominio que permita el desarrollo de la solución propuesta. Se definieron y especificaron los requerimientos tanto funcionales como no funcionales lo que permiten obtener una clara comprensión del problema a resolver. Se realizó la definición de la arquitectura a utilizar en la aplicación y del servicio de Windows permitiendo una guía para la implementación de la solución. Se realizaron los diagramas de clases del diseño aplicando

los patrones de diseño descritos así como la especificación de los diferentes tipos de clases utilizados. Además se determinó el modelo de datos a utilizar.



## Capítulo 3: Implementación y Pruebas

En el presente capítulo se traducen los elementos del diseño en elementos de implementación. Se describe cómo los elementos del modelo del diseño se implementan en términos de componentes. Definiéndose los diagramas de despliegue y componente del sistema. Se presentan algunos de los diseños de casos de pruebas que auxiliarán las pruebas de caja negra y caja blanca.

### 3.1 Estándares de codificación

La claridad en la implementación del código fuente influye directamente en lo que un programador comprende de un sistema. La legibilidad del código facilita que este pueda modificarse en el caso de realizar algún cambio en el funcionamiento del sistema implementado. Estableciendo un estándar de codificación garantiza que los programadores mantengan un código de calidad.

Un patrón de nomenclatura coherente es uno de los elementos más importantes de la previsibilidad y el descubrimiento de una biblioteca de clases administradas. El uso generalizado y la comprensión de estas directrices de denominación deberían eliminar muchas de las preguntas más comunes.

#### Estilos para la capitalización

Se contemplarán tres estilos de capitalización distintos, que se utilizarán en determinados casos (Ignasi Fosch Alonso 2006).

**Capitalización Pascal:** En este estilo se capitaliza la primera letra de cada palabra: ValorNulo.

**Capitalización Camel:** En este otro estilo se capitaliza la primera letra de cada palabra excepto la primera: valorNulo.

**Capitalización Versal:** Es aquella en la que todo se escribe en mayúsculas.

**Sensibilidad a mayúsculas:** Para evitar confusiones y garantizar la interoperabilidad entre lenguajes, se siguieron las siguientes reglas sobre el uso de mayúsculas y minúsculas:

- No utilizar *namespaces*<sup>11</sup> que solo se diferencien con en el uso de las mayúsculas.
- No utilizar funciones con nombres de parámetros que se diferencian solo en el uso de las mayúsculas.

---

<sup>11</sup> Define el nombre de un espacio de trabajo que puede ser un componente o proyecto.

- No utilizar *namespaces* con nombres de clases que se diferenciarán solo en el uso de las mayúsculas.
- No crear clases con propiedades que se diferencian solo en el uso de las mayúsculas.
- No crear clases con métodos que se diferencien solo en el uso de las mayúsculas (Cuba 2009).
- Para nombrar las clases no se recomienda usar los nombres de los *namespaces* y otras clases comúnmente usadas, por ejemplo *System*<sup>12</sup>, *Collections*<sup>13</sup> o *Forms*<sup>14</sup>.

**Evitando confusión de nombre y tipo:** En la actualidad diversos lenguajes usan diferentes términos para declarar los tipos de identificadores fundamentales. Los diseñadores de bibliotecas de clases deben establecer una terminología específica para el lenguaje. En este trabajo se utilizaron nombres que permitan obtener una descripción de sus identificadores, a diferencia de nombres que describan el tipo de identificador.

Con el fin de aumentar la legibilidad del código se utilizaron una serie de comentarios en todas las declaraciones de las clases y en las funciones más complejas. Conjuntamente se realizó una organización estructurada del código. Permitiendo una mejor lectura del mismo.

### 3.2 Tratamiento de errores

En el diseño y desarrollo de un sistema no se debe tener solamente en cuenta lo que debe ocurrir. Sino que se debe realizar un análisis profundo de las distintas situaciones que se puedan presentar y que conforman algún tipo de violación o de situación en particular. Provocando un error dentro del sistema.

Un elemento que forma parte del diseño son las excepciones, que aun cuando la mayoría de estas son identificadas en el proceso de implementación, se hace necesario definir un mecanismo para el tratamiento de ellas. En el desarrollo del análisis de los procesos para alcanzar una respuesta rápida y efectiva ante cualquier error se realiza un estudio de los errores. Que permite el lanzamiento de errores en un lenguaje de fácil comprensión para los usuarios finales. Para el tratamiento de estas excepciones se define una interfaz base que permite dejar una traza clara de todas las excepciones lanzadas dentro del sistema.

### 3.3 Implementación

Uno de los flujos de trabajo donde se desarrolla el sistema en términos de componentes, es la implementación. Estos pueden ser ejecutables, archivos de códigos fuentes, biblioteca de clases, script, entre otros. Es en este flujo donde se realiza el desarrollo de la arquitectura y en sí, el sistema como un todo. Donde se define y se aplican los patrones y estándares de codificación, con el objetivo de optimizar la

---

<sup>12</sup> Del inglés: *Sistema*.

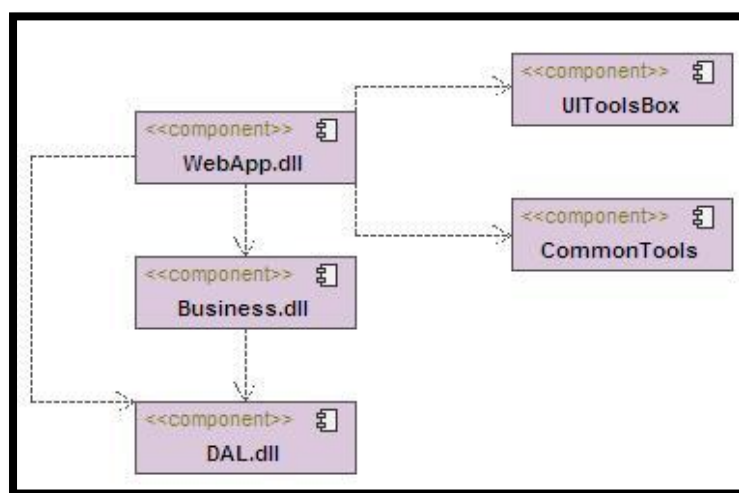
<sup>13</sup> Del inglés: *Colecciones*.

<sup>14</sup> Del inglés: *Formas*.

organización del código implementado. Otro elemento que conforma el modelo de la implementación, son los diagramas de componente y despliegue. Describiéndose la organización y dependencia de los componentes implementados. Representándose además la distribución de estos en los diferentes nodos físicos en los que funcionará la aplicación (Turossi 2008).

### 3.4 Diagrama de componentes

Según (Microsoft 2013) los diagramas de componentes son los encargados de mostrar las dependencias lógicas que existen entre los componentes de software. Siendo estos, ficheros de códigos fuentes, ejecutables o binarios. Está compuesto por componentes y las relaciones entre ellos, siendo utilizados para agrupar los elementos de modelado. En la Figura 3.10 se muestra el diagrama de componentes de la aplicación web.



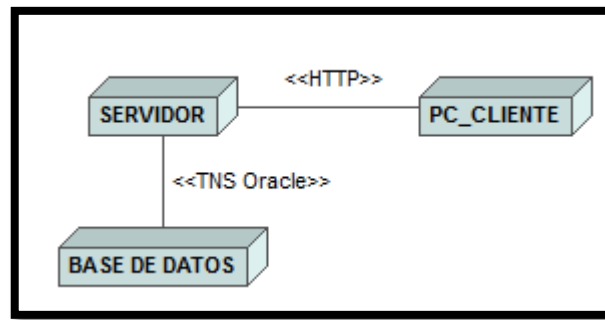
**Figura 3.6:** Diagrama de componentes de la aplicación.

Entre los principales componentes que se pueden encontrar en el sistema tenemos: El componente **WebApp** es el componente de comunicación que permite la integración de un conjunto de elementos y expone las funcionalidades al usuario. La misma utiliza para controlar el diseño y estilo de las interfaces de usuario de la aplicación el componente **UIToolsBox** hace uso además dentro de su arquitectura del componente **CommonTools** con el objetivo de llevar un control de la navegación, acceso a los recursos y la seguridad del sistema de los usuarios. El **Business** encapsula todas las funcionalidades del negocio. Para llevar a cabo la persistencia de las funcionalidades del negocio se utiliza el componente **DAL** el cual hace efectivo los cambios y persistentes los cambios del negocio.

Para visualizar el diagrama de componentes del servicio de Windows y su explicación remitirse al Anexo 5.

### 3.5 Diagrama de despliegue

Los diagramas de despliegue (Larman) permiten entender las relaciones físicas de los componentes de hardware que conforman un sistema (llamados nodos) y el reparto de los componentes de ejecución sobre dichos nodos exponiendo las relaciones que se establecen entre ellos a través de enlaces de comunicación tales como enlaces de red HTTP<sup>15</sup>, conexiones Oracle TNS<sup>16</sup>.



**Figura 3.7:** Diagrama de despliegue de la solución.

En la figura 3.11 se muestra la propuesta del despliegue de la solución del sistema. Donde la PC\_CLIENTE se conectará al servidor de aplicaciones mediante el protocolo HTTP<sup>17</sup>. El servidor de aplicaciones a su vez se conectará al servidor de base de datos, el mismo tendrá una conexión mediante el protocolo TNS.

### 3.6 Prototipo de interfaz del módulo

Para el diseño de las interfaces del módulo se tuvo en cuenta la integración de este con el módulo de administración del SUIN. El diseño de las interfaces fue definido para facilitar la navegación de los usuarios. Lo que permitirá un mejor entendimiento de los resultados que se mostrará. La figura a continuación muestra un ejemplo de una interfaz que conforma el modulo. Las restantes se encontrarán disponibles en el Anexo 2.

<sup>15</sup> Del inglés Hypertext Transfer Protocol (Protocolo de transferencia de texto). Define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web (clientes, servidores, proxies) para comunicarse.

<sup>16</sup> Del inglés Transparent Network Substrate (Sustrato de red transparente) es una capa de comunicación que utilizan las bases de datos Oracle.

<sup>17</sup> Del inglés Hypertext Transfer Protocol (Protocolo de transferencia de texto).

Fecha inicio: 10/24/2012 Fecha fin: 10/28/2012

**Tabla 1:** Trazas del servidor

5 Columnas... Limpiar datos

DateTime	Index	Type	Category	Instance id	Message	Etiqueta	Operaciones
2012-10-27 10:00:45	[160]	Warn	Pres	00000000-0000-0000-0000-000000000000	El recurso suin_proceso_estado_error se encuentra repetido	-	✎ 🗑
2012-10-25 10:00:44	[103]	Warn	UPM	00000000-0000-0000-0000-000000000000	El usuario Anneli González Hernández no tiene acceso sobre el recurso suin_mod_defuncion_pendientes	-	✎ 🗑
2012-10-25 10:00:45	[156]	Warn	Pres	00000000-0000-0000-0000-000000000000	El recurso suin_rectificar_default se encuentra repetido	-huyuu	✎ 🗑
2012-10-25 10:00:45	[157]	Warn	Pres	00000000-0000-0000-0000-000000000000	El recurso suin_rectificar_default se encuentra repetido	-huyuu	✎ 🗑
2012-10-26 10:00:45	[104]	Warn	UPM	00000000-0000-0000-0000-000000000000	El usuario Anneli González Hernández no tiene acceso sobre el recurso suin_mod_defuncion_actasPendientes	-	✎ 🗑

Entradas del (0) - (5) de 8

**Figura 9.8:** Interfaz del módulo para la gestión de trazas del servidor de despliegue del SUIN.

### 3.7 Pruebas

Una de las fases con que cuenta la metodología que se utiliza como guía en el proceso de desarrollo de este módulo es la fase de estabilización. Cuyo objetivo es realizar las pruebas pertinentes al producto que se obtiene como resultado. Las pruebas son una actividad donde el sistema es ejecutado bajo requisitos especificados, se observan los resultados y estos son registrados. Seguidamente se lleva a cabo una evaluación del sistema. Estas son un elemento importante para comprobar la calidad del sistema, representando una revisión final de las especificaciones del diseño y de la codificación. Las pruebas son clasificadas en pruebas de caja negra y caja blanca.

Visual Studio Team System Test (Grupo de Aseguramiento de la Calidad Proyecto Identificación 2010) cuenta con herramientas que realizan las pruebas integradas con Visual Studio. La herramienta Test Edition permite administrar, crear, editar y ejecutar pruebas. Este incluye diferentes tipos de pruebas: manuales, unitarias, web y de carga. Las pruebas que se realizan son ejecutadas por el entorno de desarrollo Visual Studio. Test Edition puede ejecutar grupos de pruebas o cualquier prueba individual que no sea el tipo de prueba manual desde una línea de comando.

#### 3.7.1 Pruebas de caja blanca

Estas pruebas se enfocan en los detalles procedimentales del software. Su diseño está estrechamente relacionado con el código fuente. Son consideradas como unas de las pruebas más importantes que se le realizan la solución. Con la realización de las pruebas de caja blanca disminuyen en gran medida el número de errores que existen en el software, por lo que se logra una mayor calidad y confiabilidad en los sistemas.

## Pruebas Unitarias

Las pruebas unitarias (Microsoft 2013) permiten comprobar el correcto funcionamiento de fragmentos de códigos. Lo que tiene un gran significado e importancia para el equipo de desarrollo. La herramienta de pruebas que contiene el Visual Studio permite generar este tipo de pruebas para los diferentes métodos y funcionalidades existentes en la solución. En la figura a continuación se muestra un ejemplo de las pruebas realizadas a la funcionalidad **InsertarVariables**. Después de realizada las pruebas necesarias a todas las funcionales del módulo, se obtuvo resultados satisfactorios. Lo que garantiza que la solución está desarrollada en correspondencia con el cumplimiento de los requerimientos específicos.

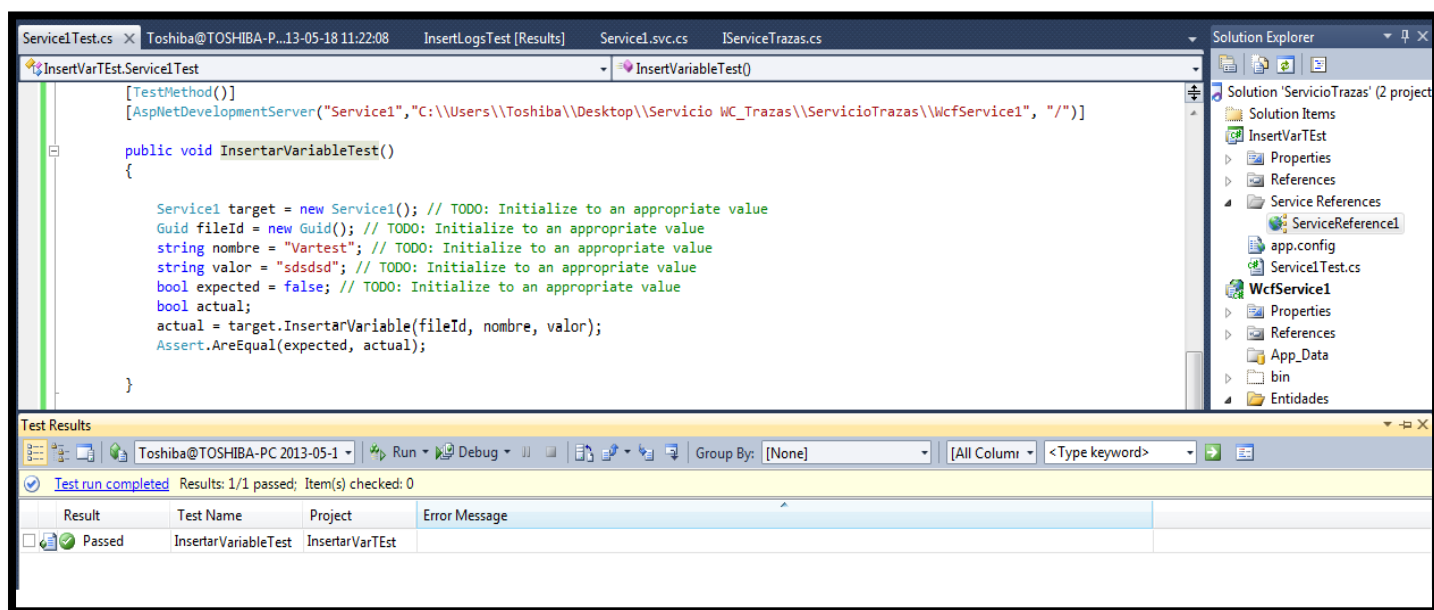


Figura 3.9: Resultado de la prueba de caja blanca realizada a la funcionalidad `InsertarVariable`.

### 3.7.2 Pruebas de caja negra

Las pruebas de caja negra (Pressman 2006) se basan fundamentalmente en los requisitos funcionales del software. Permitiendo la obtención de un conjunto de condiciones de entrada que ejerciten completamente los requerimientos funcionales del módulo. La realización de los casos de pruebas permite demostrar que las funcionalidades del sistema son operativas, que las entradas son aceptadas de manera correcta al igual que el resultado que se obtiene, así como la integridad de la información externa se mantiene. Se muestra a continuación la descripción de uno de los casos de pruebas realizados a los requisitos funcionales del sistema. Las restantes descripciones se encuentran en el Anexo 4.

Escenario	Descripción	Variable 1	Variable 2	Respuesta del sistema	Flujo Central
EC 1.0 Seleccionar opción "Trazas del Servidor".	Permite entrar en el sistema para realizar el análisis de las trazas.	N/A.	N/A.	Muestra interfaz con las opciones para realizar la carga de las trazas.	Seleccionar opción "Trazas del Servidor" del menú de administración.
EC 1.1 Seleccionar Fecha.	Permite seleccionar las fechas a partir de la cual se genera la carga.	N/A.	N/A.	El sistema muestra un DatePicker para que el usuario seleccione la fecha.	Mostrar campos para introducir las fechas. <ul style="list-style-type: none"> <li>• Fecha de Inicio.</li> <li>• Fecha Fin.</li> </ul> Permitir introducir la fecha de inicio. Permitir introducir la fecha de fin.
EC 1.2 Seleccionar opción "Cargar"	Permite cargar las trazas según la fecha especificada.	<b>Fecha Inicio</b>	<b>Fecha Fin</b>	Muestra interfaz con las trazas cargadas en formato tabular.	Seleccionar opción "Trazas del Servidor". Insertar intervalos de fecha. Seleccionar opción "Cargar".
			V 13/2/2013		
		V 12/2/2013			
		/ 12/2/2013	/ 11/2/2013	Muestra mensaje de error "Rango de Fecha Incorrecto".	
		/ Vacío.	/ Vacío.	Muestra mensaje de error "No se permiten campos Vacío".	

Tabla 3.5: Descripción del CPRF1 "Cargar Trazas".

Después de aplicar las pruebas pertinentes a la solución final del módulo de gestión de trazas en la versión 1.0. Fueron arrojando un conjunto de no conformidades las cuales se fueron eliminando durante las iteraciones por las que ha transitado el sistema. En la figura que se muestra a continuación se puede observar un resumen de las iteraciones con el resultado de las no conformidades.

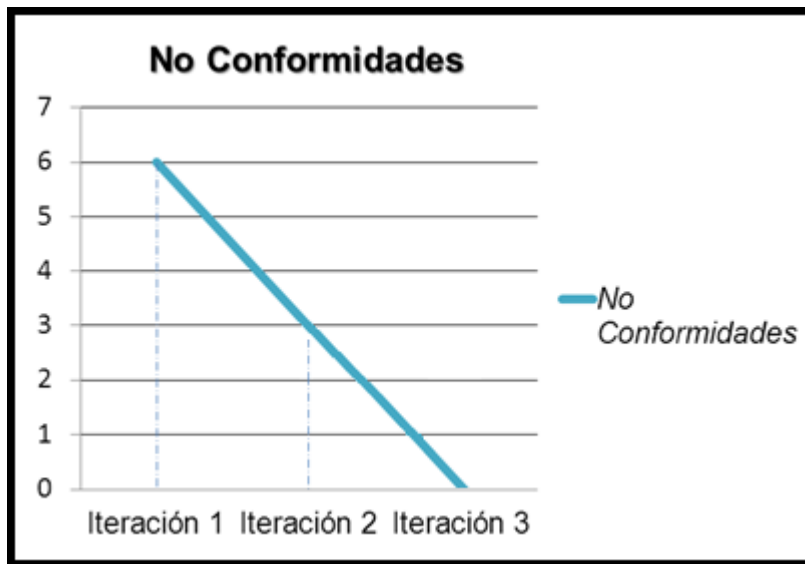


Figura 3.10: Resultados de las iteraciones.

### 3.8 Beneficios del sistema

Con el desarrollo del módulo de gestión de trazas en el servidor de despliegue del Sistema Único de Identificación Nacional de la República de Cuba. Se obtiene resultados satisfactorios en cuanto a los tiempos de respuestas ante situaciones poco tangible al análisis manual. Además de aumentar el nivel de automatización de los procesos que se realizan en las oficinas de la DIR. El módulo desarrollado brinda beneficios tales como:

- Se agrupa las trazas técnicas que se encuentran en todos los servidores.
- Se realiza un procesamiento de los ficheros logs de forma tal que la información se muestre en un formato más entendible para el usuario.
- Se lleva a cabo el procesamiento de las trazas donde se agrupan las mismas a partir de las definiciones realizadas por el usuario una vez establecido el etiquetado.
- Se expone como una aplicación web donde cada uno de los usuarios puede acceder en cualquier momento a las trazas recogidas de cada servidor de manera tal que se le puede aplicar filtro y etiquetado según la comodidad del usuario.

### Conclusiones Parciales

En este capítulo se logra el desarrollo e implementación total de la solución final lo que permite arribar a las siguientes conclusiones:



- El uso de los estándares de codificación permitió la obtención de un código legible y organizado.
- Mediante el tratamiento de las excepciones, se interceptan los errores en caso de existir tanto a nivel de aplicación como a nivel de usuario.
- El diagrama de componentes posibilitó mostrar las dependencias lógicas que existen entre los componentes de software.
- La modelación del diagrama de despliegue permitió entender las relaciones físicas de los componentes de hardware que conforman un sistema.
- El desarrollo de las pruebas brindó la posibilidad de validar la solución que cumpla con todos los requerimientos funcionales definidos.

## Conclusiones Generales

La elaboración del presente trabajo de diploma permitió obtener la primera versión del Módulo de Gestión de Trazas generadas en el Servidor de Despliegue del Sistema Único de Identificación Nacional de la República de Cuba. Este módulo garantiza conocer y analizar la información almacenada en las trazas del servidor. Aportando elementos fundamentales a los desarrolladores del SUIN permitiendo el aumento del rendimiento y el correcto funcionamiento del sistema.

Durante la realización de este trabajo:

- Se realizó un estudio a los sistemas analizadores de trazas en servidores, lo que posibilitó conocer el funcionamiento de estos y definir los requerimientos del sistema.
- La metodología, técnicas y herramientas utilizadas, permitieron el desarrollo de una aplicación acorde con los estándares y procedimientos requeridos por el cliente y el proyecto.
- La arquitectura utilizada en el módulo, brinda un alto grado de encapsulamiento de las responsabilidades, posibilitando la reutilización del código y la disminución del acoplamiento entre las clases.
- El desarrollo del módulo para la gestión de trazas en el servidor de despliegue, permite acceder y analizar la información contenida en las trazas.
- La ejecución de las pruebas para la validación del sistema propuesto permitieron detectar las no conformidades del mismo y la corrección de estas.

## **Recomendaciones**

Después de haber desarrollado una primera versión del módulo de gestión de trazas del servidor de despliegue del SUIN se recomienda:

- Implementar un interfaz de administración al servicio de Windows.
- Hacer uso de la minería de datos para la realización de un análisis más detallado.

## Referencia Bibliográfica

- "Definiciones.de." Retrieved 21 de marzo, 2013, from <http://definicion.de/modelo-de-datos/>.
- (2000-2013). "C# Station." Retrieved 7 de febrero, 2013, from <http://csharp-station.com/>.
- (2008). Altova UModel User & Reference Manual, Inc Altova.
- (2012-2013). "WebLog Expert." from <http://www.weblogexpert.com/>.
- (2013). "Sawmill, Universal Log File Analysis and Reporting." Retrieved 13 de marzo, 2013, from <http://www.sawmill.net>.
- Alapati, S. (2008). Oracle Database 11g. New Features for Administrators Exam Guide. United States, The McGraw-Hill Companies.
- Alapati, S. R. (2009). Expert Oracle Database 11g Administration. United States of America.
- Altova "Ayuda ER/Studio v8.0.0.5865."
- Altova. (2005-2013). "Altova." from <http://www.altova.com/es/umodel/uml-code-generation.html>.
- Barrett, B. (2012). "The Webalizer." Retrieved 13 de marzo, 2013, from <http://www.webalizer.org>.
- Carmen de Pablos Heredero, J. J. A. L. H., Santiago Martín Romo Romero, Sonia Medina Salgado (2012). Organización y transformación en la empresa, ESIC.
- César de la Torre Llorente, U. Z. C., Miguel Angel Ramos Barroso, Javier Calvaro Nelson (2010). Guía de Arquitectura N-Capas orientada al Dominio con .NET 4.0. España.
- Colombia, M. (2000). "Servicios de Consultoría Microsoft." Retrieved 15, 2013, from <http://www.microsoft.com/colombia/portafolio/msf.htm>.
- Corporation, M. (2013). "Microsoft." Retrieved 07 de febrero, 2013.
- Corporation, M. (2013). "MSDN." Retrieved 11 de febrero, 2013, from <http://msdn.microsoft.com/es-es/library/bb399572.aspx>.
- Cuba, P. I. d. (2009). Estándares de Codificación. Ciudad de la Habana, Universidad de las Ciencias Informáticas.
- Cuéllar, J. F. (2009). CALIDAD Y SEGURIDAD A NIVEL DE FILAS EN BBDD ORACLE. Madrid, Universidad Carlos III de Madrid.
- Fraile, L. Microsoft Solution Framework Agile. MVP Time System.
- García, S. M. (2012). "¿QUÉ ES UN SERVIDOR Y CUÁLES SON LOS PRINCIPALES TIPOS DE SERVIDORES?" APRENDER A PROGRAMAR.COM: 1-5.
- Grupo de Aseguramiento de la Calidad Proyecto Identificación, M. y. E. (2010). Manual de Pruebas utilizando Microsoft Visual Studio Team System. La Habana.

- Ignasi Fosch Alonso, J. A. R. (2006). "Guía de estilo de codificación." 21.
- Jesús M. González Barahona, J. S. P., Gregorio Robles (2003-2007) "Introducción al software libre."
- Larman, C. UML y Patrones, Introducción al análisis y diseño orientado a objetos.
- Larman, C. (1999). UML y Patrones. Introducción al Análisis y Diseño Orientado a Objetos. México, PRENTICE HALL.
- Ltd., W. (2012). "WebSpy." Retrieved 13 de marzo, 2013, from <http://www.webspy.com/>.
- Luis Ramiro Villarroel Gonzalez, C. A. M. Y. (2008). APLICACIÓN DE LA METODOLOGÍA MSF V4.0 A LA DEFINICIÓN E IMPLEMENTACIÓN DE ARQUITECTURAS ORIENTADAS A OBJETOS EN VISUAL STUDIO .NET 2005, CASO PRÁCTICO G5 SHARING FILES. DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN. SANGOLQUI, ESCUELA POLITECNICA DEL EJÉRCITO: 217.
- MacDonald, M. Beginning ASP.NET 4.5 in C#.
- Microsoft. "MSDN, LINQ." from <http://msdn.microsoft.com/es-es/library/bb397926.aspx>.
- Microsoft. (2012). "msdn." MSF for CMMI Process Improvement 6.0, from <http://msdn.microsoft.com/es-es/Library/dd997574.aspx>.
- Microsoft. (2013). "Download Center." Retrieved 7 de febrero, 2013.
- Microsoft. (2013). "msdn." Retrieved 4 de mayo, 2013, from [http://msdn.microsoft.com/es-es/library/ms182516\(v=vs.80\).aspx](http://msdn.microsoft.com/es-es/library/ms182516(v=vs.80).aspx).
- Microsoft. (2013). "msdn." Diagramas de componentes de UML: Referencia Retrieved 22 de abril, 2013, from <http://msdn.microsoft.com/es-es/library/dd409390.aspx>.
- Microsoft. (2013). "Visual Studio." Retrieved 11 de febrero, 2013, from <http://msdn.microsoft.com/es-ES/library/vstudio/hh425099>.
- MSF, M. T. (2002) "MSF Process Model v 3.1."
- Narváez, P. Concepción y desarrollo de aplicaciones Web con Visual Studio 2005, Microsoft Most Valuable Professional.
- Norberto López Mazón, J. P. v., Juan Carlos Trujillo Mondéjar (2011). Diseño y explotación de almacenes de datos: Conceptos básicos de modelado multidimensional, Club Universitario.
- Oracle. "Oracle Database." Retrieved 11 de febrero, 2013, from <http://www.oracle.com/es/products/database/overview/index.html>.
- Pressman, R. S. (2006). Ingeniería de Software: Un enfoque práctico. Mexico DF, Editorial McGraw Hill.
- Pressman, R. S. (2010). A Practitioner's Approach. New York, McGraw-Hill.

Roly Aruquipa Huaycho, G. G. R., Laura María Huanca Poma, Tomas Quelca Quelca, Omar Tito Endara Análisis y Diseño de Sistemas II. Ingeniería de Sistemas, universidad Salesiana de Bolivia.

S.L., C. (2010). "Catrian, Managed Services Provider. "Retrieved 13 de marzo, 2013, from <http://catrian.com/>.

Schmuller, J. Aprendiendo UML en 24 horas. Mexico, Prentice Hall.

Software, U. D. d. I. d. "Patrones de "Gang of Four"."

Turossi, A. G. (2008). "Scribd. "Retrieved 28 de marzo, 2013, from <http://es.scribd.com/doc/57383775/implementacion>.

## Bibliografía Consultada

- Alapati, S. R. (2009). *Expert Oracle Database 11G Administration*.
- Bill Evjen, S. H. (2010). *Professional ASP.NET 4 in C# and VB*.
- César De La Torre Llorente, J. C. (2010). *Guía de Arquitectura N Capas orientada al Dominio con NET 4.0*.
- Daniele Bochicchio, S. M. (2011). *ASP.NET 4.0 in Practice*.
- Date, C. (2001). *Introducción a los sistemas de bases de datos*. Mexico.
- DotNet Mania # 1*. (2004).
- ERIKA CAMACHO, F. C. (2004). *Guía Arquitectura v 2*.
- Klein, S. (2010). *Pro Entity Framework 4.0*.
- Larman, C. (s.f.). *UML y Patrones. 2da Edicion*.
- Mackey, A. (2010). *Introducing NET 4.0 with Visual Studio 2010*.
- Magennis, T. (2010). *LINQ to Objects using C# 4 0*.
- Magennis, T. (2010). *LINQ TO OBJECTS USING C# 4.0 USING AND EXTENDING LINQ TO OBJECTS AND PARALLEL LINQ (PLINQ)*.
- Marco Russo, P. P. (2010). *Programming Microsoft LINQ in Microsoft NET Framework 4.0*.
- Mayo, J. (2010). *Visual Studio 2010*. United States.
- Microsoft. (s.f.). *Journal Architect No. 9*.
- Pablo Cibraro, K. C. (2010). *Professional WCF 4 0*.
- Parra, J. D. (2008). *Guía de Patrones, Prácticas y Arquitectura .NET*.
- Sharp, J. (2010). *Microsoft Visual C# 2010 Step by Step*. Washington.
- Sharp, J. (2010). *Windows Communication Foundation 4 Step by Step*.
- Zeldman, J. (s.f.). *Diseño con estándares Web*.

## Glosario de Términos

- **Base de Datos:** Conjunto exhaustivo no redundante de datos estructurados organizados independientemente de su utilización y su implementación en máquina, accesibles en tiempo real y compatibles con usuarios concurrentes con necesidad de información diferente y no predicable en tiempo.
- **Biblioteca:** Es un conjunto de subprogramas utilizados para desarrollar software. Las bibliotecas contienen código y datos, que proporcionan servicios a programas independientes, es decir, pasan a formar parte de estos. Esto permite que el código y los datos se compartan y puedan modificarse de forma modular.
- **Framework:** Es un marco de aplicación o conjunto de bibliotecas orientadas a la reutilización a muy gran escala de componentes software para el desarrollo rápido de aplicaciones.
- **Fichero:** En informática, un archivo o fichero también es un conjunto de información que se almacena en forma virtual para ser leído y/o accedido por medio de una computadora.
- **XAML:** (Lenguaje Extensible de Formato para Aplicaciones en español), es el lenguaje de formato para la interfaz de usuario para la Base de Presentación de Windows. Fue diseñado para soportar las clases y métodos de la plataforma de desarrollo .NET que tienen relación con la interacción con el usuario, en especial el despliegue en pantalla. XAML es un lenguaje declarativo basado en XML, optimizado para describir gráficamente interfaces de usuarios visuales ricas desde el punto de vista gráfico.
- **Módulo:** Componente auto controlado de un sistema que posee una interfaz bien definida hacia otros componentes.
- **Servidor:** Un servidor es una computadora que maneja peticiones de datos, correo, servicios de redes y transferencia de archivos de otras computadoras (clientes). También puede referirse a un software específico. Una computadora puede tener distintos software de servidor, proporcionando muchos servidores a clientes en la red.
- **Servicio Windows:** Los servicios no son nada más ni nada menos que programas o aplicaciones cargadas por el propio sistema operativo. Estas aplicaciones tienen la particularidad que se encuentran corriendo en segundo plano (Background).
- **Servicio web:** Es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos.

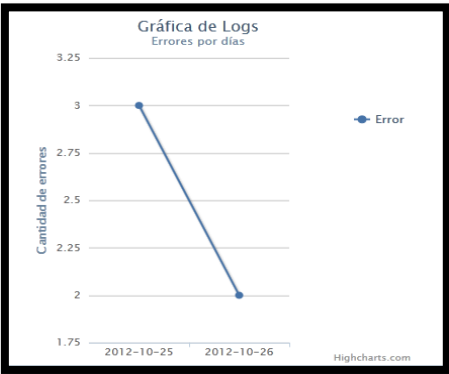


- **Script:** Es un programa usualmente simple, que por lo regular se almacena en un archivo de texto plano. Los script son casi siempre interpretados, pero no todo programa interpretado es considerado un script. El uso habitual de los scripts es realizar diversas tareas como combinar componentes, interactuar con el sistema operativo o con el usuario.

## Anexos

### Anexo 1: Especificación de Requisitos.

#### RF2. Graficar errores por día.

<b>Propósito:</b>	Permite mostrar la información analizada.
<b>Roles:</b>	Administrador Nacional.
<b>Precondiciones:</b>	<ol style="list-style-type: none"> <li>1. El usuario debe estar autenticado en el sistema.</li> <li>2. Debe estar cargada al menos una traza.</li> </ol>
<b>Entidades tratadas:</b>	Ver Diccionario de datos.
<b>Descripción:</b>	<ol style="list-style-type: none"> <li>2.1 Seleccionar la opción "Graficar".</li> <li>2.2 Mostrar interfaz con un resumen de la información analizada por las diferentes opciones:</li> </ol>
<b>Validaciones:</b>	Para realizar las operaciones: <ol style="list-style-type: none"> <li>1. Ver las validaciones de los conceptos tratados en el Diccionario de datos.</li> </ol>
<b>Poscondiciones:</b>	<ol style="list-style-type: none"> <li>1. Se muestra el resumen del análisis de la información existente.</li> </ol>
<b>Prototipo:</b>	 <p><b>Figura 3 Resultado del Análisis</b></p>

**Tabla 6:** Especificación del RF "Mostrar Resultado del Análisis".

#### RF3. Filtrar Información.

<b>Propósito:</b>	Realizar un filtrado de la información manejada.
<b>Roles:</b>	Administrador Nacional.
<b>Precondiciones:</b>	<ol style="list-style-type: none"> <li>1. El usuario debe estar autenticados en el sistema.</li> <li>2. Debe estar cargada al menos una traza.</li> </ol>
<b>Entidades tratadas:</b>	Ver Diccionario de datos.

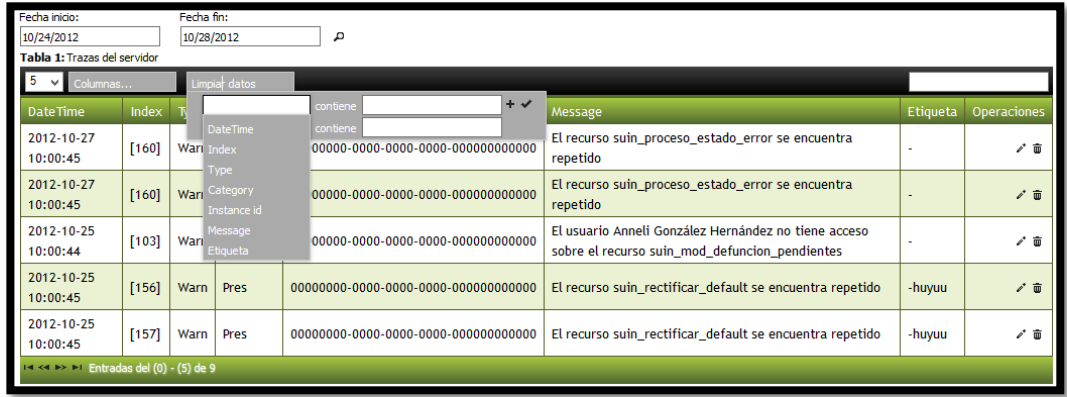
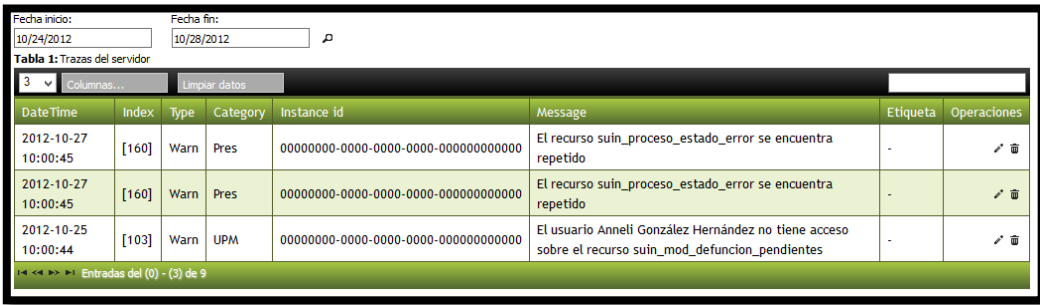
<b>Descripción:</b>	<p>3.1 Seleccionar opción “Limpiar Datos”, “Columnas”.</p> <p>3.1.1 Seleccionar el valor de la columna por la cual se quiere filtrar y el texto que contiene si se selecciona la opción “Limpiar Datos” para realizar un filtrado horizontal.</p> <p>3.1.1.1 Seleccionar opción “Adicionar” para adicionar otra columna para realizar el filtrado.</p> <p>3.1.2 Seleccionar el valor de la(s) columna(s) que se quiere trabajar para realizar un filtrado vertical.</p> <p>3.2 Mostrar los resultados del filtrado actualizando la tabla donde se encuentran las trazas.</p> <p>3.3 Insertar mensaje en el cuadro de texto para realizar un filtrado en toda la tabla.</p>
<b>Validaciones:</b>	<p>Para realizar las operaciones:</p> <p>1. Ver las validaciones de los conceptos tratados en el Diccionario de datos.</p>
<b>Poscondiciones:</b>	1. Se realiza una búsqueda más específica y detallada.
<b>Prototipo:</b>	 <p>The screenshot shows a web application interface for filtering information. At the top, there are date range filters: 'Fecha inicio: 10/24/2012' and 'Fecha fin: 10/28/2012'. Below this is a table titled 'Tabla 1: Trazas del servidor'. The table has columns: 'DateTime', 'Index', 'Type', 'Message', 'Etiqueta', and 'Operaciones'. A dropdown menu is open over the 'Index' column, showing options like 'DateTime', 'Index', 'Type', 'Category', 'Instance id', 'Message', and 'Etiqueta'. The table contains several rows of trace data, including timestamps, index values (e.g., [160], [103], [156], [157]), types (Warn, Pres), and messages. The bottom of the interface shows 'Entradas del (0) - (5) de 9'.</p>

Figura 4 Filtrar Información.

Tabla 7: Especificación del RF "Filtrar Información".

**RF4. Eliminar Basura.**

<b>Propósito:</b>	Eliminar todas las trazas que no se consideren necesarias para realizar el análisis.
<b>Roles:</b>	Administrador Nacional del Sistema.
<b>Precondiciones:</b>	<p>1. El usuario debe estar autenticado en el sistema.</p> <p>2. Debe estar cargada al menos una traza.</p>
<b>Entidades tratadas:</b>	Ver Diccionario de datos.
<b>Descripción:</b>	<p>4.1 Seleccionar la(s) trazas a eliminar.</p> <p>6.1. Seleccionar opción “Eliminar” donde se encuentran la columna de operaciones.</p> <p>4.2 Mostrar las opciones “Eliminar” y “Cancelar”.</p> <p>4.4.1 Eliminar las trazas, si se selecciona la opción “Aceptar”.</p>

	<p>4.4.1.1 Mostrar mensaje de confirmación.</p> <p>4.4.2 Cancelar la operación si se selecciona la opción “Cancelar”.</p> <p>4.4 Mostrar interfaz con un listado actualizado de las trazas.</p>
<b>Validaciones:</b>	<p>Para realizar las operaciones:</p> <p>1. Ver las validaciones de los conceptos tratados en el Diccionario de datos.</p>
<b>Poscondiciones:</b>	<p>1. Se eliminan las trazas seleccionadas.</p>
<b>Prototipo:</b>	 <p><b>Figura 5 Eliminar Basura.</b></p>

**Tabla 8:** Especificación del RF "Eliminar Basura".

## RF5. Establecer Etiquetado.

<b>Propósito:</b>	Establecer características más específicas de la información de las trazas.
<b>Roles:</b>	Administrador Nacional del Sistema.
<b>Precondiciones:</b>	<p>1. El usuario debe estar autenticado en el sistema.</p> <p>2. Debe estar cargada al menos una traza.</p>
<b>Entidades tratadas:</b>	Ver Diccionario de datos.
<b>Descripción:</b>	<p>5.1. Seleccionar la traza a etiquetar.</p> <p>5.2. Seleccionar opción “Etiquetado” donde se encuentran la columna de operaciones.</p> <p>5.3. Mostrar cuadro de texto para establecer el etiquetado.</p> <p>5.3.1. Establecer etiqueta que se desea asignar.</p> <p>5.4. Mostrar las opciones “Aceptar”.</p> <p>5.4.1. Establecer el etiquetado, si se selecciona la opción “Aceptar”.</p>
<b>Validaciones:</b>	<p>Para realizar las operaciones:</p> <p>1. Ver las validaciones de los conceptos tratados en el Diccionario de datos.</p>
<b>Poscondiciones:</b>	<p>1. Se realiza el etiquetado a la información contenida en las trazas.</p>
<b>Prototipo:</b>	

Fecha inicio: 10/24/2012 Fecha fin: 10/28/2012

Tabla 1: Trazas del servidor

5 Columnas... Limpiar datos

Date Time	Index	Type	Category	Instance id	Message	Etiqueta	Operaciones
2012-10-27 10:00:45	[160]	Warn	Pres	00000000-0000-0000-0000-000000000000	El recurso suin_proceso_estado_error se encuentra repetido	-	✎🗑
2012-10-25 10:00:44	[103]	Warn	UPM	00000000-0000-0000-0000-000000000000	El usuario Anneli González Hernández no tiene acceso sobre el recurso suin_mod_defuncion_pendientes	-	✎🗑
2012-10-25 10:00:45	[156]	Warn	Pres	00000000-0000-0000-0000-000000000000	El recurso suin_rectificar_default se encuentra repetido	-huyuu	✎🗑
2012-10-25 10:00:45	[157]	Warn	Pres	00000000-0000-0000-0000-000000000000	El recurso suin_rectificar_default se encuentra repetido	-huyuu	✎🗑
2012-10-26 10:00:45	[104]	Warn	UPM	00000000-0000-0000-0000-000000000000	El usuario Anneli González Hernández no tiene acceso sobre el recurso suin_mod_defuncion_actasPendientes	-	✎🗑

Entradas del (0) - (5) de 8

Figura 6 Establecer Etiquetado.

Tabla 9: Especificación del RF "Establecer Etiquetado".

**RF6. Establecer Configuración del Servicio.**

<b>Propósito:</b>	Permite establecer la configuración de las trazas del servicio.
<b>Roles:</b>	No Procede.
<b>Precondiciones:</b>	No Procede.
<b>Entidades tratadas:</b>	Ver Diccionario de datos.
<b>Descripción:</b>	1.4. Seleccionar la estructura definida en el fichero de configuración. 1.5. Procesar la estructura.
<b>Validaciones:</b>	Para realizar las operaciones: 2. Ver las validaciones de los conceptos tratados en el Diccionario de datos.
<b>Poscondiciones:</b>	2. Se obtiene los campos que conforman las trazas y sus valores.
<b>Prototipo:</b>	No Procede.

Tabla 10: Especificación del RF "Establecer Configuración del Servicio".

**RF7. Cargar Ficheros.**

<b>Propósito:</b>	Permite la carga de los ficheros que contienen los logs en cada servidor a partir de una configuración de tiempo especificada.
<b>Roles:</b>	No Procede.
<b>Precondiciones:</b>	No Procede.
<b>Entidades tratadas:</b>	Ver Diccionario de datos.
<b>Descripción:</b>	2.1 Buscar los logs que se encuentran especificados en las direcciones en el fichero de configuración. 2.2 Copiar a una dirección controlada por el servicio. 2.3 Eliminar el log del directorio especificado.

	2.4 Interpretar cada fichero a partir del formato establecido por la ubicación. 2.5 Descartar los ficheros que no se encuentren en el formato especificado. 2.6 Almacenar el log en la base de datos.
<b>Validaciones:</b>	Para realizar las operaciones: 2. Ver las validaciones de los conceptos tratados en el Diccionario de datos.
<b>Poscondiciones:</b>	2. Se carga los logs según el directorio especificado y se almacena en la base de datos.
<b>Prototipo:</b>	No Procede.

Tabla 11: Especificación del RF "Cargar Fichero".

**RF8. Generar Log.**

<b>Propósito:</b>	Permite generar los propios logs de la aplicación a partir del funcionamiento de la misma.
<b>Roles:</b>	No Procede.
<b>Precondiciones:</b>	No Procede.
<b>Entidades tratadas:</b>	Ver Diccionario de datos.
<b>Descripción:</b>	3.4 Generar logs a partir de los errores que ocurren en el sistema.
<b>Validaciones:</b>	Para realizar las operaciones: 2. Ver las validaciones de los conceptos tratados en el Diccionario de datos.
<b>Poscondiciones:</b>	2. Se genera los logs con los errores que ocurren en el módulo.
<b>Prototipo:</b>	No Procede.

Tabla 12: Especificación del RF "Generar Log".

**Anexo 2: Prototipos de Interfaz de Usuario.**

Figura 11: Interfaz para insertar la fecha de inicio de la carga.

Fecha inicio:  Fecha fin:

**Tabla 1: Trazas del servidor**

1 Columns...

May 2013

Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

Figura 12: Interfaz para insertar la fecha de fin de la carga.

Fecha inicio:  Fecha fin:

**Tabla 1: Trazas del servidor**

5 Columns... Limpiar datos

DateTime	Index	Type	Category	Instance id	Message	Etiqueta	Operaciones
2012-10-27 10:00:45	[160]	Warn	Pres	00000000-0000-0000-0000-000000000000	El recurso suin_proceso_estado_error se encuentra repetido	-	
2012-10-27 10:00:45	[160]	Warn	Pres	00000000-0000-0000-0000-000000000000	El recurso suin_proceso_estado_error se encuentra repetido	-	
2012-10-25 10:00:44	[103]	Warn	UPM	00000000-0000-0000-0000-000000000000	El usuario Anneli González Hernández no tiene acceso sobre el recurso suin_mod_defuncion_pendientes	-	
2012-10-25 10:00:45	[156]	Warn	Pres	00000000-0000-0000-0000-000000000000	El recurso suin_rectificar_default se encuentra repetido	-huyuu	
2012-10-25 10:00:45	[157]	Warn	Pres	00000000-0000-0000-0000-000000000000	El recurso suin_rectificar_default se encuentra repetido	-huyuu	

Entradas del (0) - (5) de 9

Figura 13: Interfaz después de llevado a cabo la carga de las trazas.

Fecha inicio:  Fecha fin:

**Tabla 1: Trazas del servidor**

5 Columns... Limpiar datos

DateTime	Index	Type	Category	Instance id	Message	Etiqueta	Operaciones
2012-10-27 10:00:45	[160]	Warn	Pres	00000000-0000-0000-0000-000000000000	El recurso suin_proceso_estado_error se encuentra repetido	-	
2012-10-27 10:00:45	[160]	Warn	Pres	00000000-0000-0000-0000-000000000000	El recurso suin_proceso_estado_error se encuentra repetido	-	
2012-10-25 10:00:44	[103]	Warn	UPM	00000000-0000-0000-0000-000000000000	El usuario Anneli González Hernández no tiene acceso sobre el recurso suin_mod_defuncion_pendientes	-	
2012-10-25 10:00:45	[156]	Warn	Pres	00000000-0000-0000-0000-000000000000	El recurso suin_rectificar_default se encuentra repetido	-huyuu	
2012-10-25 10:00:45	[157]	Warn	Pres	00000000-0000-0000-0000-000000000000	El recurso suin_rectificar_default se encuentra repetido	-huyuu	

Entradas del (0) - (5) de 9

Figura 14: Interfaz que muestra las opciones de filtrado.

**Anexo 3:** Pruebas Unitarias.

**Anexo 4:** Especificación de los Casos de Pruebas basado en los Requisitos Funcionales.

**RF2. Graficar Error por día.**

Escenario	Descripción	Variable 1	Respuesta del sistema	Flujo Central
EC 2.0 Seleccionar la opción "Graficar Información"	Permite mostrar una gráfica a partir de los datos de las trazas.	N/A.	Muestra gráfica a partir de los datos de las trazas.	Seleccionar la opción "Graficar Información".

**Tabla 13:** Descripción del CPRF2 "Mostrar Resultados".**RF3 Filtrar Información.**

Escenario	Descripción	Variable 1	Variable 2	Respuesta del sistema	Flujo Central
EC 3.1 Seleccionar opción "Limpiar Datos"	Opción que permite realizar una búsqueda en el contenido de las trazas.	<b>Campos de la Tabla.</b>	<b>Contenido</b>	Campo de texto para insertar el contenido del filtrado y campo de selección para seleccionar el valor de la columna a filtrar.	Click sobre el botón donde se realiza el filtrado.
		V Date Time, Index, Type, Category, Instance Id, Message, Etiqueta.	V 2012-10-25, [124], Warn, Pres, 000.		
			/ **, #, @#\$.	Muestra un mensaje de error.	
EC 3.2 Insertar datos en el textBox del filtrado.	Campo de texto para insertar el contenido del filtrado.	<b>Filtrado de Información.</b>	N/A.	Muestra los elementos que cumplen con el valor pasado.	Click sobre el textBox donde se realiza el filtrado. Insertar datos en el campo de filtrado.
		V 103, Warn, 2012-10-26			
		/ Warn1, 1ª3, 2012;3;12, @#\$.		Se muestra la tabla sin coincidencias.	
EC 3.3		<b>Columnas</b>	N/A.		



Seleccionar columnas del campo de selección.	Permite seleccionar los valores a trabajar dependiendo de las variables en las columnas.	V Date Time, Index, Type, Category, Instance Id, Message, Etiqueta.		Muestra la información en la tabla de las columnas seleccionadas.	Click sobre el campo selección "Columnas". Seleccionar las columnas con las cuales se va a trabajar.
EC 3.4 Ocultar columnas.	Permite ocultar las columnas que no son necesarias en el momento del análisis.	N/A.	N/A.	Ocultar las columnas a las cuales se les dan click.	Click encima de columna a ocultar.

**Tabla 14:** Descripción del CPRF1 "Filtrar Información".

#### Descripción de Variables.

No	Nombre de Campo	Clasificación	Valor Nulo	Descripción
Variable 1	Campos de la Tabla.	Campo de selección.	No.	Campo que permite seleccionar los nombres de las columnas.
Variable 2	Contenido.	Campo de texto.	No.	Campo que permite introducir los datos para realizar el filtrado. Admite letras y dígitos.
Variable 3	Filtrado.	Campo de texto.	Sí.	Campo de texto que permite introducir los datos para realizar filtrado. Admite letras y dígitos.
Variable 4	Columnas.	Campo de selección.	Sí.	Campo que permite seleccionar los nombres de las columnas.

**Tabla 15:** Descripción de Variables del CPRF3 "Filtrar Información".

#### RF4 Eliminar Basura.

Escenario	Descripción	Variable 1	Respuesta del sistema	Flujo Central
-----------	-------------	------------	-----------------------	---------------

EC Seleccionar opción "Eliminar".	4.1	Permite eliminar una o varias trazas de la tabla de trazas.	N/A.	Muestra mensaje de confirmación.	Seleccionar la traza. Seleccionar opción "Eliminar" de la columna de operaciones.
EC Seleccionar opción "Eliminar" del mensaje de confirmación.	4.1.1	Elimina las trazas seleccionadas.	N/A.	Elimina las trazas seleccionadas.	Seleccionar la(s) traza(s). Seleccionar opción "Eliminar". Seleccionar opción "Eliminar" del mensaje de confirmación. Mostrar listado de trazas actualizado.
EC Seleccionar opción "Cancelar" del mensaje de confirmación.	4.1.2	Cancela la operación.	N/A.	Retorna a la página anterior.	Seleccionar la(s) traza(s). Seleccionar opción "Eliminar Traza". Seleccionar opción "Cancelar" del mensaje de confirmación. Retorna a la página anterior.

Tabla 16: Descripción del CPRF4 "Eliminar Basura".

**RF5 Establecer Etiquetado.**

Escenario	Descripción	Variable 1	Respuesta del sistema	Flujo Central
EC Seleccionar opción "Etiquetado".	5.1 Permite etiquetar las trazas seleccionadas.	N/A.	Muestra campo de texto para realizar el etiquetado.	Seleccionar la traza a etiquetar. Seleccionar opción "Etiquetado" la columna de operaciones.
EC 5.2 Introducir datos.	Introducir datos para el etiquetado de la(s) traza(s)	N/A.		
EC Seleccionar opción "Aceptar".	5.3 Establecer etiqueta a la traza.	<b>Asignar Etiqueta</b>	Establece etiqueta a la(s) traza(s).	Seleccionar la traza a etiquetar. Seleccionar opción "Etiquetado" la columna de operaciones. Introducir etiqueta. Seleccionar opción "Aceptar".
		V El mensaje.		

Tabla 17: Descripción del CPRF5 "Establecer Etiquetado".

**Descripción de Variables.**

No	Nombre de Campo	Clasificación	Valor Nulo	Descripción
Variable 1	Asignar Etiqueta.	Campo de texto	No	Campo de texto que permite introducir los datos para establecer el etiquetado. Admite letras y dígitos.

**Tabla 18:** Descripción de Variables del CPRF5 "Establecer Etiquetado".**RF6 Establecer Configuración del Servicio.**

Escenario	Descripción	Variable 1	Respuesta del sistema	Flujo Central
EC1 Seleccionar estructura.	Permite seleccionar la estructura del log definida en el fichero de configuración.	N/A.	El sistema lee la estructura del fichero XML.	
EC2 Procesar la estructura.	Permite procesar la estructura que se especifica en el fichero de configuración.	N/A.	El servicio procesa la estructura especificada en el fichero XML.	

**Tabla 19:** Descripción del CPRF6 "Establecer Configuración del Servicio".**RF7 Cargar Fichero.**

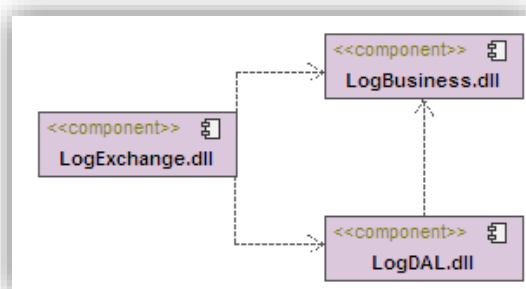
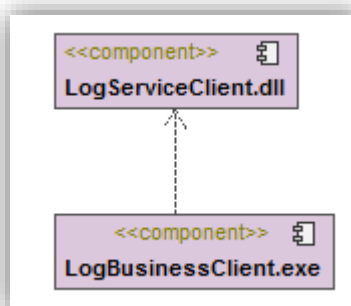
Escenario	Descripción	Variable 1	Respuesta del sistema	Flujo Central
EC1 Buscar logs	Permite obtener los logs desde la dirección especificada.	N/A.	El servicio busca los log desde la dirección especificada en el fichero XML.	
EC2 Interpretar fichero.	Permite interpretar la información que contiene el log.	N/A.	El servicio interpreta el contenido del fichero a partir de la estructura establecida en el fichero XML.	
EC3 Copiar logs a la base de datos.	Permite almacenar la información contenida en el log.	N/A.	El servicio copia la información contenida en el log a la base de datos.	
EC4 Eliminar logs del directorio.	Permite eliminar los logs del directorio especificado.	N/A.	El servicio elimina los logs que se encuentran almacenados en el directorio especificado.	

**Tabla 20:** Descripción del CPRF6 "Cargar Fichero".**RF8 Generar Log**

Escenario	Descripción	Variable 1	Respuesta del sistema	Flujo Central
EC1 Generar logs.	Permite generar log a partir de errores que ocurren en la aplicación.	N/A.	El servicio genera logs sobre errores que ocurren en la aplicación.	

**Tabla 21:** Descripción del CPRF6 "Generar Log".**Anexo 5:** Diagrama de componentes del servicio de windows.

A continuación se presenta el diagrama de componentes del servicio que se utilizará para la carga de los logs. Seguidamente el que corresponde al servicio que se encontrará en las pc clientes.

**Figura 15:** Diagrama de componentes del servicio.**Figura 16:** Diagrama de componentes del servicio en el cliente.

En las figuras 19 y 20 se muestra una representación mediante el diagrama de componentes, referente al funcionamiento del servicio que utilizará el módulo, para la carga de las trazas y el almacenamiento de la base de datos. El componente **LogExchange** permite invocar a los métodos que están en el servicio (funcionalidad de negocio). **LogBussines** contiene la lógica del negocio en el se encuentran las interfaces

que ejecutan las funcionalidades, realizando peticiones a la base de datos. Tiene relación con el componente **LogDAL** ya que este responde a las peticiones que realiza LogBussines. El componente **LogBussinesClient** se ejecuta en la máquina cliente ordenando a **LogServiceClient** que realice la conexión al servicio de Windows que es donde se ejecutan las funcionalidades.