



Pontificia Universidad Católica de Valparaíso

Facultad de Ingeniería

Escuela de Ingeniería Informática

Ingeniería de Ejecución en Informática

SISTEMA SDS MANAGER

Autor:

Javier Alberto De Barbieri Magnone

Informe final del Proyecto para optar al Título profesional de

Ingeniero de Ejecución en Informática

Profesor Guía:

Silvana Roncagliolo de la Horra

JULIO 2007

Dedicada a Dios, a la Universidad y en especial a mi Padre que siempre me acompañará y a mi Madre que sin su apoyo incondicional no podría haber llegado hasta donde me encuentro en estos momentos.

Gracias

Javier De Barbieri Magnone

Resumen

En este proyecto se muestra el análisis, desarrollo, implementación y puesta en marcha de un sistema llamado SDS Manager, el cuál fue construido con la finalidad de satisfacer la necesidad de Kapsch TrafficCom Chile S.A. de manejar las solicitudes de servicios que provienen de los mantenimientos que se realizan en forma periódica a las autopistas urbanas concesionadas en la ciudad de Santiago de Chile. Este sistema se realizó para administrar dichas solicitudes de servicio de la forma más ágil posible. Junto con manejar las solicitudes de servicio, este sistema es capaz de entregar valores estadísticos y cálculos mensuales que son de gran utilidad para dichas concesionarias.

Abstract

This project is to show the analysis, development, implementation and commissioning of a system called SDS Manager, which was built with the aim to meet the need for the company Kapsch TrafficCom Chile S.A. handle requests for services from maintenance to

be carried out on a regular concession to the urban highways in the city of Santiago de Chile. This system was made to run those applications service providers more flexible way possible. Along with handling requests for service, this system is able to deliver monthly statistical values and calculations which are of great utilities to these dealerships.

CAPITULO 1: Introducción

Las tecnologías de hoy en día, permiten agilizar los tiempos de respuesta para muchos propósitos. Así es que aprovechando al máximo los recursos que se tienen, se pueden realizar múltiples tareas en una sola herramienta.

Con esta premisa de utilizar las tecnologías que existen hoy, la empresa Kapsch TrafficCom Chile S.A. desarrolló un sistema llamado “SDS Manager”, el cuál minimiza los tiempos de respuesta de las solicitudes de servicio (SDS) de cada una de las autopistas urbanas con las cuales se tiene contrato de mantenimiento.

Entre las autopistas con las cuales se posee un contrato de mantenimiento vigente se encuentran Autopista Central, Costanera Norte y Vespucio Norte Express.

El procedimiento que se sigue en la empresa es el siguiente: llega una solicitud de servicio (SDS) de un cliente vía correo electrónico, el procedimiento solicitado se realiza en terreno y posteriormente se completa y re-envía por la misma vía los resultados de este procedimiento.

Lo que se quiere lograr con este sistema es automatizar al máximo la entrada y el proceso de la información, junto con entregar información estadística, necesaria para la generación de informes.

1.1 Descripción de la Organización

El cliente para el cual se desarrolló el software es KAPSCH TRAFICCOM CHILE S.A. Esta empresa es una multinacional, cuya casa matriz se encuentra en Viena, Austria. Algunos de los países en que Kapsch está presente son: Argentina, China, Malasia, Dinamarca, Alemania, Suecia, Irlanda, Brasil, Francia, Suiza, Grecia, Filipinas, Eslovenia, Noruega, Polonia, Turquía, Austria, Sudáfrica, Inglaterra y Chile.

En 1991 la empresa SAAB - Combitech desarrolló en Suecia un sistema para el cobro de peajes en forma FREE FLOW, el cual permite a las autopistas realizar cobros en forma expedita y sin la necesidad de que los vehículos se detuvieran o disminuyeran su velocidad. Posteriormente esta empresa se independizó de SAAB y su nombre fue cambiado a Combitech TrafficCom.

En el año 2003 y con la necesidad de Autopista Central de crear la primera autopista concesionada de Santiago de Chile, Combitech participa en el proceso de licitación para la adjudicación de la implementación del sistema de cobro. Esta licitación fue ganada por esta empresa y en el año 2004, previo instalación de los pódicos se instaló el sistema.

Posteriormente, la empresa Combitech fue adquirida por un holding de empresas Austriacas llamada Kapsch TrafficCom AB, lo cuál obligó al cambio de nombre a Kapsch TrafficCom Chile S.A.

Kapsch instaló y realiza las mantenciones de todos los pódicos de cobro de tres de las autopistas que han adquirido su sistema: Autopista Central, Costanera Norte y Vespucio Norte Express.

La primera concesionaria que compró este sistema en Chile fue Autopista Central, la cual en la actualidad posee 28 pódicos y 18 cabinas técnicas.

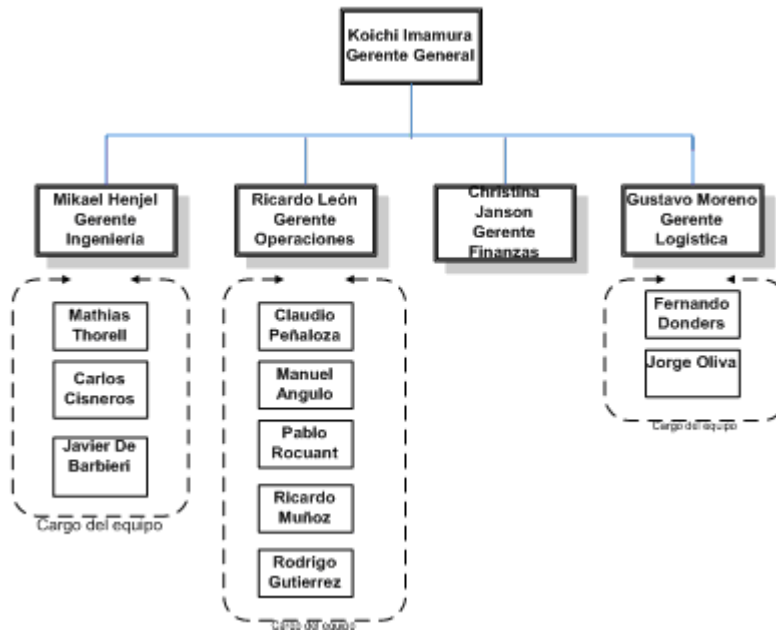
La siguiente concesionaria fue Costanera Norte, la cual posee 17 pódicos y 9 cabinas técnicas.

La última concesionaria que adquirió el sistema MLFF de Kapsch fue Vespucio Norte Express la cual posee 15 pódicos y 11 cabinas técnicas.

1.2 Organización jerárquica de la empresa

En la figura número 1 se apreciará el organigrama de la empresa en la sucursal de Chile.

Figura 1: Organigrama de Kapsch Chile.



1.3 Funcionamiento del Sistema MLFF (Multi Lane Free Flow)

El funcionamiento del Sistema MLFF consiste en lo siguiente.

Al acercarse un vehículo a un pórtico, un sensor que se encuentra a una altura de 11.5 metros, VDC Su (Vehicle Detection Clasification Sensor), detecta que entró a la zona de cobro, junto con determinar la clase del vehículo según las dimensiones del vehículo: sí es automóvil, motocicleta o camioneta corresponde a clase 1; camión sin acoplado o bus, corresponde a clase 2; y camión con acoplado que corresponde a clase 3.

Después que el sensor VDC informa que existe un nuevo vehículo en la zona de detección al VDC Controller (Controlador Vehicle Detection Casification), éste comunica su presencia al controlador más importante de todos los equipos que se encuentran en la cabina técnica, el ALC (Automatic Lane Controller). Este es el encargado de enviar una señal al VR Su (Vehicle Registration Sensor) el cuál es una cámara que registrará una fotografía de 12 bits de la placa patente delantera del vehículo. Dicha fotografía es almacenada en un buffer que se conoce como TSMC, que a su vez es el encargado de enviar la mensajería y fotografías al Sistema Central de la Concesionaria.

El ALC también recibe la información en forma paralela de la transacción que realiza el TAG con la antena TRX. Se tienen tres posibles resultados: el primero de ellos es una transacción exitosa, la segunda es transacción inválida y la tercera es no haber registrado la transacción.

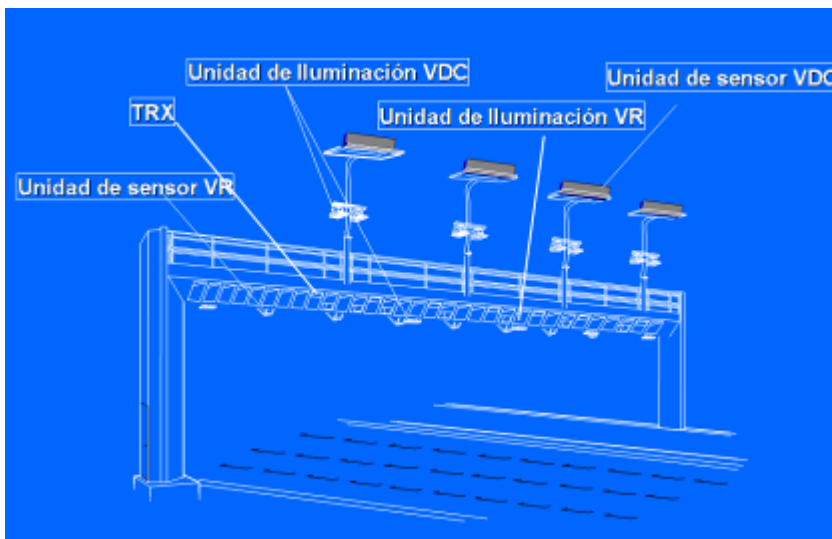
Si la transacción ha sido exitosa, la fotografía que fue almacenada temporalmente en el TSMC es eliminada.

Si la transacción fue inválida, se puede deber a que la clase del TAG que se está utilizando en la transacción, no corresponde a la que está entregando el VDC. También se puede deber a que existen problemas con el equipo del TAG. En ese caso es sólo necesario acercarse a la concesionaria propietaria de dicho TAG. Como la transacción fue inválida, la fotografía que se encuentra almacenada en el TSMC es analizada por un sistema de OCR (Optical Character Recognizer) el cuál es capaz de registrar la placa patente en un 80% de las veces. De todas maneras es enviada al Sistema Central para su posterior análisis manual.

Si no se produjo transacción con el TAG, se tratará de igual manera que si se tratara de un paso no válido por un pórtico. Es almacenada la fotografía en el TSMC, analizada por un OCR y posteriormente enviada al Sistema Central.

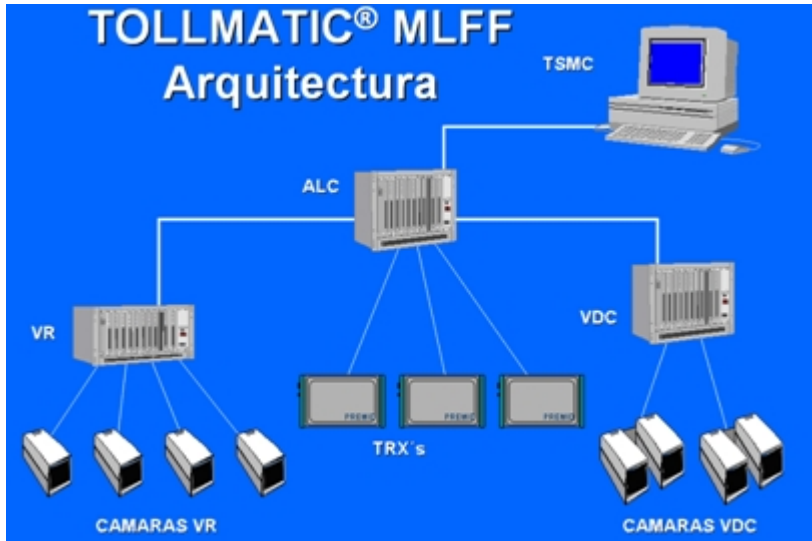
En la figura número 2 se apreciará los diferentes sensores que se encuentran en un pórtico.

Figura 2: Sensores en Pórtico.



En la figura número 3 se aprecia la distribución de los equipos instalados en la cabina técnica.

Figura 3: Sistemas en Cabina Técnica.



CAPITULO 2: Objetivos del Sistema

2.1 Objetivo General del Sistema

El objetivo general del proyecto fue desarrollar un sistema que sea capaz de administrar las Solicitudes de Servicios (SDS) de una forma segura y ágil para la empresa, además de proveer la información estadística necesaria de los informes que se utilizan a nivel gerencial.

2.2 Objetivos Específicos del Sistema

Para poder cumplir con el objetivo general del proyecto se definieron los siguientes objetivos específicos.

- Desarrollar un sistema multiusuario, donde cada uno de los usuarios que ingresen al sistema puedan manejar la información de las SDS y los reportes que generan.
- Administrar de forma correcta y ágil cada una de las SDS que se manejan en la empresa.
- Automatizar la mayor cantidad de funciones, para así evitar que se cometan errores al manejar y gestionar la información contenida dentro de dicho sistema.
- Crear un ambiente visual que ofrezca un manejo ágil y sencillo, sin exposiciones visuales que puedan afectar la concentración y correcto uso del sistema, puesto que éste tendrá un enfoque de carácter interno y no con énfasis comercial o publicitario.
- Crear un sistema modulado lo cual permita eventualmente reconstruir o incluir más funciones sin que éstas modifiquen la estructura del sistema.
- Obtener como resultado final un sistema no totalmente genérico y que pueda tener expectativas de redistribución sin tener que realizar modificaciones mayores.

CAPITULO 3: Problemática y Solución Propuesta

En este capítulo se muestra el procedimiento anterior que la empresa utilizaba en lo que se refiere a las SDS y la problemática que existe. Posteriormente se muestra cual es la solución propuesta.

3.1 Procedimiento anterior

Una Solicitud de Servicio (SDS) se produce cuando en alguno de los Sistemas o Subsistemas ubicados en pórticos o cabinas técnicas, emite algún error y se requiere dejar un registro o constancia de esto. Dicho error es reportado a la concesionaria a través de una alarma generada por los mismos dispositivos. Después de que la persona encargada en el sistema central recoge la falla, se avisa telefónicamente al personal de Mantenimiento de Kapsch, y si la situación lo amerita, el personal de la autopista genera la SDS. Esta solicitud es generada en formato Word (ver anexo B documento SDS) estándar para todas las autopistas. Este archivo Word es adjuntado a un correo y enviado al correo de mantenimiento de Kapsch.

Posteriormente a eso, el personal se hace presente en el sitio para solucionar la falla o alarma que se ha presentado. Luego de solucionar la falla, se utiliza el mismo archivo Word que llegó desde la autopista y llena en los campos de respuesta. Luego de haber modificado el documento Word, se envía como archivo adjunto de vuelta al cliente a través de un correo electrónico.

Existen tres niveles de solicitudes de servicio. Estos dependen según la urgencia con la cual se requiere resolver el problema, de manera de afectar lo menos posible el buen funcionamiento del sistema.

- Nivel 1: son alarmas que generan una incidencia cuya comunicación a Kapsch marca el inicio de una falta de disponibilidad en el punto de cobro.

- Nivel 2: son alarmas que no afectan la disponibilidad del punto de cobro, pero generan una incidencia cuya comunicación a Kapsch marca el inicio de la medición del tiempo para la reparación TTR (Time To Repair).
- Nivel 3: son alarmas que no afecta la disponibilidad del punto de cobro y no requiere un tiempo definido para su reparación, y su tratamiento será acordado entre las partes en el momento de la ocurrencia.

Si la solicitud de servicio corresponde a una solicitud de nivel 3, en la mayor parte de los casos es derivada al equipo de Ingeniería. Si la solicitud de servicio es de nivel 1 o 2, es atendida por el equipo de operaciones.

El documento que actualmente se está usando para registrar las actividades asociadas a la solicitud de servicio se encuentra en el anexo B. La planilla de resumen de solicitudes de servicio (ver anexo C), contiene información sobre cada solicitud de servicio, el responsable, la fecha de creación, la fecha de término, horas trabajadas, horas de espera y horas de transporte. Esta información resumida se utiliza para realizar el cálculo de valores de cobro por la ejecución de cada solicitud. Además, esta información sirve como información estadística y de control.

3.2 Problemática del procedimiento anterior

Entre los problemas que existían anteriormente con el tratamiento de las SDS se destacan los siguientes.

- No era posible manejar los estados intermedios de una solicitud, por ejemplo, en que fase de análisis de resolución del problema se encuentra y que labores se han realizado dentro de cada fase. Los únicos estados que se manejaban eran “abierta” (solicitud que aún no ha sido resuelta) y “cerrada” (solicitud ya resuelta).
- No existía control sobre las solicitudes que están en vías de ser cerrada, es decir, se ha enviado un Email al cliente solicitando el cierre, pero esta información no queda registrada en ninguna parte, excepto en el Email de la persona que solicita el cierre.

- El procedimiento anterior no contempla la posibilidad de que una solicitud de servicio fuese rechazada. El registro de los cambios o nuevos análisis para mejorar el resultado de la solicitud quedan registrados en el documento (ver anexo B, documento SDS), pero la solicitud con la respuesta inicial presentada al cliente se pierde.
- Una vez que la solicitud pasaba al equipo de Ingeniería, no es posible que el equipo de operaciones pueda saber cuál es el estado de la solicitud, ni tampoco que tipos de análisis se han realizado.
- No era posible mantener un control de los cambios de equipamientos registrados en un SAR, y la información contenida en el Unit Tracer, por ejemplo ver si el cambio de equipos fue registrado en el sistema.

Existen otras funcionalidades que el sistema anterior no poseía, por ejemplo no era posible obtener de manera rápida y simple los reportes de SDS cuando estos eran solicitados (ver tipo en anexo A, reporte de SDS). Para esto, una persona debía hacerse cargo mensualmente de realizarlos manualmente.

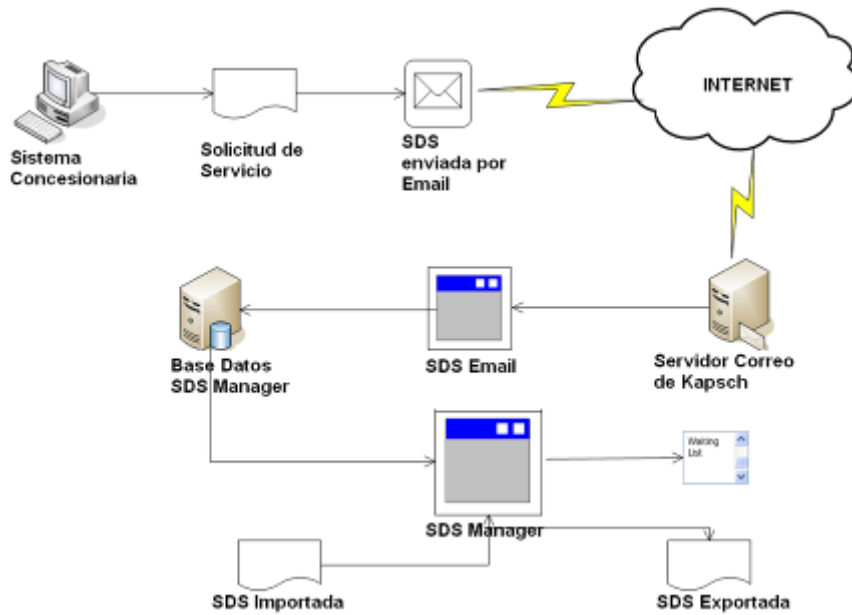
3.3 Solución Utilizada

Para solucionar la problemática presentada anteriormente, se planteó realizar una aplicación de escritorio en la cual permita que todas las SDS que se generen se puedan manejar desde el sistema, sin la necesidad de tener que almacenar un archivo Word para cada una. Esto agilizó mucho el proceso y ha permitido realizar estadísticas en forma automatizada y no manualmente como se realizaba anteriormente.

La solución consta de dos aplicaciones, una será el SDS Manager en sí y una segunda aplicación que rescata las SDS desde una casilla Email automáticamente. Esta segunda aplicación se llama SDS Email y está 24 horas al día y 7 días a la semana ejecutándose en el servidor, importando automáticamente las SDS desde el correo de mantenimiento.

En la figura número 4 se muestra el funcionamiento de los sistemas SDS Manager y SDS Email.

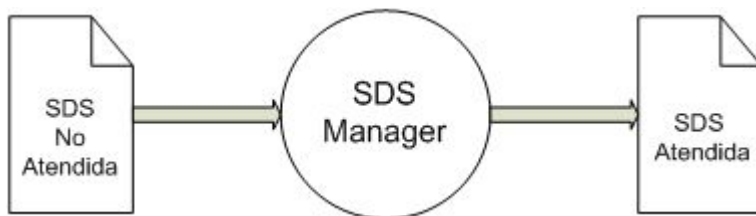
Figura 4: Esquema de flujo de documentos.



En el caso que el sistema de importación automática desde el correo de mantenimiento no funcionara, existe una función llamada IMPORT SDS, la cual permite importar desde el mismo SDS Manager los documentos Word para su utilización.

La figura número 5 muestra la importación de los documentos y su posterior exportación.

Figura 5: Importación y Exportación de Documentos.



CAPITULO 4: Análisis de Factibilidad y Riesgos

El estudio de factibilidad ayuda a la decisión de la realización de un proyecto. Para esto fue necesario aclarar la disponibilidad de recursos financieros, ya que sin éstos se hace imposible la implementación del sistema, los recursos tecnológicos que se requieren y la capacidad operacional que existe en la institución, los recursos legales que son importantes para poder implementar y desarrollar el sistema. En consecuencia, se debe evaluar cada uno de los puntos anteriormente señalados para así poder tener una visión en cuanto a la posibilidad real de implementación del proyecto.

4.1 Análisis de Factibilidad Técnica

Verifica los recursos tecnológicos necesarios para la implantación del sistema, estudia si el trabajo para el proyecto puede desarrollarse con el software, hardware y personal existente.

Para poder realizar un Análisis de Factibilidad Técnica más adecuada, se subdividió en dos fases: un análisis de Recursos de Software, y un análisis de Recursos de Hardware.

4.1.1 Análisis de Recursos de Software

En la actualidad, todos los equipos que existen en la empresa, poseen el sistema operativo Windows XP Profesional, el cual es totalmente compatible con la plataforma de programación la cual se utilizó para la implementación del SDS Manager.

Es válido señalar que todos los equipos que se utilizan en la empresa poseen sus respectivas licencias de Microsoft, tales como Windows XP Profesional y Microsoft Office.

Para poder realizar las exportaciones a documentos Word o Excel, es necesario utilizar una herramienta de Microsoft Office. Es válido señalar que en la empresa se dispone de Microsoft Office 2003, el cual es absolutamente compatible con la aplicación.

4.1.2 Análisis de Recursos de Hardware

La totalidad de equipos que se disponen en Kapsch, poseen recursos suficientes para la implementación del sistema. Se cuenta con equipos Pentium IV de 1.6 Gigahertz, 512 en RAM y 40 Gigas de Disco Duro.

La configuración de la red interna de la empresa, también facilita la comunicación del Sistema SDS Manager, ya que posee suficiente velocidad para que los datos se puedan transmitir de forma adecuada.

4.2 Análisis de Factibilidad Operacional

Todas las personas que utilizan el sistema son profesionales y técnicos del área informática y electrónica, los cuales poseen conocimientos más que básicos de computación.

También es válido señalar que no es el primer sistema o aplicación de escritorio que utilizará en esta área, por lo tanto están familiarizados con el uso de sistemas de este tipo.

4.3 Análisis de Factibilidad Económica

El análisis de factibilidad Económica se refiere a todos los gastos que se debieron incurrir en la construcción de este sistema. Los mayores gastos en que se podría haber incurrido hubiera sido la de adquirir las licencias de software de los sistemas operativos y de las herramientas de desarrollo. Como todas las licencias se encuentran disponibles en la empresa, esto no es un impedimento para la realización del sistema.

El otro tipo de gasto en que la empresa tuvo que incurrir, fue el pago de los honorarios del desarrollador del sistema. Estos fueron bajos, ya que el desarrollador fue contratado como alumno en práctica por la misma para la realización del sistema.

En conclusión, no existió ningún impedimento de tipo económico para el desarrollo del sistema.

4.4 Análisis de Factibilidad Legal

El objetivo de este análisis, es verificar si el proyecto que se realizó cumple con las normativas y leyes existentes para poder llevar a cabo el sistema. Con respecto a esto, el sistema no viola ninguna regla ni incurre en infracciones que impidan el buen funcionamiento de éste.

En Chile desde 1993 existe una ley de delitos informáticos que debe respetarse, la que se detalla a continuación.

Ley N°19.223: Ley relativa a delitos informáticos, la cual señala en sus cuatro artículos que aquel que modifique, altere, difunda los datos contenidos en el sistema o bien se apodere o use indebidamente la información será sancionado.

4.5 Conclusión de Factibilidad

Dado a todos los análisis que se han realizado, se pudo concluir que este es un proyecto absolutamente viable, ya que no presentó dificultades de ninguna de ningún tipo. La empresa posee absolutamente todos los requerimientos de factibilidad y pudo llevarse a cabo completamente.

4.6 Análisis de Riesgo

El análisis de riesgo da a conocer las distintas posibilidades por lo cual algún proyecto puede fallar. Para tomar alguna decisión, lo primero y más importante que se debe considerar son los riesgos del proyecto. ¿Cuáles son los factores que lo pueden hacer fallar? Cuanto mayor sea el riesgo, hay que prestarle mayor atención.

En general, los riesgos se pueden clasificar en 4 categorías

1. Riesgos de Requerimientos: ¿Cuáles son los requerimientos del sistema? El gran peligro es que se construya el sistema erróneo, un sistema que no haga lo que quiere el cliente. En la etapa de elaboración es donde se entienden bien los requerimientos y sus prioridades relativas. Los requerimientos son importantes y es donde las técnicas de UML

son especialmente provechosas. El punto de partida son los casos de uso. Éstos, por lo tanto son los motores de todo el proceso de desarrollo.

2. Riesgos Tecnológicos: ¿Cuáles son los riesgos tecnológicos que habrá que enfrentar? En este caso, conviene plantearse la siguiente pregunta: ¿Se tiene la experiencia suficiente para realizar este trabajo? Lo más importante que hay que hacer al abordar los riesgos tecnológicos es construir prototipos que prueben las partes tecnológicas con las que uno piensa trabajar.

3. Riesgos Funcionales: ¿Se puede tener la asesoría y los expertos que se requieren? Para este caso en particular se pretende hacer un breve entrenamiento, esto es, una clase para que, los que ocupen este sistema, dominen por lo menos lo básico de su utilización.

4. Riesgos Políticos: ¿Existen fuerzas políticas que se puedan interponer en el camino y afectar seriamente el proyecto?

Existen diversas maneras de poder tratar los riesgos, pero antes, estos riesgos tienen que ser identificados y priorizados, para que posteriormente el equipo de desarrollo decida como tratarlos, básicamente cuentan con cuatro elecciones.

- Evitarlo: algunos riesgos pueden y deberían evitarse, quizá mediante una replanificación del proyecto o un cambio en los requisitos.
- Limitarlo: otros riesgos deberían restringirse de modo que solo afecten a una pequeña parte del proyecto.
- Atenuarlo: existen riesgos que pueden atenuarse ejercitándolos y observando si aparecen o no. Si aparece, el aspecto positivo es que el equipo ha aprendido más sobre él.
- Controlarlo: hay riesgos que no pueden atenuarse. Lo único que puede hacer el equipo es controlarlos y observar si aparecen. Si aparecen, el equipo debe seguir un plan de contingencia. En caso de que aparezca un riesgo crítico, se debe analizar la situación y preguntarse si se debiese parar el proyecto, ya que sólo se ha gastado una cantidad de recursos limitada.

La importancia de priorizar los riesgos radica en que el tratamiento implica tiempo, por lo que en una organización raramente puedan tratarlo todos a la vez.

La tabla número 1 muestra los posibles riesgos, la probabilidad de ocurrencia y efecto en el sistema así como el plan de contingencia para disminuir dichos riesgos.

Riesgos	Probabilidad y Efecto	Efecto	Plan de contingencia
El proyecto no sea factible de realizar	Baja	Muy Grave	Investigar sobre las distintas opciones que existen para que el proyecto se pueda realizar, ya sea por problemas de costos, tiempo, etc
Desconocimiento de las herramientas de desarrollo.	Alta	Grave	Estudiar las herramientas con anticipación a las etapas de construcción.
El tiempo requerido para desarrollar el software esta subestimado	Moderado	Grave	Utilizar herramientas de fácil uso para el desarrollador del sistema (C++ Builder) y poseer buena cantidad de información de la herramienta.
El tamaño del software esta subestimado	Moderado	Mediano	Realizar una buena delimitación de los requerimientos del sistema.
Problemas entre la			Crear vías de comunicación lo

comunicación que efectúan las bases de datos	Bajo	Mediano	más eficientemente posibles entre las bases de datos que interactúan con el sistema.
Se proponen cambios en los requerimientos que requieren rehacer el diseño	Muy Bajo	Grave	Valorar el impacto de los requerimientos nuevos y maximizar la información oculta en ellos.

Tabla 1: Posibles Riesgos

CAPITULO 5: Paradigma y Metodología

5.1 Paradigma

Para abordar la problemática y los requerimientos del sistema, para la realización del Sistema SDS Manager se utilizó el Modelo Lineal Secuencial. El modelo lineal secuencial sugiere un enfoque sistemático para el desarrollo de software que comienza con un nivel de sistema, progresa con el análisis, diseño, codificación, prueba y mantenimiento [1].

5.1.1 Análisis del sistema

Debido a que el software es siempre parte de un sistema mayor, el trabajo comienza estableciendo los requisitos de todos los elementos del sistema y luego asignando algún subconjunto de estos requisitos al software [1]. Esta parte abarca los requisitos globales a nivel del sistema con una pequeña parte de análisis y diseño a un nivel superior.

5.1.2 Análisis de los requisitos del software

Para comprender la naturaleza de los programas que hay que construir, el ingeniero de software debe comprender el ámbito de la información del software, así como la función, el rendimiento y las interfaces requeridas [1]. Los requisitos, tanto del sistema como del software, se documentan y se revisan con el cliente.

5.1.3 Diseño

El diseño del software es un proceso de muchas etapas que se enfoca sobre cuatro atributos distintos: la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz [1]. Al igual que en el análisis de los requisitos, el diseño se documenta y forma parte de la configuración del software.

5.1.4 Codificación

Aquí el diseño debe traducirse en una forma legible para la máquina. Si el diseño se realiza de una manera detallada, la codificación puede realizarse mecánicamente [1].

5.1.5 Prueba

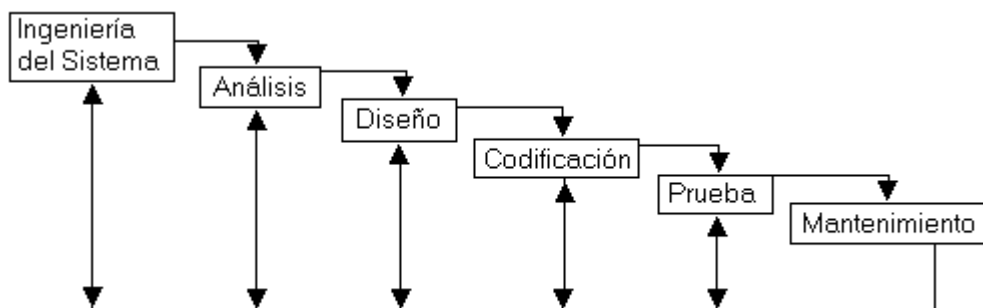
Una vez que se ha generado el código, comienza la prueba del programa. La prueba se centra en la lógica interna del software, asegurando que todas las sentencias se han probado, y en las funciones externas, realizando pruebas que aseguren que la entrada definida producen los resultados que realmente se requieren [1].

5.1.6 Mantenimiento

El software indudablemente, sufrirá cambios después de que se entregue al cliente. Los cambios ocurrirán debido a que se hayan encontrado errores, a que el software deba adaptarse a cambios del ambiente externo, o debido a que el cliente requiera ampliaciones funcionales o del rendimiento [1]. El mantenimiento del software aplica cada uno de los pasos precedentes del ciclo de vida de un programa existente en vez de a uno nuevo.

A continuación la figura número 6 muestra el ciclo de vida clásico

Figura 6: Ciclo de Vida Clásico [2].



5.2 Metodología Utilizada

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software.

En este caso, para el desarrollo del sistema SDS Manager se utiliza Orientación a Objetos, lo cual facilitará la reutilización de componentes y otorgará la posibilidad de utilizar todas las propiedades que ofrece esta metodología.

La programación Orientada a Objetos es una forma de pensar, una filosofía de lo cual surge una cultura nueva que incorpora técnicas y metodologías diferentes.

Se utilizó UML [3] para visualización, especificación y documentación de cada una de las partes que comprende el desarrollo de este sistema.

UML (Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema de software orientado a objetos. El principal objetivo de la creación de UML es posibilitar el intercambio de modelos entre las distintas herramientas CASE orientadas a objetos del mercado. Para ello era necesario definir una notación y una semántica en común, y para esto se creó UML [3].

5.3 Arquitectura planteada como solución

La arquitectura planteada como solución consiste en una aplicación distribuida. En esta arquitectura la capacidad de proceso está repartida entre el servidor y los clientes. La administración de datos es remota y las principales características de esta administración son:

- En el cliente residen tanto la interfaz como los procesos de la aplicación.
- Las bases de datos están en el servidor.

Las ventajas de utilizar una administración de datos remota son:

- Ser muy adecuada para las aplicaciones de apoyo a las decisiones del usuario final.

- Facilidad de desarrollo y mantención de las aplicaciones, debido a la independencia de la lógica de los datos con respecto a los programas que manipulan dicha información.
- Se descargan los programas del HOST.

Este tipo de arquitectura distribuida, es una forma de compartir una aplicación de escritorio en una red. Así cada uno de los computadores conectados a esta red, pueden ingresar al sistema.

Las funciones de un programa servidor son:

- Esperar las solicitudes de los clientes.
- Ejecutar muchas solicitudes al mismo tiempo.
- Empezar y operar actividades de tareas en segundo plano.
- Mantenerse activa en forma permanente.

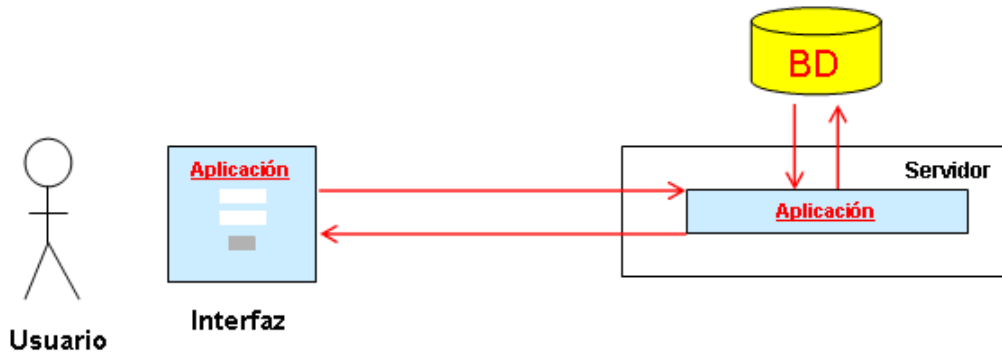
En la funcionalidad de un programa distribuido se pueden distinguir 3 capas o niveles, siendo estos:

1. Procesador de aplicaciones o reglas del negocio (Nivel lógico)
2. Manejador de Base de Datos (Nivel de almacenamiento)
3. Interfaz del usuario (Nivel de presentación)

5.4 Nivel Lógico y de Almacenamiento

En la figura número 7 se muestra el diagrama de arquitectura lógica que se utilizará para este proyecto.

Figura 7: Diagrama de la Arquitectura Lógica.



Cuando el usuario o Administrador inicie una sesión, el sistema despliega una interfaz donde se ingresa el LOGIN y su PASSWORD en la aplicación. Una vez ingresados los datos requeridos, el sistema recibe los datos y pide la confirmación al sistema, el cual recoge la información y la corrobora con la BASE DE DATOS, que contiene la información de los distintos usuarios y a qué perfil de usuario corresponden los datos que se están verificando. Finalizada la confirmación se devuelve a la aplicación de escritorio la información del tipo de usuario al que corresponde, desplegando la interfaz correspondiente.

5.5 Componentes Lógicos y de Almacenamiento

Este sistema tiene una sola interfaz: SDS Manager. Esta es una aplicación de escritorio por un tema de seguridad y políticas internas de la empresa.

Las bases de datos del sistema están ubicadas en un servidor situado en las oficinas centrales de la empresa. Esta aplicación también puede ser ejecutada desde la casa matriz en Suecia remotamente vía VPN interna.

La base de datos que se utiliza es Microsoft Access y la aplicación fue diseñada con C++ Builder 5.0 junto con unas librerías internas de la empresa. La razón de la elección se basa simplemente por las políticas internas de la empresa y porque se poseen licencias de ambas herramientas. C++ Builder es una herramienta algo antigua, pero es muy poderosa y estable a la hora de realizar Software de este tipo. Es válido señalar que en la empresa existe un programa llamado Unit Tracer el cual fue realizado con esta tecnología.

El servidor en el cual se almacena la aplicación y la base de datos es uno interno que ya existe en la empresa, por lo cual no es necesario de un hardware adicional para este fin.

En cuanto a los usuarios, sin importar de qué tipo de usuario se trate, necesitan de un computador compatible equipado con los estrictos parámetros de seguridad para estar en la red interna de esta empresa.

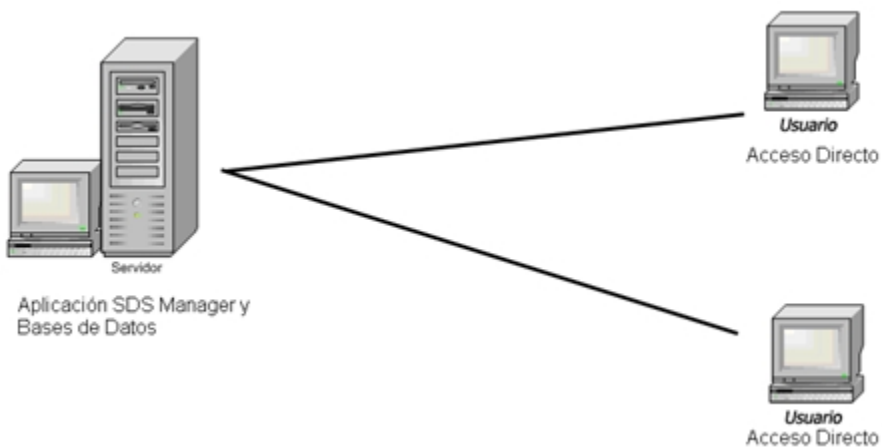
5.5.1 Nivel de Presentación

La GUI (interfaz gráfica de usuario) permite el despliegue en color de alta resolución e interacción utilizando el mouse. Esto permite dar ventaja ya que son relativamente fáciles de aprender y usar para interactuar con el sistema.

Durante la captación de requerimientos también se analizaron las actividades de usuario que fueron implementadas por el sistema. La interfaz utiliza términos familiares para los usuarios y los objetos que el sistema manipula deben estar directamente relacionados con su entorno. Debe haber familiaridad, consistencia, y mínima sorpresa.

En la figura número 8 se muestra la arquitectura física que posee el sistema.

Figura 8: Arquitectura física del sistema.



Disposición física de las entidades (mínimas) involucradas en el Sistema:

- Terminal(es) Usuario(s), en donde se ejecutan las aplicaciones.

- Servidor, donde se almacena el programa y sus bases de datos.

CAPITULO 6: Desarrollo

En esta fase del proyecto, se comenzará detallando el análisis de requerimientos funcionales, no funcionales y posteriormente el desarrollo de la implementación del sistema.

6.1 Requerimientos Funcionales

Para captar y reunir los requerimientos, en una etapa inicial se realizaron una serie de reuniones con el cliente, las que además permitieron observar el funcionamiento interno de la empresa, pudiendo así, tener una visión mucho más clara del entorno del negocio donde se usará el sistema que se desarrolla.

6.1.1 Requerimientos Funcionales con Interfaces Externas

En la figura número 9 se muestra un diagrama de los diferentes sistemas anexos que interactuarán con el SDS Manager, y en la tabla 2 se detallan sus componentes.

Figura 9: Interfaces Externas

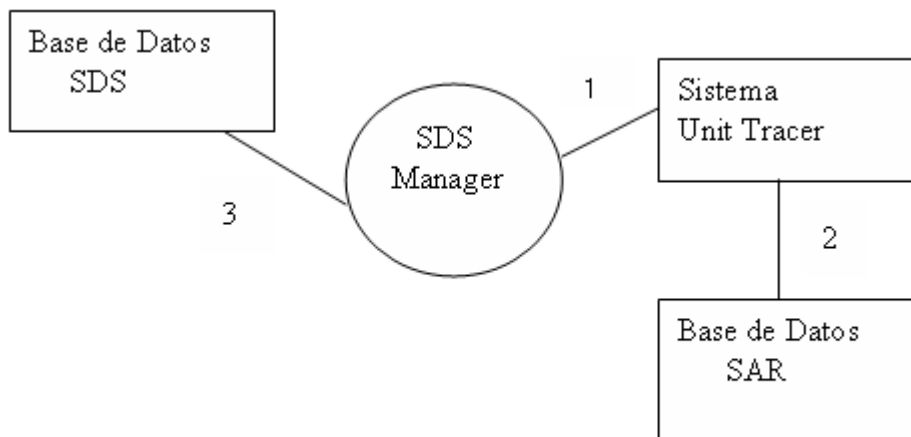


Tabla 2: Interfaces Externas

Nro	Nombre de Interfase	Descripción de Interfase
1	SDS Manager– Sistema Unit Tracer	SDS ingresadas como solicitudes de cambio de equipamiento en el sitio.
2	Unit Tracer – Base de datos SAR	La base de datos de SAR es consultada y la información es mostrada en Unit Tracer.
3	SDS Manager – Bases de Datos SDS	Es la conexión del sistema con la base de datos que contiene las Solicitudes de servicio

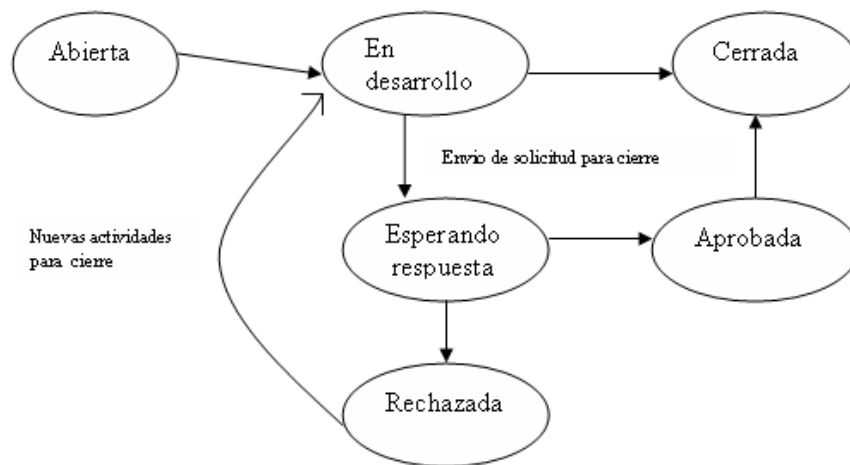
R01.	<p>El sistema deberá contar con una interfaz a la Base de Datos SDS. Que cumplan las siguientes funcionalidades.</p> <ul style="list-style-type: none"> - Ingreso o creación de una nueva solicitud de servicio (SDS) por parte del cliente - Actualización de estado de la SDS - Actualización de las actividades, observaciones u otros registros asociados a la SDS - Búsqueda de una o varias SDS según
------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	estado, tipo de alarma, nivel de severidad, sitio y número.
R02.	<p>El sistema tendrá una interfaz al Sistema Unit Tracer, con las siguientes funcionalidades.</p> <ul style="list-style-type: none"> - En el registro de cambio será posible agregar en el detalle del ítem, la información que aparece en el sistema Unit Tracer (ver anexo D). - Será posible llamar al Sistema Unit Tracer a través del SDS Manager, con un botón de opciones.
R03.	El sistema deberá ser capaz de vincular una o varias SAR asociadas a una SDS.

6.1.2 Requerimientos de Estados de una SDS

R04.	Las SDS pasarán por los estados mostrados en la figura número 10.

Figura 10: Estados de una SDS



- R05. Una solicitud tomará siempre como estado inicial el estado “abierta”.
- R06. El sistema muestra la información de los estados y la descripción de cada una.
- R07. La solicitud estará en estado “En desarrollo” cuando al menos se realice una primera actividad de desarrollo, ya sea, creación de SAR, creación de SDS, en espera de respuesta desde Suecia, etc.
- En el caso de que una SDS haya sido rechazada, se realizarán nuevas actividades para su cierre, en tal caso, la SDS toma nuevamente el estado de “En desarrollo”.
- R08. Una SDS queda en estado “Esperando Respuesta” cuando las SDS han sido enviadas al cliente para solicitar su cierre.

En parte de los casos, la solicitud de cierre no se realiza, y sólo se avisa al cliente que ha sido cerrada.

- R09. Las SDS quedan en estado “Aprobada”, cuando la respuesta del cliente ha sido de aprobación.

En parte de los caso, después del estado “En Desarrollo” la SDS es cerrada automáticamente, sin esperar una respuesta de aprobación.

- R10. Las SDS quedan en estado “Cerrada”, cuando se ha ingresado la información al sistema relacionada con el cierre.

- R11. Las SDS quedan en estado de “Rechazada” cuando el cliente no ha aprobado la respuesta o no está conforme con las actividades que se realizaron para su cierre.

- R12. Una solicitud de nivel 1,2 y 3 está “En Desarrollo” cuando ésta ha sido revisada por parte del Grupo de Mantenimiento, para ver posibles detalles de redacción de parte del cliente. Además, se realiza el trabajo de resolución en terreno.

6.1.3 Flujo de Datos de SDS Manager

- R13. Los Emails que lleguen desde VNE, CN o ANS, con nuevas solicitudes de

servicio, serán la entrada al sistema SDS Manager.

- R14. La nueva solicitud de servicio será ingresada en forma manual, es decir los siguientes datos serán copiados desde la solicitud (documento en Word) al nuevo sistema.

Correlativo
Lugar de ejecución
Equipo o sistema afectado
Descripción de la incidencia
Fecha y Hora de reporte
Tipo de reporte
Nivel de severidad
Fecha y hora de la incidencia
Estado inicial de la SDS
Nombre del que reporta la incidencia
Nombre de la persona a quién se reporta
Firma
Empresa

- R15. La solicitud de servicio puede ser leída automáticamente desde el Email. Existirá un proceso que tomará la información desde el documento en Word y la traspasará a la Base de Datos del SDS Manager. Además, la información adicional que pueda traer el Email, será también leída y guardada en la Base de

Datos.

- R16. Si el proceso de lectura automática del Email falla, se emitirá un nuevo Email confirmando que la información no ha sido ingresada al SDS Manager. La información de la solicitud de servicio deberá ser ingresada de forma manual.

6.1.4 Perfiles de Usuario

- R17. La Base de Datos del SDS Manager, contendrá un listado de los usuarios autorizados a usar el sistema SDS Manager.

- R18. Los niveles de acceso serán los siguientes.

- Acceso sólo lectura
- Acceso lectura/escritura
- Administrador

6.1.5 Modo de Inicialización

- R19. El SDS Manager deberá ser capaz de conectarse a la Base de Datos SDS.

- R20. El SDS Manager deberá ser capaz de conectarse al sistema Unit Tracer.
- R21. El sistema deberá permitir la conexión a la Base de Datos del SAR.
- R22. En caso de que el SDS Manager no logre conectarse a la Base de Datos SDS, el usuario será avisado sobre el problema, y el sistema se cerrará.
- R23. Si el SDS Manager no logra conectarse al sistema Unit Tracer, el sistema seguirá trabajando sin la funcionalidad, y el usuario será avisado sobre este detalle.
- R24. Si el SDS Manager no logra conectarse a la Base de Datos SAR, el sistema seguirá trabajando y el usuario será avisado sobre este detalle
- R25. Si el SDS Manager logra conectarse con la Base de Datos, se verificará el nivel de acceso que tiene el usuario. Si el usuario no está registrado, el sistema emitirá un mensaje y luego finalizará.

6.1.6 Modo de Edición

- R26. El SDS Manager será capaz de funcionar en diversos modos.

6.1.7 Modos de Validación

R27. El SDS Manager tendrá diversos modos de chequeo de la información antes de que sea guardada en la Base de Datos SDS.

6.1.8 Modos de Impresión

R28. El sistema será capaz de producir copias de los reportes que se generen.

6.1.9 Modo de Búsqueda

R29. El sistema será capaz de permitir diversos modos de búsquedas de SDS.

- Búsqueda por nivel (1, 2 o 3).
- Búsqueda por fecha (fecha desde y hasta)
- Búsqueda por concesionaria

6.1.10 Modo de Salida

R30. Antes de salir del sistema, el SDS Manager, se desconectará de todos los sistemas de manera apropiada.

6.1.11 Dominios

En la tabla 3 se muestra como se deben manejar los diferentes dominios.

Dominio	Descripción de Requerimientos
Comunicación con la Base de Datos	Comportamiento de la Base de Datos para el sistema
Contenido y modelo de la Base de Datos	Requerimientos del contenido de la Base de Datos y su modelo entidad-relación
Vista	Requerimientos sobre la forma en que la información será desplegada
Validación de Datos	Requerimientos sobre los datos a validar y sobre como se va a validar
Impresión	Requerimientos sobre los reportes impresos que se pueden obtener

Tabla 3: Dominios

6.2 Requerimientos No Funcionales

Corresponde a aquellos requerimientos que recogen los atributos de calidad del sistema a desarrollar.

6.2.1 Respuesta en tiempo

El sistema debe permitir que las transacciones que realiza este programa sean en el menor tiempo posible. El tiempo de respuesta entre que se solicite una acción y se realice, no sea superior a 0.5 segundos.

6.2.2 Fiabilidad

El sistema debe entregar la información que se solicita de forma correcta. Debe ser estable y en un entorno dado para un propósito específico. Para poder lograr esto, se deben contar con consultas exactas a la Base de Datos.

6.2.3 Seguridad y Protección

Debe permitir el acceso restringido a los usuarios de acuerdo al perfil asignado a cada uno, con esto disminuir al máximo el riesgo que la información sea utilizada por personas de fuera de la empresa.

6.2.4 Disponibilidad

El sistema debe entregar la información a sus usuarios cuando éstos la soliciten, debe tener la capacidad de permanecer activo y funcionando. Esto se logrará gracias a que el servidor de la empresa deberá estar funcionando siempre para que pueda accederse al programa.

6.2.5 Configurabilidad

El sistema deberá ser capaz de almacenar cambios en su configuración, como por ejemplo, la posibilidad de cambiar el idioma de la interfaz.

6.2.6 Modularidad y Escalabilidad

El sistema debe poder soportar modificaciones si se requiere agregar alguna función, sin afectar mayormente a su arquitectura.

6.2.7 Interoperabilidad

El sistema deberá relacionarse en forma armónica con otros sistemas y arquitecturas.

6.2.8 Automatización

El sistema debe tener la máxima automatización para evitar, en lo posible el ingreso y recolección manual.

6.3 Casos de Uso

6.3.1 Actores

Los actores son entidades con un comportamiento definido dentro del Sistema, ya sea entidades que utilizan el Sistema para concretar un objetivo, así como, aquellas que proveen información al Sistema o están interesados en su comportamiento.

En la tabla 4 se muestran los diferentes actores que interactúan en el sistema.

Actor	Función	Tipo Cuenta (Nivel Acceso)	Personas
Administrador	Utiliza completamente el Software. Puede crear, importar, exportar SDS junto con generar todos los reportes, administrar las cuentas de usuarios entre otras funciones.	FULL	Gerente Operaciones, Personal de Ingeniería
Operario	Este tipo de Actor puede crear, importar, exportar, y administrar las SDS. Los actores de administradores pueden ejecutar	READ/WRITE	Personal de Operaciones y Administradores.

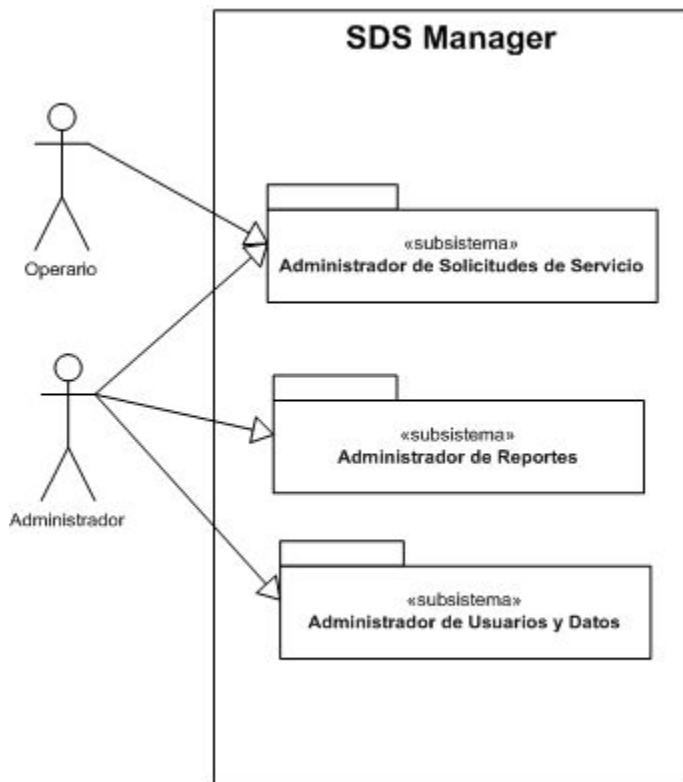
	las tareas de los operarios.		
--	------------------------------	--	--

Tabla 4: Actores del Sistema

6.3.2 Diagrama General UML

En la figura número 11 se muestra el diagrama general del Sistema SDS Manager.

Figura 11: Diagrama General UML

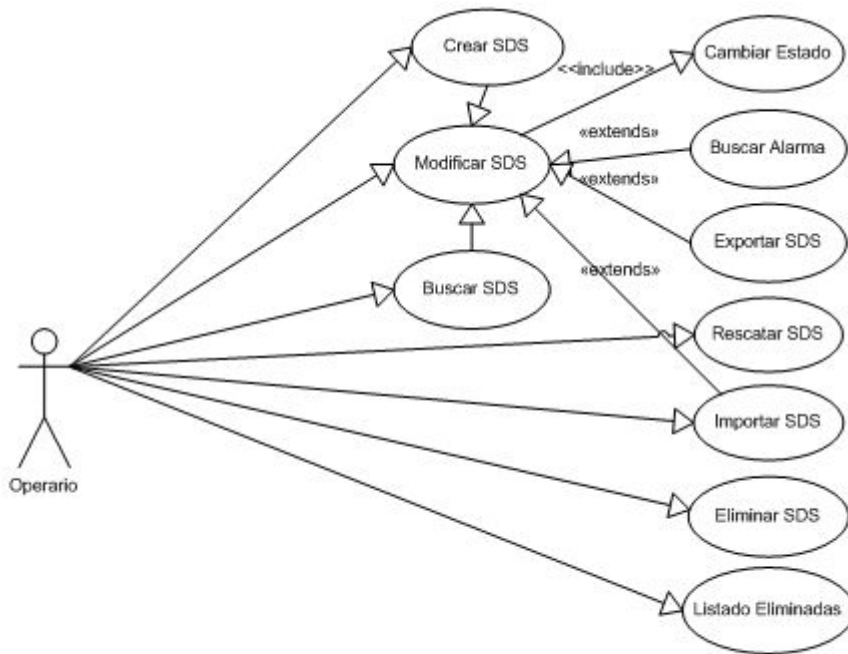


Administración de Solicitudes de Servicios (SDS)

En esta sección del informe se definirán los casos de uso referentes a la manipulación de las SDS del sistema.

La figura número 12 se muestra el diagrama de uso “Administración de Solicitudes de Servicio”.

Figura 12: Diagrama “Administración de Solicitudes de Servicio”



Nota: Al no existir diferencias para todos los Casos de Uso de este subsistema entre “Operario” y “Administrador”, en el texto se referencia solo al primer tipo.

6.3.2.1 Crear SDS

Este Caso de Uso representa la creación de una SDS manualmente, sin la necesidad de ser importada desde el documento Word. Esto se utilizará en el caso de que exista la necesidad de crear una SDS de una concesionaria antes que ella la envíe a la Empresa.

6.3.2.1.1 Especificación de Caso de Uso

En la tabla 5 se muestra la especificación de Caso de Uso.

Caso de Uso	Crear SDS
Actor Primario	Operario

Participante e intereses	Operario: Desea crear una nueva SDS que no se ingresó automáticamente al sistema
Precondiciones	El Operario se identifica en el sistema
Poscondiciones	El Operario almacena la información como SDS
Escenario Principal	<ol style="list-style-type: none"> 1.- Usuario selecciona la opción NEW SDS 2.- El sistema despliega interfaz NEW SDS 3.- Sistema despliega un combo de listado de proyectos 4.- Usuario selecciona el proyecto a crear la SDS 5.- El sistema muestra la el correlativo de la última SDS de ese proyecto 6.- El usuario acepta y se creará una nueva SDS 7.- Esta interfaz cierra y se abre la interfaz de MODIFICACION
Extensiones	<p>5 a- El sistema despliega un número SDS que no corresponde a la que se desea ingresar</p> <ul style="list-style-type: none"> - Usuario ingresa manualmente el número y el correlativo - Sistema vuelve al paso 6. <p>6 a- Usuario no desea acepta la creación de la nueva SDS</p>

	<ul style="list-style-type: none"> - Usuario solicita salir de esta interfaz - Sistema vuelve a interfaz anterior
--	-------------------------------------------------------------------------------------------------------------------------------------------

Tabla 5: Crear SDS

6.3.2.1.2 Diagrama de Secuencia

A continuación se muestra el Diagrama de Secuencia 1.

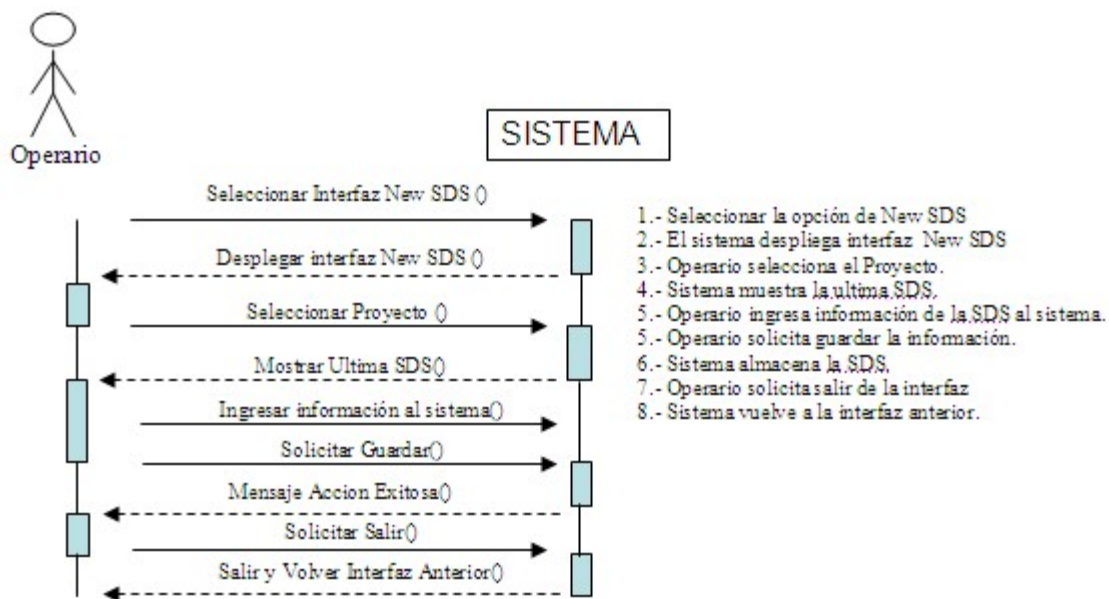
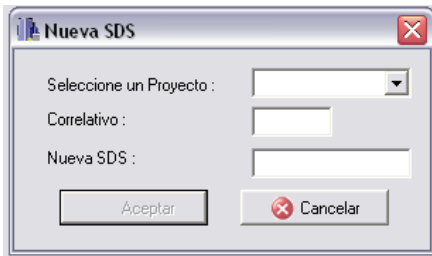


Diagrama de Secuencia 1: Crear SDS

6.3.2.1.3 Interfaz relacionada

La Ventana 1 representa la interfaz relacionada con el Caso de Uso.



Ventana 1: Nueva SDS

6.3.2.2 Modificar SDS

Este Caso de Uso representa una SDS en modo de modificación. En este modo puede ser visualizada, modificada y cambiado de estado. Este podría decirse que es el Caso de Uso más importante.

6.3.2.2.1 Especificación de Caso de Uso

En la tabla 6 se muestra la especificación de Caso de Uso.

Caso de Uso	Modificar SDS
Actor Primario	Operario
Participante e intereses	Operario: Desea modificar la información de una SDS previamente ingresada o que haya sido creada recientemente UnitTracer: Se solicita información sobre los últimos cambios de Sistemas o subsistemas
Precondiciones	El Operario se identifica en el sistema y se esta mostrando en pantalla la SDS que desea modificar
Poscondiciones	El Operario almacena o modifica la información de la SDS

<p>Escenario Principal</p>	<ol style="list-style-type: none">1.- El usuario visualiza la información de la SDS que desea modificar2.- El usuario ingresa y modifica la información de la SDS que se está mostrando3.- El usuario solicita guardar la información4.- El sistema almacena los cambios e informa de la transacción al usuario5- El sistema al almacenar, elimina la SDS que esta en el listado de espera (WAITING LIST) si es que esta proviniera de dicho listado6.- Sistema vuelve a interfaz anterior
<p>Extensiones</p>	<ol style="list-style-type: none">1 a- Usuario no es la SDS que solicita modificar<ul style="list-style-type: none">- Usuario abre el formulario de búsqueda- Sistema cierra la ventana de la SDS- Abre el formulario de búsqueda (SEARCH)2 a- Usuario no ingresó la información correctamente<ul style="list-style-type: none">- El sistema verifica cada una de las entradas de datos- Muestra el error en pantalla (como POPUP)2 b- Usuario solicita imprimir

	<ul style="list-style-type: none"> - Sistema genera impresión del local - Sistema vuelve a interfaz anterior <p>2 c- Usuario desea ingresar una alarma y no posee el código</p> <ul style="list-style-type: none"> - Abre el formulario de listado de alarmas <p>3 a- Usuario no desea guardar la información</p> <ul style="list-style-type: none"> - Usuario solicita salir de esta interfaz y El sistema informa que se realizaron cambios y que no han sido almacenados - Usuario se niega a Sistema vuelve a interfaz anterior
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 6: Modificar SDS

6.3.2.2.2 Diagrama de Secuencia

A continuación se muestra el Diagrama de Secuencia número 2.

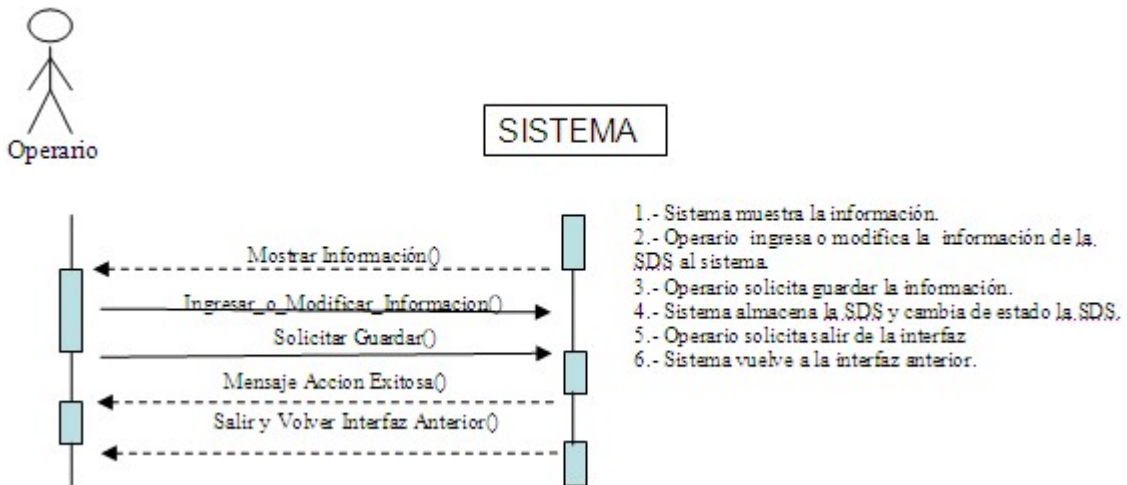
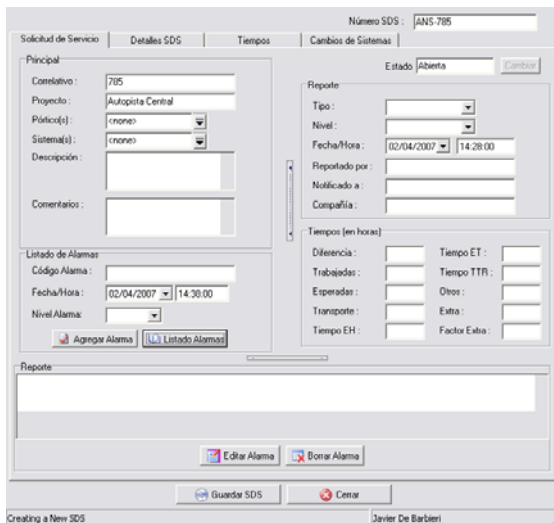


Diagrama de Secuencia 2: Modificar SDS

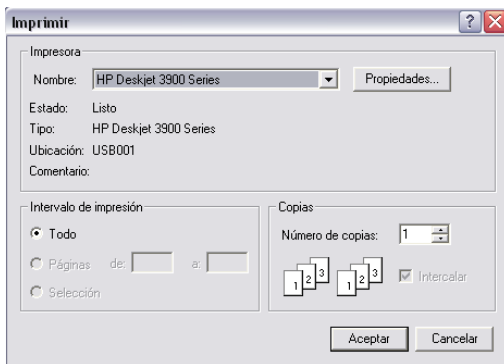
6.3.2.2.3 Interfaces relacionadas

Las ventanas 2 a 5 representan las interfaces relacionadas.

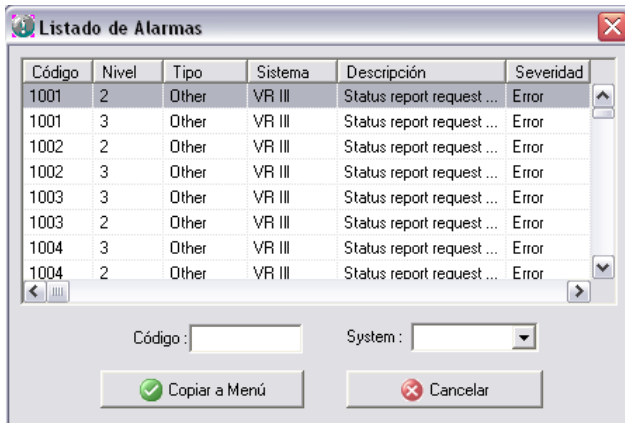


Ventana 2: Menú Principal

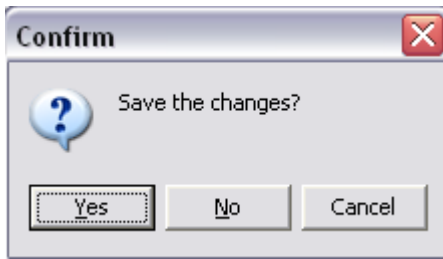
Ventana 3: Menú de Impresión



Ventana 4: Listado de Alarmas



Ventana 5: Confirmación de Guardar Cambios



6.3.2.3 Rescatar SDS

Este Caso de Uso es representado el rescate de una SDS que está en el listado de WAITING LIST para poder ser respondida.

6.3.2.3.1 Especificación de Caso de Uso

En la tabla 7 se muestra la especificación de Caso de Uso.

Caso de Uso	Rescatar SDS
Actor Primario	Operario

Participante e intereses	Operario: Desea rescatar desde el listado de WAITING LIST una SDS que no ha sido atendida
Precondiciones	El Operario se identifica en el sistema
Poscondiciones	El Operario almacena la información como SDS y es borrada la SDS del WAITING LIST
Escenario Principal	<ol style="list-style-type: none"> 1.- Usuario selecciona la opción WAITING LIST 2.- El sistema despliega interfaz WAITING LIST 3.- Sistema despliega lista de SDS no atendidas 4.- Usuario selecciona desde el listado de SDS la que desea atender 5.- El sistema muestra la información correspondiente a la SDS, junto con realizar varias verificaciones y correcciones a los datos 6.- Usuario visualiza la información de la SDS 7.- Usuario ingresa la información de Detalle de SDS y demás campos 8.- Usuario solicita guardar la información 9.- Usuario cierra ventana de SDS 10.- Sistema vuelve a interfaz anterior
Extensiones	4 a- Usuario no encuentra entre las SDS la que el que busca

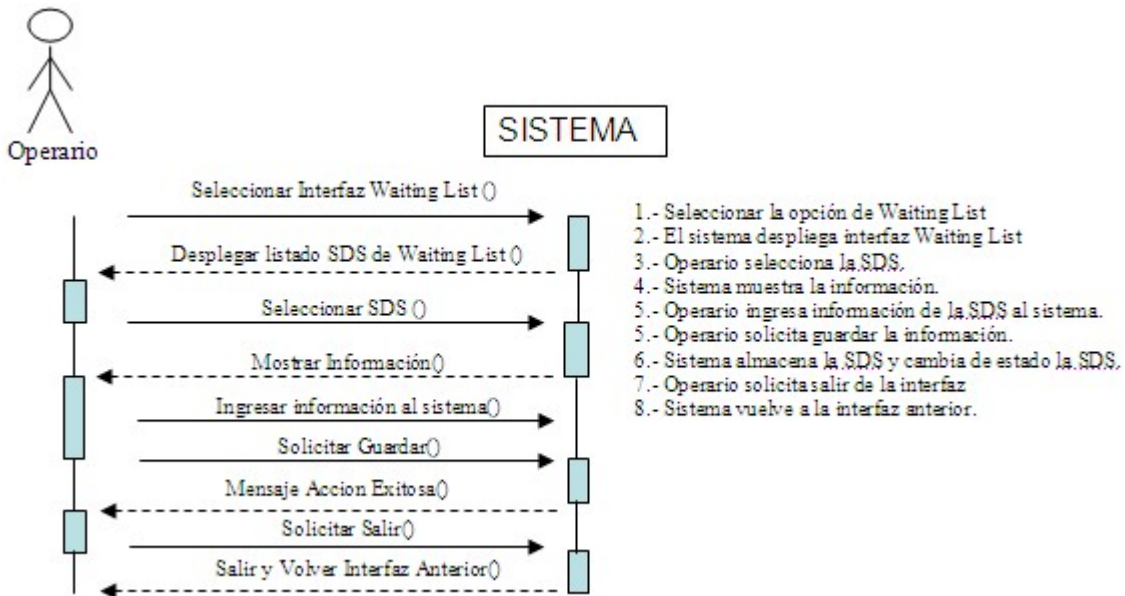
	<ul style="list-style-type: none"> - Usuario solicita salir de esta interfaz - Sistema vuelve a interfaz anterior
6 a-	<p>Usuario solicita imprimir</p> <ul style="list-style-type: none"> - Usuario impresión del local - Sistema vuelve a interfaz anterior
8 a-	<p>Usuario no desea guardar la información</p> <ul style="list-style-type: none"> - Usuario solicita salir de esta interfaz - Sistema vuelve a interfaz anterior

Tabla 7: Rescatar SDS

6.3.2.3.2 Diagrama de Secuencia

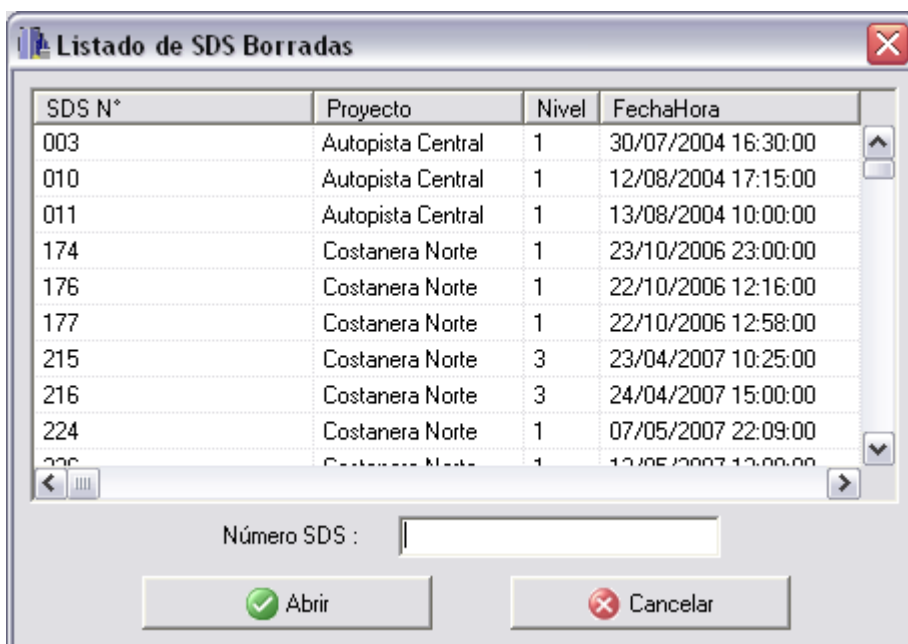
A continuación se muestra el Diagrama de Secuencia 3.

Diagrama de Secuencia 3: Rescatar SDS



6.3.2.3.3 Interfaces relacionadas

La Ventana 6 representa la interfaz relacionada con el Caso de Uso.



Ventana 6: Listado de SDS Borradas

6.3.2.4 Importar SDS

Este Caso de Uso representa la importación desde un documento Word hasta el sistema de SDS Manager.

6.3.2.4.1 Especificación de Caso de Uso

En la tabla 8 se muestra la especificación de Caso de Uso.

Caso de Uso		Importar SDS
Actor Primario		Operario
Participante	e	Operario: Desea rescatar desde el listado una SDS en archivo Word

intereses	hasta el sistema
Precondiciones	El Operario se identifica en el sistema
Poscondiciones	El Operario almacena la información como SDS dentro del sistema
Escenario Principal	<ol style="list-style-type: none"> 1.- Usuario selecciona la opción IMPORT SDS 2.- El sistema despliega interfaz IMPORT SDS 3.- Sistema despliega un formulario para ubicar el archivo a importar 4.- Usuario selecciona el documento Word a importar 5.- El sistema verifica si ya existe en WAITING LIST 6.- El sistema extrae del documento Word la información necesaria que contiene y la muestra en la pantalla del Software 7.- Usuario ingresa o modifica la información de la SDS y demás campos 8.- Usuario solicita guardar la información 9.- Usuario cierra ventana de SDS 10.- Sistema vuelve a interfaz anterior
Extensiones	<ol style="list-style-type: none"> 4 a- Usuario no encuentra entre las SDS la que el que busca - Usuario solicita salir de esta interfaz - Sistema vuelve a interfaz anterior

	<p>5 a- La SDS existe en el sistema</p> <ul style="list-style-type: none"> - Se muestra en pantalla un mensaje que se está ingresada - Se modifica la información almacenada en WAITING LIST y vuelve <p>6 a- Usuario solicita imprimir</p> <ul style="list-style-type: none"> - Sistema genera impresión del local - Sistema vuelve a interfaz anterior <p>8 a- Usuario no desea guardar la información</p> <ul style="list-style-type: none"> - Usuario solicita salir de esta interfaz - Sistema vuelve a interfaz anterior
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 8: Importar SDS

6.3.2.4.2 Diagrama de Secuencia

A continuación se muestra el diagrama de secuencia 4 asociado a este Caso de Uso.

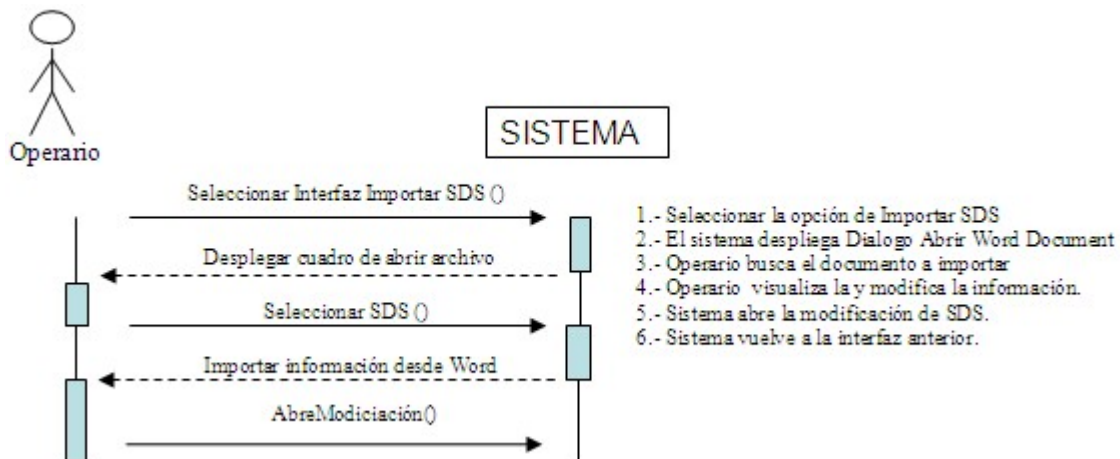
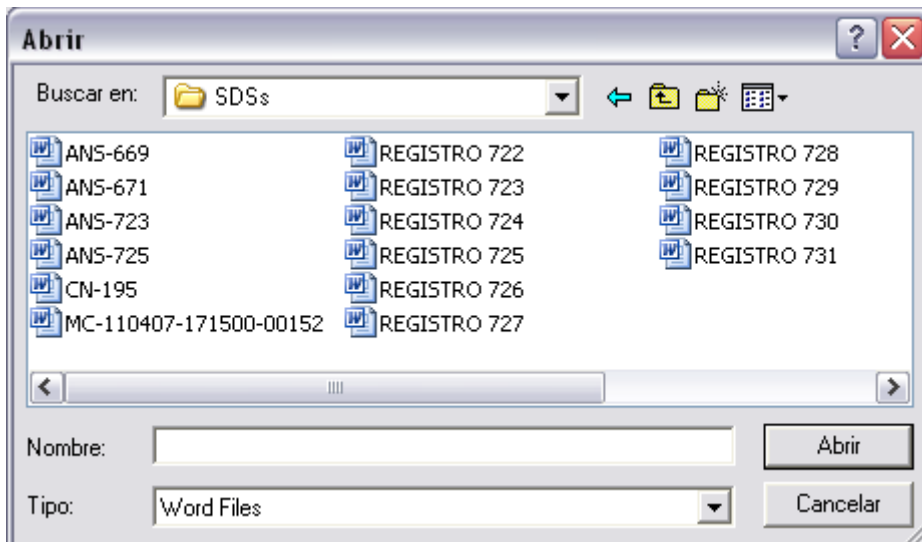


Diagrama de Secuencia 4: Importar SDS

6.3.2.4.3 Interfaces relacionadas

En la Ventana 7 se muestra la interfaz relacionada con el Caso de Uso.



Ventana 7: Importar SDS

6.3.2.5 Exportar SDS

Este Caso de Uso tiene relación con la exportación de una SDS en un documento Word que posteriormente será enviado a la concesionaria.

6.3.2.5.1 Especificación de Caso de Uso

En la tabla 9 se muestra la especificación de Caso de Uso.

Caso de Uso	Exportar SDS
Actor Primario	Operario

Participante e intereses	Operario: Desea exportar una SDS a formato Word
Precondiciones	El Operario se identifica en el sistema y tiene una SDS abierta
Poscondiciones	El Operario obtiene como resultado un archivo Word de la SDS que necesita
Escenario Principal	<ol style="list-style-type: none"> 1.- Usuario selecciona la opción Export SDS 2.- El sistema despliega una ventana para ingresar la ubicación y nombre del archivo, sugiriendo el número de la SDS 3.- Usuario ingresa nombre del archivo 4.- El sistema abre el archivo Word solicitado 5.- Usuario cierra Word y vuelve a interfaz anterior
Extensiones	<p>3 a- Usuario no desea exportar la SDS</p> <ul style="list-style-type: none"> - Usuario solicita salir de esta interfaz - Sistema vuelve a interfaz anterior

Tabla 9: Exportar SDS

6.3.2.5.2 Diagrama de Secuencia

A continuación se muestra el Diagrama de Secuencia 4.

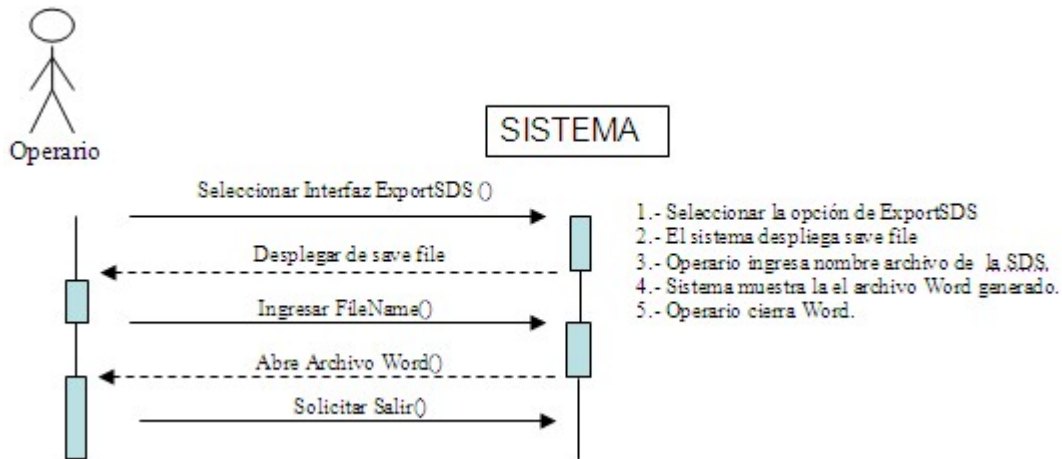
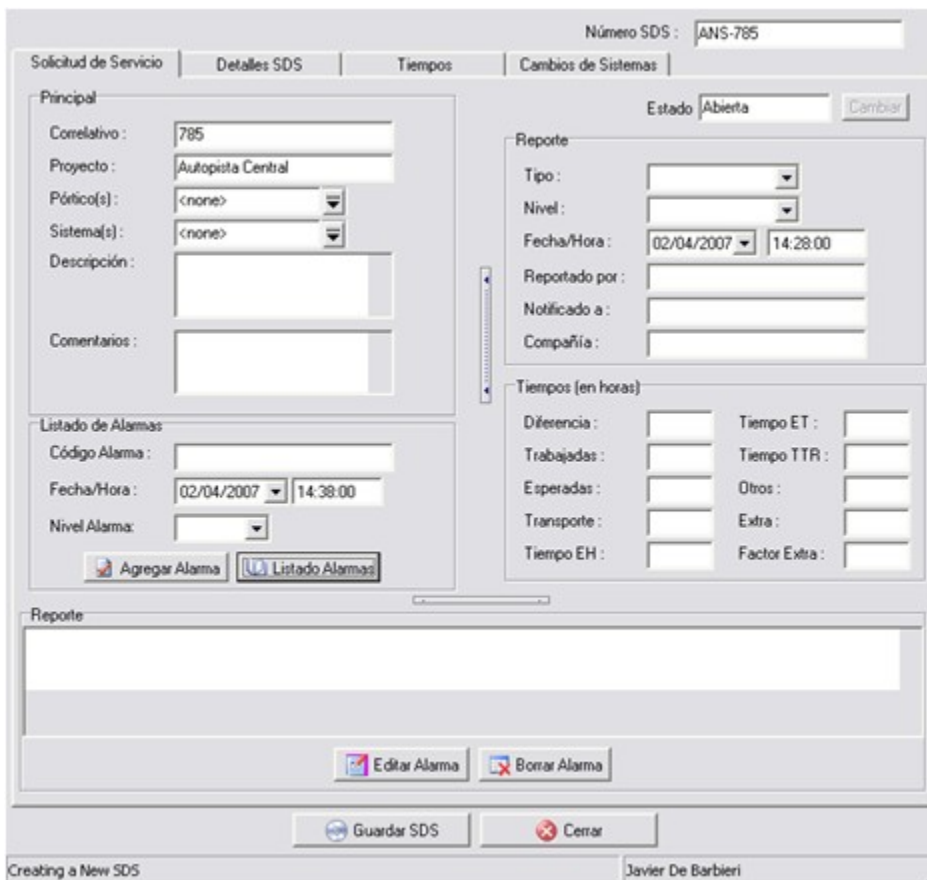


Diagrama de Secuencia 4: Exportar SDS

6.3.2.5.3 Interfaces relacionadas

Las siguientes ventanas representa la interfaz relacionada con el Caso de Uso.



Ventana 8: Menú Principal



SOLICITUD DE MANTENIMIENTO

Reporte

Correlativo	697	Fecha y hora de Reporte	26/03/2007 10:00
Lugar de la Ejecución	AC-05n1	Tipo Reporte	Correctivo
Equipo o Sistema Afectado	VDC	Nivel de Severidad	Nivel 1
Descripción de Incidencia	Aumento de fallas en la detección de vehículos...	Fecha y hora de Incidencia	26/03/2007 10:00
Estado de SoS	Abierta		

Reporto la Incidencia

Nombre	Autopista Central	Firma	WP^JBIwYUGUVSBTb~.*	-o
Comunicado a	Rodrigo Gutierrez	Empresa	Kapsch	

Acciones de Mantenimiento

Aviso Comienzo Trabajos	
Chequeo Seguridad	
Fecha y hora de Inicio	27/03/2007 10:55
Actividades:	Verificación de alarmas, monitoreo de ,contadores.,Extracción de archivos de ,controladores.,Reinicio de controladores del ,segmento afectado.,Verificación de ANS confirmando ,estadísticamente la normalidad del ,sistema en el segmento afectado.
Impacto en el Sistema	Segun información entregada por ,ANS la detección de TAG 's en este ,sitio ha bajado en un 10% ,aproximadamente. El sistema sigue ,operativo, y no acusa problema ,alguno.
Documentos de Apoyo	
Pruebas Realizadas	
Resultado de las Pruebas	
Materiales o/cant.	
Equipos	
Instrumentos de Medición:	
Línea base antigua	
Línea base nueva	
Condiciones de Operación	
Observaciones	El cambio del controlador VDC no ,tuvo incidencia en la normalización ,del sistema. Luego del reinicio del ,ALC, el TRX3 no es detectado por el ,controlador. Posterior al reinicio del ,ALC

Documento Word

6.3.2.6 Agregar Alarma

Este Caso de Uso corresponde a mostrar el listado de alarmas que existen en el sistema para que puedan ser agregadas a la SDS lo más fácilmente posible.

6.3.2.6.1 Especificación de Caso de Uso

En la tabla 10 se muestra la especificación de Caso de Uso.

Caso de Uso Agregar Alarma	
Actor Primario	Operario
Participante e intereses	Operario: Desea ingresar una alarma a una SDS
Precondiciones	El Operario es identificado en el sistema y debe tener una SDS en modo de modificación
Poscondiciones	El Operario actualiza la información como SDS
Escenario Principal	<ol style="list-style-type: none"> 1.- Usuario selecciona la opción agregar alarma desde ALARM LIST 2.- El sistema despliega interfaz ALARM LIST 3.- El sistema despliega listado de todas las alarmas 4.- Usuario busca la alarma que desea agregar ayudado por un TEXTBOX y un COMBOBOX que aceleran la búsqueda 5.- El Usuario selecciona la alarma 6.- Usuario acepta 7.- El sistema pega la alarma seleccionada en la SDS 8.- Sistema vuelve a interfaz anterior

<p>Extensiones</p>	<p>4 a- Usuario no encuentra la alarma que solicita</p> <ul style="list-style-type: none"> - Usuario solicita salir de esta interfaz - Sistema vuelve a interfaz anterior <p>8 a- Usuario no desea copiar ninguna alarma en la SDS</p> <ul style="list-style-type: none"> - Usuario solicita salir de esta interfaz - Sistema vuelve a interfaz anterior
---------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 10: Agregar Alarma

6.3.2.6.2 Diagrama de Secuencia

A continuación se muestra el Diagrama de Secuencia 5.

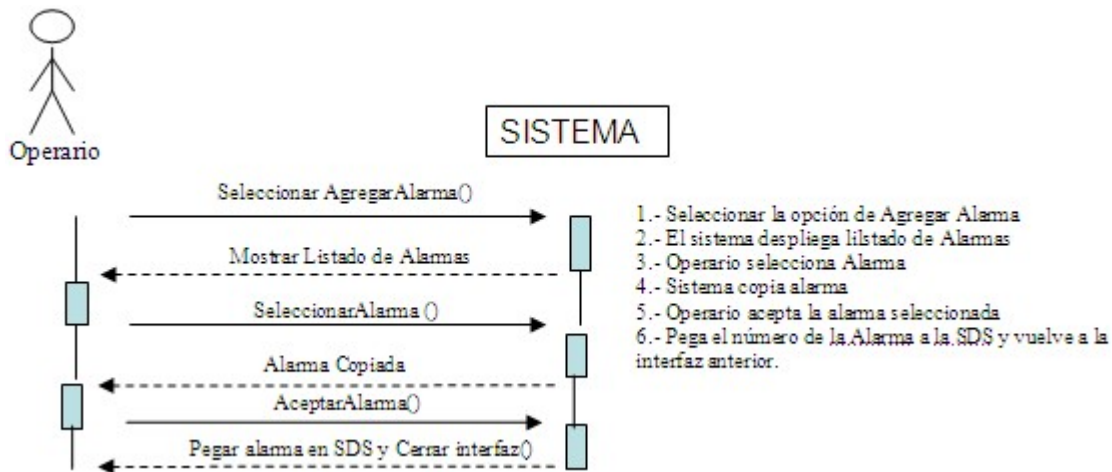
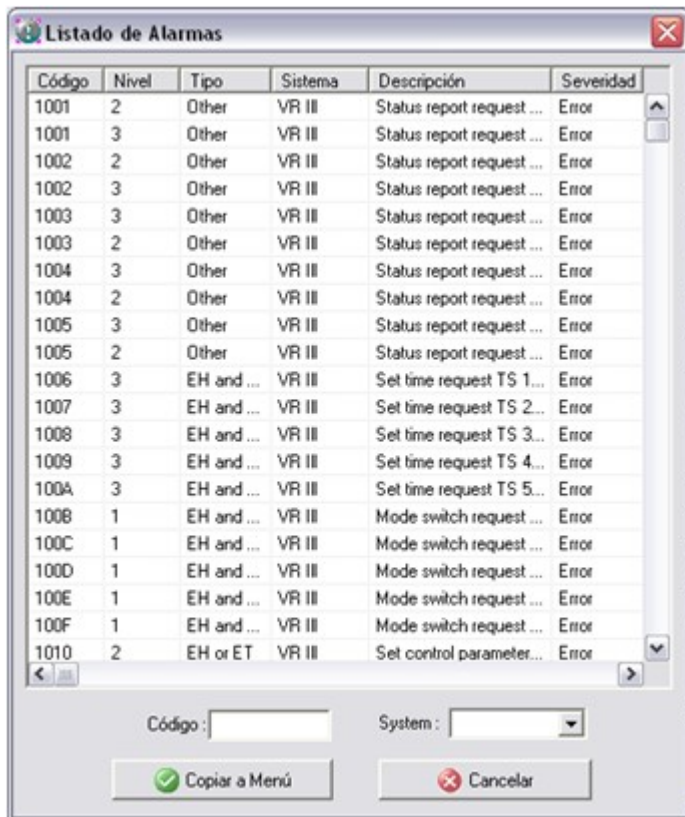


Diagrama de Secuencia 5: Agregar Alarma

6.3.2.6.3 Interfaces relacionadas

La Ventana 9 representa la interfaz relacionada con el Caso de Uso.

Ventana 9: Listado de Alarmas



6.3.2.7 Buscar SDS.

Este Caso de Uso muestra el listado de SDS que se poseen almacenados en la base de datos.

6.3.2.7.1 Especificación de Caso de Uso

En la tabla 11 se muestra la especificación de Caso de Uso.

Caso de Uso		Buscar SDS
Actor Primario		Operario
Participante	e	Operario: Desea encontrar una determinada SDS

intereses	
Precondiciones	El Operario se identifica en el sistema
Poscondiciones	El Operario encuentra la SDS que esta solicitando
Escenario Principal	<ol style="list-style-type: none"> 1.- Usuario selecciona la opción Search SDS 2.- El sistema despliega interfaz Search SDS 3.- Sistema despliega lista de SDS 4.- Usuario selecciona desde el listado de SDS la que desea ver 5.- El sistema muestra la información correspondiente a la SDS 6.- Usuario visualiza la información de la SDS 7.- Usuario cierra ventana de SDS 8.- Sistema vuelve a interfaz anterior
Extensiones	<p>4 a- Usuario no encuentra entre las SDS la que el que busca</p> <ul style="list-style-type: none"> - Usuario solicita salir de esta interfaz - Sistema vuelve a interfaz anterior <p>6 a- Usuario solicita imprimir</p> <ul style="list-style-type: none"> - Sistema genera impresión del local

	- Sistema vuelve a interfaz anterior
--	---------------------------------------------

Tabla 11: Buscar SDS

6.3.2.7.2 Diagrama de Secuencia

A continuación se muestra el Diagrama de Secuencia 6.

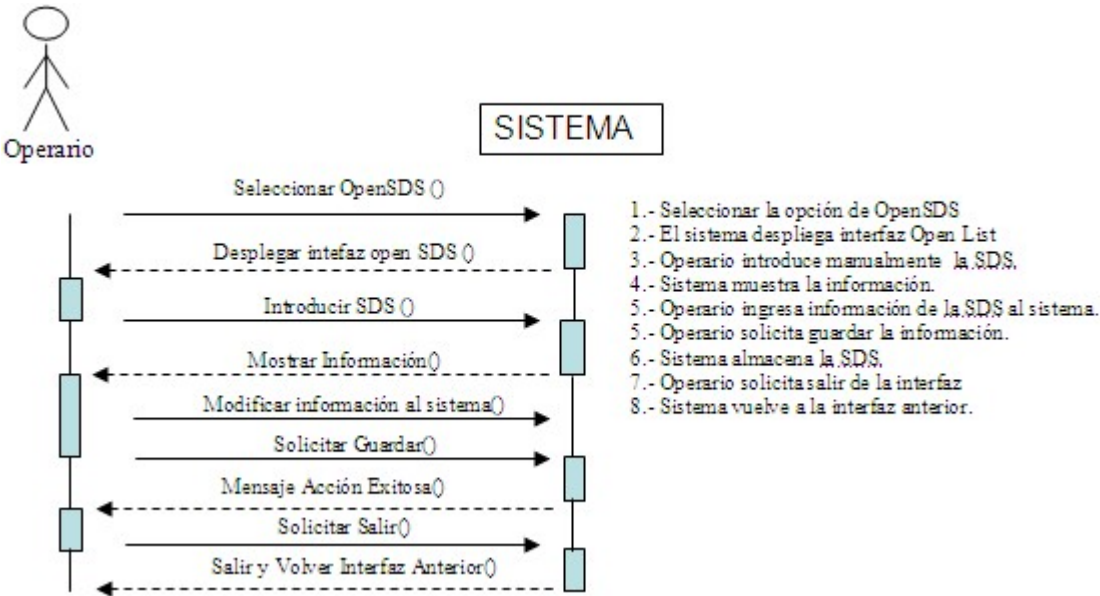


Diagrama de Secuencia 6: Abrir SDS

6.3.2.7.3 Interfaces relacionadas

La Ventana 10 representa la interfaz relacionada con el Caso de Uso.

Ventana 10: Búsqueda SDS

Busqueda SDS

Proyecto :

Fecha desde : 21/03/2007 a 19/06/2007

Nivel : Estado :

SDS N°	Proyecto	Nivel	Estado	Fecha Hora
ANS-691	Autopista Central	1	Cerrada	22/03/2007 1:
ANS-693	Autopista Central	1	Abierta	23/03/2007 1:
ANS-697	Autopista Central	1	Abierta	26/03/2007 10:
ANS-708	Autopista Central	3	Cerrada	02/04/2007 1:
ANS-709	Autopista Central	2	Cerrada	03/04/2007 6:

Número de SDS :

6.3.2.8 Cambiar Estado SDS

Este Caso de Uso se refiere al cambio manual de estado de una SDS.

6.3.2.8.1 Especificación de Caso de Uso

En la tabla 12 se muestra la especificación de Caso de Uso.

Caso de Uso	Cambiar estado SDS
Actor Primario	Operario
Participante e intereses	Operario: Desea cambiar el estado de una SDS
Precondiciones	El Operario se identifica en el sistema y tiene una SDS en modo de modificación.
Poscondiciones	El Operario cambia el estado de una SDS
Escenario	1.- Usuario selecciona la opción Cambiar estado

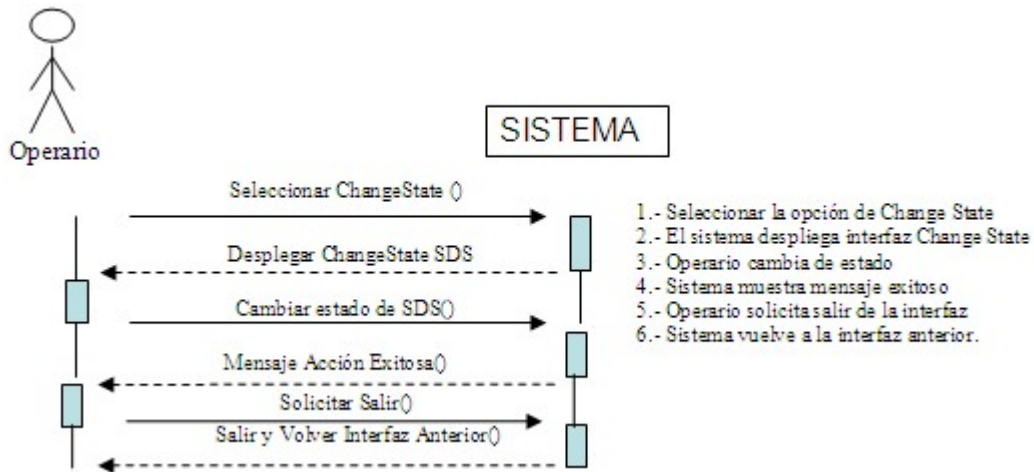
Principal	<p>2.- El sistema despliega interfaz Cambiar Estado</p> <p>3.- Usuario cambia al estado que necesita</p> <p>4.- Usuario acepta</p> <p>5.- Sistema vuelve a interfaz anterior</p>
Extensiones	<p>4 a- Usuario no acepta el cambio</p> <ul style="list-style-type: none"> - Sistema mantiene el estado de la SDS - Usuario solicita salir de esta interfaz - Sistema vuelve a interfaz anterior

Tabla 12: Cambiar Estado

6.3.2.8.2 Diagrama de Secuencia

A continuación se muestra el Diagrama de Secuencia 7.

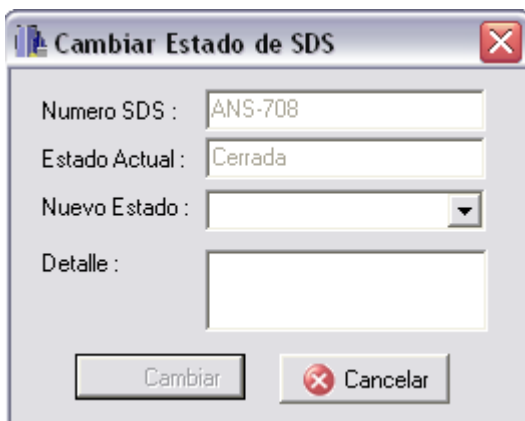
Diagrama de Secuencia 7: Cambiar Estado



6.3.2.8.3 Interfaces relacionadas

La Ventana 11 representa la interfaz relacionada con el Caso de Uso.

Ventana 11: Cambiar estado SDS



6.3.2.9 Eliminar SDS

Este Caso de Uso referencia a el cambio de estado de una SDS a modo Eliminado.

6.3.2.9.1 Especificación de Caso de Uso

En la tabla 13 se muestra la especificación de Caso de Uso.



Caso de Uso Eliminar SDS	
Actor Primario	Operario
Participante e intereses	Operario: Desea mover al listado de SDS eliminadas
Precondiciones	El Operario se identificado en el sistema y debe saber en número de la SDS a eliminar
Poscondiciones	El Operario envía esta SDS al listado de Eliminadas
Escenario Principal	<ol style="list-style-type: none"> 1.- Usuario selecciona la opción Eliminar SDS 2.- El sistema despliega interfaz Eliminar SDS 3.- Sistema solicita ingresar SDS a eliminar 4.- Usuario ingresa manualmente la SDS a eliminar 5.- El sistema solicita la confirmación de la eliminación 6.- Usuario acepta 7.- Sistema informa que se ha copiado al listado de SDS borradas 8.- Sistema vuelve a la interfaz anterior
Extensiones	<p>4 a- Usuario no desea eliminar ninguna SDS</p> <p>- Usuario solicita salir de esta interfaz</p>

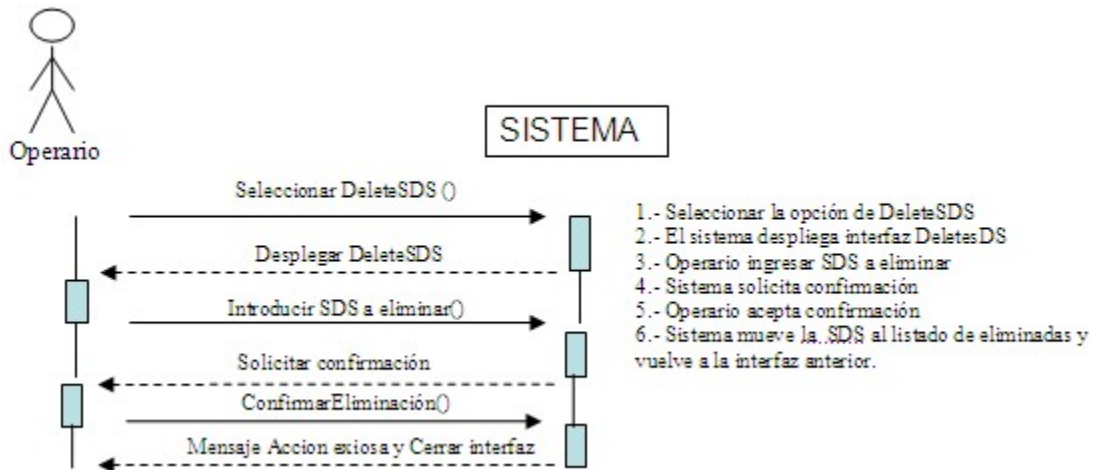
	<p>- Sistema vuelve a interfaz anterior</p> <p>6 a- Usuario no acepta la eliminación de la SDS</p> <p>- Sistema vuelve a interfaz anterior</p>
--	-------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 13: Eliminar SDS

6.3.2.9.2 Diagrama de Secuencia

A continuación se muestra el Diagrama de Secuencia 8.

Diagrama de Secuencia 8: Borrar SDS



6.3.2.9.3 Interfaces relacionadas

La Ventana 12 representa la interfaz relacionada con el Caso de Uso.

Ventana 12: Búsqueda SDS



6.3.2.10 Listado SDS Eliminadas

Este Caso de Uso referencia a el listado de SDS que han sido eliminadas del sistema para su eliminación permanente o rescate desde el listado.

6.3.2.10.1 Especificación de Caso de Uso

En la tabla 14 se muestra la especificación de Caso de Uso.

Caso de Uso	Listado SDS eliminadas
Actor Primario	Operario
Participante e intereses	Operario: Desea eliminar definitivamente o rescatar una SDS que ha sido eliminada
Precondiciones	El Operario se identificado en el sistema
Poscondiciones	El Operario actualiza el listado de SDS eliminados
Escenario Principal	1.- Usuario selecciona la opción Listado de SDS Eliminadas 2.- El sistema despliega interfaz Listado de SDS Eliminadas 3.- EL Sistema despliega todas las SDS que han sido eliminadas 4.- Usuario selecciona la SDS a eliminar o rescatar 5.- El sistema solicita la confirmación de la eliminación o rescate 6.- Usuario acepta 7.- Sistema informa que se ha eliminado o rescatado la SDS del

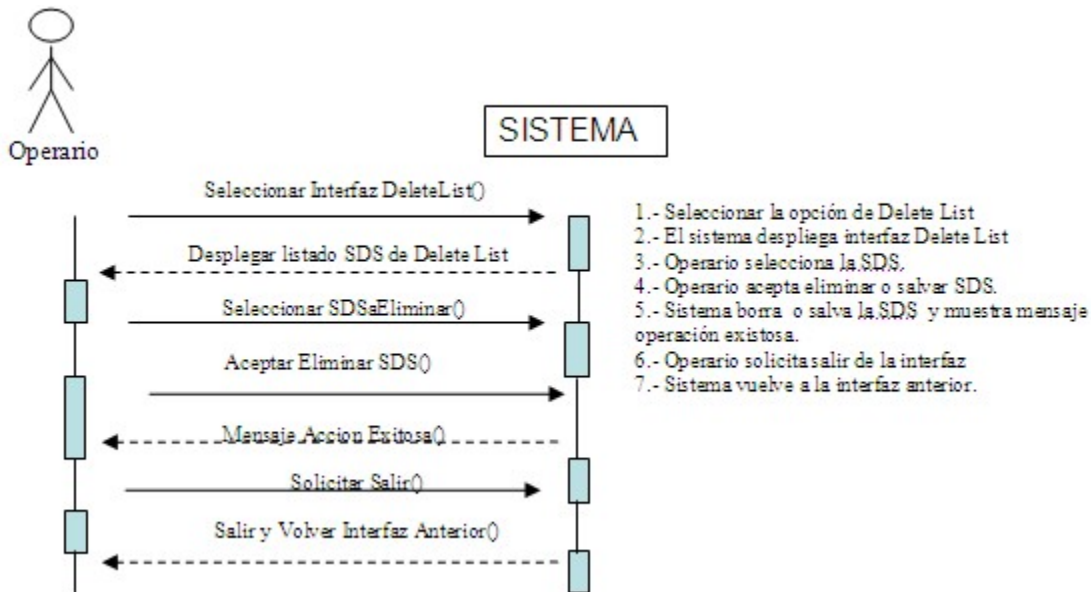
	listado 8.- Sistema vuelve a la interfaz anterior
Extensiones	4 a- Usuario no desea eliminar o rescatar ninguna SDS - Usuario solicita salir de esta interfaz - Sistema vuelve a interfaz anterior 6 a- Usuario no acepta la eliminación o el rescate de ninguna SDS - Sistema vuelve a interfaz anterior

Tabla 14: Listado SDS eliminadas

6.3.2.10.2 Diagrama de Secuencia

A continuación se muestra el Diagrama de Secuencia 9.

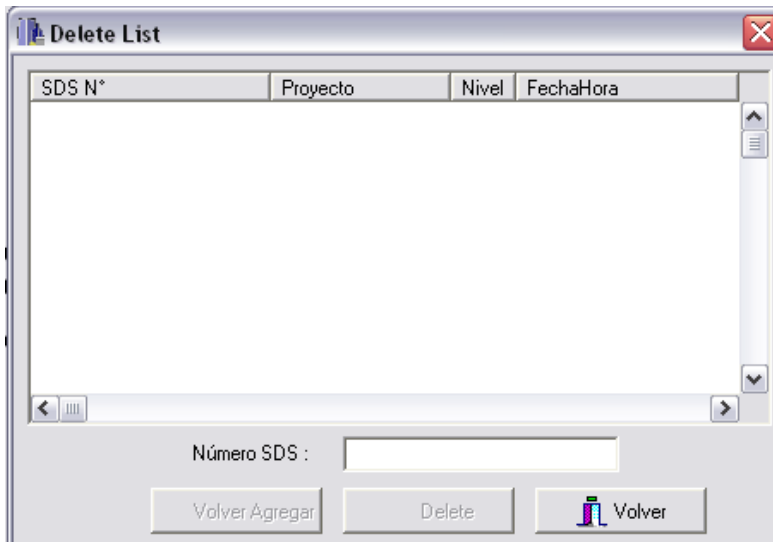
Diagrama de Secuencia 9: Borrar desde Listado



6.3.2.10.3 Interfaces relacionadas

Las Ventana 13 representa la interfaz relacionada con el Caso de Uso.

Ventana 13: Borrar de Listado

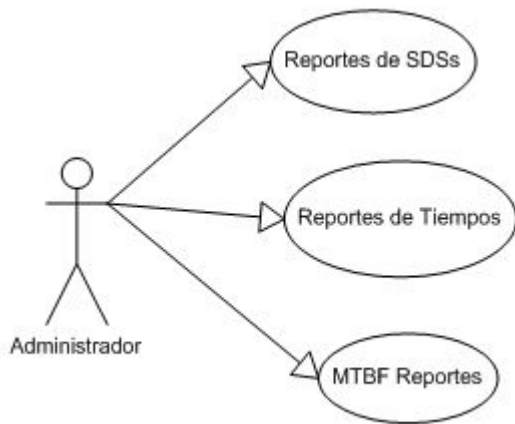


6.3.3 Administrador de Reportes

En esta sección del informe se definirán los casos de uso referentes a la entrega de reportes que posee el sistema.

En la figura 13 se muestra el diagrama de Caso de Uso “Administración de Reportes”.

Figura 13: Diagrama Caso de Uso “Administración de Reportes”



6.3.3.1 Reportes de SDS

Este Caso de Uso representa los reportes de SDS que el sistema debe entregar al Gerente de Operaciones. Parecido al anexo

6.3.3.1.1 Especificación de Caso de Uso

En la tabla 15 se muestra la especificación de Caso de Uso.

Caso de Uso		Reportes de SDS
Actor Primario		Administrador
Participante e intereses	e	Administrador: Desea generar un reporte
Precondiciones		El Administrador se identifica en el sistema.
Poscondiciones		El Administrador obtiene los resultados de reportes sobre las SDS

<p>Escenario Principal</p>	<ol style="list-style-type: none"> 1.- Administrador selecciona la opción SDS Reports 2.- El sistema despliega interfaz SDS Reports 3.- Usuario selecciona desde el SDS Report el año y/o mes de las SDS, y filtros de nivel, proyecto, etc. 4.- El sistema despliega el listado de las SDS solicitadas 5.- Administrador visualiza la información de las SDS 6.- Administrador cierra ventana de SDS Reports 7.- Sistema vuelve a interfaz anterior
<p>Extensiones</p>	<ol style="list-style-type: none"> 5 a- Administrador solicita imprimir <ul style="list-style-type: none"> - Sistema genera impresión - Sistema vuelve a interfaz anterior 5 b- Administrador solicita exportar a Excel <ul style="list-style-type: none"> - Sistema solicita nombre archivo y ubicación - Sistema vuelve a interfaz anterior 5 c- Administrador solicita ver grafico <ul style="list-style-type: none"> - Sistema despliega gráfico - Sistema vuelve a interfaz anterior

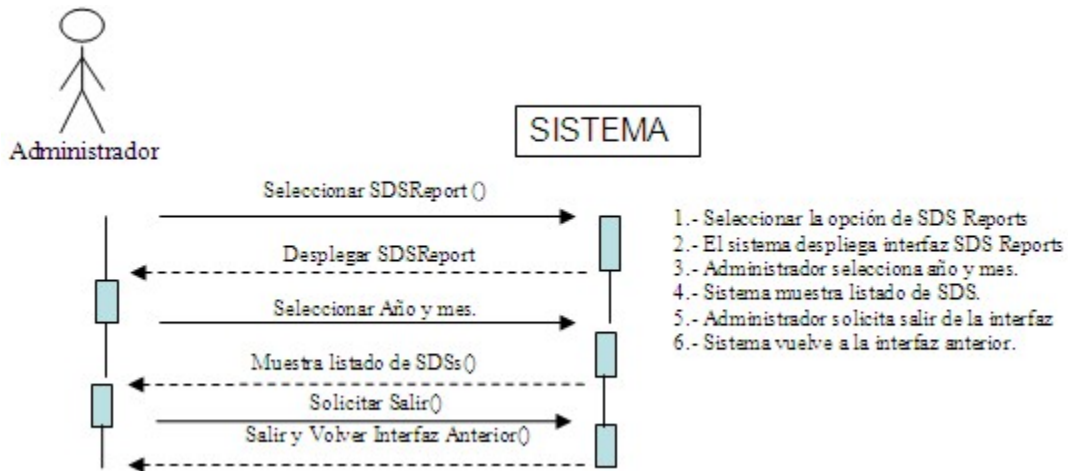
	<p>5 d- Administrador solicita ver SDS</p> <ul style="list-style-type: none"> - Sistema despliega SDS - Sistema vuelve a interfaz anterior
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 15: Reportes SDS

6.3.3.1.2 Diagrama de Secuencia

A continuación se muestra el Diagrama de Secuencia 10.

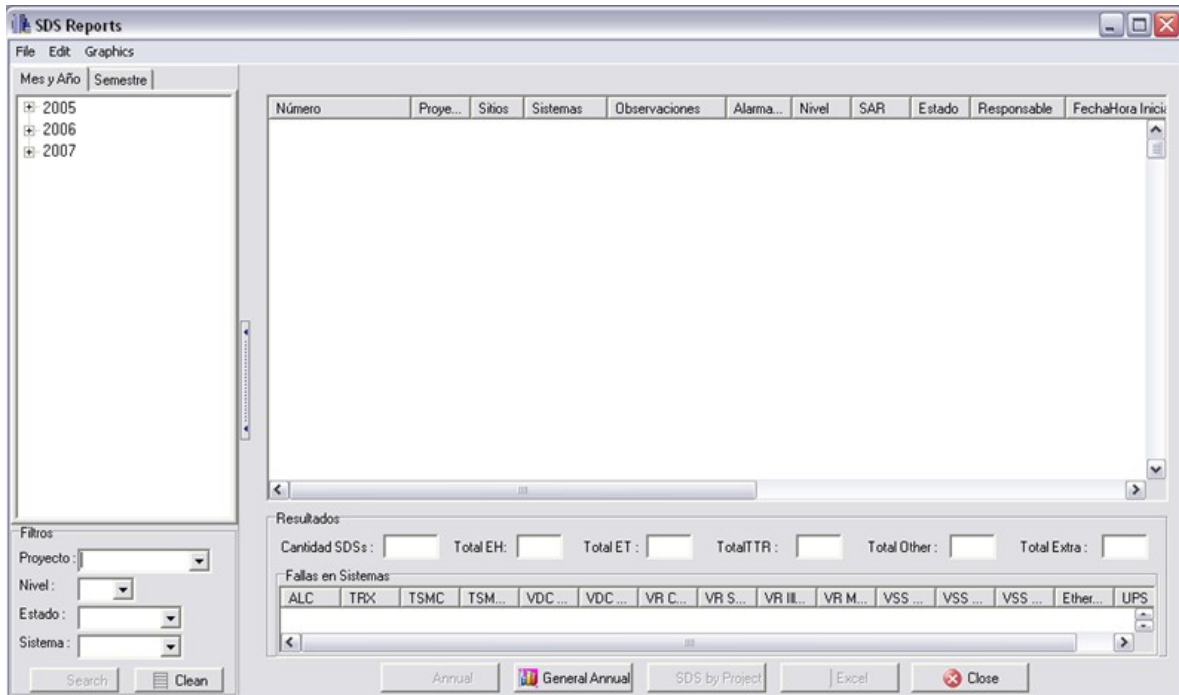
Diagrama de Secuencia 10: SDS Report



6.3.3.1.3 Interfaces relacionadas

Las Ventana 14 representa la interfaz relacionada con el Caso de Uso.

Ventana 14: Reporte de SDS



6.3.3.2 Reportes de Tiempos

6.3.3.2.1 Especificación de Caso de Uso

En la tabla 16 se muestra la especificación de Caso de Uso.

Caso de Uso	Reportes de Tiempos
Actor Primario	Administrador
Participante e intereses	Administrador: Desea generar un reporte de tiempos
Precondiciones	El Administrador se identifica en el sistema
Poscondiciones	El Administrador obtiene los resultados de reportes sobre los tiempos EH, ET, TTR y OTHERs mensuales

<p>Escenario Principal</p>	<ol style="list-style-type: none"> 1.- Administrador selecciona la opción Reporte de Tiempos 2.- El sistema despliega interfaz Reporte de Tiempos 3.- Usuario selecciona desde el Reporte de Tiempos el año de los tiempos y el proyecto a visualizar 4.- El sistema despliega el año con sus respectivos tiempos EH, ET, OTHERS, EXTRA 5.- Administrador visualiza la información de los tiempos 6.- Administrador cierra ventana 7.- Sistema vuelve a interfaz anterior
<p>Extensiones</p>	<ol style="list-style-type: none"> 5 a- Administrador solicita imprimir <ul style="list-style-type: none"> - Sistema genera impresión - Sistema vuelve a interfaz anterior 5 b- Administrador solicita exportar a Excel <ul style="list-style-type: none"> - Sistema solicita nombre archivo y ubicación - Sistema vuelve a interfaz anterior 5 c- Administrador solicita ver grafico <ul style="list-style-type: none"> - Sistema despliega gráfico - Sistema vuelve a interfaz anterior

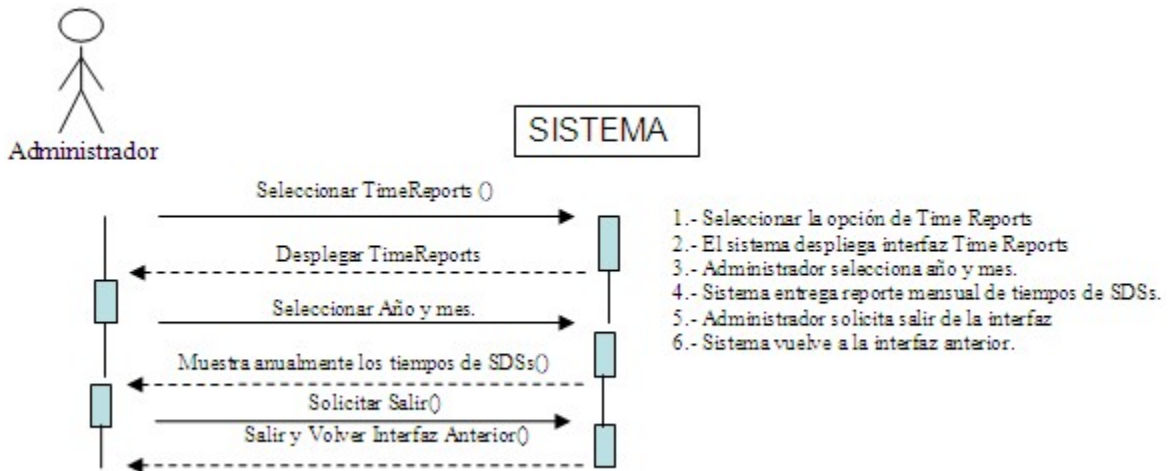
	<p>5 d- Administrador solicita ver SDS</p> <ul style="list-style-type: none"> - Sistema despliega SDS. - Sistema vuelve a interfaz anterior.
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 16: Reporte de SDS

6.3.3.2.2 Diagrama de Secuencia

A continuación se muestra el Diagrama de Secuencia 12.

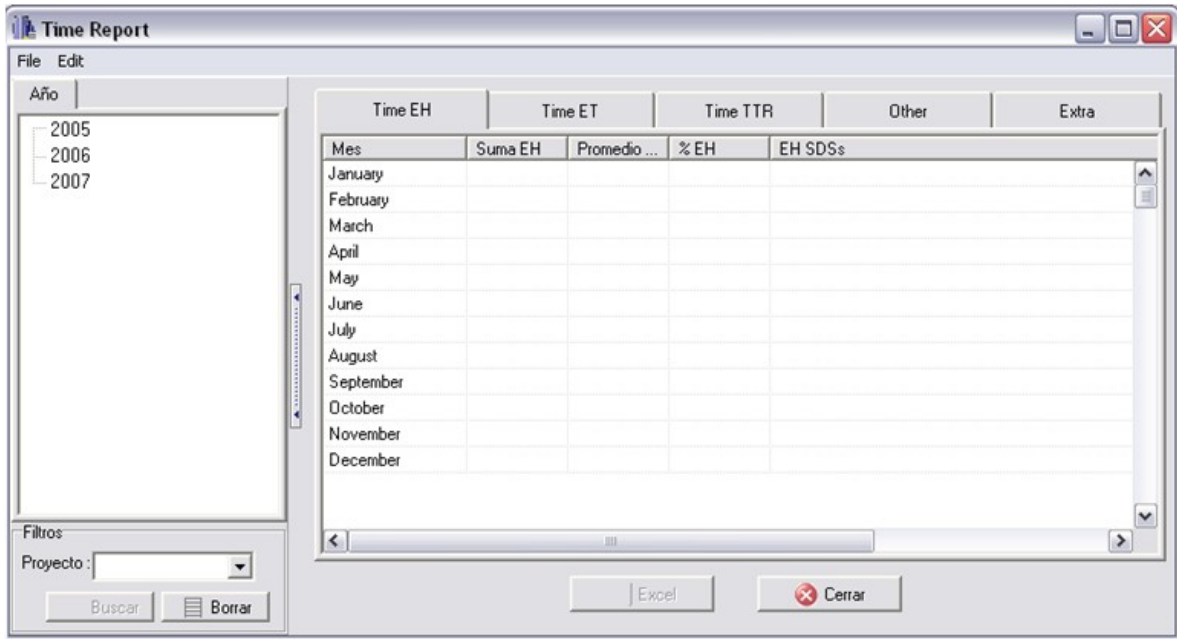
Diagrama de Secuencia 12: Time Reports



6.3.3.2.3 Interfaces relacionadas

La Ventana 15 representa la interfaz relacionada con el Caso de Uso.

Ventana 15: Reportes de Tiempo



6.3.3.3 Reportes de MTBF

6.3.3.3.1 Especificación de Caso de Uso

En la tabla 17 se muestra la especificación de Caso de Uso.

Caso de Uso	Reportes de MTBF
Actor Primario	Administrador
Participante e intereses	Administrador: Desea generar un reporte de MTBF
Precondiciones	El Administrador se identifica en el sistema
Poscondiciones	El Administrador obtiene los resultados de reportes sobre los MTBF mensuales

<p>Escenario Principal</p>	<p>1.- Administrador selecciona la opción Reporte de MTBF</p> <p>2.- El sistema despliega interfaz Reporte de MTBF</p> <p>3.- Usuario selecciona desde el Reporte de MTBF el año y el mes inicial de los tiempos y el proyecto a visualizar</p> <p>4.- El sistema despliega el año con sus respectivos MTBF</p> <p>5.- Administrador visualiza la información de las MTBF</p> <p>6.- Administrador cierra ventana</p> <p>7.- Sistema vuelve a interfaz anterior</p>
<p>Extensiones</p>	<p>5 a- Administrador solicita imprimir</p> <ul style="list-style-type: none"> - Sistema genera impresión del local - Sistema vuelve a interfaz anterior <p>5 b- Administrador solicita exportar a Excel</p> <ul style="list-style-type: none"> - Sistema solicita nombre archivo y ubicación - Sistema vuelve a interfaz anterior <p>5 c- Administrador solicita ver grafico</p> <ul style="list-style-type: none"> - Sistema despliega gráfico - Sistema vuelve a interfaz anterior <p>5 d- Administrador solicita ver SDS</p>

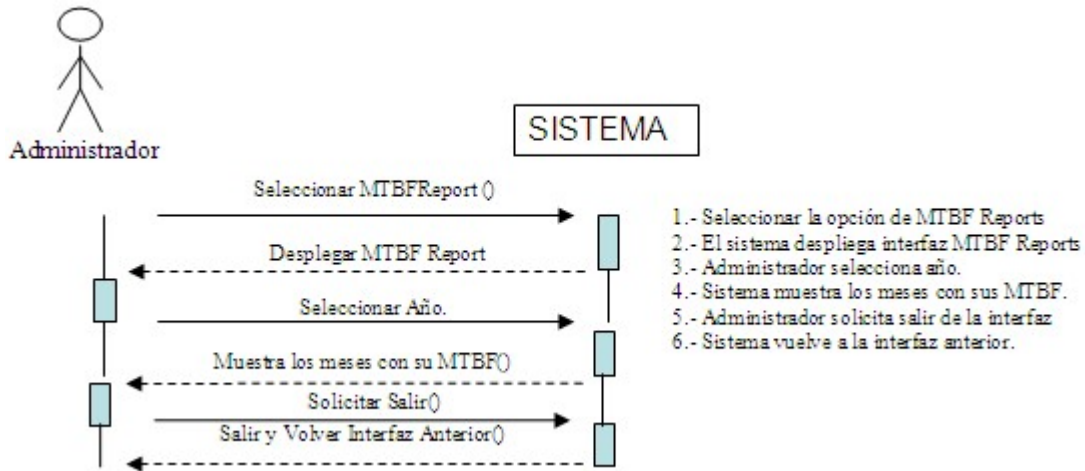
	<p>- Sistema despliega SDS</p> <p>- Sistema vuelve a interfaz anterior</p>
--	------------------------------------------------------------------------------------------

Tabla 17: Reportes MTBF

6.3.3.3.2 Diagrama de Secuencia

A continuación se muestra el Diagrama de Secuencia 13.

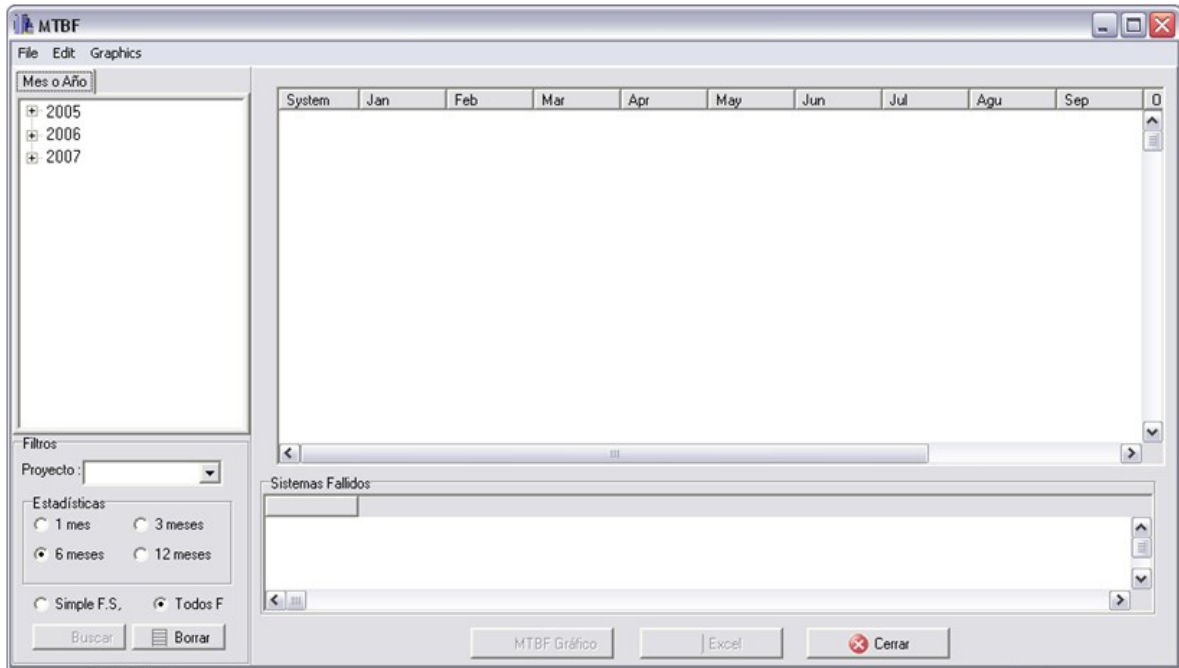
Diagrama de Secuencia 13: Reporte MTBF



6.3.3.3.3 Interfaces relacionadas

La Ventana 16 representa la interfaz relacionada con el Caso de Uso.

Ventana 16: Reportes MTBF

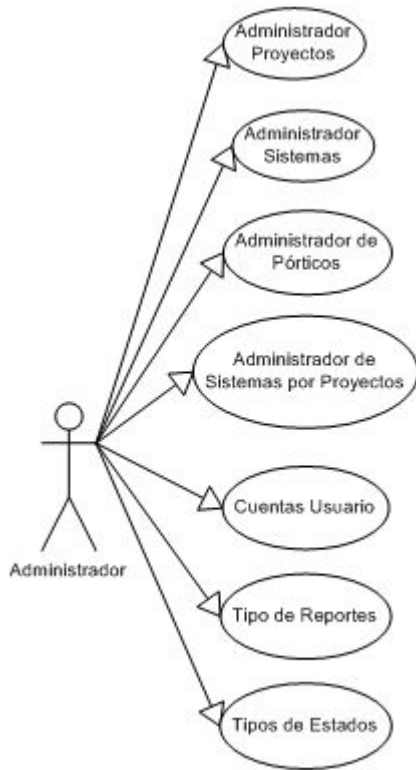


6.3.4 Administración de Datos y Usuarios

En esta sección del informe se definirán los casos de uso referentes a los administradores de datos que existen en el sistema.

En la figura 14 se muestra el diagrama correspondiente al Caso de Uso “Administración de Datos y Usuarios”

Figura 14: Caso de Uso “Administración de Datos y Usuarios”



6.3.4.1 Administrador de Proyectos

6.3.4.1.1 Especificación de Caso de Uso

En la siguiente tabla se muestra la especificación de Caso de Uso.

Caso de Uso		Administración de Proyectos
Actor Primario		Administrador
Participante e intereses	e	Administrador: Desea Administrar los Proyectos del Sistema
Precondiciones		El Administrador se identifica en el sistema
Poscondiciones		El Administrador realiza cambios en los proyectos de sistema

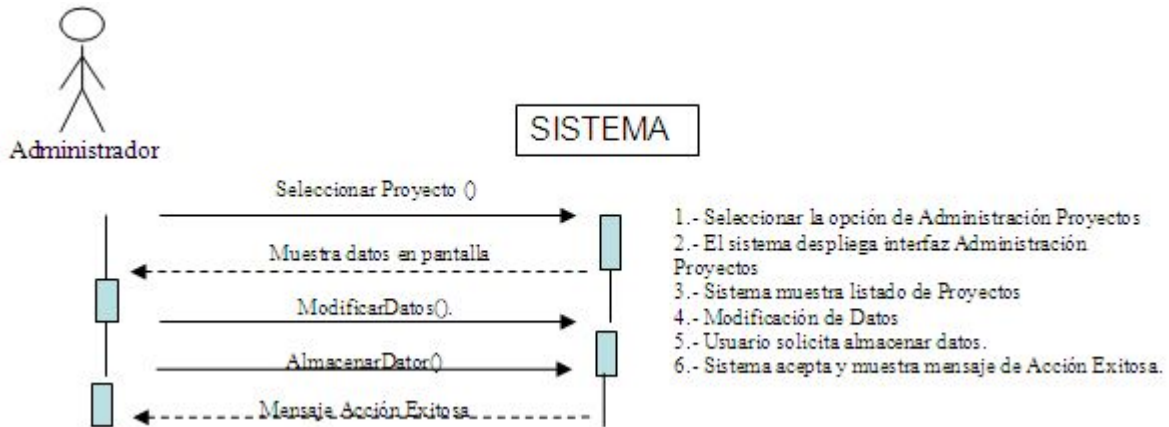
<p>Escenario Principal</p>	<p>1.- Usuario selecciona la opción Administración de Proyectos</p> <p>2.- El sistema despliega interfaz Administración de Proyectos</p> <p>3.- Usuario selecciona un proyecto a modificar</p> <p>4.- Usuario ingresa datos</p> <p>5.- Usuario selecciona Guardar cambios</p> <p>6.- Sistema despliega mensaje acción exitosa</p>
<p>Extensiones</p>	<p>3 a- Usuario solicita ingresar nuevo proyecto</p> <ul style="list-style-type: none"> - Sistema muestra los Textbox - Vuelve a 4 <p>3 a Usuario solicita eliminar proyecto</p> <ul style="list-style-type: none"> - Usuario selecciona proyecto a eliminar - Usuario acepta la eliminación - Vuelve a opción 5 <p>5 a Usuario no desea Guardar cambios</p> <ul style="list-style-type: none"> - Sistema deshace los cambios

Tabla 18: Administración de Proyectos

6.3.4.1.2 Diagrama de Secuencia

A continuación se muestra el Diagrama de Secuencia 13 de este Caso de Uso.

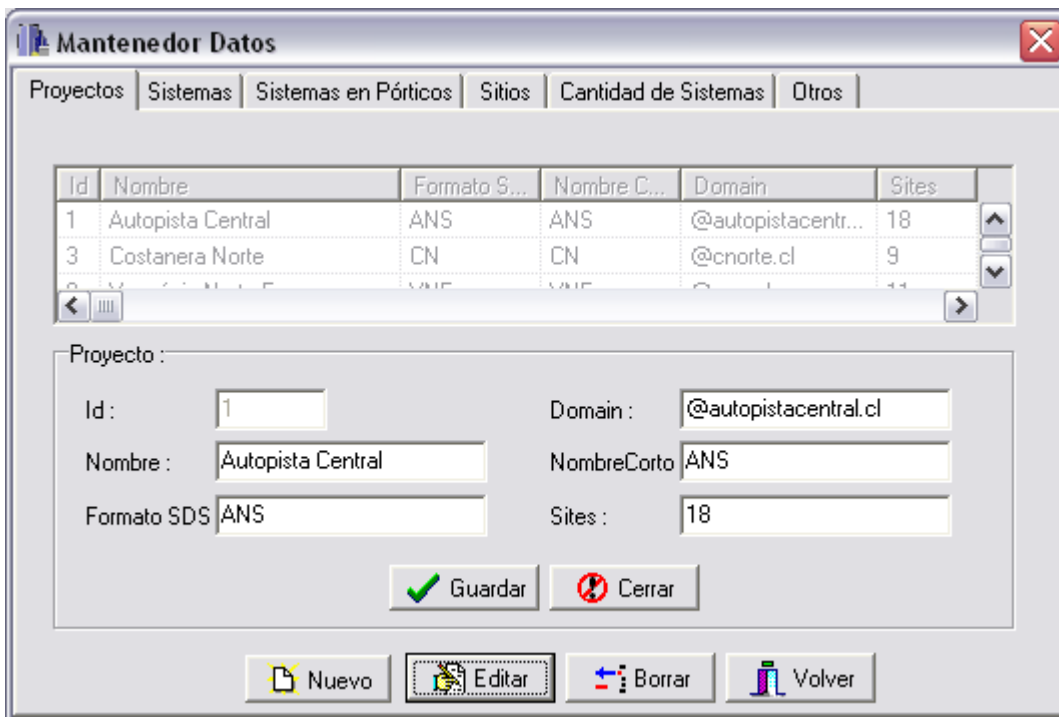
Diagrama de Secuencia 13: Administrador de proyectos



6.3.4.1.3 Interfaces relacionadas

La Ventana 17 representa la interfaz relacionada con el Caso de Uso.

Ventana 17: Mantenedor Proyectos



6.3.4.2 Administrador de Sistemas

6.3.4.2.1 Especificación de Caso de Uso

En la siguiente tabla se muestra la especificación de Caso de Uso.

Caso de Uso Administración de Sistemas	
Actor Primario	Administrador
Participante e intereses	Administrador: Desea Administrar los Sistemas que se poseen en la empresa
Precondiciones	El Administrador se identifica en el sistema
Poscondiciones	El Administrador realiza cambios en los Sistemas
Escenario Principal	1.- Usuario selecciona la opción Administración de Sistemas 2.- El sistema despliega interfaz Administración de Sistemas 3.- Usuario selecciona el sistema a modificar 4.- Usuario ingresa datos 5.- Usuario solicita almacenar 6.- Sistema despliega mensaje acción exitosa
Extensiones	3 a- Usuario solicita ingresar nuevo proyecto

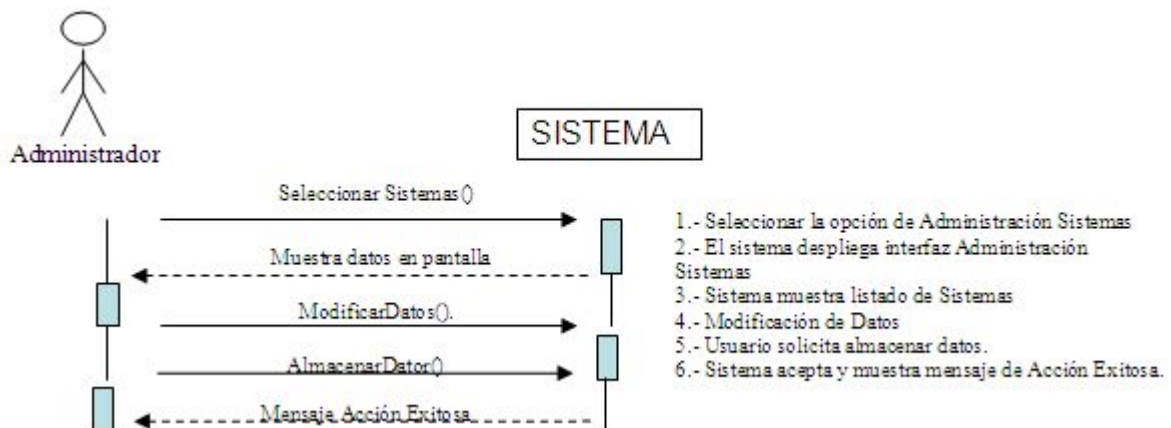
	<ul style="list-style-type: none"> - Sistema muestra los Textbox - Vuelve a 4 <p>3 a Usuario solicita sistema</p> <ul style="list-style-type: none"> - Usuario selecciona proyecto a eliminar - Usuario acepta la eliminación - Vuelve a opción 6 <p>5 a Usuario no desea Guardar cambios</p> <ul style="list-style-type: none"> - Sistema deshace los cambios
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 19: Administración de Sistemas

6.3.4.2.2 Diagrama de Secuencia

A continuación se muestra el Diagrama de Secuencia 14.

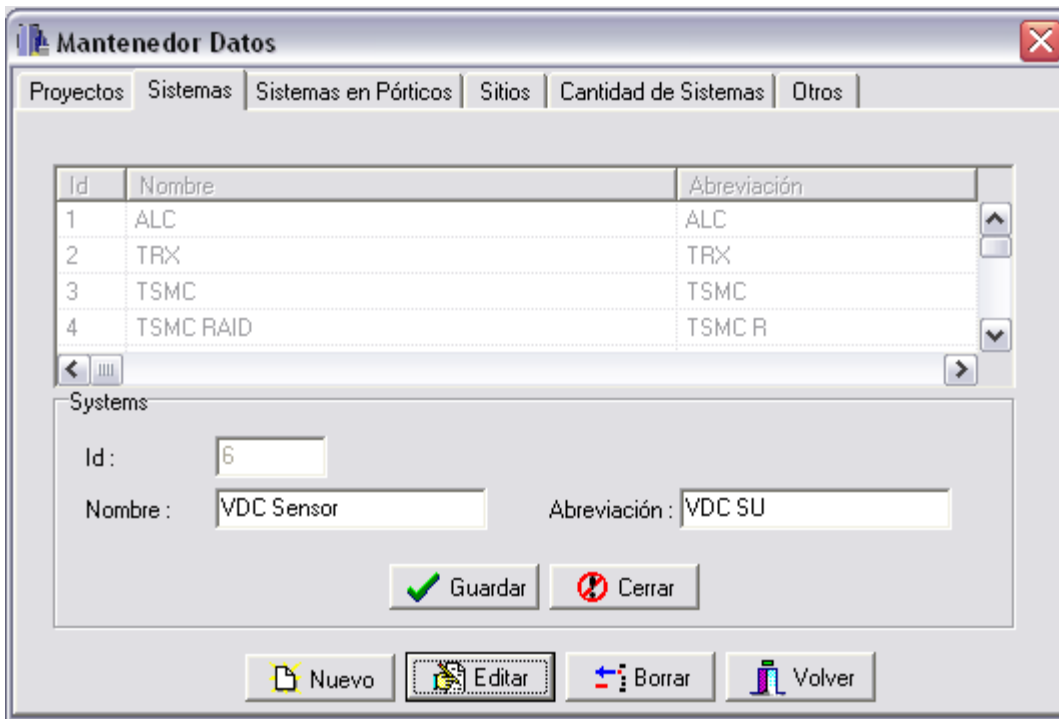
Diagrama de Secuencia 14: Administrador de Sistemas



6.3.4.2.3 Interfaces relacionadas

Las Ventana 18 representa la interfaz relacionada con el Caso de Uso.

Ventana 18: Mantenedor Sistemas



6.3.4.3 Administrador de Sistemas en Pórticos

6.3.4.3.1 Especificación de Caso de Uso

En la tabla 20 se muestra la especificación de Caso de Uso.

Caso de Uso	Administración de Sistemas en Pórticos
Actor Primario	Administrador
Participante e intereses	Administrador: Desea Administrar los Pórticos que están en el Sistema
Precondiciones	El Administrador se identifica en el sistema
Poscondiciones	El Administrador realiza visualiza en los Pórticos

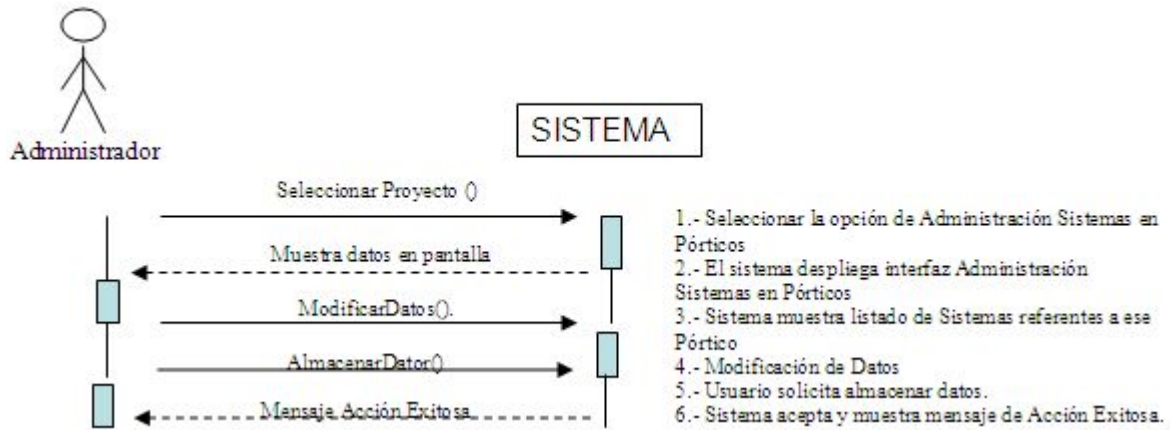
<p>Escenario Principal</p>	<p>1.- Usuario selecciona la opción Administración de Sistema en Pórticos</p> <p>2.- El sistema despliega interfaz Administración de Sistemas en Pórticos</p> <p>3.- Usuario selecciona el proyecto</p> <p>4.- Sistema despliega el listado de Sitios</p> <p>5.- Usuario selecciona el Pórtico a visualizar</p> <p>6.- Usuario solicita salir</p>
<p>Extensiones</p>	<p>5 a- Usuario no encuentra pórticos</p> <p>- Vuelve a 4</p>

Tabla 20: Administración Sistema Pórticos

6.3.4.3.2 Diagrama de Secuencia

A continuación se muestra el Diagrama de Secuencia 15.

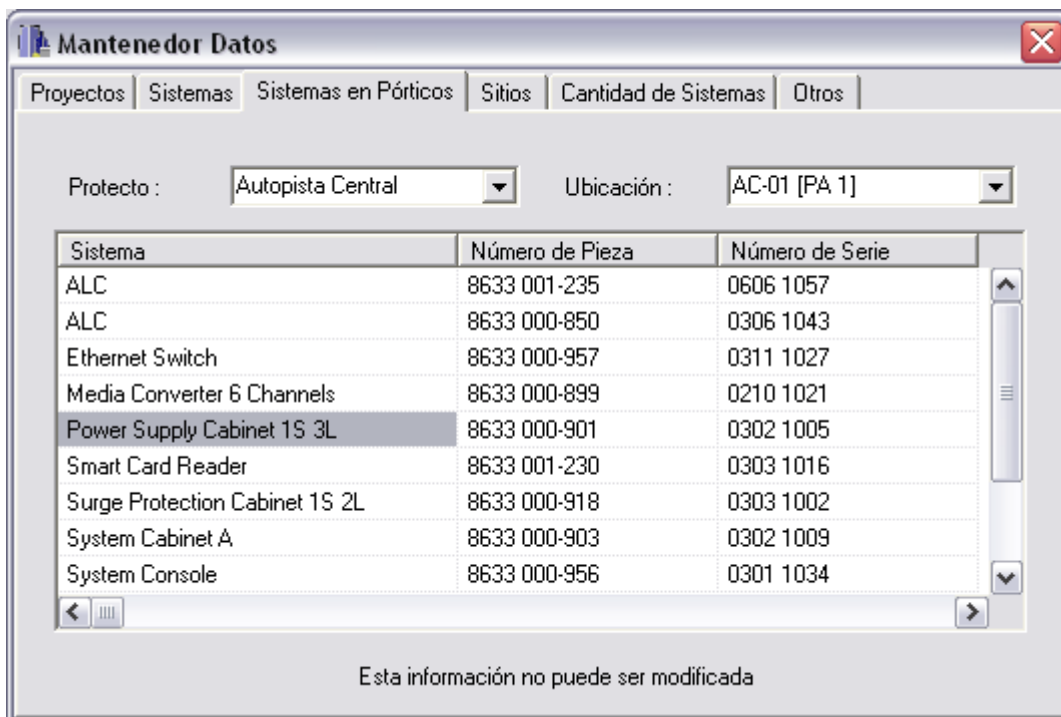
Diagrama de Secuencia 15: Administrador Sistema Pórticos



6.3.4.3.3 Interfaces relacionadas

La Ventana 19 representa la interfaz relacionada con el Caso de Uso.

Ventana 19: Administrador Sistemas Pórticos



6.3.4.4 Administrador de Sitios

6.3.4.4.1 Especificación de Caso de Uso

En la tabla 21 se muestra la especificación de Caso de Uso.

Caso de Uso Administración de Sitios	
Actor Primario	Administrador
Participante e intereses	Administrador: Desea Administrar los sitios que se poseen en la empresa
Precondiciones	El Administrador se identifica en el sistema
Poscondiciones	El Administrador realiza cambios en los sitios
Escenario Principal	<ol style="list-style-type: none"> 1.- Usuario selecciona la opción Administración de Sitios 2.- El sistema despliega interfaz Administración de Sitios 3.- Usuario selecciona el proyecto del sitio a modificar 4.- Sistema despliega el listado de sitios que posee el proyecto seleccionado 5.- Usuario selecciona el sitio que desea modificar 6.- Usuario ingresa datos 7.- Usuario solicita almacenar 8.- Sistema despliega mensaje acción exitosa
Extensiones	5 a- Usuario solicita ingresar nuevo sitio

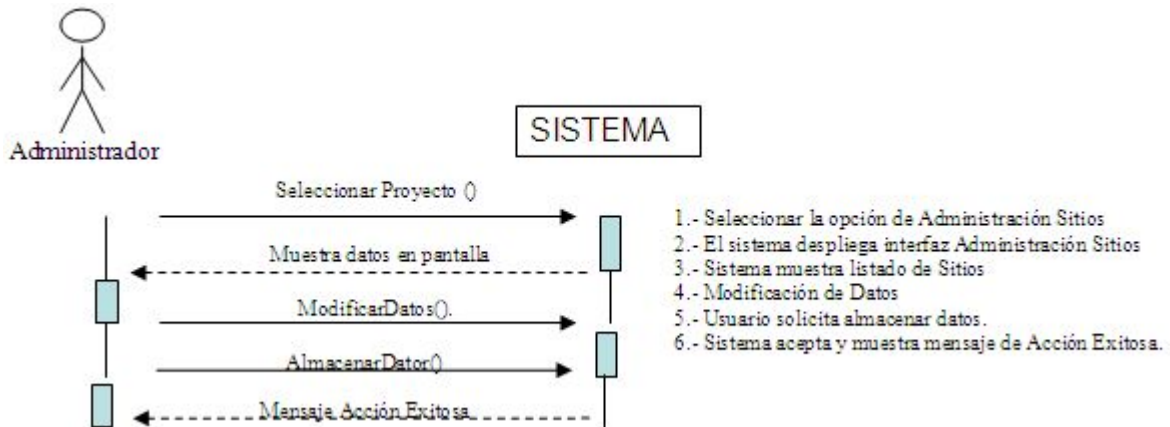
	<ul style="list-style-type: none"> - Sistema muestra los Textbox - Vuelve a 4 5 a Usuario solicita eliminar sitio - Usuario selecciona proyecto a eliminar - Usuario acepta la eliminación - Vuelve a opción 6 7 a Usuario no desea Guardar cambios - Sistema deshace los cambios
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 21: Administrador de Sitios

6.3.4.4.2 Diagrama de Secuencia

A continuación se muestra el Diagrama de Secuencia 16.

Diagrama de Secuencia 16: Administrador de Sitios



6.3.4.4.3 Interfaces relacionadas

Las Ventana 20 representa la interfaz relacionada con el Caso de Uso.

Ventana 20: Mantenedor Sitios

The screenshot shows a software window titled "Mantenedor Datos" with a tabbed interface. The active tab is "Sitios". Below the tabs, there is a "Proyecto" dropdown menu set to "Autopista Central". A table lists sites with columns "Nombre" and "Proyecto". The table contains three rows: "AC-01", "AC-01s1", and "AC-03". Below the table is a form for editing a site, with fields for "Id" (value: 5), "Nombre" (value: AC-03), and "Proyecto" (dropdown: Autopista Central). At the bottom of the form are "Guardar" and "Cerrar" buttons. At the bottom of the window are "Nuevo", "Editar", "Borrar", and "Volver" buttons.

6.3.4.5 Administrador Cantidad de Sistemas

6.3.4.5.1 Especificación de Caso de Uso

En la tabla 22 se muestra la especificación de Caso de Uso.

Caso de Uso	Administración de Cantidad de Sistemas
Actor Primario	Administrador
Participante e intereses	Administrador: Desea Administrar la cantidad de Sistemas que se poseen en la empresa por cada concesionaria
Precondiciones	El Administrador se identifica en el sistema
Poscondiciones	El Administrador realiza cambios en los la cantidad de

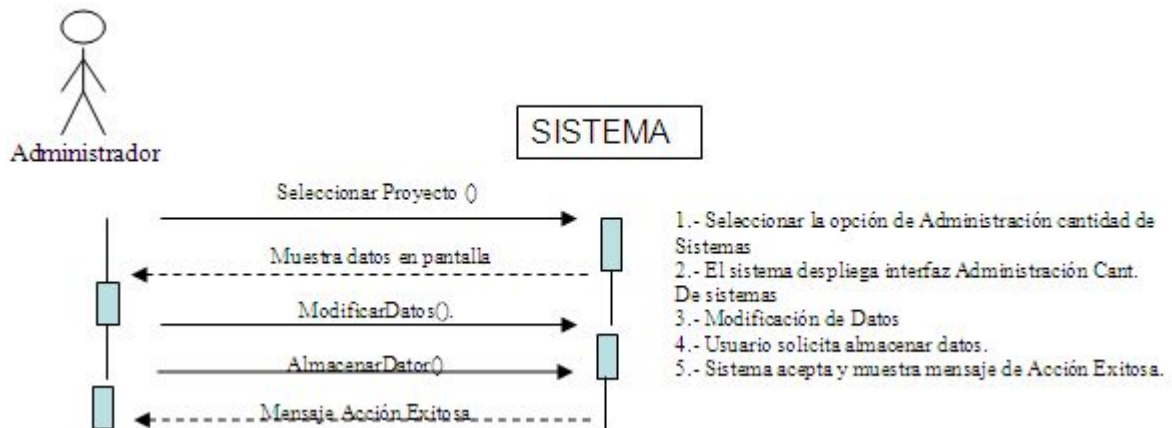
	Sistemas
Escenario Principal	<p>1.- Usuario selecciona la opción Administración de Cantidad de Sistemas</p> <p>2.- El sistema despliega interfaz Administración de Cantidad de Sistemas</p> <p>3.- Usuario selecciona el proyecto que se encuentra el sistema que se desea modificar su cantidad</p> <p>4.- Sistema despliega el listado de cantidades</p> <p>5.- Usuario selecciona el sistema a cambiar las cantidades</p> <p>6.- Usuario ingresa datos</p> <p>7.- Usuario solicita almacenar</p> <p>8.- Sistema despliega mensaje acción exitosa</p>
Extensiones	<p>5 a- Usuario solicita ingresar nuevo proyecto</p> <p>- Sistema muestra los Cuadro de Texto</p> <p>- Vuelve a 4</p> <p>7 a Usuario no desea Guardar cambios</p> <p>- Sistema deshace los cambios</p>

Tabla 22: Administrador Cantidad de Sistemas

6.3.4.5.2 Diagrama de Secuencia

A continuación se muestra el Diagrama de Secuencia 17.

Diagrama de Secuencia 17: Cantidad de Sistemas



6.3.4.5.3 Interfaces relacionadas

Las Ventana 21 representa la interfaz relacionada con el Caso de Uso.

Ventana 21: Cantidad de Sistemas

6.3.4.6 Cuentas de Usuario

6.3.4.6.1 Especificación de Caso de Uso

En la tabla 23 se muestra la especificación de Caso de Uso.

Tabla 23: Cuentas de Usuario

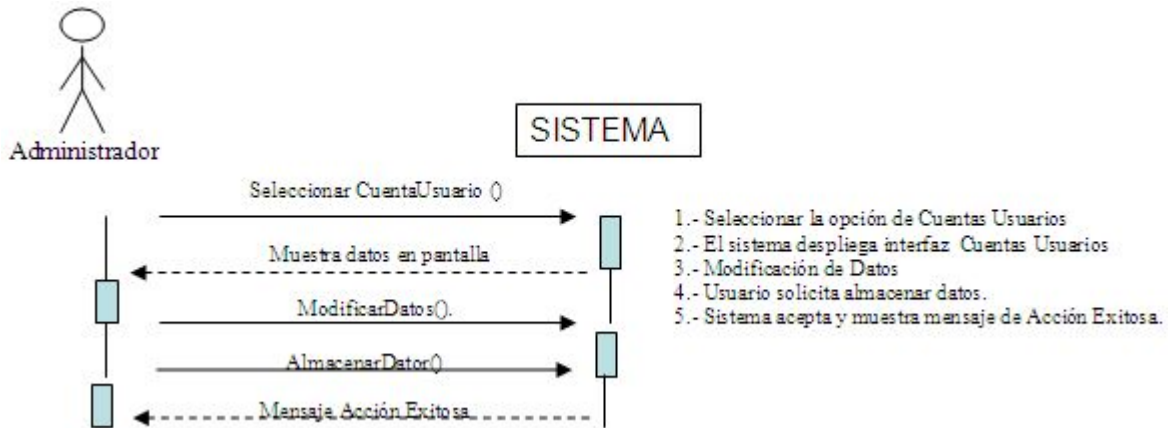
Caso de Uso	Cuentas de Usuario
Actor Primario	Administrador
Participante e intereses	Administrador: Desea Administrar las cuentas de Usuarios del Sistema
Precondiciones	El Administrador se identifica en el Sistema
Poscondiciones	El Administrador realiza cambios en las cuentas de Usuarios del Sistema

<p>Escenario Principal</p>	<ol style="list-style-type: none"> 1.- Usuario selecciona la opción Cuentas de Usuario 2.- El sistema despliega interfaz Cuentas de Usuario 3.- Usuario selecciona la cuenta de usuario que desea modificar 4.- Sistema la información en los cuadros de texto 5.- Usuario ingresa datos 6.- Usuario solicita almacenar 7.- Sistema despliega mensaje acción exitosa
<p>Extensiones</p>	<ol style="list-style-type: none"> 5 a- Usuario solicita ingresar nuevo proyecto - Sistema muestra los cuadro de texto en blanco - Vuelve a 6 6 a Usuario no desea Guardar cambios - Sistema deshace los cambios

6.3.4.6.2 Diagrama de Secuencia

A continuación se muestra el Diagrama de Secuencia 20.

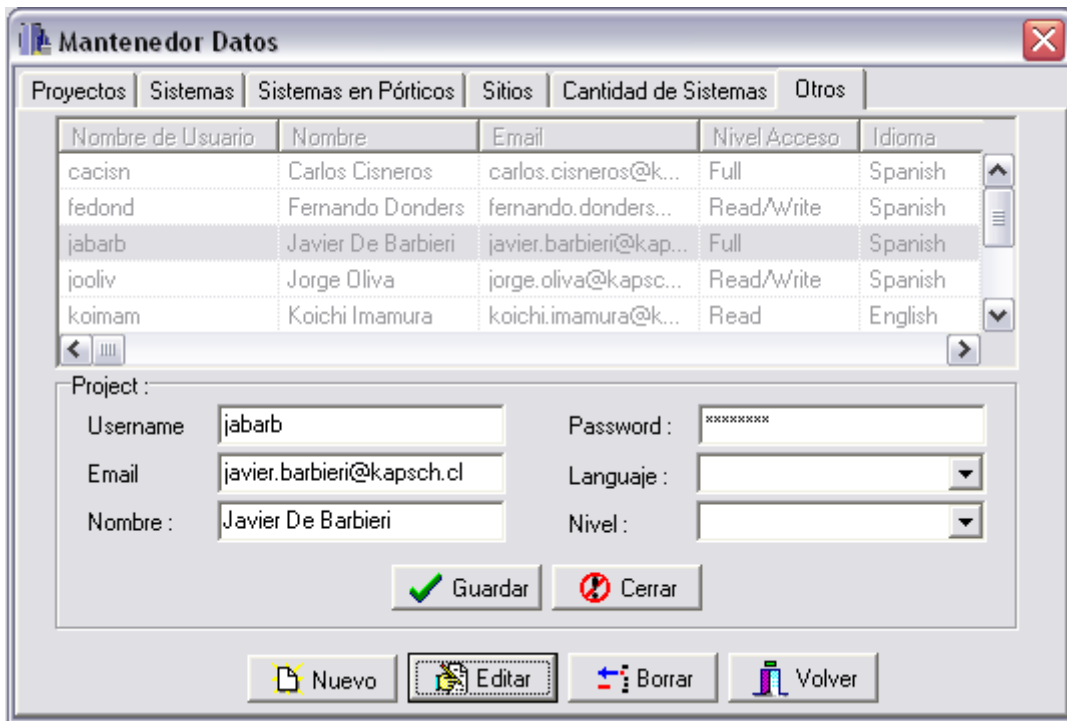
Diagrama de Secuencia 20: Administrador Usuarios



6.3.4.6.3 Interfaces relacionadas

La Ventana 22 representa la interfaz relacionada con el Caso de Uso.

Ventana 22: Administrador Usuarios



6.3.4.7 Tipo Reporte

6.3.4.7.1 Especificación de Caso de Uso

En la tabla 24 se muestra la especificación de Caso de Uso.

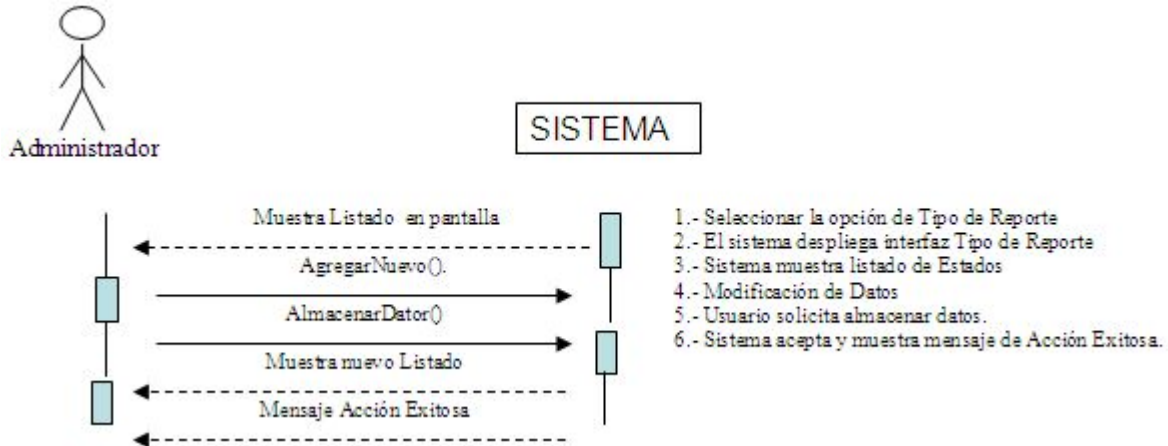
Caso de Uso	Tipo Reporte
Actor Primario	Administrador
Participante e intereses	Administrador: Desea Administrar los tipos de Reporte que se ingresan al sistema
Precondiciones	El Administrador se identifica en el sistema
Poscondiciones	El Administrador realiza cambios en los tipos de reporte
Escenario Principal	<ol style="list-style-type: none"> 1.- Usuario selecciona la opción Tipo Reporte 2.- El sistema despliega interfaz Tipo Reporte 3.- Usuario Ingresa el nuevo tipo de Repote 4.- Usuario solicita agrega el nuevo reporte 5.- Sistema agrega al listado el nuevo reporte
Extensiones	<p>3 a- Usuario solicita eliminar un reporte</p> <ul style="list-style-type: none"> - Usuario selecciona el reporte a eliminar - Usuario selecciona ELIMINAR - Sistema elimina el reporte y actualiza el listado de reportes

Tabla 24: Tipo Reporte

6.3.4.7.2 Diagrama de Secuencia

A continuación se muestra el Diagrama de Secuencia 21.

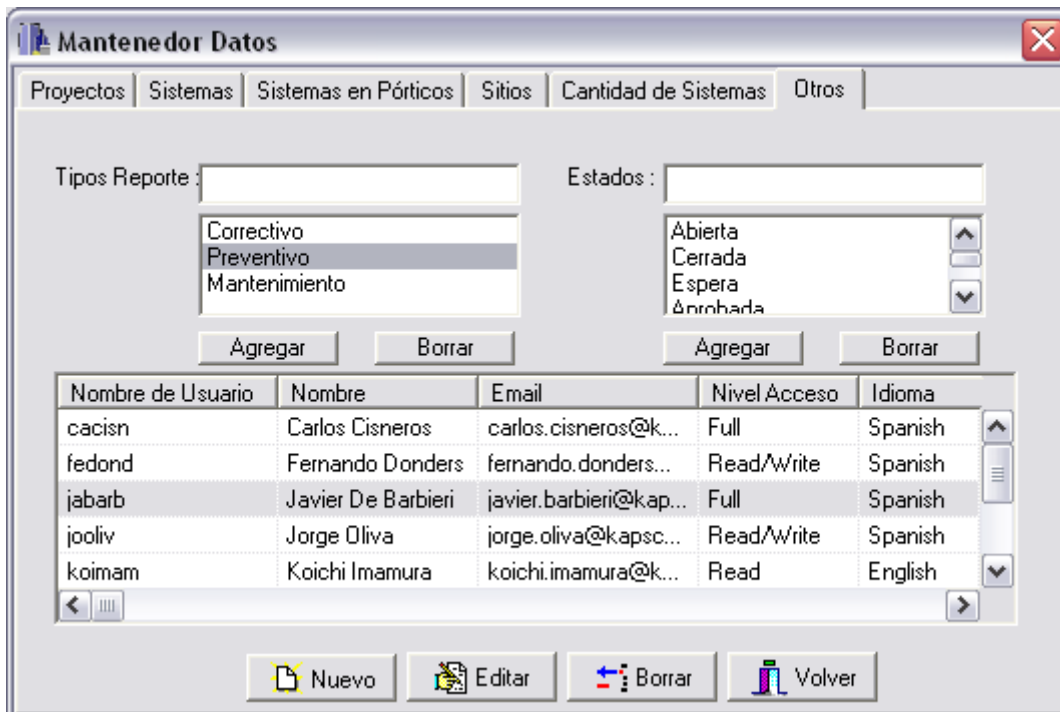
Diagrama de Secuencia 21: Tipo de Reportes



6.3.4.7.3 Interfaces relacionadas

La Ventana 23 representa la interfaz relacionada con el Caso de Uso.

Ventana 23: Tipo Reportes



6.3.4.8 Tipo de Estados

6.3.4.8.1 Especificación de Caso de Uso

En la tabla 25 se muestra la especificación de Caso de Uso.

Caso de Uso	Tipo de Estados
Actor Primario	Administrador
Participante e intereses	Administrador: Desea Administrar los tipos de Estados de las SDS
Precondiciones	El Administrador se identifica en el sistema
Poscondiciones	El Administrador realiza cambios en los Estados de SDS
Escenario Principal	1.- Usuario selecciona la opción Tipo Estados 2.- El sistema despliega interfaz Tipo Estados 3.- Usuario Ingresa el nuevo tipo de Estado 4.- Usuario solicita agrega el nuevo Estado 5.- Sistema agrega al listado el nuevo Estado
Extensiones	3 a- Usuario solicita eliminar un Estado - Usuario selecciona el Estado a eliminar

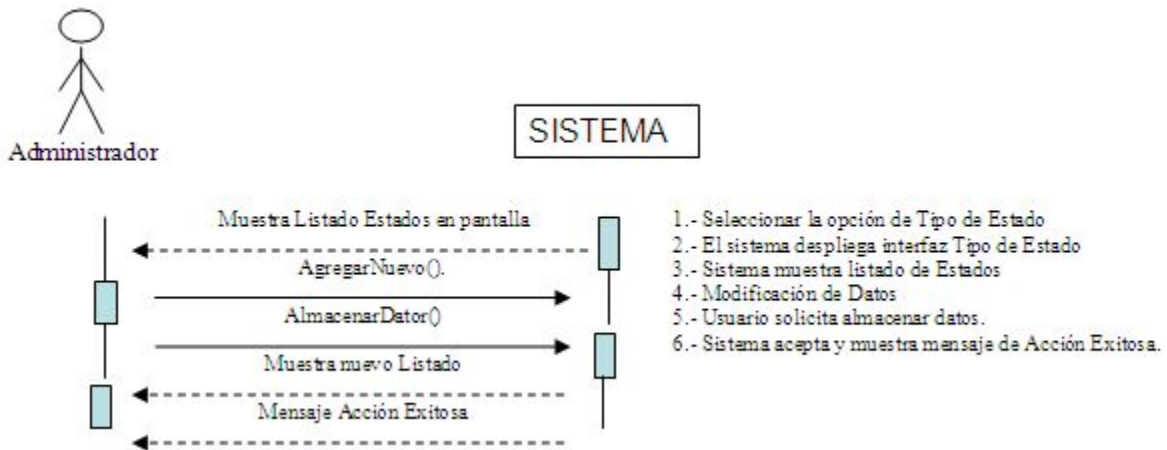
	<p>- Usuario selecciona Eliminar</p> <p>- Sistema elimina el estado y actualiza el listado de estados</p>
--	-------------------------------------------------------------------------------------------------------------------------

Tabla 25: Tipo de Estados

6.3.4.8.2 Diagrama de Secuencia

A continuación se muestra el Diagrama de Secuencia 22.

Diagrama de Secuencia 22: Administrador de Estados



6.3.4.8.3 Interfaces relacionadas

La Ventana 24 representa la interfaz relacionada con el Caso de Uso.

Ventana 24: Estados

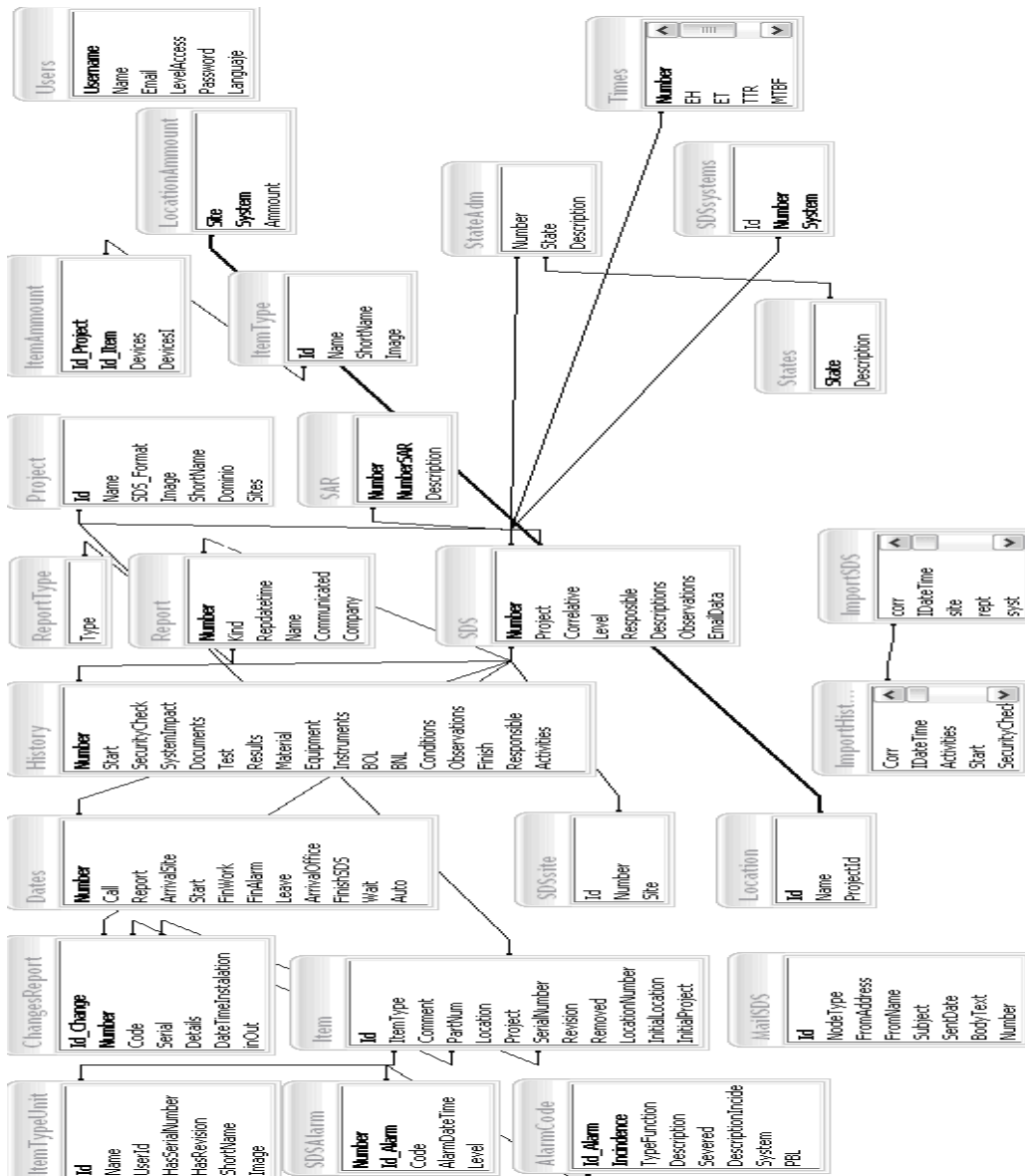


6.4 Base de Datos

6.4.1 Modelo Relacional

La figura número 15 muestra el modelo relacional del sistema SDS Manager.

Figura 15: Diagrama Relacional Base de Datos



6.4.2 Descripción de las Tablas

A continuación se describirán cada una de las tablas utilizadas en el sistema SDS Manager

6.4.2.1 Tabla SDS:

Esta tabla contiene toda la información que identifica a la SDS, cuando ésta es recién creada, como cuando esta es cerrada.

Nombre	Tipo de llave	Descripción
Number	Primary	Número de SDS
Project		Número ID de la concesionaria
Correlative		Número Correlativo de SDS
Level		Nivel de severidad 1, 2 o 3
Responsible		Responsable de los trabajos
Descriptions		Descripción de la falla
Observations		Fecha en que el reporte es llenado

6.4.2.2 Historial SDS:

Esta tabla contiene todas las actividades que realizan para la resolución de una SDS.

Nombre	Tipo de llave	Descripción
Number	Primary	Número de SDS
Stara		Inicio de Actividades
SecurityCheck		Persona quien autoriza la seguridad
SystemImpact		Sistema impactado
Documents		Documentos utilizados en el procedimiento
Test		Pruebas realizadas al sistema

Results		Resultados de las pruebas al sistema
Material		Material utilizado en el procedimiento
Equipment		Equipamiento utilizado en el procedimiento
BOL		Línea Base antigua
BNL		Línea Base nueva
Contiditions		Condiciones de trabajo
Observations		Observaciones del trabajo
Finish		Hora cierre de SDS
Responsible		Responsable del cierre de SDS
Activities		Actividades realizadas en la SDS

6.4.2.3 AlarmCode:

Esta tabla contiene a todas las alarmas de los diferentes sistemas.

Nombre	Tipo de Llave	Descripción
Id_Alarm	Primary	Identificador de Alarma
Incidente		Nivel de incidencia de la alarma (1, 2 o 3)
TypeFunction		Tipo de tiempo empleado (ET, EH, OTHER)
Description		Descripción de la alarma

Severed		Tipo de Severidad (error, warning, fatal)
DescriptionIncidence		Descripción extra de la alarma
System		Id de Sistema afectado
PBL		Línea de Base a la cual pertenece

6.4.2.4 ChangesReport:

Esta tabla contiene la información sobre los reportes de cambios de sistemas o subsistemas.

Nombre	Tipo de Llave	Descripción
Id_Change	Primary	Identificador de cambio
Number	Primary, Foreign	Número de SDS
Code		Código del sistema o subsistema afectado
Serial		Número serial del sistema afectado
Details		Detalle del cambio
DateTimeInstalation		Fecha y hora de instalación o desinstalación
inOut		Si el sistema salió u entró

6.4.2.5 Dates:

La siguiente tabla contiene las todas las fechas y horas que se manejaran en el sistema.

Nombre	Tipo de Llave	Descripción
Number	Primary	Número de SDS
Call		Fecha y Hora de llamada a la compañía
Report		Fecha y Hora en que se crea el reporte
ArrivalSite		Fecha y Hora en que se llega al sitio
Stara		Fecha y Hora en que se comienza a trabajar en el sitio
FinWork		Fecha y Hora en que se finaliza los trabajos
Leave		Fecha y Hora en que se sale del sitio de trabajo
ArrivalOffice		Fecha y Hora en que llega de vuelta a la oficina
Wait		Tiempo en horas y minutos que se utiliza en esperar
Auto		Si se calcula los tiempos automáticamente o no (“Yes” o “No”)

6.4.2.6 ImportSDS:

Tabla en la cual se almacenan los datos provenientes del subsistema SDSEmail para la importación directa desde el correo.

Nombre	Tipo de Llave	Descripción
Corr	Primary	Número de la SDS

IDateTime		Fecha y Hora de inicio de la incidencia
Site		Sitio(s) afectado
Rept		Datos del reporte
Syst		Sistemas afectados
Leve		Nivel de incidencia de SDS
Desc		Descripción de la incidencia
FDateTime		Fecha y Hora del reporte
Stat		Estado de la SDS
Send		Enviado por
Sing		Recibido por
Sent		Enviado a
Comp		Compañía enviado
Proj		Enviado a Proyecto
Ok		Fue guardada como SDS

6.4.2.7 ImportHistory:

Tabla en la cual se almacena los datos de historial cuando son importados.

Nombre	Tipo de Llave	Descripción
--------	---------------	-------------

Number	Primary	Número de SDS
IdateTime		Fecha y Hora de inicio de actividades
Activities		Actividades realizadas en la SDS
Star		Inicio de actividades
SecurityCheck		Persona quien autoriza la seguridad
SystemImpact		Sistema impactado
Documents		Documentos utilizados en el procedimiento
Test		Pruebas realizadas al sistema
Results		Resultados de las pruebas al sistema
Material		Material utilizado en el procedimiento
Equipment		Equipamiento utilizado en el procedimiento
BOL		Línea Base antigua
BNL		Línea Base nueva
Contiditions		Condiciones de trabajo
Observations		Observaciones del trabajo
Finish		Hora cierre de SDS
Fdatetime		Hora de finalizacion de SDS
Responsible		Responsable del cierre de SDS

6.4.2.8 Item:

Listado de sistemas y subsistemas que existen en la compañía, ya sea con su parte número y su serial number.

Nombre	Tipo de Llave	Descripción
Id	Primary	Identificador
ItemType		Identificador de tipo de sistema
Comment		Comentario sobre el cambio
Location		Ubicación
Project		Identificador de proyecto
SerialNumber		Número de serie
Revision		Que tipo de revisión se realizó
Removed		Si fue removido o no del sitio
LocationNumber		Número de sitio
InitialLocation		Lugar donde fue instalado por primera vez
InitialProject		Proyecto inicial donde fue instalado

6.4.2.9 ItemAmmount:

Cantidad de SISTEMAS por proyecto.

Nombre	Tipo de	Descripción
--------	---------	-------------

Llave		
Id_Project	Primary	Identificador de proyecto
Id_Item	Primary	Identificador de ítem
Devices		Total de dispositivos del proyecto
DevicesI		Total de dispositivos instalados del proyecto

Nota: Esta tabla esta vinculada con ItemType que corresponde solamente a los sistemas importantes de los pórticos, no a los subsistemas.

6.4.2.10 ItemType:

Corresponden a todos los Sistemas más importantes de los pórticos (ALC, VDC, VDC SU, VR, VR SU, etc.).

Nombre	Tipo de Llave	Descripción
Id	Primary	Identificador del ítem
Name		Nombre de sistema
ShortName		Nombre del sistema abreviado
Image		Número de imagen asociada al sistema

6.4.2.11 ItemTypeUnit:

Corresponde a todos sistemas y subsistemas de los pórticos.

Nombre	Tipo de Llave	Descripción
Id	Primary	Identificador del Item
Name		Nombre de Sistema
UserId		Especifica quien creo este ItemType
HasSerialNumber		Indica si tiene serial número o no
HasRevision		Indica si ha sido revisado o no
ShortName		Nombre del sistema abreviado
Image		Número de imagen asociada al sistema

6.4.2.12 Location:

Listado de Sitios y pórticos.

Nombre	Tipo de Llave	Descripción
Id	Primary	Identificador del sitio
Name		Nombre del sitio
ProjectId		Identificador del proyecto

6.4.2.13 LocationAmmount:

Cantidad de sistemas por sitio.

Nombre	Tipo de Llave	Descripción
Site	Primary	Nombre del sitio
System		Nombre del sistema
Ammount		Cantidad de sistemas

6.4.2.14 Project:

Almacena datos sobre los proyectos (concesionarias)

Nombre	Tipo de Llave	Descripción
Id	Primary	Identificador del proyecto
Name		Nombre de proyecto
SDS_Format		Formato de SDS según proyecto
Image		Imagen asociada al proyecto
ShortName		Nombre abreviado del proyecto
Dominio		Dominio de email del remitente (ej: @autopistacentral.cl)
Sites		Número total de sitios

6.4.2.15 Report:

Datos de los reportes que se producen cuando existe una SDS.

Nombre	Tipo de Llave	Descripción
Number	Primary	Número de SDS
Kina		Tipo de reporte
Repdatetime		Fecha y hora del reporte
Name		Nombre de la persona que recibe la información
Comunnicated		Comunicado a
Company		Compañía a que fue enviado (Kapsch)

6.4.2.16 ReportType:

Tipos de reportes que existen (open, close, wait, etc.).

Nombre	Tipo de Llave	Descripción
Type		Tipo de reporte

6.4.2.17 SAR:

Número de SAR asociado(s) a una SDS.

Nombre	Tipo de Llave	Descripción
---------------	----------------------	--------------------

Number	Primary	Es la SDS asociada a la actividad
NumberSAR		Número SAR asociado
Descripción		Descripción del SAR asociado

6.4.2.18 SDSAlarm:

Alarmas asociadas a la SDS.

Nombre	Tipo de Llave	Descripción
Number	Primary	Número de SDS asociada
Id_Alarm	Primary	Número interno de ubicación de alarma
Code		Código de alarma asociada a SDS
AlarmDateTime		Fecha y Hora de alarma
Level		Nivel de la alarma

6.4.2.19 SDSSite:

Sitios asociados a las SDS.

Nombre	Tipo de Llave	Descripción
Id	Primary	Identificador interno de Sitio
Number	Foreign	Número de SDS

Site		Sitio donde fue la SDS
------	--	-------------------------------

6.4.2.20 SDSSystems:

Sistema asociado a SDS.

Nombre	Tipo de Llave	Descripción
Id		Identificador interno de sistema
Number	Primary	Número de SDS
System		Sistema afectado por la SDS

6.4.2.21 StateAdm:

Estado de la SDS.

Nombre	Tipo de Llave	Descripción
Number	Primary	Es la SDS asociada a la actividad
Description		Descripción de la actividad en cuestión
State	Foreign	Nombre del estado en que se encuentra

6.4.2.22 Status:

Muestra los diferentes tipos de estados de las SDS.

Nombre	Tipo de Llave	Descripción
State	Primary	Estado de las SDS
Description		Descripción de la actividad en cuestión

6.4.2.23 Times:

Tiempos de SDS separados por tipos.

Nombre	Tipo de Llave	Descripción
Number	Primary	Número de SDS
EH		Tiempos EH
ET		Tiempos ET
TTR		Tiempos TTR
MTBF		Tiempos MTBF o Others
Factor		Factor aplicado
Extra		Tiempo extra

6.4.2.24 Users:

Datos referentes a los usuarios del sistema.

Nombre	Tipo de Llave	Descripción
--------	---------------	-------------

Username	Primary	Nombre de usuario
Name		Nombre de la persona que ocupa el Software
LevelAccess		Nivel de Acceso
Password		Contraseña (encriptada)
Lenguaje		Si es en ingles o español

6.5 Detalles de Implementación

La implementación de este sistema ha sido en forma sistemática. Cada una de las pantallas de la interfaz con el sistema se comienza con el diseño físico de las mismas. Esto comprende en ubicar los diferentes cuadros de texto, listas, botones, entre otros, en el lugar más estratégico. Esto no es una tarea sencilla, ya se que debe ser muy cuidadoso y creativo para poder ubicar en que lugar debe ir cada uno de los componentes visuales, para que el uso del software sea lo mas intuitivo posible [4].

6.5.1 BORLAND C++ BUILDER 5.0

Junto con la interfaz, se debe tener presente que se debe realizar toda una ingeniería interna para que los diferentes componentes que se crearon funcionen correctamente. Para esto se deben realizar varios algoritmos de codificación en cada una de estas interfaces. El sistema que utiliza C++ BUILDER es generar dos archivos por cada formulario que se utiliza.

- El primero que se crea es el archivo C++ (extensión CPP) en el cual se trabaja con dicho lenguaje y con algunas funciones propias de C++ BUILDER. En este archivo se pueden crear funciones y llamadas a otras. Para acceder a los componentes físicos del formulario, sólo basta con escribir el nombre del componente que se asignó, y escribir el signo de

puntero (->). Ahí automáticamente se desplegarán todas las funciones y objetos relacionados con ese componente.

Por ejemplo, existe un cuadro de texto llamado TextBox1, para guardar en la variable de texto Nombre el contenido de ese campo, se debe escribir como.

Nombre = TextBox1->Text;

Lo que sucede internamente, es que la información contenida en el TextBox1 es referenciada gracias a un puntero que solicita rescatar solamente el texto que está contenido dentro de él.

En este tipo de archivo también se pueden crear funciones. Estas funciones deben ser declaradas de la siguiente manera.

<tipo> __fastcall <nombre clase>::"nombre de funcion"(parámetro 1, ... parámetro n){...}

El tipo que se menciona anteriormente, puede ser alguno de los tipos de variables u objetos que existen, o también puede ser simplemente un VOID. Es válido señalar que si se utiliza un tipo diferente a VOID se deberá hacer un retorno de valor.

El nombre de la clase corresponde al nombre que se le asignó internamente a dicha clase.

A continuación se muestra un ejemplo.

void __fastcall TReports::fill_grid(String mes, String ano, String proj, String lev, String st){...}

Es válido señalar que al crear una función debe ser declarada en su archivo HEADER.

Al crear un nuevo formulario el archivo de C++ tiene la siguiente información por defecto.

```

#include <vcl.h>
#pragma hdrstop
#include "Unit6.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm6 *Form6;
//-----
__fastcall TForm6::TForm6(TComponent* Owner)
    : TForm(Owner)
{
}
//-----

```

Se aprecia que contiene el include a “Unit6.h” que corresponde a su archivo header y que contiene su propio constructor.

Para poder crear una función, se debe escribir de la forma anteriormente descrita.

- El otro tipo de archivo que se genera automáticamente en C++ BUILDER corresponde al mencionado recientemente Archivos Header (archivo con extensión H). En estos archivos se generan automáticamente todos los componentes necesarios para la compilación. Este archivo, como dice su nombre, es el archivo de cabecera, donde se deben declarar todas las funciones creadas y se deben agregar las importaciones a archivos. Acá también se pueden crear las variables globales para ser utilizadas en el presente formulario.

El formato que se utiliza para declarar una función en el archivo header es el siguiente.

<Tipo> __fastcall “nombre función” (parámetro 1, ..., parámetro N);

Un ejemplo para la función anteriormente mostrada es.

```
void __fastcall fill_grid(String mes, String ano, String proj, String lev, String st);
```

```

#ifndef Unit6H
#define Unit6H
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
//-----
class TForm6 : public TForm
{
__published: // IDE-managed Components
    void __fastcall FormCreate(TObject *Sender);
private: // User declarations
public: // User declarations
    __fastcall TForm6(TComponent* Owner);
};
//-----
extern PACKAGE TForm6 *Form6;
//-----
#endif

```

A simple vista este archivo parece muy complicado de entender, pero no lo es. Hay que tener presente que existen cuatro partes claves en este tipo de archivo. La primera corresponde a la referencia a los archivos de importación, la segunda a los componentes que se crean en forma pública, la tercera los componentes que se utilizan en forma privada y la última corresponde a los componentes que se utilizarán en forma pública. La diferencia entre componentes en forma pública y publicada, es que los públicos corresponden a los componentes no visuales, sino solamente internos. En cambio las publicadas son exclusivamente los componentes visuales.

6.5.2 Manejo de las Bases de Datos

El manejo de las bases de datos se ha simplificado mucho con la utilización de una clase que se disponen en la empresa. Esta clase llamada MTDATABASE, contiene algunos componentes muy fáciles de utilizar.

Por ejemplo para conectar un programa a una base de datos, es necesario crear un objeto del tipo MTADODBASE. Posteriormente crear una nueva referencia al objeto del tipo MTADODBASE. Para conectar a la base de datos, es necesario utilizar el objeto ya creado y utilizar OPEN seguido de la dirección de conexión. Y para realizar consultas simplemente utilizar la función EXECUTE.

Por ejemplo.

```
MTADODBase *Db; // crea el objeto de nombre Db
Db = new MTADODBase(this); // crea una nueva referencia al objeto Db.
Db->Open("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=SDS.mdb;Persist Security info=False") //
abre la conexión a la base de datos.
TStringList arr = Db->Execute ("select * from SDS"); // realiza la consulta e inserta la información en el
string list.
```

Esta clase ha sido muy efectiva para la creación de este software.

La tabla 26 de consultas muestra las consultas más complejas que se utilizan en el software.

Formulario	Consulta	Detalle
Report	Select * from project, SDS, Times, dates, report, stateadm, states where sds.number = dates.number and sds.number = times.number and sds.number = report.number and stateadm.number = sds.number and States.state = stateadm.state and project.id = sds.project and stateadm.activo = 'Yes' and (project.name = "" + proj + "" or "" + proj + "" = "") and (states.description = "" + st + "" or "" + st + "" = "")	Esta consulta es relativamente larga, ya que esta consulta muchas tablas referentes a las SDS.

MTBF	Select * from dates, sdssystems, sds, project, itemtype where dates.number = sdssystems.number and sds.project = project.id and sds.number = dates.number and itemtype.id = sdssystems.system and call > 0	Esta consulta no es tan extensa y no posee valores para filtrarla, ya que los resultados de ésta son almacenados y procesados en una matriz. Este formulario demora unos segundos en abrir, ya que llena la matriz cada vez que se abre el formulario.
TimesEHET	select * from times, dates, SDS, project where times.number = dates.number and dates.number = sds.number and project.id = sds.project	Esta consulta no es extensa, ya que no necesita tener valores para ser filtrada. Los resultados son almacenados y procesados en una matriz. Este formulario no demora mucho en abrir, ya que los datos que se almacenan en dicha matriz no son muchos.

Tabla 26: Consultas

6.5.3 Programación de Interfaces

6.5.3.1 Main Menu

Esta interfaz se subdivide en cuatro páginas diferentes. En este formulario se utilizó un componente que incluye C++ Builder que permite trabajar separadamente cada uno de las páginas separadamente.

En las figuras 16, 17, 18 y 19 se muestra las diferentes interfaces del Menú Principal del SDS Manager.

Figura 16: Main Menu 1 “SDS”

SDS Manager

File Edit Report Options View Help

SDS Number

SDS | History | Times | Systems Changes

Main

Correlative :

Project :

Sites :

Systems :

Description :

Comments :

Signature :

Alarms

Code Alarm :

Alarm (D/T) :

Level :

Report

Type :

Level :

Report (D/T) :

Reported by :

Notified to :

Company :

Times (in hours)

Diference : Time ET :

Work : Time TTR :

Wait : Other :

Transportation : Extra :

Time EH : Factor Extra :

Alarm List

Id	Code	Level	Severed	Description	Time	Date	Function Type	System

Figura 17: Main Menu 2 “History”

SDS Manager

File Edit Report Options View Help

Solicitud de Servicio | Detalles SDS | Tiempos | Cambios de Sistemas

Acciones de Mantenimiento

SAR : Agregar

Comienzo :

Impacto en Sistema :

Chequeo Seguridad :

Documentos :

Actividades :

Pruebas :

Resultado :

Materiales :

Equipamento :

Instrumentos :

Linea Base Antigua :

Linea Base Nueva :

Condiciones :

Observaciones :

Ejecutado por :

Responsable :

Figura 18: Main Menu 3 "Times"

SDS Manager

File Edit Report Options View Help

SDS Number

SDS History Times Systems Changes

Initial Times

Call Time : 12/03/2007 18:27:45

Report Time : 12/03/2007 18:27:45

Arrival to site : 12/03/2007 18:27:45

Time wait before : 0:00:00

Start to Work : 12/03/2007 18:27:45

Change

Final Times

Finish to Work : 12/03/2007 18:27:45

Finish Alarm : 12/03/2007 18:27:45

Leave the site : 12/03/2007 18:27:45

Arrival to office : 12/03/2007 18:27:45

Finish SDS : 12/03/2007 18:27:45

Change

Times (in hours)

Diference :

Work :

Wait :

Transportation :

Automatic

Time EH :

Time ET :

Time TTR :

Other :

Extra :

Factor Extra : %

Save SDS Close

Figura 19: Main Menu 4 "Systems Changes"

SDS Manager

File Edit Report Options View Help

Número SDS :

Solicitud de Servicio Detalles SDS Tiempos Cambios de Sistemas

Seleccione

Select

Sistema :

Fecha Hora cambi 27/02/2007 9:50:00

Detalles :

Sistema Saliente

Código :

Número Serie :

Rev :

Sistema Entrante

Código :

Número Serie :

Rev :

Agregar Limpiar

List of Systems Changes

Id	Sistema	Fecha Hora	Código Sale	Serial Sale	Rev	Código Entra	Serial Entra	Rev	Detalle

Editar Borrar

Guardar SDS Cerrar

Para programar cada una de estas interfaces de una forma más eficiente y práctica, se pueden separar en páginas cada una de las pantallas.

A continuación se muestra el proceso para crear la primera de las páginas.

1) Se crea el FRAME de SDS_TAB y se agregan los botones, cuadros de texto, grillas, accesos a otros componentes del Software, etc. Este FRAME es tratado como un formulario más, incluso tiene su propio C++ file y HEADER file.

Las figuras 20, 21, 22 y 23 muestran los diferentes TAB creados.

Figura 20: SDS Tab

Code	Id	Level	Severity	Description	Time	Date	Function Type	System

Figura 21: SDS History Tab

Code	Id	Level	Severity	Description	Time	Date	Function Type	System

Figura 22: SDS Times Tab

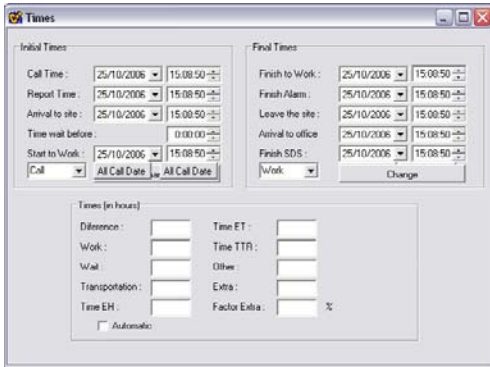
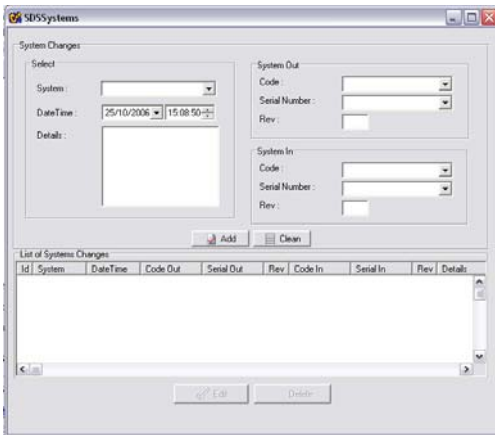


Figura 23: SDS Systems Tab



2) Este FRAME es programado en el archivo de C++ y si se crean funciones deben ser declaradas en su respectivo HEADER File.

La tabla 27 muestra una función utilizada en SDS_TAB.

```

void __fastcall TSds_Tab::ins_sds() // Agrega a la Base de datos una nueva SDS
{
    AnsiString sds = "insert into SDS values (" + Trim(MenuSDS->SDS->Text) + ", " +
    IntToStr(num_proj(ComboBox2->Text)) + ", " +
    corr->Text + ", " +
    IntToStr(StrToIntDef(Trim(level->Text),0)) + ", " +
    Trim(sign->Text) + ", " +
    Trim(descripcion->Text) + ", " +
    Trim(obs->Text) + ", " + company->Text + ")"; // insert into SDS table
    MenuSDS->Db->Execute(sds); // insert into report
    AnsiString tim = "insert into Times values (" + Trim(MenuSDS->SDS->Text) + ", " +
    Trim(eh->Text) + ", " +
    Trim(et->Text) + ", " +
    Trim(ttr->Text) + ", " +
    Trim(mtbef->Text) + ", " +
    Trim(factor->Text) + ", " +
    Trim(extra->Text) + ")"; // insert into times
    MenuSDS->Db->Execute(tim); // insert into times
    AnsiString rep = "insert into Report values (" + Trim(MenuSDS->SDS->Text) + ", " +
    Trim(type->Text) + ", " +
    rep->Time.TimeString() + ", " +
    rep->Date.DateString() + ", " +
    Trim(report->Text) + ", " +
    Trim(notified->Text) + ", " + ")";
    MenuSDS->Db->Execute(rep); // insert into report
    rep = "insert into stateadm values (" + Trim(MenuSDS->SDS->Text) + ", " + Edit3->Text + ", " + ")";
    MenuSDS->Db->Execute(rep);
    ins_sds_site();
    ins_sds_system();
    fill_alarm();
    ins_sds_alarm();
}

```

Tabla 27: SDS_TAB

Este código que se mostró anteriormente corresponde a la función que se utiliza para insertar una nueva SDS dentro del sistema SDS Manager. Cada una de las variables que aparecen con una ‘->’ corresponde a un puntero del Cuadro de Texto del formulario SDS_TAB.

3) Se agrega el componente de tabuladores y dentro de la primera pestaña se agrega otro componente llamado “FRAMES”, el cual permite ingresar el FRAME ya programado dentro del principal. Si uno quisiera trabajar dentro del mismo formulario principal, se debe hacer referencia a los punteros de los diferentes componentes que existen en los FRAMES. A modo de ejemplo se muestra la tabla 28 con un trozo de código sacado del formulario principal del programa.

Tabla 28: Frames

```

TMenuSDS *MenuSDS;
__fastcall TMenuSDS::TMenuSDS(TComponent* Owner)
    : TForm(Owner)
{
    // Este es el constructor del formulario, que se ejecutará al iniciar el programa
    Db = NULL;
    Screen->Cursor = crHourGlass;
    splash->pro->Position++;
    splash->cant->Caption = IntToStr(splash->pro->Position * 100 / 15);
}
-----
void __fastcall TMenuSDS::FormCreate(TObject *Sender)
{
    // Esta function es ejecutada automáticamente por C++ Builder cada vez que se crea el formulario.
    if (splash != NULL) splash->Lbl->Caption = "Opening database...";
    Application->ProcessMessages();

    Save = false;
    first = true;
    Db = new MTADODBase(this);
    if (Db->Open("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=SDS.mdb;Persist Security
Info=False")) // open the database, SE UTILIZA ESTE Db EN TODO EL PROGRAMA.
    {
        TStringList *arr = Db->Execute("Select * from ReportType"); // get the list of the Reports Type
        for (int i=1; i < arr->Count;i++)
        {
            AnsiString Str= Db->Str[arr]["Type"][i]; // reading the database
            SDSTAB->type->Items->Add(Str); // set the information of the Reports Type to the
Combobox
        }
        mat = Db->Execute("select * from project");
        delete arr; // kill the tstringlist
    }
    else
        MessageDlg("There's a problem with the Database", mtWarning, TMsgDlgButtons() <<
mbOK, 0);
    int i = Import->waiting();
    bar->SimpleText = "There's " + IntToStr(i) + " SDSs in Waiting state!";
    SYSTAB->systemas();
    Screen->Cursor = crDefault;
    splash->pro->Position++;
}

```

En este código se muestran los comandos que se utilizan al ser ejecutado el programa. Es válido señalar que se crea un objeto llamado “Db” para la conexión a la base de datos, y se utilizará en todo el programa, sin necesidad de abrir nuevamente la base de datos.

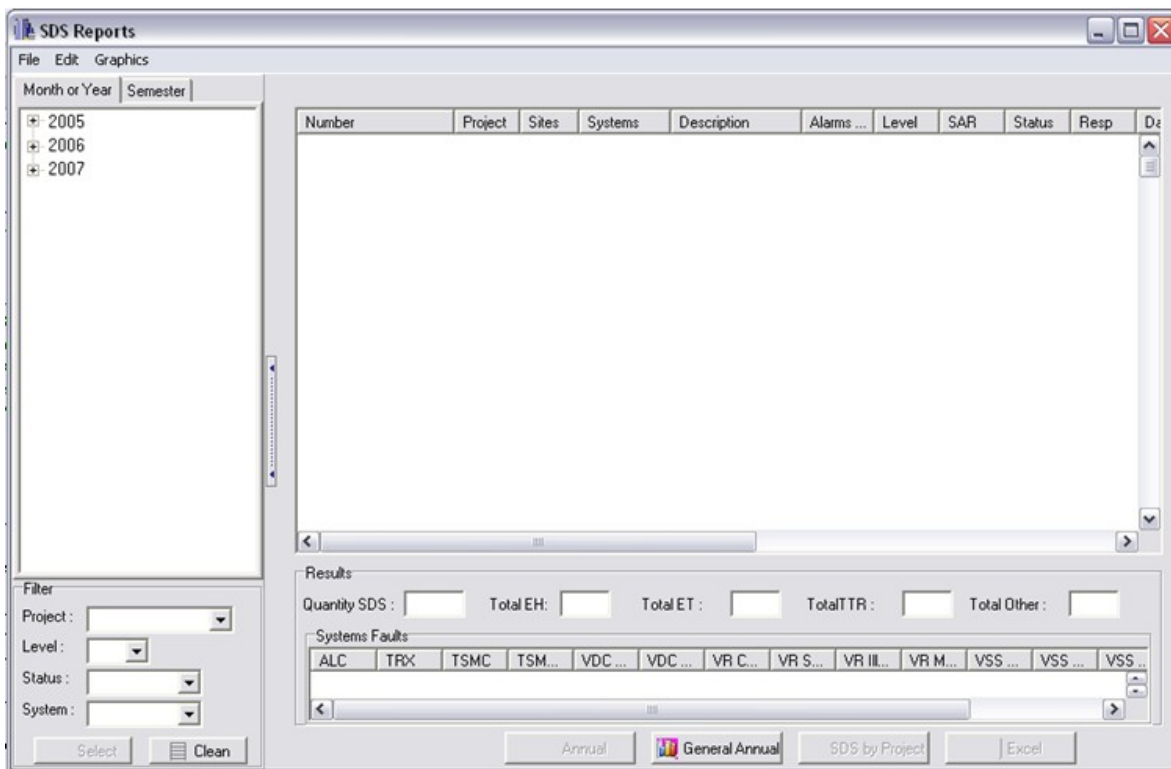
6.5.3.2 SDS Report

Esta interfaz se subdivide en 2 partes: la primera se utiliza para seleccionar el año, mes o semestre. Más abajo existe un filtro de proyecto, nivel, estado y sistema; y la segunda parte se utiliza para mostrar las tablas de todas las SDS y abajo los sistemas afectados.

El funcionamiento de esta interfaz es relativamente simple: al seleccionar un año, se desplegará el listado de meses. Adicionalmente se puede seleccionar un mes en específico a consultar. Si se desea se puede filtrar la consulta en el cuadrado inferior izquierdo. Ahí existen 4 tipos diferentes de filtros. Posteriormente se debe presionar el botón de “SELECT” y se desplegará al costado derecho el listado de SDS que corresponda.

La figura 24 muestra la interfaz gráfica.

Figura 24: Interfaz Grafica de Reportes

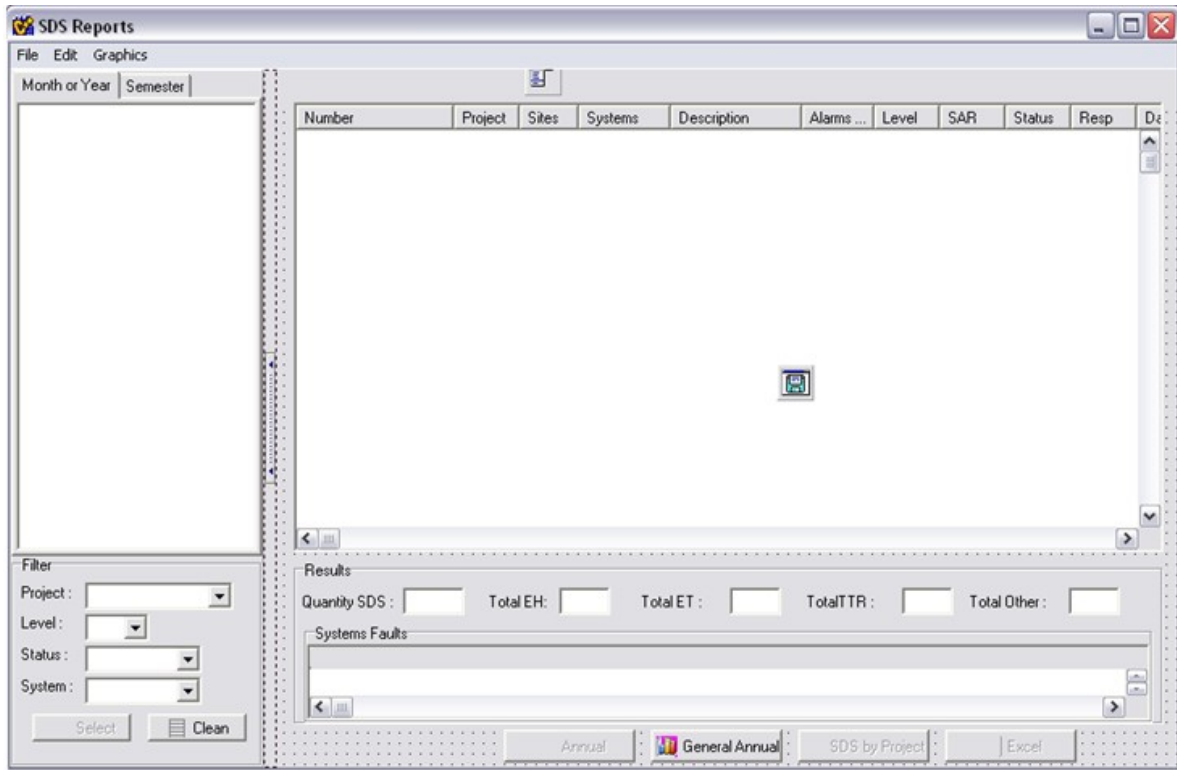


La programación se realizó del siguiente modo.

- 1) Se crearon las interfaces con el editor de C++ Builder, agregando tablas, árboles de vista (costado superior izquierdo), COMBOBOX (costado inferior izquierdo) y botones.

La figura 25 muestra el diseño de la interfaz.

Figura 25: Diseño Interfaz



2) La codificación en sí se realizó una vez que la pantalla o formulario fue terminado.

La tabla 29 muestra las funciones que permiten el llenado del listado de SDS.

Tabla 29: Funciones Listado SDS

```

void __fastcall TReports::fill_grid(String mes, String ano, String proj, String lev, String st, String
syst, int sem)
{
    TDateTime dt;
    double fecha;
    String num, aux, aux2, aux3;
    for (int i = 0; i < 25; i++)
        matriz[i] = 0;
    float et = 0.0, eh = 0.0, ttr = 0.0, ot = 0.0;
    int fl = 0;
    grilla->Clear();

    String sql = "Select * from project, SDS, Times, dates, report, stateadm where sds.number =
dates.number and sds.number = times.number and sds.number = report.number and
stateadm.number = sds.number and project.id = sds.project"; // get the list of the Reports Type
    if (proj != "")
        sql = sql + " and project.name = " + proj + "";
    if (lev != "")
        sql = sql + " and sds.level = " + lev;
    if (st != "")
        sql = sql + " and stateadm.state = " + st + "";
    if (syst != "")
        ar6 = MenuSDS->Db->Execute ("select * from sdssystems, itemtype where ItemType.id
= SDSsystems.System and itemtype.shortname = " + syst + "");
    else
        ar6 = MenuSDS->Db->Execute ("select * from sdssystems, itemtype where ItemType.id
= SDSsystems.System");
    ar = MenuSDS->Db->Execute (sql + " order by SDS.number");
    for (int i = 1, j = 0; i < ar->Count; i++)
    {
        String FECHA = MenuSDS->Db->Str[ar][i];
        int Ano = open_sds->year(StrToDouble(FECHA)), Mes = open_sds-
>month(StrToDouble(FECHA));
        if (Ano == StrToInt(ano))
        {
            if (mes == "" || (Mes == month_num(mes) && sem == 0) || (Mes < 7 && sem ==
1) || (Mes >= 7 && sem == 2))
            {
                if ((MenuSDS->Db->Str[ar][i] == IntToStr(num_proj(proj)) || proj
== "") &&
                    (MenuSDS->Db->Str[ar][i] == st || st == "")) &&

```

```

(MenuSDS->Db->Str[arr]["Level"][i] == lev || lev == "")
{
num = MenuSDS->Db->Str[arr]["SDS.Number"][i];
grilla->NumRows++;
fl ++; // count the total number of sds's
grilla->Cell[j][0] = MenuSDS->Db->Str[arr]["SDS.Number"][i];
grilla->Cell[j][1] = MenuSDS->Db->Str[arr]["ShortName"][i];
grilla->Cell[j][4] = clean_str(MenuSDS->Db->Str[arr]["Descriptions"][i]);
grilla->Cell[j][6] = MenuSDS->Db->Str[arr]["Level"][i];
grilla->Cell[j][8] = MenuSDS->Db->Str[arr]["state"][i];
grilla->Cell[j][9] = MenuSDS->Db->Str[arr]["responsible"][i];
dt = FECHA.ToDouble();
grilla->Cell[j][10] = dt.DateTimeString();
aux = MenuSDS->Db->Str[arr]["FINISHSDS"][i]; dt = aux.ToDouble();
grilla->Cell[j][11] = dt.DateTimeString();
grilla->Cell[j][16] = MenuSDS->Db->Str[arr]["et"][i];
if (MenuSDS->Db->Str[arr]["et"][i] != "")
    et = et + StrToDouble(MenuSDS->Db->Str[arr]["et"][i]);
grilla->Cell[j][15] = MenuSDS->Db->Str[arr]["eh"][i];
if (MenuSDS->Db->Str[arr]["eh"][i] != "")
    eh = eh + StrToDouble(MenuSDS->Db->Str[arr]["eh"][i]);
grilla->Cell[j][17] = MenuSDS->Db->Str[arr]["ttr"][i];
if (MenuSDS->Db->Str[arr]["ttr"][i] != "")
    ttr = ttr + StrToDouble(MenuSDS->Db->Str[arr]["ttr"][i]);
grilla->Cell[j][18] = MenuSDS->Db->Str[arr]["mtbf"][i];
if (MenuSDS->Db->Str[arr]["mtbf"][i] != "")
    ot = ot + StrToDouble(MenuSDS->Db->Str[arr]["mtbf"][i]);
aux = "";
}

total->Text = IntToStr(fl);
fill_group();
ET->Text = FloatToStrF(et,ffGeneral,8,2);
EH->Text = FloatToStrF(eh,ffGeneral,8,2);
TTR->Text = FloatToStrF(ttr,ffGeneral,8,2);
OTHER->Text = FloatToStrF(ot,ffGeneral,8,2);
Screen->Cursor = crDefault;
}

```

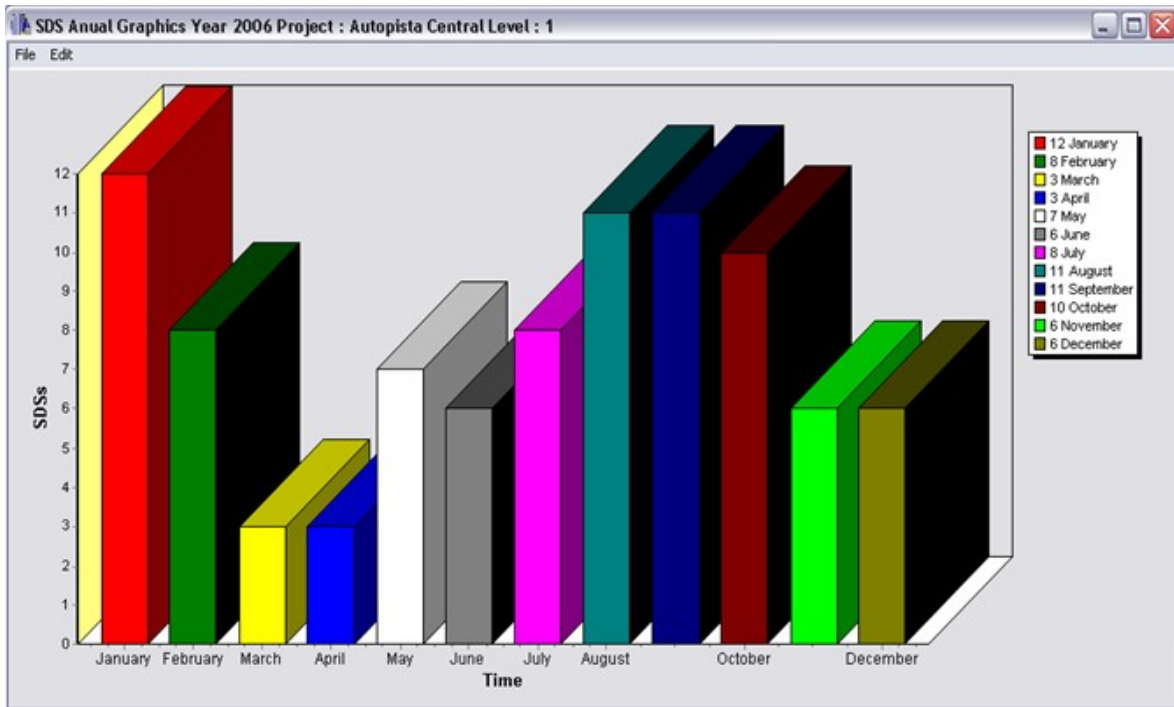
Junto con la posibilidad de ver el listado de SDS, también se da la opción de mostrar tres tipos diferentes de gráficos.

6.5.3.2.1 SDS Annual Graphics

Este gráfico de barras muestra todas las SDS que ocurrieron en el año que está seleccionado.

En la figura 26 se muestra el gráfico resultante.

Figura 26: Grafico anual

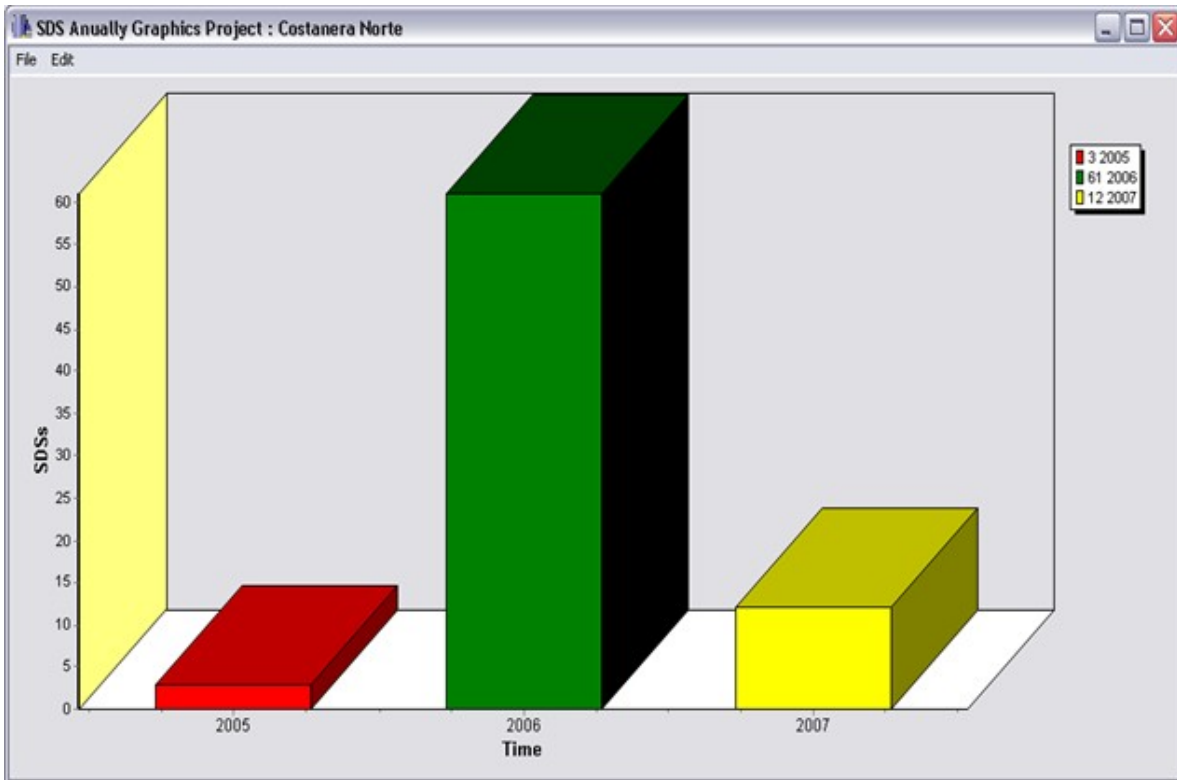


6.5.3.2.2 SDS Annually Graphics

Gráfico de barras que muestra las SDS que han ocurrido anualmente en una concesionaria determinada.

En la figura 27 se muestra el gráfico resultante.

Figura 27: Gráfico anual por Concesionaria

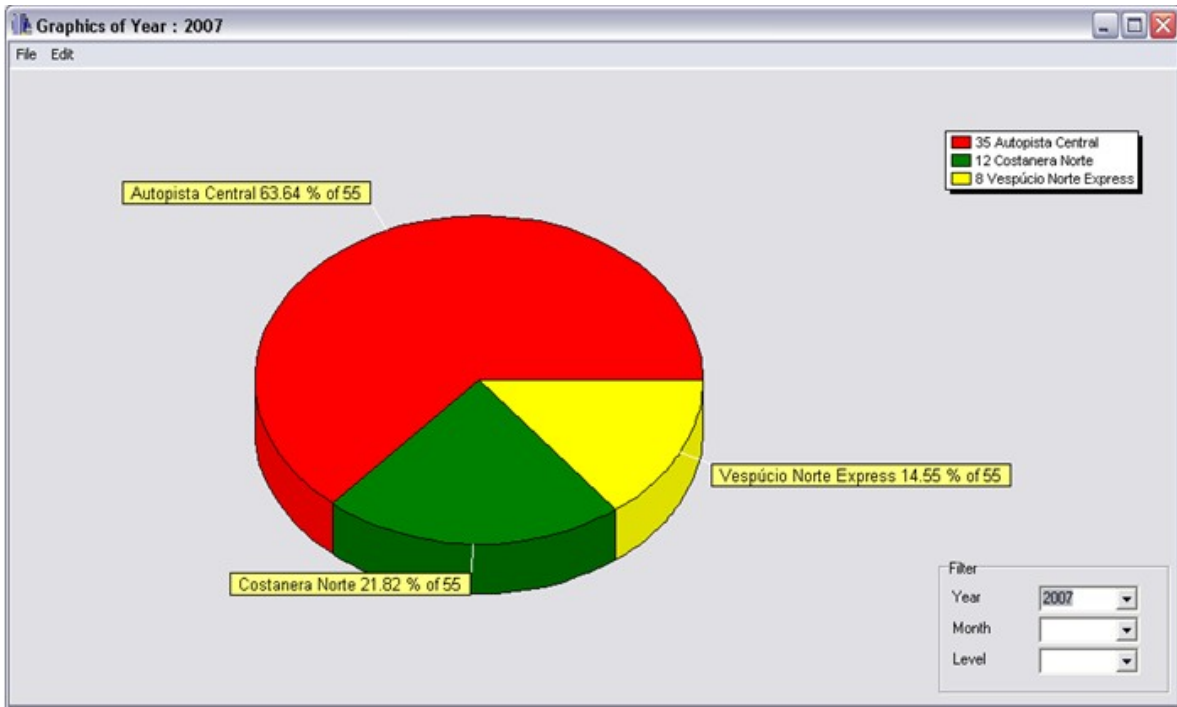


6.5.3.3 Graphics of Year

Gráfico de torta que muestra la cantidad de SDS ocurridas por cada uno de los proyectos en determinado año o mes. Este es el único gráfico que puede ser actualizado sin necesidad de abrir el formulario de reportes.

En la figura 28 se muestra el gráfico resultante.

Figura 28: Gráfico de torta

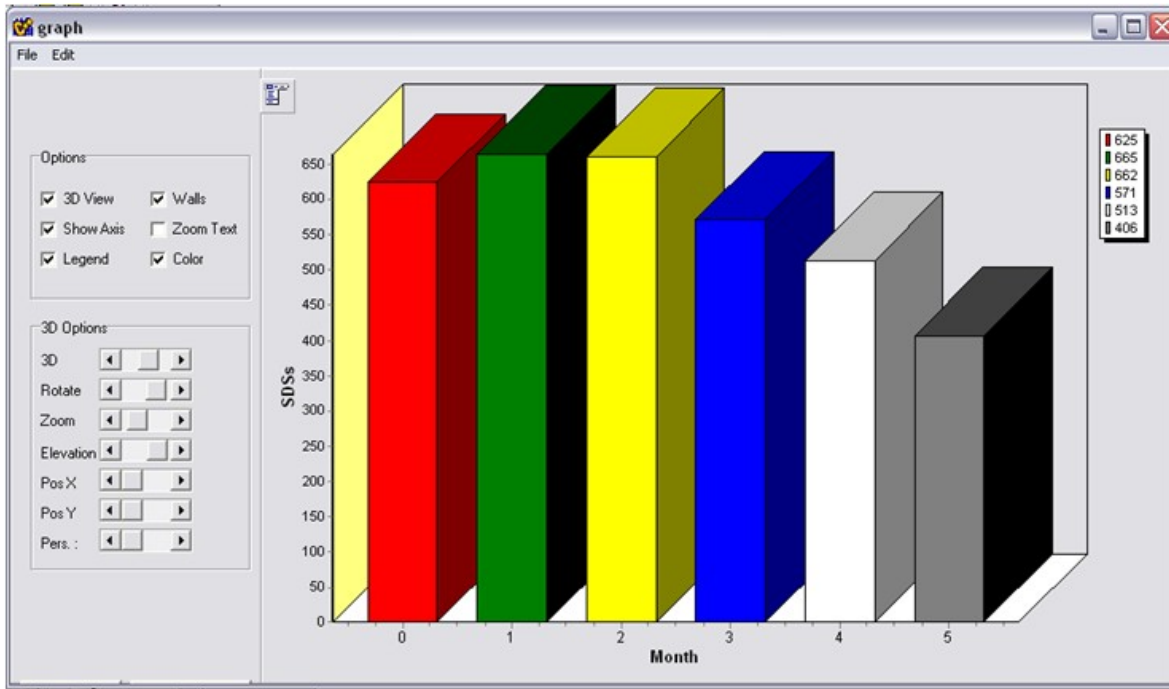


La programación de estos gráficos de barra fue realmente simple.

1) Se creó un formulario y se agregó el componente de gráficos.

La figura 29 muestra el componente.

Figura 29: Componente Gráfico



El componente de gráfico (derecha) por defecto muestra esos gráficos de barras con ese ejemplo, y las opciones del lado izquierdo son botones y opciones vinculados al gráfico.

3) Este componente de gráficos es muy simple de utilizar, se crearon funciones en cada uno de los reportes que muestran gráficos.

Un ejemplo.

La siguiente función lo que hace es utilizar los SCROLLBAR de la izquierda de la pantalla en diferentes opciones para el gráfico.

```
void __fastcall Tgraph::fill_box()
{
    Chart2->View3DOptions->Rotation = ScrollBar1->Position;
    Chart2->View3DOptions->Zoom = ScrollBar2->Position;
    Chart2->View3DOptions->Vert Offset = ScrollBar5->Position;
    Chart2->View3DOptions->Perspective = ScrollBar6->Position;
    show_box();
}
```

La siguiente función llena el formulario mostrado anteriormente.


```

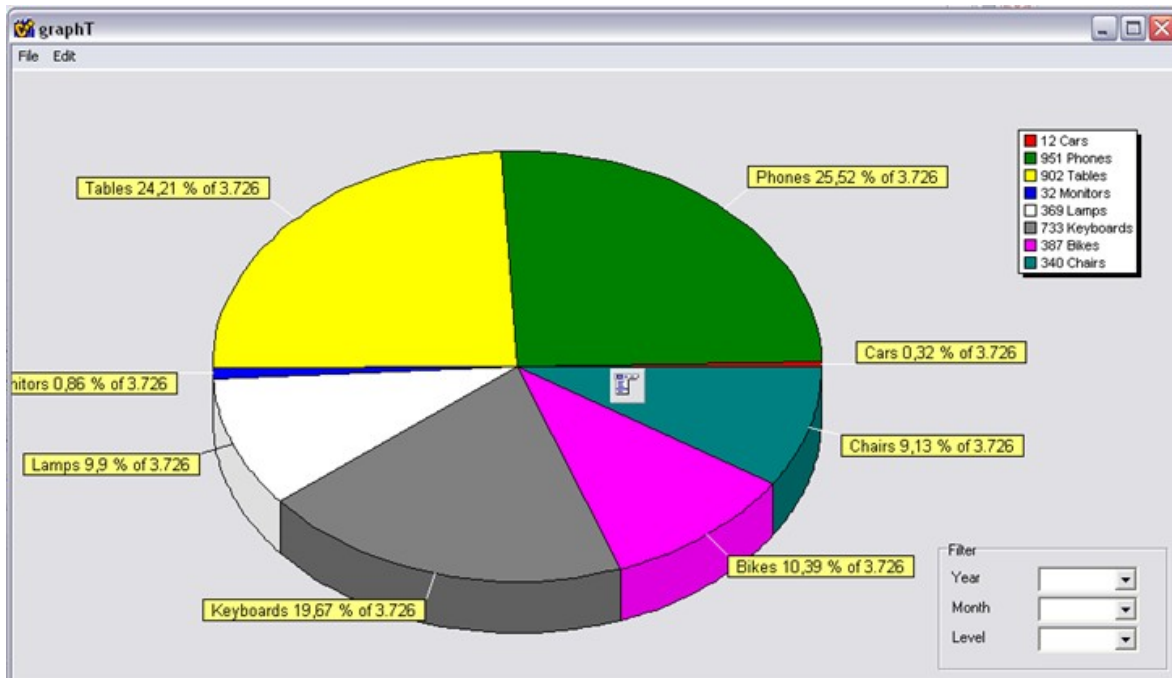
void __fastcall TReports::fill_graf_anual(String ano, String mes2, String proj, String lev, String st, int sem)
{
    int mensual[20], mes1, anual[20], ano1;
    String Fecha;
    String sql = "Select * from project, SDS, Times, dates, report, stateadm, States where sds.number = dates.number
and sds.number = times.number and sds.number = report.number and stateadm.number = sds.number and project.id =
sds.project and stateadm.activo = 'Yes' order by SDS.number"; // get the list of the Reports Type
    arr = MenuSDS->Db->Execute (sql);
    for (int i = 0; i < 20; i++){mensual[i] = 0; anual[i] = 0;}
    for (int i = 1; i < arr->Count; i++)
    {
        Fecha = MenuSDS->Db->Str[arr]["call"][i];
        int Anno = year(Fecha.ToDouble());
        if ((IntToStr(Anno) == ano || ano == "") &&
            (MenuSDS->Db->Str[arr]["level"][i] == lev || lev == "") &&
            (MenuSDS->Db->Str[arr]["project"][i] == IntToStr(num_proj(proj)) || proj == "") &&
            (MenuSDS->Db->Str[arr]["states.description"][i] == st || st == ""))
        {
            mes1 = month(Fecha.ToDouble());
            mensual[mes1] = mensual[mes1] + 1;
            ano1 = year(Fecha.ToDouble());
            anual[ano1 - ano1] = anual[ano1 - ano1] + 1;
        }
    }
    graph->Series1->Clear();
    if (sem == 0)
    {
        for (int i = 1; i <= 12; i++)
        {
            int a = mensual[i];
            AnsiString aux = mes(i);
            graph->Series1->Title = ano;
            graph->Series1->Add(a, aux, clTeeColor);
        }
    }
    else
    {
        for (int i = 0; i <= 20; i++)
        {
            int a = anual[i];
            AnsiString aux = IntToStr(ano1 + i);
            if (a > 0)
                graph->Series1->Add(a, aux, clTeeColor);
        }
    }
}
}

```

El otro tipo de gráfico es el de torta. Este gráfico está hecho en otro formulario diferente al de barras, ya que muestra un distinto tipo de información.

La figura número 30 muestra el formulario para gráfico de tortas.

Figura 30: Gráfico de Tortas



En el costado derecho de la pantalla se muestra un filtro por el cual el gráfico es modificado. En este se muestran los años, meses y el nivel de las SDS que se desea filtrar.

6.6 Testing y Prueba

El diseño de pruebas para el software o para otros productos de ingeniería puede requerir tanto esfuerzo como el propio diseño inicial del producto. Sin embargo, los ingenieros del software, por razones que se han tratado, a menudo tratan las pruebas como algo sin importancia, desarrollando casos de prueba que parezcan adecuados, pero que tienen poca garantía de ser completos. Recordando el objetivo de las pruebas, se debe diseñar pruebas que tengan la mayor probabilidad de encontrar el mayor número de errores con la mínima cantidad de esfuerzo y tiempo posible.

Cualquier producto de ingeniería (y de muchos otros campos) puede probarse de una de estas dos formas: conociendo la función específica para la que fue diseñado el producto, se pueden llevar a cabo pruebas que demuestren que cada función es completamente operativa y, al mismo tiempo, buscando errores en cada función; conociendo el funcionamiento del producto, se pueden desarrollar pruebas que aseguren que todas las piezas encajan, o sea, que la operación interna se ajusta a las especificaciones y que todos los componentes

internos se han comprobado de forma adecuada. El primer enfoque de prueba se denomina prueba de caja negra y el segundo, prueba de caja blanca.

Cuando se considera el software de computador, la prueba de caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. O sea, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa (por ejemplo, archivos de datos) se mantiene. Una prueba de caja negra examina algunos aspectos del modelo fundamental del sistema sin tener mucho en cuenta la estructura lógica interna del software. La prueba de caja blanca del software se basa en el minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del software proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones y/o bucles. Se puede examinar el “estado del programa” en varios puntos para determinar si el estado real coincide con el esperado o mencionado.

A primera vista parecería que una prueba de caja blanca muy profunda lleva a tener programas cien por cien correctos. Todo lo que se debe hacer, es definir todos los caminos lógicos, desarrollar casos de prueba que los ejerciten y evaluar los resultados, es decir, generar casos de prueba que ejerciten exhaustivamente la lógica del programa.

Desgraciadamente, la prueba exhaustiva presenta ciertos problemas logísticos. Incluso para pequeños programas, el número de caminos lógicos posibles puede ser enorme. Por ejemplo, considere un programa de 100 líneas de código en lenguaje C. Después de la declaración de algunos datos básicos, el programa contiene dos bucles que se ejecutan de 1 a 20 veces cada uno, dependiendo de las condiciones especificadas en la entrada. Dentro del bucle interior, se necesitan cuatro construcciones IF THEN ELSE.

¡Existen aproximadamente 10 caminos posibles que se pueden ejecutar en este programa!

Las pruebas son un conjunto de actividades que se pueden planificar por adelantado y llevar a cabo sistemáticamente.

Por esta razón, se puede definir en el proceso de la ingeniería del software una planilla para las pruebas del software: un conjunto de pasos en los que se pueden situar los métodos

específicos de diseño de casos de prueba. Se han propuesto varias estrategias de prueba del software en una gran variedad de libros diferentes. Todas proporcionan al ingeniero del software una plantilla para la prueba y todas tienen las siguientes características generales: Las pruebas comienzan a nivel de módulo y trabajan “hacia fuera”, hacia la integración de todo el sistema basado en computadores. Según el momento, son apropiadas diferentes técnicas de prueba. La prueba la lleva a cabo el responsable del desarrollo del software y (para grandes proyectos) un grupo independiente de pruebas. La prueba y la depuración son actividades diferentes, pero la depuración se debe incluir en cualquier estrategia de prueba.

Una estrategia de prueba del software debe incluir pruebas de bajo nivel que verifiquen que todos los pequeños segmentos de código fuente se han implementado correctamente, así como pruebas de alto nivel que validen las principales funciones del sistema frente a los requisitos del cliente. Una estrategia debe proporcionar una guía al profesional y proporcionar un conjunto de hitos para el jefe de proyecto. Debido a que los pasos de la estrategia de prueba se dan a la vez cuando aumenta la presión de los plazos fijados, se debe poder medir el progreso y los problemas deben aparecer lo antes posible.

6.6.1 Tipos de Pruebas

A continuación se describirá conceptualmente los diferentes tipos de prueba que existen.

6.6.1.1 Pruebas de Caja Blanca

Las Pruebas de caja blanca o caja de cristal, es una forma de realizar casos de prueba basada en la estructura de control del diseño procedimental. Mediante este método el Ingeniero del software obtiene los casos que.

- Garantice la ejecución de al menos una vez, de todos los caminos independientes de cada módulo.
- Ejercite todas las decisiones lógicas en sus vertientes verdadera y falsa.
- Ejecute los bucles en sus límites.
- Ejerciten las estructuras internas de datos para asegurar su validez.

Llegar a automatizar este tipo de pruebas es casi imposible, dado que para hacerlo se necesita pleno conocimiento de la estructura interna del software. Por lo que su estudio no se profundizará en mayor medida.

6.6.1.2 Pruebas de Caja Negra

Las pruebas de caja negra, también llamadas pruebas funcionales, ven el sistema como una caja donde su comportamiento sólo se puede determinar estudiando las entradas y las salidas relacionadas.

Las pruebas de caja negra intentan encontrar errores en las siguientes categorías.

1. Funciones incorrectas o ausentes
2. Errores de interfaz
3. Errores de estructuras de datos
4. Errores de rendimiento
5. Errores de inicialización y de terminación.

A diferencia de las pruebas de caja blanca, las cuales se utilizan generalmente en la etapa de prueba. Las pruebas de caja negra se suelen llevar en una etapa posterior a esta fase de pruebas. Ya que se ignora intencionadamente la estructura de control y centra su atención en la información.

El principal problema para el ingeniero de pruebas es la selección de las entradas que tengan una mayor probabilidad de pertenecer al conjunto que provoca un comportamiento anómalo en el software. Es posible utilizar conocimiento del dominio para identificar casos de prueba que probablemente revelen la presencia de algún defecto. De otra forma también es posible utilizar procedimientos sistemáticos para la selección de datos de prueba.

6.6.1.3 Bancos de trabajo de pruebas

Hoy en día existen herramientas que permiten manejar un arsenal de recursos para el desarrollo de pruebas, y su uso ayuda a disminuir en gran manera el costo del proceso de pruebas. Estas herramientas de prueba están integradas a bancos de trabajo de prueba.

Las herramientas incluidas en un banco de pruebas son las siguientes.

1. Administrador de pruebas: Controla la ejecución de las pruebas del software, lleva un registro de los datos de prueba, los resultados esperados y los recursos del programa probados.
2. Generador de datos de prueba: Genera datos de prueba para el programa a ser testeado o probado. Se realiza seleccionando datos de la base de datos o utilizando patrones para generar datos aleatorios de manera correcta.
3. Predictor: Genera predicciones de los resultados de prueba esperados, los predictores son versiones previas o prototipos del programa. La ejecución de pruebas paralelas de dos códigos es la técnica en la cual se realizan la ejecución en paralelo del predictor con el programa a ser probado. Luego se resaltan las diferencias entre ellos.
4. Comparador de archivos: Compara los resultados de las pruebas del programa con resultados de las pruebas anteriores y reporta las diferencias entre ellos. Los comparadores son esenciales en pruebas de regresión donde se comparan los resultados de las ejecuciones de programas antiguos versus nuevos. Las diferencias muestran los potenciales problemas en las nuevas versiones del software.
5. Generador de informes: Provee la definición del informe y los recursos de generación para los resultados de la prueba.
6. Analizador dinámico: Agrega código a un programa para contar el número de veces que se ejecuta cada instrucción. Luego de realizar las pruebas, genera un perfil de ejecución que muestra que tan seguido se ejecutan las instrucciones del programa.

7. Simulador: Provee de diferentes tipos de simuladores. Los simuladores Objetivo simulan la máquina sobre la cual se ejecuta el programa. Los de Interfaces de usuario, son programas guiados por secuencias de comandos que simulan interacciones múltiples de usuarios de manera simultáneas. Utilizar simuladores de E/S quiere decir que el tiempo de la secuencia de transacciones se repite.

La creación de un herramienta de banco de pruebas esta íntimamente ligada a la aplicación que se desea testear. Y se tiene que adaptar para satisfacer el plan de pruebas de cada sistema.

La cantidad de tiempo y esfuerzo de crear un banco de pruebas adecuado es considerablemente importante. Por lo que generalmente los bancos de prueba completos, sólo se desarrollan cuando los sistemas a testear son grandes y vale la pena la inversión en herramientas CASE de alta calidad que apoyen en el proceso de pruebas.

6.6.2 Casos de Prueba aplicados al proyecto

En los casos de prueba se tiene como objetivo principal la comprobación del software, que el software cumpla con todos los requerimientos de seguridad y estabilidad que se requieren en este tipo de productos.

Como primer paso, para hacer un buen testeo del software, se realizó una buena inspección visual manual del código. Esta visión demostró que el código es legible y no posee errores a simple vista.

Como no existe la posibilidad de practicar pruebas muy exhaustivas, se realizaron pruebas manuales con personas de la empresa, y en ciertas partes se utilizó pequeños programas para testear el software.

6.6.2.1 Pruebas de Caja Blanca

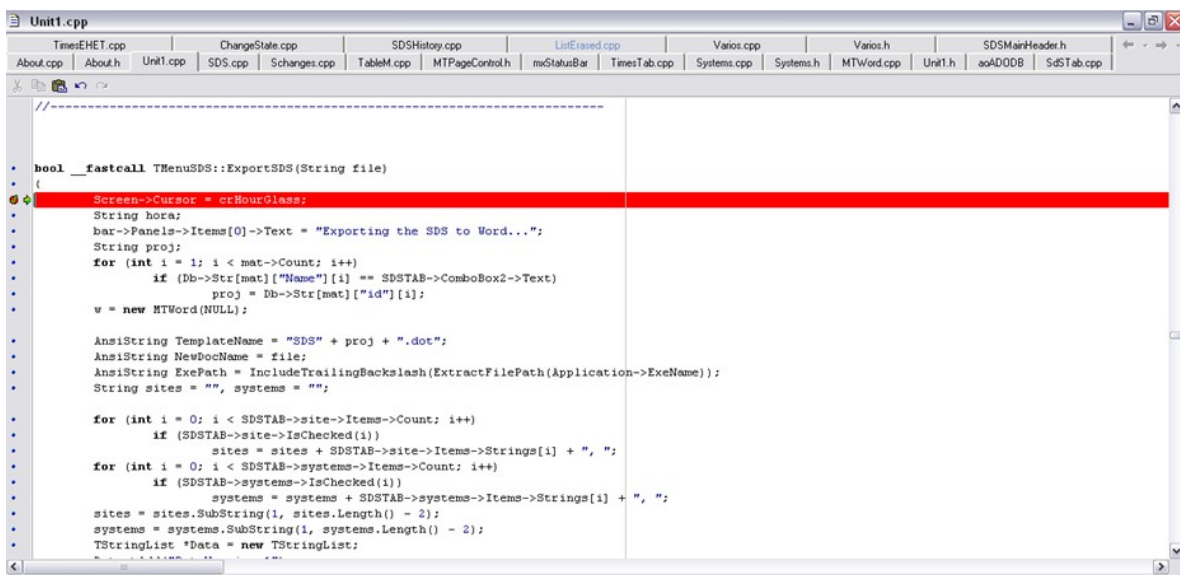
Las pruebas de caja blanca que se realizaron en este proyecto abarcaron gran parte del tiempo, ya que se fue revisando el código formulario a formulario, verificando que todas las estructuras y opciones que se utilizan están en forma correcta.

6.6.2.1.1 Cobertura de Seguimientos

Se comenzó las pruebas de caja blanca con una fase llamada “Cobertura de Segmentos”, en la cual se ejecutaron línea a línea de código con el DEBUG que dispone el Borland C++ Builder 5.0. Esta prueba realmente es muy lenta, pero no fue realizada al 100% del código, ya que el Software posee unas 2000 a 3000 líneas de código. Se reviso prácticamente el 60% del código y tardó aproximadamente una semana y media.

En la figura 31 se muestra una imagen de Cobertura de Seguimientos.

Figura 31: Cobertura de Seguimientos



The image shows a screenshot of a code editor window titled 'Unit1.cpp'. The editor displays C++ code for a function named 'ExportSDS'. A red horizontal bar highlights the line: `Screen->Cursor = crHourGlass;`. The code includes various declarations and loops for processing data from a database and updating a menu. The editor's interface includes a menu bar at the top with options like 'File', 'Edit', 'View', 'Tools', 'Window', and 'Help', and a toolbar below it. The code is displayed in a monospaced font with syntax highlighting.

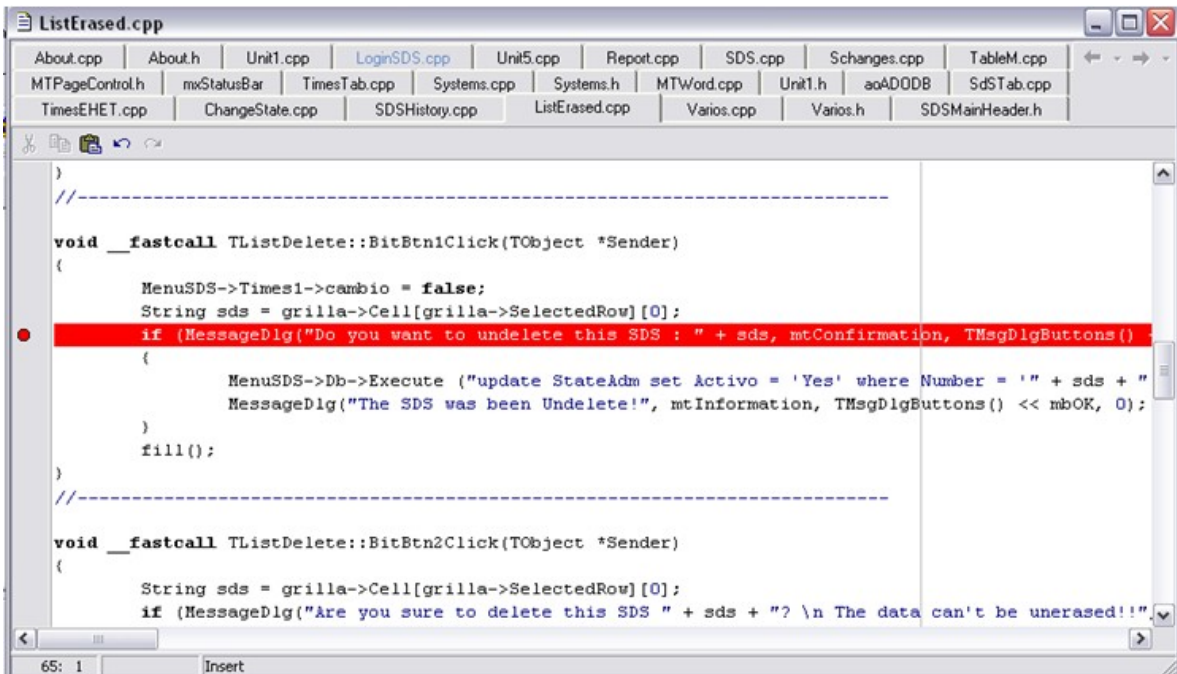
6.6.2.2 Cobertura en Ramas

Posteriormente se continuó con una cobertura llamada “Cobertura de Ramas” en la cual se prueban cada uno de las condiciones que se emplean en la realización del Código.

Se realizó un 70 % de la revisión de las condiciones que se utilizaron en el sistema, en todas las funciones IF.

La figura 32 muestra una imagen de la etapa de Cobertura en Ramas.

Figura 32: Cobertura en Ramas



```
void __fastcall TListDelete::BitBtn1Click(TObject *Sender)
{
    MenuSDS->Times1->cambio = false;
    String sds = grilla->Cell[grilla->SelectedRow][0];
    if (MessageDlg("Do you want to undelete this SDS : " + sds, mtConfirmation, TMsgDlgButtons()
    {
        MenuSDS->Db->Execute ("update Stateadm set Activo = 'Yes' where Number = '" + sds + "
        MessageDlg("The SDS was been Undelete!", mtInformation, TMsgDlgButtons() << mbOK, 0);
    }
    fill();
}

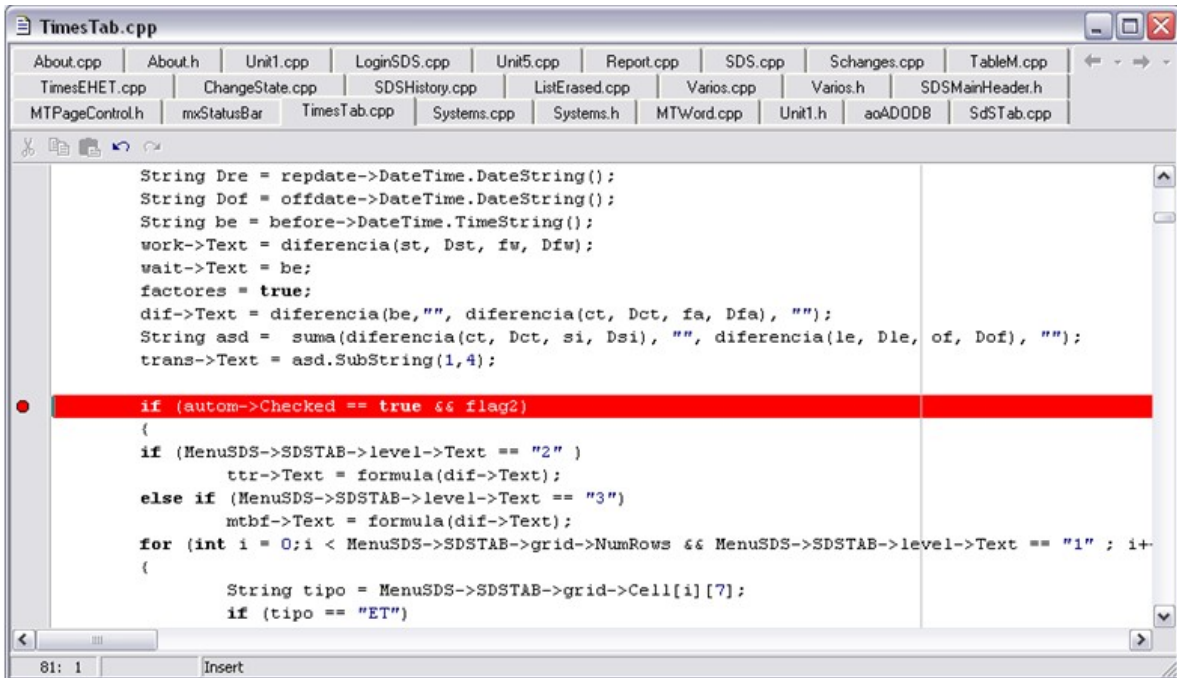
void __fastcall TListDelete::BitBtn2Click(TObject *Sender)
{
    String sds = grilla->Cell[grilla->SelectedRow][0];
    if (MessageDlg("Are you sure to delete this SDS " + sds + "? \n The data can't be unerased!!",
```

6.6.2.3 Coberturas de Condición y Decisión

Posteriormente de la cobertura de ramas, se continuó con las “Coberturas de Condición y Decisión”, estas son más complejas que las anteriores ya que de dos condiciones diferentes se pueden hacer hasta 4 caminos diferentes. Esto funciona si existen n condiciones, los tipos de caminos pueden ser hasta 2^n .

La figura 33 muestra una imagen de Cobertura de Condición y Decisión.

Figura 33: Cobertura de Condición y Decisión



```
String Dre = repdate->DateTime.DateString();
String Dof = offdate->DateTime.DateString();
String be = before->DateTime.TimeString();
work->Text = diferencia(st, Dst, fw, Dfw);
wait->Text = be;
factores = true;
dif->Text = diferencia(be, "", diferencia(ct, Dct, fa, Dfa), "");
String asd = suma(diferencia(ct, Dct, si, Dsi), "", diferencia(1e, D1e, of, Dof), "");
trans->Text = asd.SubString(1,4);

if (autom->Checked == true && flag2)
{
    if (MenuSDS->SDSTAB->level->Text == "2" )
        ttr->Text = formula(dif->Text);
    else if (MenuSDS->SDSTAB->level->Text == "3")
        mtbf->Text = formula(dif->Text);
    for (int i = 0; i < MenuSDS->SDSTAB->grid->NumRows && MenuSDS->SDSTAB->level->Text == "1" ; i++)
    {
        String tipo = MenuSDS->SDSTAB->grid->Cell[i][7];
        if (tipo == "ET")
```

Este tipo de cobertura tomó mucho tiempo en ser realizada.

6.6.2.4 Cobertura en Bucles

Posteriormente a las pruebas de cobertura/decisión, se continuó con las pruebas de “Cobertura de Bucles”. Estas pruebas se realizaron a más del 90 % de los FOR que tiene el programa. Este tipo de pruebas tomaron mucho tiempo en ser realizadas.

La figura 34 muestra una imagen de Cobertura en Bucles.

Figura 34: Cobertura en Bucles

```
TimesTab.cpp
About.cpp | About.h | Unit1.cpp | LoginSDS.cpp | Unit5.cpp | Report.cpp | SDS.cpp | Schanges.cpp | TableM.cpp
TimesEHET.cpp | ChangeState.cpp | SDSHistory.cpp | ListErased.cpp | Varios.cpp | Varios.h | SDSMainHeader.h
MTPageControl.h | mxStatusBar | TimesTab.cpp | Systems.cpp | Systems.h | MTWord.cpp | Unit1.h | aoADODB | SdSTab.cpp

dif->Text = diferencia(be,"", diferencia(ct, Dct, fa, Dfa), "");
String asd = suma(diferencia(ct, Dct, si, Dsi), "", diferencia(le, Dle, of, Dof), "");
trans->Text = asd.SubString(1,4);

if (autom->Checked == true && flag2)
{
if (MenuSDS->SDSTAB->level->Text == "2" )
    ttr->Text = formula(dif->Text);
else if (MenuSDS->SDSTAB->level->Text == "3")
    mtbf->Text = formula(dif->Text);
for (int i = 0; i < MenuSDS->SDSTAB->grid->NumRows && MenuSDS->SDSTAB->level->Text == "1" ; i+
{
    String tipo = MenuSDS->SDSTAB->grid->Cell[i][7];
    if (tipo == "ET")
        et->Text = formula(dif->Text);
    else if (tipo == "EH and ET")
    {
        et->Text = formula(dif->Text);
        eh->Text = formula(dif->Text);
    }
}
}
87: 1 | Insert
```

6.6.3 Pruebas de Caja Negra

Las pruebas de caja negra corresponden a todas las pruebas que se realizan al ejecutar el sistema. Se realizaron muchos tipos de pruebas de caja negra.

6.6.3.1 Bases de Datos

El primer tipo de prueba que se realizó fueron las de eliminar o modificar los caracteres no válidos para la inserción de información dentro de una Base de Datos. Estos caracteres son los apóstrofes y las comillas. En un primer momento el programa soportaba este tipo de caracteres y al momento de insertar o actualizar en la base de datos producía error. El apostrofe fue reemplazado por un tilde y las comillas fueron eliminadas al ser detectadas dentro de un cuadro de texto.

Otro tipo de prueba que se realizó referente a las Bases de Datos fue la utilización de palabras reservadas en las sentencias SQL. Estas fueron modificadas una vez que terminó el tiempo de testing.

En la figura 35 se aprecia la función “Clean String”, la cual es utilizada y llamada cada vez que es utilizado un cuadro de texto.

Figura 35: Función Clean String

```
String Clean_String(String cln)
{
    for (int i = 1; i < cln.Length() + 1; i++)
    {
        if (cln.SubString(i, 1) == "")
            cln = cln.SubString(1, i-1) + "" + cln.SubString(i + 1, cln.Length());
        if (cln.SubString(i, 1) == "\")
            cln = cln.SubString(1, i-1) + "" + cln.SubString(i + 1, cln.Length());
    }
    return cln;
}
```

6.6.3.2 ComboBox

La utilización de ComboBox (listado de opciones tratado como cuadro de texto) tiene sus ventajas y sus desventajas. Es cierto que es muy rápido de acceder a ciertas palabras que son utilizadas, pero puede producir error en caso de que se escriba sobre el ComboBox algo que no corresponda con el listado del mismo. Por eso al probar los ComboBox fue necesario crear una función que revisara si la información de texto contenida en este corresponde a lo de su listado.

En la figura 36 se aprecia la función “Revisa Combo” la cual es utilizada y llamada cada vez que es utilizado un ComboBox.

Figura 36: ComboBox

```
bool revisa_combo(TComboBox *combo, String st)
{
    if (st == "")
        return true;
    for (int i = 0; i < combo->Items->Count; i++)
        if (combo->Items->Strings[i] == st)
            return true;
    return false;
}
```

6.6.3.3 Fechas

El problema que existió con las fechas fue que cada una de las concesionarias tiene una diferente forma de usar las fechas, y el SDS Manager tuvo que adaptarse a cada una de las formas de usarlas.

Para eso se tuvo cambiar todas las fechas del sistema y las bases de datos a formato de número Double. Se tuvo que crear muchos tipos de funciones para rescatar separadamente los días, meses y años.

En la figura 37 se aprecian las funciones que se utilizan para separar las fechas.

Figura 37: Funciones de Fechas

```


int day(String date)
{
    TDateTime aux = StrToDate(date);
    Word Year, Month, Day;
    DecodeDate(aux, Year, Month, Day);
    return (int)Day;
}
int month(String date)
{
    TDateTime aux;
    aux = StrToDate(date);
    Word Year, Month, Day;
    DecodeDate(aux, Year, Month, Day);
    return (int)Month;
}
int year(String date)
{
    TDateTime aux;
    aux = StrToDate(date);
    Word Year, Month, Day;
    DecodeDate(aux, Year, Month, Day);
    return (int)Year;
}
double DateInv(int dia, int mes, int ano)
{
    TDateTime dt = EncodeDate((Word)ano, (Word)mes, (Word)dia);
    return dt;
}

```

6.6.3.4 Importación y Exportación de Documentos Word

En la importación y exportación de documentos Word hubo muchos problemas ya que las tres concesionarias no tienen el mismo formato de documento Word de SDS, y hubo que adecuarse a cada una de ellas. Como fue mencionado anteriormente las fechas de las concesionarias no tienen el mismo formato, y hubo que hacer una forma especial para cada una de ellas.

Las figuras 38 y 39 muestran una SDS en formato Word antes y después de ser atendida vía SDS Manager.



SOLICITUD DE MANTENIMIENTO

Reporte

Correlativo	722	Fecha y hora de Reporte	16/04/2007 18:20
Lugar de la Ejecución	AC-18	Tipo Reporte	Correctivo
Equipo o Sistema Afectado	VDC	Nivel de Severidad	Nivel 3
Descripción de Incidencia	Alarma 3715	Fecha y hora de Incidencia	16/04/2007 12:58
Estado de SdS	Abierta		

Reporto la Incidencia

Nombre	Juan Carlos Carvajal	Firma	~\J^_JB\jw\yYGT\SWBT~*00000
Comunicado a	Pablo Rocuant	Empresa	Kapsch

Acciones de Mantenimiento

Aviso Comienzo Trabajos	
Chequeo Seguridad	
Fecha y hora de Inicio	
Actividades:	
Impacto en el Sistema	
Documentos de Apoyo	
Pruebas Realizadas	
Resultado de las Pruebas	
Materiales c/cant.	
Equipos	
Instrumentos de Medición:	
Línea base antigua	

Figura 38: SDS antes de ser atendida



SOLICITUD DE MANTENIMIENTO

Reporte

Correlativo	722	Fecha y hora de Reporte	16/04/2007 18:20
Lugar de la Ejecución	AC-18	Tipo Reporte	Correctivo
Equipo o Sistema Afectado	VDC	Nivel de Severidad	Nivel 3
Descripción de Incidencia	Alarma 3715	Fecha y hora de Incidencia	16/04/2007 12:58
Estado de SdS	Abierta		

Reporto la Incidencia

Nombre	Juan Carlos Carvajal	Firma	~\U^_JB\jwYyYGTVSWBT~*00000 0-0
Comunicado a	Pablo Rocuant	Empresa	Kapsch

Acciones de Mantenimiento

Aviso Comienzo Trabajos	Operador de Turno
Chequeo Seguridad	Luis Torres
Fecha y hora de Inicio	17/04/2007 11:30
Actividades:	Verificación de alarma a través de LCTU Realineamiento de sensor VDC1. La imagen sobre la consola muestra líneas curvas verticales sobre las cámaras VDC Left de sensores 1 y 2 (condición anormal) Luego del realineamiento, se levantan alarmas 3000 VDC Unreliable y 3802 SIB 1 BIT error. Se reemplaza VDC Controller, cancelando alarmas.
Impacto en el Sistema	Sin EH durante la atención de la alarma
Documentos de Apoyo	8633 801 063b
Pruebas Realizadas	Verificación de Imágenes Superiores Verificación de Contadores

Figura 39: SDS después de ser atendida

CAPITULO 7: Conclusiones

Al finalizar el presente, se debe destacar el importante rol que desempeñó el desarrollador del sistema ayudado de cada una de las herramientas que entregan los diferentes paradigmas de desarrollo de análisis y diseño. En el caso particular de la metodología orientada a objetos, junto con UML, los cuales ofrecen estándares mundiales para el desarrollo de las distintas etapas de un proyecto informático. Es así como los diagramas de casos de uso permiten establecer una mejor comunicación y un mejor entendimiento de los requerimientos establecidos por el cliente.

En la actualidad el sistema SDS Manager está siendo utilizado como una herramienta esencial de trabajo para el manejo de las solicitudes de servicio de cada una de las concesionarias con las cuales se mantiene contrato de servicio. También es utilizado para realizar los informes mensuales de tiempos de respuesta y listados de solicitudes, los cuales se entregan a los clientes.

Concluyendo, se ha visto ante el desafío de desarrollar un sistema sobre una plataforma de programación que es relativamente antigua. Al enfrentarse a esta problemática, se puede concluir que existen diversos manuales, y que el uso de Internet ha sido una herramienta fundamental para la programación de ciertos módulos del software.

CAPITULO 8: Glosario y Referencias

8.1 Glosario

Término	Definición
ALC	Equipo de cabina técnica encargado de gestionar el paso de los vehículos por los pórticos. El ALC es el encargado de automatizar todo el sistema de cobro de peaje.
Autopista Central (ANS)	Autopista urbana concesionada de la ciudad de Santiago de Chile, que atraviesa dicha ciudad de norte a sur. Esta autopista, es cliente de la empresa Kapsch TrafficCom Chile S.A.
Combitech	Nombre de la empresa Kapsch TrafficCom ante de ser adquirido por el grupo Austriaco Kapsch Group.
Costanera Norte (CN)	Autopista urbana concesionada de la ciudad de Santiago de Chile, que atraviesa dicha ciudad de oriente a poniente, ubicada en la ribera y bajo el río Mapocho. Esta autopista es cliente de la empresa Kapsch TrafficCom Chile S.A.
Free Flow	Tipo de peajes en el cual el vehiculo no debe detener su marcha para que se realice el cobro. Este tipo de peajes es efectivo hasta los 160 Kilómetros por hora.
Kapsch TrafficCom A.B.	Corresponde a la sucursal Sueca del grupo Kapsch Group (Kapsch AG).

KTSC (Kapsch TrafficCom Chile)	Corresponde a la sucursal chilena del grupo Kapsch Group (Kapsch AG).
SAAB	Empresa estatal Sueca, dedicada a la fabricación de vehículos de alta gama y aviones de guerra.
SAR	Servicio de mantenimiento de equipos que se envían a Suecia para su reparación.
SDS	Una Solicitud de Servicio corresponde a un evento de anomalía o de configuración defectuosa en alguno de los sistemas de los pódicos de los clientes hacia la empresa Kapsch TrafficCom Chile S.A.
TAG	Dispositivo que se encuentra ubicado en los parabrisas de los vehículos que contienen la información del tipo e identificación única del vehículo que esta pasando bajo un pódico. Conocido también como Televía y Transponder.
UnitTracer	Aplicación creada por Kapsch TrafficCom Chile para el control de inventario. Creada en Borland C++ 5.0 con base de datos Access.
Vespucio Norte Express (VNE)	Autopista urbana concesionada de la ciudad de Santiago de Chile, que atraviesa dicha ciudad de oriente a poniente, ubicada al norte de la ciudad de Santiago.
VDC	Sensor ubicado en lo mas alto del pódico que detecta cuando un vehículo se acerca al pódico y que tipo de TAG es el que debería tener.
VR	Cámara que registra la fotografía frontal de los vehículos. Posteriormente esta fotografía es analizada por un OCR para obtener la placa patente.

8.2 Referencias

[1] Ian Sommerville, Ingeniería de Software, Edición 2003.

[2] Roger S. Pressman, Ingeniería de Software, un enfoque práctico, Quinta Edición.

[3] Graig Larman, Uml y Patrones. Quinta edición.

[4] Bob Swart, Mark Cashman, Borland C++ Builder 6 Developer's Guide, Primera Edición.

ANEXO A

Tabla de índices mensuales MTBF

	ENERO	FEBRERO	MARZO	ABRIL	MAYO	JUNIO	Promedio Semestre	JULIO
Calendar time Mes	744	672	744	720	744	720	724	744
N° Tolling Points	12	12	12	16	16	16	14	16
ΣDT (for each device)	1.67	0	2.01	0	2.00	5.83	1.92	4.37
Disponibilidad ET	99.98%	100.00%	99.98%	100.00%	99.98%	99.95%	99.98%	99.96%
ΣDT (for each device)	2.50	10.90	58.93	15.18	11.97	10.95	18.40	26.27
Disponibilidad EH	99.97%	99.86%	99.34%	99.87%	99.90%	99.90%	99.81%	99.78%
Σ TTR (for each Sds)	23.13	7.39	6.93	16.58	28.10	40.58	20.45	6.13
N° Sds	5	2	2	7	9	15	6.67	3
MTTR	4.63	3.70	3.47	2.37	3.12	2.71	3.33	2.04
Disp. ET Contractual								99.9%
Disp. EH Contractual								99.6%
MTTR Contractual								

ANEXO B

Formato Archivo Word SDS

Correlativo	001	Fecha de Reporte	25/7/04

Lugar de la Ejecución	AC04	Hora de Reporte	18:00
Equipo o Sistema Afectado	TSMC		
Descripción de solicitud	Limpieza de bases de datos y colas de mensajería		

Solicita el Servicio

Nombre	Carlos López	Firma	

Acciones

Aviso Comienzo Trabajos	Carlos Cisneros / CPA	Chequeo Seguridad	NA
Fecha de Inicio	26/7/2004	Hora Inicio	15:30
Actividades:	Limpieza de base de datos del TSMC		
Impacto en el Sistema			
Documentos de Apoyo	KTSC-04-302 rev A		
Materiales c/cant.		Equipos	LCTU

Instrumentos de Medición:			
Condiciones de Operación	ALCs y TSMC en Test & Installation mode		
Observaciones			
Fecha Término	26/7/2004	Hora Término	17:04
Aviso de término	Vía telefónica al sr. Carlos Paratori		

Responsable de Acciones

Nombre	Carlos Cisneros	Firma	

ANEXO C

Listado de SDS

Sitio	Observaciones	codigo alarma	CAT	Nivel	Status	SAR
AC-15	Acusa Alarma de Nivel 1. Id Hexadecimal : 10EF Card reader error.	10EF		1	closed	
AC-07	VR SU 4 del AC-07-s1 no responde después de corte de energía	4033		1	closed	
AC-13	Alarma 2100 - EH Failure Alarm	2100		1	closed	
AC-03	Alarma 4264 - Too many failed (subsequent) OCRs have occurred. Ill	4264		1	closed	
AC-16	Alarma 3200 - VR communication lost	2200-3200		1	closed	4
AC-08	VR SU communication lost	2200-3200		1	closed	4
AC-09	VR communication lost	2200-3200		1	closed	4
AC-13	Alarmas reiteradas de OCR en el pórtico.	4064		1	closed	
AC-04	Communication with VDC controller is lost (2210h)	2210		1	closed	
AC-16	Communication with the SU 1 is lost.	4030		1	closed	
AC-16	Communication with the SU 1 is lost.	4030		1	closed	4
AC-08	Communication with the SU 1 is lost.	4030		1	closed	

ANEXO D

Reporte de cambios de Sistemas



Reporte de Cambios en Control de Configuración

Reporte

In / Out	Artículo confs	Serial number	Req	Item details	Installation Date
Out 1					
In 1					
Out 2					
In 2					
Out 3					
In 3					
Out 4					
In 4					
Out 5					
In 5					
Out 6					
In 6					
Out 7					
In 7					
Out 8					
In 8					
Out 9					