UNIVERSIDAD AUSTRAL DE CHILE
CAMPUS PUERTO MONTT
ESCUELA DE INGENIERIA EN COMPUTACION

"Análisis, Diseño, Construcción de un Sistema de Información de Mantenciones en Centros de Cultivos, bajo Arquitectura de Múltiples Capas"

Seminario de Titulación para optar al Título de Ingeniero en Computación

PROFESOR PATROCINANTE:
Sr. Óscar Salazar Cerna

PROFESOR COPATROCINANTE:
Sra. Claudia Zil Bontes

ERICK MARCOS NAYÁN BARRÍA

PUERTO MONTT - CHILE
2008

Universidad Austral de Chile
Escuela de Ingeniería en Computación

Los Pinos s/n, Balneario Pelluco
Campus Puerto Montt
Puerto Montt · Chile
Casilla 1327 · Fono: 56 65 260990
Fax: 56 65 277156
Email: ecomputa@uach.cl
www.uach.cl

Puerto Montt, 28 de Julio de 2008

COMUNICACIÓN INTERNA N° 156/08

DE : Sra. Sandra Ruiz Aguilar
DIRECTORA ESCUELA DE INGENIERIA EN COMPUTACION

A : Dr. Renato Westermeier H. – **DIRECTOR CAMPUS PUERTO MONTT**
Sra. Cristina Barriga – **REGISTRO ACADEMICO**
Sra. Alba Vásquez - **ENCARGADA DE TITULACIÓN CAMPUS PUERTO MONTT**

C.c : Sr. Erick Nayán Barría
Sr. Oscar Salazar Mapocho
Sra. Claudia Zil Bontes
Sr. Moisés Coronado Delgado

MOTIVO:

Informar a usted, las calificaciones obtenidas por el alumno de Ingeniería en Computación **Sr. Erick Marcos Nayán Barría** Rut 15.301.750-6, en su informe de Titulación "***Análisis, diseño, construcción de un sistema de información de mantención en Centros de Cultivos, bajo arquitectura de múltiples capas***"

Prof. Oscar Salazar Mapocho	6.5
Prof. Claudia Zil Bontes	6.0
Prof. Moisés Coronado Delgado	5.5
Promedio Seminario	**6.0**

Sin otro particular, le saluda atentamente,

UNIVERSIDAD AUSTRAL DE CHILE
ESCUELA DE INGENIERIA EN COMPUTACION
CAMPUS PUERTO MONTT

SANDRA RUIZ AGUILAR
DIRECTORA

SRA/mva

PUERTO MONTT, 23 de Julio de 2008

De : Sr. Oscar Salazar
PROFESOR PATROCINANTE

A : Sra. Sandra Ruiz Aguilar
DIRECTORA ESCUELA INGENIERÍA EN COMPUTACIÓN

MOTIVO:

Informar a Usted la calificación obtenida por el alumno ***Erick Marcos Nayán Barría*** en su Seminario de Titulación "**Análisis, Diseño, Construcción de un Sistema de Información de Mantenciones en Centros de Cultivos, Bajo Arquitectura de Múltiples Capas ":**

NOTA: 6,5

JUSTIFICACION:

Excelente trabajo realizado pero el documento demasiado extenso.

OTRAS OBSERVACIONES:

Algunos problemas en la redacción y falsó poder de síntesis. Estoy seguro que para demostrar que se es ingeniero no se requiere un documento tan extenso.

Sr. Oscar Salazar
PROFESOR PATROCINANTE

PUERTO MONTT, 18 de Julio del 2008

De : Sra. Claudia Zil Bontes
PROFESORA CO-PATROCIMANTE

A : Sra. Sandra Ruiz Aguilar
DIRECTORA ESCUELA INGENIERÍA EN COMPUTACIÓN

MOTIVO:

Informar a Usted la calificación obtenida por el alumno ***Erick Marcos Nayán Barría*** en su Seminario de Titulación "**Análisis, Diseño, Construcción de un Sistema de Información de Mantenciones en Centros de Cultivos, Bajo Arquitectura de Múltiples Capas "**:

NOTA: 6,0

JUSTIFICACION:

Sistema bien desarrollado y documentado.
Uso de características no comunes de las de clase.

OTRAS OBSERVACIONES:

Sra. Claudia Zil Bontes
PROFESORA CO-PATROCIMANTE

PUERTO MONTT, 23-Jul-2008

De : Sr. Moisés Coronado Delgado
PROFESOR INFORMANTE

A : Sra. Sandra Ruiz Aguilar
DIRECTORA ESCUELA INGENIERÍA EN COMPUTACIÓN

MOTIVO:

Informar a Usted la calificación obtenida por el alumno ***Erick Marcos Nayán Barría*** en su Seminario de Titulación "**Análisis, Diseño, Construcción de un Sistema de Información de Mantenciones en Centros de Cultivos, Bajo Arquitectura de Múltiples Capas ":**

NOTA: 5,5

JUSTIFICACION:

- Buena Presentación.
- Falto un poco mas de detalle

OTRAS OBSERVACIONES:

Sr. Moisés Coronado Delgado
PROFESOR INFORMANTE

Dedicado a Mis Padres Sonia y Luis

INDICE

1. Introducción ..... - 1 -
1.1 Resumen de capítulos ..... - 4 -
2 Objetivos ..... - 7 -
2.1 Objetivo General ..... - 7 -
2.2 Objetivos Específicos ..... - 7 -
3 Planteamiento del Problema ..... - 8 -
3.1 Antecedentes ..... - 8 -
3.1.1 Definición del Problema ..... - 8 -
3.1.2 Identificación de Esfuerzos Anteriores para Resolverlo ..... - 10 -
3.1.3 Solución Propuesta ..... - 10 -
3.1.4 Definición Equipo de trabajo ..... - 16 -
3.2 Justificación ..... - 17 -
3.2.1 Situación Sin Proyecto ..... - 17 -
3.2.2 Situación con Proyecto ..... - 18 -
3.3 Delimitación ..... - 19 -
4 Metodología ..... - 21 -
5 Recursos ..... - 23 -
5.1 Hardware ..... - 23 -
5.2 Software ..... - 24 -
5.2.1 Software para el desarrollo del proyecto ..... - 24 -
5.2.2 Software para el servidor de producción del proyecto ..... - 25 -
6 Análisis del Sistema ..... - 27 -
6.1 Descripción y Conceptos Generales del sistema ..... - 27 -
6.1.1 Módulo Activo ..... - 34 -
6.1.1.1 Caso de uso: Ingreso Nuevo Activo. ..... - 34 -
6.1.1.2 Caso de uso: Búsqueda Activo Oracle. ..... - 35 -
6.1.1.3 Caso de uso: Modificar Activo. ..... - 36 -
6.1.1.4 Caso de uso: Eliminación de Activo. ..... - 36 -
6.1.1.5 Caso de uso: Ingreso /Modificación/Eliminación Ítems de Activo - 37 -
6.1.1.6 Caso de uso: Ingreso /Modificación/Eliminación Recepciones Parciales de Activo. ..... - 38 -
6.1.1.7 Caso de uso: Actualización Código AF del Activo. ..... - 39 -
6.1.1.8 Caso de uso: Baja Activo. ..... - 40 -
6.1.1.9 Caso de uso: Registro Información Básica. ..... - 40 -
6.1.2 Módulo Mantenciones ..... - 43 -
6.1.2.1 Caso de uso: Ingreso Mantención Teórica. ..... - 43 -
6.1.2.2 Caso de uso: Creación Plan Mantención para Modelo. ..... - 44 -
6.1.2.3 Caso de uso: Ejecutar Plan de Mantención de un Activo. ..... - 45 -
6.1.2.4 Caso de uso: Ejecutar Mantenciones Efectivas de un Activo. ..... - 46 -

6.1.2.5 Caso de uso: Ejecutar Mantención NO planificada. - 47 -
6.1.2.6 Caso de uso: Registro Mantención Calendario Google. - 48 -
6.1.3 Módulo Reportes. - 49 -
6.1.3.1 Caso de uso: Generar reporte. - 49 -
6.1.4 Casos de Uso transversales para todos los módulos del Sistema. - 51 -
6.1.4.1 Caso de uso: Sincronizador Datos Central / Área. - 51 -
6.1.4.2 Caso de uso: Autenticación de Usuario. - 52 -
6.1.2 Análisis de Tecnologías a Utilizar y su Aplicación - 52 -
6.1.2.1 Modelo N-Capas - 52 -
6.1.2.2 Evolución del modelo de N-Capas a Webservices - 54 -
6.1.2.3 WebServices - 55 -
6.1.2.3.1Requisitos que debe cumplir un WebServices - 56 -
6.1.2.3.2 Elementos que componen un WebServices - 57 -
6.1.2.3.2.1 XML (eXtended Markup Languaje) - 57 -
6.1.2.3.2.2 SOAP - 59 -
6.1.2.3.2.3 WSDL (WebService Description Language) - 60 -
6.1.2.3.2.4 Disco - 61 -
6.1.2.3.2.5 UDDI - 61 -
6.1.2.3.2.6 SOAP, WSDL, UDDI, y WebServices - 61 -
6.1.2.3.2.7 Posibles riesgos de los WebServices - 62 -
6.1.2.3.2.8 Calidad de los WebServices - 63 -
6.1.2.3.2.9 Estandarización de los WebServices - 63 -
6.1.2.4 Reporting Services - 64 -
6.1.2.5 Réplica de Base de Datos - 66 -
6.1.2.5.1Réplica en SQL Server 2005. - 67 -
6.2 Diseño del Sistema - 71 -
6.2.1 Diagramas de Actividad - 72 -
6.2.1.1 Diagrama de actividad: ingreso información básica. - 72 -
6.2.1.2 Diagrama de actividad: búsqueda activo en Oracle - 74 -
6.2.1.3 Diagrama de actividad: item’s Activo - 76 -
6.2.1.4 Diagrama de actividad: ingreso mantención teórica - 78 -
6.2.1.5 Diagrama de actividad: ejecutar plan de mantención de un activo - 80 -
6.2.1.6 Diagrama de actividad: registro mantención calendario Google - 82 -
6.2.1.7 Diagrama de actividad: reportes - 83 -
6.2.1.8 Diagrama de actividad: sincronizador datos central / área - 84 -
6.2.2 Diagrama Modelo de Clases - 85 -
6.2.3 Diseño Sitio Web - 88 -
6.2.3.1 Especificación de pantallas - 89 -
6.2.3.2 Estandarización de nombres de página - 89 -
6.2.3.3 Jerarquía de archivos - 92 -

6.2.3.4 Estandarización de pantallas ..... - 93 -
6.2.4 Diseño de Procedimientos de Carga Inicial de Datos. ..... - 95 -
7 Diseño de Base de Datos ..... - 96 -
7.1 Diseño Conceptual ..... - 97 -
7.1.1 Entidades ..... - 97 -
7.1.1.1 Entidades Módulo Activos. ..... - 98 -
7.1.1.2 Entidades Módulo Mantenciones ..... - 101 -
7.1.1.3 Entidades transversales para todos los módulos ..... - 103 -
7.1.2 Relaciones. ..... - 104 -
7.1.2.1 Relaciones Módulo Activos. ..... - 104 -
7.1.2.2 Relaciones Módulo Mantenciones ..... - 108 -
7.1.2.3 Relaciones transversales para todos los módulos ..... - 112 -
7.1.3 Identificar y asociar atributos con una entidad o relación ..... - 113 -
7.1.3.1 Módulo Activos. ..... - 113 -
7.1.3.2 Módulo Mantenciones. ..... - 120 -
7.1.3.3 Módulo Reportes ..... - 122 -
7.1.3.4 Identificación de Atributos con entidades o relaciones transversales a todos los módulos. ..... - 122 -
7.1.4 Documentación de los Dominios de Atributos ..... - 125 -
7.1.5 Identificación de Atributos para Claves Primarias y Candidatas - 126 -
7.1.7 Diagrama Modelo Conceptual BDPROMO ..... - 128 -
7.2 Modelo Lógico BDPROMO (“Base de Datos Sistema Programa Mantención Operaciones”) ..... - 130 -
7.2.1 Mapeo del Modelo Conceptual al Modelo Físico ..... - 130 -
7.2.1.1 Relaciones “Muchos - a - Muchos” ..... - 130 -
7.2.1.1.1Relaciones “Muchos - a- Muchos” para módulo activos ..... - 130 -
7.2.1.1.2 Relaciones “Muchos - a- Muchos” para módulo mantenciones - 133 -
7.2.1.1.3..... Relaciones “Muchos - a- Muchos” Transversales para todos los módulos ..... - 135 -
7.2.1.2 Relaciones “Complejas” ..... - 135 -
7.2.1.2.1Relaciones “Complejas” módulo Activos ..... - 135 -
7.2.1.2.2Relaciones “Complejas” módulo mantenciones ..... - 136 -
7.2.2 Derivar relaciones del modelo de datos lógico ..... - 137 -
7.2.3 Validar el Modelo Utilizando Normalización ..... - 141 -
7.2.3.1 Primera Forma Normal (1FN) ..... - 142 -
7.2.3.2 Segunda Forma Normal (2FN) ..... - 142 -
7.2.3.3 Tercera Forma Normal (3FN) ..... - 144 -
7.2.4 Transacciones de Usuario ..... - 147 -
7.2.5 Diagrama ER Final ..... - 150 -
7.2.6 Definir restricciones de integridad ..... - 151 -
7.2.6.1 Integridad referencial ..... - 151 -

7.2.6.2 Restricciones de la empresa ..... - 152 -
7.3 Construir y validar el modelo de datos lógico global ..... - 154 -
7.4 Traducir el modelo lógico global para el DBMS ..... - 154 -
7.4.1 Diseñar las relaciones bases para el DBMS especificado. ..... - 154 -
7.4.2 Diseñar las restricciones de la empresa para el DBMS especificado. - 155 -
7.4.3 Implementación ..... - 155 -
7.5 Diseñar representación Física ..... - 159 -
7.5.1 Analizar transacciones ..... - 159 -
7.5.2 Elegir organización de archivos ..... - 160 -
7.5.3 Elegir índices secundarios ..... - 160 -
7.5.4 Introducción de redundancia controlada ..... - 160 -
7.5.5 Estimar los requerimientos de espacio en disco ..... - 161 -
7.6 Diseñar mecanismos de seguridad ..... - 161 -
7.6.1 Diseñar vistas de usuario ..... - 161 -
7.6.2 Diseñar reglas de acceso ..... - 162 -
7.7 Monitoreo y refinamiento del sistema operacional. ..... - 162 -
8 Construcción ..... - 163 -
8.1 Principales Pantallas ..... - 166 -
8.1.1 Pantalla Inicio Sistema ..... - 166 -
8.1.2 Pantalla Menú Principal ..... - 166 -
8.1.3 Pantalla Menú Activos ..... - 167 -
8.1.4 Pantalla Ingreso Mantenciones Teóricas ..... - 168 -
8.1.5 Pantalla Plan Mantención Teórica. ..... - 169 -
8.1.5 Pantalla Información Básica Activo. ..... - 172 -
8.1.6 Pantalla Lista Mantenciones Efectivas. ..... - 175 -
8.1.7 Pantalla Ingreso detalle Mantenciones Efectivas. ..... - 176 -
8.2 Implementación de la Réplica de base de datos ..... - 178 -
9 Pruebas ..... - 182 -
9.1 Conceptos generales de las pruebas realizadas ..... - 182 -
9.1.1 Pruebas de Unidad. ..... - 182 -
9.1.2 Pruebas de Integración. ..... - 183 -
9.1.3 Pruebas de Sistema y Evaluación ..... - 183 -
9.1.4 Prueba de Seguridad ..... - 185 -
9.2 Plan de pruebas. ..... - 186 -
9.2.1 Pruebas Funcionales ..... - 186 -
9.2.2 Estrategias de las Pruebas ..... - 187 -
9.2.3 Tipos de Pruebas ..... - 188 -
10 Conclusiones ..... - 192 -
11 Bibliografía ..... - 195 -
12 Anexos ..... - 196 -

SINTESIS

Las empresas del sector salmonero, dada las distribuciones geográficas bajo las cuales están inmersas, están obligadas a mantener una serie de instalaciones en lugares tan dispersos como de difícil acceso. Dichos centros operativos requieren de activos para su funcionamiento. Es por ello que les resulta indispensable contar con dichos activos en óptimas condiciones para asegurar un ciclo productivo con los menores inconvenientes.

Con este antecedente se implementó un sistema sobre plataforma Web cuyo objetivo principal fue apoyar un completo registro y control de los activos de una empresa salmonera, en cada una de sus instalaciones, obteniendo así una plataforma integrada y homogénea que permitió contar con información de gestión consolidada y centralizada, a disposición de toda la compañía para apoyar las decisiones en el ámbito operativo, administrativo y contable.

Para poder realizar un sistema que entregue una solución acorde con los requerimientos solicitados y obtener un producto de calidad, se utilizó una metodología ágil para el proceso de desarrollo, ya que facilita y posibilita una realimentación continua entre el equipo desarrollador y el cliente, en donde el

principal objetivo es la satisfacción del cliente, buscando dar al cliente (usuario), el software que el necesita y cuando lo necesita.

Una vez realizadas todas las etapas que propone la metodología seleccionada, se obtuvo como resultado un producto de software capaz de administrar, gestionar y proveer de información consolidada de los activos del área farming de una importante compañía salmonera.

ABSTRACT

The companies of the salmon farming sector, due the geographical distributions under which they are working, are forced to support a series of facilities in places as dispersed as of difficult access. The above mentioned operative centers needs assets for his functioning. For that, it turns to be indispensable to have the assets mentioned above in optimal conditions to assure a productive cycle with the minimum inconvenient.

With this precedent, a Web plataform system was implemented which principal target was to support a complete record and control of the assets of a salmon farming company, in each of his facilities, obtaining an integrated and homogeneous platform that allowed to be provided with information of consolidated and centralized management, available for the whole company to support their decisions in the operative, administrative and countable ambience.

To be able to realize a system that delivers a solution according to the requirements and obtain a quality product, an agile methodology was used for the development process, since it facilitates and there makes a continuous feedback possible between the team developer and the client, where the

principal target is the satisfaction of the client, thinking about how to give to the client (user), the software that he need and when he needs it.

As soon as there were realized all the stages that the chosen methodology proposes, it was obtained a software product capable of administering, managing and providing consolidated information of the assets of the farming division of an important salmon farming company.

1. Introducción

El explosivo crecimiento industrial y comercial de la región en el área de la salmonicultura, sustentado en la gran producción y exportación de sus productos y derivados a partes del mundo, ha permitido que las empresas del salmón, en su afán de mantenerse y posicionarse mejor en un mercado cada vez más competitivo y exigente, posean una gran infraestructura que por lo general se encuentra distribuida entre las regiones décima y duodécima. Lo anterior se debe a la descentralización y expansión geográfica de sus recursos, que tiene cómo consecuencia la necesidad de generar la producción necesaria, para satisfacer un mercado que tiene un crecimiento sostenido.

Como se mencionó anteriormente, la necesidad de generar alimentos de calidad y satisfacer la gran demanda de los productos del salmón, hace primordial el control y correcto funcionamiento de todos los recursos que funcionan activa e integralmente en todo el proceso que conlleva la producción de alimentos. La modernización y automatización de las herramientas de control, seguimiento y planificación, se hace de vital importancia para lograr este objetivo que va de la mano con la innovación tecnológica que está experimentando la industria salmonera.

Por ello, la actualización de herramientas para el control de recursos permite proveer de mecanismos efectivos para determinar la ubicación de ellos y detectar si estos desempeñan a cabalidad con su “tarea”.

Para solucionar esta problemática, el área de cultivos de una empresa de la industria del salmón, ha decidido implementar un sistema que permita controlar y planificar las mantenciones preventivas y correctivas que realiza sobre sus activos, aplicándolo en los diversos centros de cultivos que posee y en las oficinas centrales que se encuentran ubicadas en la ciudad de Puerto Montt.

Desde el punto de vista informático, se propone la creación de un sistema sobre plataforma Web y que tenga cómo objetivo centralizar la información de los activos, también registrar la planificación de las mantenciones preventivas y correctivas sobre ellos, describiendo el procedimiento realizado a los activos que se encuentran en distintos puntos geográficos de la décima y undécima región.

Aunque las distancias geográficas sean no despreciables, el continuo funcionamiento en el control de los activos se realizará, aunque NO exista “conexión” vía Internet, ya que la solución informática propuesta, permitirá realizar tareas localmente, es decir, se registrará la información en una base de

datos local, la cual se sincronizará con la base de datos central una vez que la conectividad sea reestablecida.

Cabe señalar que la creación e implementación de este sistema facilitará enormemente la toma de decisiones en el aspecto financiero, ya que permitirá una proyección administrativa más acabada para destinar de mejor forma los recursos monetarios en la compra o reparación de activos. Además la aplicación contemplará un completo registro de los activos que tiene la compañía en el área de cultivos, así cómo la interacción con el ERP de la compañía. Con esto se busca integrar y lograr que todas las plataformas funcionen de forma homogénea y con fluidez para entregar tiempos de respuesta aceptables.

El rol del alumno será el de diseñar y desarrollar un sistema de información basado en una plataforma Web, que sea capaz de centralizar todas las características de los activos del área farming que posee la compañía y establecer mecanismos efectivos para el control de los mismos, además de otras funcionalidades que el sistema deberá ofrecer para establecer y disponer de un acceso eficiente y eficaz para los datos que requieran los usuarios.

Para lograr esto el alumno deberá lograr un conocimiento del negocio bastante importante, y consensuar opiniones que se generaran durante todo el desarrollo

del sistema por parte del grupo de trabajo que dispondrá la empresa salmonera para estos efectos. Todo esto para poder entregar funcionalidades que puedan satisfacer claramente los objetivos planteados.

Para llevar a cabo el desarrollo del sistema antes mencionado, se utilizará una metodología adaptada XP (Programación Extrema), y la base de datos seguirá las técnicas propuestas por Thomas Connolly y Carolyn Begg.

1.1 Resumen de capítulos

Capítulo 1: Introducción

En este capítulo se introduce al lector acerca del contenido de fondo de este proyecto de tesis.

Capítulo 2: Objetivos

Definiciones de los objetivos generales y específicos que alcanzara este proyecto de tesis

Capítulo 3: Planteamiento del Problema

Capítulo que detalla los antecedentes que propician el desarrollo de este proyecto de tesis.

Capítulo 4: Metodología

En este capítulo se señala y describe la metodología utilizada para el desarrollo del proyecto.

Capítulo 5: Recursos

Capítulo en el cual se detallan los recursos, tanto de hardware como de software utilizados para el desarrollo del sistema que da origen a este seminario de tesis.

Capítulo 6: Análisis y Diseño del Sistema

Capítulo que describe todo lo relacionado con la etapa de análisis, algunos procesos que se destacan son: casos de uso, requerimientos del sistema etc. También se documenta todo lo relacionado con el diseño de la aplicación.

Capítulo 7: Diseño Base de datos

Capítulo que tiene como objetivo documentar y explicar todo lo relacionado con el diseño de la base de datos.

Capítulo 8: Construcción del Sistema

Capítulo donde se muestra documentación de la construcción del sistema.

Capítulo 9: Pruebas de Sistema

Capítulo que detalla las pruebas realizadas sobre el sistema y evaluación de los resultados.

Capítulo 10: Conclusiones

Capítulo en el que se describen las conclusiones obtenidas al término del proyecto.

Capítulo 11: Bibliografía

En este capítulo se hace referencia a todas las fuentes de información utilizadas para apoyar el desarrollo de este seminario de tesis.

Capítulo 12: Anexos

Último capítulo de este informe, el cual muestra contenidos específicos para que el lector se interiorice completamente en este seminario de tesis.

2 Objetivos

2.1 Objetivo General

Desarrollar una herramienta informática para apoyar el control de activos del área farming de una empresa salmonera.

2.2 Objetivos Específicos

- Centralizar la información referente a los activos del área de cultivos que posee una empresa del rubro salmonero

- Integrar sistemas ya existentes en la compañía específicamente su ERP, con el módulo de activos. Esto para la obtención de información previamente ingresada en dichos sistemas. Además el módulo de mantención integrará la planificación de las mantenciones con el calendario que ofrece Google.

- Capacidad de trabajo independiente de todos los módulos del sistema de la aplicación central, mediante bases de datos distribuidas que comunicarán y sincronizarán los centros de cultivos con las oficinas centrales de la compañía.

3 Planteamiento del Problema

3.1 Antecedentes

3.1.1 Definición del Problema

La salmonicultura a nivel mundial ha crecido aceleradamente, con una expansión de mercado que hoy abarca todo el mundo. Por esto la industria del Salmón en Chile ha debido adaptarse a este crecimiento, adquiriendo una gran tecnología y maquinaria que sea capaz de apoyar la generación de los distintos productos que se generan a partir del salmón.

La cantidad de recursos utilizados en la producción de alimentos por parte de las diversas empresas ligadas a la industria salmonera, es no menor y por ende se hace cada vez más difícil de controlar, administrar, mantener y realizar un posterior seguimiento de estos, por estar diversificados en plantas de procesos, centros de cultivos, etc., que se encuentran en la extensa y complicada geografía del sur de Chile.

En consecuencia, se crea una problemática para las empresas, si no se tienen tecnologías de información que puedan proporcionar datos fidedignos del lugar

en donde se encuentran los recursos y el estado en que están operando. De acuerdo a lo mencionado anteriormente se producen los siguientes problemas:

a. Nula centralización de información de los activos referente al real estado en que se encuentran funcionando.
b. Lento acceso a la información que dice relación con las mantenciones que se realizan a cada uno de los activos.
c. Difícil detección de los activos que generan más pérdidas de dinero y poder así detectar oportunamente cuando reemplazarlos.
d. Escaso análisis para la planificación de mantenciones a los diversos activos, por tener información poco actualizada, oportuna y confiable.
e. Alto grado de duplicidad e inconsistencia de información.
f. La utilización de documentos en papel para el registro de mantenciones y control de activos, tiene como resultado manejar grandes volúmenes de información con un alto costo en tiempo del personal encargado, además de su engorroso y poco seguro manejo.
g. Escasa documentación para realizar estadísticas de los activos que requieren de mayores mantenciones y así poder extender su vida útil.

3.1.2 Identificación de Esfuerzos Anteriores para Resolverlo

A la fecha no existen avances anteriores por solucionar esta problemática, ya que la empresa en cuestión, centraliza la totalidad de su información en un ERP[1] llamado “Oracle E-Business Suite Financials”, el cual no satisface la resolución de los problemas mencionados anteriormente.

3.1.3 Solución Propuesta

El ingreso a un constante mercado competitivo, como en el que se encuentra inmerso esta empresa, vendrá acompañado de una previsible adaptación a los nuevos sistemas de información y a su evolución tecnológica. Por esto, se considera necesario el desarrollo de un sistema que permita controlar, administrar, y registrar las mantenciones que se realizan a cada uno de los activos que tiene el área de cultivos. Todo esto debe ser soportado por bases de datos que registraren toda la información mencionada anteriormente para la generación de reportes con datos reales y confiables. Éste es un punto importante para que en el futuro se analicen estos factores y se puedan planificar las mantenciones de activos de manera correcta, ya que de esta forma se podría confirmar que elementos son los que necesitan de una mayor inversión monetaria, y así en un futuro ver las mejores alternativas en la compra de nuevos activos.

[1] Enterprise Resourse Planning, sistema de gestión de información estructurado

Con estos antecedentes la solución propuesta comprende los requerimientos planteados por el área de cultivo de una empresa del rubro salmonero durante reuniones realizadas en sus oficinas, que incluyen: mantenciones preventivas y correctivas de activos, limitación a información que es responsabilidad del usuario, información orientada también a proveedores, inclusión de procedimientos de mantenciones, registro de mediciones variables, registro del costo de cada mantención, baja de equipos, manejo de carta Gantt para mantenciones planificadas, completo registro de los activos divididos en las principales sub-áreas que pertenecen al área de cultivos de la compañía, integración del módulo de activos con el ERP, visualización de un RSS con información relevante para el área de la salmonicultura, reportes que entreguen información orientada a inventarios y mantenciones según filtros ingresados por el usuario, registros de equipos eliminados, registro de los activos que se han trasladado de área o centro.

Este Sistema de información se desarrollará en una arquitectura Web, que constará de 3 módulos que están interconectados entre sí:

- Módulo Activos: el objetivo de este módulo es centralizar toda la información concerniente a la identificación de los activos que tiene el área de cultivos además de mecanismos que permitan la búsqueda, registro y modificación de la información asociada a cada activo. Además

este módulo se integrará con el ERP de la compañía, es decir, utilizará procedimientos que permitan interactuar con motores de bases de datos ORACLE que soportan y almacenan la información de dichos sistemas, evitando así duplicidad de información entre sistemas que pertenecen a la misma compañía.

- Módulo Mantenciones: compuesto por las planificaciones de las mantenciones que se desarrollarán a todos los activos previamente ingresados en el Sistema. Este módulo entrega mecanismos para el registro de todas las actividades que se realizan en una mantención. Conjuntamente con la creación de mantenciones en el Sistema, este módulo registrará un evento con el resumen de cada una de las mantenciones creadas, en el calendario que ofrece Google[2]. Dicho resumen se creará en la cuenta de correo del usuario que supervise la realización de la mantención planificada, el usuario supervisor se establecerá según rol o perfil de usuario. Esta integración se logrará utilizando las herramientas que ofrece Google para integrar la plataforma de desarrollo Microsoft Visual Studio.NET 2003 con el mencionado calendario gráfico que posee una cuenta de correo Google.

[2] Motor de búsqueda en Internet, que además provee de distintos servicios como por ejemplo: correo, calendario, Google Heart etc.

- Módulo Reportes: compuesto por la información consolidada que se desplegará según filtros que serán ingresados por el usuario que requiera visualizar dicha información. Este módulo será desarrollado en su totalidad con la herramienta Reporting Services que ofrece soluciones de servidor y permite crear informes tradicionales en papel cómo informes Web dinámicos.

Lo apartado y lejano que podrían encontrarse algunos centros de cultivos con la aplicación central, conllevaría un posible problema de sincronización de información que se reflejaría finalmente en verdaderas “islas de información”, y que produciría un desconocimiento total de la ubicación y estado en que se encuentran operando los activos y no se cumpliría el objetivo de centralizar la información de la compañía. Para mitigar este problema y lograr capacidad de trabajo local e independiente de forma continua, se propone la utilización de todos los módulos del sistema implementados en los centros de cultivos, con un motor de base de datos SQL SERVER 2005 en su versión “express”, que almacenaría la información del sistema y que permitiría apoyarse en los mecanismos, que provee dicho motor de base de datos, para sincronizar la información con el sistema central, utilizando para ello los métodos de publicación, distribución y suscripción de datos. Además la información que replicará el sistema central, sólo “llegará”, a las áreas respectivas en donde se

utilizará dicha información, debido en gran parte al diseño de la base de datos que soportará al sistema y obviamente gracias a los mecanismos que ofrece el motor de base de datos empleado.

Cabe señalar que el sistema contempla el manejo de roles de usuario para el acceso al sistema y visualización de información según privilegios. Todo esto controlado y validado por la base de datos que soporta al sistema.

Basados en los requerimientos planteados., se ha diseñado la presente solución que se gráfica en el figura [1].

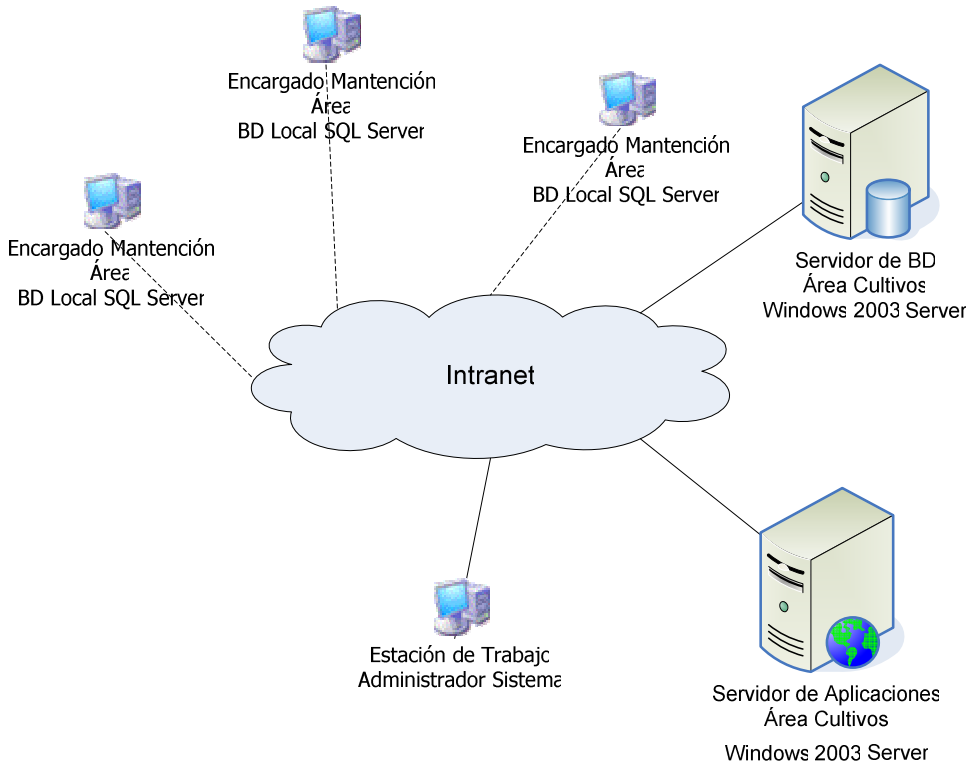


Figura Nº [1]: Solución Propuesta.

3.1.4 Definición Equipo de trabajo

A continuación se describen las responsabilidades de cada uno de los participantes en el proyecto. Este equipo de trabajo estará compuesto por las siguientes personas:

- *Sr. Óscar Salazar Cerna*: Profesor Patrocinante, participa como guía y tutor de los diferentes objetivos del proyecto, apoyo en el análisis y desarrollo del proyecto. Además se encarga de planificar la Gestión de riesgos y control del proyecto.

- *Sr. Erick Nayán Barría*: Captura, especificación y validación de requisitos, interactuando con el cliente y los usuarios mediante entrevistas, encargado de la elaboración del análisis, diseño y construcción del Sistema en cuestión y posterior puesta en marcha.

3.2 Justificación

3.2.1 Situación Sin Proyecto

Hoy en día se torna de vital importancia poseer tecnologías de información acordes con el acelerado proceso de cambio del mercado que tiene cómo objetivo: disminuir costos de inversión y producción de sus recursos de forma eficiente y eficaz.

Por esto, el hecho de no contar con procesos de gestión y control de recursos automatizados y de fácil acceso, retarda de forma considerable cualquier esfuerzo de realizar diagnósticos preventivos, para los variados recursos que podría poseer la empresa.

Cabe señalar que la optimización en el funcionamiento de los diversos activos utilizados en las distintas fases que comprende la elaboración y consecución de los productos del salmón, esta ligada de forma directa con la centralización de la información concerniente a cada uno de los elementos que permiten dicha tarea. Esto se traduce en planes acordes de mantenimiento para la variada gama de recursos, a la eficaz reacción ante fallas y posteriores reparaciones de éstos.

Finalmente la falta de mecanismos para solucionar los problemas anteriormente descritos, afecta de sobre manera la producción que en definitiva se refleja en costos monetarios para la compañía y que tiene por misión solucionar este proyecto a desarrollar.

3.2.2 Situación con Proyecto

La situación con la puesta en marcha del proyecto, propone mejoras sustanciales en el actual trabajo que se realiza con los recursos de la compañía, distribuidos en diversos centros de cultivos. Esto producto de la centralización total de la información que se reflejará en un sistema robusto, que proporcionará todos los datos necesarios para la realización de diagnósticos eficaces y pertinentes de cada uno de los recursos cuando sea requerido.

La implementación de este sistema permite ventajas esenciales, para poseer un registro total de los activos del área de cultivos, en conjunto con los planes de mantenciones que tendrán los distintos activos que pertenecen a la compañía y que se encuentran dispersos en sus diferentes centros de cultivos. Con esta herramienta se tendría accesibilidad oportuna y eficiente a la información de los activos desde cualquier centro de cultivo, facilitando de sobre manera el control y seguimiento de las prestaciones de un activo en particular.

La integración del sistema con el ERP de la compañía ofrece un ambiente homogéneo para los usuarios, ya que la información será consistente y no existirá en ningún caso duplicidad de información.

3.3 Delimitación

De acuerdo a la filosofía de la metodología elegida para este proyecto y que se detallará más adelante, todos los resultados tangibles de éste son objetos de modificaciones a lo largo del desarrollo del proceso, con lo cual, sólo al término del proyecto se podrán señalar las reales limitaciones del proyecto.

Basándose en lo descrito anteriormente y de acuerdo a una estimación realizada por el equipo de trabajo que desarrollará el sistema, y apoyada en el levantamiento de requerimientos realizada a la contraparte del proyecto y quien actúa como representante de la empresa en el desarrollo de este sistema, se pueden establecer algunas limitaciones en el alcance total del proyecto, cómo se detallan a continuación:

- No contemplará la identificación física de los activos que controlará el sistema.

- La información que se visualice en el sistema será el resultado del poblamiento inicial, según datos entregados por la empresa y del exclusivo ingreso de información por parte de los futuros usuarios del sistema.

Este proyecto constituye el análisis, diseño, construcción e implantación del sistema. No involucra acciones anexas a la puesta en marcha, como configuraciones de servidor, configuración de seguridad y capacitación de usuarios.

4 Metodología

El enfoque metodológico en el cuál se basará este proyecto de tesis será la metodología de desarrollo XP[3] [Kent Beck 1999], que es la más destacada de los Procesos Ágiles de Desarrollo de Software y que basa su desarrollo en la adaptabilidad y no en la previsibilidad. Las características esenciales en que basa su desarrollo son las siguientes:

- Satisfacción del cliente: esta metodología trata de dar al cliente el software que él necesita y cuando lo necesita.
- Potenciar al máximo el trabajo en grupo. Tanto el jefe de proyecto, los clientes y desarrolladores, son parte del equipo y están involucrados en el desarrollo del software.
- La necesidad de un proceso de desarrollo que permita obtener resultados de forma prematura.
- Un proceso iterativo permite una comprensión creciente de los requerimientos a la vez que se va haciendo crecer el sistema.
- Un proceso de desarrollo que se vaya creando a medida que se obtienen o que se desarrollan y maduran sus componentes.

[3] Programación Extrema

- Evidenciar tempranamente los errores cometidos en cada etapa del ciclo de vida del proceso de desarrollo del software.
- Los procesos se aceptan y se acuerdan, no se imponen.

Dado estos argumentos, se utilizará la metodología para el proceso de desarrollo denominada “Ciclo iterativo de Desarrollo de Software”, que corresponderá a la gráfica de la figura [2]:

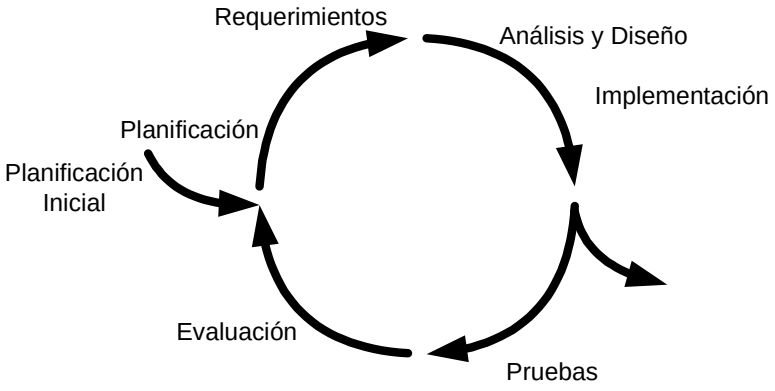


Figura Nº [2]: Ciclo iterativo de Desarrollo de Software.

La metodología “Ciclo iterativo de Desarrollo de Software”, no ofrece un marco metodológico para modelar una base de datos, por lo que decidió implementar la metodología denominada “Ciclo de Vida de Bases de Datos” [Connolly1999], propuesta por Thomas Connolly y Carolyn Begg, ya que se ajusta de mejor forma para el desarrollo propuesto para el desarrollo de esta Tesis.

5 Recursos

5.1 Hardware

El hardware para el desarrollo del proyecto requiere las siguientes características:

- Procesador: Mobile AMD Sempron 1.8 Ghz
- Memoria RAM: 512 MB DDR2
- Disco Duro: *80 GB*
- Red: Ethernet Realtek RTL8139

Las características que debe cumplir el equipo para la producción del sistema son las siguientes:

- Procesador: Xeon 3.5 GHZ o superior
- Memoria RAM: 2 GB DDR2
- Tarjeta Dual LAN 10/100/1000
- Tarjeta de Video 8 MB

5.2 Software

Este punto hará referencia al software utilizado para llevar acabo el desarrollo, y producción del proyecto.

5.2.1 Software para el desarrollo del proyecto

Sybase PowerDesigner 10.0: PowerDesigner es un conjunto de herramientas que permiten modelar y combinar distintas técnicas estándar de modelamiento: modelamiento de aplicación a través de UML, técnicas de Modelamiento de Procesos Empresariales y técnicas tradicionales de modelamiento de bases de datos.

Microsoft Visual Studio.NET 2003: Visual Studio .NET es un conjunto de herramientas integrado para la construcción y desarrollo de servicios Web XML, aplicaciones basadas en Microsoft Windows®, y soluciones Web.

Active Server Page (ASP .NET): ASP .NET es una tecnología de Microsoft para generar páginas Web de forma dinámica, mezclando código de scripts del lado del servidor (incluyendo acceso a base de datos) con HTML y código del lado del servidor. A diferencia de ASP clásico, ASP .NET reemplaza los lenguajes

interpretados como VBScript o JScript por lenguajes *compilados* a código intermedio (llamado MSIL o Microsoft Intermediate Language) como Visual Basic, C#, o cualquier otro lenguaje que soporte la plataforma .NET.

Subversion: herramienta para controlar las versiones del proyecto.

Microsoft Office: suite o conjunto de programas de Microsoft para trabajo de oficina; incluye Word, Excel, Project etc.

5.2.2 Software para el servidor de producción del proyecto

Windows Server 2003: es un sistema operativo de propósitos múltiples capaz de manejar una gran gama de funciones de servidor, en base a las necesidades, tanto de manera centralizada como distribuida.

SQL Server Reporting Services 2000: es una completa solución de servidor que permite crear, administrar y entregar tanto informes tradicionales en papel como informes Web interactivos.

Internet Information Server 6.0 (IIS): Es una serie de servicios para computadores que funcionan con Windows. Los servicios que ofrece son: FTP,

SMTP, NNTP y HTTP/HTTPS, pudiendo dejar un computador cliente como servidor WEB.

Microsoft SQL Server 2005: Es un sistema de gestión de bases de datos relacionales (SGBD) basada en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. SQL Server 2005 se caracteriza por la facilidad de uso, escalabilidad y fiabilidad en el almacenamiento de los datos.

*Microsoft SQL Server 2005 Express Edition: E*s un producto gratuito basado en la tecnología SQL Server 2005 que utiliza el mismo motor de base de datos confiable y de alto rendimiento que el resto de las versiones de SQL Server 2005.

6 Análisis del Sistema

De acuerdo a las características que presenta este proyecto de Tesis, el software es parte fundamental para concretar los objetivos planteados por el equipo de trabajo, que se centraliza finalmente en un sistema con toda la funcionalidad e integración total propuesta y llevada a cabo con éxito.

Dicho sistema será guiado por la metodología de desarrollo XP, que se puede englobar dentro de las metodologías ligeras, que son aquellas que priorizan los resultados directos y que reducen la documentación tradicional que ofrecen la mayoría de las metodologías. XP es una metodología que permite un desarrollo de software adaptable más que previsible, se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. Lo cual posibilita un software escalable y de un mejoramiento futuro de forma dinámica.

6.1 Descripción y Conceptos Generales del sistema

El sistema desarrollado para este proyecto consiste en una aplicación de arquitectura web, que tiene como principal objetivo centralizar la información referente a los activos del área de cultivos y permitir el completo registro de las mantenciones que se realizan a los distintos activos de una empresa del rubro

salmonero. Todos los antecedentes que permitirán la implementación del sistema se mostrarán en detalle en las siguientes secciones del presente documento.

Para un mejor entendimiento se presentarán a continuación conceptos generales que serán utilizados en el presente documento y que determinarán el diccionario técnico del proyecto.

Conceptos Generales
PROMO: Nombre del sistema y que significa Programa Mantención Operaciones.
Activo: Conjunto de bienes con valor monetario que se reflejan en la contabilidad de la empresa.
Mantención: Acción o efecto de mantener un activo en el tiempo.
Centro: Lugar donde se desarrolla una actividad en particular.
Área: Sector geográfico que comprende varios centros.
FA: Aplicación financiera de la compañía.
Código AF: Cada activo tiene asignado un código que pertenece a la Aplicación Financiera de la compañía. Este código impide la duplicación de activos en las distintas aplicaciones.
Mantenciones Teóricas: Se denomina a los planes de mantenciones que se generan según un modelo en especial.
Mantenciones Efectivas: se produce cuando se ejecuta el plan de mantención para un activo en particular, generado según una mantención teórica.
Minutas: Documento en donde se contempla las actividades a realizar según los requerimientos establecidos en las reuniones.

Tabla Nº [1]: Conceptos Generales.

Para el análisis, se torna de suma importancia la captura de requerimientos, que permiten encontrar las funcionalidades que el sistema debe cumplir con éxito y así satisfacer los objetivos propuestos. Para esto, XP, propone “historias de usuarios”, el cual representa la necesidad del usuario en cuanto a lo que debe realizar el sistema. Para esto se estableció la siguiente figura [3], que muestra el flujo de información que proporciona una historia de usuario.

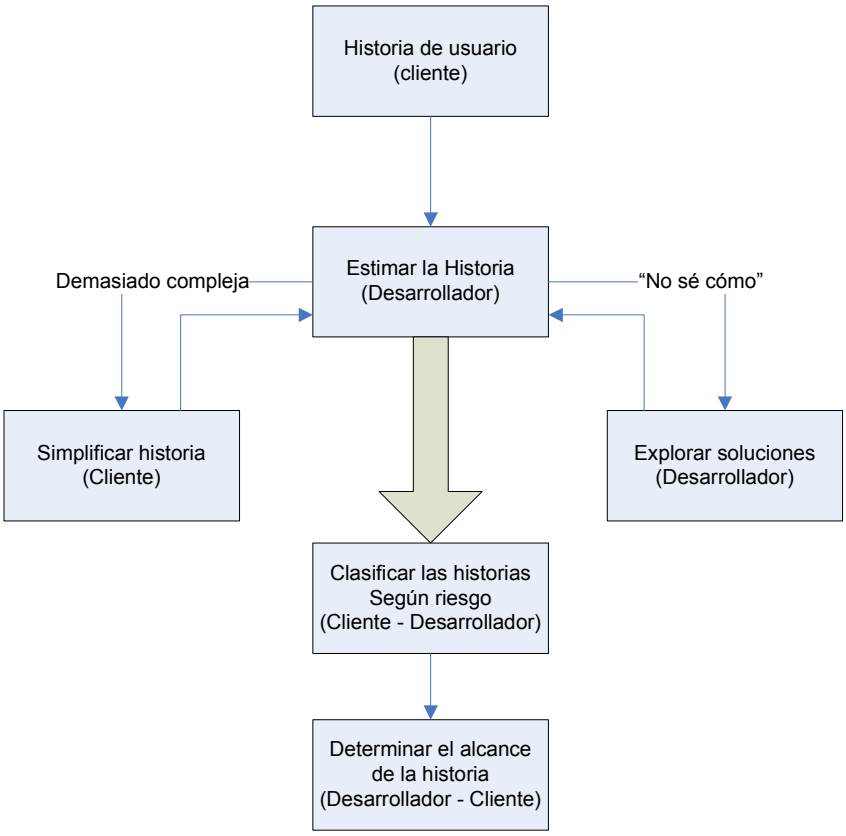


Figura Nº [3]: Flujo información historia de usuario.

De acuerdo a este flujo de información se puede desprender el siguiente plan de iteración para las historias de usuario.

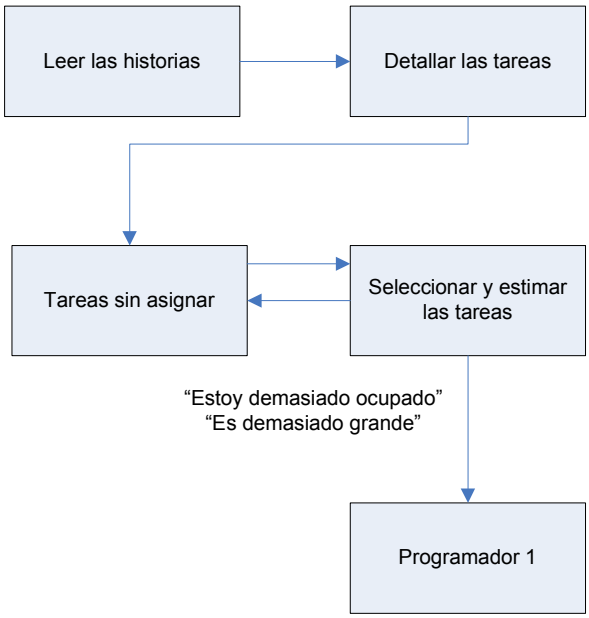


Figura Nº [4]: Plan de iteración para las historias de usuarios.

Cabe mencionar que el desarrollo de las tareas será realizada por un programador. Evidentemente que el caso ideal sería que existieran más programadores para agilizar el desarrollo de las tareas.

A continuación se describen en la siguiente tabla [2] los requerimientos más importantes, ya que la totalidad de los requerimientos serán detallados con los casos de usos.

Resumen requerimientos	
ID	**Detalle requerimiento**
Req. 1.	Autenticar a cada usuario que opere con el sistema.
Req. 2.	Mantener un registro de todos los activos, para el área de cultivos.
Req. 3.	Mantener, para cada activo, una ficha técnica que lo describa técnicamente hablando.
Req. 4.	Cada activo debe ser asignado a un centro, el cual depende de un área, y a nivel de subgerencia serán planificadas y ejecutadas sus mantenciones.
Req. 5.	Registrar la planificación de una mantención, describiendo el procedimiento a realizar.
Req. 6.	Integración del sistema con el ERP de la compañía.
Req. 7.	Registrar la ejecución de una mantención, ingresando sus observaciones y costos.
Req. 8.	Definir mantenciones teóricas, que se debieran ejecutar cuando una variable sobrepase una cantidad especificada.
Req. 9.	Registrar bajas de equipos.
Req. 10.	Generar avisos cuando sucedan eventos de planificación.
Req. 11.	Mantener un registro de mediciones de variables realizadas a los activos durante su vida útil.
Req. 12.	Permitir la sincronización de las distintas bases de datos que se encuentran ubicadas en cada unos de los centros de cultivos de la compañía.

Tabla Nº [2]: Resumen Requerimientos.

En vista que las “historias de usuario”, tienen el mismo propósito que los “Casos de Usos”, para efectos de orden y mejor visualización se representarán de forma gráfica con un “Diagrama de Casos de Uso”, las acciones que el software debe contemplar, además de los actores involucrados en la acción y que deben llevar a cabo, según los requisitos obtenidos mediante las “historias de usuarios”, que

se plantearon en las reuniones de trabajo, por parte de los usuarios y que son detalladas en minutas.

Utilizando los casos de uso, se facilita la evolución y el continuo mejoramiento del software a través del tiempo.

De acuerdo a la adaptación que se ha realizado de las historias de usuario a casos de uso, se utilizarán las definiciones entregadas por la diagramación U.M.L[4]. Luego, del diagrama del modelo de casos de uso, se detallará cada uno de los casos de uso expuestos en dicho diagrama.

A continuación se presenta el diagrama [1] de casos de uso para el módulo de activos correspondiente al ciclo actual de desarrollo de software.

[4] Lenguaje Unificado de Modelado

Diagrama Nº [1]: Modelo de Casos de Uso Módulo Activos.

Los actores presentes son:

- Administrador Sistema.

- Administrador Oracle.

A continuación se detallan los casos de uso para el módulo activos.

6.1.1 Módulo Activo

6.1.1.1 Caso de uso: Ingreso Nuevo Activo.

ACTOR (ES): *administrador sistema*	**Sistema**
1. Selecciona la opción “Activos”.	2. Lista árbol con tipos de activos.
3. Selecciona un tipo de activo.	4. Lista opciones: o Lista Activos PROMO. o Lista Activos AF. o Lista Mantenciones Teóricas.
5. Selecciona opción “Lista Activos PROMO”.	6. Muestra lista con activos ingresados, en caso de no haber activos ingresados, la lista se muestra sin elementos.
7. Presiona botón “Ingreso Nuevo Activo”.	8. Muestra posible información básica del activo a ingresar.
9. Ingresa información del activo.	
10. Presiona Botón “Guardar”.	11. Guarda el nuevo activo ingresado, quedando en estado pendiente para la definición en el Sistema FA
	12. Envía un correo electrónico al actor Administrador Oracle (encargado de finanzas), avisándole de la creación del nuevo activo.
Excepciones:	

a. Si el actor presiona botón “Volver”, entonces se visualizará la página anterior. b. Si el actor presiona botón “Buscar Oracle” (caso de uso que se detallará más adelante), entonces el sistema busca el activo en el sistema financiero. c. El actor puede realizar una búsqueda tipo Google en la lista de activos ingresados, es decir, basta ingresar cualquier término para realizar la búsqueda.
Observaciones:
d. El activo no se puede crear si no tiene como información básica, al menos los siguientes elementos: • Orden de Compra

6.1.1.2 Caso de uso: Búsqueda Activo Oracle.

ACTOR (ES): *administrador sistema*	**Sistema**
1. Selecciona pantalla de ingreso de información para un activo.	
2. Ingresa número de orden compra, presiona botón “Buscar Oracle”.	3. Muestra una lista de líneas de compra asociadas a la orden de compra.
4. Selecciona una línea en particular.	5. Los campos de información a ingresar se completan automáticamente.
6. Presiona botón “Guardar”.	7. Guarda el nuevo activo ingresado.
	8. Envía un correo electrónico al actor Administrador Oracle (encargado de finanzas), avisándole de la creación del nuevo activo.
Excepciones:	
a. Si el actor presiona botón “Volver”, entonces se visualizará la página anterior.	
Observaciones:	
b. No se puede realizar la búsqueda en el sistema financiero, si no se tienen al menos los siguientes elementos: • Orden de Compra	

6.1.1.3 Caso de uso: Modificar Activo.

ACTOR (ES): *administrador sistema*	**Sistema**
1. Selecciona pantalla lista activos, según tipo de activo.	2. Lista activos ingresados.
3. Selecciona activo.	4. Muestra formulario con información del activo con campos de ingreso inhabilitados.
5. Presiona botón “Modificar”.	6. Muestra formulario con información del activo con campos de ingreso habilitados.
7. Modifica información del activo.	
8. Presiona botón “Guardar”	9. Guarda información del activo.
Excepciones:	
a. Si el actor presiona botón “Volver”, entonces se visualizará la página anterior. b. Si el actor presiona botón “Eliminar”, ocurre el caso de uso detallado en el punto 6.1.1.5. c. Si el actor presiona botón “Ejecutar Plan Mantención”, se crea un plan de mantenciones para el activo.	
Observaciones:	
d. El sistema registra al usuario responsable de la modificación de la ubicación física del activo.	

6.1.1.4 Caso de uso: Eliminación de Activo.

ACTOR (ES): *administrador sistema*	**Sistema**
1. Selecciona pantalla lista activos, según tipo de activo	2. Lista activos ingresados.
3. Presiona botón “Eliminar”.	4. “lanza”, mensaje “¿Esta Seguro de Eliminar el Activo?”
5. Acepta Eliminar el activo	6. Elimina y cambia estado del activo a “ELIMINADO”.
Excepciones:	

a. Si el actor presiona botón “Volver”, entonces se visualizará la página anterior. b. Si el actor presiona botón “Modificar”, se deshabilitarán los campos de ingreso de información. c. Cuando se “lanza” el mensaje “¿Esta Seguro de Eliminar el Activo?”, y se presiona “Cancelar”, la operación de eliminación de activo no se realiza.
Observaciones:
d. El activo se elimina de forma virtual, ya que se debe mantener un histórico de los activos eliminados.

6.1.1.5 Caso de uso: Ingreso / Modificación / Eliminación Ítems de Activo.

ACTOR (ES): *administrador sistema*	**Sistema**
1. Selecciona la opción “Activos”.	2. Lista tipos de activos.
3. Selecciona el tipo de activo “Fondeo” - “Líneas de Fondeo”.	4. Lista opciones: o Lista Activos PROMO. o Lista Activos AF. o Lista Mantenciones Teóricas.
5. Selecciona opción “Lista Activos PROMO”.	6. Muestra lista con activos ingresados.
7. Selecciona activo.	8. Muestra formulario con información del activo con campos de ingreso inhabilitados.
9. Selecciona la opción “Ítems Activo”	10. Muestra posible información del ítem de activo a ingresar.
11. Ingresa información del ítem de activo.	
12. Presiona botón “Guardar”.	13. Guarda el nuevo ítem ingresado y se agrega en la lista (grilla) de ítems de activos.
14. En el proceso descrito en el punto 9, selecciona un ítem de activo que se encuentra en la lista (grilla).	15. Completa automáticamente los campos a ingresar con la información del ítem de activo seleccionado. Dichos campos

	se visualizan deshabilitados.
16. Presiona botón “modificar”.	17. Habilita campos de ingreso de información.
18. Modifica información del ítem de activo.	
19. Presiona botón “Guardar”.	20. Además de guardar la información del ítem de activo, “Lanza”, un mensaje “¡Datos guardados con Éxito!”.
21. En el proceso descrito en el punto 14, el actor además puede eliminar un ítem de activo presionando el botón “Eliminar”.	22. Elimina el ítem de activo permanentemente.
Excepciones:	
a. Si el actor presiona el botón “Volver”, entonces se visualizará la página anterior.	
Observaciones:	
b. No se puede crear el ítem de activo, si no se tiene al menos los siguientes elementos: • Orden de Compra	

6.1.1.6 Caso de uso: Ingreso / Modificación / Eliminación Recepciones Parciales de Activo.

ACTOR (ES): *administrador sistema*	**Sistema PROMO**
1. Selecciona pantalla lista activos, según tipo activo.	2. Lista activos ingresados.
3. Selecciona activo.	4. Muestra formulario con información del activo con campos de ingreso inhabilitados.
5. Selecciona la opción “Recepciones Activo”.	6. Muestra lista de recepciones parciales del activo.
7. Presiona botón “Agregar Nuevo”.	8. Muestra formulario en donde se ingresa la información de la recepción parcial.
9. Ingresa información.	

10. Presiona botón “Guardar”	11. Guarda la información ingresada.
12. En el proceso descrito en el punto 6, selecciona una orden de recepción que se encuentra en la lista.	13. Muestra información asociada a la orden de recepción seleccionada.
14. Presiona botón “Modificar”.	15. Habilita los campos de ingreso de información.
16. Modifica la información de la orden de recepción.	
17. Presiona botón “Guardar”.	18. Guarda la información ingresada.
Excepciones:	
a. Si el actor presiona el botón “Volver”, entonces se visualizará la página anterior.	
Observaciones:	

6.1.1.7 Caso de uso: Actualización Código AF del Activo.

ACTOR (ES): *administrador Oracle*	**Sistema**
1. Selecciona la opción “Activos”.	2. Lista tipos de activos.
3. Selecciona el tipo de activo.	4. Lista opciones: o Lista Activos PROMO. o Lista Activos AF. o Lista Mantenciones Teóricas.
5. Selecciona opción “Lista Activos AF”.	6. Muestra lista, según tipo activo seleccionado.
7. Selecciona la opción “Edición”, de un activo en particular	8. Muestra “caja de texto”, en la columna “Código AF”.
9. Ingresa valor en la “caja de texto”.	
10. Selecciona la opción “Actualizar”	11. Guarda el activo con el “código AF”.
Excepciones:	
a. Si el actor presiona la opción “Cancelar”, desaparecerá la caja de texto.	
Observaciones:	

b. Cuando se guarda el activo con su código AF, este desaparece de la lista de activos ORACLE.

6.1.1.8 Caso de uso: Baja Activo.

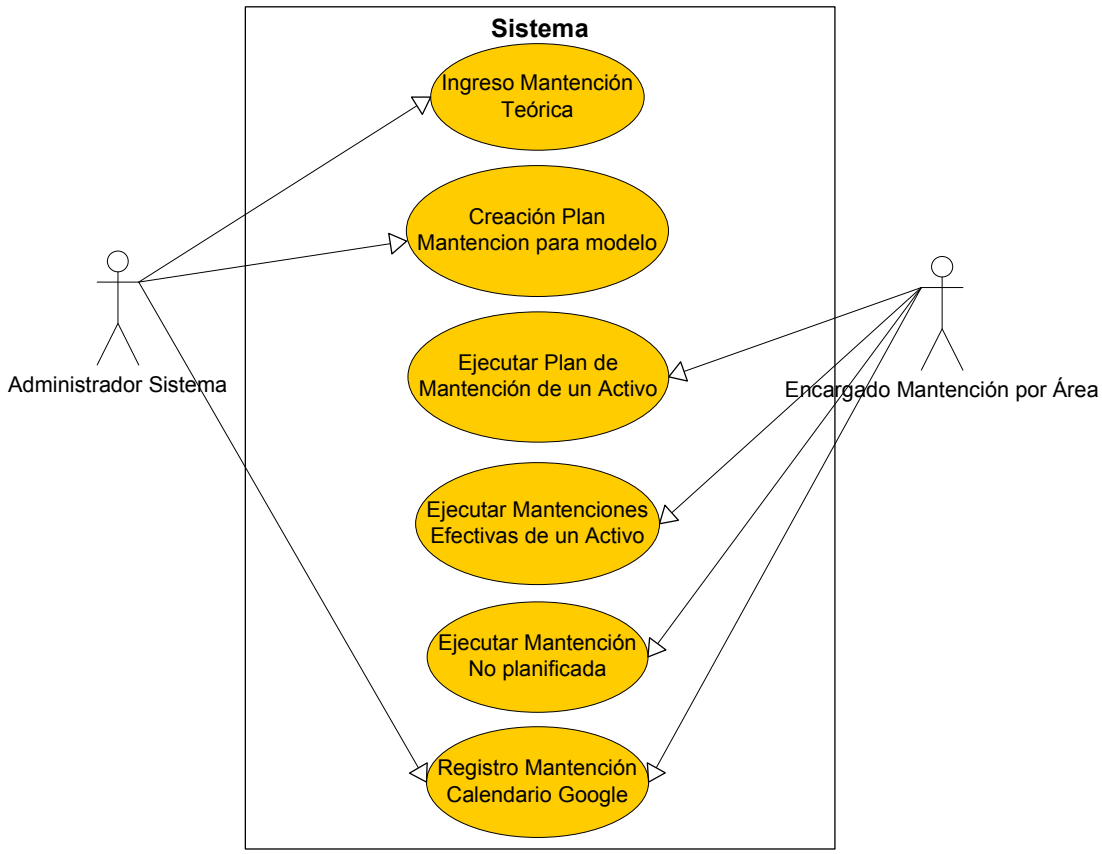
ACTOR (ES): *administrador sistema*	**Sistema**
1. Selecciona pantalla lista activos, según tipo activo.	2. Lista activos ingresados.
3. Selecciona activo.	4. Muestra formulario con información del activo con campos de ingreso inhabilitados.
5. Selecciona la opción “Mantenciones”.	6. Muestra Lista con mantenciones.
7. Selecciona mantención con estado “Planificada”.	8. Muestra botón “Baja Equipo”.
9. Presiona botón “Baja Equipo”.	10. “Lanza”, un mensaje “¿Esta seguro de Dar de Baja este Activo?”.
11. Presiona “Aceptar”.	12. El activo queda con estado “Baja”.
Excepciones:	
a. Si el actor presiona botón “Volver”, entonces se visualizará la página anterior. b. Si el actor presiona botón “Modificar”, se habilitaran los campos de ingreso de información. c. Si el actor presiona botón “Imprimir”, se imprimirá la página. d. Cuando el sistema “lanza” el mensaje “¿Esta seguro de Dar de Baja este Activo?”. Si el actor presiona “Cancelar”, la operación de baja activo no se realiza.	
Observaciones:	
e. No es necesario que el activo allá finalizado su vida útil para que pueda darse de Baja.	

6.1.1.9 Caso de uso: Registro Información Básica.

ACTOR (ES): *administrador sistema*	**Sistema**

1. Selecciona pantalla “Administración”.	2. Lista opciones de administración del sistema.
3. Selecciona opción “Información básica”.	4. Muestra formulario para el ingreso de usuario y contraseña.
5. Ingresa usuario y contraseña correcta	6. Muestra menú de todos los mantenedores de sistema.
7. selecciona mantenedor.	8. Muestra formulario que lista información ingresada en el sistema.
9. Presiona botón “Agregar Nuevo”.	10. Muestra formulario para el ingreso de información básica al sistema.
11. Ingresa información.	
12. Presiona botón “Enviar”.	13. valida información para posteriormente guardarla en caso que la información sea correctamente ingresada.
Excepciones:	
Observaciones:	
a. El actor también puede modificar información de los mantenedores, cómo también puede eliminar dicha información, eso si, cumpliendo con las reglas que solicita el sistema para realizar estas acciones.	

A continuación se presenta el diagrama de casos de uso para el módulo de mantenciones correspondiente al ciclo actual de desarrollo de software.



Digrama Nº [2]: Modelo de Casos de Uso Módulo Mantenciones.

Los actores presentes son:

- Administrador Sistema.
- Encargado Mantención por Área.

A continuación se detallan los casos de uso para el módulo mantenciones.

6.1.2 Módulo Mantenciones

6.1.2.1 Caso de uso: Ingreso Mantención Teórica.

Actor: *Administrador Sistema*	**Sistema**
1. Selecciona la opción “Activos”.	2. Lista árbol con tipos de activos.
3. Selecciona un tipo de activo.	4. Lista opciones: o Lista Activos PROMO. o Lista Activos AF. o Lista Mantenciones Teóricas.
5. Selecciona opción “Lista Mantenciones Teóricas”.	6. Muestra lista con mantenciones ingresadas en caso que existan, de lo contrario la lista se muestra sin elementos.
7. Presiona botón “Ingreso Nueva Mantención”.	8. Muestra formulario con posible información a ingresar.
9. Ingresa información.	
10. Selecciona modelo para determinar, cual es la frecuencia de las actividades que tendrá la mantención teórica.	
11. Presiona botón “Guardar”.	12. Guarda la información de la mantención teórica y habilita opción “Matriz Plan Mantención”.
13. En el proceso descrito en el punto 9, realiza “Click” en la opción “Habilitar Frecuencia Mantenciones”.	14. Deshabilita opción para seleccionar un modelo.
	15. Deshabilita la opción “Matriz Plan Mantención”. Por lo tanto no se pueden crear planes de mantenciones.
16. Ingresa información.	

17. Presiona botón “Guardar”.	18. Guarda información de la mantención teórica.
Excepciones:	
a. Si el actor presiona botón “Volver”, entonces se visualizará la página anterior. b. Si el actor presiona botón “Eliminar Actividades”, se elimina la actividad seleccionada. c. Si el actor intenta crear una nueva mantención teórica para un modelo determinado, el sistema “lanza”, el siguiente mensaje: “No se puede insertar la mantención porque ya existe una mantención teórica asociada al modelo seleccionado”.	
Observaciones:	

6.1.2.2 Caso de uso: Creación Plan Mantención para Modelo.

Actor: *Administrador Sistema*	**Sistema**
1. Selecciona pantalla mantenciones teóricas según tipo activo.	2. Muestra lista mantenciones teóricas ingresadas.
3. Selecciona mantención teórica.	4. Muestra Información de la mantención teórica.
5. Selecciona opción “Matriz Plan Mantención”.	6. Crea plan de mantención.
	7. Muestra matriz del plan de mantención creado.
8. “Chequea”, las frecuencias en que se realizarán las actividades.	
9. Presiona botón “Guardar”.	10. Guarda la información del plan de mantención.
Excepciones:	
a. Si el actor presiona el botón “Volver”, entonces se visualizará la página anterior.	
Observaciones:	

b. Según el proceso descrito en el punto 6, sólo se creará el plan de mantención si se ha ingresado la frecuencia de días en que se realizarán las actividades creadas en la mantención teórica. La información de la frecuencia en días se ingresa en el mantenedor “Edades Mantención por Modelo”, que se encuentra en la opción “Administración” – “Información Básica”. c. El actor puede modificar el plan de mantención presionando el botón “Modificar”, sólo si no existe un activo asociado a dicha mantención teórica. En caso que no se cumpla dicha condición, puede realizar los cambios que estime conveniente para luego presionar el botón “Guardar”, y así registrar los cambios en el plan de mantención.

6.1.2.3 Caso de uso: Ejecutar Plan de Mantención de un Activo.

Actor: *Encargado Mantención por área*	**Sistema**
1. Selecciona pantalla lista activos, según tipo activo.	2. Lista activos ingresados.
3. Selecciona activo.	4. Muestra formulario con información del activo con campos de ingreso inhabilitados.
5. Presiona botón “Modificar”.	6. Muestra botón “Ejecutar Plan Mantención”.
7. Presiona botón “Ejecutar Plan Mantención”.	8. Crea primera mantención con estado “Planificada”, asignando la fecha de ejecución según indique el plan de mantención asociado al modelo del activo.
Excepciones:	
a. Si el actor presiona el botón “Modificar”, sólo aparecerá el botón “Ejecutar Plan Mantención”, si no existen mantenciones efectivas asociadas al activo, además de esto, no se podrá modificar el modelo del activo. El actor se percatará de este evento cuando el sistema, muestre el siguiente mensaje:” No se puede modificar el modelo del activo, porque tiene asociado mantenciones efectivas”.	
Observaciones:	

a. En la opción “Mantenciones”, se puede visualizar la mantención con estado “Planificada”. b. Sólo se crearan mantenciones efectivas, si el modelo seleccionado para el activo, tiene un plan de mantención teórica asociado a dicho modelo.

6.1.2.4 Caso de uso: Ejecutar Mantenciones Efectivas de un Activo.

ACTOR (ES): *Encargado Mantención por área*	**Sistema**
1. Selecciona pantalla lista activos, según tipo activo.	2. Lista activos ingresados.
3. Selecciona activo.	4. Muestra formulario con información del activo con campos de ingreso inhabilitados.
5. Selecciona opción “Mantenciones”	6. Muestra lista mantenciones efectivas, cuyos estados pueden ser: “Planificada” o “Ejecutada”.
7. Selecciona mantención con estado “Planificada”.	8. Muestra formulario con información y actividades a realizar en la mantención efectiva.
9. Selecciona botón “Modificar”.	
10. Visualiza campos de ingreso deshabilitados.	
11. Ingresa información y selecciona las actividades realizadas en la mantención.	
12. Presiona botón “Guardar”.	13. Guarda la mantención efectiva y cambia el estado de la mantención a “EJECUTADA”.
	14. Muestra botón “Ejecutar Mantención”.
15. Presiona botón “Ejecutar Mantención”.	16. Genera la próxima mantención con estado “PLANIFICADA”, con las actividades y fecha

	definida según plan de mantención.
Excepciones:	
a. Si en la pantalla lista “Mantenciones Efectivas”, el actor selecciona una mantención con estado “EJECUTADA”, visualizará la pantalla “Detalle Mantenciones Efectivas”, con los campos deshabilitados y con la posibilidad de realizar las siguientes tareas: • Presionar botón “Imprimir”. • Presionar botón “Volver”. • Presionar botón “Baja Equipo”.	
Observaciones:	

6.1.2.5 Caso de uso: Ejecutar Mantención NO planificada.

ACTOR (ES): *Encargado Mantención por área*	**Sistema**
1. Selecciona pantalla lista mantenciones efectivas.	2. Lista mantenciones efectivas.
3. Selecciona botón “Agregar Nueva Mantención Efectiva”.	4. Muestra formulario para ingresar detalle de la mantención efectiva.
5. Ingresa información	
6. Presiona botón “Guardar”.	7. Guarda la mantención efectiva con estado ejecutada.
Excepciones:	
Observaciones:	
a. No se puede crear la mantención efectiva no planificada si no se ingresan los siguientes elementos: • Orden de compra • Campo de ingreso Observaciones. b. Las mantenciones no planificadas sólo se crean en casos de emergencia e imprevistos que puede presentar el activo.	

6.1.2.6 Caso de uso: Registro Mantención Calendario Google.

ACTOR (ES): *administrador sistema, Encargado mantención por área.*	**Sistema**
1. Selecciona la opción “Información Básica”, de un activo seleccionado previamente.	2. Muestra formulario con la información del activo seleccionado.
3. Presiona botón “Modificar”.	4. Muestra botón “Ejecutar Plan Mantención”.
5. Presiona botón “Ejecutar Plan Mantención”.	6. Crea resumen de la mantención, en calendario Google del usuario que ejecuto el plan de mantención.
7. De acuerdo a lo descrito en el punto 1, el actor selecciona la opción “Mantenciones”, de un activo previamente seleccionado.	8. Muestra lista de las mantenciones efectivas del activo.
9. Selecciona mantención con estado “Planificada”.	10. Muestra formulario con la información de la mantención.
11. Presiona botón “Modificar”.	
12. Ingresa información de la mantención realizada.	
13. Presiona botón “Guardar”.	
14. Presiona botón “Ejecutar Mantención”.	15. Crea resumen de la mantención, en calendario Google del usuario que ejecuto el plan de mantención.
Excepciones:	
Observaciones:	
a. El actor podrá crear un resumen de la mantención en calendario Google, sólo si existe una conexión Internet.	

A continuación se presenta el diagrama de casos de uso para el módulo de reportes correspondiente al ciclo actual de desarrollo de software.

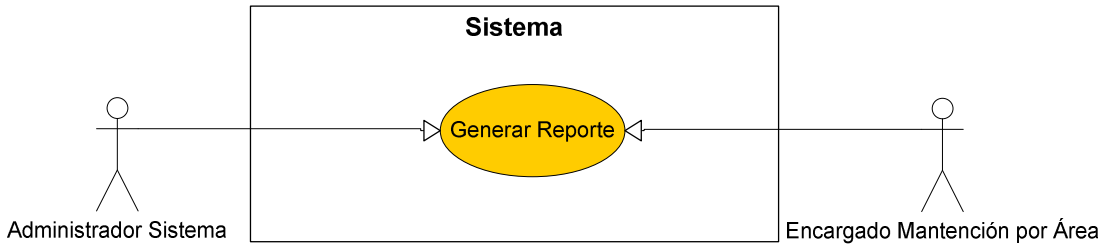


Diagrama Nº [3]: Modelo de Caso de Uso Módulo Reportes.

Los actores presentes son:

- Administrador Sistema.
- Encargado Mantención por Área.

A continuación se detallan los casos de uso para el módulo reportes.

6.1.3 Módulo Reportes.

6.1.3.1 Caso de uso: Generar reporte.

ACTOR (ES): *administrador sistema Encargado Mantención por Área*	**Sistema**
1. Selecciona la opción “Administración”.	2. Lista opciones: o Información Básica. o Configuración. o Activos Eliminados. o Activos Estado Baja. o Reporte Inventario. o Reporte Inatenciones.

3. Selecciona reporte a generar.	4. Muestra opciones para generar reporte.
5. Selecciona datos para filtrar la información a visualizar.	
6. Selecciona botón “Generar Reporte”.	7. Se conecta con servidor de reportes.
	8. Muestra reporte solicitado.
Excepciones:	
Observaciones:	
b. El actor podrá generar reportes de acuerdo a sus privilegios de usuario. El Actor Encargado mantención por área sólo podrá generar reportes limitados a la información concerniente al área que pertenece.	

A continuación se presenta el diagrama de casos de uso transversales para todos los módulos del sistema correspondiente al ciclo actual de desarrollo de software.

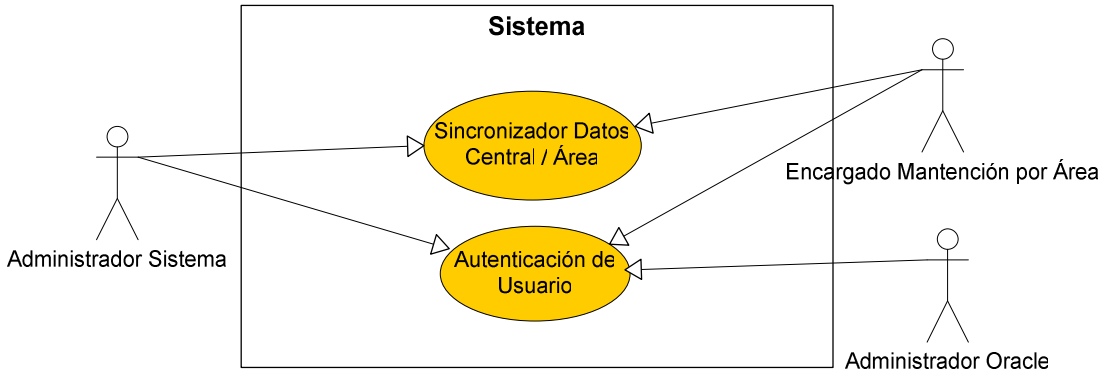


Diagrama Nº [4]: Modelo de Caso de Usos Transversales para todos los módulos del sistema.

Los actores presentes son:

- Administrador Sistema.
- Administrador Oracle.
- Encargado Mantención por Área.

A continuación se detallan los casos de usos transversales para todos los módulos del sistema.

6.1.4 Casos de Uso transversales para todos los módulos del Sistema.

6.1.4.1 Caso de uso: Sincronizador Datos Central / Área.

Actor: administrador sistema, *encargado Mantención por área*	**Sistema**
1. Realiza cambios en la base de datos de la central/área.	2. Marca los datos que sufrieron algún cambio. Por ejemplo: • Datos insertados • Datos modificados • Datos eliminados
	3. De acuerdo al tiempo establecido para la sincronización de las bases de datos, se verifica el estado de los cambios.
	4. Finalmente consolida información actualizada en ambas bases de datos.
Excepciones:	
Observaciones:	
a. En caso que no existiera conexión vía Internet entre la central / área, la información se actualizará sólo una vez reestablecida dicha comunicación	

6.1.4.2 Caso de uso: Autenticación de Usuario.

ACTOR (ES): *administrador sistema, encargado mantención por área, Administrador Oracle*	**Sistema**
1 Ingresa a página de inicio.	2 Solicita ingreso de usuario y contraseña.
3 Ingresa usuario y contraseña correcta.	4 Valida la información ingresada, permite o deniega acceso al sistema.
	5 En caso denegar acceso "lanza", un mensaje: "El Usuario No existe".
Excepciones:	
Observaciones:	
a. El sistema muestra información según perfil o rol del usuario.	

6.1.2 Análisis de Tecnologías a Utilizar y su Aplicación

6.1.2.1 Modelo N-Capas

El modelo N-capas, es la arquitectura predominante para realizar aplicaciones o soluciones multiplaformas en la mayoría de las empresas hoy en día.

Para clarificar que ofrece realizar aplicaciones con esta arquitectura se mencionarán algunas ventajas tales como:

- Desarrollos paralelos entre las capas: Esto es posible dado que las funcionalidades de cada capa se encuentran identificadas y

diferenciadas, permitiendo desarrollar conjuntamente las diferentes capas que componen la solución.

- Mantenimiento más sencillo: Debido a la misma encapsulación de las funcionalidades de la solución, el mantenimiento de cualquier aplicación desarrollada en un ambiente distribuido en n-capas se torna más simple, reduciendo tiempos de actualizaciones y mantenciones de la solución.
- Mayor flexibilidad de la solución: Debido a la distribución de la lógica de la solución, es posible añadir nuevas funcionalidades o módulos a la solución en forma más transparente, sin interferir en los desarrollos ya realizados y así acelerando las nuevas implementaciones.
- Mayor escalabilidad: Una de las principales ventajas de una aplicación distribuida diseñada correctamente es su buena escalabilidad, es decir, que puede manejar muchas peticiones con el mismo rendimiento simplemente añadiendo más hardware. El crecimiento es casi lineal y no es necesario añadir más código para conseguir una buena escalabilidad.

A diferencia de lo que se pudiera pensar, el desarrollo en “n capas” no es un producto o un estándar, se encapsula en un concepto estratégico que ayuda a la construcción y despliegue lógico de un sistema distribuido.

Por último y para conceptualizar la utilización de esta arquitectura en este proyecto de Tesis, se puede concluir que los sistemas de “n capas” subdivididos ayudan a facilitar el desarrollo rápido de aplicaciones y su posterior despliegue, con beneficios incrementales fruto de los esfuerzos del desarrollo en paralelo coordinado, resultando un decremento del tiempo de desarrollo y de sus costos involucrados.

6.1.2.2 Evolución del modelo de N-Capas a Webservices

Tomando como punto de partida lo comentado en el punto 6.1.2.1., las aplicaciones tienden a ser divididas lógicamente, enfocando en cada división un grupo de funcionalidades especificas de cada solución. Lo más común es que la elección sea una división en tres partes (presentación, lógica del negocio y finalmente almacenamiento y comunicación con repositorio de datos), aunque también existen otras posibilidades dependiendo del nivel de encapsulamiento que se pretende imprimir a la solución. A pesar de que las aplicaciones de N niveles surgieron en un principio para solucionar algunos de los problemas relacionados con las aplicaciones cliente/servidor tradicionales, con la llegada del Web esta arquitectura ha pasado a desempeñar un papel predominante en la nueva programación.

La arquitectura en la que se basan los WebServices, es una arquitectura que se maneja en forma distribuida. Dada esta característica, los Webservices logran una vinculación con el modelo de aplicaciones de “n-capas”. Esto se debe a que gran parte de su desarrollo se encuentra en diferentes plataformas y pueden estar en diferentes ubicaciones.

6.1.2.3 WebServices

Los WebServices o Servicios Web surgen ante la necesidad de establecer una estandarización de las comunicaciones entre las distintas plataformas y lenguajes de programación. Dada esta problemática, anteriormente se habían realizado intentos de crear estándares, pero no lograron un éxito adecuado, alguno de ellos como DCOM[5] (Microsoft) o CORBA (ORB) no lograron una penetración significativa ya que dependían directamente de la implementación del vendedor. [Liz02]

Otro problema que surgió era la utilización de RCP[6] para realizar las comunicaciones entre diferentes nodos. Esto debido a problemas de conectividad que se presentaban al momento de intentar establecer comunicaciones entre nodos, ya que la seguridad impuesta por los firewall bloqueaba este tipo de comunicaciones. Es por eso que en 1999 se comenzó a

[6] Arquitectura de Negocios que permite la transferencia de servicios 1 a 1

plantear un nuevo estándar, el cual terminaría utilizando XML, SOAP, WSDL y UDDI.

6.1.2.3.1 Requisitos que debe cumplir un WebServices

Los WebServices deben cumplir un conjunto de requisitos mínimos para ser denominados como tal, los cuales son:

- **Interoperabilidad:** utilización por clientes de distintas plataformas.
- **Amigabilidad con Internet:** soportar a clientes que utilicen los Webservices desde Internet.
- **Interfaces fuertemente tipadas:** No debe existir ambigüedad acerca del tipo de dato enviado y recibido desde un WebService.
- **Posibilidad de aprovechar los estándares de Internet existentes**
- **Soporte para cualquier lenguaje**
- **Soporte para cualquier infraestructura de componente distribuida**
- **Descubrimiento:** la forma que tiene el cliente para “resolver”, donde se encuentra el WebServices, se logra mediante un proceso llamado “descubrimiento” (discovery).
- **Descripción:** implica meta datos estructurados sobre la interfaz que intenta utilizar la aplicación cliente, así como documentación escrita sobre el WebService, incluyendo ejemplos de uso (typelib).

- **Formato del mensaje:** el uso de un mecanismo estándar para codificar los datos asegura que los datos que codifica el cliente los interpreta correctamente el servidor.
- **Codificación**
- **Transporte:** una vez que se ha dado el formato al mensaje y se han serializado los datos en el cuerpo del mensaje, se debe transferir entre el cliente y el servidor utilizando algún protocolo de transporte, por ejemplo TCP/IP.

6.1.2.3.2 Elementos que componen un WebServices

Para la descripción de los componentes de un WebServices se citarán textos textuales de libros que se especificarán en la Bibliografía del documento.

6.1.2.3.2.1 XML (eXtended Markup Languaje)

Es un lenguaje de marcas que ofrece un formato para la descripción de datos estructurados, es decir, XML es un metalenguaje, dado que con él podemos definir un lenguaje propio de presentación y a diferencia de HTML, que es centrado en la representación de la información, XML se centra en la información en si misma. La particularidad más importante de XML es que no posee etiquetas predefinidas con anterioridad, ya que es el propio diseñador el que las va creando de acuerdo a su necesidad, dependiendo directamente del contenido del documento. [Har05]

En el siguiente ejemplo se despliega una tabla con un código típico HTML y una versión equivalente en XML, se puede apreciar que es más interpretable la información que intenta representar la versión en XML.

HTML	XML
<table> <tr> <td>Titulo</td> <td>Autor</td> <td>Precio</td> </tr> <td>La novia</td> <td>Juan Perez</td> <td>10000</td> </tr> <td>Clavel</td> <td>Pedro Reyes</td> <td>15000</td> </tr> <table>	<LIBROS> <libro> <titulo>El novio</titulo> <autor>Erick Nayan</autor> <precio>10000</precio> </libro> <libro> <titulo>Clavel</titulo> <autor>Gabriela</autor> <precio>15000</precio> </libro> </LIBROS>

Tabla Nº [3]: Tabla comparativa gramática HTML y XML

La principal fortaleza del XML es también, actualmente, su debilidad principal. El XML es demasiado genérico para ser utilizado sin definir externamente la sintaxis exacta de un documento. Tal estandarización es más simple de alcanzar dentro de una sola aplicación corporativa, más conseguir el mismo resultado en un contexto más amplio requiere un enorme esfuerzo para llegar a algunas definiciones estándares.

Desgraciadamente, cuando se trata de integrar sistemas de diferentes organizaciones, especialmente de una manera abierta, el XML muestra algunas limitaciones. El XML trata de la descripción de los datos, pero solamente es útil cuando las personas están de acuerdo en la manera en que los datos deben ser descritos.

6.1.2.3.2.2 SOAP

Es un estándar propuesto por Microsoft, IBM y otros al consorcio WWW (W3C) para el intercambio de mensajes entre servicios Web y los consumidores de este servicio.

El protocolo SOAP esta construido sobre XML y solo describe el formato de los mensajes dejando abierta la posibilidad de usar varios transportes, aunque actualmente el transporte usado es HTML. La elección de HTML como transporte se debe a que es el transporte más utilizado por las organizaciones a nivel actual para establecer cualquier tipo de comunicación con su entorno.

El protocolo define un “sobre” en el que se empaqueta el requerimiento donde se especifica el destinatario de la llamada, el nombre del método que se invoca y opcionalmente una serie de parámetros con tipos definidos. La respuesta a

este requerimiento se empaqueta de la misma forma, en un “sobre” que contiene el resultado del método invocado.

La utilidad de SOAP consiste en que con un conjunto de servicios simples se puede implementar aplicaciones que entreguen funcionalidades valiosas mediante la integración de estos servicios básicos o mediante el uso programático de estos. [URL 1].

6.1.2.3.2.3 WSDL (WebService Description Language)

Una de las ideas centrales detrás de los WebServices o Servicios Web es que las aplicaciones futuras estarán conformadas de una colección de servicios habilitados en una red. Mientras exista dos servicios equivalentes que se publiciten a la red de una forma estándar y neutra, en teoría una aplicación podrá seleccionar una de ellas en base a criterios establecidos de antemano como el precio o rendimiento del servicio. Además algunos servicios podrán permitir que fueran replicados entre máquinas, facilitando así que una aplicación que se ejecuta en una máquina mejore el rendimiento al replicar servicios a unidades de almacenamiento locales. [Chr01]

6.1.2.3.2.4 Disco

DISCO (Discovery of WebService), es una especificación creada para ayudar a determinar como pueden encontrar los servicios Web en un servidor. Un archivo DISCO contiene el Identificador de recursos uniforme (URI[7]), de cada servicio Web disponible en ese equipo. El URI apunta normalmente al documento WSDL, que a su vez apunta al servicio Web en cuestión.

6.1.2.3.2.5 UDDI

UDDI[8] es un WebService en línea que puede ser utilizado desde las aplicaciones para descubrir de forma dinámica otros servicios en línea, todos ellos perfectamente integrados en una interfaz XML simple.

6.1.2.3.2.6 SOAP, WSDL, UDDI, y WebServices

SOAP a diferencia de XML-RCP[9], incluye una infraestructura a su alrededor. No es un simple protocolo de comunicaciones entre computadores, sino que además se rodea de términos como WSDL y UDDI, para comprender mejor esta afirmación obsérvese la siguiente figura [5]:

[7] Uniform Resource Identifier
[8] Universal Description Discovery Integration.
[9] eXtensible Markup Language-Remote Call Procedure.

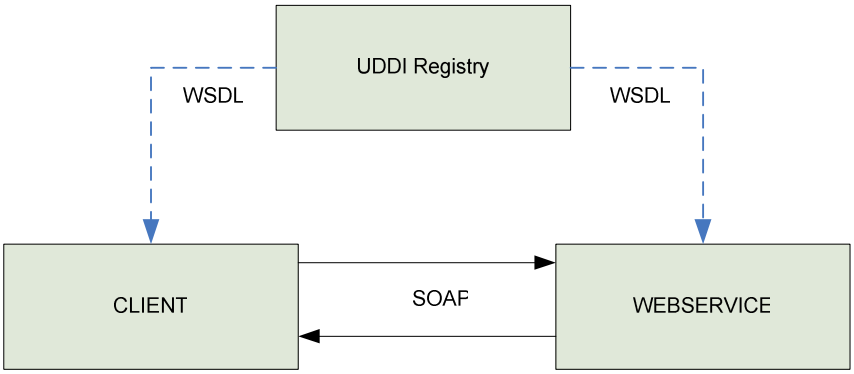


Figura Nº [5]: Vinculación entre UUDI, cliente SOAP y WebService

Las repercusiones que pueden ofrecer los WebServices a los usuarios actuales de Internet son mínimas, ya que los WebServices son herramientas que están directamente involucradas con los programadores de funcionalidades que intentan ofrecer métodos estandarizados a funcionalidades que son comunes a muchos usuarios. [URL 1]

6.1.2.3.2.7 Posibles riesgos de los WebServices

Las expectativas que se presentan frente a esta tecnología son amplias, mas hay que considerar que la aplicación de esta tecnología presenta ciertos riesgos, los cuales son:

- Los WebServices hacen uso de las mismas tecnologías que han sido atacadas en innumerable de ocasiones. Utilizando WebServices la seguridad de una empresa puede verse comprometida debido a la

ausencia de técnicas de seguridad estándar para poder aplicar este tipo de soluciones.

- La calidad de los WebServices es un parámetro que no queda especificado con la suficiente claridad, ya que se puede caer en una expectativa de servicio la cual no cumple con los requisitos preestablecidos.
- Al ser una tecnología en desarrollo, la mayoría de los protocolos que se ven involucrados no han alcanzado un nivel de madurez tecnológico para ofrecer las garantías de seguridad que algunos clientes esperan recibir.

6.1.2.3.2.8 Calidad de los WebServices

Actualmente ya existen en el mercado herramientas específicamente diseñadas para medir la calidad de los WebServices, pero siguen siendo necesaria una estandarización sobre el tema. Los resultados sobre la calidad de los diferentes WebServices servirán como parámetro referencial de comparación y aportarán al cliente de estos servicios a buscar un servicio específico que le ofrezca las garantías necesarias para operar con él. [Dan01]

6.1.2.3.2.9 Estandarización de los WebServices

Los WebServices, como se ha explicado anteriormente, están basado en XML, que ha sido universalmente aceptado en un sin fin de protocolos y sistemas que

operan actualmente. Pero la situación para el resto de los protocoles que se ven involucrados en la tecnologías de los WebServices es distinta, esto debido a que se encuentran en permanente desarrollo y por lo tanto son objetos a ofrecer cierta evolución. Dada esta situación, muchas empresas están a la expectativa de lo que pueda suceder con las determinaciones finales para cada protocolo, más bien esperando una estandarización generalizada para invertir en esta tecnología.

Actualmente, los protocolos que se ven involucrados en esta tecnología se ven trabados en la estandarización debido a las técnicas de seguridad que las envuelven, solo SOAP es considerado por el World Wide Web Consortium (W3C) y se encuentra en tramites de estandarización. SOAP y WSDL son a esta altura ampliamente utilizados pero por el momento UDDI no ha tenido el mismo nivel de penetración de los protocolos anteriormente descritos.

6.1.2.4 Reporting Services

Los reportes del sistema se realizarán con esta herramienta que ofrece un ambiente para almacenar, crear y ver reportes.

Reporting Services (RS) se divide en:

- Servidor de informes. Es el componente principal de RS, El servidor de informes se implementa como un servicio de Microsoft Windows y como

un servicio Web. Se encarga de generar los reportes a través de los servicios Web y de la seguridad de los mismos.

- Administrador de informes. Es una herramienta de administración de los informes. Se accede a ésta a través del explorador Web (IE), que lleva por debajo un portal Share Point o desde el MSSS Management Studio. A través de éste podemos asignar permisos, crear carpetas, ver informes, crear nuevos informes con Report Builder, crear suscripciones (o instantáneas, etc).
- Base de datos del servidor de informes. Es una base de datos SQL Server 2005 donde almacena toda la información que tiene que ver con los informes, con la seguridad, suscripciones, instantáneas y demás extensiones del mismo.
- Herramienta de configuración de Reporting Services. Esta es la encargada de configurar el servidor de informes, entre ellos, el estado del servidor, Directorio virtual del servidor de informes y administrador de informes, Identidad del servicio de Windows, identidad del servicio Web, Instalación de base de datos, claves de cifrado, Inicialización, Configuración de correo electrónico y cuenta de ejecución.

Finalmente el completo análisis de la base de datos se encuentra en el capítulo diseño de la base de datos.

6.1.2.5 Réplica de Base de Datos

La réplica es un conjunto de tecnologías destinadas a la copia y distribución de datos y objetos desde una base de datos a otra, para luego sincronizar ambas bases de datos y mantener su coherencia. La réplica permite distribuir datos a diferentes ubicaciones y a usuarios remotos o móviles mediante redes locales y de área extensa, conexiones de acceso telefónico, conexiones inalámbricas e Internet.

El mecanismo de replicación debe al menos proveer las siguientes características:

- Independencia: Permitir que cada servidor almacene y mantenga su propia base de datos facilita el acceso inmediato y eficaz de sus datos que se usan más frecuentemente.
- Consistencia o coherencia transaccional: La réplica permite validar que la información emitida por un servidor coincide con la información recibida por el servidor destino.
- Tolerancia a fallas: se pueden producir conflictos, y cuando ocurren, debe poder detectarlos y resolverlos.
- Monitorización: Permitir supervisar el rendimiento de la réplica, estableciendo advertencias y medidas de rendimiento.

6.1.2.5.1 Réplica en SQL Server 2005.

La replica en SQL Server 2005 se basa en un modelo que permite conexiones tanto rápidas como lentas soportadas tanto por Wan's y Lan's. Con la duplicación se puede distribuir automáticamente copias de transacciones de datos desde un servidor fuente a uno o varios servidores destinos o a una o varias localizaciones remotas. La duplicación en SQL Server 2005 está basada en el Log de transacciones: las transacciones son marcadas como duplicación, estas leen el Log de transacciones de la base de datos fuente y aplican lo ocurrido en la base de datos destino.

SQL Server 2005 utiliza una metáfora de tipo editorial (Diario, Revista), para representar los componentes de una topología de réplica, que incluyen el publicador, el distribuidor, los suscriptores, las publicaciones, los artículos y las suscripciones.

Una topología de réplica define la relación entre los servidores y las copias de los datos, y aclara la lógica que determina cómo fluyen los datos entre los servidores. Hay varios procesos de réplica (denominados *agentes*) que son responsables de copiar y mover los datos entre el publicador y los suscriptores.
A continuación se visualiza un diagrama con los componentes y procesos que participan en la réplica.

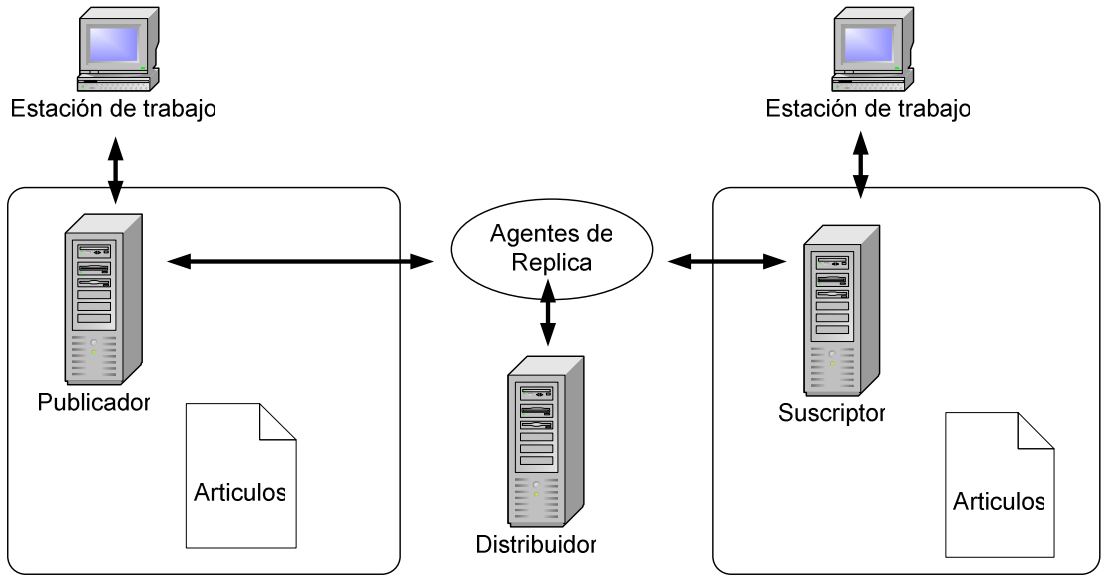


Diagrama Nº [5]: Proceso de la réplica.

Descripción de cada uno de los componentes de la réplica

- **Publicador:** Es una instancia de base de datos que permite que los datos estén disponibles para otras ubicaciones a través de la réplica. El publicador puede tener una o más publicaciones, cada una de las cuales representa un conjunto de objetos y datos relacionados lógicamente para replicar.

- **Distribuidor:** Es una instancia de base de datos que funciona como almacén para datos específicos de réplica asociados con uno o más publicadores. Cada publicador está asociado con una sola base de datos (conocida como la base de datos de distribución) en el distribuidor. La

base de datos de distribución almacena los datos de estado de la réplica, metadatos acerca de la publicación y, en algunos casos, funciona como cola para los datos que se transfieren del publicador a los suscriptores. En muchos casos, una sola instancia de servidor de bases de datos funciona como publicador y como distribuidor. Esto se conoce como un distribuidor local. Cuando el publicador y el distribuidor se configuran en instancias distintas del servidor de bases de datos, el distribuidor se denomina un distribuidor remoto.

- **Suscriptores:** Es una instancia de base de datos que recibe datos replicados. Un suscriptor puede recibir datos de varios publicadores y publicaciones. En función del tipo de réplica elegida, el suscriptor también puede devolver los datos modificados al publicador o volver a publicar los datos en otros suscriptores.
- **Articulo:** Identifica un objeto de base de datos incluido en una publicación. Una publicación puede contener diferentes tipos de artículos, como tablas, vistas, procedimientos almacenados y otros objetos. Cuando las tablas se publican como artículos, se pueden usar filtros para restringir las columnas y filas de datos que se envían a los suscriptores.
- **Publicación:** Es un conjunto de uno o más artículos de una base de datos. La agrupación de varios artículos en una publicación permite

especificar más fácilmente un conjunto de objetos y datos de bases de datos relacionados lógicamente, que se replican como una unidad.

- **Suscripción:** Es una solicitud de una copia de una publicación que se entrega a un suscriptor. La suscripción define qué publicación se recibirá, dónde y cuándo. Hay dos tipos de suscripciones: de inserción y de extracción.

6.2 Diseño del Sistema

En esta etapa de la iteración, XP enfatiza que no se puede establecer una definición temprana de una arquitectura estable para el sistema, si no que dicha arquitectura se asume evolutiva y que los posibles inconvenientes que se generarían por no contar con ella explícitamente en el comienzo del proyecto se solventan con la existencia de una “Metáfora del Sistema”, que es equivalente a la arquitectura que soportará la aplicación en un comienzo y que servirá como vocabulario para hablar sobre el dominio del problema y que posteriormente ayudará a la nomenclatura de clases y métodos del sistema. Todo esto destacando el diseño simple y siguiendo estas características:

- Superar con éxito todas las pruebas.
- No tener lógica duplicada.
- Reflejar claramente la intención de implementación de los programadores.
- Tener el menor número posible de clases y métodos.

Evidentemente, que el diseño de una aplicación de software tiende al cambio y, por ende sufre constantes cambios y reestructuraciones con el objetivo de adaptarse de mejor forma a los nuevos requerimientos. Por esto la representación de las funcionalidades del sistema, mediante distintos diagramas

flexibiliza la mantención de un diseño apropiado, permitiendo además corregir los problemas con mayor facilidad.

A continuación se presentarán una serie de diagramas o modelos que identificarán los procesos que son necesarios para el funcionamiento de la aplicación, así como los procesos que permiten la funcionalidad del sistema, así como también se abordará el completo diseño de la base de datos en el capítulo número 7 del presente documento.

6.2.1 Diagramas de Actividad

Los diagramas que se presentarán a continuación pretenden graficar los procesos considerados más complejos de implementar, representados con una visión simplificada de lo que ocurre durante la operación o proceso, Esto porque se considera innecesario mostrar todos los procesos que componen el sistema.

6.2.1.1 Diagrama de actividad: ingreso información básica.

Muestra el proceso necesario para que el usuario ingrese la información a las tablas “maestras”, que componen la base de datos que soporta al sistema, permitiendo el funcionamiento y buen desempeño de dicho sistema. Cabe

señalar que este diagrama no se presenta por lo complejo de seguir, si no, por que es parte fundamental e integral para que el sistema ofrezca la prestación necesaria para el usuario final.

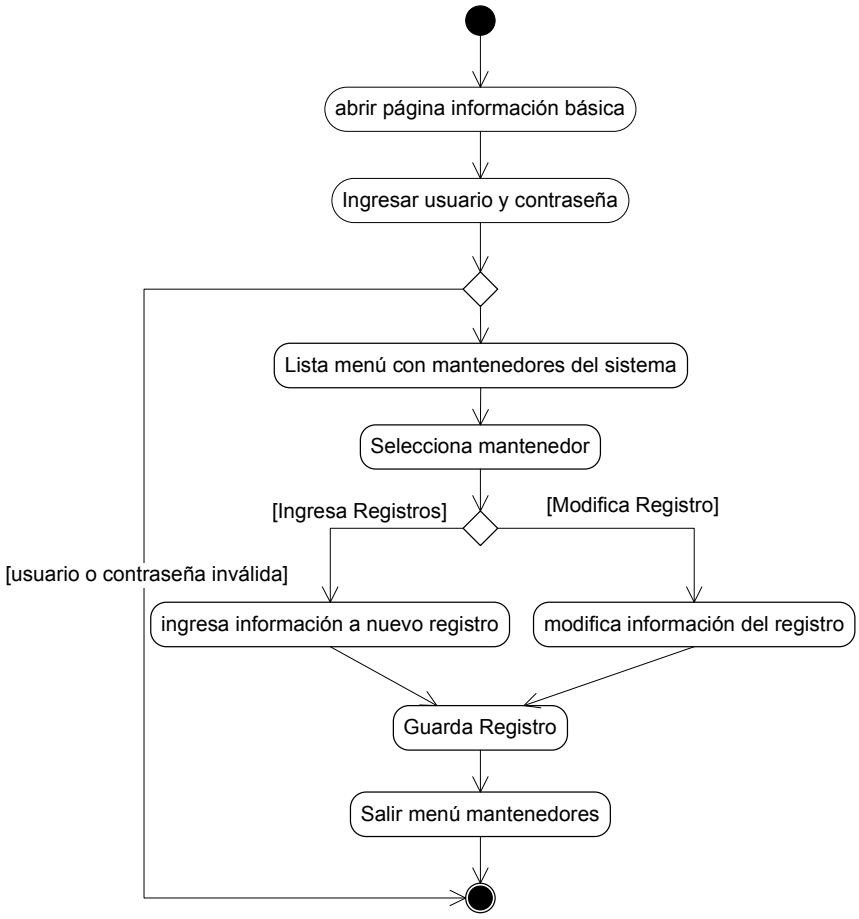


Diagrama Nº [6]: Ingreso información básica.

6.2.1.2 Diagrama de actividad: búsqueda activo en Oracle

Proceso que describe la búsqueda de información de un activo en los sistemas contables que la compañía utiliza para almacenar actualmente dicha información, cabe mencionar que el motor de bases de datos que soporta los sistemas de la compañía, es “Oracle 10.2.0.3 Enterprise Edition”.

Algunas consideraciones para el siguiente diagrama de actividad:

- OC: Orden de compra con la cual se compra uno o más activos.
- MP2: sistema perteneciente a la compañía, en donde almacena los activos que se encuentran aún sin una asignación formal de un número o código de activo.
- Oracle Financial: sistema en donde se almacenan los activos de los distintos departamentos que posee la compañía.

Las búsquedas de activos, según orden de compras y que se encuentran en otros sistemas, se realizarán diseñando e implementando consultas en lenguaje PL/SQL, nativo de ORACLE.

Cabe señalar que estas consultas se realizarán bajo supervisión y posterior aprobación de las personas pertenecientes al departamento de TI de la compañía y serán las encargadas de brindar soporte por parte de la compañía para que el proyecto se lleve a cabo sin inconvenientes.

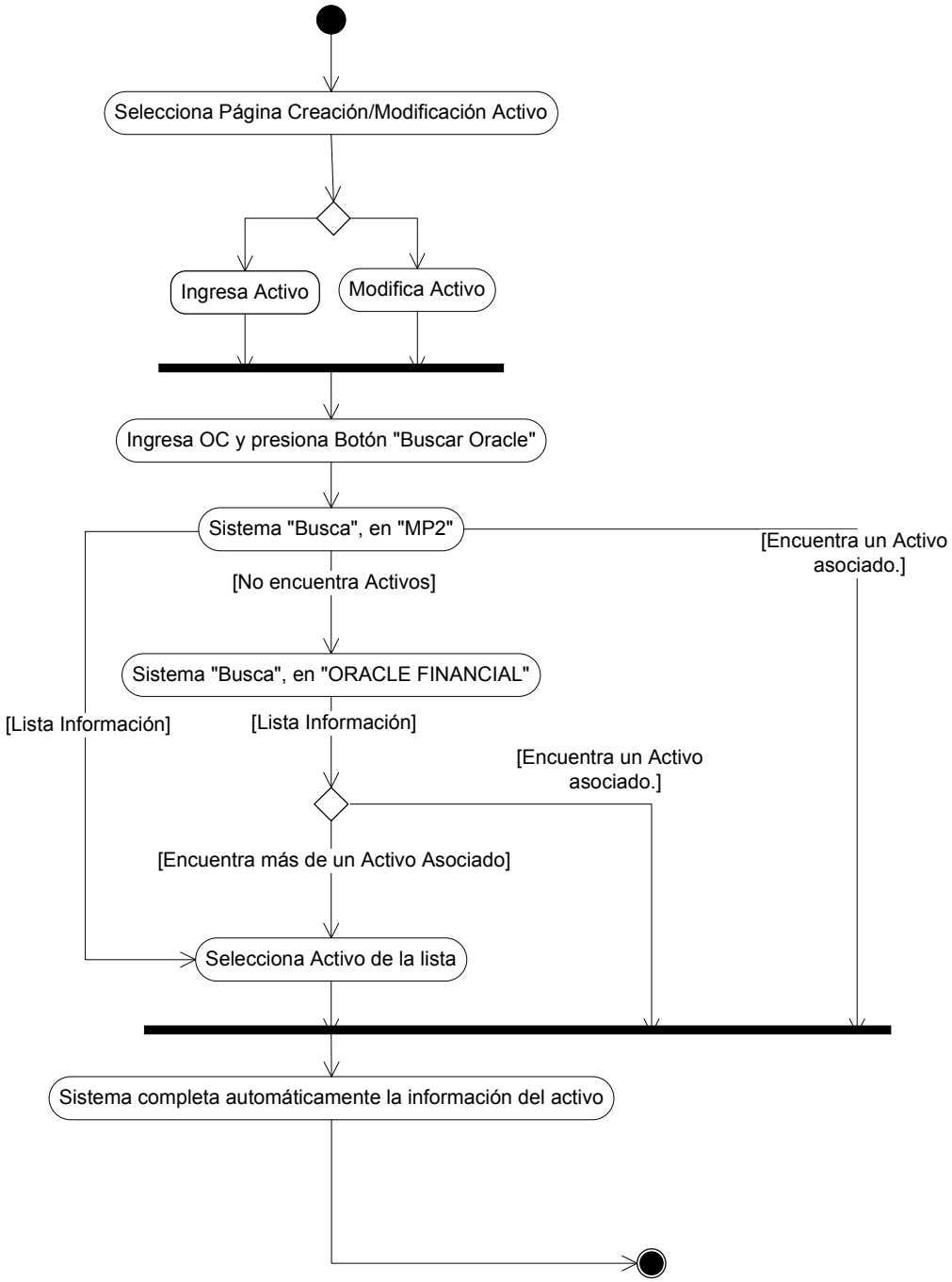


Diagrama Nº [7]: Búsqueda activo en Oracle.

6.2.1.3 Diagrama de actividad: item’s Activo

Proceso que representa el ingreso de ítem’s de activos del tipo activo “línea de fondeo”, entiéndase por ítem de activo a una parte integral del activo. En la siguiente Figura [3], se visualizan dichas partes del activo.

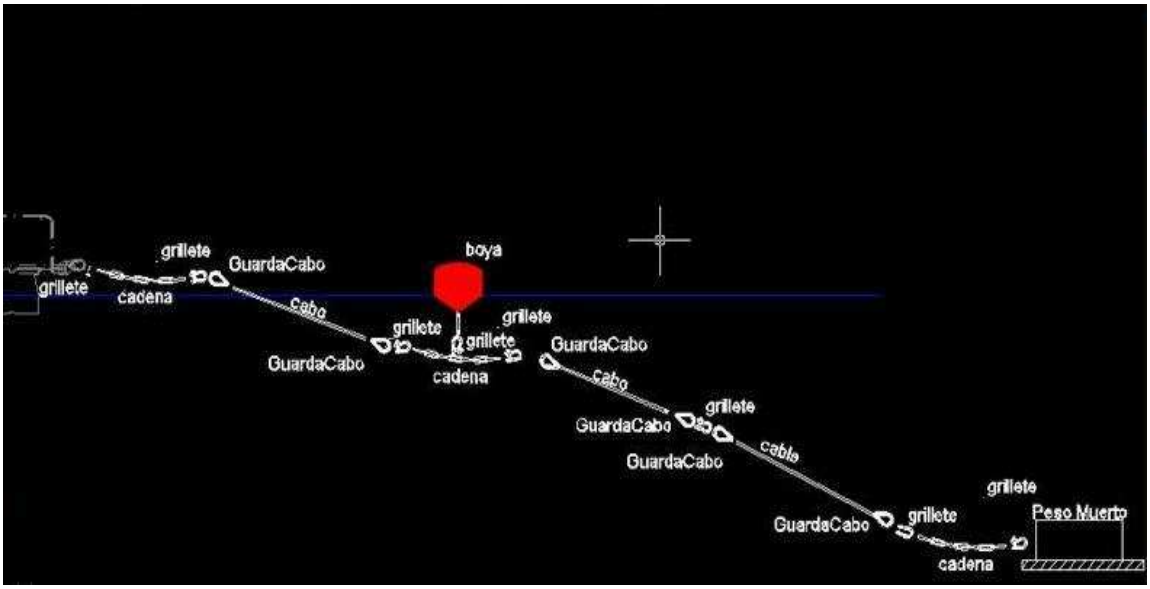


Figura Nº [6]: Item’s de activo.

Los componentes o ítem de una línea de fondeo son las siguientes: Cadena, Cabo, Cable, Boya, Muerto, Grillete, Grillete Boya, Guarda Cabo.

Cabe señalar que las líneas de fondeo son los únicos activos que poseen ítem de activos. A continuación se presenta el diagrama de actividad.

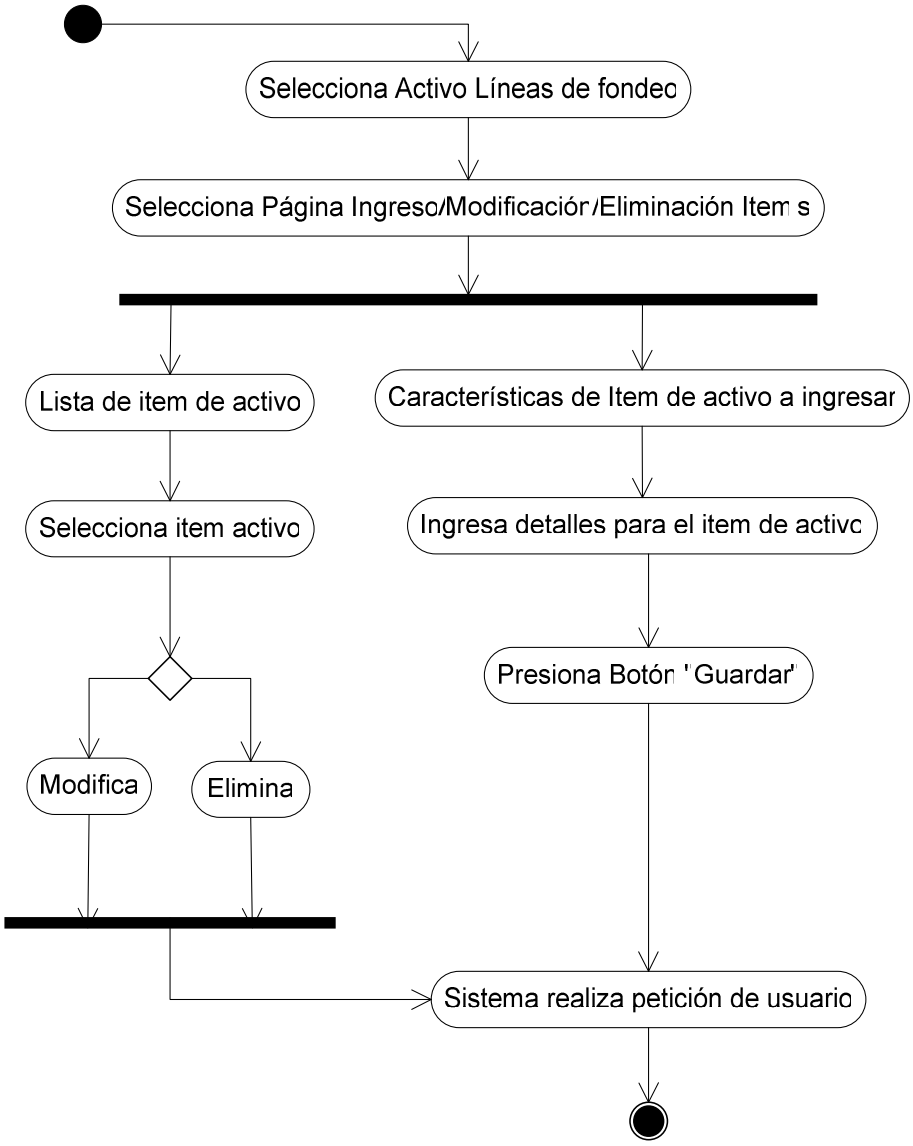


Diagrama Nº [8]: Item's de activo.

6.2.1.4 Diagrama de actividad: ingreso mantención teórica

Proceso que describe el ingreso de las mantenciones teóricas, según tipo de activo seleccionado previamente por el usuario administrador o encargado del área en que realizarán las mantenciones. Las mantenciones teóricas establecen las actividades que se deben realizar a un conjunto de activos que tienen características similares y que se encuentran definidos para un tipo de activo.

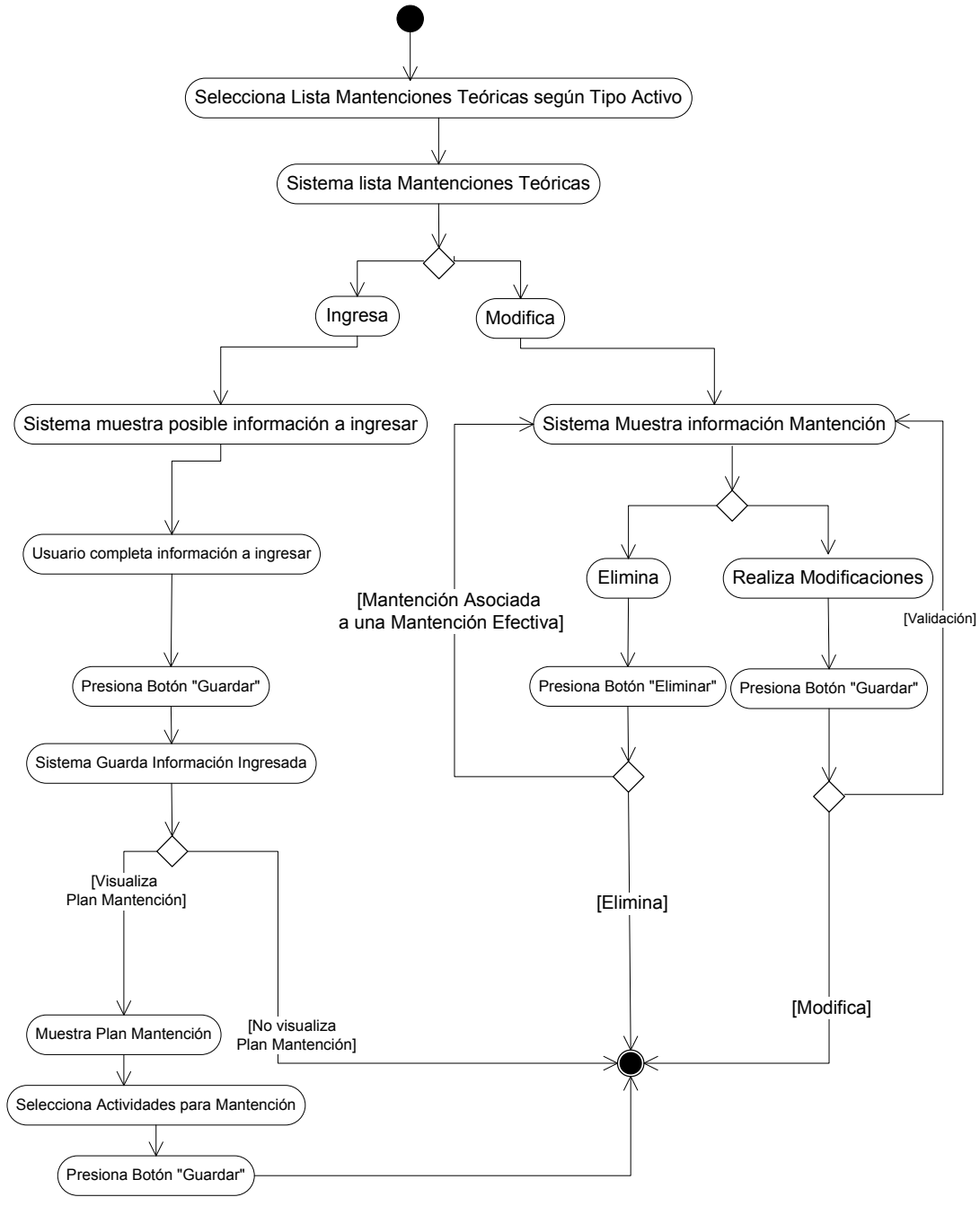


Diagrama Nº [9]: Ingreso Mantención Teórica.

6.2.1.5 Diagrama de actividad: ejecutar plan de mantención de un activo

Proceso que muestra las actividades necesarias para ejecutar los planes de mantenciones teóricas establecidas previamente para los activos.

Cabe mencionar que los planes de mantenciones teóricas se establecen agrupados en tipos de activos, por lo tanto las mantenciones establecidas para cierto tipos de activos, sólo podrán ser asociados a los activos que pertenezcan a dichos tipos de activos.

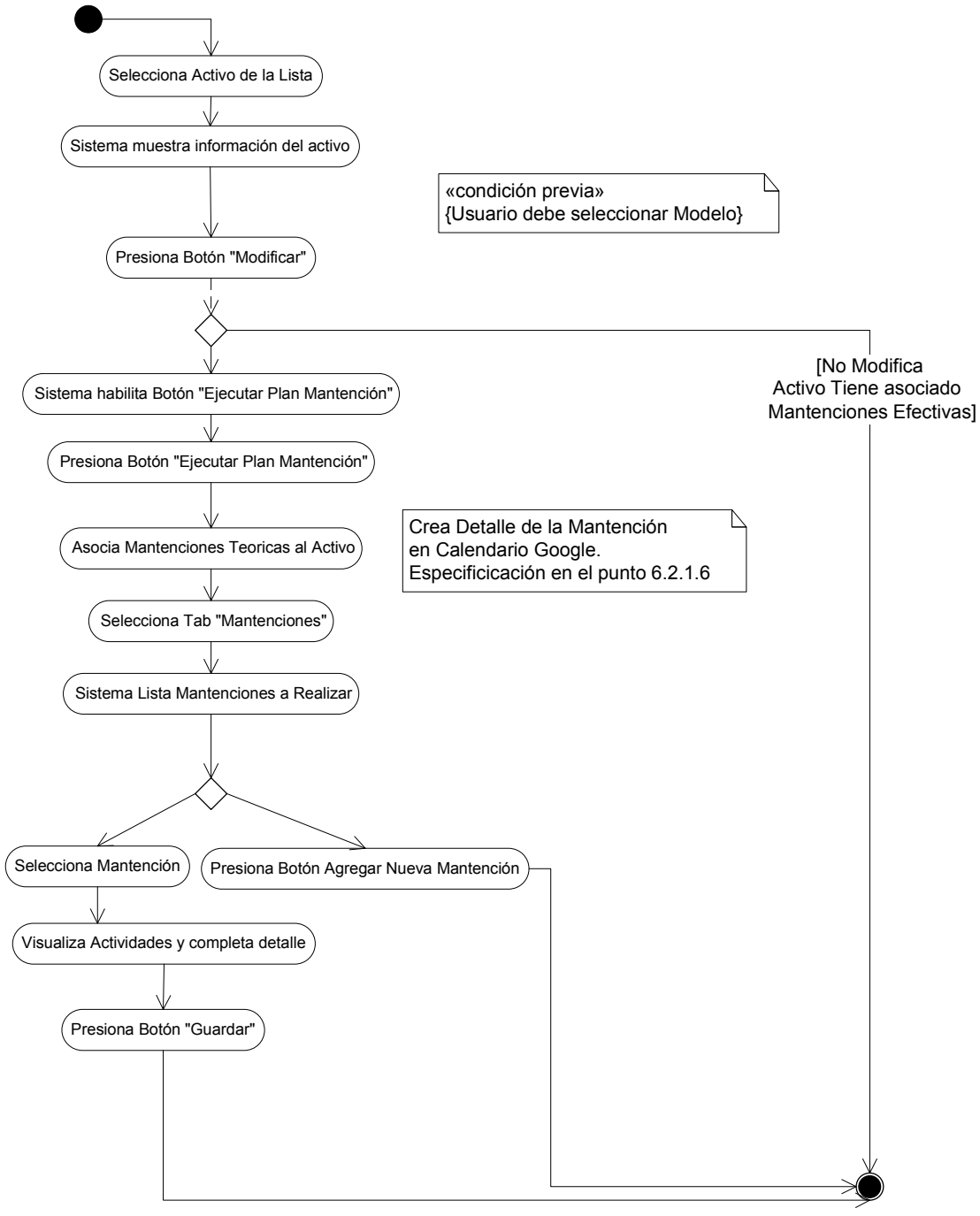


Diagrama Nº [10]: Ejecutar Plan de mantención de un activo.

6.2.1.6 Diagrama de actividad: registro mantención calendario Google

Proceso que grafica el registro de las mantenciones en el calendario Gráfico que ofrece el Buscador Google.

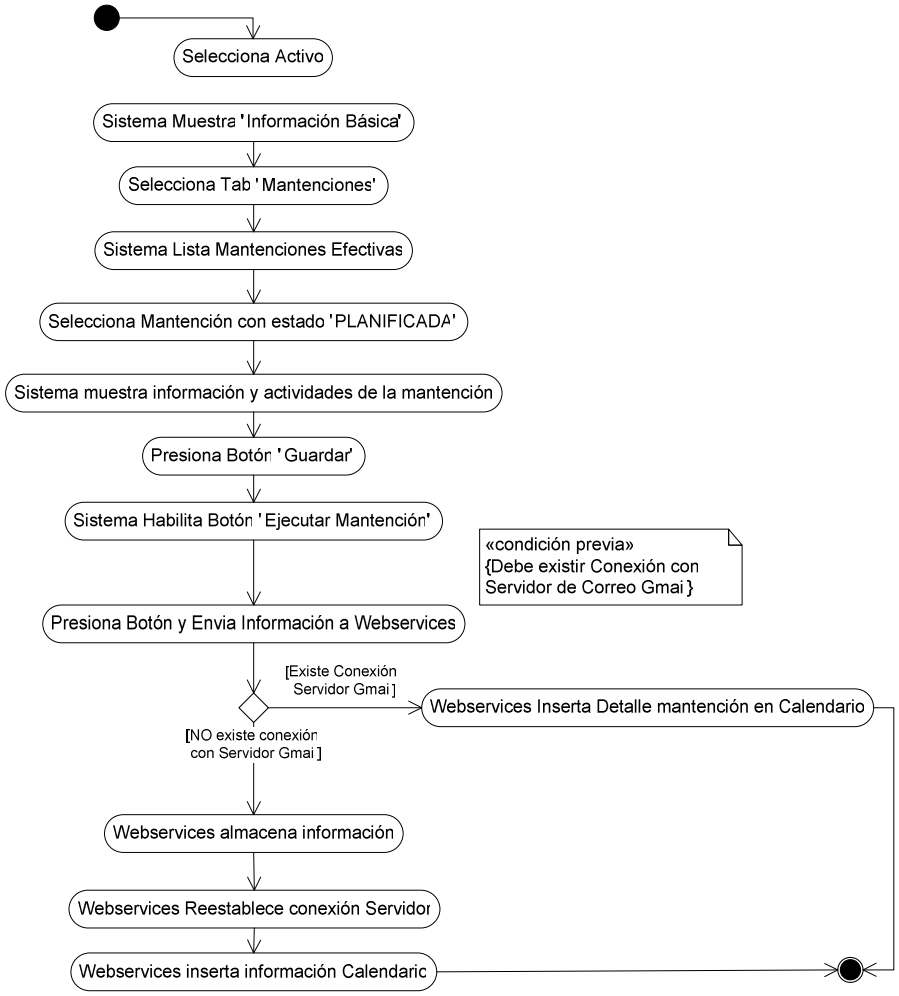


Diagrama Nº [11]: Registro Mantención Calendario Google.

6.2.1.7 Diagrama de actividad: reportes

Proceso que muestra las actividades necesarias para la generación de los reportes de información consolidada que ofrece el sistema PROMO.

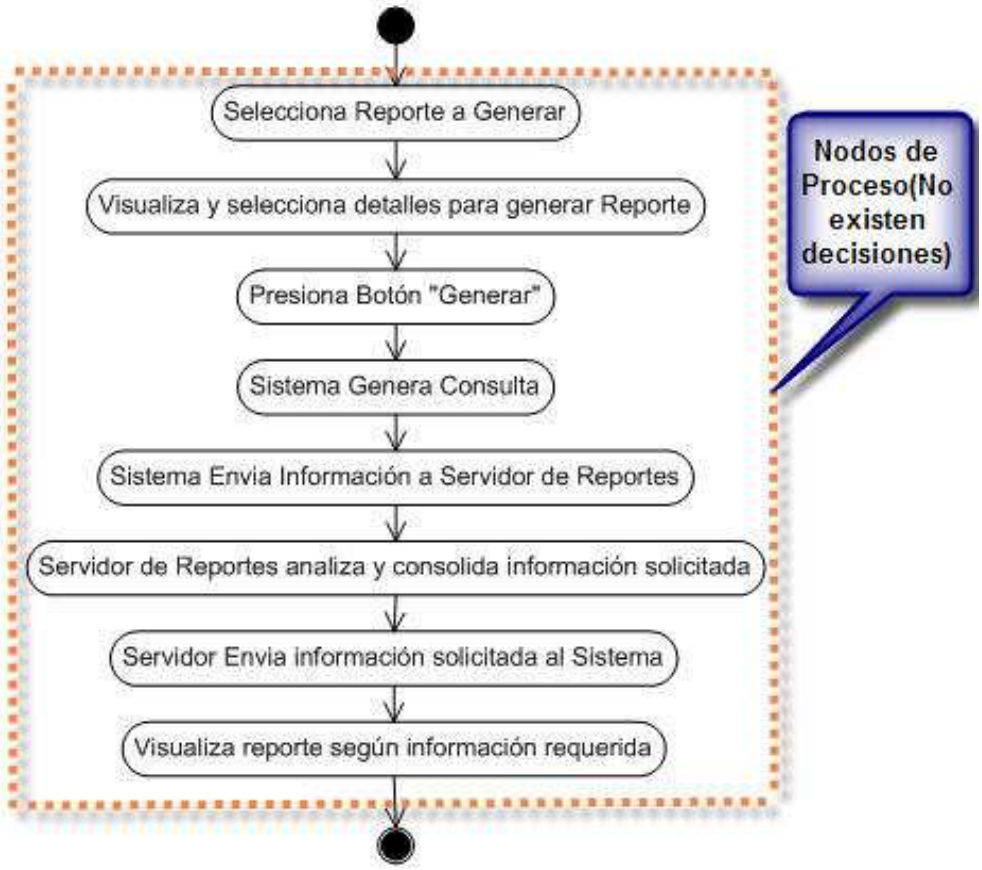


Diagrama Nº [12]: Reportes.

6.2.1.8 Diagrama de actividad: sincronizador datos central / área

Proceso que muestra los recursos necesarios para realizar la sincronización de información, entre la aplicación central y los distintos centros de cultivo que tiene la compañía.

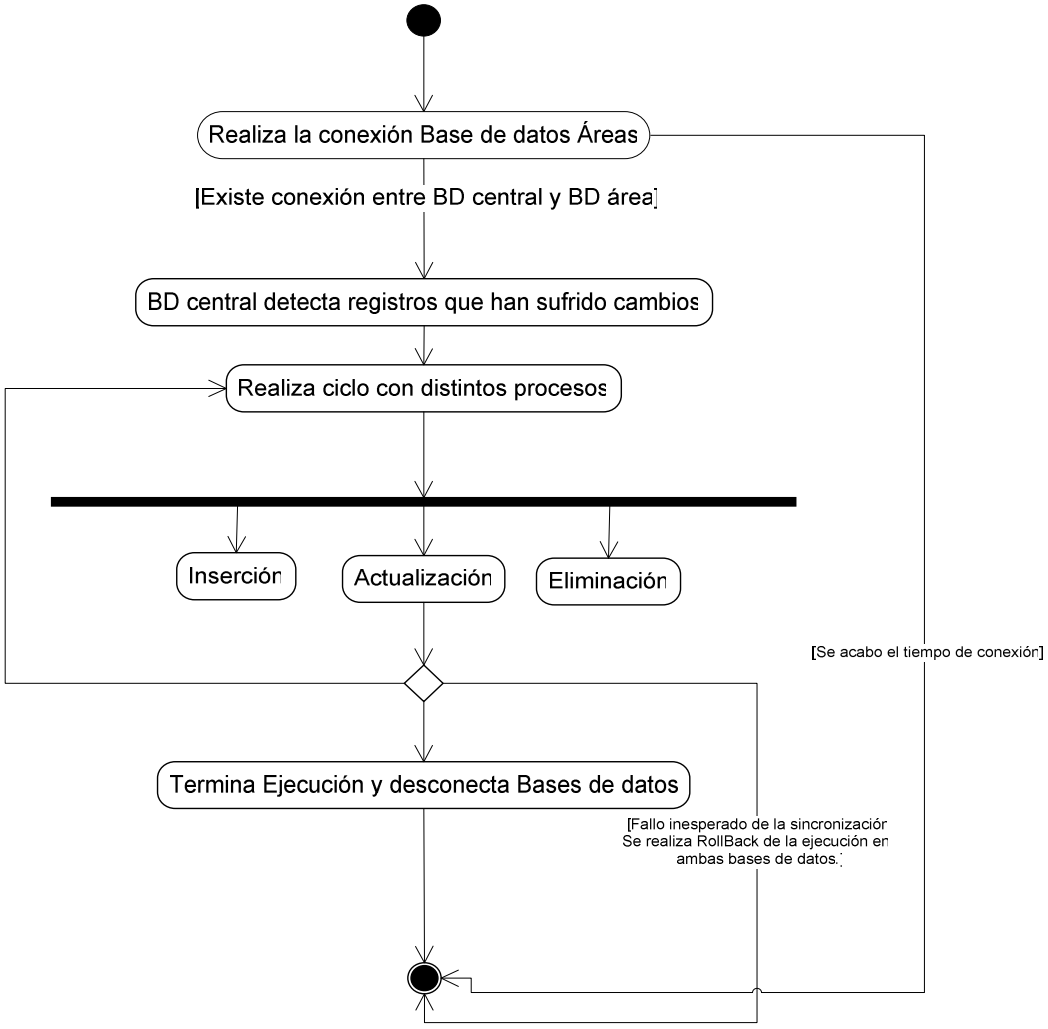


Diagrama Nº [13]: sincronizador datos central / área.

6.2.2 Diagrama Modelo de Clases

Basándose en el paradigma de la orientación a objeto en el cual se desarrolló el sistema, el diagrama de clases otorga una representación estática para desarrollar e implementar una aplicación. Con esta representación del sistema, se pretenden establecer los siguientes puntos:

- Los tipos de objetos del sistema.
- Las relaciones estáticas que existen entre ellos.
- Los atributos y operaciones de las clases.
- Las restricciones a las clases y sus asociaciones.

A continuación se presenta el modelo de clases para el módulo Activos.

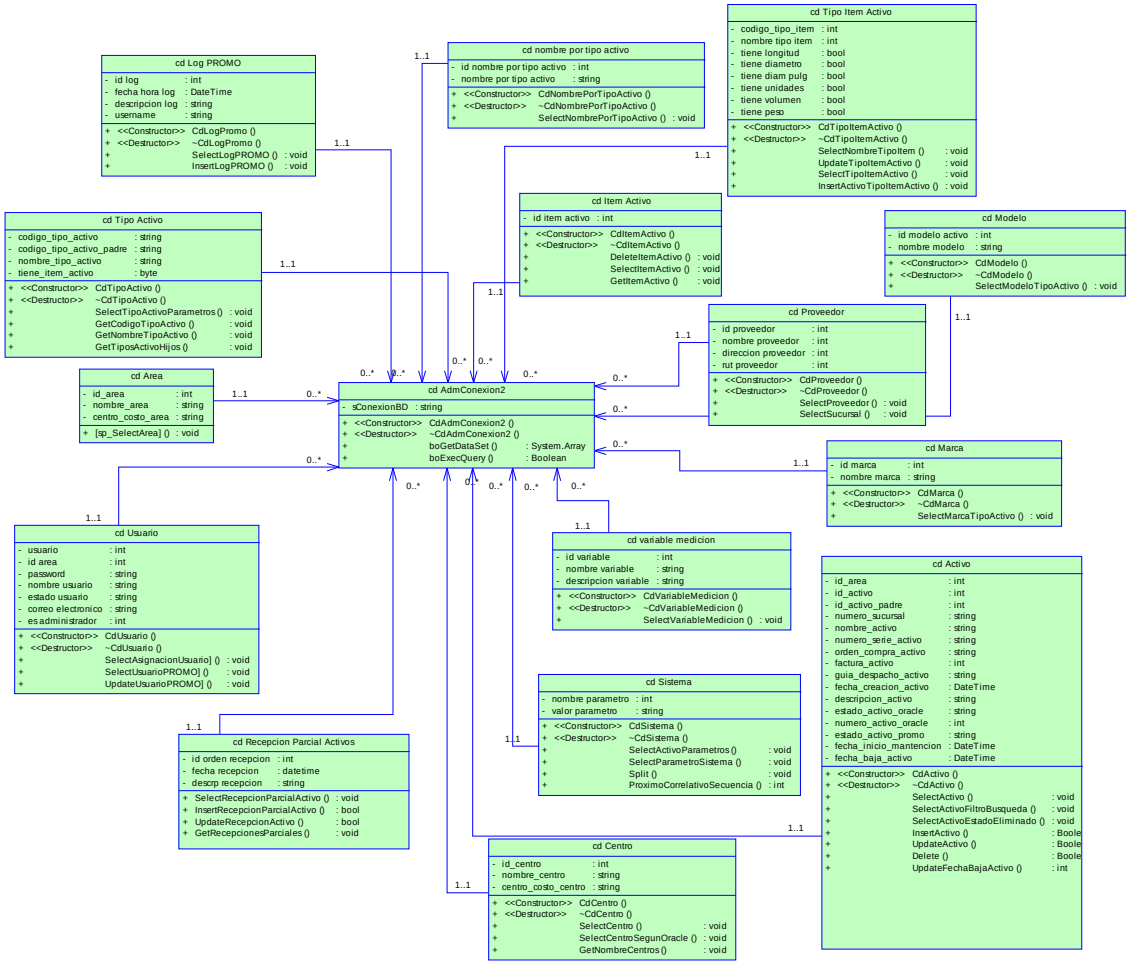


Diagrama Nº [14]: Modelo de Clases

A continuación se presenta el modelo de clases para el módulo Mantención.

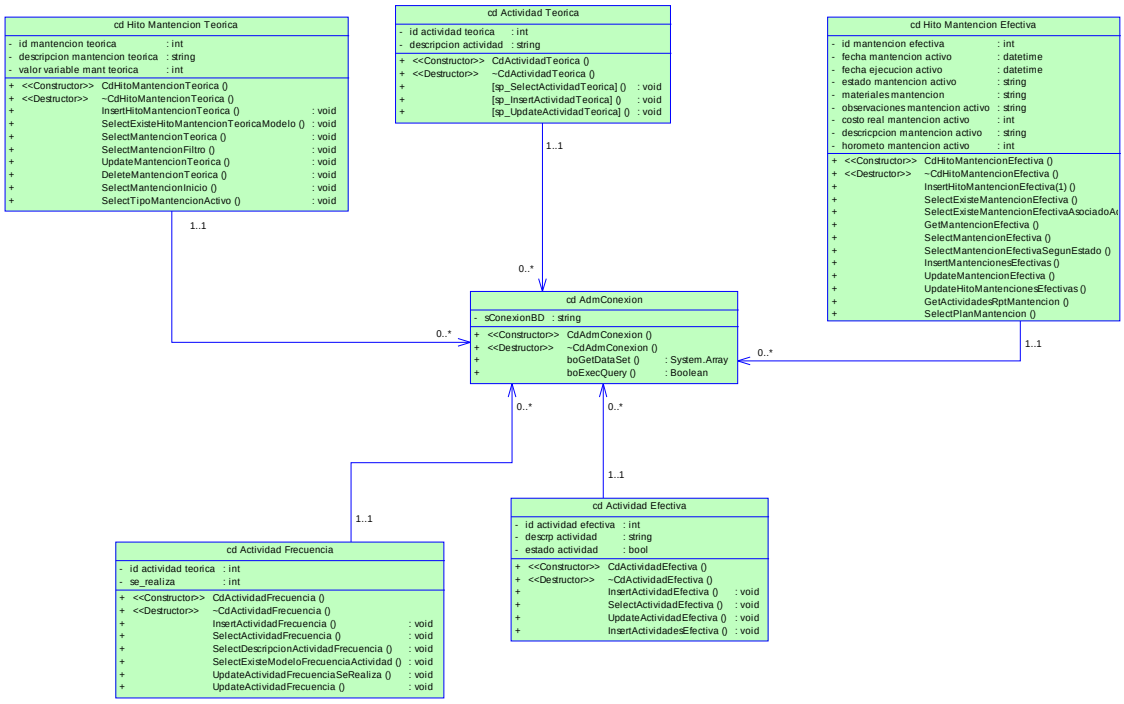


Diagrama Nº [15]: Modelo de Clases

6.2.3 Diseño Sitio Web

La importancia del diseño en la interacción interfaz – usuario es fundamental para lograr los objetivos e ideales trazados para el proyecto. Por esto se debe centrar un gran esfuerzo para prevalecer las necesidades del usuario final del sitio Web.

Cabe destacar que pretender diseñar un sitio amigable, intuitivo y de fácil manejo, es un poco complejo por lo ambiguo de algunos de los puntos mencionados, ya que no se contempla realizar encuestas de satisfacción a los usuarios. Pero sí, se pretende mostrar prototipos de posibles pantallas, donde los usuarios podrán realizar objeciones y comentar sus opiniones en cuanto a colores, estilos etc.

Para lograr el máximo de navegabilidad, manejo y fácil accesibilidad, se realizarán algunas especificaciones tales como:

- Especificación de las pantallas a utilizar.
- Estandarizar los nombres de las páginas Web.
- Especificar una jerarquía de archivos.
- Estandarización de pantallas.

6.2.3.1 Especificación de pantallas

Se mostrara un resumen de la especificación de pantallas en el siguiente punto.

6.2.3.2 Estandarización de nombres de página

Esta actividad busca proporcionar nombres significativos y que se relacionen someramente con los contenidos de las mismas páginas. Esto conlleva a una mejor estructuración del sitio Web.

A continuación se visualizará una tabla con las páginas con sus respectivos nombres de archivo.

Estandarización de Nombres de Pantallas PROMO		
ID	**Nombre**	**Nombre de Archivo**
1	Formulario lista características de los activos, según tipo de activo.	Lista_Activos.aspx
2	Formulario lista características de los activos con estado “BAJA”, según tipo de activo.	Lista_ActivosBaja.aspx
3	Formulario lista características de los activos con estado “Eliminado”, según tipo de activo.	Lista_ActivosEliminados.aspx
4	Formulario lista características de los activos que corresponden al tipo de activo “Líneas de Fondeo”.	Lista_ActivosLineasFondeo.aspx
5	Formulario lista características de los activos, que se encuentran en el sistema “Oracle financial”, pero no el sistema PROMO.	Lista_ActivosOracle.aspx

6	Formulario lista características de los activos que tienen recepciones parciales, según tipo de activo.	ListaRecepcionParcialActivo.aspx
7	Formulario lista características de las mantenciones teóricas según tipo de activo.	ListaTipoMantencionActivo.aspx
8	Formulario que lista las características de los activos, que han sufrido algún traslado de área o centro de cultivo.	LogPROMO.aspx
9	Formulario que muestra un conjunto de opciones o menú para visualizar páginas que utilizaran con frecuencia los administradores del sistema	Menu_Administracion.aspx
10	Formulario que muestra la página principal del sistema.	Menu_Principal.aspx
	Formulario que visualiza el árbol con los distintos departamentos del área de cultivos (farming).	Menu_Activos.aspx
11	Formulario que muestra las distintas opciones y filtros para generar el reporte de inventario.	Reporte_Inventario.aspx
12	Formulario que permite Crear/ Modificar/ Eliminar/ Asociar planes de mantención según tipo de activo.	ver_ActivosInfoBasica.aspx
13	Formulario que permite realizar cambios de contraseñas a los usuarios del sistema.	ver_CuentaAdmin.aspx
14	Formulario detalle mantenciones efectivas.	ver_DetalleMantEfectivas.aspx
15	Formulario detalle mantenciones teóricas según tipo de activo.	ver_MantencionActivo.aspx
16	Formulario lista mantenciones efectivas.	Ver_MantencionEfectiva.aspx
17	Formulario que permite generar mantenciones reactivas.	ver_MantEfectivaNueva.aspx
18	Formulario de ingreso recepciones parciales de activo.	ver_OrdRecepActivoNuevo.aspx
19	Formulario de modificación de recepciones parciales de activo.	ver_OrdRecepInfoActivo.aspx
20	Formulario que presenta la matriz de actividades a realizar en una	ver_PlanMantencionTeorica.aspx

	mantención efectiva de activo.	
21	Formulario de ingreso/modificación/eliminación de Ítems de activo.	ver_ItemsActivosInfoBasica.aspx
22	Formulario de ingreso del sistema PROMO.	index.aspx

Tabla Nº [4]: Estandarización de nombres de página PROMO.

Las páginas que tienen por finalidad “alimentar”, al sistema (Información básica), se dividirán en:

- Lista información.
- Agregar/ Editar/ Eliminar información.

Estas páginas tienen un control de acceso, que se visualiza en la siguiente figura [7]:

Figura Nº [7]: Login Maestros PROMO.

El conjunto de pantallas que agrupa las páginas maestras del sistema se visualizan a continuación:

Figura Nº [8]: Maestros PROMO.

6.2.3.3 Jerarquía de archivos

La estructuración de las diferentes páginas, permite representar los contenidos del sitio en un orden jerárquico y de fácil visualización.

La jerarquía de archivos se visualizara mediante un mapa de navegación. Cabe señalar que los maestros del sistema se visualizarán como List xxxxx.aspx y Agregar / Modificar / Eliminar.aspx para generalizar, ya que tienen la misma secuencia de navegación guiada por la figura [9].

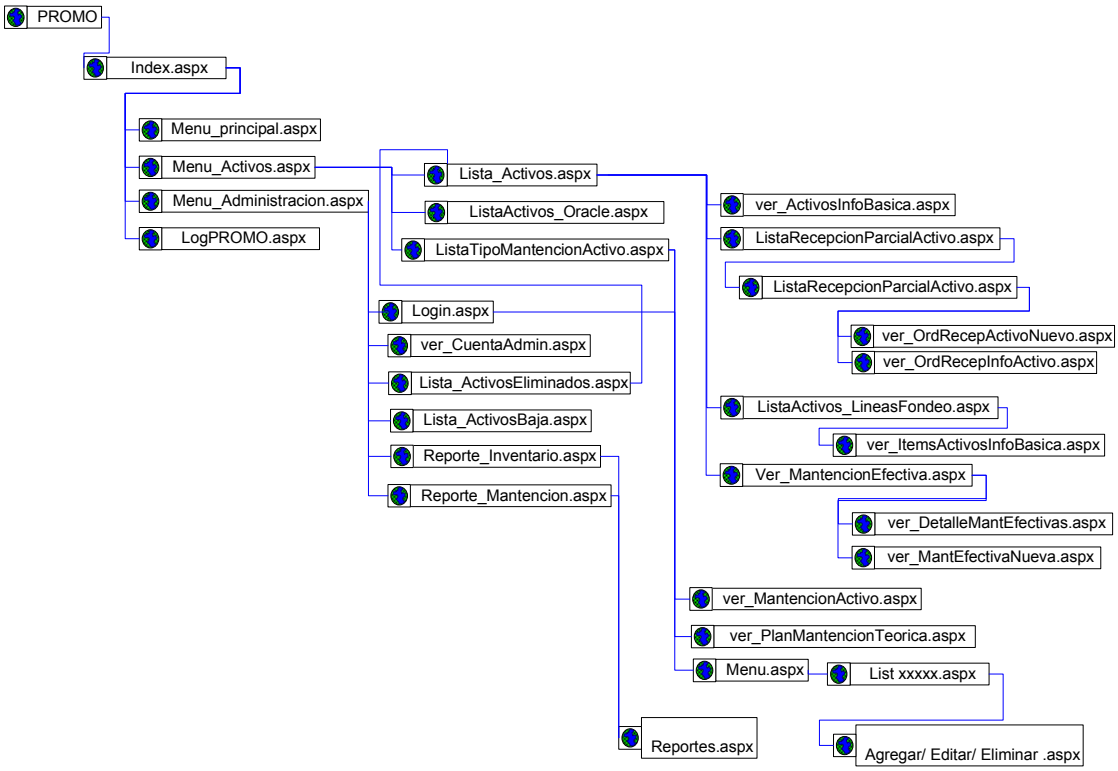


Figura Nº [9]: Mapa del Sitio Web.

6.2.3.4 Estandarización de pantallas

Con la presentación de prototipos a los usuarios finales se logró estandarizar la estructura de las diferentes interfaces que componen el sistema y como éstas presentan la información. Esta composición de las páginas contempla los estilos a utilizar en el sitio.

Esta estandarización contempla el manejo de un control “Encabezado”, en donde se establece:

- A la izquierda el logo de la compañía.
- Un conjunto de “pestañas”, que permite la navegabilidad del sitio.
- A la derecha un datos como: Usuario, Área, Desde, Fecha, Hora, Salir, Acerca.

Estas características se visualizan en la siguiente figura [10].

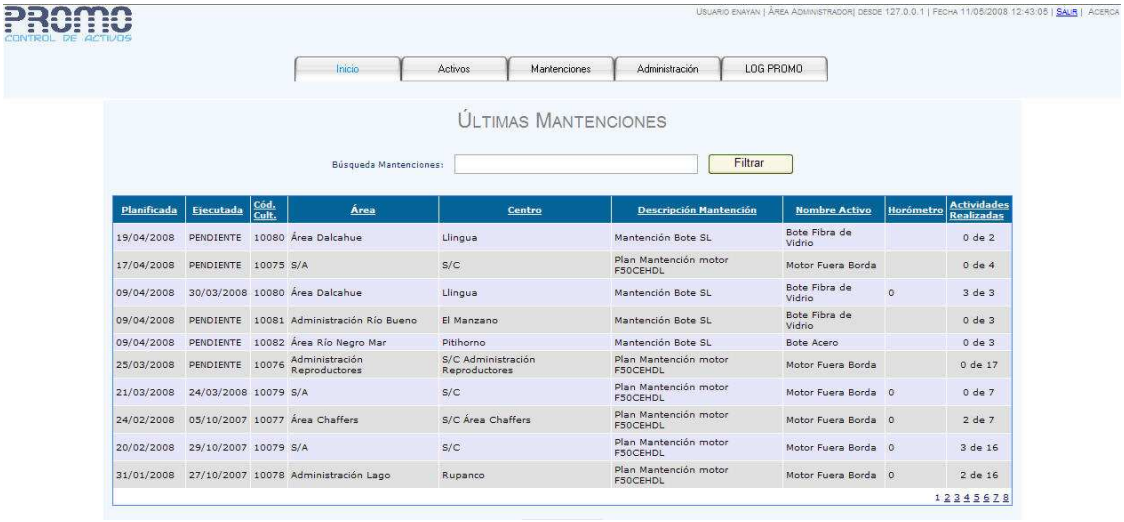
Figura Nº [10]: Encabezado página.

Además todas las páginas tendrán un “pie”, de página. Como muestra la figura [11].

Figura Nº [11]: Pie de página.

Finalmente se logra establecer un estándar para todas las páginas del sitio, permitiendo así que sólo se modifique la parte central de las páginas en donde se presenta la información que corresponda.

Las especificaciones que corresponden a colores de controles, estilos de letras etc. se logrará mediante un archivo de estilos o CSS.



Planificada	Ejecutada	Cód. Cult.	Área	Centro	Descripción Mantención	Nombre Activo	Horómetro	Actividades Realizadas
19/04/2008	PENDIENTE	10080	Área Dalcahue	Llingua	Mantención Bote SL	Bote Fibra de Vidrio		0 de 2
17/04/2008	PENDIENTE	10075	S/A	S/C	Plan Mantención motor F50CEHDL	Motor Fuera Borda		0 de 4
09/04/2008	30/03/2008	10080	Área Dalcahue	Llingua	Mantención Bote SL	Bote Fibra de Vidrio	0	3 de 3
09/04/2008	PENDIENTE	10081	Administración Río Bueno	El Manzano	Mantención Bote SL	Bote Fibra de Vidrio		0 de 3
09/04/2008	PENDIENTE	10082	Área Río Negro Mar	Pitihorno	Mantención Bote SL	Bote Acero		0 de 3
25/03/2008	PENDIENTE	10076	Administración Reproductores	S/C Administración Reproductores	Plan Mantención motor F50CEHDL	Motor Fuera Borda		0 de 17
21/03/2008	24/03/2008	10079	S/A	S/C	Plan Mantención motor F50CEHDL	Motor Fuera Borda	0	0 de 7
24/02/2008	05/10/2007	10077	Área Chaffers	S/C Área Chaffers	Plan Mantención motor F50CEHDL	Motor Fuera Borda	0	2 de 7
20/02/2008	29/10/2007	10079	S/A	S/C	Plan Mantención motor F50CEHDL	Motor Fuera Borda	0	3 de 16
31/01/2008	27/10/2007	10078	Administración Lago	Rupanco	Plan Mantención motor F50CEHDL	Motor Fuera Borda	0	2 de 16

1 2 3 4 5 6 7 8

Figura Nº [12]: Formato Principal Sistema PROMO.

En la figura [12], queda de manifiesto el estándar de representación de la información del sistema.

6.2.4 Diseño de Procedimientos de Carga Inicial de Datos.

La carga inicial de datos se realizará utilizando el “wizard” de importación que ofrece el motor de base de datos utilizado que soporta al sistema. Dicha información se encontrará en planillas Excel. Para realizar con éxito la migración de la información se deben respetar los formatos preestablecidos para dichas planillas.

7 Diseño de Base de Datos

La información que se presenta a continuación, se basa en la metodología propuesta por Thomas Connolly y Carolyn Begg, que permite la implementación y documentación de la base de datos a través de pasos preestablecidos que permiten un desarrollo escalable y confiable.

Esta metodología propone las siguientes etapas:

- Diseño conceptual
- Diseño lógico
- Diseño físico

Cabe señalar que el diseño de la base de datos que se describirá a continuación es un diseño inicial y que sufrirá cambios, ya que los módulos serán escalados para soportar los nuevos módulos de TI[10] y Redes[11] que serán implementados como una segunda fase del sistema.

[10] Tecnologías de información
[11] Referencia a las redes que mantienen a los salmones en sus jaulas.

7.1 Diseño Conceptual

Permite la construcción de un modelo de flujo de datos, para orientar el desarrollo del software.

La metodología propone identificar los siguientes componentes que forman parte del modelo abordado, estos son:

- Entidades.
- Relaciones.
- Atributos.
- Dominio de Atributos.
- Claves Candidatas.
- Claves Primarias.

7.1.1 Entidades

Del análisis de la toma de requerimientos, que fueron representados a través de los casos de uso se desprendieron las siguientes entidades para el modelo

conceptual de la base de datos. Las entidades se presentaran de acuerdo al módulo al cual pertenece.

Cabe mencionar que para el módulo de reportes no se identifican entidades y relaciones, ya que el módulo en cuestión sólo genera información consolidada de la base de datos en general.

7.1.1.1 Entidades Módulo Activos.

NOMBRE DE ENTIDAD	: Activo
DESCRIPCIÓN	: Almacena la información que identifica únicamente a un activo en particular.
ALIASES	: ACTIVO.
OCURRENCIA	- Ninguno, uno o muchos activos.

NOMBRE DE ENTIDAD	: TipoActivo
DESCRIPCIÓN	: Almacena información que define los tipos de activo que soportará la aplicación.
ALIASES	: TIPOACTIVO.
OCURRENCIA	- Ninguno o uno por cada activo

NOMBRE DE ENTIDAD	: ItemTipoActivo
DESCRIPCIÓN	: Almacena información que define los casos en que un tipo de activo puede tener ítems de tipo de activo.
ALIASES	: ITEMTIPOACTIVO.
OCURRENCIA	- Ninguno o uno por cada tipo de activo.

NOMBRE DE ENTIDAD : NombrePorTipoActivo

DESCRIPCIÓN : Almacena información que define los nombres que tendrán los activos según el tipo de activo.

ALIASES : NOMBREPORTIPOACTIVO.

OCURRENCIA - Ninguno o uno por cada activo.
- Ninguno o uno por cada tipo de activo

NOMBRE DE ENTIDAD : ItemActivo

DESCRIPCIÓN : Almacena información que identifica los ítems de activo.

ALIASES : ITEMACTIVO.

OCURRENCIA - Ninguno, uno o muchos por cada activo.

NOMBRE DE ENTIDAD : TipoItemActivo

DESCRIPCIÓN : Almacena las características que tendrán los ítems de activo.

ALIASES : TIPOITEMACTIVO.

OCURRENCIA - Ninguno, uno o más por cada Item de activo.

NOMBRE DE ENTIDAD : MarcaTipoActivo

DESCRIPCIÓN : Almacena información que determina la marca de los activos.

ALIASES : MARCATIPOACTIVO.

OCURRENCIA - Puede pertenecer a cero o más activos.

NOMBRE DE ENTIDAD : ModeloTipoActivo

DESCRIPCIÓN : Almacena información que determina la marca de los activos.

ALIASES : MODELOTIPOACTIVO.

OCURRENCIA - Puede pertenecer a cero o más activos.

NOMBRE DE ENTIDAD : ExtProveedor

DESCRIPCIÓN : Almacena información que determina los

proveedores.

ALIASES : EXTPROVEEDOR.

OCURRENCIA - Puede tener cero o más sucursales.
- Puede pertenecer a ninguno, uno o más activos.

NOMBRE DE ENTIDAD : ExtSucursal

DESCRIPCIÓN : Almacena información que determina las sucursales de los proveedores.

ALIASES : EXTSUCURSAL.

OCURRENCIA - Puede pertenecer solo a un proveedor.
- Puede pertenecer a ninguno, uno o más activos.

NOMBRE DE ENTIDAD : RecepcionParcialActivo

DESCRIPCIÓN : Almacena información referente a las recepciones parciales que se pueda tener de un activo.

ALIASES : RECEPCIONPARCIALACTIVO.

OCURRENCIA - Puede pertenecer a ninguno, uno o más activos.

NOMBRE DE ENTIDAD : ValorVariableDeActivo

DESCRIPCIÓN : Almacena información referente a las diferentes mediciones que se puedan realizar de un activo en particular.

ALIASES : VALORVARIABLEDEACTIVO.

OCURRENCIA - Ninguno o un activo.

NOMBRE DE ENTIDAD : VariableMedicion

DESCRIPCIÓN : Almacena información referente a la descripción de las distintas variables de medición que pueda tener un activo.

ALIASES : VARIABLEMEDICION.

OCURRENCIA - Ninguno, uno más por cada valor variable de activo.

7.1.1.2 Entidades Módulo Mantenciones

NOMBRE DE ENTIDAD	: HitoMantencionTeorica
DESCRIPCIÓN	: Almacena información referente a la descripción de las mantenciones teóricas que posiblemente se realicen a los activos pertenecientes a un tipo de activo.
ALIASES	: HITOMANTENCIONTEORICA.
OCURRENCIA	- Ninguno, uno o más veces.

NOMBRE DE ENTIDAD	: DetalleMantencion
DESCRIPCIÓN	: Almacena información que determina que característica tendrá la mantención teórica, que se realizara a un activo.
ALIASES	: DETALLEMANTENCION.
OCURRENCIA	- Ninguno, una o más veces por cada hito mantención teórica.

NOMBRE DE ENTIDAD	: TipoMantencionActivo
DESCRIPCIÓN	: Almacena información que determina si una mantención teórica, se realizara de forma interna o externa.
ALIASES	: TIPOMANTENCIONACTIVO.
OCURRENCIA	- Ninguno o una vez por cada hito mantención teórica.

NOMBRE DE ENTIDAD	: ActividadTeorica
DESCRIPCIÓN	: Almacena información que determina las actividades que se realizaran en una mantención teórica.
ALIASES	: ACTIVIDADTEORICA.
OCURRENCIA	- Ninguno o una vez por cada hito mantención teórica.

NOMBRE DE ENTIDAD	: HitoMantencionEfectiva

DESCRIPCIÓN : Almacena información referente a las mantenciones teóricas que se han ejecutado y que se transforman en mantenciones efectivas.

ALIASES : TIPOMANTENCIONACTIVO.

OCURRENCIA - Ninguno, una o más veces por cada hito mantención teórica.

NOMBRE DE ENTIDAD : Actividadefectiva

DESCRIPCIÓN : Almacena información referente a las actividades teóricas que son ejecutadas y que se transforman en actividades efectivas.

ALIASES : ACTIVIDADEFECTIVA.

OCURRENCIA - Ninguno o una vez por cada actividad teórica.

NOMBRE DE ENTIDAD : FrecuenciaTipoActivo

DESCRIPCIÓN : Almacena información referente a la frecuencia de tiempo en que se realizaran los hitos de mantención efectiva, según modelo y tipo de activo.

ALIASES : FRECUENCIATIPOACTIVO.

OCURRENCIA - Ninguno o una vez por cada hito mantención efectiva.

NOMBRE DE ENTIDAD : ActividadFrecuencia

DESCRIPCIÓN : Almacena información que determina las actividades y la frecuencia de tiempo en que serán realizadas las actividades de las mantenciones teóricas según modelo y tipo de activo.

ALIASES : ACTIVIDADFRECUENCIA.

OCURRENCIA - Ninguno o una vez por cada actividad teórica.

NOMBRE DE ENTIDAD : VariableParaTipoActivo

DESCRIPCIÓN : Almacena información que determina las variables de medición según Tipo Activo, que utilizaran para realizar Hitos de Mantención Teórica.

ALIASES	: VARIABLEPARATIPOACTIVO.
OCURRENCIA	- Ninguno o una vez por cada Hito Mantención teórica.

7.1.1.3 Entidades transversales para todos los módulos

NOMBRE DE ENTIDAD	: Usuario
DESCRIPCIÓN	: Almacena información referente a los usuarios del sistema.
ALIASES	: USUARIO.
OCURRENCIA	- Ninguno, uno o más usuarios.

NOMBRE DE ENTIDAD	: GrupoUsuario
DESCRIPCIÓN	: Almacena información referente a los grupos de usuarios que tendrá el sistema.
ALIASES	: GRUPOUSUARIO.
OCURRENCIA	- Ninguno, uno o más grupos de usuarios.

NOMBRE DE ENTIDAD	: Secuencia
DESCRIPCIÓN	: Almacena información que determina los valores de inicio que tendrán los números correlativos que utilice el sistema.
ALIASES	: SECUENCIA.
OCURRENCIA	- Ninguno, uno o más secuencias.

NOMBRE DE ENTIDAD	: ParametroSistema
DESCRIPCIÓN	: Almacena información que determina los valores que pueden ser modificados en el tiempo según criterio del administrador del sistema.
ALIASES	: PARAMETROSISTEMA.
OCURRENCIA	- Ninguno, uno o más parámetros.

NOMBRE DE ENTIDAD	: LogPromo
DESCRIPCIÓN	: Almacena información referente a los cambios de área o centro que se pueda realizar de un activo.
ALIASES	: LOGPROMO.
OCURRENCIA	- Ninguno, uno o más log's Promo.

7.1.2 Relaciones.

Con la identificación de las entidades, a continuación se exhiben las siguientes relaciones.

7.1.2.1 Relaciones Módulo Activos.

TIPO DE ENTIDAD	: Activo
TIPO DE RELACIÓN	: ActivoTieneExtSucursal
DESCRIPCIÓN	- Un activo puede tener una sucursal y una sucursal puede estar asociado a n activos.
TIPO ENTIDAD	: ExtSucursal
CARDINALIDAD	: 1, n

TIPO DE ENTIDAD	: Activo
TIPO DE RELACIÓN	: ActivoTieneMarcaTipoActivo
DESCRIPCIÓN	- Un activo puede tener una MarcaTipoActivo y una MarcaTipoActivo puede estar asociado a n activos.
TIPO ENTIDAD	: MarcaTipoActivo
CARDINALIDAD	: 1, n

TIPO DE ENTIDAD : Activo

TIPO DE RELACIÓN : ActivoTieneModeloTipoActivo

DESCRIPCIÓN - Un activo puede tener un ModeloTipoActivo y un ModeloTipoActivo puede estar asociado a n activos.

TIPO ENTIDAD : ModeloTipoActivo

CARDINALIDAD : 1, n

TIPO DE ENTIDAD : Activo

TIPO DE RELACIÓN : ActivoTipoActivo

DESCRIPCIÓN - Un activo puede pertenecer a un TipoActivo y un TipoActivo puede estar asociado a n activos.

TIPO ENTIDAD : TipoActivo

CARDINALIDAD : 1, n

TIPO DE ENTIDAD : Activo

TIPO DE RELACIÓN : ActivoItemActivo

DESCRIPCIÓN - Un activo puede tener n ItemActivo y un ItemActivo puede estar asociado a un solo activo.

TIPO ENTIDAD : ItemActivo

CARDINALIDAD : 1, n

TIPO DE ENTIDAD : Activo

TIPO DE RELACIÓN : ActivoPerteneceArea

DESCRIPCIÓN - Un activo puede pertenecer a una Area y una Area puede estar asociado a n activos.

TIPO ENTIDAD : Area

CARDINALIDAD : 1, n

TIPO DE ENTIDAD : Activo

TIPO DE RELACIÓN : ActivoPerteneceCentro

DESCRIPCIÓN - Un activo puede pertenecer a un Centro y un

	Centro puede estar asociado a n activos.
TIPO ENTIDAD	: Centro
CARDINALIDAD	: 1, n
TIPO DE ENTIDAD	: Activo
TIPO DE RELACIÓN	: ActivoTieneRecepcionParcialActivo
DESCRIPCIÓN	- Un activo puede tener n RecepcionParcialActivo y una RecepcionParcialActivo puede estar asociado a un solo activo.
TIPO ENTIDAD	: RecepcionParcialActivo
CARDINALIDAD	: 1, n
TIPO DE ENTIDAD	: Activo
TIPO DE RELACIÓN	: ActivoTieneNombrePorTipoActivo
DESCRIPCIÓN	- Un activo puede tener un NombrePorTipoActivo y un NombrePorTipoActivo puede estar asociado a n activos.
TIPO ENTIDAD	: NombrePorTipoActivo
CARDINALIDAD	: 1, n
TIPO DE ENTIDAD	: Activo
TIPO DE RELACIÓN	: ActivoValorVariabledeActivo
DESCRIPCIÓN	- Un activo puede tener n ValorVariabledeActivo y un ValorVariabledeActivo puede estar asociado a un solo activo.
TIPO ENTIDAD	: ValorVariabledeActivo
CARDINALIDAD	: 1, n
TIPO DE ENTIDAD	: TipoActivo
TIPO DE RELACIÓN	: TipoActivoTieneModeloTipoActivo
DESCRIPCIÓN	- Un TipoActivo puede estar asociado a n ModeloTipoActivo y un ModeloTipoActivo puede estar asociado a n TipoActivo.

TIPO ENTIDAD : ModeloTipoActivo

CARDINALIDAD : n, n

TIPO DE ENTIDAD : TipoActivo

TIPO DE RELACIÓN : TipoActivoTieneMarcaTipoActivo

DESCRIPCIÓN - Un TipoActivo puede estar asociado a n MarcaTipoActivo y una MarcaTipoActivo puede estar asociado a n TipoActivo.

TIPO ENTIDAD : MarcaTipoActivo

CARDINALIDAD : n, n

TIPO DE ENTIDAD : TipoActivo

TIPO DE RELACIÓN : TipoActivoExtProveedor

DESCRIPCIÓN - Un TipoActivo puede estar asociado a n ExtProveedor y un ExtProveedor puede estar asociado a n TipoActivo.

TIPO ENTIDAD : ExtProveedor

CARDINALIDAD : n, n

TIPO DE ENTIDAD : TipoActivo

TIPO DE RELACIÓN : TipoActivoNombrePorTipoActivo

DESCRIPCIÓN - Un TipoActivo puede estar asociado a n NombrePorTipoActivo y un NombrePorTipoActivo puede estar asociado a un solo TipoActivo.

TIPO ENTIDAD : NombrePorTipoActivo

CARDINALIDAD : 1, n

TIPO DE ENTIDAD : Area

TIPO DE RELACIÓN : AreaTieneCentro

DESCRIPCIÓN - Una Area puede estar asociado a n Centro y un Centro puede estar asociado a una sola Area.

TIPO ENTIDAD : Centro

CARDINALIDAD : 1, n

TIPO DE ENTIDAD : ItemActivo

TIPO DE RELACIÓN : ItemActivoTieneTipoItemActivo

DESCRIPCIÓN - Una ItemActivo puede tener un solo TipoItemActivo y un TipoItemActivo puede estar asociado a n ItemActivo.

TIPO ENTIDAD : TipoItemActivo

CARDINALIDAD : 1, n

TIPO DE ENTIDAD : ExtProveedor

TIPO DE RELACIÓN : ExtProveedorTieneExtSucursal

DESCRIPCIÓN - Un ExtProveedor puede tener n ExtSucursal y una ExtSucursal puede estar asociado a un solo ExtProveedor.

TIPO ENTIDAD : ExtSucursal

CARDINALIDAD : 1, n

TIPO DE ENTIDAD : MarcaTipoActivo

TIPO DE RELACIÓN : MarcaTipoActivoTieneModeloTipoActivo

DESCRIPCIÓN - Una MarcaTipoActivo puede tener n ModeloTipoActivo y un ModeloTipoActivo puede estar asociado a una sola MarcaTipoActivo.

TIPO ENTIDAD : ModeloTipoActivo

CARDINALIDAD : 1, n

7.1.2.2 Relaciones Módulo Mantenciones

TIPO DE ENTIDAD : HitoMantencionTeorica

TIPO DE RELACIÓN : HitoMantencionTeoricaTieneDetalleMantencion

DESCRIPCIÓN - Un HitoMantencionTeorica puede tener n

	DetalleMantencion y un DetalleMantencion puede estar asociado a un solo HitoMantencionTeorica.
TIPO ENTIDAD	: DetalleMantencion
CARDINALIDAD	: 1, n
TIPO DE ENTIDAD	: HitoMantencionTeorica
TIPO DE RELACIÓN	: HitoMantencionTeoricaTieneTipoMantencionActivo
DESCRIPCIÓN	- Un HitoMantencionTeorica puede tener un TipoMantencionActivo y un TipoMantencionActivo puede estar asociado a n HitoMantencionTeorica.
TIPO ENTIDAD	: TipoMantencionActivo
CARDINALIDAD	: 1, n
TIPO DE ENTIDAD	: HitoMantencionTeorica
TIPO DE RELACIÓN	: HitoMantencionTeoricaTieneActividadTeorica
DESCRIPCIÓN	- Un HitoMantencionTeorica puede tener n ActividadTeorica y una ActividadTeorica puede estar asociado a un solo HitoMantencionTeorica.
TIPO ENTIDAD	: ActividadTeorica
CARDINALIDAD	: 1, n
TIPO DE ENTIDAD	: HitoMantencionTeorica
TIPO DE RELACIÓN	: HitoMantencionTeoricaTieneHitoMantencionEfectiva
DESCRIPCIÓN	- Un HitoMantencionTeorica puede tener n HitoMantencionEfectiva y un HitoMantencionEfectiva puede estar asociado a un solo HitoMantencionTeorica.
TIPO ENTIDAD	: HitoMantencionEfectiva
CARDINALIDAD	: 1, n
TIPO DE ENTIDAD	: HitoMantencionTeorica
TIPO DE RELACIÓN	: HitoMantencionTeoricaTieneModeloTipoActivo

DESCRIPCIÓN	- Un HitoMantencionTeorica puede tener un ModeloTipoActivo y un ModeloTipoActivo puede estar asociado a n HitoMantencionTeorica.
TIPO ENTIDAD	: ModeloTipoActivo
CARDINALIDAD	: 1, n
TIPO DE ENTIDAD	: Activo
TIPO DE RELACIÓN	: ActivoTieneHitoMantencionEfectiva
DESCRIPCIÓN	- Un Activo puede tener n HitoMantencionEfectiva y un HitoMantencionEfectiva puede estar asociado a un solo Activo.
TIPO ENTIDAD	: HitoMantencionEfectiva
CARDINALIDAD	: 1, n
TIPO DE ENTIDAD	: HitoMantencionEfectiva
TIPO DE RELACIÓN	: HitoMantencionEfectivaTieneFrecuenciaTipoActivo
DESCRIPCIÓN	- Un HitoMantencionEfectiva puede tener una FrecuenciaTipoActivo y una FrecuenciaTipoActivo puede estar asociado a n HitoMantencionEfectiva.
TIPO ENTIDAD	: FrecuenciaTipoActivo
CARDINALIDAD	: 1, n
TIPO DE ENTIDAD	: HitoMantencionEfectiva
TIPO DE RELACIÓN	: HitoMantencionEfectivaTieneActividadEfectiva
DESCRIPCIÓN	- Un HitoMantencionEfectiva puede tener n ActividadEfectiva y una ActividadEfectiva puede estar asociado a un solo HitoMantencionEfectiva.
TIPO ENTIDAD	: ActividadEfectiva
CARDINALIDAD	: 1, n
TIPO DE ENTIDAD	: ActividadTeorica

TIPO DE RELACIÓN	: ActividadTeoricaTieneActividadFrecuencia
DESCRIPCIÓN	- Una ActividadTeorica puede tener n ActividadFrecuencia y una ActividadFrecuencia puede estar asociado a una sola ActividadTeorica.
TIPO ENTIDAD	: ActividadFrecuencia
CARDINALIDAD	: 1, n

TIPO DE ENTIDAD	: ActividadTeorica
TIPO DE RELACIÓN	: ActividadTeoricaTieneActividadEfectiva
DESCRIPCIÓN	- Una ActividadTeorica puede tener n ActividadEfectiva y una ActividadEfectiva puede estar asociado a una sola ActividadTeorica.
TIPO ENTIDAD	: ActividadEfectiva
CARDINALIDAD	: 1, n

TIPO DE ENTIDAD	: FrecuenciaTipoActivo
TIPO DE RELACIÓN	: FrecuenciaTipoActivoTieneActividadFrecuencia
DESCRIPCIÓN	- Una FrecuenciaTipoActivo puede tener n ActividadFrecuencia y una ActividadFrecuencia puede estar asociado a una sola FrecuenciaTipoActivo.
TIPO ENTIDAD	: ActividadFrecuencia
CARDINALIDAD	: 1, n

TIPO DE ENTIDAD	: FrecuenciaTipoActivo
TIPO DE RELACIÓN	: FrecuenciaTipoActivoPerteneceTipoActivo
DESCRIPCIÓN	- Una FrecuenciaTipoActivo puede pertenecer a un TipoActivo y un TipoActivo puede estar asociado a n FrecuenciaTipoActivo.
TIPO ENTIDAD	: TipoActivo
CARDINALIDAD	: 1, n

TIPO DE ENTIDAD	: FrecuenciaTipoActivo
TIPO DE RELACIÓN	: FrecuenciaTipoActivoTieneModeloTipoActivo
DESCRIPCIÓN	- Una FrecuenciaTipoActivo puede tener un ModeloTipoActivo y un ModeloTipoActivo puede estar asociado a n FrecuenciaTipoActivo.
TIPO ENTIDAD	: ModeloTipoActivo
CARDINALIDAD	: 1, n

7.1.2.3 Relaciones transversales para todos los módulos

TIPO DE ENTIDAD	: Usuario
TIPO DE RELACIÓN	: UsuarioPerteneceGrupoUsuario
DESCRIPCIÓN	- Usuario puede pertenecer a n GrupoUsuario y un GrupoUsuario puede estar asociado a n Usuario.
TIPO ENTIDAD	: GrupoUsuario
CARDINALIDAD	: 1, n

TIPO DE ENTIDAD	: Usuario
TIPO DE RELACIÓN	: UsuarioPerteneceArea
DESCRIPCIÓN	- Usuario puede pertenecer a una Area y una Area puede estar asociado a un solo Usuario.
TIPO ENTIDAD	: Area
CARDINALIDAD	: 1, n

7.1.3 Identificar y asociar atributos con una entidad o relación

Del análisis de las entidades y relaciones antes mencionadas, se presentaron los siguientes atributos para cada una de ellas.

Algunas consideraciones a tomar en cuenta:

- Para este modelo no se asignaron valores por defecto.
- En la columna “Tipo Dato”, solo se incluye información del tipo de dato y ejemplos cuando corresponda, las características que tienen relación con: “tamaño” y “rango”, serán especificadas en el siguiente apartado 7.1.4.
- Los nombres de las entidades serán abreviados para una mejor visualización de las tablas.

7.1.3.1 Módulo Activos.

Entidad Activo

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
IdArea	Entero	PK	NO	Identificador del área al cual pertenece el activo.
IdActivo	Entero	PK	NO	Identificador del activo.

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
NombreActivo	Varchar (100)		SI	Descripción y / o información adicional del nombre del activo.
NumeroSerieActivo	Varchar (50)		SI	Almacena número serie del activo.
OrdCompraActivo	Varchar (100)		NO	Almacena la orden de compra con la cual fue adquirido el activo.
FacturaActivo	Entero		SI	Almacena el número de factura del activo.
GuiaDespActivo	Entero		SI	Almacena número de la guía de despacho al cual pertenece el activo.
FechaCreaActivo	Datetime		SI	Almacena fecha en que se creo el activo.
DescripciónActivo	Varchar (300)		SI	Descripción y / o información adicional del activo.
EstadoActivoOracle	Varchar (50) "INGRESADO","NO INGRESADO"		SI	Establece si el activo se encuentra ingresado en la aplicación Oracle Financial.
NumActivoOracle	Entero		SI	Almacena el número con que se encuentra el activo en la aplicación Oracle Financial
EstadoActPROMO	Varchar (50) "ACTIVO", "BAJA", "ELIMINADO"		SI	Establece el estado en que tiene el activo en el sistema.

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
FechaBajaActivo	Datetime		SI	Almacena la fecha en que se estableció la "baja", del activo.

Entidad Tipo Activo

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
CodigoTipoActivo	Varchar (30)	PK	NO	Nombre único que identifica los tipos de activo que existen en el Sistema.
NombreTipoActivo	Varchar (30)		NO	Descripción y / o información que establece el nombre del tipo de activo.
TieneItemActivo	bit		SI	Establece la existencia de ítems de activo para el tipo de activo.

Entidad Area

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
IdArea	Entero	PK	NO	Identificador del área.
NombreArea	Varchar (50)		SI	Descripción y / o información que establece el nombre del área.
CentroCostoArea	Varchar (50)		SI	Almacena el centro de costo al cual pertenece el área.

Entidad Centro

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
IdCentro	Entero	PK	NO	Identificador del centro.

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
NombreCentro	Varchar (50)		SI	Descripción y / o información que establece el nombre del centro.
CentroCostoCentro	Varchar (50)		SI	Almacena el centro de costo al cual pertenece el centro.

Entidad NombrePorTipoActivo

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
IdNomPorTipoAct	Entero	PK	NO	Identificador del Nombre por tipo de activo.
NomPorTipoAct	Varchar (100)		SI	Descripción y / o información que establece los nombres del tipo de activo.

Entidad VariableMedicion

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
IdVariableMedicion	Entero	PK	NO	Identificador de la variable de medición.
NomVarMedicion	Varchar (100)		NO	Descripción y / o información que establece el nombre de la variable de medición.
DesVarMedicion	Varchar (100)		SI	Descripción y / o información adicional.

Entidad ExtSucursal

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
NumeroSucursal	Entero	PK	NO	Identificador de la sucursal.

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
NombreSucursal	Varchar (100)		SI	Descripción y / o información que establece el nombre de la sucursal.
DireccionSucursal	Varchar (50)		SI	Descripción y / o información que establece la dirección.

Entidad ExtProveedor

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
IdProveedor	Entero	PK	NO	Identificador del proveedor.
NombreProveedor	Varchar (100)		SI	Descripción y / o información que establece el nombre del proveedor.
DireccionProveedor	Varchar (100)		SI	Descripción y / o información que establece la dirección.

Entidad ModeloTipoActivo

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
IdModeloTipoAct	Entero	PK	NO	Identificador del modelo según tipo de activo.
NomModeloTipoAct	Varchar (50)		SI	Descripción y / o información que establece el nombre del modelo según tipo de activo.

Entidad MarcaTipoActivo

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
IdMarcaTipoAct	Entero	PK	NO	Identificador de la marca según tipo de activo.
NomMarcaTipoAct	Varchar (50)		SI	Descripción y / o información que establece el nombre de la marca.

Entidad ItemActivo

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
IdItemActivo	Entero	PK	NO	Identificador item activo
DescripcionItemAct	Varchar (50)		SI	Descripción y / o información adicional.
NumFactItemAct	Entero		SI	Almacena el número de factura del item de activo.
GuiaDespItemAct	Entero		SI	Almacena el número de la guía de despacho al cual pertenece el item de activo.
OrdCompraItemAct	Varchar (50)		SI	Almacena el número de orden de compra del item de activo.
LongitudItemActivo	Float		SI	En caso que el item de activo tenga esta característica, se establece la longitud del item de activo.
DiametroItemActivo	Float		SI	Almacena el diámetro del item de activo
DiametroPulgItemActivo	Varchar (50) EJ: “12.5”		SI	Almacena el diámetro en pulgadas del item de activo.
unidadesItemActivo	Float		SI	Almacena la cantidad de unidades del Item de activo.

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
volumenItemActivo	Float		SI	Almacena volumen del item de activo.
PesoTItemActivo	Float		SI	Almacena el peso del item de activo.
FechaCreaItemActivo	Datetime		SI	Establece la fecha de creación del item de activo
NumActOracleItemAct	Entero		SI	Identifica el número que tiene el item de activo en la aplicación "Oracle Financial".
IdProveedorItemActivo	Entero		SI	Almacena el identificador del proveedor.

Entidad TipoItemActivo

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
CodigoTipoItemAct	Varchar (50)	PK	NO	Identificador item activo
NomTipoItemAct	Varchar (100)		NO	Descripción y / o información del nombre.
TieneLongItemAct	Boleano		NO	Establece si el item de activo posee Longitud.
TieneDiamItemAct	Boleano		NO	Establece si el item de activo posee Diámetro.
TieneDiamPulgItemAct	Boleano		NO	Establece si el item de activo posee Diámetro en pulgadas.
TieneUnidItemAct	Boleano		NO	Establece si el item de activo posee Unidades
TieneVolumenItemActivo	Boleano		NO	Establece si el item de activo posee Volumen.

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
TienePesoTItemActivo	Boleano		NO	Establece si el item de activo posee Peso.

Entidad RecepcionParcialActivo

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
IdOrdenRecAct	Entero	PK	NO	Identificador Orden de Recepción Parcial activo.
FechaRecParcialAct	Datetime		SI	Establece la fecha de recepción parcial.
DesRecParcialAct	Varchar (100)		SI	Descripción y / o información adicional.

7.1.3.2 Módulo Mantenciones.

Entidad HitoMantencionTeorica

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
IdMantTeoricaAct	Entero	PK	NO	Identificador mantenciones teóricas de activo
DesMantTeorica	Varchar (300)		NO	Descripción y / o información adicional.
ValorVarMantTeorica.	Entero		SI	Identificador del valor de variable de medición.

Entidad DetalleMantencion

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
IdDetalleMantencion	Entero	PK	NO	Identificador detalle mantenciones teóricas de activo

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
DesDetMant	Varchar (300)		SI	Descripción y / o información adicional.

Entidad TipoMantencionActivo

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
CodTipoMantAct	Varchar (30)	PK	NO	Identificador tipo mantención
NombreTipoMant	Varchar (100)		SI	Descripción y / o información adicional.

Entidad ActividadTeorica

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
IdActividadTeorica	Varchar (30)	PK	NO	Identificador de la actividad teorica de un hito de mantención teórica
DescripcionActividad	Varchar (100)		NO	Descripción y / o información adicional.

Entidad HitoMantencionEfectiva

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
IdMantEfectivaActivo	Varchar (30)	PK	NO	Identificador mantención efectiva.
FechaMantActivo	Datetime		NO	Fecha planificada para realizar la mantención.
FechaEjecActivo	Datetime		SI	Fecha en que se ejecuto la mantención.

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
EstadoMantActivo	Varchar (50)		NO	Estado en que se encuentra la mantención, por EJ: "PLANIFICADA", "EJECUTADA"
MaterialesMantActivo	Varchar (300)		SI	Descripción y / o información adicional.
ObsMantActivo	Varchar (300)		SI	Descripción y / o información adicional.
DesMantencionActivo	Varchar (100)		SI	Descripción y / o información adicional.

7.1.3.3 Módulo Reportes

Al no identificarse entidades para este módulo, se entiende que no existen atributos para relacionar con entidades o relaciones.

7.1.3.4 Identificación de Atributos con entidades o relaciones transversales a todos los módulos.

Entidad Usuario

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
Username	Char (10)	PK	NO	Identificador del usuario.

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
Password	Char (10)		NO	Almacena la contraseña del usuario.
NombreUsuario	Varchar (10)		NO	Almacena el nombre de usuario
EsAdministrador	Bit		NO	Identifica si el usuario tiene privilegios de administrador del Sistema.
CorreoElectronico	Varchar (10)		NO	Descripción del correo electrónico del usuario.

Entidad GrupoUsuario

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
CodigoGrupo	Varchar (30)	PK	NO	Identificador del grupo de usuario.
GlosaGrupo	Varchar (50)		SI	Establece el nombre del grupo de usuario.

Entidad ParametroSistema

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
Nombre Parametro	Entero	PK	NO	Identificador del parámetro generalizado.
ValorParametro			SI	Descripción y / o información adicional.

Entidad Secuencia

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
CodigoSecuencia	Char (20)	PK	NO	Identificador de la secuencia.
ProximoCorrelativo	Entero		SI	Descripción y / o información adicional.

Entidad LOG

Nombre	Tipo de Dato	Restricción (PK, AK)	Nulo	Descripción
IdLog	Entero	PK	NO	Identificador de la secuencia.
FechaHoraLog	Datetime		NO	Fecha y hora en que se traslada un activo desde un Área o Centro de cultivo.
DescripcionLog	Varchar (300)		NO	Descripción y / o información adicional.
Username	Varchar (50)		NO	Nombre del usuario que esta trasladando un activo desde un Área o Centro de cultivo.
IdArea	Entero		NO	Identificador de área origen del activo.
IdActivo	Entero		NO	Identificador del activo.

7.1.4 Documentación de los Dominios de Atributos

En la siguiente tabla se mostrarán los dominios de los atributos.

Nombre	Característica	Descripción
D_IDENTIFICADOR	Entero	Números enteros con signo
D_CANTIDAD	Entero	Números enteros con signo
D_CODIGO_LARGO	Char(20)	Cadena caracteres con largo máximo 20.
D_CORRELATIVO	Entero	Números enteros con signo
D_DECIMAL	Float	Números decimales
D_FECHA	Datetime	Fechas
D_GLOSA	Varchar(50)	Cadena alfanumérica con largo máximo 50
D_GLOSA_FRECUENCIA	Varchar(5000)	Cadena alfanumérica con largo máximo 5000
D_GLOSA_LARGA	Varchar(100)	Cadena alfanumérica con largo máximo 100
D_GLOSA_TEXTO	Varchar(300)	Cadena alfanumérica con largo máximo 300
D_NOMBRE_LARGO	Varchar(100)	Cadena alfanumérica con largo máximo 100
D_NOTICIA	Varchar(100)	Cadena alfanumérica con largo máximo 100
D_RUT	Entero	Números enteros con signo
D_RUT_GLOSA	Varchar(10)	Cadena alfanumérica con largo máximo 10
D_SERIE	Varchar(50)	Cadena alfanumérica con largo máximo 50
D_USUARIO	Char(10)	Cadena caracteres con largo máximo 20.
D_VERDADERO_O_FALSO	Boleano	Numero binario(1 o 0)

Tabla Nº [5]: Dominio de atributos

7.1.5 Identificación de Atributos para Claves Primarias y Candidatas

A continuación se presenta una tabla con la identificación de las claves primarias y candidatas de cada unas de las entidades.

Nombre	Alias	Tipo	Pertenece
IdArea	IDAREA	PK,FK	Entidad 'Activo'
IdActivo	IDACTIVO	PK	Entidad 'Activo'
CodigoTipoActivo	CODIGOTIPOACTIVO	PK	Entidad 'TipoActivo'
IdNombrePorTipo Activo	IDNOMBREPORTIPO ACTIVO	PK	Entidad 'NombrePorTipoActivo'
IdArea	IDAREA	PK	Entidad 'Area'
IdCentro	IDCENTRO	PK	Entidad 'Centro'
NumeroSucursal	NUMEROSUCURSAL	PK	Entidad 'ExtSucursal'
IdProveedor	IDPROVEEDOR	PK	Entidad 'ExtProveedor'
IdMarcaTipo Activo	IDMARCATIPO ACTIVO	PK	Entidad 'MarcaTipoActivo'
IdModeloTipo Activo	IDMODELOACTIVO	PK	Entidad 'ModeloTipoActivo'
IdValorVariable DeActivo	IDVALORVARIABLE DEACTIVO	PK	Entidad 'ValorVariable DeActivo'
IdVariablede Medicion	IDVARIABLEDE MEDICION	PK	Entidad 'Variablede Medicion'
IdItemTipoActivo	IDITEMTIPOACTIVO	PK	Entidad 'ItemTipoActivo'
IdItemActivo	IDITEMACTIVO	PK	Entidad 'ItemActivo'
CodigoTipoItem Activo	CODIGOTIPOITEM ACTIVO	PK	Entidad 'TipoItemActivo'
IdOrden RecepcionParcial Activo	IDORDEN RECEPCIONPARCIAL ACTIVO	PK	Entidad 'RecepcionParcialActivo'
IdHitoMantencion Teorica	IDHITOMANTENCION TEORICA	PK	Entidad 'HitoMantencionTeorica'
IdDetalle Mantencion	IDDETALLE MANTENCION	PK	Entidad 'DetalleMantencion'
CodigoTipo MantencionActivo	CODIGOTIPO MANTENCIONACTIVO	PK	Entidad 'TipoMantencionActivo'

Nombre	Alias	Tipo	Pertenece
Idactividad Teorica	IDACTIVIDAD TEORICA	PK	Entidad ‘ActividadTeorica’
IdHitoMantencion Efectiva	IDHITOMANTENCION EFECTIVA	PK	Entidad ‘HitoMantencionEfectiva’
Username	USERNAME	PK	Entidad 'Usuario'
CodigoGrupo	CODIGOGRUPO	PK	Entidad ‘GrupoUsuarios’
CodigoSecuencia	CODIGOSECUENCIA	PK	Entidad 'CodigoSecuencia'
IdLog	IDLOG	PK	Entidad ‘LogPromo’
Nombre Parametro	NOMBRE PARAMETRO	PK	Entidad ‘ParametroSistema’

Tabla Nº [6]: Lista de claves primarias

7.1.7 Diagrama Modelo Conceptual BDPROMO

A continuación se presenta el modelo conceptual de datos para el módulo Activos.

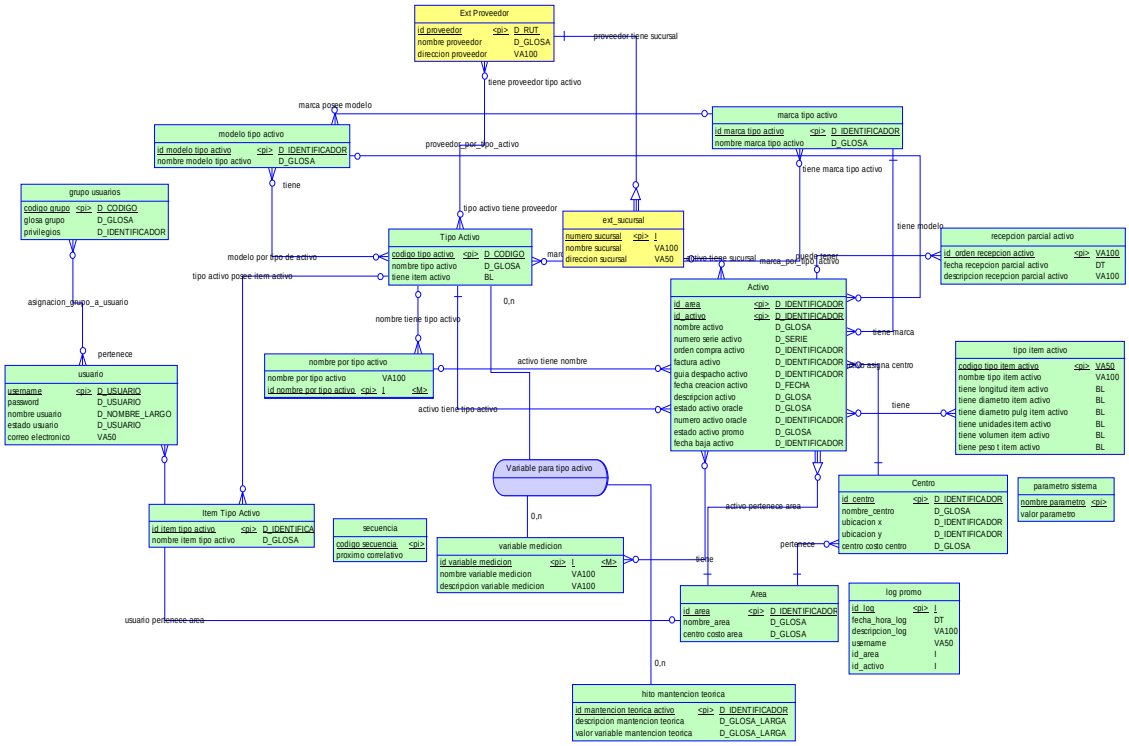


Diagrama Nº [16]: Modelo Conceptual Módulo Activos.

A continuación se presenta el modelo conceptual de datos para el módulo Mantenciones.

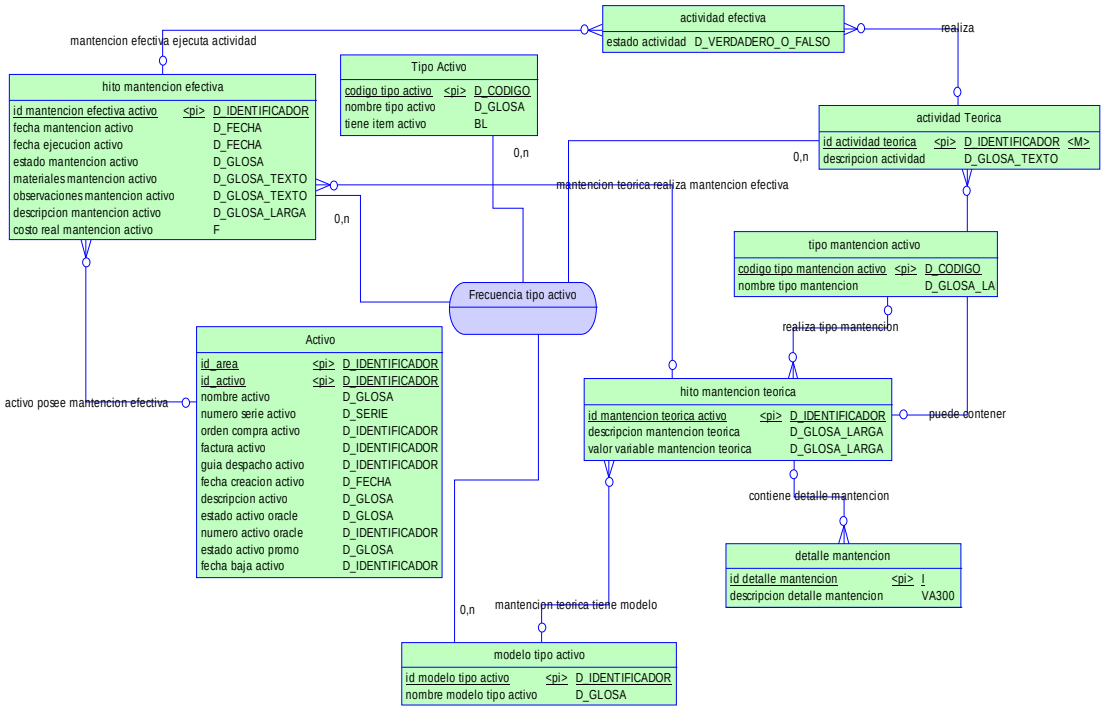


Diagrama Nº [17]: Modelo Conceptual Módulo Mantenciones.

7.2 Modelo Lógico BDPROMO (“Base de Datos Sistema Programa Mantención Operaciones”)

A continuación, se presenta el desarrollo para lograr la implementación del modelo lógico de la base de datos “BDPROMO”

7.2.1 Mapeo del Modelo Conceptual al Modelo Físico

En esta etapa se pretende redefinir algunas estructuras que podrían generar dificultades en la implementación final de la base de datos.

7.2.1.1 Relaciones “Muchos - a - Muchos”

7.2.1.1.1 Relaciones “Muchos - a- Muchos” para módulo activos

La relación muchos-a-muchos formada por las relaciones “ModeloTipoActivo” y “TipoActivo”, se transformó tal como muestra la siguiente figura.

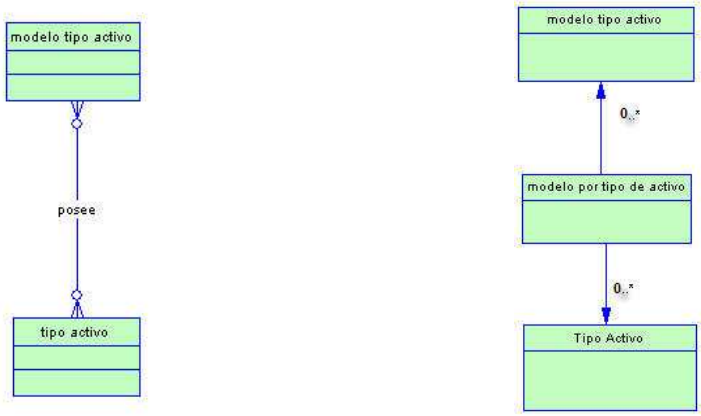


Figura Nº [13]: Relación Muchos – A – Muchos "ModeloTipoActivo – A – TipoActivo".

Así mismo la relación muchos-a-muchos formada por las relaciones "Activo" y "VariableMedicion", se transformó tal como muestra la siguiente figura.

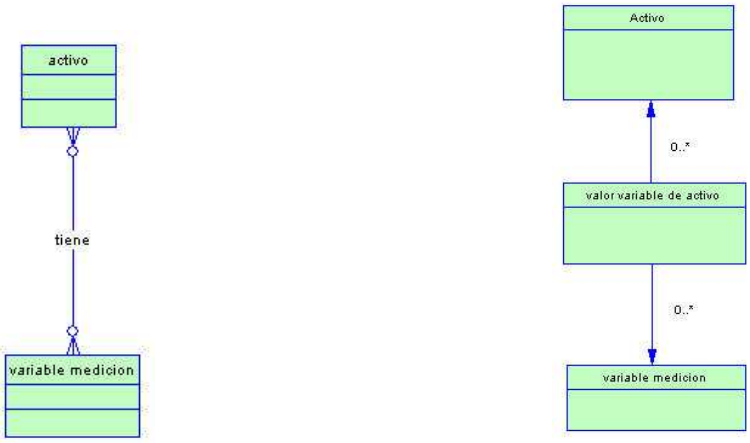


Figura Nº [14]: Relación Muchos – A – Muchos "Activo – A – VariableMedicion".

La relación muchos-a-muchos formada por las relaciones "TipoActivo" y "VariableMedicion", se transformó tal como muestra la siguiente figura.

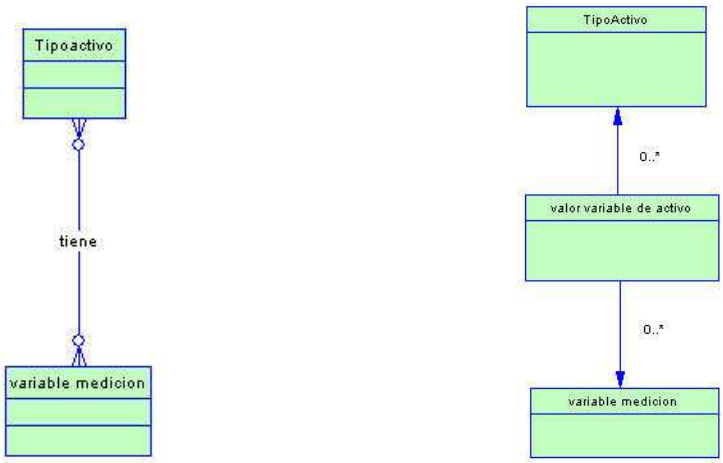


Figura Nº [15]: Relación Muchos – A – Muchos “TipoActivo – A – VariableMedicion”.

La relación muchos-a-muchos formada por las relaciones “ExtProveedor” y “TipoActivo”, se transformó tal como muestra la siguiente figura.

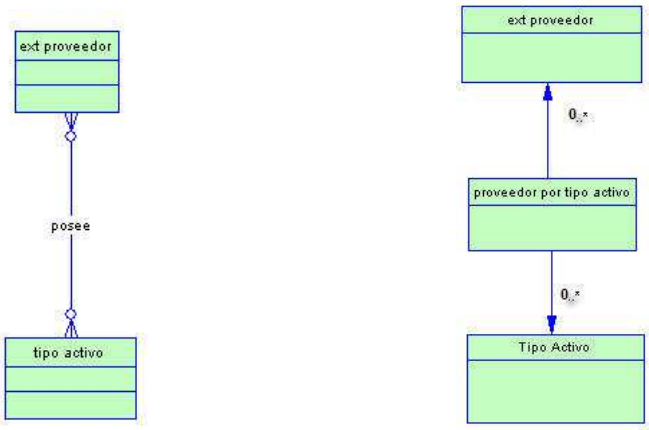


Figura Nº [16]: Relación Muchos – A – Muchos “ExtProveedor – A – TipoActivo”.

La relación muchos-a-muchos formada por las relaciones “Activo” y “TipoItemActivo”, se transformó tal como muestra la siguiente figura.

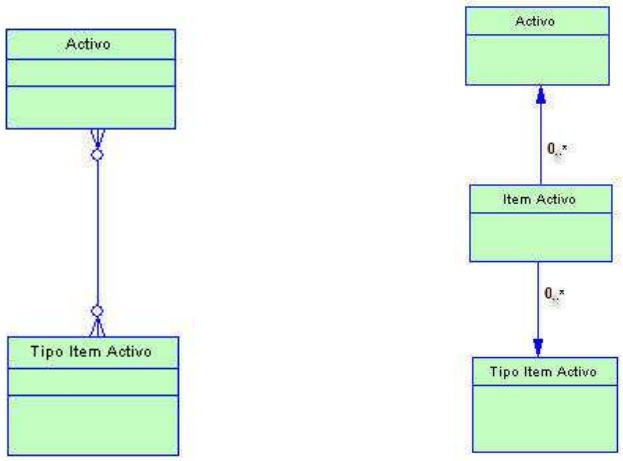


Figura Nº [17]: Relación Muchos – A – Muchos “Activo – A – TipoItemActivo”.

7.2.1.1.2 Relaciones “Muchos - a- Muchos” para módulo mantenciones

La relación muchos-a-muchos formada por las relaciones “ActividadTeorica” y “HitoMantencionEfectiva”, se transformó tal como muestra la siguiente figura.

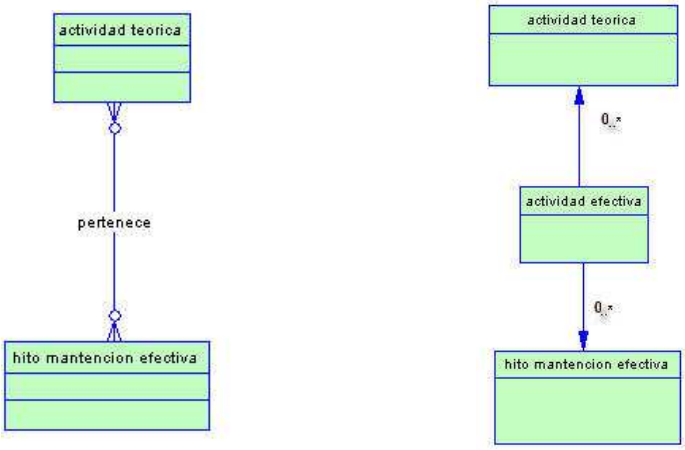


Figura Nº [18]: Relación Muchos – A – Muchos "ActividadTeorica – A – HitoMantencionEfectiva".

La relación muchos-a-muchos formada por las relaciones "ActividadTeorica" y "FrecuenciaTipoActivo", se transformó tal como muestra la siguiente figura.

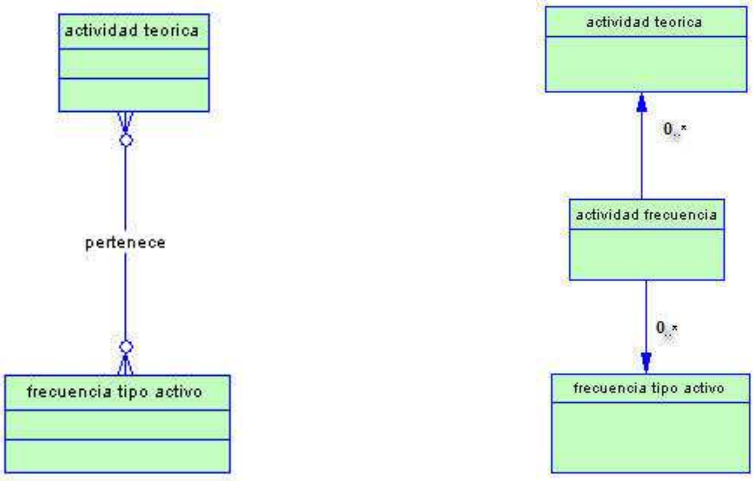


Figura Nº [19]: Relación Muchos – A – Muchos "ActividadTeorica – A – FrecuenciaTipoActivo".

7.2.1.1.3 Relaciones “Muchos - a- Muchos” Transversales para todos los módulos

La relación muchos-a-muchos formada por las relaciones “GrupoUsuarios” y “Usuario”, se transformó tal como muestra la siguiente figura.

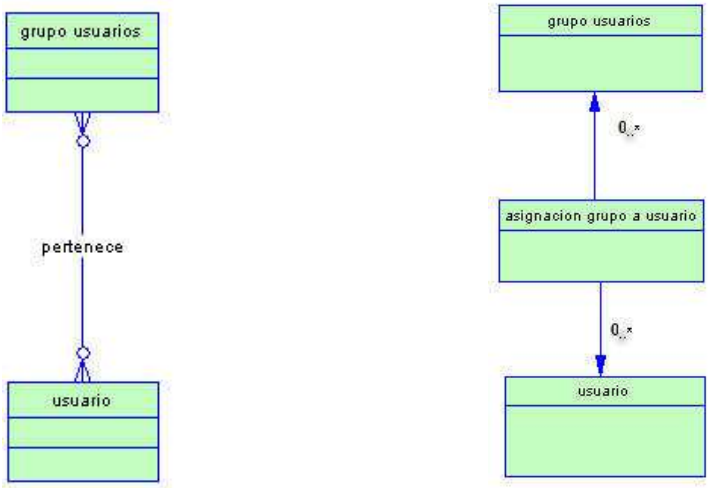


Figura Nº [20]: Relación Muchos – A – Muchos “GrupoUsuarios – A – Usuario”.

7.2.1.2 Relaciones “Complejas”

7.2.1.2.1 Relaciones “Complejas” módulo Activos

La relación compleja “VariableParaTipoActivo”, se transformó insertando una entidad intermedia, como muestra la siguiente figura.

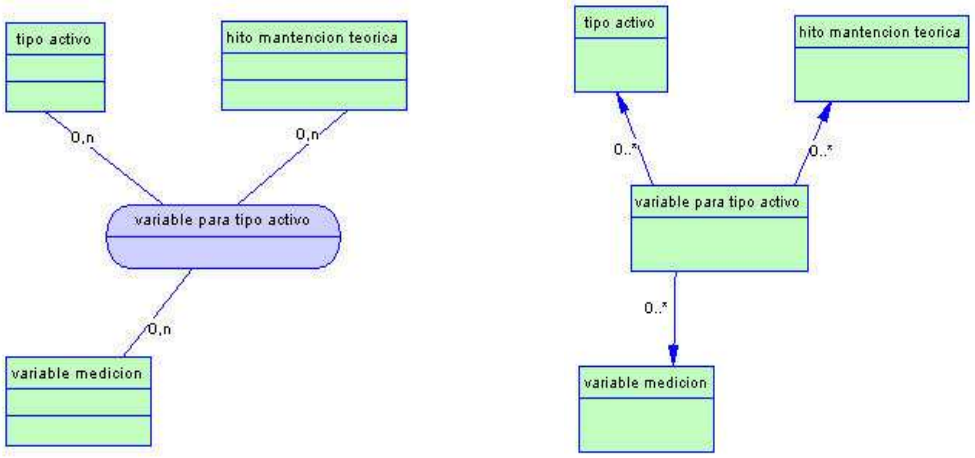


Figura Nº [21]: Relación Compleja "VariableParaTipoActivo"

7.2.1.2.2 Relaciones "Complejas" módulo mantenciones

La relación compleja "FrecuenciaTipoActivo", se transformó insertando una entidad intermedia, como muestra la siguiente figura.

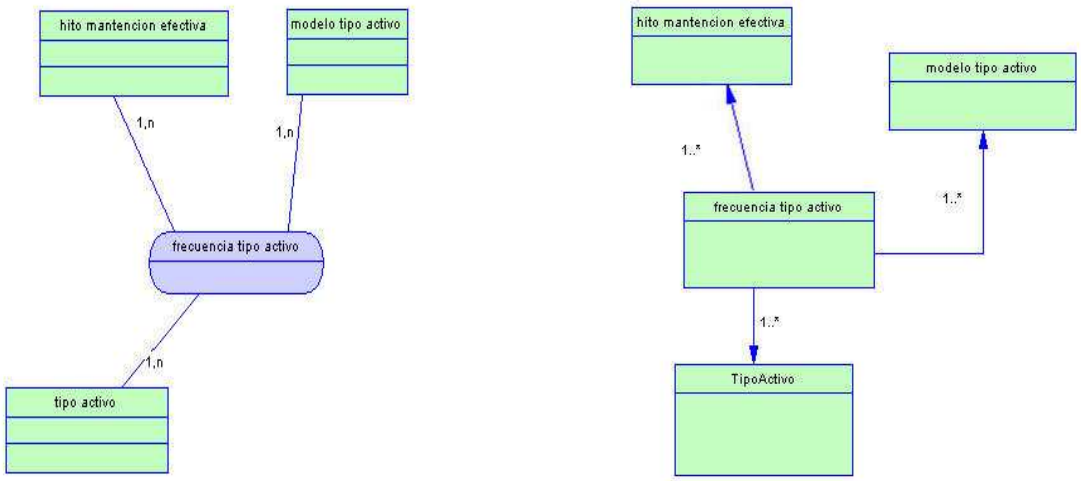


Figura Nº [22]: Relación Compleja "FrecuenciaTipoActivo"

7.2.2 Derivar relaciones del modelo de datos lógico

A continuación se presenta y describen las relaciones del modelo en lenguaje DBDL ("Database Definition Language" o Lenguaje de definición de base de datos):

Activo (Id_Area, Id_Activo, Codigo_Tipo_Activo, Id_Activo_Padre, Id_Marca_ Tipo_Activo, Id_Centro, Id_Modelo_Tipo_Activo, Id_Proveedor, Numero_sucursal, Id_Nombre_Por_Tipo_Activo, Nombre_Activo, Numero_Serie_Activo, Orden_Compra_Activo, Factura_Activo, Guia_Despacho_Activo, Fecha_Creacion_Activo, Descripcion_Activo, Estado_Activo_Oracle, Numero_Activo_Oracle, Estado_Activo_Promo, Fecha_Baja_Activo)
Primary Key (Id_Area, Id_Activo)
Foreign Key (Id_Area) **References** Area (Id_Area)
Foreign Key (Codigo_Tipo_Activo) **References** TipoActivo (Codigo_Tipo_Activo)
Foreign Key (Id_Marca_ Tipo_Activo) **References** MarcaTipoActivo (Id_Marca_Tipo_Activo)
Foreign Key (Id_Centro) **References** Centro (Id_Centro)
Foreign Key (Id_Modelo_Tipo_Activo) **References** ModeloTipoActivo (Id_Modelo_Tipo_Activo)
Foreign Key (Id_Proveedor) **References** ExtProveedor (Id_Proveedor)
Foreign Key (Numero_Sucursal) **References** ExtSucursal (Numero_Sucursal)
Foreign Key (Id_Nombre_Por_Tipo_Activo) **References** NombrePorTipoActivo (Id_Nombre_Por_Tipo_Activo)

TipoActivo (Codigo_Tipo_Activo, Codigo_Tipo_Activo_Padre, Nombre_Tipo_Activo, tiene_item_activo)
Primary Key (Codigo_Tipo_Activo)
Foreign Key (Codigo_Tipo_Activo_Padre) **References** TipoActivo (Codigo_Tipo_Activo_Padre)

Area (Id_Area, Nombre_Area, Centro_Costo_Area)
Primary Key (Id_Area)
Alternate Key (Nombre_Area)

Centro (Id_Centro, Id_Area, Nombre_Centro, Ubicacion_x, Ubicacion_y, Centro_Costo_Centro)
Primary Key (Id_Centro)
Foreign Key (Id_Area) **References** Area (Id_Area)

NombrePorTipoActivo (Id_Nombre_Por_Tipo_Activo, Codigo_Tipo_Activo, Nombre_Por_Tipo_Activo)
Primary Key (Id_Nombre_Por_Tipo_Activo)
Foreign Key (Codigo_Tipo activo) **References** TipoActivo (Codigo_Tipo activo)

VariableMedicion (Id_Variable_Medicion, Nombre_Variable_Medicion, activo, Descripcion_Variable_Medicion)
Primary Key (Id_Nombre_Por_Tipo_Activo)
Foreign Key (Codigo_Tipo_activo) **References** TipoActivo (Codigo_Tipo activo)
ExtProveedor (Id_Proveedor, Nombre_Proveedor, Direccion_Proveedor)
Primary Key (Id_Proveedor)
Alternate Key (Nombre_Proveedor)

ExtSucursal (Numero_Sucursal, Id_Proveedor, Nombre_Sucursal, Direccion_Sucursal)
Primary Key (Numero_Sucursal, Id_Proveedor)
Foreign Key (Id_Proveedor) **References** ExtProveedor (Id_Proveedor)

ModeloTipoActivo (Id_Modelo_Tipo_Activo, Id_Marca_Tipo_Activo, Nombre_Modelo_Tipo_Activo)
Primary Key (Id_Modelo_Tipo_Activo)
Foreign Key (Id_Marca_Tipo_Activo) **References** MarcaTipoActivo (Id_Marca_Tipo_Activo)

MarcaTipoActivo (Id_Marca_Tipo_Activo, Nombre_Marca_Tipo_Activo)
Primary Key (Id_Marca_Tipo_Activo)
Alternate Key (Nombre_Proveedor)

ItemActivo (Id_item_activo, Id_Area, Id_Activo, Codigo_Tipo_Item_Activo, Descripcion_Item_Activo, Numero_Factura_Item_Activo, Guia_Despacho_Item_Activo, Orden_Compra_Item_Activo, Longitud_Item_Activo, Diametro_Item_Activo, Diametro_Pulg_Item_Activo, Unidades_Item_Activo, Volumen_Item_Activo, Peso_t_Item_Activo, Fecha_Creacion_Item_Activo, Numero_Activo_Oracle_Item_Activo,

Id_Proveedor_Item_Activo)
Primary Key (Id_item_activo, Id_Area, Id_Activo)
Foreign Key (Id_Area) **References** Activo (Id_Area)
Foreign Key (Id_Activo) **References** Activo (Id_Activo)

TipoItemActivo (Codigo_Tipo_Item_Activo, Nombre_Tipo_Item_Activo, Tiene_Longitud_Item_Activo, Tiene_Diametro_Item_Activo, Tiene_Diametro_Pulg_Item_Activo, Tiene_Unidades_Item_Activo, Tiene_Volumen_Item_Activo, Tiene_Peso_t_Item_Activo)
Primary Key (Codigo_Tipo_Item_Activo)
Alternate Key (Nombre_Tipo_Item_Activo)

RecepcionParcialActivo (Id_Orden_Recepcion_Activo, Id_Area, Id_Activo, Fecha_Recepcion_Parcial_Activo, Descripcion_Recepcion_Parcial_Activo)
Primary Key (Id_Orden_Recepcion_Activo)
Foreign Key (Id_Area) **References** Activo (Id_Area)
Foreign Key (Id_Activo) **References** Activo (Id_Activo)

HitoMantencionTeorica (Id_Mantencion_Teorica_Activo, Codigo_Tipo_Mantencion_Activo, Id_Variable_Medicion, Codigo_Tipo_Activo, Descripcion_Mantencion_Teorica, Valor_Variable_Mantencion_Teorica, Id_Modelo_Tipo_Activo)
Primary Key (Id_Mantencion_Teorica_Activo)
Foreign Key (Codigo_Tipo_Mantencion_Activo) **References** TipoMantencionActivo (Codigo_Tipo_Mantencion_Activo)
Foreign Key (Id_Variable_Medicion) **References** VariableParaTipoActivo (Id_Variable_Medicion)
Foreign Key (Codigo_Tipo_Activo) **References** VariableParaTipoActivo (Codigo_Tipo_Activo)
Foreign Key (Id_Modelo_Tipo_Activo) **References** ModeloTipoActivo (Id_Modelo_Tipo_Activo)

DetalleMantencion (Id_Detalle_Mantencion, Id_Mantencion_Teorica_Activo, Descripcion_Detalle_Mantencion)
Primary Key (Id_Detalle_Mantencion)
Foreign Key (Id_Mantencion_Teorica_Activo) **References** HitoMantencionTeorica (Id_Mantencion_Teorica_Activo)

TipoMantencionActivo (Codigo_Tipo_Mantencion_Activo, Nombre_Tipo_Mantencion)

Primary Key (Codigo_Tipo_Mantencion_Activo)
Alternate Key (Nombre_Tipo_Mantencion)

ActividadTeorica (Id_Actividad_Teorica, Id_Mantencion_Teorica_Activo, Descripcion_Actividad)
Primary Key (Id_Actividad_Teorica)
Foreign Key (Id_Mantencion_Teorica_Activo) **References** HitoMantencionTeorica (Id_Mantencion_Teorica_Activo)

HitoMantencionEfectiva (Id_Mantencion_Efectiva_Activo, Id_Area, Id_Activo, Id_Mantencion_Teorica_Activo, Codigo_Tipo_Activo, Fecha_Mantencion_Activo, Fecha_Ejecucion_Activo, Estado_Mantencion_Activo, Materiales_Mantencion_Activo, Observaciones_Mantencion_Activo, Descripcion_Mantencion_Activo, Costo_Real_Mantencion_Activo, Frecuencia_Dias, Id_Modelo_Tipo_Activo)
Primary Key (Id_Mantencion_Efectiva_Activo)
Foreign Key (Id_Area) **References** Activo (Id_Area)
Foreign Key (Id_Activo) **References** Activo (Id_Activo)
Foreign Key (Id_Mantencion_Teorica_Activo) **References** HitoMantencionTeorica (Id_Mantencion_Teorica_Activo)
Foreign Key (Codigo_Tipo_Activo) **References** FrecuenciaTipoActivo (Codigo_Tipo_Activo)

ActividadEfectiva (Id_Actividad_Efectiva, Id_Actividad_Teorica, Id_Mantencion_Efectiva_Activo, Estado_Actividad)
Primary Key (Id_Actividad_Efectiva)
Foreign Key (Id_Actividad_Teorica) **References** ActividadTeorica (Id_Actividad_Teorica)
Foreign Key (Id_Mantencion_Efectiva_Activo) **References** HitoMantencionEfectiva (Id_Mantencion_Efectiva_Activo)

ActividadFrecuencia (Id_Actividad_Teorica, Codigo_Tipo_Activo, Frecuencia_Dias, Id_Modelo_Tipo_Activo, Se_Realiza)
Primary Key (Id_Actividad_Teorica)
Foreign Key (Codigo_Tipo_Activo) **References** FrecuenciaTipoActivo (codigo_tipo_activo)
Foreign Key (Frecuencia_Dias) **References** FrecuenciaTipoActivo (Frecuencia_Dias)
Foreign Key (Id_Modelo_Tipo_Activo) **References** FrecuenciaTipoActivo (id_modelo_tipo_activo)

FrecuenciaTipoActivo (Codigo_Tipo_Activo, Frecuencia_Dias, Id_Modelo_Tipo_Activo)
Primary Key (Codigo_Tipo_Activo, Frecuencia_Dias, Id_Modelo_Tipo_Activo)

Usuario (Username, Password, Id_Area, Nombre_Usuario, Estado_Usuario, Correo_Electronico)
Primary Key (Username)
Foreign Key (Id_Area) **References** Area (Id_Area)

GrupoUsuario (Codigo_Grupo, Glosa_Grupo, Privilegios)
Primary Key (Codigo_Grupo)
Alternate Key (Codigo_Grupo)

ParametroSistema (Nombre_Parametro, Valor_Parametro)
Primary Key (Nombre_Parametro)
Alternate Key (Valor_Parametro)

Secuencia (Codigo_Secuencia, Proximo_Correlativo)
Primary Key (Codigo_Secuencia)
Alternate Key (Proximo_Correlativo)

Log PROMO (Id_Log, Fecha_Hora_Log, Descripcion_Log, Username, Id_Area, Id_Activo)
Primary Key (Id_Log)
Alternate Key (Fecha_Hora_Log)

7.2.3 Validar el Modelo Utilizando Normalización

Este punto de la metodología busca asegurar que el modelo resultante sea consistente, con bajos niveles de redundancia y máxima estabilidad.

Para obtener estos resultados se pretende analizar y validar los grupos de

atributos en cada relación, para ello se realizo la comprobación de las siguientes formas normales:

7.2.3.1 Primera Forma Normal (1FN)

Un modelo lógico se encuentra en primera forma normal, cuando se cumplen ciertas normas que dicen relación con la inexistencia de grupos de datos no repetitivos o multivalorados, en caso de haberlos se deben separar para formar sus propias relaciones.

De acuerdo al análisis presentado se entiende que el modelo se encuentra en primera forma normal, ya que las relaciones cumplen con las características citadas anteriormente.

7.2.3.2 Segunda Forma Normal (2FN)

Se asume que un modelo se encuentra en segunda forma normal, cuando cumple las reglas de la primera forma normal y todos sus atributos que no son claves dependen por completo de la clave primaria. De acuerdo con esta definición, cada tabla que tiene un atributo único como clave, esta en segunda

forma normal. Por ende, el análisis que se presenta a continuación sólo se baso en las tablas que poseen claves compuestas.

Activo (Id_Area, Id_Activo, Codigo_Tipo_Activo, Id_Activo_Padre, Id_Marca_ Tipo_Activo, Id_Centro, Id_Modelo_Tipo_Activo, Id_Proveedor, Numero_sucursal, Id_Nombre_Por_Tipo_Activo, Nombre_Activo, Numero_Serie_Activo, Orden_Compra_Activo, Factura_Activo, Guia_Despacho_Activo, Fecha_Creacion_Activo, Descripcion_Activo, Estado_Activo_Oracle, Numero_Activo_Oracle, Estado_Activo_Promo, Fecha_Baja_Activo)
Entonces: Nombre_Activo, Numero_Serie_Activo, Orden_Compra_Activo, Factura_Activo, Guia_Despacho_Activo, Fecha_Creacion_Activo, Descripcion_Activo, Estado_Activo_Oracle, Numero_Activo_Oracle, Estado_Activo_Promo, Fecha_Baja_Activo. Dependen completamente de la clave primaria (Id_Area, Id_Activo, Codigo_Tipo_Activo, Id_Activo_Padre, Id_Marca_ Tipo_Activo, Id_Centro, Id_Modelo_Tipo_Activo, Id_Proveedor, Numero_sucursal, Id_Nombre_Por_Tipo_Activo) y no solo de una parte de ella.

ItemActivo (Id_Area, Id_Activo, Id_Item_Activo, Codigo_Tipo_Item_Activo, Descripcion_Item_Activo, Numero_Factura_Item_Activo, Guia_Despacho_Item_Activo, Orden_Compra_Item_Activo, Longitud_Item_Activo, Diametro_Item_Activo, Diametro_Pulg_Item_Activo, Unidades_Item_Activo, Volumen_Item_Activo, Peso_T_Item_Activo, Fecha_Creacion_Item_Activo, Numero_Activo_Oracle_Item_Activo, Id_Proveedor_Item_Activo)
Entonces: Descripcion_Item_Activo, Numero_Factura_Item_Activo, Guia_Despacho_Item_Activo, Orden_Compra_Item_Activo, Longitud_Item_Activo, Diametro_Item_Activo, Diametro_Pulg_Item_Activo, Unidades_Item_Activo, Volumen_Item_Activo, Peso_T_Item_Activo, Fecha_Creacion_Item_Activo, Numero_Activo_Oracle_Item_Activo, Id_Proveedor_Item_Activo. Dependen completamente de la clave primaria (Id_Area, Id_Activo, Id_Item_Activo, Codigo_Tipo_Item_Activo) y no solo de una parte de ella.

ExtSucursal (Numero_Sucursal, Id_Proveedor, Nombre_Sucursal, Direccion_Sucursal)
Entonces: Nombre_Sucursal, Direccion_Sucursal. Dependen completamente de la clave primaria (Numero_Sucursal, Id_Proveedor) y no solo de una parte de ella.

De acuerdo a la definición de 2FN y al análisis de las tablas que poseen claves primarias compuestas, se establece que el modelo se encuentra en segunda forma normal.

7.2.3.3 Tercera Forma Normal (3FN)

Según la definición de 3FN, un modelo se encuentra en tercera forma normal si y solo si se encuentra en 1NF y 2NF, y en la cual los atributos que no son clave primaria no son transitivamente dependiente de la clave primaria.

Dependencias funcionales:

Activo
Id_Area, Id_Activo → Codigo_Tipo_Activo, Id_Activo_Padre, Id_Marca_ Tipo_Activo, Id_Centro, Id_Modelo_Tipo_Activo, Id_Proveedor, Numero_sucursal, Id_Nombre_Por_Tipo_Activo, Nombre_Activo, Numero_Serie_Activo, Orden_Compra_Activo, Factura_Activo, Guia_Despacho_Activo, Fecha_Creacion_Activo, Descripcion_Activo, Estado_Activo_Oracle, Numero_Activo_Oracle, Estado_Activo_Promo, Fecha_Baja_Activo

TipoActivo
Codigo_Tipo_Activo → Codigo_Tipo_Activo_Padre, Nombre_Tipo_Activo, tiene_item_activo

Area
Id_Area → Nombre_Area, Centro_Costo_Area

Centro
Id_Centro → Id_Area, Nombre_Centro, Ubicacion_x, Ubicacion_y, Centro_Costo_Centro

NombrePorTipoActivo
Id_Nombre_Por_Tipo_Activo → Codigo_Tipo_Activo, Nombre_Por_Tipo_Activo

VariableMedicion
Id_Variable_Medicion → Nombre_Variable_Medicion,
Descripcion_Variable_Medicion

ExtProveedor
Id_Proveedor → Nombre_Proveedor, Direccion_Proveedor
ExtSucursal
Numero_Sucursal → Id_Proveedor, Nombre_Sucursal, Direccion_Sucursal

ModeloTipoActivo
Id_Modelo_Tipo_Activo → Id_Marca_Tipo_Activo, Nombre_Modelo_Tipo_Activo

MarcaTipoActivo
Id_Marca_Tipo_Activo → Nombre_Marca_Tipo_Activo

ItemActivo
Id_item_activo, Id_Area, Id_Activo → Codigo_Tipo_Item_Activo,
Descripcion_Item_Activo, Numero_Factura_Item_Activo,
Guia_Despacho_Item_Activo, Orden_Compra_Item_Activo,
Longitud_Item_Activo, Diametro_Item_Activo, Diametro_Pulg_Item_Activo,
Unidades_Item_Activo, Volumen_Item_Activo, Peso_t_Item_Activo,
Fecha_Creacion_Item_Activo, Numero_Activo_Oracle_Item_Activo,
Id_Proveedor_Item_Activo

TipoItemActivo
Codigo_Tipo_Item_Activo → Nombre_Tipo_Item_Activo,
Tiene_Longitud_Item_Activo, Tiene_Diametro_Item_Activo,
Tiene_Diametro_Pulg_Item_Activo, Tiene_Unidades_Item_Activo,
Tiene_Volumen_Item_Activo, Tiene_Peso_t_Item_Activo

RecepcionParcialActivo
Id_Orden_Recepcion_Activo → Id_Area, Id_Activo,
Fecha_Recepcion_Parcial_Activo, Descripcion_Recepcion_Parcial_Activo

HitoMantencionTeorica
Id_Mantencion_Teorica_Activo → Codigo_Tipo_Mantencion_Activo,
Id_Variable_Medicion, Codigo_Tipo_Activo, Descripcion_Mantencion_Teorica,

Valor_Variable_Mantencion_Teorica, Id_Modelo_Tipo_Activo

DetalleMantencion
Id_Detalle_Mantencion → Id_Mantencion_Teorica_Activo, Descripcion_Detalle_Mantencion

TipoMantencionActivo
Codigo_Tipo_Mantencion_Activo → Nombre_Tipo_Mantencion

ActividadTeorica
Id_Actividad_Teorica → Id_Mantencion_Teorica_Activo, Descripcion_Actividad

HitoMantencionEfectiva
Id_Mantencion_Efectiva_Activo → Id_Area, Id_Activo, Id_Mantencion_Teorica_Activo, Codigo_Tipo_Activo, Fecha_Mantencion_Activo, Fecha_Ejecucion_Activo, Estado_Mantencion_Activo, Materiales_Mantencion_Activo, Observaciones_Mantencion_Activo, Descripcion_Mantencion_Activo, Costo_Real_Mantencion_Activo, Frecuencia_Dias, Id_Modelo_Tipo_Activo

ActividadEfectiva
Id_Actividad_Efectiva → Id_Actividad_Teorica, Id_Mantencion_Efectiva_Activo, Estado_Actividad

ActividadFrecuencia
Id_Actividad_Teorica → Codigo_Tipo_Activo, Frecuencia_Dias, Id_Modelo_Tipo_Activo, Se_Realiza

FrecuenciaTipoActivo
Codigo_Tipo_Activo → Frecuencia_Dias, Id_Modelo_Tipo_Activo
Frecuencia_Dias → Codigo_Tipo_Activo, Id_Modelo_Tipo_Activo
Id_Modelo_Tipo_Activo → Codigo_Tipo_Activo, Frecuencia_Dias

Usuario
Username → Password, Id_Area, Nombre_Usuario, Estado_Usuario, Correo_Electronico

GrupoUsuario
Codigo_Grupo → Glosa_Grupo, Privilegios

ParametroSistema
Nombre_Parametro → Valor_Parametro

Secuencia
Codigo_Secuencia → Proximo_Correlativo

LogPROMO
Id_Log → Fecha_Hora_Log, Descripcion_Log, Username, Id_Area, Id_Activo

Se concluye que, con el análisis anterior y según la definición 3FN, el modelo se encuentra en tercera forma normal.

7.2.4 Transacciones de Usuario

Secuencia atómica de acciones en la base de datos iniciada por un usuario, a continuación se detallarán algunas transacciones definidas por el usuario. Las entidades de color morado, representan relaciones N:N y están definidas con otro color sólo para efectos visuales.

- Configurar, según necesidades del usuario, ítem de activos, para activos previamente determinados para poseer estas características.

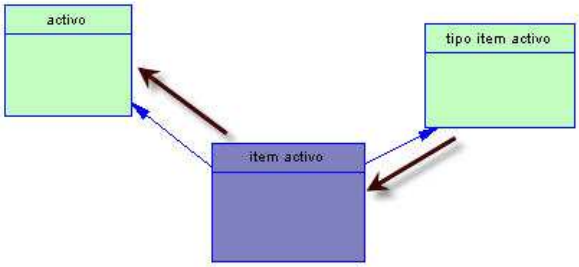


Figura Nº [23]: Transacción configurar Activo e Item Activo.

- Configurar, según necesidad del usuario, la planificación y ejecución de un plan de mantención.

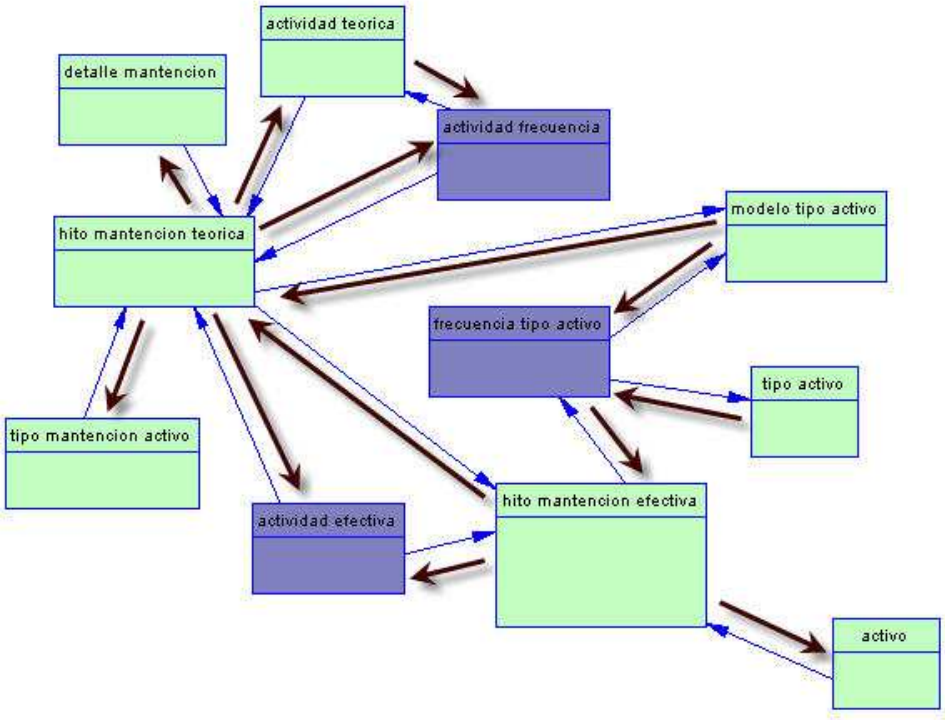


Figura Nº [24]: Transacción configurar plan de mantención de un activo

- Ejecutar acciones por parte del usuario, para generar reporte de inventario de activos.

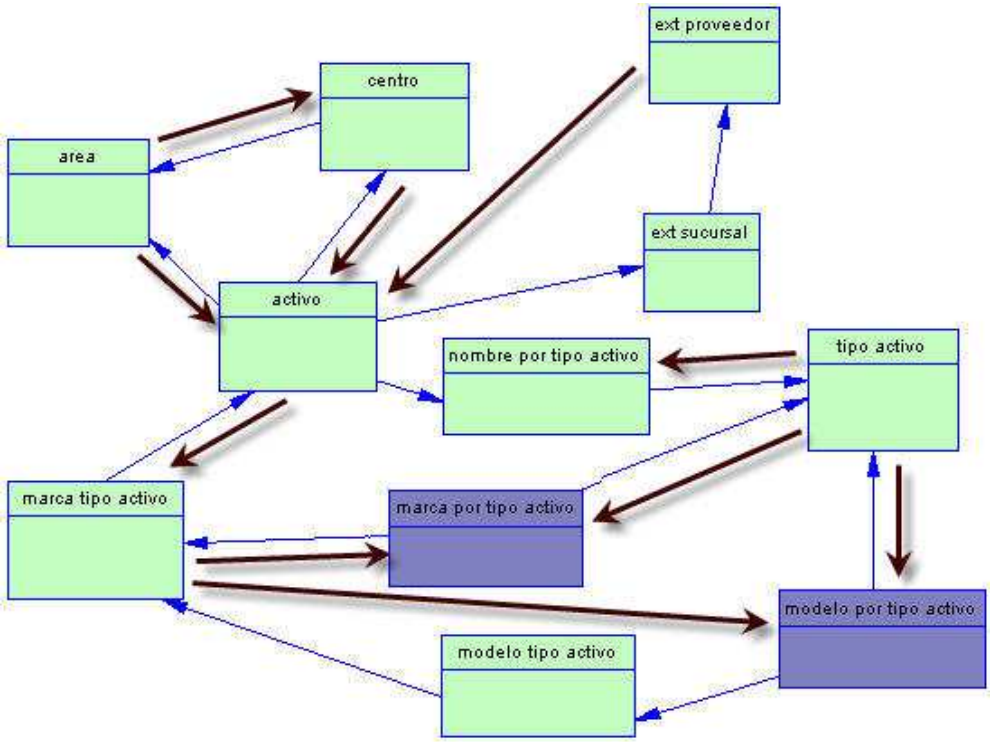


Figura Nº [25]: Transacción configurar generación reporte inventario activo

7.2.5 Diagrama ER Final

Se entiende que el análisis presentado anteriormente da cuenta de todos los atributos y características que conforman las entidades y relaciones, por ende sólo se presentarán las entidades con las relaciones.

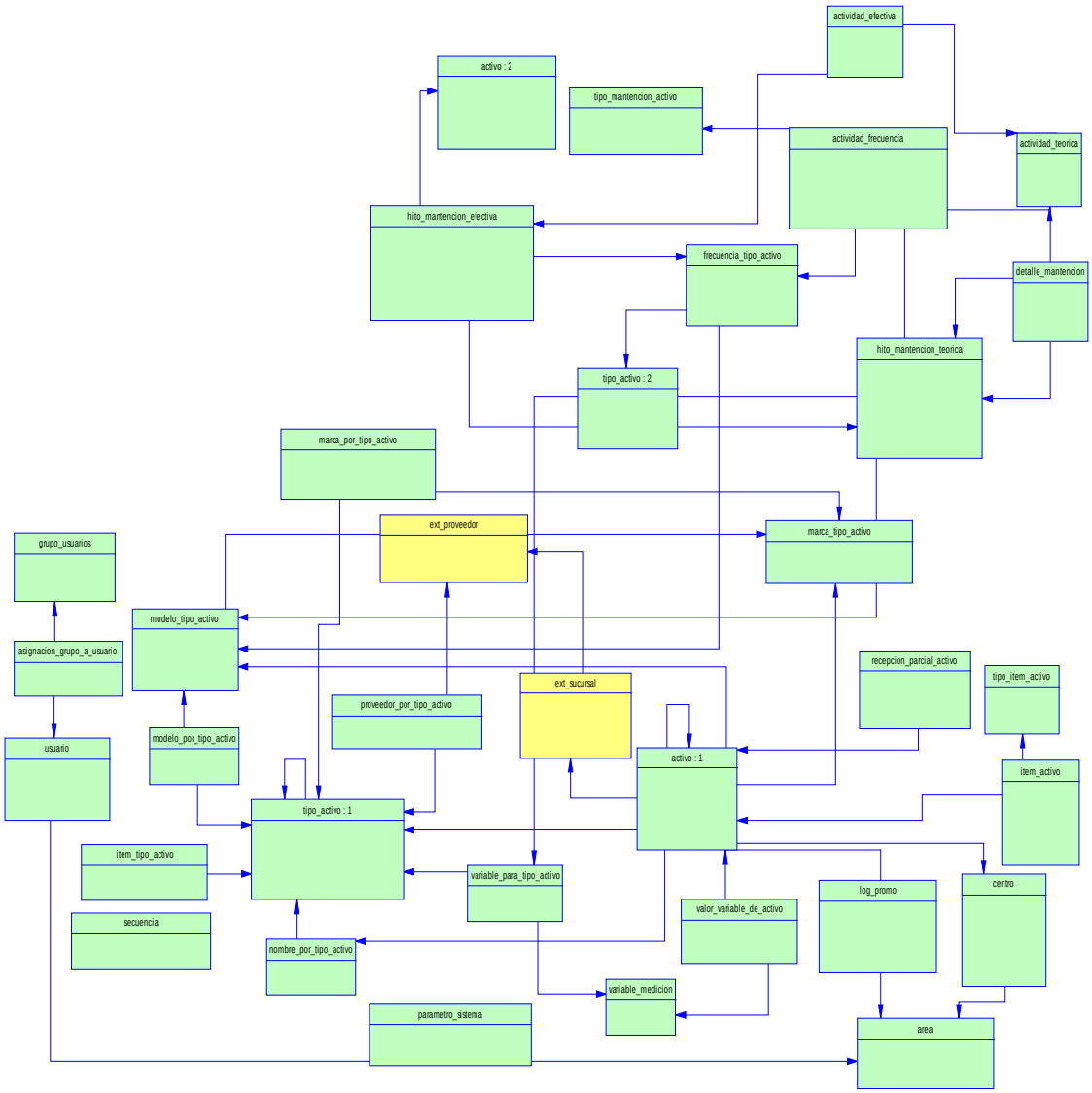


Diagrama Nº [18]: Modelo ER Final

7.2.6 Definir restricciones de integridad

Con esto se busca que una vez implementada la base de datos, esta entregue consistencia, es decir, muestre una estabilidad en el 100% de los casos en que sea requerida.

7.2.6.1 Integridad referencial

Tanto para actualizaciones y eliminaciones en tablas padres, se realizó en "cascada", por lo tanto para la implementación en SQL SERVER 2005, esta se realizó en forma declarativa, es decir, las relaciones que existen entre las llamadas entidades padres y entidades hijas, se establecerá la siguiente sintaxis en las relaciones:

```
alter table activo
add constraint FK_ACTIVO_PERTENECE_TIPO_ACTIVO foreign key
(codigo_tipo_activo)
references tipo_activo (codigo_tipo_activo)
on update cascade on delete cascade -- sintaxis que establece integridad
referencial
go
```

7.2.6.2 Restricciones de la empresa

Entre otras restricciones de la empresa, se puede destacar la NO eliminación de activos de la entidad activo, permitiendo así mantener un histórico, que permitiría ser utilizado en el futuro, para distintos análisis por parte de los administradores del sistema entre otros.

Además, se puede mencionar la restricción que se realiza en el ingreso de valores preestablecidos para los distintos “estados”, que puede tener un activo, y que pertenecen a la entidad activo, estos atributos son: “estado_activo_oracle” y “estado_activo_Promo”. Obviamente el usuario puede ingresar solo valores válidos y que son visualizados en la interfaz que propone el sistema para dicho usuario, pero para una mayor seguridad esta validación se realizó en la base de datos a través del siguiente código:

Restricción para validar el estado del activo, en el sistema Oracle
Nombre de la restricción: **CKC_ESTADO_ACTIVO_ORA_ACTIVO**
Expresión de la restricción:
([estado_activo_oracle] is null or ([estado_activo_oracle] =
'NOINGRESADO' or [estado_activo_oracle] = 'INGRESADO'))

Restricción para validar el estado del activo en el sistema

Nombre de la restricción: **CKC_ESTADO_ACTIVO_PRO_ACTIVO**

Expresión de la restricción:

([estado_activo_promo] is null or ([estado_activo_promo] = 'ELIMINADO' or [estado_activo_promo] = 'ACTIVO') or [estado_activo_promo] = 'BAJA')

Estas restricciones permiten que valores críticos que determinan el posterior uso de un activo, tengan un manejo bastante restrictivo y seguro.

7.3 Construir y validar el modelo de datos lógico global

De acuerdo a lo expuesto y analizado en los puntos anteriores, se entiende que el modelo tiene un nivel de refinamiento aceptable, y además representa la globalidad de los datos que interactúan en la problemática abordada, en este proyecto de tesis, por lo expuesto anteriormente este punto no es aplicable.

7.4 Traducir el modelo lógico global para el DBMS

El DBMS utilizado para esta implementación fue “Microsoft SQL Server Management Studio”, en su versión 2005.

7.4.1 Diseñar las relaciones bases para el DBMS especificado.

Para esto se utilizo información producida durante el diseño lógico: esquema lógico global y el diccionario de datos.

El detalle del diseño de las relaciones, se detalló en el punto 7.2.6.1 (Integridad Referencial)

7.4.2 Diseñar las restricciones de la empresa para el DBMS especificado.

Las actualizaciones que se realizan sobre las relaciones de la base de datos deben observar ciertas restricciones que imponen las reglas de negocio de la empresa. En este caso se crearon dichas reglas de negocio en el DBMS especificado en el punto anterior 7.4 y el detalle de dichas reglas se especificó en el punto 7.2.6.2.

Finalmente, cabe mencionar que no fue necesaria la utilización de triggers para tratar condiciones inválidas que pueden presentarse para el DBMS específico.

7.4.3 Implementación

El desarrollo del sistema se baso en el paradigma de la orientación a objetos, que implica beneficios en términos de modularización, reducción de complejidad, reusabilidad de la lógica etc. Por otro la lado la base de datos implementada pertenece al paradigma de la bases de datos relacionales, que son efectivamente útiles para recuperar y hacer análisis de información. Para entender mejor en que consisten cada uno de estos paradigmas se puede concluir que están pensados con fines distintos: “Los objetos modelan el estado del negocio durante la ejecución de procesos transaccionales (On line

Transaction Processing u OLTP), en tanto que las tablas de una base de datos relacional modelan los distintos estados del negocio a lo largo del tiempo con afines analíticos (On line Analitical Processing u OLAP).

Esto se traduce en limitaciones para convertir objetos en registros y viceversa. La literatura informática ha englobado este problema como Desajuste por Impedancia (o Impedance Mismatch).

Por esto se hizo imprescindible la implementación de una capa de acceso a datos que permitiera separar completamente esta lógica, con la capa de presentación, a esto se le conoce como “Objetos de Acceso a Datos”, o más comúnmente DAOs. Esta práctica permite desacoplar la lógica de lidiar con el “Desajuste por Impedancia”, de la lógica funcional de la aplicación. Además con esto se logra aumentar la independencia de las capas, con lo cual el sistema tiende a cumplir con el “Modelo de tres capas”, el cual propone separar capa de datos (SQL Server Management Studio), lógica de negocios (procedimientos almacenados), capa de presentación (aplicación PROMO).

Este proyecto de Tesis se basó en tratar de entender claramente donde las bases de datos son más hábiles que los objetos y viceversa. Y sacar el máximo provecho a eso. Por esto se considero que la lógica de acceso a datos era muy

intensiva, por lo que se decidió codificar dicha lógica en la forma de procedimientos almacenados (Stored Procedures) directamente en la base de datos. Para lograr este objetivo se tuvieron que emplear instrucciones o características especiales del DBMS utilizado y/o desarrollar algoritmos que cumpliesen con los requerimientos planteados.

Para implementar los procedimientos almacenados (SP), se muestra a continuación la estructura que se utilizó:

```
/*****************************************************************************************
*
Nombre:                 Nombre del SP
Desarrollador:          Nombre del autor
Parámetros:             Descripción de los parámetros del SP
Descripción:            Descripción de lo que realiza el SP
Ultima Modificación:    Fecha de creación o última codificación
*****************************************************************************************
/

Create PROCEDURE dbo.Nombre del SP
@ildActivo d_correlativo  Declaración de parámetros y su dominio
AS
DECLARE @ErrorSave d_identificador Declaracion de variables que utilizara el SP
BEGIN

LOGICA HA IMPLEMENTAR PARA REGLA DE NEGOCIO: esta puede contener todas las características que ofrece el DBMS específico seleccionado. Para cumplir los requerimientos planteados, se implementaron sentencias tales como: algoritmos recursivos, Cursores, ciclos, ejecución y “llamadas”, a otros SP o funciones, condiciones IF, y todas las demás funcionalidades que ofrece Transact-SQL.
```

```
IF (@@ERROR <> 0)
    BEGIN
        SET @ErrorSave = @@ERROR

        RETURN @ErrorSave
     END
```

Tabla Nº [7]: Formatos Procedimientos almacenados

7.5 Diseñar representación Física

Uno de los objetivos principales del diseño físico es almacenar los datos de modo eficiente. Para medir la eficiencia hay varios factores que se deben tener en cuenta.

7.5.1 Analizar transacciones

Cabe señalar que el sistema se encuentra terminado y entregado, pero aún no se encuentra totalmente instalado en todos los centros que pertenecen a la empresa salmonera, lo cual implica que un análisis de transacciones tomando sólo en consideración uno o dos centros, estaría fuera de la realidad y se obtendrían resultados que obviamente no representarían el desempeño del sistema en producción e implementado en el 100% de los centros.

Claro esta, que después de realizar las pruebas correspondientes, el diseño físico no será el definitivo, si no que habrá que monitorear para observar sus prestaciones e ir ajustándolo.

7.5.2 Elegir organización de archivos

La implementación del modelo de datos, se realizó utilizando el sistema de archivos otorgado por defecto por SQL Server Management Studio 2005.

7.5.3 Elegir índices secundarios

Se mantuvieron las mismas claves alternas señaladas en el punto 7.1.5, como índices secundarios.

7.5.4 Introducción de redundancia controlada

A menudo, las base de datos normalizadas no proporcionan la máxima eficiencia, con lo que es necesario “volver atrás”, y desnormalizar algunas relaciones, sacrificando los beneficios de la normalización para mejorar las prestaciones.

En el modelo de datos presentado, se evaluó desnormalizar relaciones pero finalmente se estimó, no desnormalizar algunas relaciones por las siguientes razones:

- El sistema entrega prestaciones aceptables.

- La desnormalización hace que la implementación sea más compleja.
- La desnormalización hace que se sacrifique la flexibilidad.
- La desnormalización puede hacer que los accesos a datos sean más rápidos, pero hace más complejas las actualizaciones.

7.5.5 Estimar los requerimientos de espacio en disco

No se han realizado pruebas exhaustivas de espacio en disco, ya que la alta disponibilidad en cuanto a capacidad de almacenamiento de información que proveen los servidores, que ofrece la empresa salmonera, permite que este dato no sea relevante para un análisis minucioso.

7.6 Diseñar mecanismos de seguridad

Los datos constituyen un recurso esencial para la empresa, por lo tanto la seguridad es de vital importancia.

7.6.1 Diseñar vistas de usuario

Las vistas, además de preservar la seguridad, mejoran la independencia de datos, reducen la complejidad y permiten que los usuarios vean los datos en el

formato deseado. Las transacciones tienen permisos totales para los miembros del grupo “Administrador “, de la base de datos. El acceso a los datos por parte de usuarios no miembros de dicho grupo estará supeditado a normas de seguridad propias de la empresa.

7.6.2 Diseñar reglas de acceso

Las reglas de acceso para usuarios que se deseen realizar operaciones directamente en la base de datos, deberán pertenecer o ser miembros del grupo de “Administrador”, y ser usuarios validos de la máquina Windows donde resida el servidor de base de datos. Con esto el usuario “Administrador”, tendrá permisos totales sobre todas las transacciones (inserciones, eliminaciones, actualizaciones).

7.7 Monitoreo y refinamiento del sistema operacional.

De acuerdo a lo mencionado en el punto 7.5.1, y la no implementación total del sistema en el 100% de los centros pertenecientes a la empresa, es que este punto aún no aplica.

8 Construcción

Para introducirse en este capitulo, se señalara como principales herramientas de desarrollo, el lenguaje nativo de SQL Server, Transact–SQL y el lenguaje compilado C#, que utiliza el FrameWork v1.1.4322 y que se engloba en la plataforma Microsoft Visual Studio 2003.

La implementación física de la base de datos, se logrará mediante el script generado en la etapa de diseño a través de la herramienta Power Designer 10.

Con el script generado, se crearán todas las tablas con sus respectivos índices y relaciones necesarias para implementar la base de datos que soportará finalmente al sistema.

Como demostración de lo realizado en el proyecto se implementarán los casos de uso: “Creación Plan Mantención para Modelo”, y “Ejecutar Plan de Mantención de un Activo”, junto con las piezas de software que permiten la ejecución e integración de los mencionados casos de uso que se describen en detalle en la sección Análisis, módulo mantenciones del presente documento.

Los diagramas de secuencia que se mostrarán a continuación establecerán una interacción ordenada según una secuencia temporal de eventos. En particular, muestran los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo.

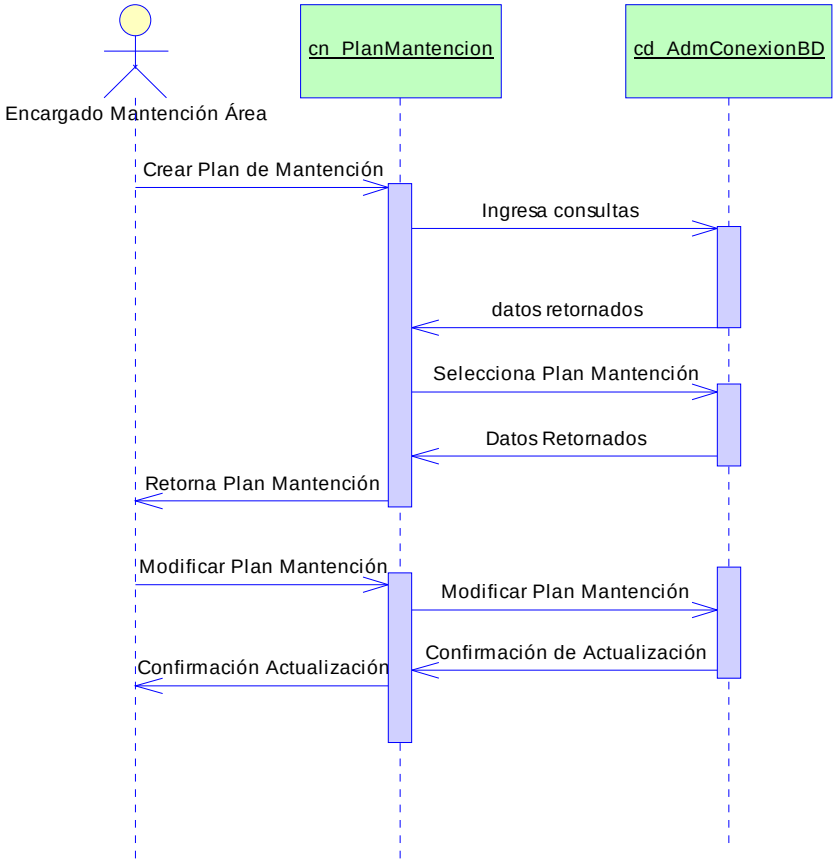


Diagrama Nº [19]: Diagrama Secuencia Creación Plan Mantención para Modelo

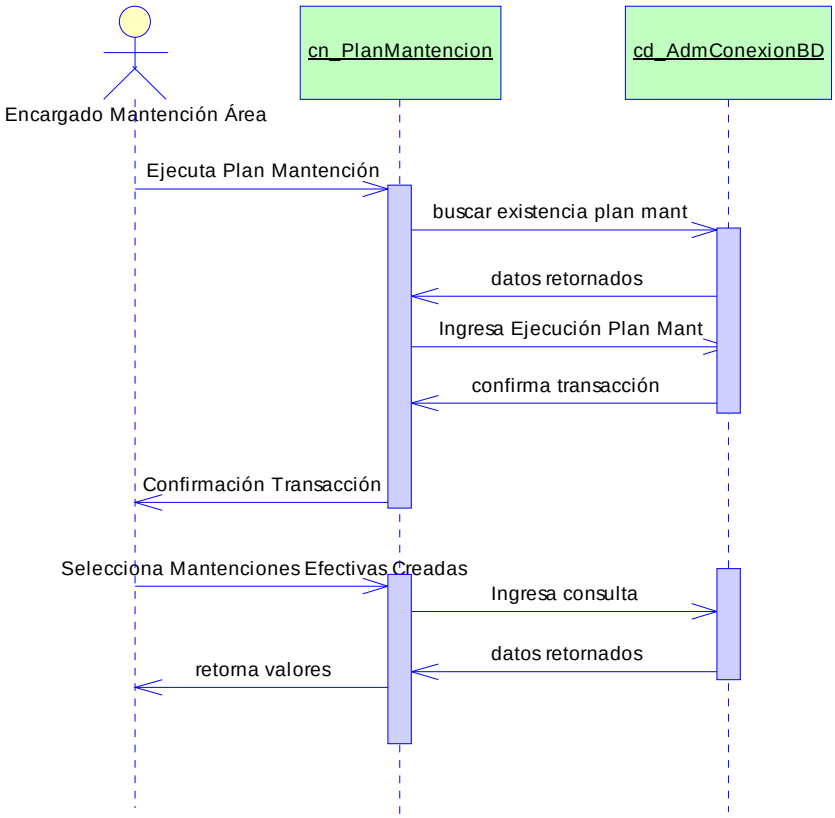


Diagrama Nº [20]: Diagrama Secuencia Ejecutar Plan de Mantención de un Activo

Una vez analizados los diagramas de secuencia, se procederá a implementar el conjunto de pasos que se deben cumplir para satisfacer los casos de usos anteriormente mencionados.

8.1 Principales Pantallas

8.1.1 Pantalla Inicio Sistema

La página de inicio del sistema comprenderá el ingreso de login (usuario) y Password (contraseña). Una vez que el usuario ingrese correctamente los datos solicitados. Una "cokie[12]", se encargará de almacenar la información del usuario mientras "dure", su sesión.

Figura Nº [26]: Pantalla Autenticación PROMO

8.1.2 Pantalla Menú Principal

Contiene una lista con el detalle de las últimas mantenciones que el sistema ha programado para los usuarios. Junto con esto se visualizará la información de un RSS[13] que podrá ser configurado por el administrador del sistema.

[12] Fragmento de información que se almacena en el disco duro del visitante de una página web
[13] Formato permite distribuir contenido sin necesidad de un navegador

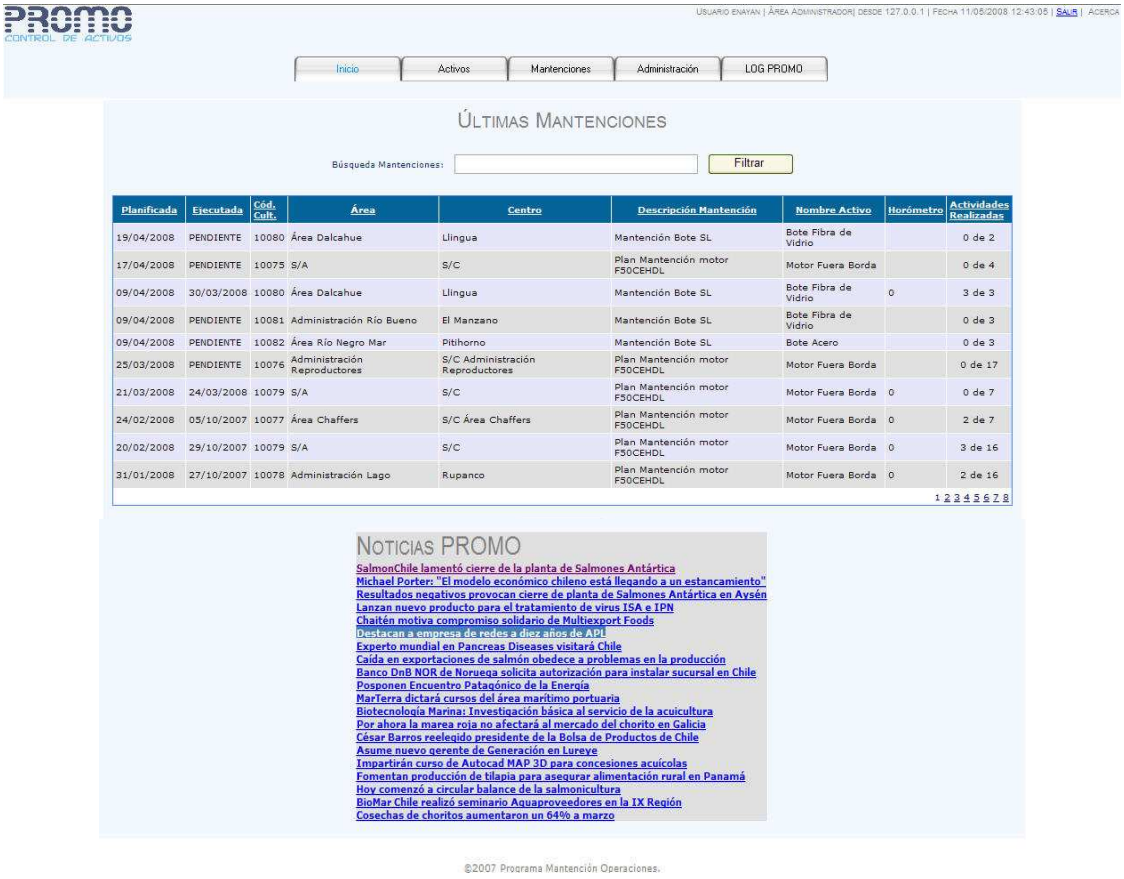


Figura Nº [27]: Pantalla menú principal.

8.1.3 Pantalla Menú Activos

Pantalla que despliega la información de los tipos de activos que contiene la base de datos. Seleccionando [2], se puede ver la pantalla “Ingreso Mantención teórica”, del tipo activo “Botes”.

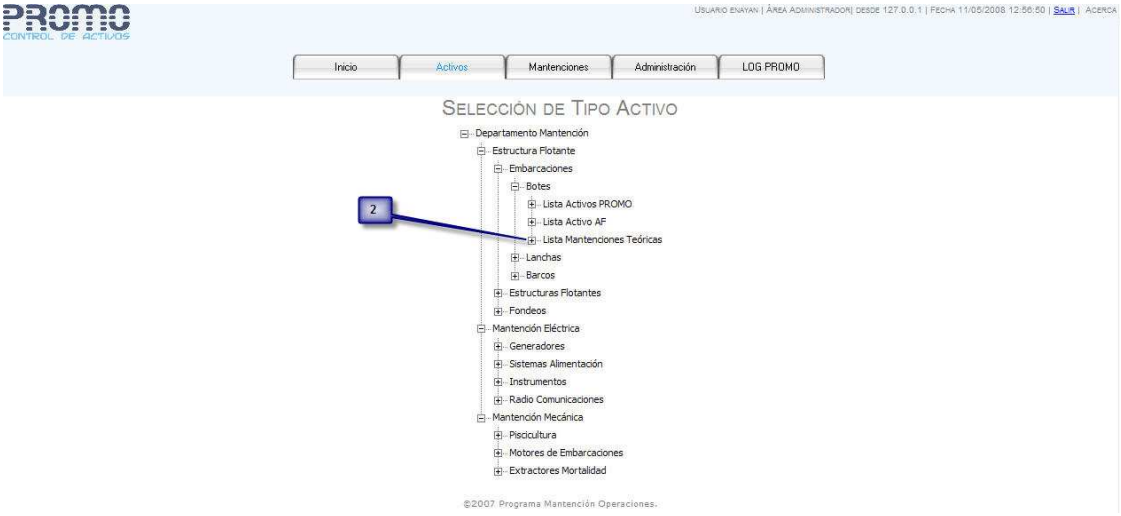


Figura Nº [28]: Pantalla Menú Activos.

8.1.4 Pantalla Ingreso Mantenciones Teóricas

Pantalla donde se ingresan las mantenciones teóricas que permiten la creación de los planes de mantención según modelos.

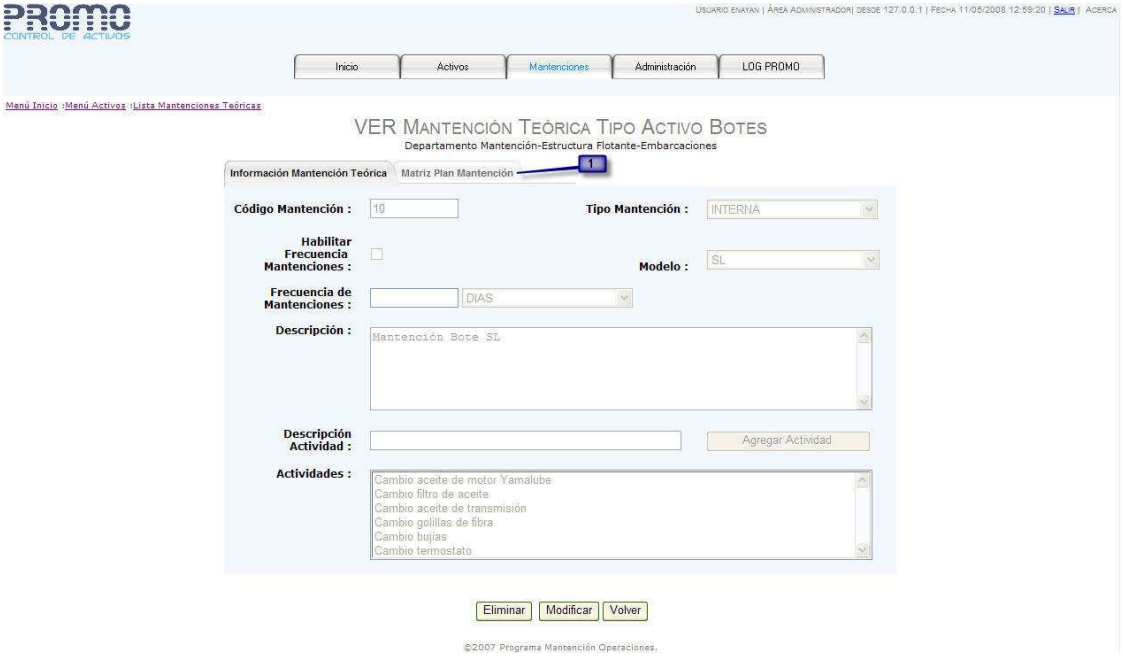


Figura Nº [29]: Pantalla Ingreso Mantenciones Teóricas.

8.1.5 Pantalla Plan Mantención Teórica.

Presionando la pestaña [1] “Matriz Plan Mantención”, de la Figura Nº [29] se gatilla una serie de reglas de negocio que realiza el procedimiento almacenado “sp_InsertFrecuenciasActividades”.

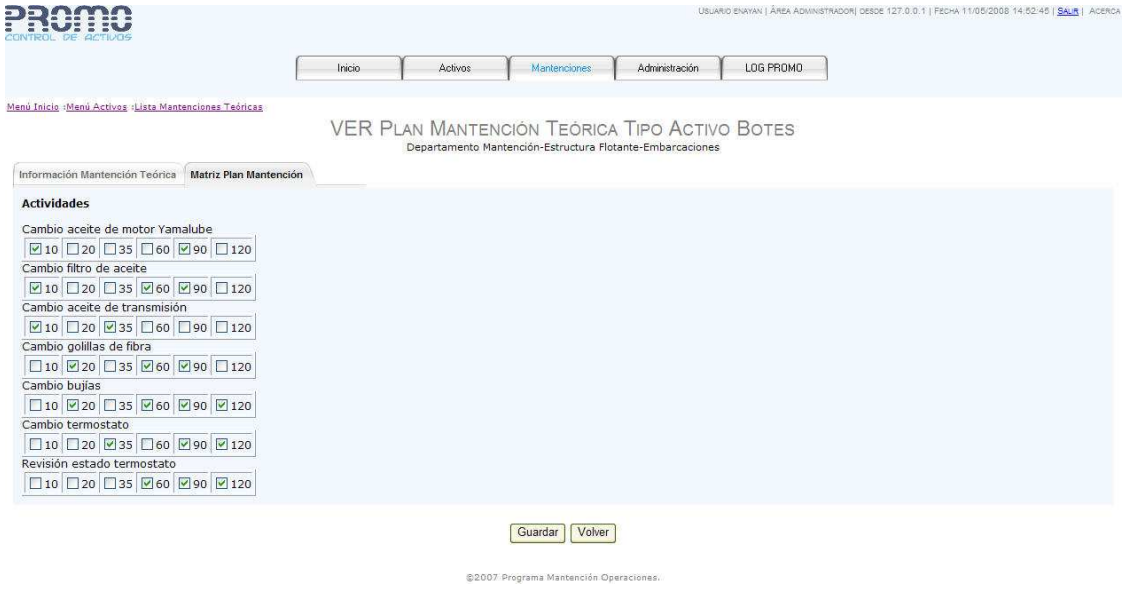


Figura Nº [30]: Pantalla Plan Mantención Teórica.

```
ALTER     PROCEDURE dbo.sp_InsertFrecuenciasActividades
@ildMantencionTeorica     d_identificador,      -- id mantención teórica
@ildCodigoTipoActivo      d_glosa,              -- id tipo activo
@ildModeloTipoActivo      d_identificador       -- id modelo tipo activo
AS
DECLARE @ErrorSave                    d_identificador
--Variables para utilizar el Cursor
DECLARE @sActividadTeorica            d_identificador
DECLARE @iCountActividadTeorica       d_identificador
DECLARE @sErrorSql                    d_glosa_texto
DECLARE @ildFrecuenciaDias            d_identificador
BEGIN
--Declaración del cursor
DECLARE MyCursorFrecuencia CURSOR FOR

SELECT id_actividad_teorica,@ildCodigoTipoActivo,frecuencia_dias
FROM ACTIVIDAD_TEORICA,FRECUENCIA_TIPO_ACTIVO
WHERE id_mantencion_teorica_activo = @ildMantencionTeorica AND
id_modelo_tipo_activo = @ildModeloTipoActivo
ORDEN BY id_actividad_teorica,frecuencia_dias
```

```
--Abrimos el Cursor
OPEN MyCursorFrecuencia FETCH NEXT FROM MyCursorFrecuencia INTO
@sActividadTeorica, @ildCodigoTipoActivo, @ildFrecuenciaDias

WHILE @@FETCH_STATUS = 0
BEGIN
        IF(@ildFrecuenciaDias IS NULL)
        SET @sErrorSql =""
-- en caso de que se cumpla la condición se crea las mantenciones efectivas.
         Exec sp_InsertActividadFrecuencia      @sActividadTeorica,
                                                @ildCodigoTipoActivo,
                                                @ildFrecuenciaDias,
                                                @ildModeloTipoActivo

FETCH NEXT FROM MyCursorFrecuencia INTO @sActividadTeorica,
@ildCodigoTipoActivo, @ildFrecuenciaDias
END

IF (@ildFrecuenciaDias IS NULL)
BEGIN
         SET @sErrorSql = 'NO se puede crear el plan de mantenciones,
porque no existe frecuencia de días para el modelo seleccionado'
         SELECT @sErrorSql AS Error
END
ELSE
BEGIN
        SELECT @sErrorSql AS Error
END
CLOSE MyCursorFrecuencia
DEALLOCATE MyCursorFrecuencia
END

IF (@@ERROR <> 0)
BEGIN
    SET @ErrorSave = @@ERROR
RETURN @ErrorSave
```

Tabla Nº [8]: Procedimiento almacenado Inserta Plan Mantención

Con las pantallas anteriormente expuestas se completan los pasos para ejecutar el caso de uso “Creación Plan de Mantención para Modelo”.

8.1.5 Pantalla Información Básica Activo.

En esta pantalla presionando el botón “Ejecutar Plan Mantención”, se asocia el activo con el plan de mantención que se creó para el modelo seleccionado.

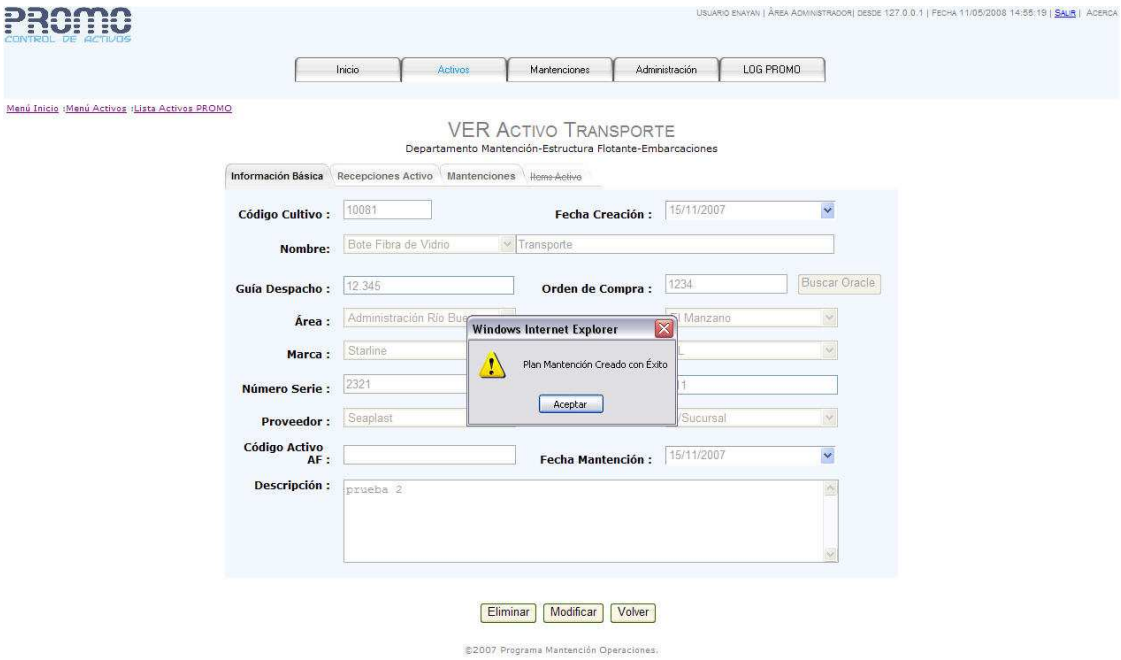


Figura Nº [31]: Pantalla Información Básica Activo.

Al ejecutarse el plan de mantención, también se inserta un detalle de la mantención en el calendario Google, que contiene el nombre del activo (Titulo), fecha, nombre del centro o área donde se realizará la mantención y correo.

Para lograr la consecución de este proceso se utiliza el WebServices llamado "cn_CalendarGoogle", que se describe a continuación:

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Diagnostics;
using System.Web;
using System.Web.Services;
//DLL necesarias para realizar las Inserciones
using Google.GData.Calendar;
using Google.GData.Client;
using Google.GData.Extensions;

namespace capa_negocios
{
 public class cn_CalendarGoogle : System.Web.Services.WebService
 {
  public cn_CalendarGoogle()
  {
   InitializeComponent();
  }

#region Código generado por el Diseñador de componentes
#endregion

//Método que realiza las inserciones en el calendario google.
[WebMethod]
public string InsertEventoCalendarGoogle(string sTitulo
                                         string sContenido
                                         string sDetalle
                                         DateTime dFecha,
```

```
                                                    string sCorreo,
                                                    ref string sMsgErr)
{
 try
  {
    //se instancia la clase CalendarService.
    CalendarService service = new CalendarService("exampleCo-exampleApp-
1");
    //se instancia a la clase EvenEntry que proporciona los métodos para crear
//   las características de la mantención
    EventEntry eNewEntryCalendar        = new EventEntry();
    eNewEntryCalendar.Title.Text         = sTitulo
    eNewEntryCalendar.Content.Content= sContenido

    //se instancia la clase Where que proporciona los métodos en donde se va a
insertar la mantención.
    Where wEvent             = new Where();
    wEvent.ValueString       = sDetalle;
    eNewEntryCalendar.Locations.Add(wEvent);

   //se instancia la clase When que proporciona los métodos cuándo se va a
insertar la mantención
    When eventTime = new When();
    eventTime.StartTime = Convert.ToDateTime(dFecha);
    eventTime.EndTime = eventTime.StartTime.AddMinutes(60);
    eNewEntryCalendar.Times.Add(eventTime);

    //se instancia la clase Uri que permite trabajar con Url's.
    Uri uPostUri        = new Uri(http://www.google.com/calendar/feeds/ +
sCorreo + "/private/full");

    //finalmente se inserta el objeto y se inserta la mantención con sus
características definidas anteriormente en el calendario Google
    AtomEntry aInserteNewEntryCalendarservice.Insert
     (uPostUri,eNewEntryCalendar);
}
catch (Exception Ex)
{
  sMsgErr = Ex.ToString();
}
  }
 }
```

```
}
```

Tabla Nº [9]: WebServices cn_CalendarGoogle.

8.1.6 Pantalla Lista Mantenciones Efectivas.

Al ejecutar el plan de mantención, se crean automáticamente las Mantenciones Efectivas mediante un conjunto de procedimientos almacenados que se mostrarán a continuación y que finalmente muestran la información en la pantalla lista mantenciones efectivas.

Los distintos procedimientos almacenados que son necesarios para realizar esta tarea son los siguientes:

- sp_InsertActivo
- sp_InsertMantencionesEfectivas (descripción en Anexo [1])
- sp_InsertHitoMantencionEfectiva, además utiliza el procedimiento almacenado (sp_ProximoCorrelativoSecuencia).
- sp_InsertActividadesEfectivas, además utiliza el procedimiento almacenado (sp_InsertActividadEfectiva).

Además se utilizaron las siguientes Funciones:

- sf_GetProximoFrecuenciaRealiza.

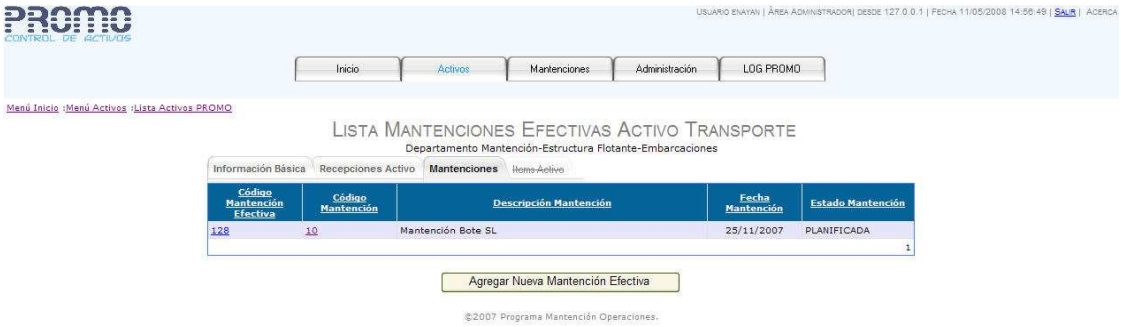


Figura Nº [32]: Pantalla Lista Mantenciones Efectivas.

8.1.7 Pantalla Ingreso detalle Mantenciones Efectivas.

Pantalla donde se ingresa el detalle de la mantención realizada al activo. Claro esta que el operario que realiza la mantención se guía por las fechas y actividades que el sistema entrega para el activo en cuestión. Una vez ejecutada la mantención el sistema creará la siguiente mantención utilizando el mismo procedimiento descrito en el punto 8.1.6.

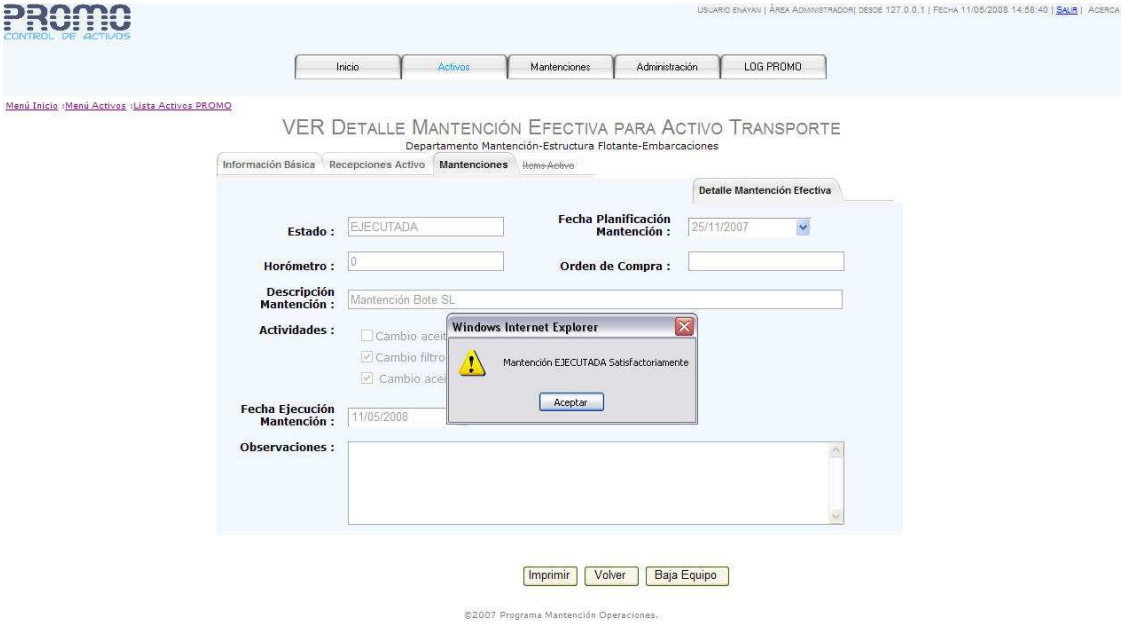


Figura Nº [33]: Pantalla Ingreso detalle Mantenciones Efectivas.

Con las pantallas anteriormente expuestas desde el punto 8.1.5 hasta el punto 8.1.7, se completan los pasos para ejecutar el caso de uso “Ejecutar Plan de Mantención de un Activo”.

8.2 Implementación de la Réplica de base de datos

Para este proyecto de tesis se utilizó la réplica de mezcla, que utiliza al servidor central como publicador y distribuidor de la réplica. Mientras que los suscriptores son los servidores remotos. Para representar este escenario se muestra el siguiente modelo.

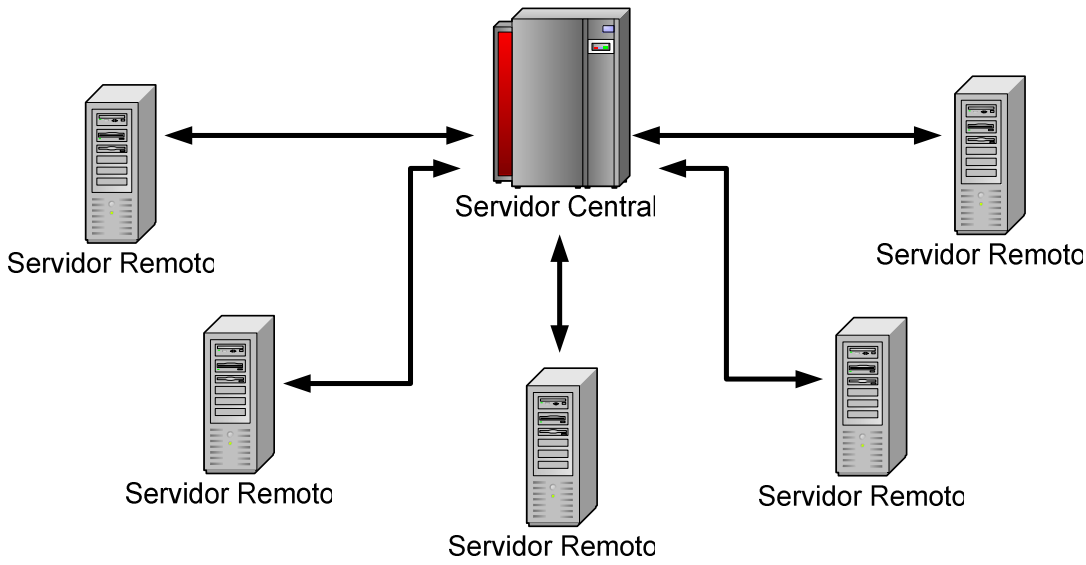


Diagrama Nº [21]: Escenario de la réplica.

Para la implementación de la réplica de mezcla se tomaron en consideración los siguientes requisitos:

- Los datos se escriben y actualizan en un servidor central y en los servidores remotos.

- Los usuarios remotos pueden realizar actualizaciones de forma independiente, sin que sea necesaria una conexión con el servidor central.
- La réplica de mezcla proporciona un mecanismo predeterminado o personalizado para la detección y resolución de conflictos que puedan ocurrir mientras se realiza la replicación.
- Permite la actualización de información solo en el servidor central.
- Los usuarios pueden sincronizar los datos a petición o en momentos programados.
- Algunas tablas requieren filtrado para que cada usuario reciba datos diferentes de una o varias tablas.
- Permite controlar cuánto tiempo puede permanecer un sitio remoto sin sincronizar.
- Permite la ejecución de lógica de negocios personalizada al sincronizar los datos.

Finalmente el requisito más importante que proporciona la replicación de mezcla radica en que resuelve conflictos y provee de filtros que proporcionan a cada servidor remoto un conjunto de datos exclusivo.

En el siguiente diagrama se muestran los componentes que se utilizaron para implementar la réplica de mezcla.

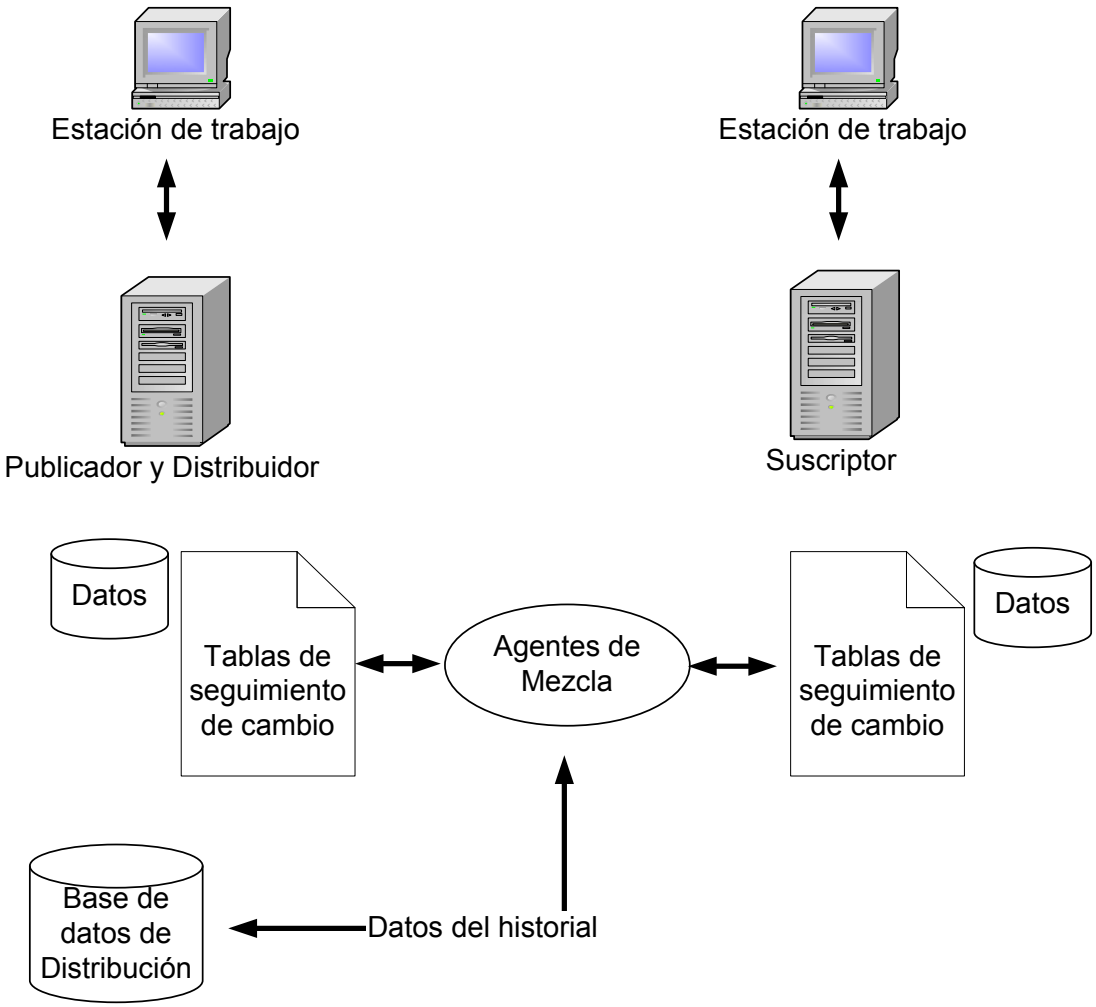


Diagrama Nº [22]: Componentes replicación de mezcla.

Para realizar la réplica de mezcla se comenzó con la generación y aplicación de la instantánea. Luego la réplica de instantáneas distribuye los datos exactamente como aparecen en un momento específico en el tiempo y no supervisa las actualizaciones de los datos. Cuando se produce la sincronización, se genera la instantánea completa y se envía a los suscriptores.

Con la réplica se utilizaron programas independientes, llamados agentes, para realizar las tareas asociadas con el seguimiento de los cambios y la distribución de los datos. De forma predeterminada, los agentes de réplica se ejecutan como trabajos programados en el Agente SQL Server. El agente que provee SQL SERVER 2005 para la réplica de mezcla se denomina Agente de Mezcla y su descripción se detalla según un extracto obtenido de la ayuda de la herramienta "Microsoft SQL Server Management Studio":

"El Agente de mezcla se utiliza con la réplica de mezcla. Aplica la instantánea inicial al suscriptor, transfiere y reconcilia los cambios incrementales de datos que se producen. Cada suscripción de mezcla tiene su propio Agente de mezcla, que se conecta con el publicador y con el suscriptor, y los actualiza. El Agente de mezcla se ejecuta en el distribuidor, para las suscripciones de inserción, o en el suscriptor, para las suscripciones de extracción. De forma predeterminada, el Agente de mezcla carga los cambios del suscriptor al publicador y, a continuación, descarga los cambios del publicador al suscriptor".

Para un mejor entendimiento se mostrará una configuración de réplica entre un motor de base de datos SQL SERVER 2005 y un motor de base de datos SQL SERVER EXPRESS en el anexo [2] y anexo [3] respectivamente.

9 Pruebas

En este capítulo se describirán las pruebas que se realizaron a los distintos artefactos de software desarrollados, ya sea en conjunto o modularizado, esto con el fin de encontrar o detectar diferencias entre las condiciones requeridas y las reales (fallas o defectos), y finalmente evaluar las características del artefacto.

Antes de continuar, se debe aclarar que el término “usuario final”, corresponde a la contraparte técnica del proyecto y que está compuesto por un equipo de trabajo, que tiene como misión generar los requerimientos y obviamente aprobar las funcionalidades acordadas para el sistema.

9.1 Conceptos generales de las pruebas realizadas

9.1.1 Pruebas de Unidad.

Las pruebas de unidad se realizaron sobre cada módulo de forma independiente. Con esto se consigue comprobar que los módulos, vistos como unidades funcionales de programas autónomos están correctamente codificados.

Dichos módulos fueron “testeados”, bajo la técnica de prueba llamada “caja blanca”.

9.1.2 Pruebas de Integración.

Estas pruebas se realizaron de forma incremental, es decir, a medida que se desarrollaban los módulos del sistema, se comprobaba el funcionamiento del sistema en su totalidad, cabe señalar que sólo al final del último ciclo del proyecto se pudo comprobar que el sistema PROMO funcionaba de forma correcta y con la prestación necesaria para satisfacer los requerimientos iniciales y los generados durante la construcción del proyecto.

9.1.3 Pruebas de Sistema y Evaluación

Las pruebas de caja de negra, se utilizaron para comprobar el comportamiento del sistema. Este tipo de pruebas se utilizaron para el ingreso de valores máximos y mínimos para los dominios permitidos como entradas. Bajo este análisis se pueden aplicar varios casos de pruebas tales como:

- Ingreso de valores que sobrepasan el valor permitido
- Ingreso de valores que están por “debajo”, del mínimo valor permitido
- Ingreso de valores alfanuméricos, donde solo se permiten ingresos numéricos.

Para las pruebas de validación se utilizaron “Pruebas de interfaces Gráficas de Usuario (Gui’s)”, que constan de las siguientes pruebas:

- Ventanas correctamente Tabuladas, es decir, al presionar la tecla TAB, debería ejecutarse de izquierda a derecha y hacia abajo en los controles que posea la ventana.
- Comprobar ortografía y gramática de la ventana.

Por último, las pruebas estéticas fueron evaluadas por los usuarios finales, mediante prototipos que fueron presentados en cada una de las reuniones que se realizaron, en donde ellos entregaban su percepción del sistema. Los comentarios y recomendaciones fueron recabados e implementados en las interfaces que componen el sistema.

También cabe mencionar que las validaciones que realiza el sistema ante ingresos de datos de forma incorrecta y posteriores mensajes que visualiza el usuario a raíz de los mismos errores, fueron aprobados ortográficamente y gramaticalmente por los usuarios. Esto busca familiarizar 100% al usuario con el sistema.

Las condiciones de navegabilidad fueron objetos de pruebas y básicamente consistieron en:

- Correcto direccionamiento de las opciones elegidas por el usuario.

- Verificar que la navegación de los distintos tipos de usuarios del sistema concuerde con el perfil asignado.

Para comprobar las condiciones de integridad de datos, se verifico que los datos ingresados por el usuario (entiéndase a través de la interfaz del sistema), no fueron “truncados”, por la base de datos.

9.1.4 Prueba de Seguridad

A continuación se mencionan algunas pruebas que se realizaron, para comprobar la seguridad y estabilidad que ofrece el sistema ante estos casos:

- Comprobación de los distintos perfiles de usuarios del sistema.
- Comprobación del ingreso de usuarios correctamente autorizados.
- Verificar la existencia de caducidad de una sesión.
- Verificar el comportamiento de la sesión cuando:
    - Se cierra y vuelve atrás el navegador.
    - Comprobar el “linkeo”, directo a una página.
- Cifrado de la clave de usuario en la base de dato.

9.2 Plan de pruebas.

Con los antecedentes mencionados anteriormente se crea un plan de pruebas que permite definir las actividades, planificar y estimar los esfuerzos, condiciones y recursos involucrados para la ejecución de las pruebas del sistema PROMO. A continuación se describe el plan de pruebas de forma general, de acuerdo a los siguientes objetivos:

- Identificar la información generada por el proyecto y los componentes de software que deberán ser probados.
- Establecer las actividades de prueba a ser realizadas.
- Recomendar y describir la estrategia de pruebas a utilizar.

9.2.1 Pruebas Funcionales

En este plan de Test se aplicarán técnicas de “Caja Negra”. Las que tendrán como objetivo principal verificar el comportamiento del sistema, comparando el resultado esperado detallado en el caso de uso contra el resultado obtenido en la ejecución de las pruebas.

Los datos de entrada serán los utilizados por las transacciones involucradas. Cada argumento de entrada puede seleccionar uno de los siguientes datos de pruebas, dependiendo este del resultado que se desea obtener (esperado),

verificando así el comportamiento del componente a testear, usando los siguientes valores de entrada:

- Valores normales para cada transacción.
- Valores límites para cada transacción.
- Valores de equivalencia.
- Valores ilegales.
- Los requerimientos para validar estas pruebas se basaron en asociar los requerimientos funcionales a los casos de usos.

9.2.2 Estrategias de las Pruebas

Las pruebas serán realizadas por funcionalidades o módulos en la medida que estos objetos sean entregados al proceso de prueba, dentro de los meses establecidos para dichas pruebas, plazo acordado previamente, considerando siempre el caso de uso que se encuentra desarrollando.

Las pruebas realizadas están enfocadas a revisar principalmente los siguientes puntos:

- Los link de las páginas deberán estar correctos.
- Los datos deberán ser almacenados y recuperados en forma correcta.

- La interacción y retroalimentación con otros sistemas de la empresa son correctos.
- El sistema deberá ser robusto para controlar valores limites.
- El sistema deberá controlar mensaje ante validaciones del sistema.

9.2.3 Tipos de Pruebas

La integridad de la bases de datos se verificará de acuerdo a la siguiente tabla.

Objetivo del Test:	Asegurar que los métodos de Acceso a la Base de Datos y los procesos asociados, funcionan apropiadamente y sin riesgo de datos corruptos.
Técnica a Utilizar:	Realizar la llamada a una Base de Datos y ejecutar algún proceso con datos válidos e inválidos. Inspeccionar la Base de Datos para verificar que los procesos se han realizado correctamente.
Criterio de Verificación:	Todos los métodos de acceso a la Base de Datos y sus procesos deben estar sin datos corruptos.
Consideraciones Especiales:	La prueba puede requerir que el Administrador de Bases de Datos diseñe un ambiente para acceder a los datos directamente. Deben invocarse los procesos manualmente. Se debe utilizar una reducida cantidad de registros para facilitar la inspección de los datos e identificar eventos erróneos.

Tabla Nº [10]: Descripción de las Pruebas para Base de datos.

Las pruebas funcionales se realizaron de acuerdo a la siguiente tabla.

Objetivo del Test:	Asegurar la funcionalidad del conjunto de casos, incluyendo la navegación en la aplicación, el ingreso de datos, el proceso y la recuperación (resultados). Que la navegación a través de los casos de prueba refleje apropiadamente las reglas del negocio y los requerimientos, incluyendo ventana a ventana, campo a campo y usando los métodos de acceso correctamente (tecla tab, movimiento del mouse, etc.) Que los objetos de las ventanas y sus características, tales como menús, tamaño, posición, estados y el foco, estén de acuerdo a los estándares.
Técnica a Utilizar:	Ejecutar cada Caso de Pruebas, su flujo y funcionalidad usando tanto datos válidos como inválidos para verificar lo siguiente: • Resultados esperados ocurren cuando los datos válidos son utilizados. • El mensaje de error es el apropiado cuando se utilizan datos inválidos • Cada Regla de Negocio se utiliza apropiadamente. • Crear y modificar los procedimientos de Pruebas para cada ventana, para verificar los estados de los objetos y de la aplicación.
Criterio de Verificación:	• Todas las pruebas planificadas se ejecutaron correctamente • Todos los defectos identificados han sido asignados. • Cada ventana debe ser verificada para mantener la consistencia con la versión maestra y verificar que esté dentro de los estándares aceptables.

Tabla Nº [11]: Descripción de las Pruebas Funcionales

Las pruebas de seguridad se comprobaron de acuerdo a la siguiente tabla.

Objetivo del Test:	Asegurar el correcto funcionamiento del sistema y el conjunto de componentes de seguridad. Que la navegación a través de los casos de prueba refleje apropiadamente la aplicación de las políticas de seguridad y los requerimientos.
Técnica a Utilizar:	Ejecutar cada Caso de Pruebas, su flujo y funcionalidad para verificar: • Que los resultados esperados ocurren cuando se ha detectado alguna vulnerabilidad. • Que el sistema como sus componentes, sea capaz de detener cualquier intento de intrusión o acceso.
Criterio de Verificación:	• Todas las pruebas planificadas se ejecutaron correctamente • Todos los defectos identificados han sido asignados para modificación.

Tabla Nº [12]: Descripción de las Pruebas de Seguridad

Dado que el sistema se encuentra entregado y aprobado, se entiende que todas las pruebas mencionadas en este capitulo, tuvieron resultados satisfactorios y que cubren todos los requerimientos establecidos en la etapa de análisis del proyecto.

Con estos antecedentes se puede asegurar que el sistema ofrece una prestación bastante aceptable y que cumple a cabalidad con lo requerido para el proyecto.

En el anexo [4], se especifica una plantilla de casos de prueba utilizados en la etapa de pruebas del sistema.

En el anexo [5], se detalla como debe funcionar el sistema según un caso de prueba especifico.

10 Conclusiones

De acuerdo a la experiencia obtenida durante el desarrollo de este proyecto de tesis, y lo expuesto en el presente documento, se puede concluir que los objetivos planteados se cumplieron satisfactoriamente. Sin embargo se debe reconocer y ser bien crítico, en que el sistema aún no ha logrado la madurez suficiente, ya que no se ha implementado en todas las áreas que la compañía posee, por lo tanto no se puede establecer con fundamentos sólidos que el sistema brinde prestaciones aceptables en ambientes muy críticos. Evidentemente que con el tiempo y con un monitoreo continuo de la aplicación, se logrará un sistema totalmente robusto y que ofrezca una excelente prestación.

Si bien, hoy en día existen muchas aplicaciones que centralizan información y ofrecen variadas características de acceso a dicha información, se podría pensar que este sistema es más de lo mismo, pero claramente no lo es, por que las prestaciones y características que ofrece este sistema lo hacen único, ya que esta totalmente basado en inquietudes e ideas de los propios usuarios representados por un grupo multidisciplinario de trabajo. Basándose en esta última frase se puede señalar que el mayor inconveniente de este proyecto de tesis fue la fase de análisis, pues consensuar ideas para un ingeniero con nula

experiencia es bastante complejo, pero que fue sobre llevado con éxito con la ayuda de la metodología de desarrollo XP o programación extrema que brinda mecanismos que permitieron cumplir de manera eficaz esta fase, esto por que propone un énfasis totalmente orientado a lograr productos o artefactos tangibles por los usuarios y no enfocado ha tanta documentación que por lo general el usuario no entiende. Para esto XP plantea el desarrollo de casos de uso y prototipos entre otros mecanismos que facilitan el análisis. Unas de estas herramientas son los prototipos semi – funcionales que establecen métricas para demostrar que lo entendido por el desarrollador representa la idea manifestada por el usuario, y así acotar los márgenes de error que pudieran existir.

Con esto se quiere dar a entender que la solución presentada se basó en métricas que facilitaron el desarrollo total del sistema. Pero también se debe señalar que la utilización de metodologías en el desarrollo de software está totalmente ligada con la experiencia del equipo desarrollador, por ende, establecer recomendaciones para desarrollos futuros con esta metodología, estaría fuera de contexto, ya que se desconoce las bondades que ofrecen otras metodologías de desarrollo de software. Lo cual no excluye mencionar y recalcar en una apreciación muy personal, en que esta metodología es una excelente opción para afrontar desarrollos de software.

Como se menciona anteriormente, la inexperiencia afecta de alguna manera el desarrollo de cualquier proyecto, independiente del contexto. Es por esto que en este seminario de tesis, se documenta en extenso las fases que permiten tener una claridad total del sistema, como son las etapas de análisis y diseño.

El desarrollo utilizado en la implementación de este sistema con la herramienta Visual Studio .NET 2003, que esta totalmente orientado a objetos y que permite un desarrollo en múltiples capas, brinda la escalabilidad que hoy en día necesitan las aplicaciones para poder mantenerse en el tiempo y adaptarse de mejor forma al constante cambio tecnológico que se vive en la actualidad. Pero hay que ser cautos en que esto va de la mano con un diseño adaptable y flexible de la aplicación y que es lo que trata de cumplir este sistema, apostando a un producto y no a un desarrollo a la medida.

Por último, se debe señalar que el énfasis que entrega la Universidad Austral de Chile, en las fases que permiten tener una mayor claridad y conceptualizar una idea de un sistema relativamente extenso, queda en deuda de acuerdo a lo mencionado anteriormente, pero se entiende que los procesos de enseñanza se ajustan y modifican a través del tiempo y conforme a las necesidades del mercado.

11 Bibliografía

[Connolly2002] Connolly and Begg. Database Systems. A practical approach to design, implementation and management. Addison Wesley, Segunda Edición. 1999.

[Liz02] Lizarraga Celaya, Carlos (2002). *Servicios Web*. Universidad de Sonora.

[Har05] Harold, Elliotte Rusty, Scout Jeans (2005) *XML* Imprescindible.

[Dan01] Daniel A. Menascé, Virgilio A. F. Almeida (2001), Capacity Planning for Web Services.

[Kent Beck 2000] Extreme Programming Explained: Embrace Change

[URL 1] http://www.webservices.org/

[URL 2] http://www.programacionextrema.org/

12 Anexos

Anexo Nº [1]: Procedimiento Almacenado sp_InsertMantencionesEfectivas

```
/*********************************************************************
Nombre:             sp_InsertMantencionesEfectivas
Desarrollador:      Erick Nayán
Parametros:         número activo,número area, fecha mantención, código tipo
                    activo, fecha ejecución de la mantención, número del modelo,
                    número mantención teórica
Descripcion:        Inserta las mantenciones efectivas de un activo en particular
Última Modificación:  06/06/2007 18:40
*********************************************************************/
ALTER PROCEDURE [dbo].[sp_InsertMantencionesEfectivas]
@ildActivo                      d_correlativo,
@ildArea                        d_identificador,
@dFechaMantencion               d_fecha,
@ildCodigoTipoActivo            d_glosa,
@dFechaEjecucionMantencion      d_fecha,
@ildModeloTipoActivo            d_identificador,
@ildMantencionTeorica           d_identificador = null
AS
```

```
DECLARE @ErrorSave                      d_identificador
DECLARE @Correlativo                    d_correlativo
DECLARE @NewCorrelativo                 d_correlativo
--Variables para utilizar el Cursor
DECLARE @idlMantencionTeorica               d_identificador
DECLARE @sDescripcionMantencionTeorica      d_glosa_larga
DECLARE @sValorVariable                     d_identificador
DECLARE @sExisteValorVariable               d_identificador
DECLARE @sFrecuenciaRealizaActividad        d_identificador
DECLARE @sCondicionRealizaActividad         d_identificador
DECLARE @sExisteMantencionEfectiva          d_identificador
DECLARE @sErrorSql                          d_glosa_texto
DECLARE @sFrecuenciaRealiza                 d_identificador
BEGIN
    IF (@ildMantencionTeorica IS NOT NULL)
    BEGIN
        DECLARE MyCursor CURSOR FOR –declaración del cursor según
descriminación señala en el IF anterior
        SELECT id_mantencion_teorica_activo,
        descripcion_mantencion_teorica,
```

```
            valor_variable_mantencion_teorica FROM
            HITO_MANTENCION_TEORICA
            WHERE (id_mantencion_teorica_activo=@ildMantencionTeorica)
            OR (id_modelo_tipo_activo = @ildModeloTipoActivo) – condición
            de ejecución del cursor
        END
        ELSE
        BEGIN
            IF(@ildCodigoTipoActivo = '2.3.1') – código identificador para item
de activo y alternativa según descriminación señalada anteriormente.
            BEGIN
                DECLARE MyCursor CURSOR FOR
                SELECT id_mantencion_teorica_activo,
                descripcion_mantencion_teorica,
                valor_variable_mantencion_teorica FROM
                HITO_MANTENCION_TEORICA
                WHERE (codigo_tipo_activo = @ildCodigoTipoActivo) OR
                (id_modelo_tipo_activo = @ildModeloTipoActivo)
            END
            ELSE
            BEGIN
```

```
                DECLARE MyCursor CURSOR FOR
                SELECT id_mantencion_teorica_activo,
                descripcion_mantencion_teorica,
                valor_variable_mantencion_teorica FROM
                HITO_MANTENCION_TEORICA
                WHERE (codigo_tipo_activo = @ildCodigoTipoActivo) AND
                (id_modelo_tipo_activo = @ildModeloTipoActivo)
            END
        END
        OPEN MyCursor -- se abre el cursor
        FETCH NEXT FROM MyCursor INTO @idlMantencionTeorica ,
        @sDescripcionMantencionTeorica, @sValorVariable – se inicia el cursor
        WHILE @@FETCH_STATUS = 0
        BEGIN
            DECLARE @Fecha as d_fecha
            IF (@sValorVariable is null) – bifurcación código según parametro
de ingreso
            BEGIN
                SELECT @sExisteMantencionEfectiva =
                count(id_mantencion_efectiva_activo) from
```

```
        HITO_MANTENCION_EFECTIVA where id_activo =
        @ildActivo
SELECT @sFrecuenciaRealizaActividad =
frecuencia_realiza_actividad FROM
HITO_MANTENCION_EFECTIVA
WHERE id_mantencion_teorica_activo = @idlMantencionTeorica
SET @sCondicionRealizaActividad = 1
IF (@sFrecuenciaRealizaActividad IS NULL) OR
(@sExisteMantencionEfectiva = 0)
BEGIN
        SELECT @sFrecuenciaRealizaActividad =
        FRECUENCIA_DIAS FROM
        sf_GetProximoFrecuenciaRealiza(@ildCodigoTipoActivo,0,
        @ildModeloTipoActivo)
        SET @sValorVariable = @sFrecuenciaRealizaActividad
        SET @Fecha =
        DATEADD(day,@sValorVariable,@dFechaMantencion)

END
ELSE
BEGIN
```

```
SELECT @sFrecuenciaRealiza = FRECUENCIA_DIAS
FROM
sf_GetProximoFrecuenciaRealiza(@ildCodigoTipoActivo,
@sFrecuenciaRealizaActividad,@ildModeloTipoActivo)
IF (@sFrecuenciaRealiza IS NULL)
BEGIN
        SET @sErrorSql = '1'
END
ELSE
BEGIN
        SELECT @sFrecuenciaRealizaActividad =
        FRECUENCIA_DIAS FROM
        sf_GetProximoFrecuenciaRealiza(
        @ildCodigoTipoActivo,
        @sFrecuenciaRealizaActividad,
        @ildModeloTipoActivo)
        SET
        @sValorVariable=@sFrecuenciaRealizaActividad
        SET @Fecha = DATEADD(day,@sValorVariable,
        @dFechaMantencion)
END
```

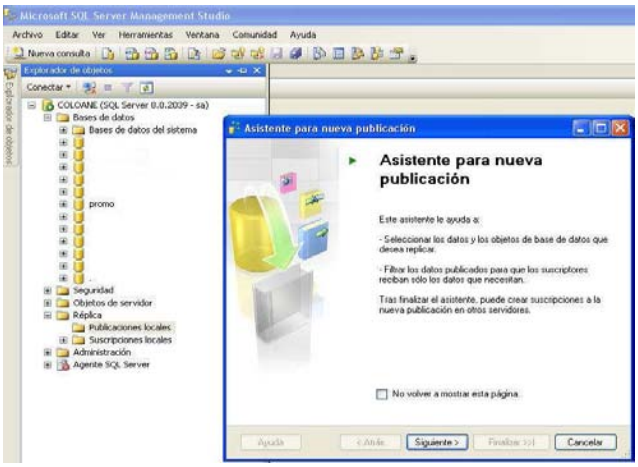
```
        END
END
ELSE
BEGIN
        SET @Fecha =
        DATEADD(day,@sValorVariable,@dFechaMantencion)
        SET @sCondicionRealizaActividad = 0
        SET @sFrecuenciaRealizaActividad = NULL
END
IF (@sErrorSql = '1')
BEGIN
        SET @sErrorSql = 'La vida útil del activo a terminado, por lo tanto
        no existen mantenciones planificadas por realizar. Se recomienda
        dar de baja al activo'
        SELECT @sErrorSql AS id_mantencion_efectiva_activo
END
ELSE
BEGIN
-- inserción de la próxima mantención con estado Planificada
Exec sp_InsertHitoMantencionEfectiva  @ildActivo,
                        @ildArea,
```

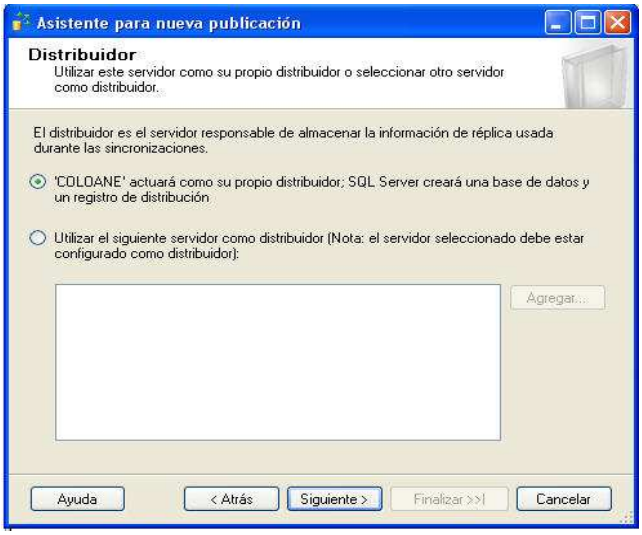
```
                    @idlMantencionTeorica,
                    @Fecha,
                    'PLANIFICADA',
                    NULL,
                    NULL,
                    NULL,
                    NULL,
                    NULL,
                    @sDescripcionMantencionTeorica,
                    NULL,
                    @sFrecuenciaRealizaActividad,
                    @@ildMantencionEfectiva = @Correlativo OUTPUT
SET @NewCorrelativo = @Correlativo
IF (@sCondicionRealizaActividad = 1)
        EXEC sp_InsertActividadesEfectivas
        @NewCorrelativo,@idlMantencionTeorica,@sValorVariable
        ,@ildModeloTipoActivo
ELSE
        EXEC sp_InsertActividadesEfectivas
        @NewCorrelativo,@idlMantencionTeorica
END
```

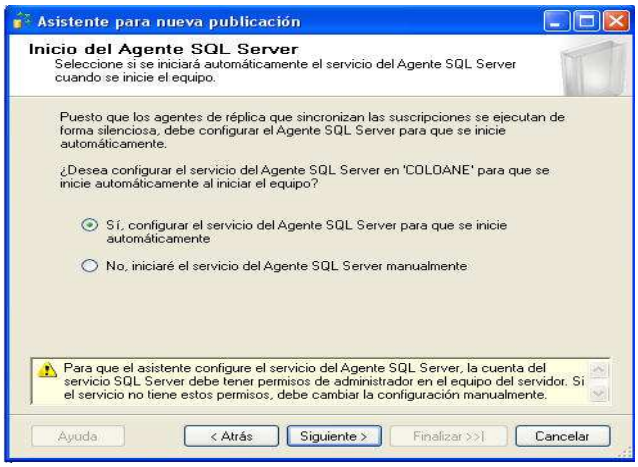
```
        FETCH NEXT FROM MyCursor INTO
        @idlMantencionTeorica,@sDescripcionMantencionTeorica,
        @sValorVariable
        END
        CLOSE MyCursor
        DEALLOCATE MyCursor
        END
IF (@@ERROR <> 0)
  BEGIN
     SET @ErrorSave = @@ERROR
     RETURN @ErrorSave
   END
```

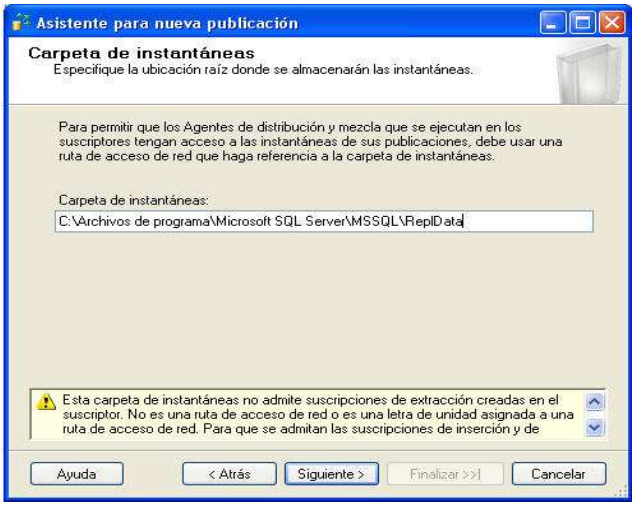
Anexo Nº [2]: Configuración de una Publicación con motor de base de datos SQL SERVER 2005

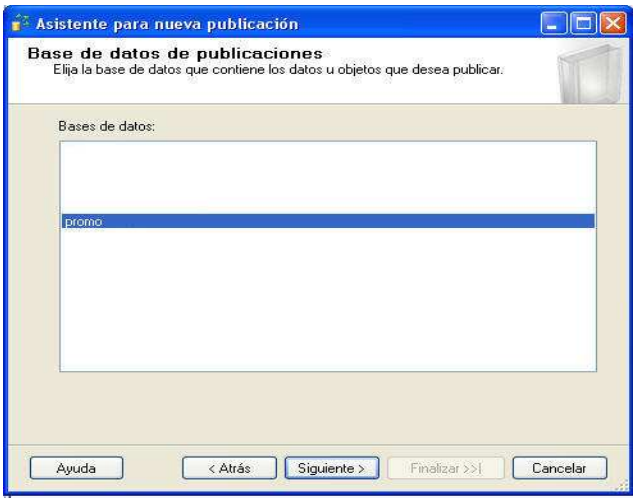
A continuación se describen los pasos necesarios para implementar una Publicación en una réplica.

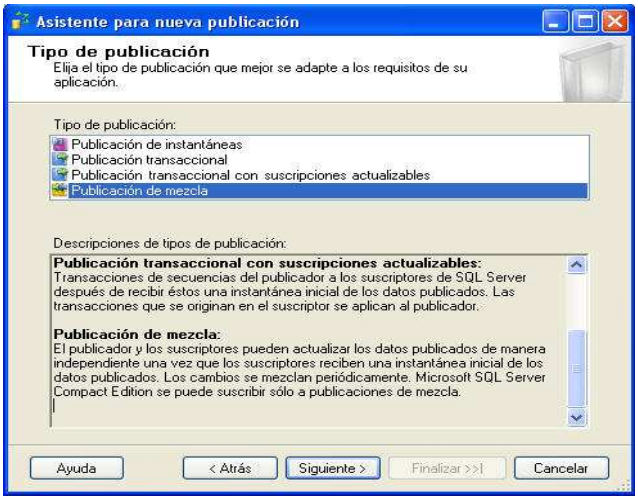


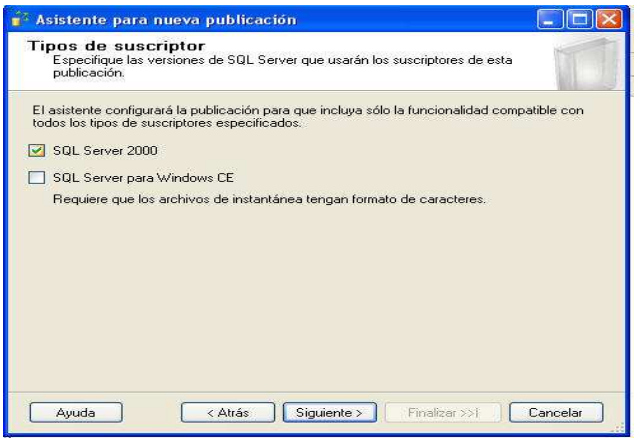


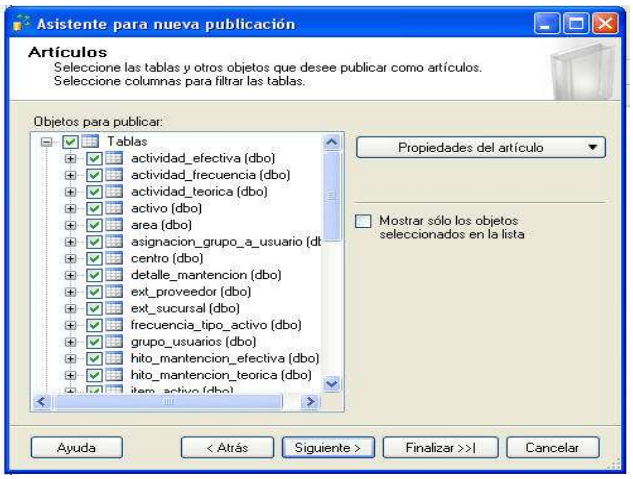

Asistente para nueva publicación
Inicio del Agente SQL Server
Seleccione si se iniciará automáticamente el servicio del Agente SQL Server cuando se inicie el equipo.
Puesto que los agentes de réplica que sincronizan las suscripciones se ejecutan de forma silenciosa, debe configurar el Agente SQL Server para que se inicie automáticamente.
¿Desea configurar el servicio del Agente SQL Server en 'COLOANE' para que se inicie automáticamente al iniciar el equipo?
Sí, configurar el servicio del Agente SQL Server para que se inicie automáticamente
No, iniciaré el servicio del Agente SQL Server manualmente
Para que el asistente configure el servicio del Agente SQL Server, la cuenta del servicio SQL Server debe tener permisos de administrador en el equipo del servidor. Si el servicio no tiene estos permisos, debe cambiar la configuración manualmente.
Ayuda
< Atrás
Siguiente >
Finalizar >>|
Cancelar

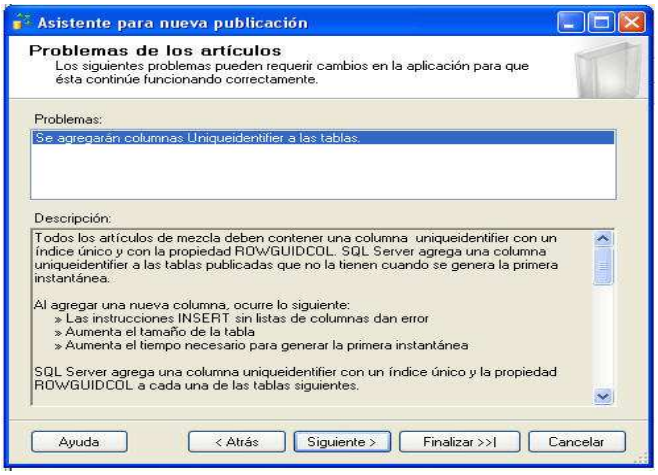

Asistente para nueva publicación
Carpeta de instantáneas
Especifique la ubicación raíz donde se almacenarán las instantáneas.
Para permitir que los Agentes de distribución y mezcla que se ejecutan en los suscriptores tengan acceso a las instantáneas de sus publicaciones, debe usar una ruta de acceso de red que haga referencia a la carpeta de instantáneas.
Carpeta de instantáneas:
C:\Archivos de programa\Microsoft SQL Server\MSSQL\ReplData
Esta carpeta de instantáneas no admite suscripciones de extracción creadas en el suscriptor. No es una ruta de acceso de red o es una letra de unidad asignada a una ruta de acceso de red. Para que se admitan las suscripciones de inserción y de
Ayuda
< Atrás
Siguiente >
Finalizar >>|
Cancelar

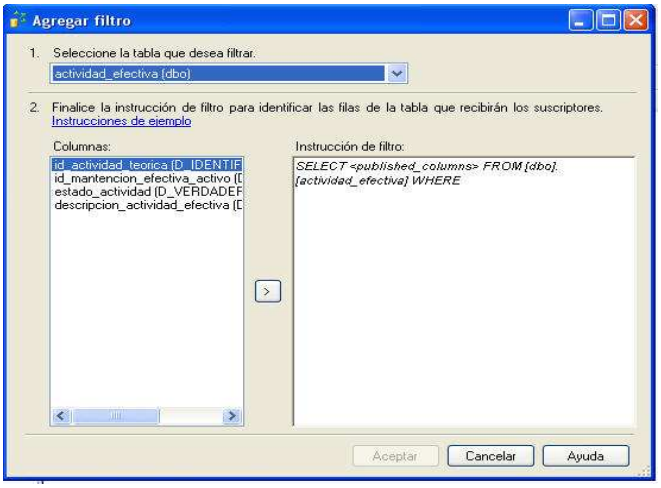

Asistente para nueva publicación
Base de datos de publicaciones
Elija la base de datos que contiene los datos u objetos que desea publicar.
Bases de datos:
promo
Ayuda
< Atrás
Siguiente >
Finalizar >>|
Cancelar

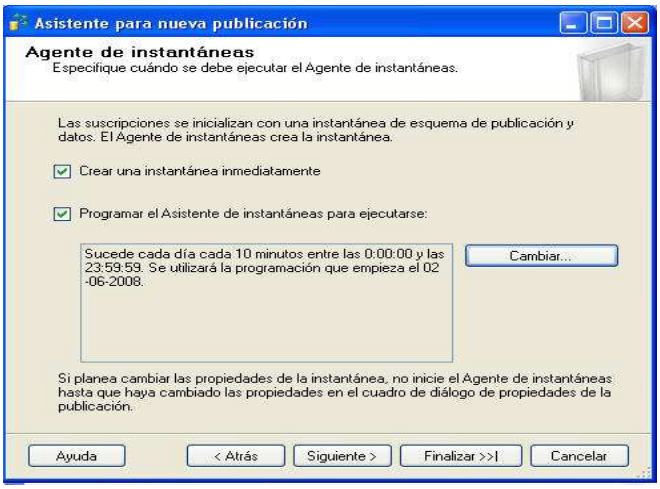

Asistente para nueva publicación
Tipo de publicación
Elija el tipo de publicación que mejor se adapte a los requisitos de su aplicación.
Tipo de publicación:
Publicación de instantáneas
Publicación transaccional
Publicación transaccional con suscripciones actualizables
Publicación de mezcla
Descripciones de tipos de publicación:
Publicación transaccional con suscripciones actualizables:
Transacciones de secuencias del publicador a los suscriptores de SQL Server después de recibir éstos una instantánea inicial de los datos publicados. Las transacciones que se originan en el suscriptor se aplican al publicador.
Publicación de mezcla:
El publicador y los suscriptores pueden actualizar los datos publicados de manera independiente una vez que los suscriptores reciben una instantánea inicial de los datos publicados. Los cambios se mezclan periódicamente. Microsoft SQL Server Compact Edition se puede suscribir sólo a publicaciones de mezcla.
Ayuda
< Atrás
Siguiente >
Finalizar >>|
Cancelar

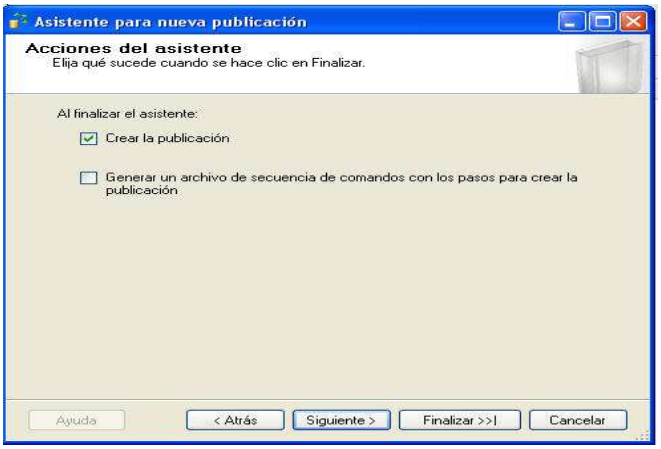

Asistente para nueva publicación
Tipos de suscriptor
Especifique las versiones de SQL Server que usarán los suscriptores de esta publicación.
El asistente configurará la publicación para que incluya sólo la funcionalidad compatible con todos los tipos de suscriptores especificados.
SQL Server 2000
SQL Server para Windows CE
Requiere que los archivos de instantánea tengan formato de caracteres.
Ayuda
< Atrás
Siguiente >
Finalizar >>|
Cancelar


Asistente para nueva publicación
Artículos
Seleccione las tablas y otros objetos que desee publicar como artículos.
Seleccione columnas para filtrar las tablas.
Objetos para publicar:
Tablas
actividad_efectiva (dbo)
actividad_frecuencia (dbo)
actividad_teorica (dbo)
activo (dbo)
area (dbo)
asignacion_grupo_a_usuario (dbo)
centro (dbo)
detalle_mantencion (dbo)
ext_proveedor (dbo)
ext_sucursal (dbo)
frecuencia_tipo_activo (dbo)
grupo_usuarios (dbo)
hito_mantencion_efectiva (dbo)
hito_mantencion_teorica (dbo)
Propiedades del artículo
Mostrar sólo los objetos seleccionados en la lista
Ayuda
< Atrás
Siguiente >
Finalizar >>|
Cancelar

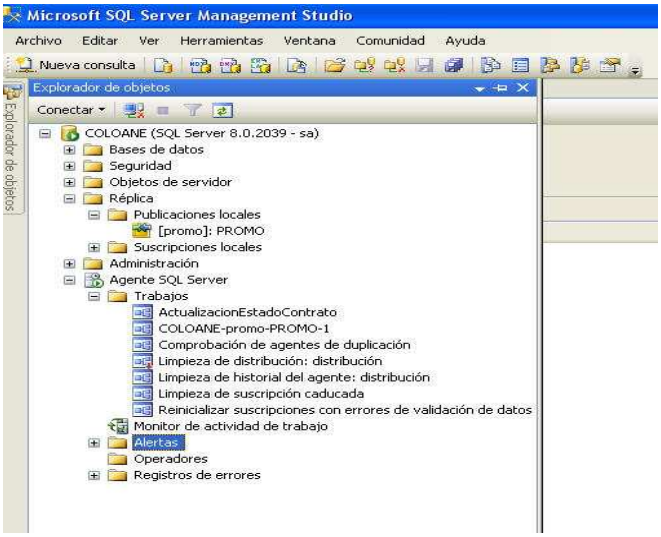

Asistente para nueva publicación
Problemas de los artículos
Los siguientes problemas pueden requerir cambios en la aplicación para que ésta continúe funcionando correctamente.
Problemas:
Se agregarán columnas Uniqueidentifier a las tablas.
Descripción:
Todos los artículos de mezcla deben contener una columna uniqueidentifier con un índice único y con la propiedad ROWGUIDCOL. SQL Server agrega una columna uniqueidentifier a las tablas publicadas que no la tienen cuando se genera la primera instantánea.
Al agregar una nueva columna, ocurre lo siguiente:
» Las instrucciones INSERT sin listas de columnas dan error
» Aumenta el tamaño de la tabla
» Aumenta el tiempo necesario para generar la primera instantánea
SQL Server agrega una columna uniqueidentifier con un índice único y la propiedad ROWGUIDCOL a cada una de las tablas siguientes.
Ayuda
< Atrás
Siguiente >
Finalizar >>|
Cancelar


Agregar filtro
1. Seleccione la tabla que desea filtrar.
actividad_efectiva (dbo)
2. Finalice la instrucción de filtro para identificar las filas de la tabla que recibirán los suscriptores.
Instrucciones de ejemplo
Columnas:
id_actividad_teorica (D_IDENTIF
id_mantencion_efectiva_activo (D
estado_actividad (D_VERDADEF
descripcion_actividad_efectiva (D
Instrucción de filtro:
SELECT <published_columns> FROM [dbo].[actividad_efectiva] WHERE
Aceptar
Cancelar
Ayuda


Asistente para nueva publicación
Agente de instantáneas
Especifique cuándo se debe ejecutar el Agente de instantáneas.
Las suscripciones se inicializan con una instantánea de esquema de publicación y datos. El Agente de instantáneas crea la instantánea.
Crear una instantánea inmediatamente
Programar el Asistente de instantáneas para ejecutarse:
Sucede cada día cada 10 minutos entre las 0:00:00 y las 23:59:59. Se utilizará la programación que empieza el 02-06-2008.
Cambiar...
Si planea cambiar las propiedades de la instantánea, no inicie el Agente de instantáneas hasta que haya cambiado las propiedades en el cuadro de diálogo de propiedades de la publicación.
Ayuda
< Atrás
Siguiente >
Finalizar >>|
Cancelar

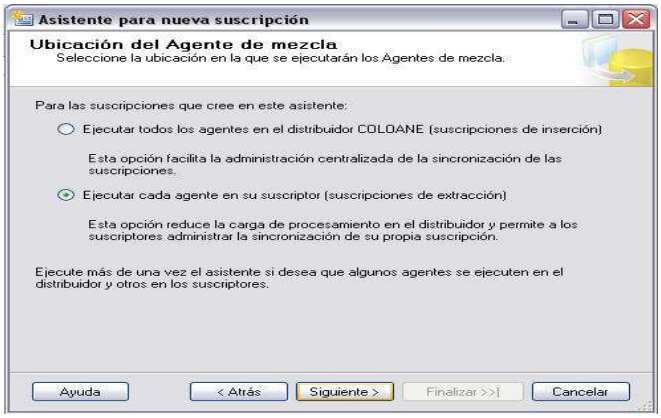

Asistente para nueva publicación
Acciones del asistente
Elija qué sucede cuando se hace clic en Finalizar.
Al finalizar el asistente:
Crear la publicación
Generar un archivo de secuencia de comandos con los pasos para crear la publicación
Ayuda
< Atrás
Siguiente >
Finalizar >>I
Cancelar

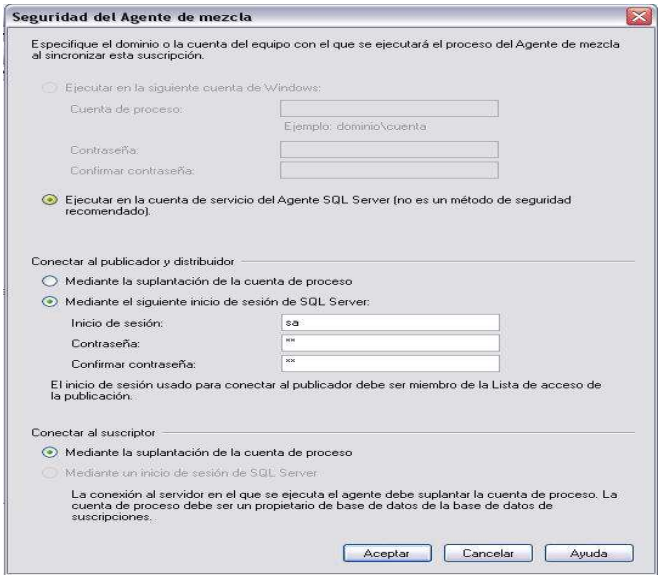

Asistente para nueva publicación
Finalización del asistente
Compruebe las opciones seleccionadas en el asistente y haga clic en Finalizar.
Nombre de publicación:
PROMO
Haga clic en Finalizar para realizar las siguientes acciones:
Cree la publicación.
El publicador 'COLOANE' se configurará con las siguientes opciones:
El publicador actuará como su propio distribuidor.
Configurar el servicio del Agente SQL Server en 'COLOANE' para que se inicie automáticamente cuando se inicie el equipo.
Utilizar 'C:\Archivos de programa\Microsoft SQL Server\MSSQL\ReplData' como carpeta raíz de instantáneas para los publicadores que usen este distribuidor.
Se creará una publicación con las siguientes opciones:
Crear una publicación de mezcla de la base de datos 'promo'.
El nivel de compatibilidad de la publicación admitirá suscriptores que sean servidores que ejecutan SQL Server 2000 o posterior.
Publicar las tablas siguientes como artículos:
'actividad_efectiva'
'actividad_frecuencia'
Ayuda
< Atrás
Siguiente >
Finalizar
Cancelar

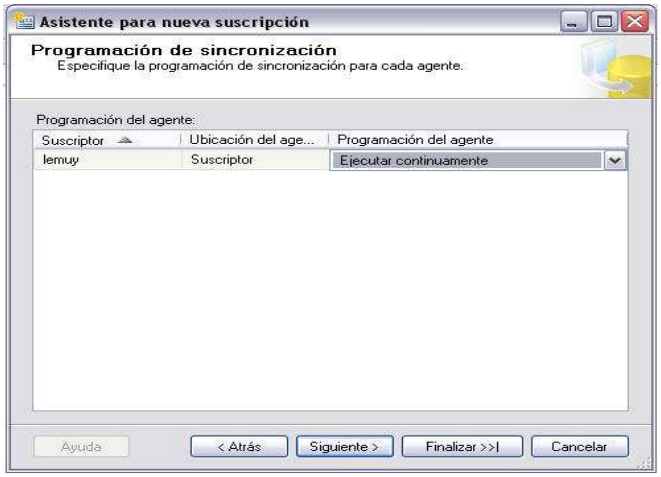

Microsoft SQL Server Management Studio
Archivo
Editar
Ver
Herramientas
Ventana
Comunidad
Ayuda
Nueva consulta
Explorador de objetos
Conectar
COLOANE (SQL Server 8.0.2039 - sa)
Bases de datos
Seguridad
Objetos de servidor
Réplica
Publicaciones locales
[promo]: PROMO
Suscripciones locales
Administración
Agente SQL Server
Trabajos
ActualizacionEstadoContrato
COLOANE-promo-PROMO-1
Comprobación de agentes de duplicación
Limpieza de distribución: distribución
Limpieza de historial del agente: distribución
Limpieza de suscripción caducada
Reinicializar suscripciones con errores de validación de datos
Monitor de actividad de trabajo
Alertas
Operadores
Registros de errores

Anexo Nº [3]: Configuración de una Suscripción con motor de base de datos SQL SERVER 2005 EXPRESS


Asistente para nueva suscripción
Publicación
Seleccione la publicación para la que desea crear una o más suscripciones.
Publicador:
coloane
Bases de datos y publicaciones:
promo
PROMO
Ayuda
< Atrás
Siguiente >
Finalizar >>|
Cancelar

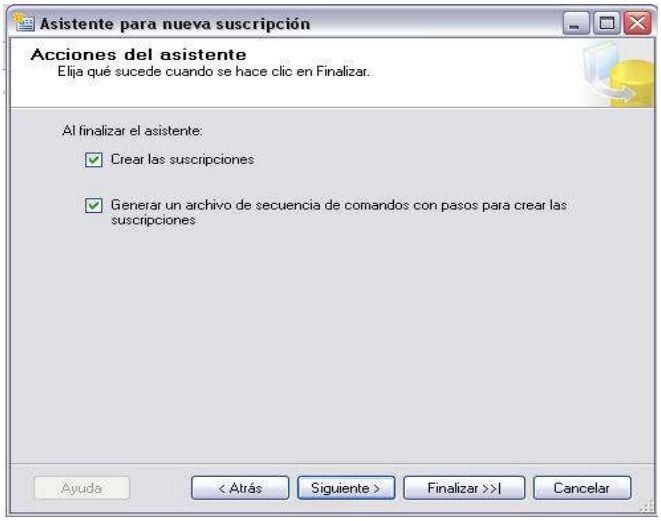

Asistente para nueva suscripción
Ubicación del Agente de mezcla
Seleccione la ubicación en la que se ejecutarán los Agentes de mezcla.
Para las suscripciones que cree en este asistente:
Ejecutar todos los agentes en el distribuidor COLOANE (suscripciones de inserción)
Esta opción facilita la administración centralizada de la sincronización de las suscripciones.
Ejecutar cada agente en su suscriptor (suscripciones de extracción)
Esta opción reduce la carga de procesamiento en el distribuidor y permite a los suscriptores administrar la sincronización de su propia suscripción.
Ejecute más de una vez el asistente si desea que algunos agentes se ejecuten en el distribuidor y otros en los suscriptores.
Ayuda
< Atrás
Siguiente >
Finalizar >>|
Cancelar

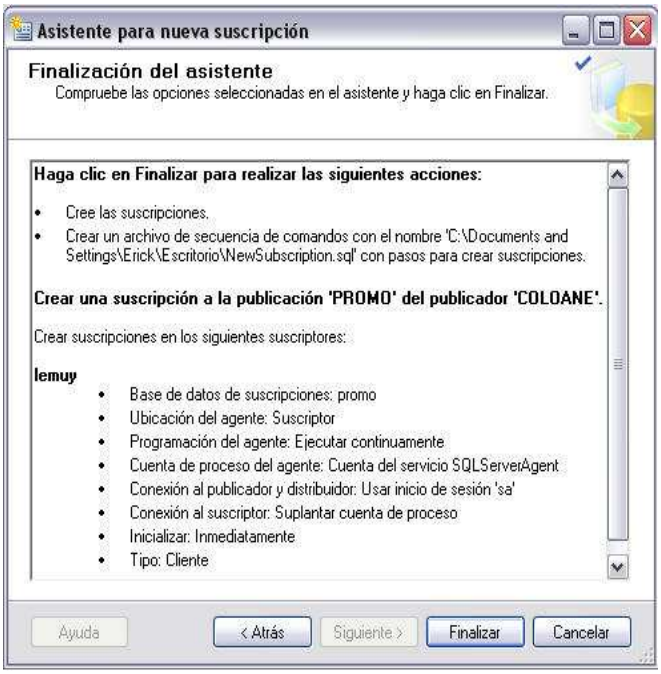

Seguridad del Agente de mezcla
Especifique el dominio o la cuenta del equipo con el que se ejecutará el proceso del Agente de mezcla al sincronizar esta suscripción.
Ejecutar en la siguiente cuenta de Windows:
Cuenta de proceso:
Ejemplo: dominio\cuenta
Contraseña:
Confirmar contraseña:
Ejecutar en la cuenta de servicio del Agente SQL Server (no es un método de seguridad recomendado).
Conectar al publicador y distribuidor
Mediante la suplantación de la cuenta de proceso
Mediante el siguiente inicio de sesión de SQL Server:
Inicio de sesión:
sa
Contraseña:
Confirmar contraseña:
El inicio de sesión usado para conectar al publicador debe ser miembro de la Lista de acceso de la publicación.
Conectar al suscriptor
Mediante la suplantación de la cuenta de proceso
Mediante un inicio de sesión de SQL Server
La conexión al servidor en el que se ejecuta el agente debe suplantar la cuenta de proceso. La cuenta de proceso debe ser un propietario de base de datos de la base de datos de suscripciones.
Aceptar
Cancelar
Ayuda

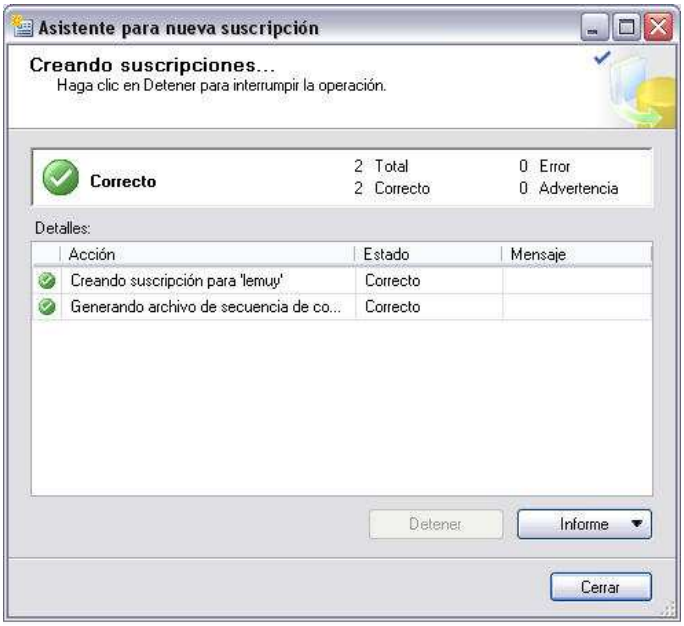

Asistente para nueva suscripción
Programación de sincronización
Especifique la programación de sincronización para cada agente.
Programación del agente:
Suscriptor
Ubicación del age...
Programación del agente
lemuy
Suscriptor
Ejecutar continuamente
Ayuda
< Atrás
Siguiente >
Finalizar >>|
Cancelar

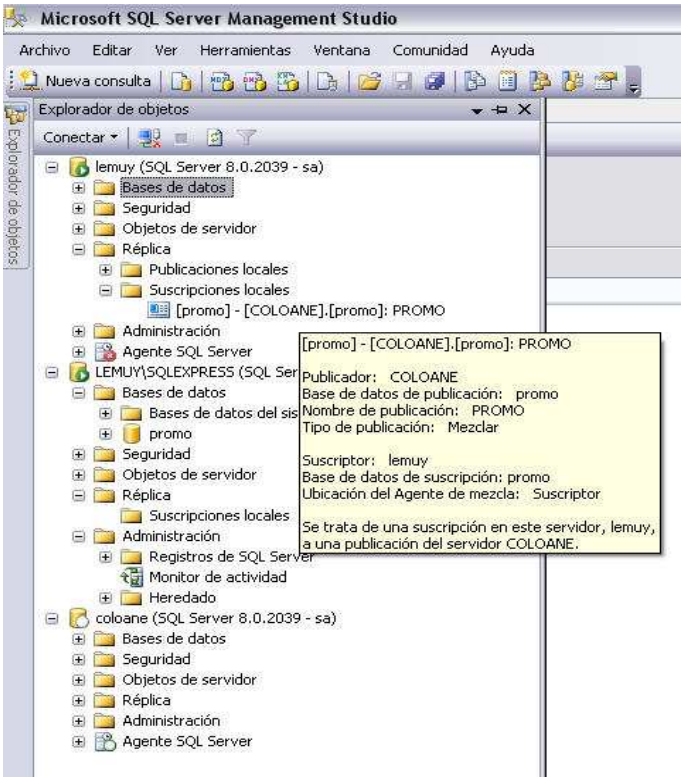

Asistente para nueva suscripción
Inicializar suscripciones
Especifique si desea inicializar las suscripciones con una instantánea de los datos y el esquema de publicación.
Propiedades de suscripción:
Suscriptor
Inicializar
Inicializar cuando
lemuy
Inmediatamente
Es necesario inicializar una base de datos de suscripciones con una instantánea de los datos y el esquema de publicación, salvo que ya se haya preparado específicamente para la suscripción.
Ayuda
< Atrás
Siguiente >
Finalizar >>|
Cancelar


Asistente para nueva suscripción
Tipo de suscripción
Especifique el tipo de cada suscripción y asigne una prioridad para la resolución de conflictos.
Propiedades de suscripción:
Suscriptor
Tipo de suscripción
Prioridad para la resolución de conflictos
lemuy
Cliente
Gana el primero en el publicador
La suscripción de un servidor puede volver a publicar los datos en otros suscriptores y ser un asociado de sincronización con otros suscriptores. Tiene su propia prioridad, un número entre 0 (prioridad más baja) y 99,99 (prioridad más alta), para la resolución de conflictos de datos. Además, los cambios realizados en artículos de sólo descarga en el suscriptor se replican en el publicador.
Ayuda
< Atrás
Siguiente >
Finalizar >>|
Cancelar


Asistente para nueva suscripción
Acciones del asistente
Elija qué sucede cuando se hace clic en Finalizar.
Al finalizar el asistente:
Crear las suscripciones
Generar un archivo de secuencia de comandos con pasos para crear las suscripciones
Ayuda
< Atrás
Siguiente >
Finalizar >>|
Cancelar


Asistente para nueva suscripción
Finalización del asistente
Compruebe las opciones seleccionadas en el asistente y haga clic en Finalizar.
Haga clic en Finalizar para realizar las siguientes acciones:
Cree las suscripciones.
Crear un archivo de secuencia de comandos con el nombre 'C:\Documents and Settings\Erick\Escritorio\NewSubscription.sql' con pasos para crear suscripciones.
Crear una suscripción a la publicación 'PROMO' del publicador 'COLOANE'.
Crear suscripciones en los siguientes suscriptores:
lemuy
Base de datos de suscripciones: promo
Ubicación del agente: Suscriptor
Programación del agente: Ejecutar continuamente
Cuenta de proceso del agente: Cuenta del servicio SQLServerAgent
Conexión al publicador y distribuidor: Usar inicio de sesión 'sa'
Conexión al suscriptor: Suplantar cuenta de proceso
Inicializar: Inmediatamente
Tipo: Cliente
Ayuda
< Atrás
Siguiente >
Finalizar
Cancelar


Asistente para nueva suscripción
Creando suscripciones...
Haga clic en Detener para interrumpir la operación.
Correcto
2 Total
2 Correcto
0 Error
0 Advertencia
Detalles:
Acción
Estado
Mensaje
Creando suscripción para 'lemuy'
Correcto
Generando archivo de secuencia de co...
Correcto
Detener
Informe
Cerrar


Microsoft SQL Server Management Studio
Archivo
Editar
Ver
Herramientas
Ventana
Comunidad
Ayuda
Nueva consulta
Explorador de objetos
Conectar
lemuy (SQL Server 8.0.2039 - sa)
Bases de datos
Seguridad
Objetos de servidor
Réplica
Publicaciones locales
Suscripciones locales
[promo] - [COLOANE].[promo]: PROMO
Administración
Agente SQL Server
LEMUY\SQLEXPRESS (SQL Ser
Bases de datos
Bases de datos del sis
promo
Seguridad
Objetos de servidor
Réplica
Suscripciones locales
Administración
Registros de SQL Server
Monitor de actividad
Heredado
coloane (SQL Server 8.0.2039 - sa)
Bases de datos
Seguridad
Objetos de servidor
Réplica
Administración
Agente SQL Server
[promo] - [COLOANE].[promo]: PROMO
Publicador: COLOANE
Base de datos de publicación: promo
Nombre de publicación: PROMO
Tipo de publicación: Mezclar
Suscriptor: lemuy
Base de datos de suscripción: promo
Ubicación del Agente de mezcla: Suscriptor
Se trata de una suscripción en este servidor, lemuy, a una publicación del servidor COLOANE.

Anexo Nº [4]: Ejemplo Plantilla Caso de Prueba utilizado en la fase de Pruebas del Sistema

Nº Caso: 1	**Tipo Clase**: Valores Límites	**Formulario.: ver_ActivosInfoBasica.aspx**
Condición Externa:	Campo factura_activo debe admitir sólo valores numéricos [0-9] factura_activo: integer, acepta valores NULL	
Clases Válidas:	**1.** Valor factura_activo = 20	
Clases Inválidas:	**2.** Valor factura_activo = 123456 **3.** Valor factura_activo = 34A **4.** Valor factura_activo = ZZ **5.** Valor factura_activo = 123: **6.** Valor factura_activo = “ ”	

Nº de Caso:	**Fecha de Prueba:**	
Clase	**Resultado Esperado**	**Resultado Obtenido**
1	Valor aceptado	Valor aceptado
2	Valor Rechazado	Valor Rechazado
3	Valor Rechazado	Valor Rechazado
4	Valor Rechazado	Valor Rechazado
5	Valor Rechazado	Valor Rechazado
6	Valor Rechazado	Valor Rechazado

Anexo Nº [5]: Especificación de un Caso de Prueba.

Caso de Prueba - 01	**Autenticación usuario al sistema**
Descripción:	El sistema deberá comportarse tal y como se describe a continuación:
Secuencia Normal:	1. El actor solicita entrar en el sistema. 2. El sistema solicita las credenciales al usuario. 3. El actor proporciona identificador y contraseña. 4. Si el actor se encuentra registrado en el sistema, el sistema permite el acceso al usuario.
Excepciones:	1. Si el usuario introdujo una credencial incorrecta, el sistema muestra un mensaje impidiéndole el ingreso al sistema. 2. si el identificador no esta registrado, el sistema vuelve al paso 2, muestra un mensaje de aviso y continúa el caso de uso.
Notas:	Consultar el requisito de almacenamiento etiquetado con el código RA – 01.