

TD-0087/05-02

658.3011
A15
S
TD_0087_05_02



Universidad de Ciencias Informáticas

Departamento de Informatización

SISTEMA DE EVALUACIÓN DEL DESEMPEÑO

SED

Documento de tesis para optar por el título de Ingeniero Informático

Autor: Isrrael Alba Reyes

Tutor: Lida Gonzáles Alvarez

Resumen

Este proyecto consiste en la creación de un sistema que sea capaz de facilitar el proceso de evaluación del desempeño de los trabajadores en la Universidad de las Ciencias Informáticas (UCI).

Al no existir en la universidad previamente ningún sistema de evaluación del desempeño, el trabajo surge a partir de la idea de crear un software en recursos humanos para controlar y guardar la información de la evaluación del desempeño de los trabajadores de la institución, aprovechando el equipamiento informático existente, garantizar el acceso del personal de recursos humanos autorizados a las evaluaciones de los trabajadores de una forma eficiente, organizada, segura y cómoda.

Índice

Introducción	9
Capítulo 1. Fundamentación teórica	13
1.1 INTRODUCCIÓN	13
1.2 EVALUACIÓN DEL DESEMPEÑO.....	13
1.3 ESTADO DEL ARTE	14
1.4 TÉCNICAS, TECNOLOGÍAS, METODOLOGÍAS Y SOFTWARE PARA LA ELABORACIÓN DEL SISTEMA.....	17
1.4.1. Intranet	17
1.4.2. El navegador (Browser).....	17
1.4.3. Aplicaciones Web	18
1.4.4. Modelo cliente servidor en tres capas.....	18
1.4.5. RUP.....	19
1.4.6. UML.....	21
1.4.7. Rational Rose.....	22
1.4.8. .Net.....	23
1.4.9. Asp.net.....	24
1.4.10. SQL Server.....	25
1.5 CONCLUSIONES	26
Capítulo 2. Características del sistema	27
2.1. INTRODUCCIÓN	27
2.2. OBJETO DE ESTUDIO.....	27
2.2.1. Definición del problema.....	27
2.2.2. Objetivos estratégicos.....	27
2.2.3. Análisis de como se ejecutan los procesos actualmente.....	28
2.3. DESCRIPCIÓN DETALLADA DE LOS PROCESOS A AUTOMATIZAR.....	28
2.4. PROPUESTA DEL SISTEMA.....	29
2.5. MODELO DEL NEGOCIO.....	29
2.6. ESPECIFICACIÓN DE LOS REQUISITOS DE SOFTWARE.....	30
2.6.1. Requerimientos Funcionales.....	30
2.6.2. Requerimientos no Funcionales.....	31
2.7. DEFINICIÓN DE LOS CASOS DE USO	31
CONCLUSIONES	31
Capítulo 3. Análisis y diseño del sistema	32
3.1. INTRODUCCIÓN	32
3.2. ANÁLISIS	32
3.2.1. Diagrama de clases de análisis	33
3.3. DISEÑO.....	34
3.3.1. Diagramas de interacción.....	34
3.3.2. Diagrama de clases.....	34
3.3.2.1. Capa de datos	35
3.3.2.2. Capa de negocio.....	36
3.3.3. Diagrama de clases Web	37
3.3.4. Descripción de las Clases	40
3.3.5. Diagrama entidad-relación.....	51
3.3.6. Tratamiento de errores	54
3.3.7. Diseño de la Interfaz.....	54
3.3.8. Concepción general de la ayuda.....	55
3.4. CONCLUSIONES	55
Conclusiones generales	56
Recomendaciones	57
Referencias Bibliográficas	58

Bibliografía	59
Anexo 1. Modelo del negocio.....	60
Anexo 2. Definición de los casos de uso del sistema	73
Anexo 3. Diagramas de interacción.....	82
Glosario de Términos	99

Introducción

El profesional de recursos humanos enfrenta diversos desafíos en el logro de su objetivo en las organizaciones, para el mejor desarrollo de la organización a la cual están integrados, sin olvidar el contexto social en el cual se desenvuelven.

En términos concretos, el principal desafío del especialista en recursos humanos es lograr el mejoramiento permanente de las organizaciones de las que forman parte, haciéndolas más eficientes y más eficaces, de manera que sean aceptables para la sociedad. Estos dos factores conducen a mejores niveles de productividad.

En la Universidad de Ciencias Informáticas (UCI) los métodos tradicionales manuales para mantener la información del personal no son viables en una organización relativamente grande. Los documentos que competen al departamento de Recursos humanos se encuentran en archivos manuales y el volumen a medida que valla pasando el tiempo es considerable porque corresponderá a documentos generales a través de los años de funcionamiento de la institución... Generar cualquier tipo de información se hace difícil, pues el proceso es manual e implica horas de trabajos.

La UCI, a través del departamento de Recursos humanos., necesitan gestionar fácilmente la información referente al personal de tal forma que responda con rapidez y agilidad a las diversas consultas y a optimizar la comunicación con los demás sectores de la institución generando información confiable, precisa y oportuna. La solución para este problema es informatizar Recursos Humanos.

La tarea de evaluar el desempeño constituye un aspecto básico de la gestión de recursos humanos en las organizaciones. La evaluación del desempeño constituye una función esencial que de una u otra manera suele efectuarse en toda organización moderna. Al evaluar el desempeño la organización obtiene información para la toma de decisiones: Si el desempeño es inferior a lo estipulado, deben emprenderse acciones correctivas; si el desempeño es satisfactorio, debe ser alentado.

Contar con un sistema formal de evaluación de desempeño permite a la gestión de recursos humanos evaluar sus procedimientos. Las decisiones sobre promociones, compensaciones y adiestramiento, un seguimiento del desempeño de sus trabajadores de donde sale como esta dicha institución y tomar dediciones en base de esta información, garantizado información sistemática y documentada proveniente del sistema de evaluación de desempeño.

Actualmente, departamento de Recursos humanos de la Universidad de Ciencias Informáticas (UCI) busca mejorar la calidad, rapidez y fácil gestión de la información sobre las evaluaciones de los Trabajadores de la institución, informatizando esta área, implementando un software de evaluación de desempeño.

El problema radica en: ¿Cómo implementar las evaluaciones de desempeño de los trabajadores de la Universidad de Ciencias Informáticas en un software que permita a la vez mejorar el flujo, gestión y calidad de la información?

Con la implantación del sistema de evaluación de desempeño ,las oficinas de recursos humanos podrán monitorear muy fácilmente las evaluaciones de los trabajadores de la entidad obteniendo así la información que requiere de manera rápida y sencilla mejorando la eficiencia de esta. Se contara con un control de usuario y acceso para restringir la entrada al sistema y solo pueda consultar la información los usuarios autorizados.

Para la realización de este sistema es necesario un estudio profundo y detallado a cerca de cuales son los aspectos con los que debe contar un sistema de evaluación del desempeño. Además de la tecnología Web para el desarrollo de aplicaciones y el análisis del sistema gestor de bases de datos a utilizar. Es imprescindible realizar un estudio de los diferentes artefactos de análisis y diseño en términos de ingeniería de software aplicados al sistema propuesto; con el fin de lograr una aplicación robusta y capaz de enfrentar los fallos del sistema y que solucione los problemas actuales.

El campo de acción de este trabajo está centrado en los sistemas de evaluación del desempeño orientados a recursos humanos.

Este trabajo tiene como objetivo general automatizar los servicios de los recursos humanos referidos a la evaluación del desempeño de los trabajadores de la institución, con un ambiente Web que permita y facilite la gestión y actualización de la información de las evaluaciones y controle la seguridad del mismo.

Como objetivos específicos tenemos:

- Analizar aspectos teóricos conceptuales de los Sistemas de evaluación de desempeño.
- Estudiar acerca de los Sistemas evaluación de desempeño que existen en el mundo.
- Implementar un Sistemas de evaluación de desempeño que se ajuste a los requerimientos de recursos humanos de la Universidad de Ciencias Informáticas.
- Implementar la seguridad del sistema.
- Implementar el sistema sobre tecnología Web para mejor uso del mismo en la red.

Tareas a desarrollar para cumplir los objetivos planteados.

- Estudio de la política de evaluación de desempeño.
- Hacer un estudio preliminar del problema y la situación actual.
- Obtener los requisitos del sistema.
- Hacer un análisis y discutir la arquitectura de la aplicación.
- Utilizar RUP como metodología de análisis y diseño de este trabajo.
- Obtener el modelo de datos más idóneo.
- Implantar el sistema de evaluación del desempeño en la Universidad de Ciencias Informáticas (UCI).

Este trabajo está dividido en tres capítulos que contienen toda la información referente a la investigación realizada, análisis y diseño del sistema a desarrollar, la información está distribuida en tres capítulos:

Capítulo I: Fundamentación teórica: Es donde se realiza un estudio del estado del arte, se describen los principales aspectos de las herramientas a utilizar para la implementación de la aplicación.

Capítulo II: Características del sistema: En este capítulo se realiza un estudio preliminar del sistema y el proceso de negocio, en el cual se especifican los objetivos y los requerimientos esperados.

Capítulo III: Análisis y diseño del sistema: aquí se muestra el diagrama de clases del análisis. En la parte del diseño se exponen los detalles relacionados con el diseño del sistema propuesto. Se utilizan para su modelado los diagramas de interacción. También se muestra el diagrama de clases Web.

Capítulo 1. Fundamentación teórica

1.1 Introducción

En este capítulo se hará un estudio detallado de los principales conceptos y tecnologías utilizadas en el desarrollo del sistema de evaluación del desempeño. Se abordan temas relacionados con la evaluación del desempeño, el uso de las tecnologías Web, lenguajes de programación, Sistemas de Gestión de Bases de datos (SGBD), etc. En este capítulo se persigue el propósito de dar a conocer el estado del arte en los Sistemas de evaluación de desempeño y la necesidad de decidir cuales de las tecnologías utilizar en el desarrollo de este trabajo.

1.2 Evaluación del desempeño

Se evalúa la actuación del ser humano desde el momento de nacer hasta el morir. Lo pesan, lo miden y lo evalúan de diferentes formas. De hecho, la evaluación se realiza antes del nacimiento, se evalúan el ritmo cardiaco, la posición fetal y otros tipos de componentes. Por supuesto que en el mundo laboral los trabajadores son sometidos a diferentes clases de evaluaciones, algunas de ellas pueden conducir a la promoción de nuevas posiciones, generalmente con un sueldo y/o un cargo y por supuesto con unos privilegios más elevados. Dentro de ese complejo mundo organizacional las recompensas directas e indirectas se pueden conseguir con un reconocimiento especial o un nuevo escenario de posibilidades de autoexpresión y de un mundo de creatividad e innovación. La evaluación se identifica, con excesiva frecuencia con los test o exámenes, es decir con las pruebas que se realizan tanto en el ámbito teórico, así como también en el sitio de trabajo.

En las organizaciones siempre existen dos grupos dentro del proceso de evaluación del desempeño. Unos que fungen como evaluadores y otros que son evaluados. Los que ocupan posiciones de liderazgo evalúan formalmente el desempeño de sus más cercanos seguidores. La actuación total de un seguidor es evaluada durante un período de tiempo determinado. Si el resultado es positivo, puede ser recompensado. Pero; si el

desempeño se encuentra por debajo del estándar los buenos líderes pueden intentar corregirlo o dependiendo de la naturaleza del resultado, que en este caso es desviación sobre lo esperado, se debe entonces reenseñar al seguidor. Las interpretaciones acerca de los resultados del desempeño se convierten en elementos de control, debido a que los trabajadores tienden a comportarse de manera que se observen bien con respecto a los criterios por los cuales han sido evaluados.

Para llevar a cabo cualquier proceso de evaluación de desempeño, es importante y muy pertinente conocer a cabalidad las políticas, las normas y los sistemas de reclutamiento, selección y empleo, así como también los de promoción, de disciplina, de desarrollo y por supuesto de evaluación del desempeño, de compensaciones, beneficios e incentivos no monetarios. Sin lugar a dudas, por cuanto ellos fijan el parámetro de actuación dentro de los escenarios organizacionales, además, que determinan la actuación de los participantes del diseño, y son determinantes de la posición actitudinal que puedan tener. En forma muy especial los programas de entrenamiento y desarrollo deben ser un punto en donde se debe profundizar, ya que el diseño e implantación de una nueva técnica reclamará la aprehensión de nuevos conocimientos y habilidades por parte de los trabajadores. Estas nuevas realidades conllevan a que los líderes organizacionales perciban la gestión de recursos humanos como una verdadera estrategia competitiva y que los procesos de evaluación de desempeño tanto informales como formales, explícitos o implícitos que existen en la organización deben ser tomados muy en cuenta.

1.3 Estado del arte

En el mundo de la informática se han desarrollado un gran número de sistemas capaces de optimizar y facilitar las labores más comunes en la evaluación del desempeño. Entre esto se encuentran:

QTraining (Software para Evaluación de Desempeño y Control de Capacitación

).

QTraining es una poderosa pero sencilla herramienta para evaluar el desempeño del personal, detectar necesidades de capacitación y administrar los cursos de capacitación de la empresa. Permite evaluar el impacto de las capacitaciones llevadas a cabo y relaciona los objetivos de desempeño con los cursos de capacitación.

KHRONOS

Gestión de Actividades

Desarrollador: Software House Ltda.

País : COLOMBIA

Versión : 3.06

Año : 2003

Mercado Objetivo: N.E.

Sistema administrativo orientado a la gestión de actividades, evaluación de desempeño e indicadores de cumplimiento; todo esto con el objetivo de mejorar la calidad y la productividad de las actividades a realizar de cada uno de los empleados de su empresa, obteniendo como resultado una serie de ventajas estratégicas, que permitirán marcar una diferencia competitiva. Khronos posee un poderoso sistema estadístico, cuya funcionalidad es realizar una visualización gráfica del cumplimiento del recurso humano frente a las labores asignadas, generando indicadores de cumplimiento, los cuales despliegan gráficamente la puntuación obtenida por cada uno de los usuarios, grupos o sucursales de acuerdo a las actividades asignadas en un período de tiempo. Khronos cuenta con una serie de funcionalidades que hacen del sistema una herramienta agradable y útil para el control, seguimiento, planeación, organización y cumplimiento de las labores en la compañía, con el fin de marcar un ritmo de trabajo dentro de ella, mejorando la productividad de cada una de las personas que la conforman.



Módulos de Gestión Humana y Beneficios

Evaluación del Desempeño

Servicios y Beneficios

- Información inmediata del desempeño del personal, a cualquier nivel (directivo o subalterno).
- Agilidad en la toma de decisiones que involucran el desempeño del personal.
- Seguimiento en el cumplimiento de metas.
- Registro de lo planeado y ejecutado del período evaluado.
- Capacitación del empleado por requerimiento de la evaluación.

eFeedback

eFeedback es un sistema, desarrollado por el **Instituto de Ingeniería del Conocimiento (IIC)**, para evaluar el desempeño profesional de los individuos mediante la medición de una serie de *competencias*, de forma objetiva y precisa. Para ello, eFeedback permite desde la evaluación por el superior Inmediato hasta una medición 360°, en la que se toma como Referencia al propio evaluado y a su equipo de trabajo habitual (Colaboradores y personas a su cargo).

Con la metodología de eFeedback se puede medir no sólo las competencias que determinan el nivel de desempeño, sino también el contexto laboral en el que se desarrollan.

Cognos Metrics Manager

Es el nuevo producto de la empresa de software canadiense Cognos. La compañía ya está definiendo para sí misma una nueva categoría: administración del desempeño corporativo (Corporate Performance Management, CPM). Cognos Metrics Manager es el ejemplo más reciente de esta visión; es un software que permite a las compañías medir su ejecución contra su estrategia de negocio. Contribuye a que las organizaciones supervisen, analicen y creen reportes activamente en todos los niveles de la empresa.

GIRH - Gestión Integral de Recursos Humanos

GIRH es un producto innovador, específicamente diseñado para empresas e instituciones. GIRH no solo le permite realizar las operaciones de cálculo de sueldos y

jornales, sino que cuenta con un conjunto de módulos integrados para realizar evaluaciones, seguimiento estadístico y gestión de los diferentes aspectos de la administración de los recursos humanos de su empresa.

Estructura modular del sistema:

- ▣ Liquidación de Haberes
- ▣ Registro y administración de licencias
- ▣ Ficha Funcional
- ▣ **Evaluación de Desempeño**
- ▣ Capacitación y Medición de Impacto
- ▣ Reclutamiento y Selección
- ▣ Satisfacción de Actores Clave

1.4 Técnicas, tecnologías, metodologías y software para la elaboración del sistema.

1.4.1. Intranet

Una intranet se define como un conjunto de computadoras interconectadas para un uso específico y generalmente interno. La intranet se puede decir que es una versión en menor escala de Internet, y generalmente de ámbito privado que, al utilizar la misma tecnología TCP/IP que Internet, permite tener todos los servicios diseñados para esta tecnología, como son, por ejemplo, programas clientes y servidores de correo electrónico, FTP, Web , compartir archivos , discos duros , etc.

1.4.2. El navegador (Browser)

El navegador es el software especial que se utiliza para acceder a información del World Wide Web y las informaciones depositadas en los Servidores Web de una Intranet o de Internet. El navegador permite al usuario navegar entre páginas relacionadas entre sí por vínculos, de manera que proporciona al servidor información sobre dónde está depositada la página que quiere ver, y automáticamente nos muestra los documentos llamados páginas Web. Los navegadores son programas comerciales. Los dos de más

utilización en la actualidad son los comercializados por Microsoft (Internet Explorer) y por Netscape Communications (Netscape Navigator). Estos programas trabajan de forma similar y ponen a disposición del usuario el mismo tipo de documentos.

1.4.3. Aplicaciones Web

La Web nos permite que una vez que la información esté disponible, se pueda acceder a ella desde cualquier computadora, desde cualquier país, por cualquier persona autorizada, usando un simple navegador.

El contenido de una página Web está basado en un lenguaje llamado *Hypertext Markup Language* (HTML), un lenguaje muy sencillo que permite describir hipertexto, es decir, texto representado por una forma estructurada y agradable, con enlaces que conducen a otros documentos o fuentes de información relacionadas, y con inserciones multimedia, imágenes, sonido, entre otros. Las aplicaciones Web poseen una interfaz HTML para recibir y mostrar la información, por ejemplo, se podría crear una aplicación que permita a los usuarios interactuar con la información almacenada en una Base de Datos.

Básicamente las aplicaciones Web se desarrollan en un ambiente Cliente-Servidor donde la mayor parte del procesamiento se realiza en el servidor. El cliente o un navegador (*browser*) es el responsable de mostrar la información al usuario y de hacer validaciones sencillas en la entrada de datos para que la información sea mostrada al usuario. Para la programación de aplicaciones Web existen dos tecnologías fundamentales, del lado del cliente y del lado del servidor.

1.4.4. Modelo cliente servidor en tres capas

Este modelo se basa fundamentalmente en un aplicación, la cual está dividida en tres módulos que tienen diferentes propósitos e interactúan entre sí pero de forma independiente, de forma tal que al hacer cambios en una no se afectan los otros dos , básicamente estos módulos o capas reciben el nombre de:

Capa Cliente (Aquí es donde va la interfaz de usuario, esta capa es la encargada de hacer un pedido y mostrar la información al usuario)

Capa intermedia o de negocio (Aquí es donde van implementadas todas las reglas de negocio, esta capa es la encargada de recibir el pedido del cliente, extrae los datos al interactuar con la capa de datos y le da una respuesta al cliente tal y como él la pidió)

Capa servidor o de datos (Aquí es donde están guardados los datos que serán utilizados por la aplicación)

1.4.5. RUP.

Evolución:

1967	Método Ericsson	Desarrollo Basado en Componentes (I. Jacobson)
1987	Proceso Objectory	Incorpora "Casos de Uso" y diseño OO.
1995-1997	P. Objectory de Rational (ROP)	Añade fases: Inicio, Elaboración, Construcción y Transición. Utiliza UML.
1998	P.Unificado de Rational 5.0	Método Unificado de I. Jacobson, G. Booch, J. Rumbaugh.
2000	P.Unificado de Rational 2000	Incorpora Método OO en Tiempo Real

Descripción:

El proceso unificado es un conjunto de actividades que transforman requisitos de usuario en un sistema software.

Consta de:

- a) Flujos de Trabajo Fundamentales:
 - Modelado del Negocio
 - Requisitos
 - Análisis y Diseño
 - Implementación

- Pruebas
 - Despliegue
 - Configuración y Gestión de Cambios, Gestión del Proyecto, Entorno.
- b) Fases: Inicio, Elaboración, Construcción, Transición
- Una Fase tiene 1 o varias Iteraciones y finaliza con un Hito.
 - Las 4 Fases producen una Generación.

RUP es:

- Iterativo e Incremental
 - Dirigido por los Casos de Uso
 - Centrada en la Arquitectura
- a) Iterativo e Incremental:
- Compuesto por pequeños proyectos que incorporan incrementalmente nueva funcionalidad y cuyo desarrollo es una iteración”
 - Inicio: Comprender el Problema y Determinar su Ámbito
 - Elaboración: Capturar Requisitos y Producir un Prototipo (Línea Base)
 - Construcción: Diseño e Implementación. 1º Producto Operacional (Beta)
 - Transición: Producir y Suministrar el Producto Final.
 - Características de un Proyecto Iterativo e Incremental
 - Obtiene un Sistema Robusto
 - Reduce el Riesgo de tener un mal producto
 - Reduce el Riesgo de no obtener el producto en el tiempo previsto
 - Permite atacar problemas con requisitos incompletos.
- b) Dirigido por los Casos de Uso:
- Casos de Uso:
 - Servicios que un actor requiere del sistema y le proporcionan un resultado
 - Proporcionan los Requisitos Funcionales del Sistema
 - Modelo de Casos de Uso:
 - Describen toda la funcionalidad del Sistema

- Filosofía de Desarrollo:
 - Modelo Casos de Uso → Modelo de Análisis → Modelo de Diseño → Modelo Implementación.
 - Trazabilidad del Sistema:
 - Cambios en Requisitos de un Caso de Uso fácil detectar las clases y componentes que afectan.
- c) Centrado en la Arquitectura
- Casos de Uso describen la Funcionalidad del Sistema
 - Arquitectura define la Forma del Sistema
 - Se describe mediante Vistas que incorporan el 5-10% de los casos de uso más relevantes:
 - Casos de Uso: Actores y Casos de Uso relevantes.
 - Análisis: Clases de Análisis (Interfaz, Control, Entidad) de C. Uso anteriores.
 - Diseño: Subsistemas y Clases de Diseño derivadas de las Clases de Análisis.
 - Despliegue: Arquitectura Física por medio de Nodos.
 - Implementación: Componentes de las clases relevantes.

1.4.6. UML.

El Lenguaje de Modelado Unificado UML es una especificación de notación orientada a objetos. Es un lenguaje de propósito general para el modelado orientado a objetos. UML es también un lenguaje de modelación visual que permite una abstracción del sistema y sus componentes, que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir.

UML no es un lenguaje de programación. Las herramientas pueden ofrecer generadores de código de UML para una gran variedad de lenguaje de programación, así como construir modelos por ingeniería inversa a partir de programas existentes.

UML también intenta solucionar el problema de propiedad de código que se da con los desarrolladores, al implementar un lenguaje de modelado común para todos los desarrollos se crea una documentación también común, que cualquier desarrollador con conocimientos de UML será capaz de entender, independientemente del lenguaje utilizado para el desarrollo.

Permite la modificación de todos sus miembros mediante estereotipos y restricciones. Un estereotipo nos permite indicar especificaciones del lenguaje al que se refiere el diagrama de UML. Una restricción identifica un comportamiento forzado de una clase o relación, es decir mediante la restricción estamos forzando el comportamiento que debe tener el objeto al que se le aplica.

UML es ahora un standard, no existe otra especificación de diseño orientado a objetos. Su utilización es independiente del lenguaje de programación y de las características de los proyectos, ya que UML ha sido diseñado para modelar cualquier tipo de proyectos, tanto informáticos como de arquitectura, o de cualquier otro ramo.

1.4.7. Rational Rose.

Rational Rose es la herramienta CASE que comercializan los desarrolladores de UML y que soporta de forma completa la especificación del UML.

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

Rational Rose utiliza un proceso de desarrollo iterativo controlado (controlled iterative process development), donde el desarrollo se lleva a cabo en una secuencia de iteraciones. Cada iteración comienza con una primera aproximación del análisis, diseño e implementación para identificar los riesgos del diseño, los cuales se utilizan para

conducir la iteración, primero se identifican los riesgos y después se prueba la aplicación para que éstos se hagan mínimos.

Cuando la implementación pasa todas las pruebas que se determinan en el proceso, ésta se revisa y se añaden los elementos modificados al modelo de análisis y diseño. Una vez que la actualización del modelo se ha modificado, se realiza la siguiente iteración.

Rose permite que haya varias personas trabajando a la vez en el proceso iterativo controlado, para ello posibilita que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo.

También es posible descomponer el modelo en unidades controladas e integrarlas con un sistema para realizar el control de proyectos que permite mantener la integridad de dichas unidades.

Se puede generar código en distintos lenguajes de programación a partir de un diseño en UML.

Rational Rose proporciona mecanismos para realizar la denominada Ingeniería Inversa, es decir, a partir del código de un programa, se puede obtener información sobre su diseño.

Rational también ofrece una alternativa a la plataforma .NET que permite crear aplicaciones de nuevas generaciones fiables y productivamente.

1.4.8. .Net.

Utiliza los Servicios Web como un medio para poder interoperar a distintas tecnologías. Permite conectar distintos sistemas operativos, dispositivos físicos, información y usuarios. Les da a los desarrolladores las herramientas y tecnologías para

hacer rápidamente soluciones de negocios que involucran distintas aplicaciones, dispositivos físicos y organizaciones.

La idea central detrás de la plataforma .NET es la de servicio. Más concretamente software como servicio y de cómo construir, instalar, consumir, integrar o agregar estos servicios para que puedan ser accedidos mediante Internet. Esto es posible debido a que tenemos la infraestructura de comunicación global que es Internet cada vez mas rápida y a un costo cada vez menor y además, a la capacidad de los procesadores que continúa incrementándose año tras año. El usuario de Internet puede con un explorador de Internet no solamente acceder a contenido como texto, imágenes o sonido, también puede hacer uso de servicios Web. Estos son los bloques de construcción o componentes sobre los cuales se basa el modelo de computación distribuida en Internet. La plataforma .NET permite usar Internet y su capacidad de distribución para que los usuarios accedan desde cualquier dispositivo, en cualquier sistema operativo y lugar a la funcionalidad que los servicios Web proveen.

Tanto la invocación de los servicios como su ejecución pueden ser hechas en cualquier dispositivo y sistema operativo, y accedido desde Internet. Los sitios se comunican entre sí y acceden a servicios y contenidos sin la intervención humana. Por eso se llama a la nueva generación de Internet "Internet inteligente".

1.4.9. Asp.net.

Cada vez es mayor el número de empresas que optan por usar la arquitectura de Internet, sus protocolos y servicios, como medio para conectar los sistemas de su propia empresa, así como la propia Internet como infraestructura de comunicación para conseguir esa conexión. En un entorno como el de Internet no podemos asumir que todos los clientes tengan un PC, funcionando con Windows como sistema operativo, para poder ejecutar las interfaces de usuario basadas en formularios Windows. Existen otros dispositivos y sistemas operativos, potenciales clientes a los que hay que ofrecer otro tipo de interfaz de acceso a nuestros servicios.

Una de las novedades más interesantes de la plataforma .Net es Asp.Net, el sucesor de Asp (Active Server Pages). Mediante Asp.Net podemos desarrollar aplicaciones, generalmente interfaces de usuarios, accesibles desde cualquier cliente que sea capaz de acceder a un servidor Web. Asp.Net se encarga de ejecutar la lógica en el servidor y generar el código HTML apropiado para el cliente que efectúe la solicitud, si importar el tipo de dispositivo, hardware o sistema operativo en que esté ejecutándose.

ASP.NET es un marco de trabajo de programación generado en CLR (Common Language Runtime) que puede utilizarse en un servidor para generar eficaces aplicaciones Web. ASP.NET ofrece varias ventajas importantes acerca de los modelos de programación Web anteriores:

- Mejor rendimiento.
- Compatibilidad con herramientas de primer nivel.
- Eficacia y flexibilidad.
- Simplicidad.
- Facilidad de uso.
- Escalabilidad y disponibilidad.
- Posibilidad de personalización y extensibilidad.
- Seguridad.

1.4.10. SQL Server.

En la actualidad, las compañías demandan una clase diferente de solución de base de datos. El rendimiento, la escalabilidad y la confiabilidad son esenciales y la anticipación al mercado es crítica. Aparte de estas cualidades empresariales fundamentales, SQL Server 2000 proporciona agilidad a sus operaciones de análisis y administración de datos al permitir a su organización adaptarse rápida y fácilmente para obtener ventaja competitiva en un entorno de cambios constantes. Desde una perspectiva de administración de datos y análisis, resulta crítico transformar los datos sin procesar en inteligencia empresarial y aprovechar las oportunidades que presenta el

Web. SQL Server 2000 es un paquete completo de base de datos y análisis de datos que abre las puertas al rápido desarrollo de una nueva generación de aplicaciones comerciales de nivel empresarial, que pueden proporcionar a su compañía una ventaja competitiva crítica. SQL Server 2000 ha obtenido importantes galardones en pruebas de referencia por su escalabilidad y velocidad. Es un producto de base de datos totalmente habilitado para Web que proporciona una compatibilidad fundamental con el Lenguaje de marcado extensible (XML, *Extensible Markup Language*) y la capacidad para realizar consultas en Internet y por encima del servidor de seguridad.

1.5 Conclusiones

Por las características ya descrita de las técnicas, tecnologías, metodologías y software en este capítulo, son la que se van a usar para crear el sistema de evaluación de desempeño de trabajadores de la UCI, debido a que gracias al uso de estas se ganara mucho en eficiencia, rapidez, facilidad y claridad del sistema.

Capítulo 2. Características del sistema.

2.1. Introducción

En este capítulo se expone el estudio realizado acerca del sistema de evaluación del desempeño de los trabajadores para la Universidad de Ciencias Informáticas. Se detalla la forma en la que se realizan las labores que serán objeto de automatización para determinar con mayor claridad las tareas que debe realizar el sistema. Se describen los procesos y la forma en que los actores interactúan con el sistema además del modelo de negocio.

2.2. Objeto de estudio.

2.2.1. Definición del problema.

En la oficina de recursos humanos conforma la evolución de los trabajadores de forma manual por lo que se hace muy trabajoso este proceso. Las evaluaciones están en archivos manuales y generar cualquier tipo de información se hace difícil pues el proceso es manual e implica horas de trabajos por lo que no satisface las necesidades existentes en estos momentos y por tanto surge la necesidad de informatizar esta área de recursos humanos.

2.2.2. Objetivos estratégicos.

Objetivos estratégicos generales

- Diseño, Implementación, e implantación de un sistema de evaluación del desempeño de trabajadores en la Universidad de Ciencias Informáticas (UCI).
- Brindar seguridad en los datos.
- Garantizar la flexibilidad del sistema.
- Mantener informada a la administración del funcionamiento del sistema.

Objetivos estratégicos específicos

- Analizar aspectos teóricos conceptuales de los Sistemas de evaluación del desempeño.
- Estudiar acerca de los Sistemas de evaluación del desempeño existentes.
- Hacer un estudio de cómo accederán los usuarios al sistema.
- Capacitar a los usuarios en el uso del sistema.
- Implementar el sistema de evaluación del desempeño que cumpla los requerimientos de la oficina de recursos humanos de la UCI.
- Montar el sistema en la red.

2.2.3. Análisis de como se ejecutan los procesos actualmente.

En las oficinas de recursos humanos actualmente se realizan las evaluaciones de los trabajadores manualmente, el jefe inmediato del trabajador realiza la evaluaciones de sus subordinados en cuanto al desempeño de este apoyándose en los parámetros ya establecido para esta tarea ya sea evaluación de adiestrados, técnicos, profesores o cuadros la cual son firmada por el jefe inmediato y el evaluado, posteriormente estas evaluaciones son llevadas a la especialista de recursos humanos que atiende a un determinado cargo, quien revisa el resultado final de la evolución si el resultado de la evaluación es inferior a lo estipulado, deben emprenderse acciones correctivas; si el desempeño es satisfactorio, debe ser alentado. En caso que falte alguna evaluación el especialista le informara al jefe inmediato del área donde falta esa evolución.

2.3. Descripción detallada de los procesos a automatizar.

Se desea automatizarse la evaluaciones del desempeño de los trabajadores de la institución, cada modelo de evaluación con sus características e información serán almacenados en una base de datos con una internas Web para mejorar el flujo de la información y consulta de esta, también la entrada de los usuarios al sistema, dándole la facilidad a los administradores de crear usuarios y darle los permisos pertinentes según el tipo de usuario. Y tendrá automatizado los reportes derivados de las evaluaciones.

2.4. Propuesta del sistema.

El sistema de evaluación del desempeño se considera que sea capaz de gestionar y guardar la información de las evaluaciones de los trabajadores desde cualquier momento y lugar conectado a la red con una interfaz única que permita acceder a cualquier tipo de evaluación.

Este sistema debe gestionar todo el proceso de guardar, modificar, eliminar y obtener información de las evaluaciones y asegurar la seguridad de este. Para lograr este propósito se creará un sistema capaz de permitir al administrador del sistema adicionar, eliminar y modificar a los usuarios dando los accesos pertinentes según el tipo de usuario, los especialistas tendrán acceso a visualizar la evaluación de los trabajadores tomando decisiones en base de la información y los jefes inmediatos tendrán acceso de insertar, eliminar, modificar y visualizar la información de los trabajadores subordinados a él. Las evaluaciones tanto de técnicos, profesores, adiestrados y cuadros cuentan con atributos comunes como:

- Periodo de evaluación (Fecha inicio de la evaluación y fecha fin).
- IdPersona al que pertenece la evaluación.

Se propone para este sistema que sea programado sobre tecnología Web, para un mejor uso en la red y las posibilidades que esta nos trae, se utilizará .NET por las características de este lenguaje de ser muy rápido en sus respuestas y compatible con el sistema operativo usado en las oficinas de recursos humanos, como SGBD se propone SQLSERVER para mantener la infraestructura de la UCI y por ser este bastante potente y resistir muchas conexiones en paralelo, además de ser muy fácil y compatible con casi todos los SGBD existentes.

2.5. Modelo del negocio

Ver Anexo 1.

2.6. Especificación de los requisitos de software.

El sistema se debe nutrir de otros sistemas que brindarán la información necesaria para el funcionamiento correcto.

El sistema de evaluación del desempeño deberá proveer la información sobre las evaluaciones de los trabajadores y permitir a los jefes inmediatos actualizar las evaluaciones.

2.6.1. Requerimientos Funcionales.

R1- Autenticar usuario.

- Verificar usuario y contraseña.

R2- Adicionar usuario.

- Insertar un nuevo usuario.
- Otorgar el tipo de usuario.

R3- Actualizar usuario.

- Eliminar un usuario.
- Modificar un usuario.

R4- Adicionar evaluación profesor.

R5- Adicionar evaluación cuadro.

R6- Adicionar evaluación adiestrado.

R7- Adicionar evaluación técnico.

R8- Actualizar evaluación profesor.

R9- Actualizar evaluación cuadro.

R10- Actualizar evaluación adiestrado.

R11- Actualizar evaluación técnico.

R12- Mostrar los adiestrados no evaluados con un mes de antelación.

R13- Mostrar los técnicos no evaluados que hallan cumplido al menos el 70% del trabajo realizado.

R14- Mostrar evaluación profesor.

R15- Mostrar evaluación cuadros.

R16- Mostrar evaluación técnicos.

R17- Mostrar evaluación adiestrado.

R18- Imprimir.

2.6.2. Requerimientos no Funcionales.

1- **Interfaz externa:** La interfaz debe ser sencilla, intuitiva, amigable manteniendo un diseño uniforme. Es decir de fácil uso.

2- **Rendimiento:** Se necesita gran rapidez búsqueda información debido a medida que valla pasando el tiempo el cúmulo de información va aumentando.

3- **Soporte:** Mantenimiento (El sistema debe estar bien documentado para que el tiempo de mantenimiento sea mínimo en caso de necesitarse.).Extensibilidad (se debe garantizar la inserción de módulos nuevos, sin afectar lo realizado hasta el momento o el buen funcionamiento.)

4- **Portabilidad:** El sistema se pondrá en función sobre plataforma Windows, que es la que más usada hasta el momento en la universidad.

5- **Seguridad:** Confiabilidad (la información manejada por el sistema está protegida de acceso no autorizado.). Integridad (la información será protegida contra corrupción y estados inconsistentes.). Disponibilidad (Los usuarios autorizados tendrán acceso a la información.)

6- **Software:** Se utilizará HTML, .NET, SQL Server.

7- **Hardware:** Se utiliza una PC de escritorio.

2.7. Definición de los casos de uso

Ver Anexo 2.

Conclusiones

En este capítulo se realizó un estudio de la reglas del negocio y se completó su análisis y modelado. Se realizó una propuesta de sistema que satisfaga las especificaciones de automatización de los procesos del negocio actuales y se completó el levantamiento de los requerimientos esperados para el sistema.

Capítulo 3. Análisis y diseño del sistema.

3.1. Introducción

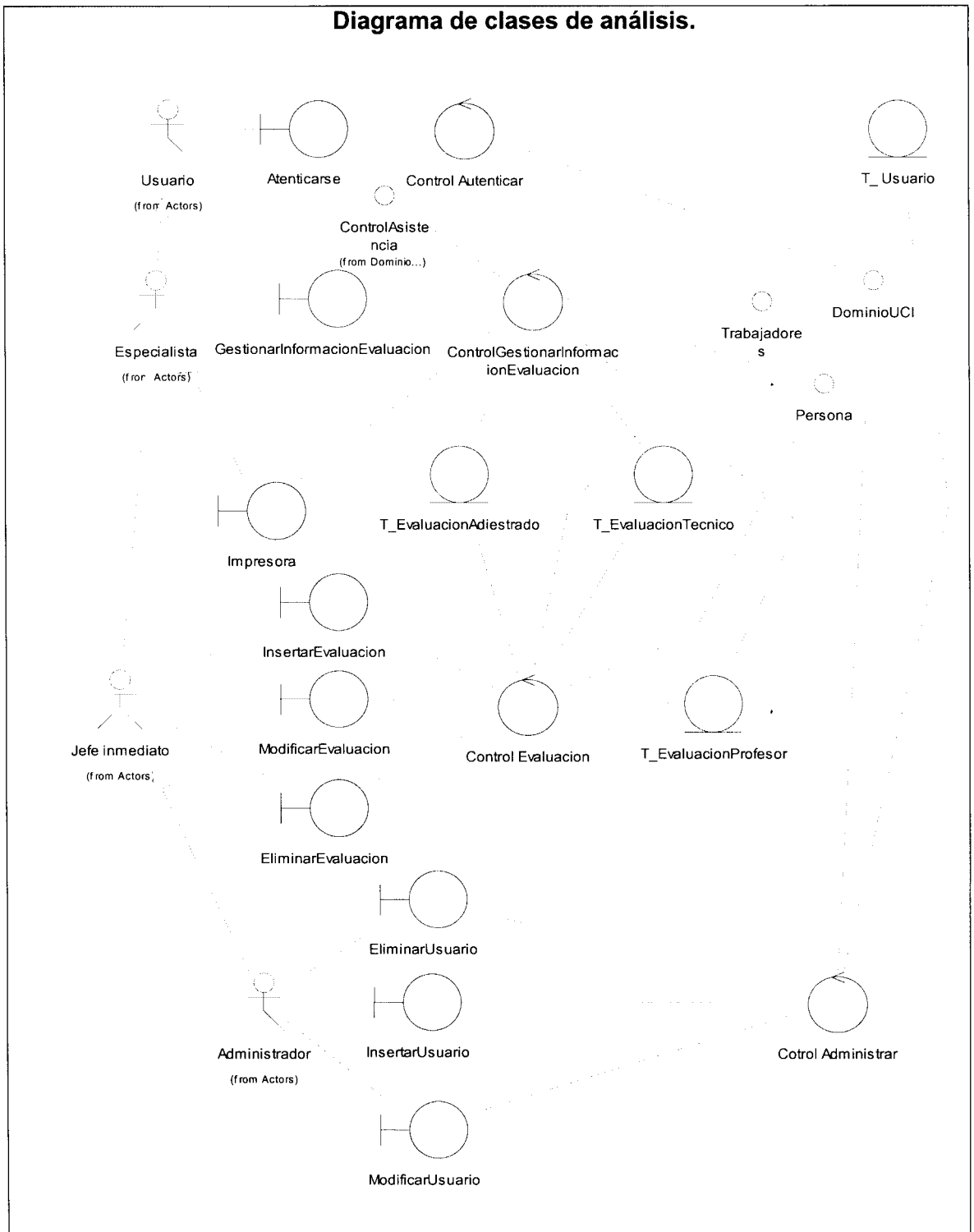
En esta etapa de desarrollo se profundiza en el análisis del sistema, luego se procede al diseño, donde quedará plasmado todo lo analizado anteriormente, en forma de diagramas que llevarán a cabo todo el flujo de los procesos. Se define el diagrama de clases del análisis de la aplicación, donde se especifica qué clases de análisis toman parte del caso de uso y sus relaciones entre ellas y las entidades. En de los diagramas de interacción del sistema se muestra una descripción gráfica de la interacción entre los actores y el sistema, además quedan reflejados los mensajes que se transmiten entre los objetos. También aquí se describen las clases que va a tener nuestra aplicación en el diseño y se obtiene el Diagrama Entidad-Relación y la descripción de sus tablas.

3.2. Análisis

En el diagrama de clases del análisis se representa una abstracción de una o varias clases y/o subsistemas en el diseño del sistema. Estas clases se clasifican en:

- De Interfaz: son usadas para modelar la interacción entre el sistema y sus actores.
- De Entidad: son usadas para modelar información que persiste en el tiempo o tiene una larga vida.
- De Control: estas clases realizan la coordinación, secuenciado de transacciones y, en definitiva, el control sobre otros objetos del sistema. [13]

3.2.1. Diagrama de clases de análisis



3.3. Diseño

3.3.1. Diagramas de interacción

Ver Anexo 3.

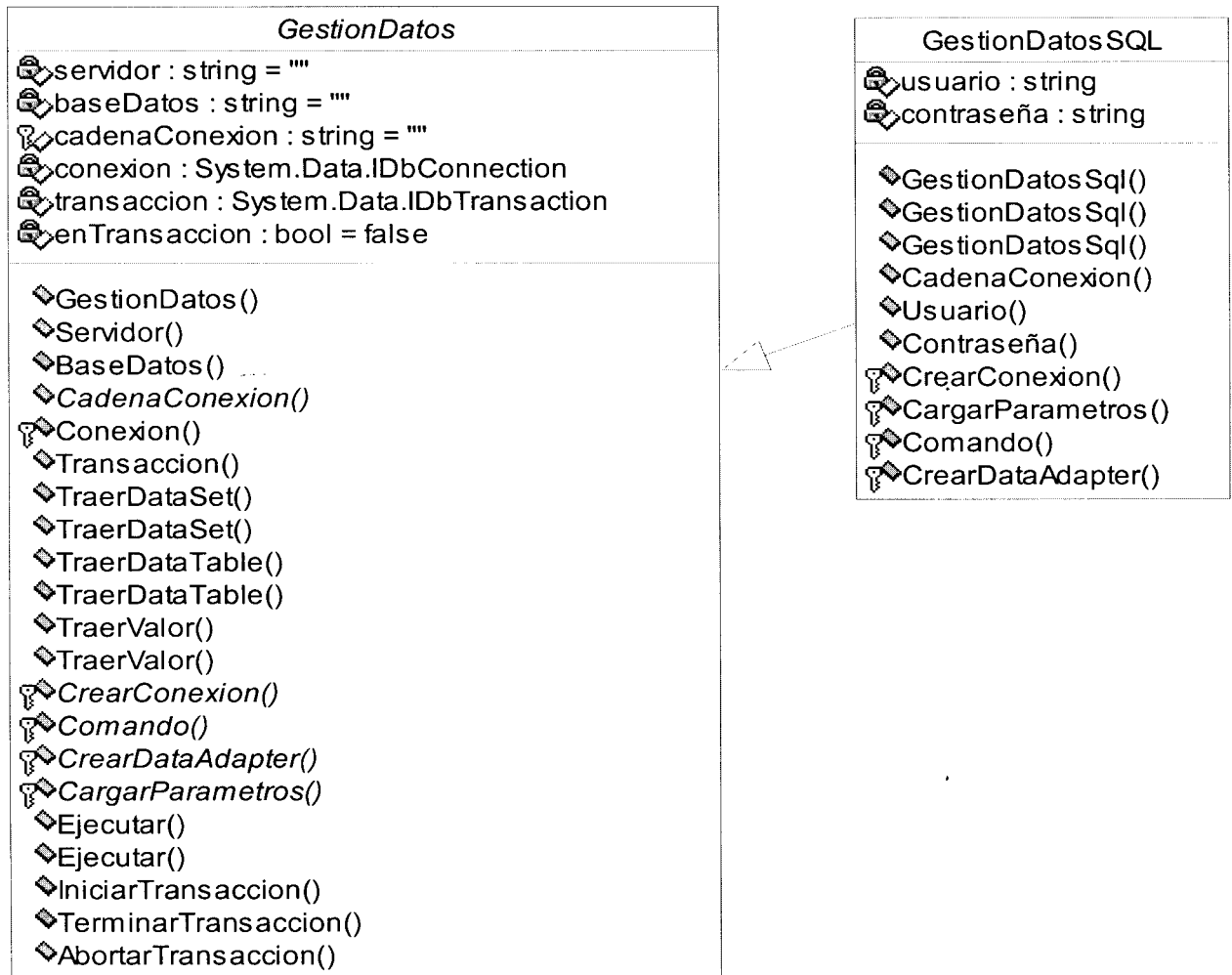
3.3.2. Diagrama de clases

Se ha diseñado una jerarquía de clases que será la encargada de gestionar toda la data de forma única teniendo en cuenta la posibilidad de emplear varios gestores de diferentes tecnologías por lo que no estarán ligadas a ninguna base de datos en específico. Estas clases se limitan a implementar métodos para realizar consultas, ya sea de acción o de selección pero en este caso solo manipularán procedimientos almacenados dada su probada eficiencia, no obstante no sería difícil realizar pequeñas modificaciones para emplear consultas SQL. Serán las encargadas además de manipular las conexiones y las transacciones.

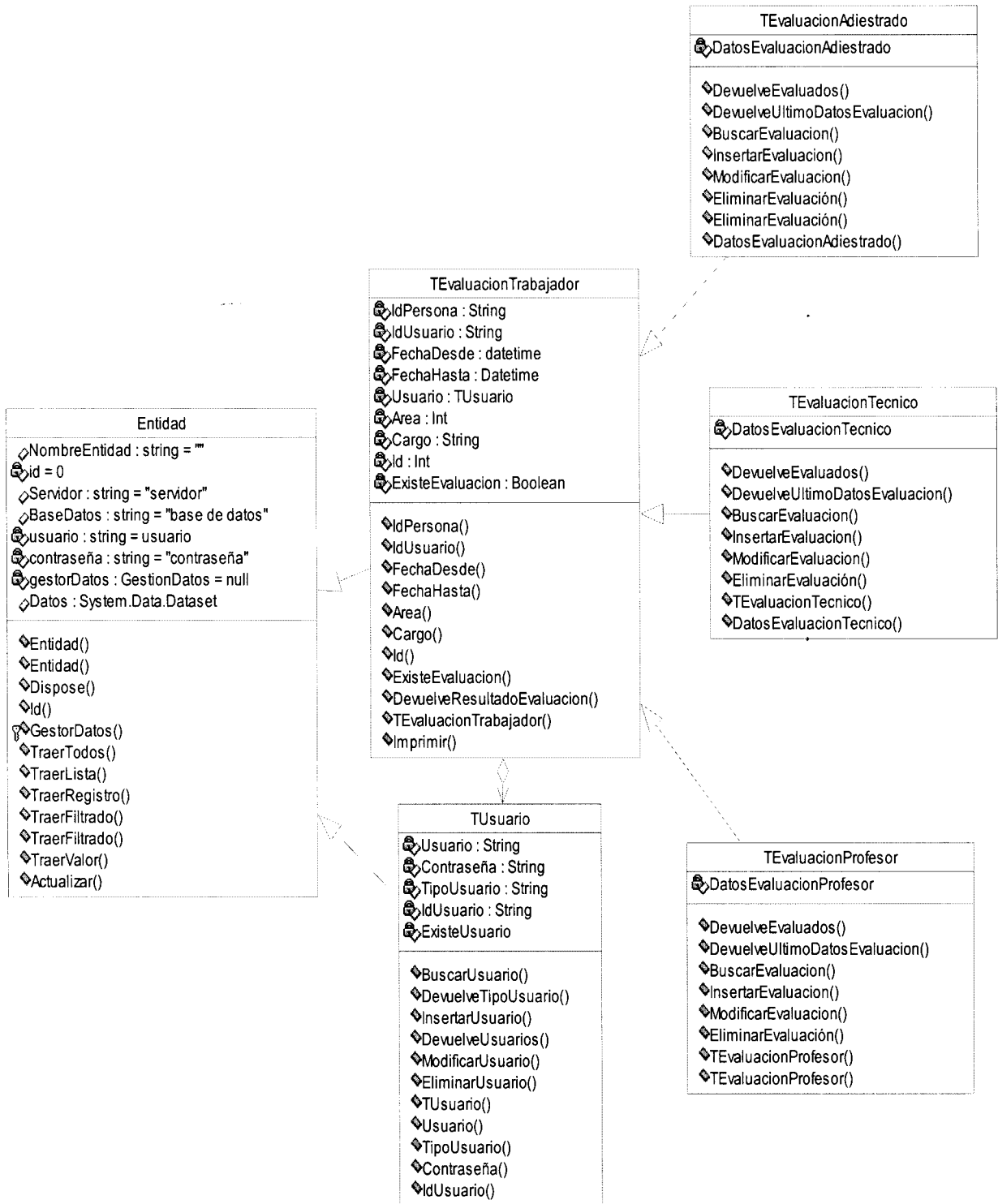
Dado que la aplicación que nos ocupa limita su funcionamiento prácticamente a manipular datos almacenados en una base de datos se creó una jerarquía que permita un comportamiento de los objetos como si fueran entidades lógicas con sus funciones de proveer datos y almacenarlos en función de las especificaciones de la entidad representada. Para este propósito se implementó una clase Entidad que se comportará como cualquier entidad de datos con sus posibles métodos de devolución, modificación e inserción de datos. Para que esta filosofía de diseño funcione correctamente cada método de la clase deberá tener implementado su contraparte en consultas SQL que en este caso estará contenido en procedimientos almacenados, siguiendo una nomenclatura específica, pero es posible con la realización de pequeños cambios implementar un mecanismo para emplear consultas SQL desde el lado de la aplicación, este tema de la implementación de las consultas se retomará con mas detalle en la sección del diseño del modelo de datos. Todas las clases que manejan la lógica del sistema heredarán de esta clase Entidad, implementando los mecanismos específicos para sus responsabilidades particulares.

Según esta filosofía de diseño la aplicación quedará perfectamente dividida en tres capas mínimas, capa de datos, capa de negocio y capa de presentación.

3.3.2.1. Capa de datos

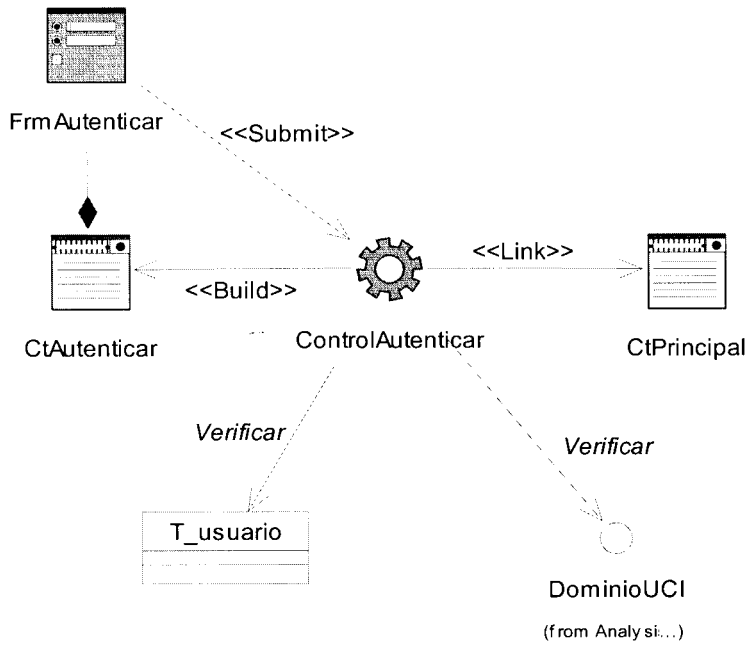


3.3.2.2. Capa de negocio

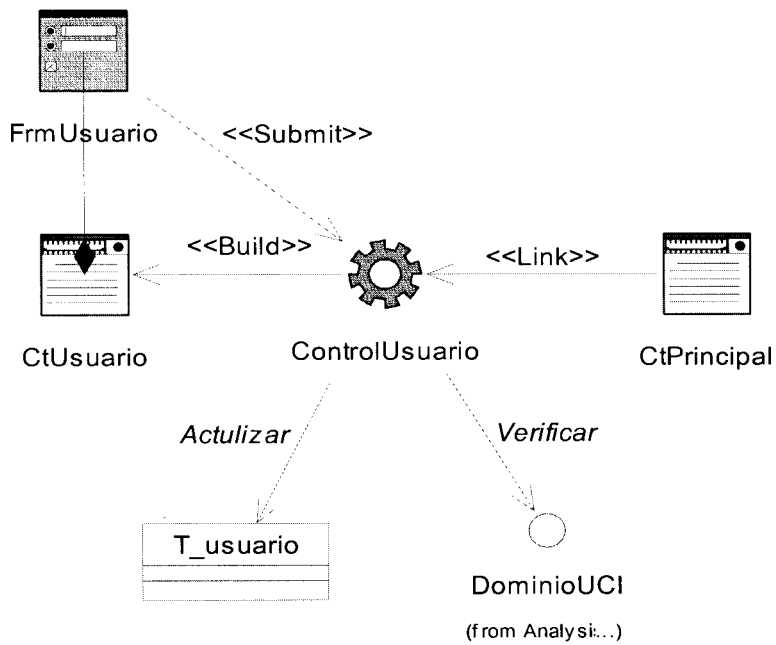


3.3.3. Diagrama de clases Web

Autenticar usuarios:

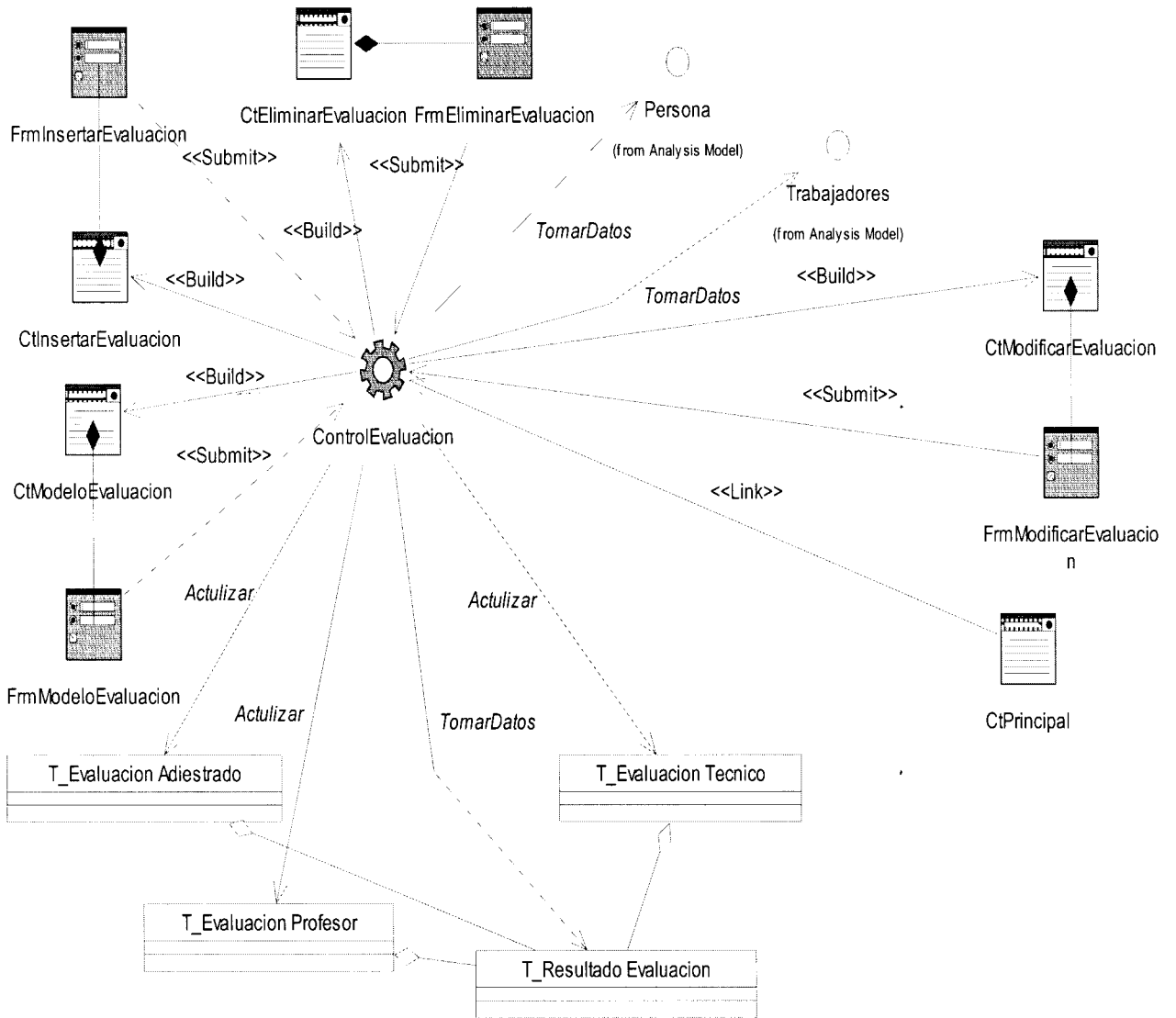


Administrar usuario:

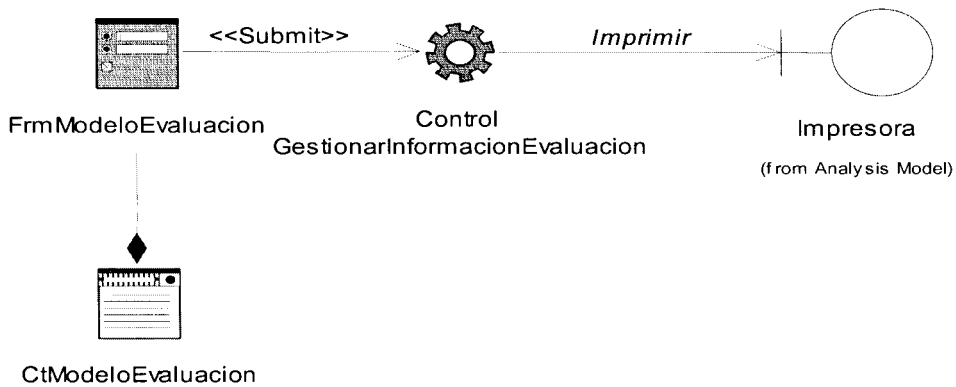


:

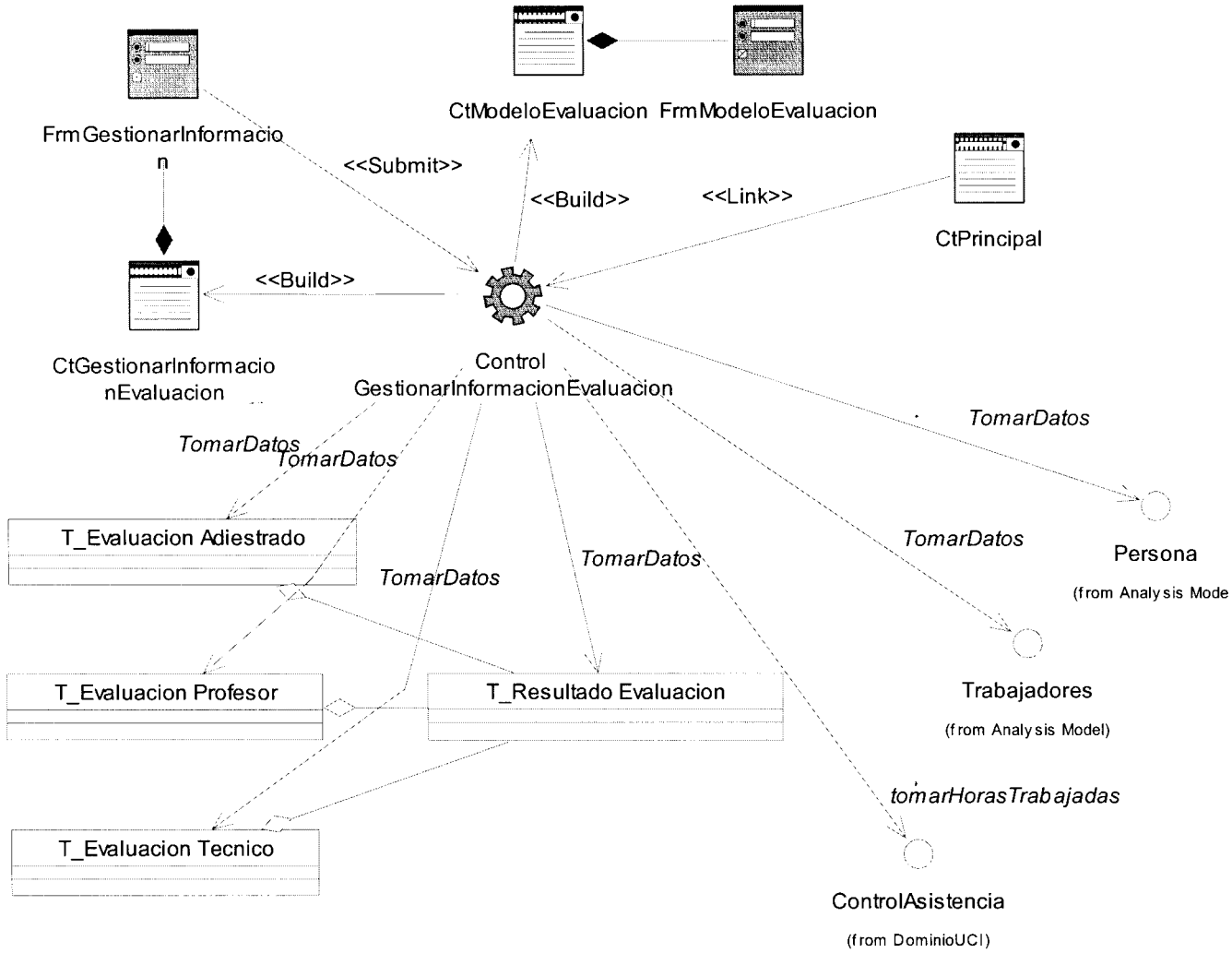
Evaluar trabajador:



Imprimir:



Gestión Información evaluación:



3.3.4. Descripción de las Clases

Nombre: GestionDatos (abstracta)	
Clase Entidad	
Atributo	Tipo
servidor	string
baseDatos	string
cadenaConexion	string
conexion	System.Data.IDbConnection
transaccion	System.Data.IDbTransaction
enTransaccion	bool
Para cada responsabilidad:	
Nombre:	GestionDatos()
Descripción:	Es el constructor por defecto de la clase.
Nombre:	string Servidor
Descripción:	Es la propiedad que encapsula al atributo servidor
Nombre:	string BaseDatos
Descripción:	Es la propiedad que encapsula al atributo baseDatos
Nombre:	string CadenaConexion
Descripción:	Es la propiedad abstracta que encapsula al atributo cadenaConexion
Nombre:	System.Data.IDbConnection Conexion
Descripción:	Es la propiedad que encapsula al atributo conexión, es de solo lectura y se encarga, en caso de ser accedida, de crear la conexión si no existe.
Nombre:	System.Data.IDbTransaction Transaccion
Descripción:	Es la propiedad que encapsula al atributo transaccion
Nombre:	System.Data.DataSet TraerDataSet(string procedimientoAlmacenado)
Descripción:	Método que devuelve un dataset dado un procedimiento almacenado existente con una consulta de selección.
Nombre:	System.Data.DataSet TraerDataSet(string procedimientoAlmacenado, params object[] argumentos)
Descripción:	Sobrecarga del método anterior que permite emplear procedimientos almacenados con parámetros.
Nombre:	System.Data.DataTable TraerDataTable(string

	procedimientoAlmacenado)
Descripción:	Método que devuelve un datatable dado un procedimiento almacenado existente con una consulta de selección.
Nombre:	System.Data.DataTable TraerDataTable(string procedimientoAlmacenado, params object[] argumentos)
Descripción:	Sobrecarga del método anterior que permite emplear procedimientos almacenados con parámetros.
Nombre:	object TraerValor(string procedimientoAlmacenado)
Descripción:	Método que devuelve un valor dado un procedimiento almacenado existente con un parámetro de salida.
Nombre:	object TraerValor(string procedimientoAlmacenado, params object[] argumentos)
Descripción:	Sobrecarga del método anterior que permite emplear procedimientos almacenados con parámetros.
Nombre:	System.Data.IDbConnection CrearConexion(string cadenaConexion)
Descripción:	Método abstracto que debe encargarse de crear la conexión para el gestor de datos específico.
Nombre:	System.Data.IDbCommand Comando(string procedimientoAlmacenado)
Descripción:	Método abstracto que debe encargarse de crear el comando para el gestor de datos específico.
Nombre:	System.Data.IDataAdapter CrearDataAdapter(string procedimientoAlmacenado, params object[] argumentos)
Descripción:	Método abstracto que debe encargarse de crear el dataadapter para el gestor de datos específico.
Nombre:	CargarParametros(IDbCommand Comando, params object[] argumentos)
Descripción:	Método abstracto que debe encargarse de la carga de los parámetros en el comando para el gestor de datos específico.
Nombre:	int Ejecutar(string procedimientoAlmacenado)
Descripción:	Método encargado de ejecutar un procedimiento almacenado que contenga una consulta de acción, devuelve el número de registros afectados.

Nombre:	int Ejecutar(string procedimientoAlmacenado, params object[] argumentos)
Descripción:	Sobrecarga del método anterior que permite emplear un procedimiento almacenado con parámetros.
Nombre:	IniciarTransaccion()
Descripción:	Método encargado de crear e iniciar una transacción.
Nombre:	TerminarTransaccion()
Descripción:	Método encargado de completar la transacción iniciada.
Nombre:	AbortarTransaccion()
Descripción:	Método encargado de cancelar una transacción iniciada.

Nombre: GestionDatosSQL (hereda de GestionDatos)	
Clase Entidad	
Atributo	Tipo
usuario	string
contraseña	string
Para cada responsabilidad:	
Nombre:	GestionDatosSql()
Descripción:	Es el constructor por defecto de la clase.
Nombre:	GestionDatosSql(string cadenaConexion)
Descripción:	Sobrecarga del constructor que inicializa el atributo cadenaConexion heredado.
Nombre:	GestionDatosSql(string servidor, string baseDatos, string usuario, string contraseña)
Descripción:	Sobrecarga del constructor que inicializa los atributos usuario y contraseña de la clase y servidor y baseDatos heredados.
Nombre:	string CadenaConexion
Descripción:	Implementación de la propiedad abstracta heredada del mismo nombre.
Nombre:	string Usuario
Descripción:	Es la propiedad de solo escritura que encapsula el atributo usuario.
Nombre:	string Contraseña

Descripción:	Es la propiedad de solo escritura que encapsula el atributo contraseña.
Nombre:	System.Data.IDbConnection CrearConexion(string cadenaConexion)
Descripción:	Implementación de la propiedad abstracta heredada del mismo nombre.
Nombre:	CargarParametros(IDbCommand comando, params object[] argumentos)
Descripción:	Implementación de la propiedad abstracta heredada del mismo nombre.
Nombre:	System.Data.IDbCommand Comando(string procedimientoAlmacenado)
Descripción:	Implementación de la propiedad abstracta heredada del mismo nombre.
Nombre:	System.Data.IDataAdapter CrearDataAdapter(string procedimientoAlmacenado, params object[] argumentos)
Descripción:	Implementación de la propiedad abstracta heredada del mismo nombre.
Nombre:	object TraerValor(string procedimientoAlmacenado)
Descripción:	Método que devuelve un valor dado un procedimiento almacenado existente con un parámetro de salida.
Nombre:	object TraerValor(string procedimientoAlmacenado, params object[] argumentos)
Descripción:	Sobrecarga del método anterior que permite emplear procedimientos almacenados con parámetros.
Nombre:	System.Data.IDbConnection CrearConexion(string cadenaConexion)
Descripción:	Método abstracto que debe encargarse de crear la conexión para el gestor de datos específico.
Nombre:	System.Data.IDbCommand Comando(string procedimientoAlmacenado)
Descripción:	Método abstracto que debe encargarse de crear el comando para el gestor de datos específico.
Nombre:	System.Data.IDataAdapter CrearDataAdapter(string procedimientoAlmacenado, params object[] argumentos)
Descripción:	Método abstracto que debe encargarse de crear el

	dataadapter para el gestor de datos específico.
--	---

Nombre: Entidad	
Clase controladora	
Atributo	Tipo
NombreEntidad	string
id	int
Servidor	string
BaseDatos	string
usuario	string
contraseña	string
gestorDatos	GestionDatos
Datos	System.Data.DataSet
Para cada responsabilidad:	
Nombre:	Entidad()
Descripción:	Es el constructor por defecto de la clase.
Nombre:	Entidad(string nombreEntidad)
Descripción:	Sobrecarga del constructor que inicializa el atributo nombreEntidad.
Nombre:	int Id
Descripción:	Propiedad que encapsula el atributo id, a asignar un identificador válido se cargan los datos referentes a la entidad creada correspondientes al id en la tabla correspondiente.
Nombre:	GestionDatos GestorDatos()
Descripción:	Crea una nueva instancia de la clase GestorDatos, en caso de no existir, según el gestor de datos correspondiente.
Nombre:	System.Data.DataTable TraerTodos()
Descripción:	Método que devuelve todos los registros de la entidad especificada correspondiente a la tabla del mismo nombre.
Nombre:	System.Data.DataTable TraerLista ()
Descripción:	Método que devuelve todos los registros de la entidad especificada correspondiente a la tabla del mismo nombre, tomando solamente los dos campos mas significativos en la forma Id y Descripción.

Nombre:	System.Data.DataRow TraerRegistro(int id)
Descripción:	Retorna el registro especificado por id.
Nombre:	System.Data.DataTable TraerFiltrado(string filtro)
Descripción:	Retorna un conjunto de registros seleccionados según el criterio especificado en filtro.
Nombre:	System.Data.DataTable TraerFiltrado(string filtro, params object[] argumentos)
Descripción:	Sobrecarga del método anterior que permite el paso de argumentos aparte del criterio de selección.
Nombre:	object TraerValor(string condicion, params object[] argumentos)
Descripción:	Devuelve un valor que cumpla con la condición especificada en condición.
Nombre:	Actualizar()
Descripción:	Actualiza los cambios realizados al conjunto de datos en la base de datos.

Nombre: TEvaluacionTrabajador (hereda de Entidad)	
Tipo de clase: Entidad	
Atributo	Tipo
IdUsuario	String
IdPersona	String
FechaDesde	Datetime
FechaHasta	Datetime
Area	Int
Cargo	String
Id	Int
ExisteEvaluacion	Boolean
Usuario	TUsuario
Para cada responsabilidad:	
Nombre:	String IdUsuario
Descripción:	Propiedad que encapsula el atributo IdUsuario.
Nombre:	String IdPersona
Descripción:	Propiedad que encapsula el atributo IdPersona.
Nombre:	Datetime FechaDesde
Descripción:	Propiedad que encapsula el atributo FechaDesde.

Nombre:	Datetime FechaHasta
Descripción:	Propiedad que encapsula el atributo FechaHasta.
Nombre:	Int Area
Descripción:	Propiedad que encapsula el atributo Area.
Nombre:	Int Id
Descripción:	Propiedad que encapsula el atributo Id.
Nombre:	String Cargo
Descripción:	Propiedad que encapsula el atributo Cargo.
Nombre:	Boolean ExisteEvaluacion
Descripción:	Propiedad que encapsula el atributo ExisteEvaluacion.
Nombre:	TEvaluacionTrabajador()
Descripción:	Es el constructor por defecto de la clase.
Nombre:	DevuelveResultadoEvaluacion(Id, var Resultado)
Descripción:	Dado el id devuelve el resultado de la evaluación.
Nombre:	Imprimir()
Descripción:	Da la orden de impresión a la impresora.

Nombre: TEvaluacionAdiestrado (hereda de TEvaluacionTrabajador)	
Tipo de clase: Entidad	
Atributo	Tipo
DatosEvaluacionAdiestrado	System.Data.DataTable
Para cada responsabilidad:	
Nombre:	DevuelveEvaluados (IdPersona, FechaDesde, var DatosEvaluacionAdiestrado)
Descripción:	Dado IdPersona y FechaDesde devuelve los datos de la evaluación de adiestrados.
Nombre:	DevuelveUltimoDatosEvaluacion (IdPersona, var DatosEvaluacionAdiestrado, var FechaDesde, var FechaHasta)
Descripción:	Dado el IdPersona devuelve el periodo de la última evaluación registrada y los datos de la evaluación de adiestrado.
Nombre:	BuscarEvaluacion (IdPersona, FechaDesde, FechaHasta): Boolean
Descripción:	Dado el IdPersona y el periodo de evaluación devuelve si esta evaluación existe en la base de datos.

Nombre:	InsertarEvaluacion (IdPersona, FechaDesde, FechaHasta, DatosEvaluacionAdiestrado)
Descripción:	Dado el IdPersona y el periodo de evaluación inserta los datos de la evaluación del adiestrado
Nombre:	ModificarEvaluacion (IdPersona, FechaDesde, FechaHasta, DatosEvaluacionAdiestrado)
Descripción:	Dado el IdPersona y el periodo evaluación modifica los datos de la evaluación del adiestrado
Nombre:	EliminarEvaluación (IdPersona, FechaDesde, FechaHasta)
Descripción:	Dado el IdPersona y el periodo evaluación elimina los datos de la evaluación adiestrado.
Nombre:	TEvaluacionAdiestrado()
Descripción:	Es el constructor por defecto de la clase.
Nombre:	System.Data.DataTable DatosEvaluacionAdiestrado
Descripción:	Propiedad que encapsula el atributo DatosEvaluacionAdiestrado

Nombre: TEvaluacionTecnico (hereda de TEvaluacionTrabajador)	
Tipo de clase: Entidad	
Atributo	Tipo
DatosEvaluacionTecnico	System.Data.DataTable
Para cada responsabilidad:	
Nombre:	DevuelveEvaluados (IdPersona, FechaDesde, var DatosEvaluacionTecnico)
Descripción:	Dado IdPersona y FechaDesde devuelve los datos de la evaluación de técnico.
Nombre:	DevuelveUltimoDatosEvaluacion (IdPersona, var DatosEvaluacionTecnico, var FechaDesde, var FechaHasta)
Descripción:	Dado el IdPersona devuelve el periodo de la última evaluación registrada y los datos de la evaluación de técnico.
Nombre:	BuscarEvaluacion (IdPersona, FechaDesde, FechaHasta): Boolean
Descripción:	Dado el IdPersona y el periodo de evaluación devuelve si esta evaluación existe en la base de datos.
Nombre:	InsertarEvaluacion (IdPersona, FechaDesde, FechaHasta, DatosEvaluacionTecnico)

Descripción:	Dado el IdPersona y el periodo de evaluación inserta los datos de la evaluación del técnico.
Nombre:	ModificarEvaluacion (IdPersona, FechaDesde, FechaHasta, DatosEvaluacionTecnico)
Descripción:	Dado el IdPersona y el periodo evaluación modifica los datos de la evaluación del técnico.
Nombre:	EliminarEvaluación (IdPersona, FechaDesde)
Descripción:	Dado el IdPersona y el periodo evaluación elimina los datos de la evaluación técnico.
Nombre:	TEvaluacionTecnico()
Descripción:	Es el constructor por defecto de la clase.
Nombre:	System.Data.DataTable DatosEvaluacionTecnico
Descripción:	Propiedad que encapsula el atributo DatosEvaluacionTecnico

Nombre: TEvaluacionProfesor (hereda de TEvaluacionTrabajador)	
Tipo de clase: Entidad	
Atributo	Tipo
DatosEvaluacionProfesor	System.Data.DataTable
Para cada responsabilidad:	
Nombre:	DevuelveEvaluados (IdPersona, FechaDesde, var DatosEvaluacionProfesor)
Descripción:	Dado IdPersona y FechaDesde devuelve los datos de la evaluación de Profesor.
Nombre:	DevuelveUltimoDatosEvaluacion (IdPersona, var DatosEvaluacionProfesor, var FechaDesde, var FechaHasta)
Descripción:	Dado el IdPersona devuelve el periodo de la última evaluación registrada y los datos de la evaluación de profesor.
Nombre:	BuscarEvaluacion (IdPersona, FechaDesde, FechaHasta): Boolean
Descripción:	Dado el IdPersona y el periodo de evaluación devuelve si esta evaluación existe en la base de datos.
Nombre:	InsertarEvaluacion (IdPersona, FechaDesde, FechaHasta, DatosEvaluacionProfesor)
Descripción:	Dado el IdPersona y el periodo de evaluación inserta los datos de la evaluación del Profesor

Nombre:	ModificarEvaluacion (IdPersona, FechaDesde, FechaHasta, DatosEvaluacionProfesor)
Descripción:	Dado el IdPersona y el periodo evaluación modifica los datos de la evaluación del Profesor
Nombre:	EliminarEvaluación (IdPersona, FechaDesde)
Descripción:	Dado el IdPersona y el periodo evaluación elimina los datos de la evaluación Profesor.
Nombre:	TEvaluacionProfesor()
Descripción:	Es el constructor por defecto de la clase.
Nombre:	System.Data.DataTable DatosEvaluacionProfesor.
Descripción:	Propiedad que encapsula el atributo DatosEvaluacionProfesor

Nombre: TUsuario (hereda de Entidad)	
Tipo de clase: Entidad	
Atributo	Tipo
Usuario	String
Contraseña	String
TipoUsuario	String
IdUsuario	String
ExisteUsuario	Boolean
Para cada responsabilidad:	
Nombre:	BuscarUsuario (Usuario): Boolean
Descripción:	Dado el Usuario dice si existe en la base de datos
Nombre:	DevuelveTipoUsuario (Usuario, var TipoUsuario)
Descripción:	Dado el Usuario devuelve el tipo de usuario.
Nombre:	InsertarUsuario (Usuario, TipoUsuario)
Descripción:	.Inserta Usuario y tipo de usuario en la base de datos.
Nombre:	DevuelveUsuarios (Usuario, Tipo)
Descripción:	Devuelve todos los usuarios y su tipo de usuario de la base de datos.
Nombre:	ModificarUsuario (Usuario, TipoUsuario)
Descripción:	Modifica el usuario y el tipo de usuario de la base de datos.
Nombre:	EliminarUsuario (Usuario)
Descripción:	Dado el Usuario lo elimina de la base de datos.

Nombre:	TUsuario()
Descripción:	Es el constructor por defecto de la clase.
Nombre:	Srting Usuario
Descripción:	Propiedad que encapsula el atributo Usuario.
Nombre:	String TipoUsuario
Descripción:	Propiedad que encapsula el atributo TipoUsuario.
Nombre:	String Contraseña
Descripción:	Propiedad que encapsula el atributo Contraseña.
Nombre:	String IdUsuario
Descripción:	Propiedad que encapsula el atributo IdUsuario

Se ha tomado la determinación de no describir las clases que físicamente tienen su equivalente como páginas Web, ya sean de cliente o servidor, siguiendo la misma metodología. Las clases (páginas) empleadas en el diseño no contienen lógica del negocio en su contenido dada su separación en la capa de presentación o de cliente, por este motivo solo se limitan a contener manipuladores para los eventos correspondientes y como atributos emplean las instancias de los componentes de Web Forms de .Net. La descripción se basará en los roles que juegan estas páginas dentro de la aplicación.

Nombre: ControlAutenticar
Es la clase encargada de permitir a los usuarios registrarse al sistema, generando un formulario para que el usuario pueda introducir su clave y contraseña y el sistema tenga confirmación de las páginas restringidas.

Nombre: ControlUsuario
Es la clase encargada de permitir al administrador insertar, modificar y eliminar los usuarios del sistema. Permitirle darle los accesos a los usuarios. Genera un formulario para permitir estas operaciones y mostrando los usuarios existentes en el sistema.

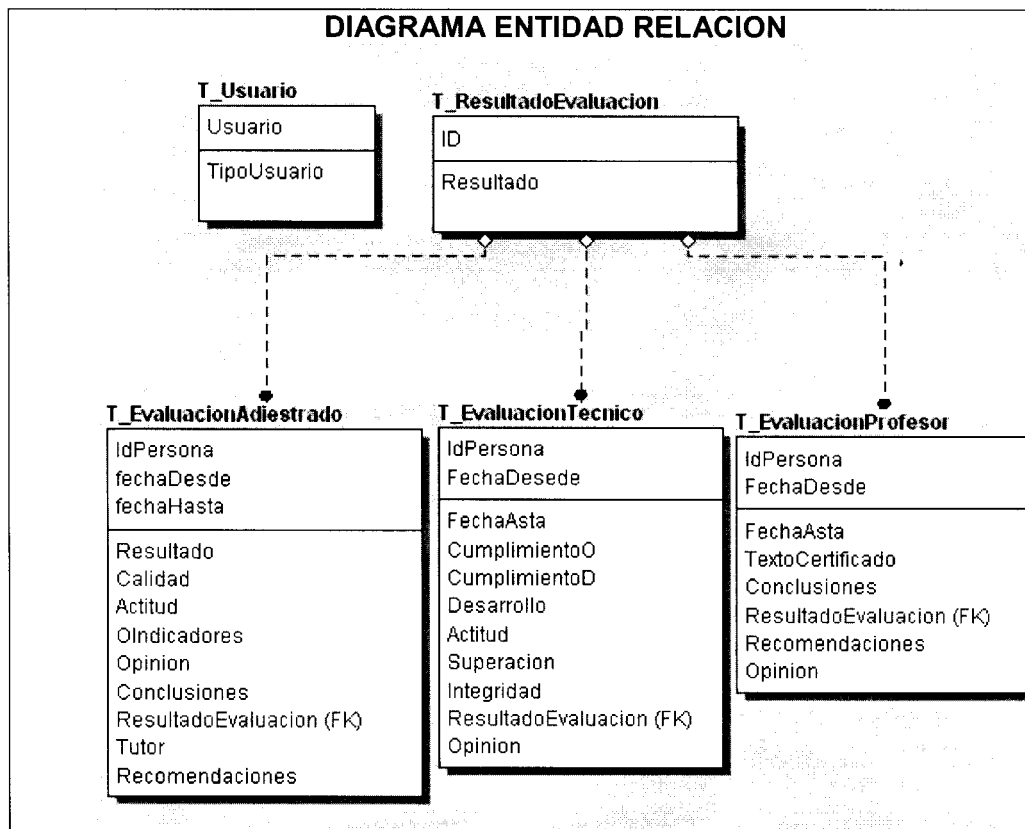
Nombre: ControlEvaluacion

Es la clase encargada de permitir al usuario buscar al trabajador que desea insertarle, eliminarle o modificarle la evaluación. Genera un formulario para permitir introducir el trabajador que se desea actualizarle su evaluación y le muestra los trabajadores a actualizar.

Nombre: ControlGestionInformacionInformacionEvaluacion

Es la clase encargada de permitirle al usuario buscar al o los trabadores que se le desea visualizar la información respecto a la evaluación. Genera un formulario que le permite al usuario introducir los datos para realizar la búsqueda deseada visualizando dicha información. Y le permite al usuario imprimir la información visualizada.

3.3.5. Diagrama entidad-relación.



Nombre: T_EvaluacionProfesor		
Descripción: Contiene los datos de las de la evaluación de los profesores.		
Atributo	Tipo	Descripción
IdPersona	nvarchar	Identificador de la persona ha la que se evalúa.
FechaDesde	datetime	Fecha de inicio del periodo de evaluación.
FechaHasta	datetime	Fecha fin del periodo de evaluación.
TextoCertificado	text	Texto donde se evalúa al profesor.
Conclusiones	nchar	Las conclusiones de la evaluación.
ResultadoEvaluacion	bigint	Identificador de el resultado de la evaluación.
Recomendaciones	nchar	Recomendaciones derivadas de la evaluación.
Opinion	nchar	Opinión del evaluado.

Nombre: T_EvaluacionTecnico		
Descripción: Contiene los datos de las de la evaluación de los técnicos.		
Atributo	Tipo	Descripción
IdPersona	nvarchar	Identificador de la persona ha la que se evalúa.
FechaDesde	datetime	Fecha de inicio del periodo de evaluación.
FechaHasta	datetime	Fecha fin del periodo de evaluación.
CumplimientoO	nchar	Cumplimiento y calidad de los objetivos y tareas asignadas.
CumplimientoD	nchar	Cumplimiento de la disciplina laboral y tecnológica.
Desarrollo	nchar	Desarrollo demostrado e iniciativas en el cumplimiento de las tareas.
Actitud	nchar	Actitud ante el trabajo.
Superacion	nchar	Superación y desarrollo técnico profesional.
Integridad	nchar	Integridad y contribución al trabajo colectivo de su unidad organizativa y cumplimiento de indicadores específicos de su actividad.
ResultadoEvaluacion	bigint	Identificador de el resultado de la evaluación.
Opinion	nchar	Opinión del evaluado.

Nombre: T_EvaluacionAdiestrado		
Descripción: Contiene los datos de las de la evaluación de los técnicos.		
Atributo	Tipo	Descripción
IdPersona	nvarchar	Identificador de la persona ha la que se evalúa.
FechaDesde	datetime	Fecha de inicio del periodo de evaluación.
FechaHasta	datetime	Fecha fin del periodo de evaluación.
Resultado	nchar	Resultado del cumplimiento de los objetivos y tareas previstas (a partir del plan que se diseño para la etapa que se evalúa).
Calidad	nchar	Calidad del trabajo realizado.
Actitud	nchar	Actitud ante el trabajo, disciplina laboral y social.
OIndicadores	nchar	Otros indicadores (Incluye actividades de superación, preparación como reserva de cuadros y otras actividades extraplan).
Opinion	nchar	Opinión del evaluado.
Conclusiones	nchar	Resumen de la evaluación.
ResultadoEvaluacion	bigint	Identificador de el resultado de la evaluación.
Recomendaciones	nchar	Recomendaciones derivadas de la evaluación.
Tutor	nchar	Nombre del tutor.

Nombre: T_ResultadoEvaluacion		
Descripción: Contiene los datos de el resultado de las evaluaciones.		
Atributo	Tipo	Descripción
Id	bigint	Identificador de el resultado de la evaluación.
Resultado	char	Contiene el resultado de la evaluación.

Nombre: T_Usuario		
Descripción: Contiene los datos de el resultado de las evaluaciones.		
Atributo	Tipo	Descripción
Usuario	nvarchar	Contiene el usuario del dominio UCI.
TipoUsuario	nchar	Contiene el tipo de usuario del sistema.

3.3.6. Tratamiento de errores

El tratamiento de los errores es fundamental para que el sistema funcione bien, garantizando la eficiencia y claridad de la aplicación. El tratamiento de errores está dirigido a los errores que pueden ocurrir durante la interacción del usuario con el sistema a la hora de introducir datos, como a los errores y excepciones que pudieran generarse a la hora de correr la aplicación debido a su funcionamiento e interacción con sistemas y mecanismos externos.

En este caso, fue elegida una forma de tratar estos errores de manera que las operaciones de inserción, eliminación y/o modificación de registros en la Base de Datos se realice de forma correcta, fueron establecidos mecanismos que visualicen la información para evitar en la medida de lo posible la introducción manual de los datos y minimizar los errores que pueda tener la aplicación, siendo válida toda la información.

En el caso de la inserción se implementarán funciones que validen la entrada de datos y en caso de haber errores, se mostrarán mensajes que hagan una aclaración ejemplificada de cómo se desea que realmente se presente la información.

Otros errores pueden ser generados por el Gestor de Base de Datos, estos se capturan antes de mostrarse al usuario y una vez tratados por el sistema, son mostrados al usuario de una manera entendible para él.

3.3.7. Diseño de la Interfaz.

El diseño de la interfaz es un punto fundamental a la hora de la presentación de la aplicación ya que es la capa de presentación al usuario y por lo tanto debe ser lo más amigable y entendible posible. En el diseño de de la interfaz hay que tener en cuenta la organización de los elementos en la pantalla, donde se coloca la información a mostrar y cómo se estructuran. La paginas de la interfaz se diseñaran de forma uniforme garantizando el equilibrio en la organización de la información, La optimización de la cantidad de elementos en la pantalla asiendo mas fácil manejar la información mostrada en la pantalla y la unidad. No se usaran colores fuertes ni brillantes para la comodidad y visibilidad de los usuarios que interactúen con el sistema. Las páginas no se deben

encontrar muy cargadas, solo la información necesaria para mayor claridad. El vocabulario a utilizar no habrá problemas ya que resulta muy familiar a los usuarios que harán uso de ella.

3.3.8. Concepción general de la ayuda.

El sistema mostrará una interfaz Web amigable y suficientemente clara para el usuario, por lo que la utilización de una ayuda no es considerada necesario.

3.4. Conclusiones

En este capítulo se abordó el análisis y el diseño del sistema a implementar, desarrollando el modelo de clases del análisis y se elaboraron los diagramas de interacción como evolución de la expansión de los casos de uso. Se obtuvo el diagrama de clases del diseño de la aplicación, se describieron las clases teniendo en cuenta sus atributos y responsabilidades, se desarrollaron los diagramas de diseño Web y se realizó la descripción de las páginas en función de sus roles específicos. Se obtuvo el diagrama entidad relación y se realizó la descripción de las tablas teniendo en cuenta la información que contiene y los campos que contiene. En este capítulo se abordó acerca del tratamiento de errores y diseño de Interfaz.

Conclusiones generales

Durante el desarrollo de este trabajo se fueron creando las condiciones para la implementación del sistema de evaluación del desempeño, Analizándose el negocio que interviene en las evaluaciones de los trabajadores dándose cuenta de la necesidad de implantar un sistema de este tipo para el desarrollo de los recursos humanos en la UCI. Después de un fuerte análisis y diseño de la aplicación desarrollando todos los diagramas pertinentes para la correcta implementación del sistema. Al culminar este trabajo se cumple con el objetivo fundamental trazado, diseñar e implementar un sistema de evaluación del desempeño que satisfaga las necesidades de recursos humanos de la UCI.

Recomendaciones

- Realizar el análisis, diseño y implementación sobre las evaluaciones de los cuadros ya que en estos momentos no existe una infraestructura en la base de datos trabajadores para poder evaluarlos.
- Añadirles más funcionalidades al sistema a medida que se le haga necesario a los trabajadores de recursos humanos.

Referencias Bibliográficas

1. Larman, Craig. UML y Patrones. Introducción al análisis y diseño orientado a objetos. Prentice Hall Hispanoamérica, México, 1999.
2. Jacobson, Ivar y Booch, Grady y Rumbaugh, James. El proceso unificado de software. Primera edición. Pearson Educación, S.A. 2000.
3. Ceria, Santiago. *Ingeniería de Software I. Casos de Uso. Un Método Práctico para Explorar Requerimientos*.
4. Desarrollo basado en RUP bajo la herramienta Rational Rose <http://lml.ls.fi.upm.es/mdp/si/> (4/4/2004).
5. Curso práctico de desarrollo de aplicaciones con Visual Studio .NET <http://www.microsoft.com/spanish/msdn/comunidad/uni.net/> (4/4/2004).
6. Villariño, Luis. *UML para Web*. 2002.
7. Fowler, Martín. "UML Gota a Gota". Primera Edición. Addison Wesley Longman. 1999.
8. Meyer, B. *Construcción de software orientado a objetos*. Prentice-Hall. 1998.

Bibliografía

1. Larman, Craig. *UML y Patrones, Introducción al análisis y diseño orientado a objetos*. Prentice-Hall, 2002.
2. Fowler, Martín. "UML Gota a Gota". Primera Edición. Addison Wesley Longman. 1999.
3. Meyer, B. *Construcción de software orientado a objetos*. Prentice-Hall. 1998.
4. Ceria, Santiago. *Ingeniería de Software I. Casos de Uso. Un Método Práctico para Explorar Requerimientos*.
5. Desarrollo basado en RUP bajo la herramienta Rational Rose
<http://lml.ls.fi.upm.es/mdp/si/> (4/4/2004)
6. Universidad .NET
<http://www.microsoft.com/spanish/msdn/comunidad/uni.net/> (4/4/2004)
7. Curso práctico de desarrollo de aplicaciones con Visual Studio .NET
<http://www.microsoft.com/spanish/msdn/comunidad/uni.net/> (4/4/2004)
8. Villariño, Luis. *UML para Web*. 2002.
9. Booch, G., Rumbaugh, J., Jacobson, I. *El Lenguaje Unificado de Modelado*. Addison-Wesley. 1999.
10. Booch, G., Rumbaugh, J., Jacobson, I. *The Unified Software Development Process*. Addison-Wesley. 1999.
11. Cockburn, A. *Using Goal-Based Use Cases*. JOOP, 1997
12. García Molina, J. *Towards Use Case and Conceptual Models through Business Modeling*. Conference on Conceptual Modelling. 2000
13. Larman, C. *Applying UML and Patterns. An Introduction to Object-Oriented Analysis and Design*. Prentice-Hall, 1998.
14. Gamma, E. *Design Patterns, Elements of Reusable Object-Oriented Software*. Addison-Wesley. 1995.
15. Larman, Craig. *UML y Patrones, Introducción al análisis y diseño orientado a objetos*. Prentice-Hall, 2002.

Anexo 1. Modelo del negocio

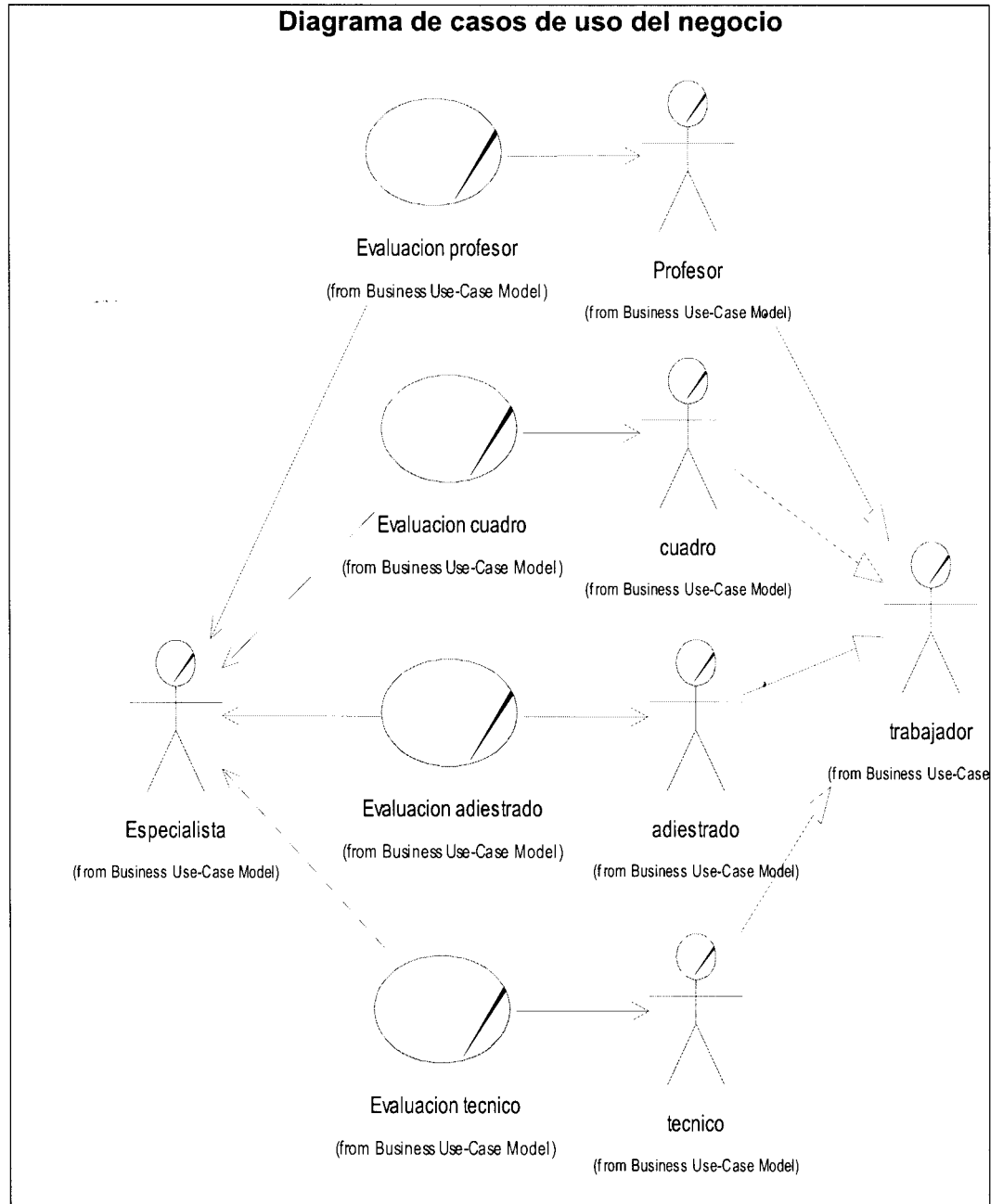
Actores del negocio.

Actores del negocio	Justificación
Trabajadores	Es el que beneficia o perjudica la evaluación que le realizan.
Profesores	Es un trabajador que se le hace una evaluación en específico.
Adiestrados	Es un trabajador que se le hace una evaluación en específico.
Técnicos	Es un trabajador que se le hace una evaluación en específico.
Cuadros	Es un trabajador que se le hace una evaluación en específico.
Especialista	Es el que le interesa y revisa la información de las evaluaciones de los trabajadores.

Trabajadores del negocio

Trabajadores del negocio	justificación
Jefe inmediato	Son los encargados de evaluar a los trabajadores subordinados a el.
Tutor	Son los encargados de evaluar al adiestrado que tiene a cargo junto con el jefe inmediato.

Diagramas de casos de uso del negocio



Descripción de los casos de uso del negocio:

Nombre del caso de uso del negocio:	Evaluación profesor
Actores del negocio:	Profesor, Especialista
Propósito:	Evaluar a los profesores.
Resumen: El caso de uso inicia cuando el curso escolar a terminado y el jefe inmediato evalúa a sus subordinados teniendo en cuenta los parámetros establecidos en la evaluación y el desempeño de estos ,el jefe inmediato y su subordinado firman la evaluación. El jefe inmediato le entrega las evaluaciones a la especialista que los atiende y esta analiza las evaluación. El caso de uso finaliza cuando la especialista en base a la evaluación toma decisiones.	
Flujo de trabajo	
Acción del actor	Respuesta del negocio
4.El profesor revisa evaluación 5.El profesor firma la evaluación. 7. Especialista analiza la evaluación. 8. Especialista toma desiciones en base de la evaluación.	1.Cuando termina el curso escolar el jefe inmediato analiza el desempeño de su subordinado 2. El jefe inmediato evalua a sus subordinados. 3. El jefe inmediato firma la evaluación. 6.El jefe inmediato entrega la evaluacion al especialista
Mejoras:	El jefe inmediato podar evaluar a sus subordinados mediante la Web y mejorara el flujo de información con la especialista.

Diagrama de actividad (Evaluacion Profesor)

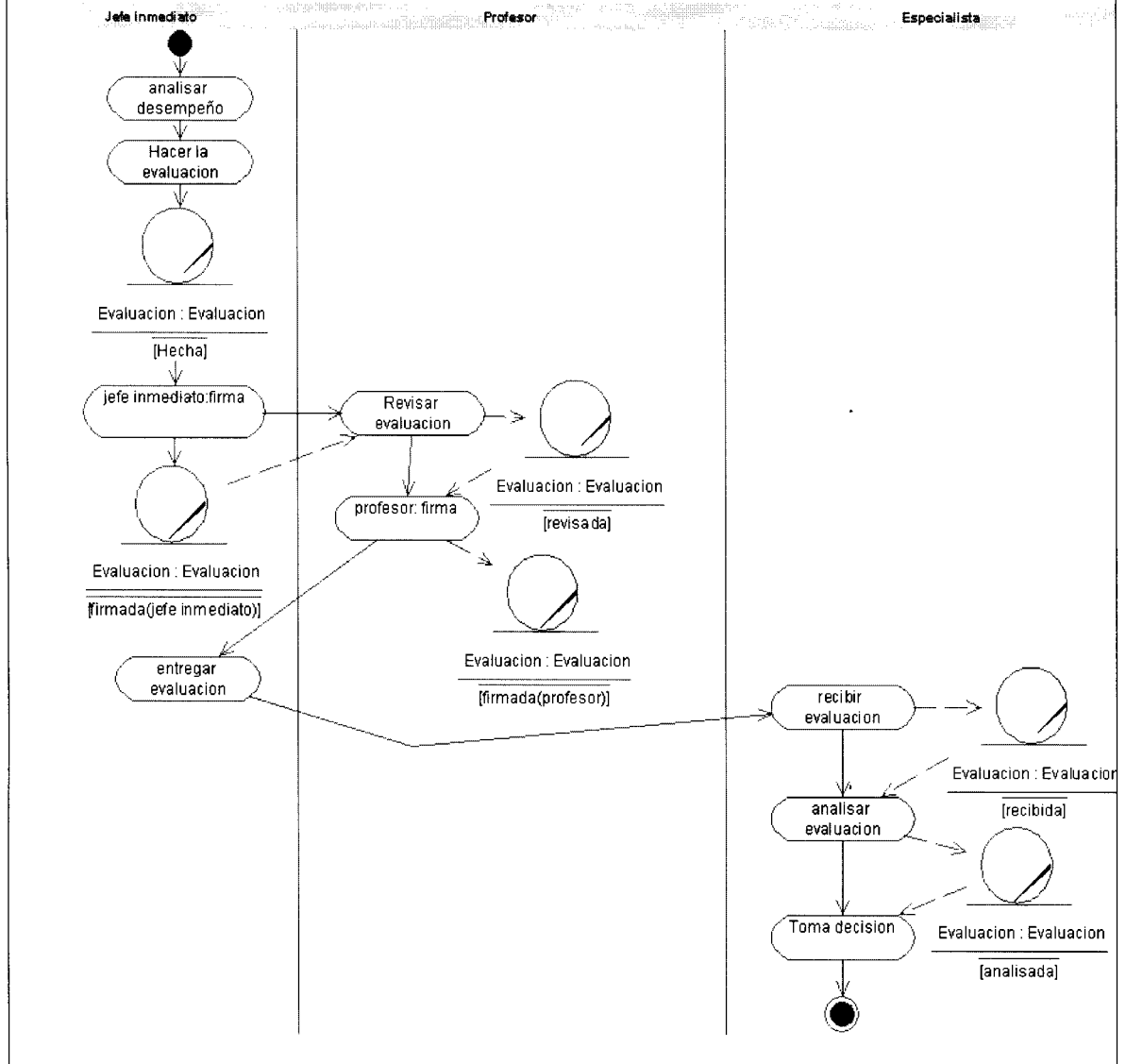
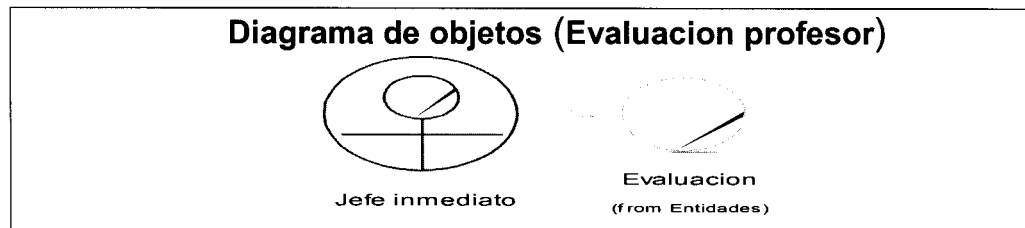


Diagrama de objetos (Evaluacion profesor)



Nombre del caso de uso del negocio:	Evaluación cuadro
Actores del negocio:	Cuadro, Especialista
Propósito:	Evaluar a los cuadros.
<p>Resumen: El caso de uso inicia cuando el cuadro cumple 2 años de desempeño del cargo que ocupa o de su anterior evaluación, el jefe inmediato le solicita a sus subordinados la auto evaluación, los cuadros hacen su autoevaluación y se la entregan a su jefe inmediato esta la revisa y le hace la evaluación a sus subordinados teniendo en cuenta su desempeño y la autoevaluación realizada, el jefe inmediato y sus subordinados firman la evaluaciones, el jefe inmediato le entrega las evaluaciones a la especialista que lo atiende y esta las analiza. Termina cuando la especialista en base a la evaluación toma decisiones.</p>	
Flujo de trabajo	
Acción del actor	Respuesta del negocio
<p>2. El trabajador hace su autoevaluación.</p> <p>7. El cuadro revisa evaluación</p> <p>8. El cuadro firma la evaluación.</p> <p>10. Especialista analiza la evaluación.</p> <p>11. Especialista toma decisiones en base de la evaluación.</p>	<p>1. Cuando el cuadro cumple 2 años en el cargo en que ocupa o de su anterior evaluación el jefe inmediato le solicita la autoevaluación.</p> <p>3. El jefe inmediato revisa la autoevaluación.</p> <p>4. El jefe inmediato analiza el desempeño de sus subordinados</p> <p>5. El jefe inmediato evalúa a sus subordinados.</p> <p>6. El jefe inmediato firma la evaluación.</p> <p>9. El jefe inmediato entrega la evaluación al especialista.</p>
Mejoras:	El jefe inmediato podrá evaluar a sus subordinados mediante la Web y mejorará el flujo de información con la especialista.

Diagrama de actividad (Evaluacion cuadros)

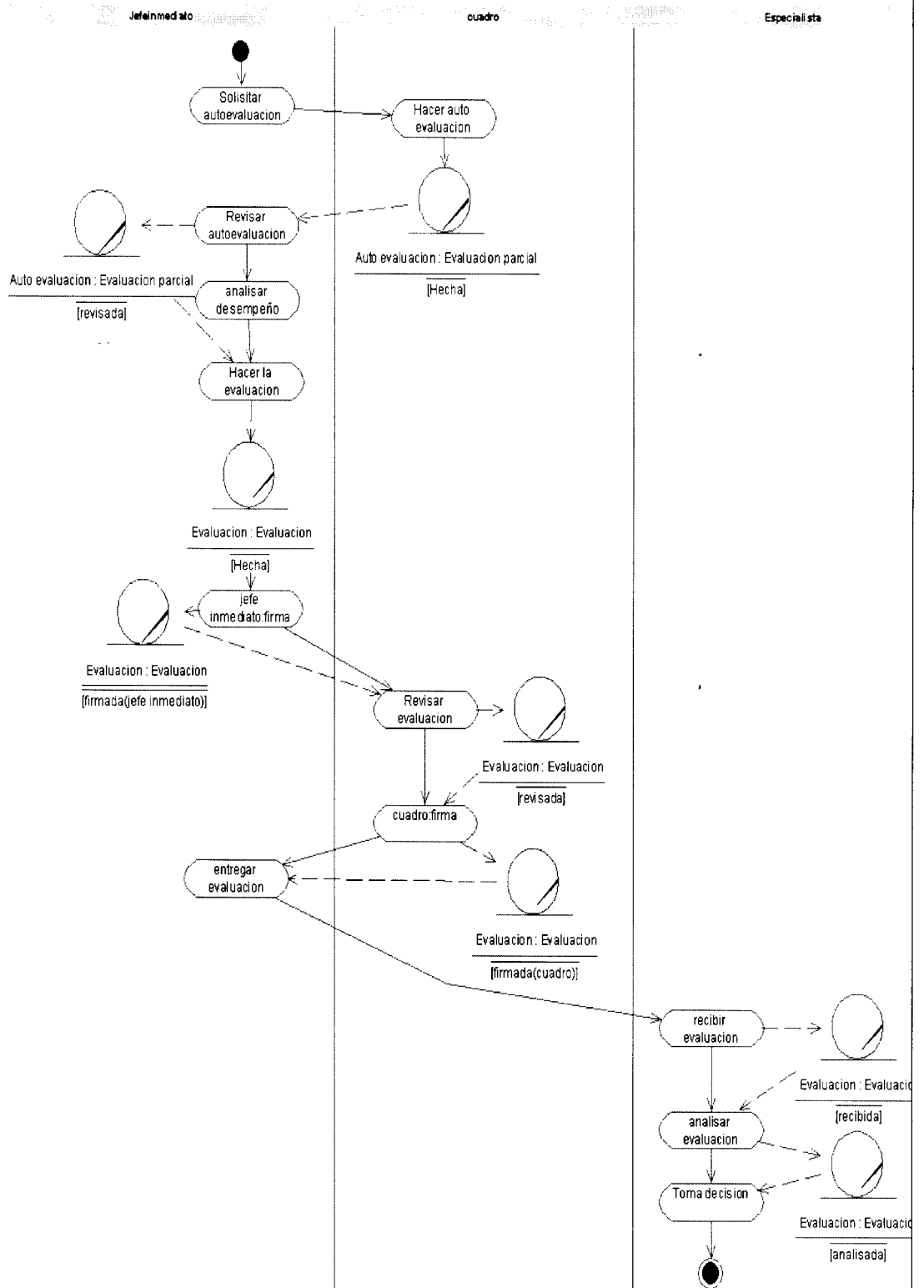
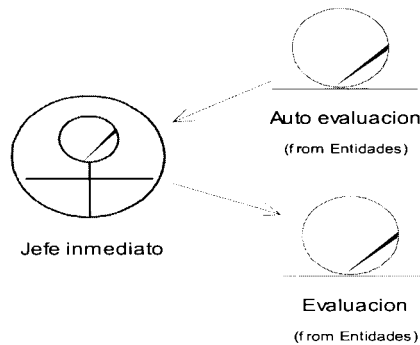


Diagrama de objetos (Evaluacion Cuadro)



Nombre del caso de uso del negocio:	Evaluación técnico
Actores del negocio:	Técnico, especialista
Propósito:	Evaluar a los técnicos.
<p>Resumen: El caso de uso inicia cuando el técnico cumple con el 70% del trabajo realizado. El jefe inmediato analiza el desempeño de sus subordinados y teniendo en cuenta los parámetros regidos por la evaluación le hace la evaluación a los técnicos subordinados a el .El técnico y el jefe inmediato firma la evaluación. El jefe inmediato le entrega las evaluaciones a la especialista que lo atiende y esta las analiza. Termina cuando la especialista en base a la evaluación toma decisiones.</p>	
Flujo de trabajo	
Acción del actor	Respuesta del negocio
<p>4.El técnico revisa evaluación</p> <p>5. El técnico firma la evaluación.</p> <p>7. Especialista analiza la evaluación.</p> <p>8. Especialista toma desiciones en</p>	<p>1.Cuando el tecnico cumple con el 70% del trabajo realizado el jefe inmediato analiza el desempeño.</p> <p>2. El jefe inmediato hace la evaluación.</p> <p>3. El jefe inmediato firma la evaluación.</p> <p>6. El jefe inmediato entrega la evaluacion al especialista.</p>

base de la evaluación.

Mejoras:

El jefe inmediato podar evaluar a sus subordinados mediante la Web y mejorara el flujo de información con la especialista.

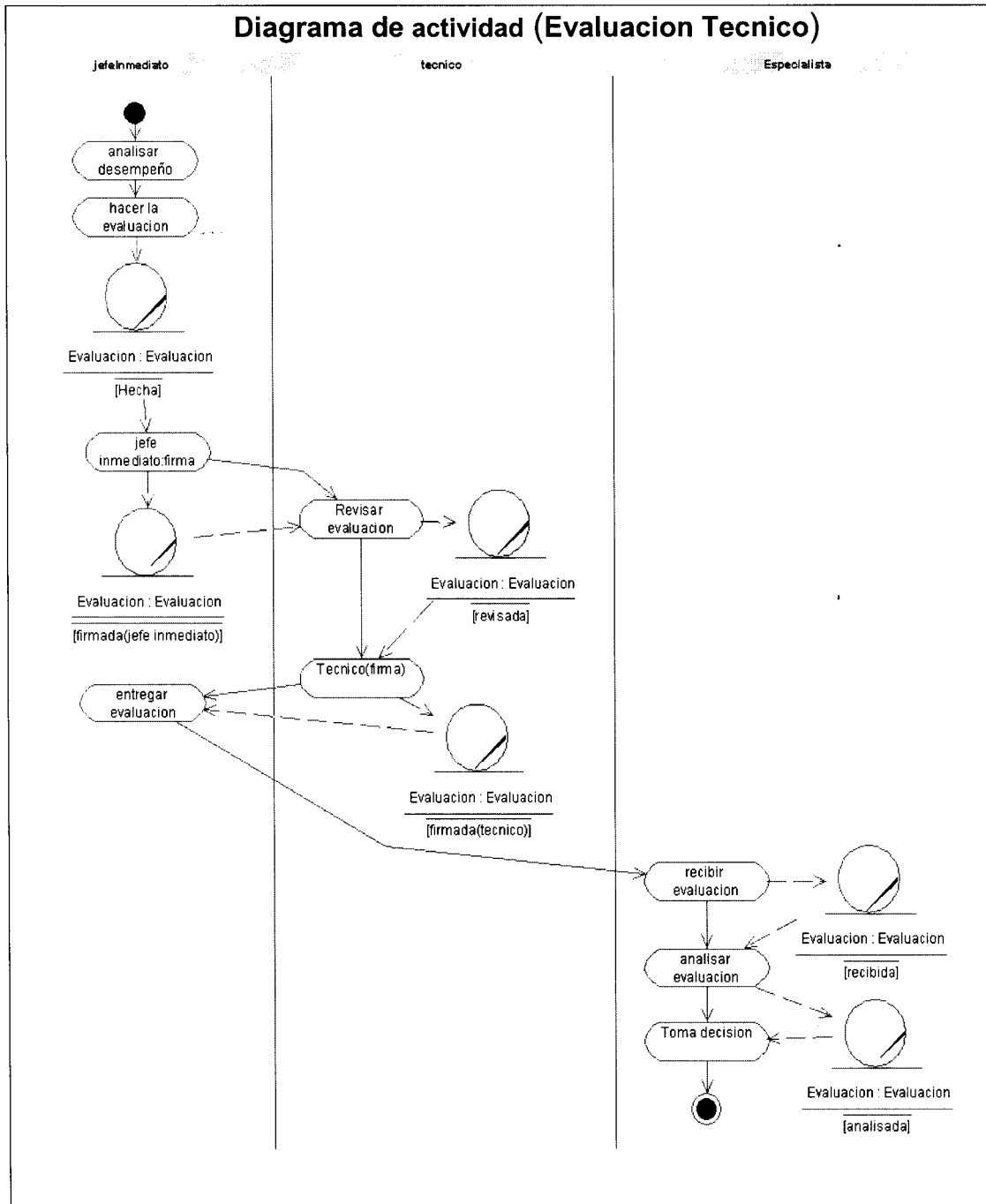
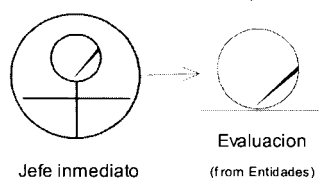


Diagrama de Objeto (Evaluación Técnico)



Nombre del caso de uso del negocio:	Evaluación adiestrado
Actores del negocio:	adiestrado
Propósito:	Evaluar a los adiestrados.
<p>Resumen: El caso de uso inicia cuando el adiestrado cumple 1 año de trabajo o haber cumplido 1 año de su anterior evaluación parcial. Si se cumple lo primero el jefe inmediato junto con el tutor del adiestra analizan el desempeño del adiestrado junto con los parámetros regidos por la evaluación y le hacen la evaluación parcial, en caso que se cumpla lo segundo El jefe inmediato junto con el tutor del adiestra analizan el desempeño del adiestrado junto con los parámetros regidos por la evaluación y le hacen la evolución parcial y la evaluación final analizando las dos parciales. El jefe inmediato, tutor y adiestrado firma la evaluación. El jefe inmediato le entrega las evaluaciones a la especialista que lo atiende y esta las analiza. Termina cuando la especialista en base a la evaluación toma decisiones.</p>	
Flujo de trabajo	
Acción del actor	Respuesta del negocio
<p>d). El adiestrado revisa la evaluación. e). El adiestrado firmar la evaluación.</p>	<p>1. Si el adiestrado cumple un año de estar trabajando o de su anterior evaluación. a)- El jefe inmediato y el tutor anilsan el desempeño del adiestrado. b)- El jefe inmediato con la alluda del tutor hace la evaluacion parcial. c)- El jefe inmediato y el tutor firma la evaluación.</p> <p>2. Si el adiestrado cumple un año de su anterior evaluación. a). Si no se cumple el paso 2 se salta al paso 3. b). El jefe in mediato junto con el tutor analizan las dos evaluaciones</p>

<p>e). El adiestrado revisa la evaluación.</p> <p>f). El adiestrado firmar la evaluación.</p> <p>4. Especialista analiza la evaluación.</p> <p>5. Especialista toma desiciones en base de la evaluación.</p>	<p>parciales.</p> <p>c). El jefe inmediato con la alluda del tutor hace la evaluacion final.</p> <p>d). El jefe inmediato y el tutor firma la evaluación.</p> <p>3. El jefe inmediato entrega las evaluaciones al especialista.</p>
<p>Mejoras:</p>	<p>El jefe inmediato y el tutor podar evaluar a sus subordinados mediante la Web y mejorara el flujo de información con la especialista.</p>

Diagrama de Actividad 1 (Evaluacion Adiestrado)

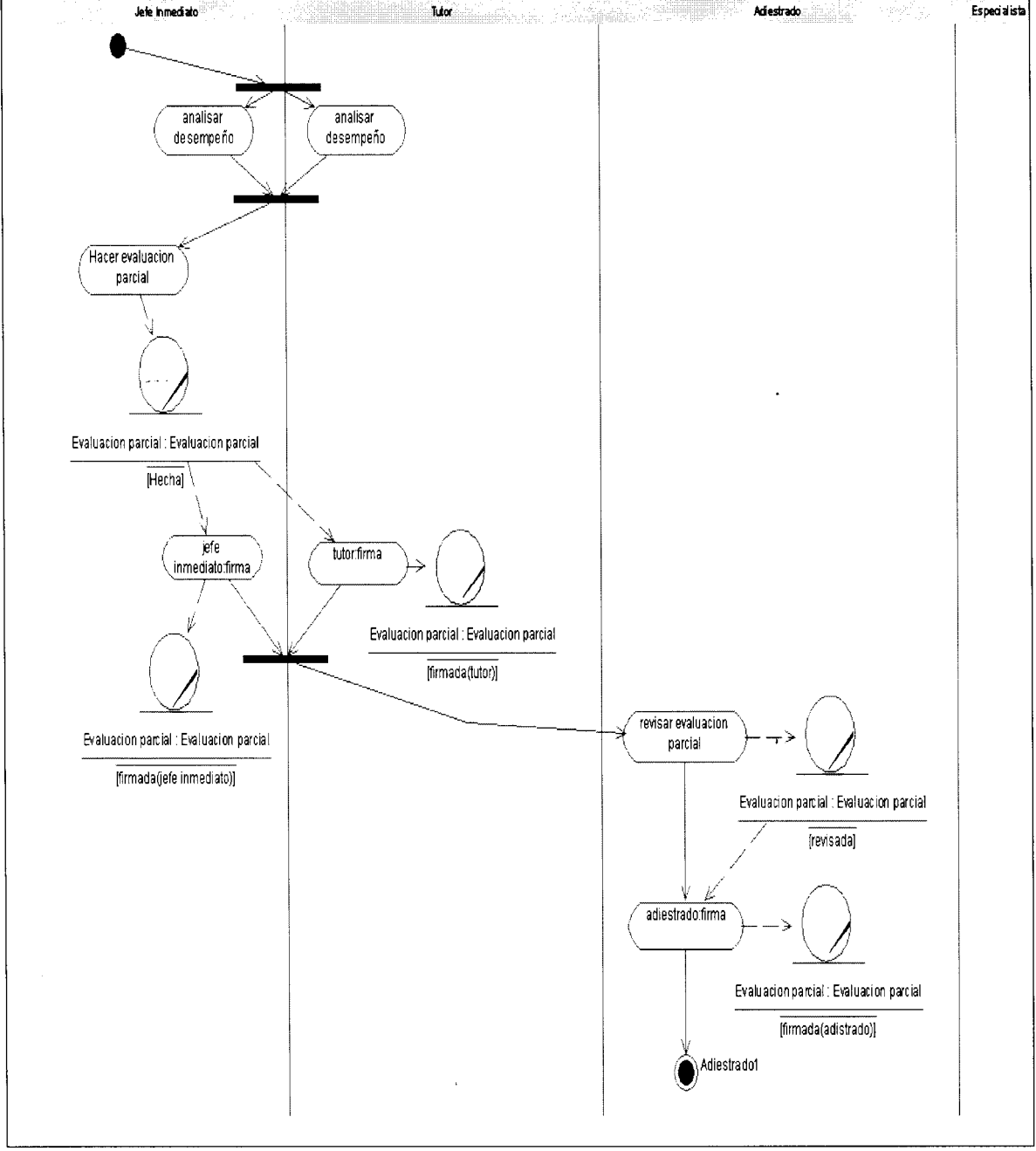


Diagrama de Actividad 2 (Evaluacion Adiestrado)

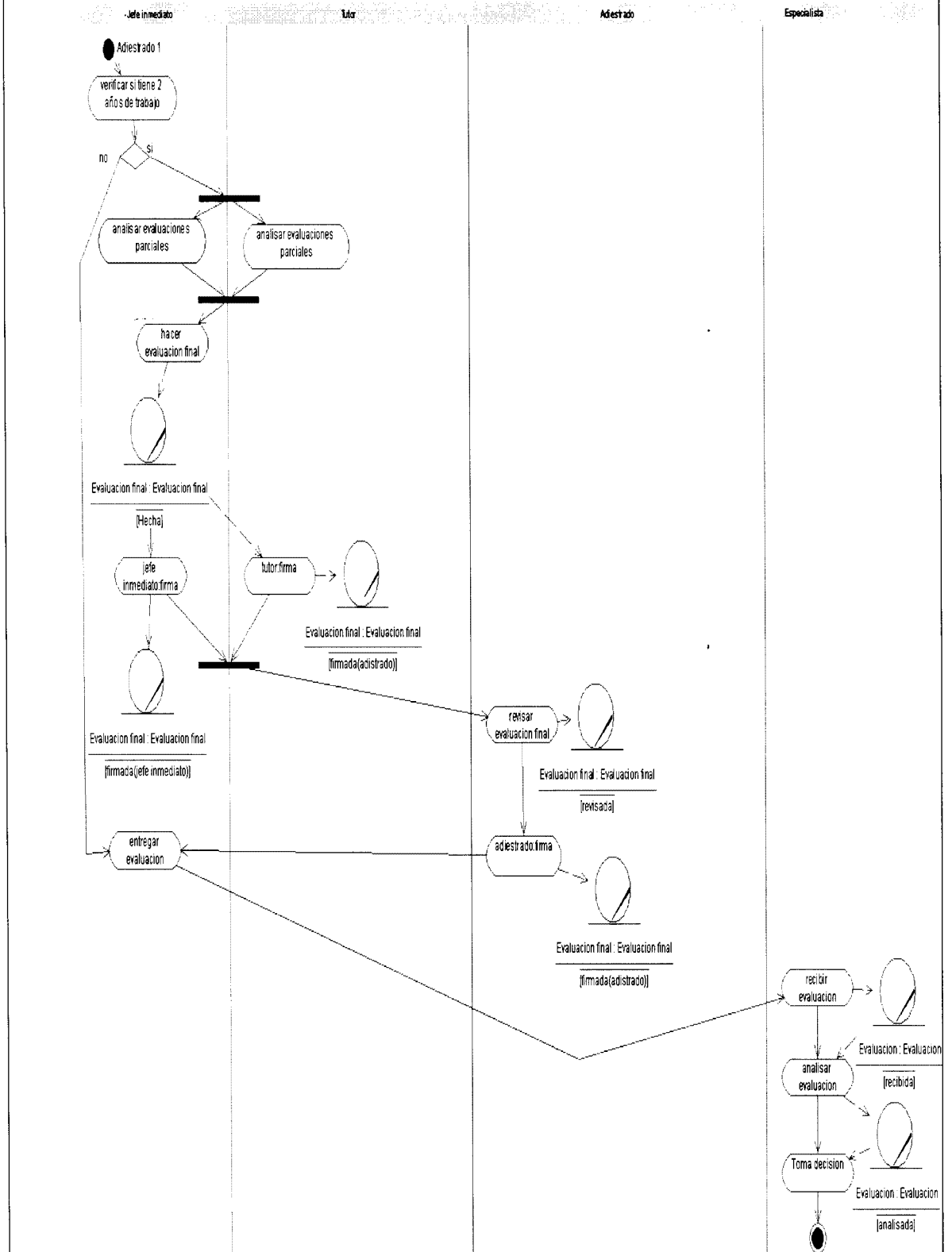
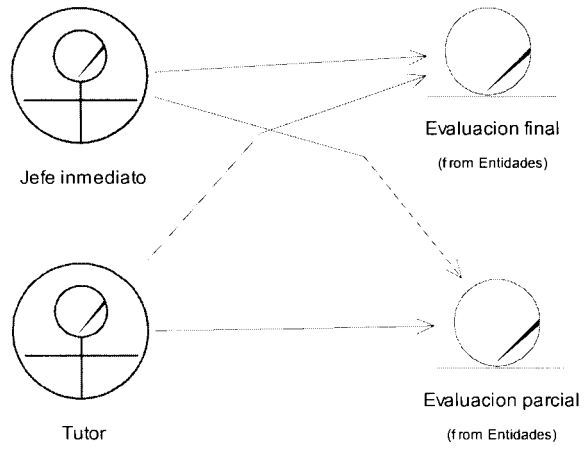


Diagrama de Objeto (Evaluacion Adiestrado)



Anexo 2. Definición de los casos de uso del sistema

Descripción de los actores del sistema

Actores	Justificación
Usuario	Son los únicos que tienen acceso al sistema.
Especialista	Son los que utilizan el servicio del sistema de evaluación para obtener información de la evaluación de los trabajadores.
Jefe inmediato	Son los que elaboran las evaluaciones utilizando este sistema.
Administrador	Son los que se encargan de administrar el sistema.

Descripción de los casos de uso del sistema

CU_01	Autenticar usuario.
Actor	Usuario(Inicia)
Descripción	El caso de uso inicia cuando cualquiera de los actores accede al sistema. En este proceso el sistema le pide al usuario: (Usuario y Contraseña), y por aquí se conoce el tipo de usuario y a que lugares del sistema puede acceder y con que permisos.
Referencia	R1

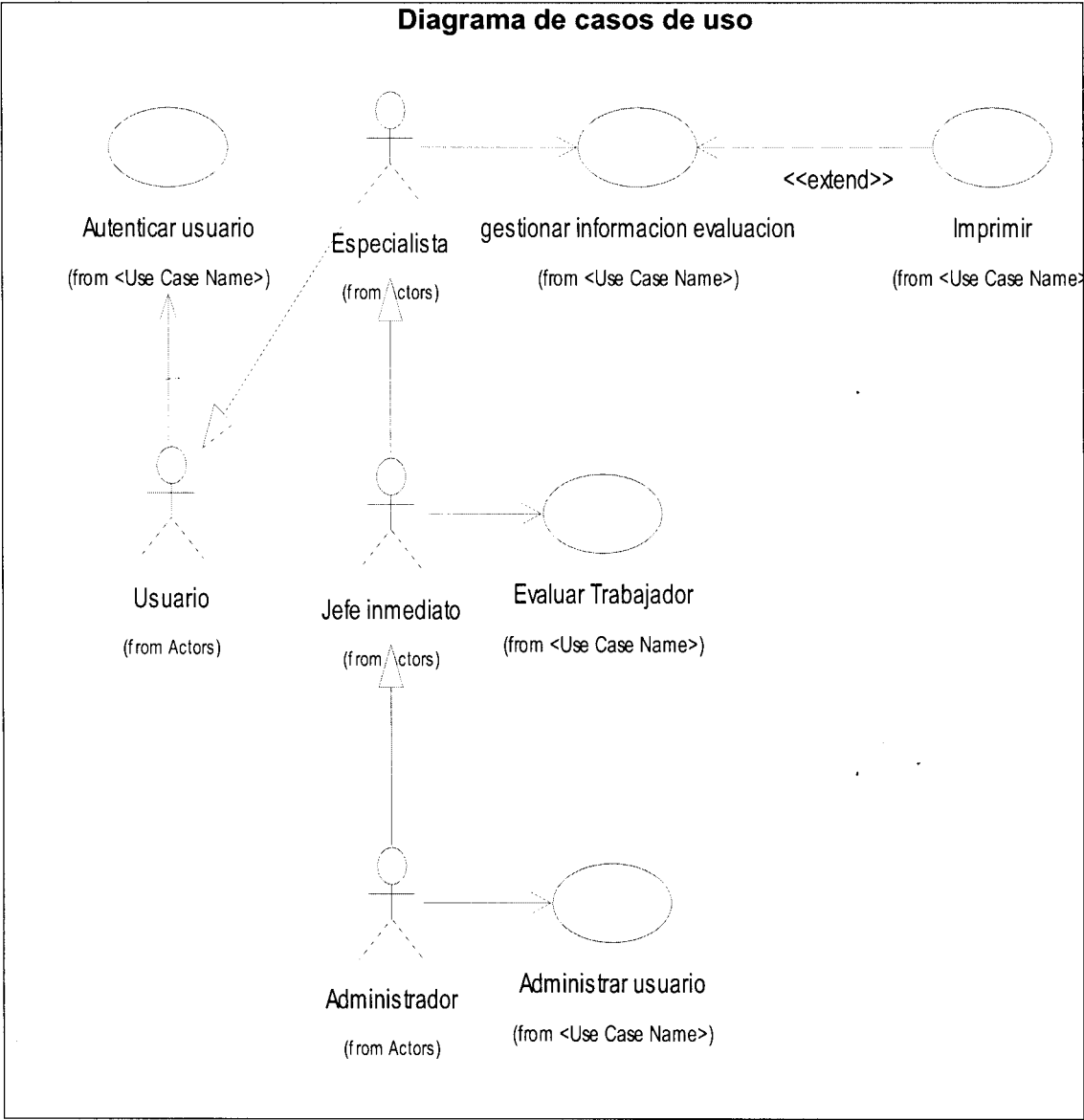
CU_02	Evaluar trabajador
Actor	Jefe inmediato(inicia)
Descripción	El caso de uso inicia cuando el jefe inmediato solicita insertar, modificar o eliminar la evaluación.
Referencia	R4,R5,R6,R7,R8,R9,R10,R11

CU_03	Gestionar información evaluados
Actor	Jefe inmediato(inicia), Especialista(inicia)
Descripción	El caso de uso inicia cuando el jefe inmediato o el especialista solicitan una lista de trabajadores evaluados o no evaluados según el periodo de evaluación. O solicitan la evaluación de un trabajador en específico.
Referencia	R12,R13,R14,R15,R16,R17, CU_05

CU_04	Administrar usuario
Actor	Administrador(inicia)
Descripción	El caso de uso inicia cuando un Administrador desea adicionar o actualizar un Usuario y darles el acceso correspondiente al sistema.
Referencia	R2,R3

CU_05	Imprimir
Actor	
Descripción	El caso de uso inicia cuando en el CU_3 (imprimir=True) y manda imprimir la impresora.
Referencia	R18

Diagrama de casos de uso del sistema



Casos de uso por ciclo de desarrollo

Ciclo	Cód	Nombre de caso de uso	Justificación de la selección.
1	CU_02	Evaluar trabajador	Representa las tareas de mayor uso en el sistema, contienen la información sobre las evaluaciones y permite la obtención de información derivada de las evaluaciones.
	CU_03	Gestionar información evaluados	
	CU_05	Imprimir	
2	CU_01	Autenticar usuario	No interviene directamente en el funcionamiento del sistema pero es fundamental para garantizar el control y administración del sistema.
	CU_04	Administrar usuario	

Casos de uso expandidos

Caso de uso	
CU_01	Autenticar usuario
Propósito	Se realiza la autenticación del usuario para poder mostrar las opciones correspondientes a cada uno de los autenticados.
Resumen:	El caso de uso inicia cuando el jefe inmediato, especialista o administrador se autentican introduciendo clave y contraseña para ser reconocidos por el sistema para poder interactuar con este.
Referencias	R1
Acción del actor	Respuesta del sistema
<p>1- Comienza cuando un actor se va a autenticar en el sistema.</p> <p>3- El actor introduce su nombre de usuario y contraseña.</p>	<p>2- El sistema muestra un formulario donde podrá escribir su nombre de usuario y contraseña.</p> <p>4- El sistema solicita verificar si el nombre de usuario escrito existe, si existe obtiene el tipo de usuario y si la contraseña esta correcta el sistema muestra de acuerdo al tipo de usuario la página con las opciones que puede realizar.</p>
Puntos de extensión.	

Caso de uso	
CU-02	Evaluar Trabajador
Propósito	Insertar, Actualizar y eliminar las evaluaciones de los trabajadores
Resumen: El caso de uso inicia cuando el jefe inmediato solicita insertar, modificar o eliminar la evaluación.	
Referencias	R4,R5,R6,R7,R8,R9,R10,R11
Acción del actor	Respuesta del sistema
<p>1- Comienza cuando el jefe inmediato solicita actualizar la evaluación de un trabajador.</p> <p>3- El jefe inmediato selecciona la opción que desea realizar.</p> <p>3.1- Insertar evaluación de un trabajador. Ver Insertar evaluación.</p> <p>3.2- Modificar evaluación de un trabajador. Ver Modificar evaluación.</p> <p>3.3- Eliminar la evaluación de un trabajador. Ver eliminar evaluación.</p>	<p>2- El sistema le da la posibilidad de insertar la evaluación o modificar y eliminar las que ya estén realizadas.</p>
Insertar evaluación	
Acción del actor	Respuesta del sistema
<p>1- Comienza cuando el jefe inmediato solicita insertar los datos de la evaluación del trabajador.</p> <p>3- El jefe inmediato introduce la información para buscar al trabajador.</p> <p>5- El jefe inmediato elige al trabajador que desea insertarle la evaluación.</p> <p>7- El jefe inmediato inserta los datos de la evaluación.</p>	<p>2- El sistema le muestra un Formulario (número del solapín) para buscar el trabajador al cual le desea realizar la evaluación (En caso de que el jefe inmediato no entre el número de solapín busca a todos los trabajadores subordinados a él).</p> <p>4- El sistema muestra (nombre, apellido, periodo de evaluación de la última evaluación insertada, resultado evaluación) de los trabajadores encontrados.</p> <p>6- El sistema muestra un formulario con el modelo de evaluación correspondiente al cargo del trabajador.</p> <p>8- El sistema inserta los datos de la evaluación en la base de datos.</p>
Modificar evaluación	
Acción del actor	Respuesta del sistema

<p>1- Comienza cuando el jefe inmediato solicita modificar los datos de la evaluación del trabajador.</p> <p>3- El jefe inmediato introduce la información para buscar al trabajador.</p> <p>5- El jefe inmediato elige al trabajador que desea modificar la evaluación.</p> <p>7- El jefe inmediato modifica los datos de la evaluación.</p>	<p>2- El sistema le muestra un formulario (número del solapín, periodo de evaluación) para buscar el trabajador al cual le desea modificar la evaluación (En caso de que el jefe inmediato no entre el número de solapín, busca a todos los trabajadores subordinados a él, que tengan el periodo de evaluación igual que el introducido en el formulario).</p> <p>4- El sistema muestra (nombre, apellido, periodo de evaluación, resultado evaluación) de los trabajadores encontrados.</p> <p>6- El sistema muestra un formulario con el modelo de evaluación correspondiente al tipo de evaluación con los datos a modificar.</p> <p>8- El sistema modifica los datos de la evaluación en la base de datos.</p>
---	---

Eliminar evaluación

Acción del actor	Respuesta del sistema
<p>1- Comienza cuando el jefe inmediato solicita eliminar los datos de la evaluación del trabajador.</p> <p>3- El jefe inmediato introduce la información para buscar al trabajador.</p> <p>5- El jefe inmediato elige al trabajador que desea eliminar la evaluación y se la elimina.</p>	<p>2- El sistema le muestra un formulario (número del solapín, periodo de evaluación) para buscar el trabajador al cual le desea eliminar la evaluación (En caso de que el jefe inmediato no entre el número de solapín, busca a todos los trabajadores subordinados a él, que tengan el periodo de evaluación igual que el introducido en el formulario).</p> <p>4- El sistema muestra (nombre, apellido, periodo de evaluación, resultado evaluación) de los trabajadores encontrados.</p> <p>6- El sistema elimina la evaluación de la base de datos.</p>

Puntos de extensión.

(11)

Caso de uso	
CU_05	Imprimir
Propósito	Mandar un mensaje de impresión a la impresora conectada.
Resumen: El caso de uso inicia cuando en el CU_3 (imprimir= True) y manda imprimir la impresora.	
Referencias	R19
Acción del actor	Respuesta del sistema
1 El caso de uso inicia cuando en el CU_3 (imprimir= True).	2- El sistema manda un mensaje de impresión a la impresora de lo solicitado por el CU_3.
Puntos de extensión.	

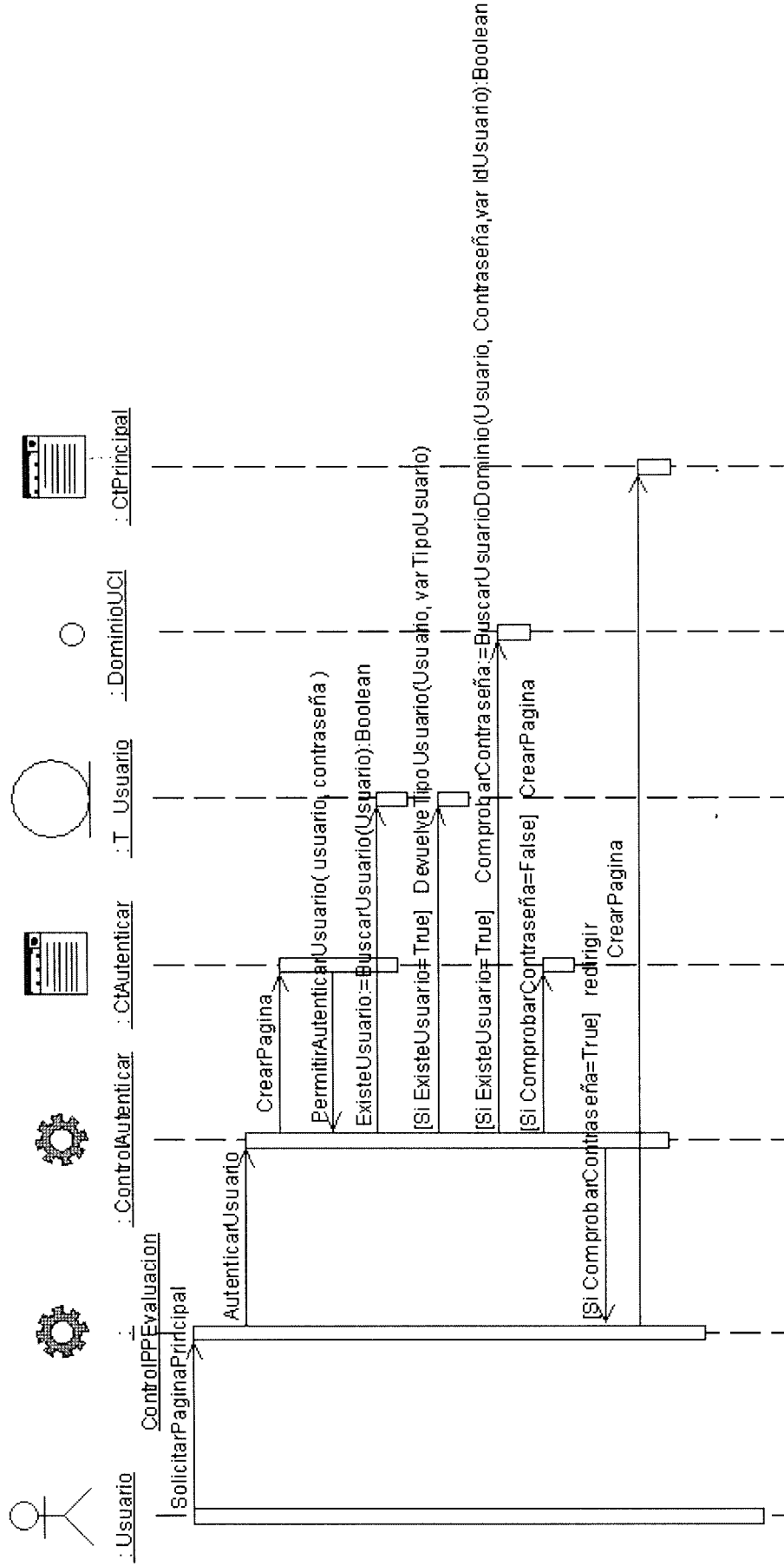
Caso de uso	
CU_03	Gestionar información evaluados
Propósito	Gestionar la información de los trabajadores en base a la evaluación.
Resumen: El caso de uso inicia cuando el jefe inmediato o el especialista solicitan una lista de trabajadores evaluados o no evaluados según el periodo de evaluación. O solicitan de un trabajador en específico.	
Referencias	R12,R13, CU_05
Acción del actor	Respuesta del sistema
<p>1- El caso de uso inicia cuando el jefe inmediato o la especialista solicita gestionar la información de sus trabajadores en base a la evaluación.</p> <p>3- El jefe inmediato o la especialista introduce los datos y la condición de búsqueda.</p> <p>5- Si el jefe inmediato o la especialista quiere mostrar la evaluación de un trabajador lo selecciona (En caso que el sistema le muestre los evaluados).</p>	<p>2- El sistema le muestra un formulario con (número de solapín [en caso de que quiera que le muestre un trabajador en específico], periodo de evaluación, condición a buscar [evaluados o no evaluados]).</p> <p>4- El sistema le muestra en caso de los no evaluados (número de solapín, are, cargo, Por ciento de trabajo realizado.) de los encontrados. En caso de los evaluados (número de solapín, periodo de evaluación, are, cargo, resultado evaluación).</p> <p>6- El sistema le muestra la evaluación del trabajador seleccionado.</p>
Puntos de extensión.	

Caso de uso	
CU_04	Administrar usuario
Propósito	Insertar, actualizar, eliminar usuarios
Resumen: El caso de uso inicia cuando el administrador solicita al sistema insertar, actualizar o eliminar un usuario.	
Referencias	R2,R3
as	
Acción del actor	Respuesta del sistema
<p>1- Comienza cuando el administrador solicita administrar usuario.</p> <p>3- El administrador selecciona la opción que desea realizar.</p> <p>3.1- Insertar usuario. Ver Insertar usuario.</p> <p>3.2- Modificar usuario. Ver Modificar usuario.</p> <p>3.3- Eliminar usuario. Ver eliminar usuario.</p>	<p>2- El sistema le muestra un formulario con (nombre usuario, tipo de usuario [jefe inmediato, especialista, administrador y le muestra todos los usuarios insertados con (nombre usuario, tipo de usuario. El sistema le da la posibilidad de insertar la evaluación, actualizar y eliminar las que ya estén realizadas.</p>
Insertar usuario.	
Acción del actor	Respuesta del sistema
<p>1- Comienza cuando el administrador solicita insertar un usuario.</p> <p>3- El administrador inserta al usuario.</p>	<p>2- El sistema le muestra (nombre usuario, tipo de usuario [jefe inmediato, especialista adiestrado, especialista Técnico, especialista Profesor administrador].</p> <p>4- El sistema inserta al usuario en la base de datos.</p>
Modificar usuario.	
Acción del actor	Respuesta del sistema
<p>1- El caso de uso comienza cuando el administrador solicita actualizar un usuario.</p> <p>3- El administrador selecciona al usuario y lo actualiza.</p>	<p>2- El sistema le muestra un formulario con (nombre usuario, tipo de usuario. De los usuarios que existen en el sistema.</p> <p>4- El sistema actualiza la información en la base de datos.</p>
Eliminar usuario.	

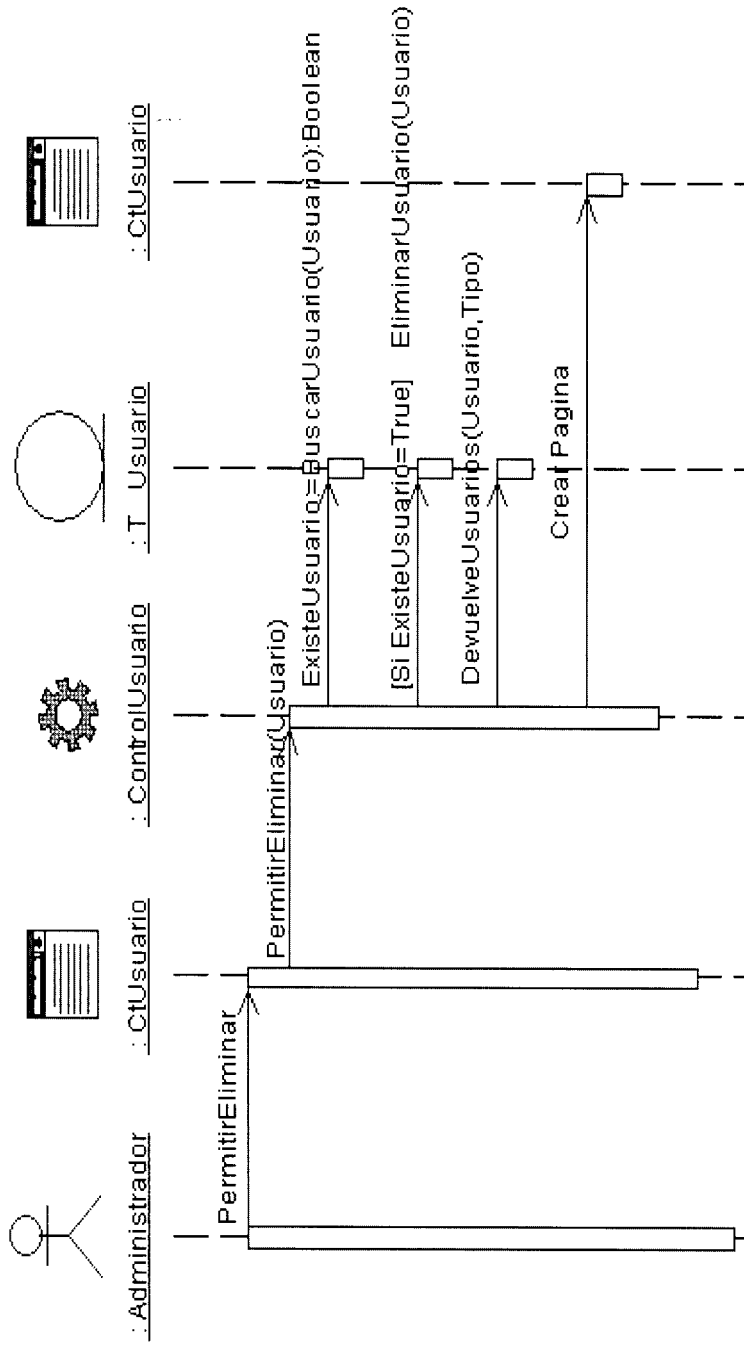
Acción del actor 1- El caso de uso comienza cuando el administrador solicita eliminar un usuario. 3- El administrador selecciona al usuario y lo elimina.	Respuesta del sistema 2- El sistema le muestra un formulario con (nombre usuario, tipo de usuario. De los usuarios que existen en el sistema. 4- El sistema elimina la información en la base de datos.
Puntos de extensión.	

Anexo 3. Diagramas de interacción

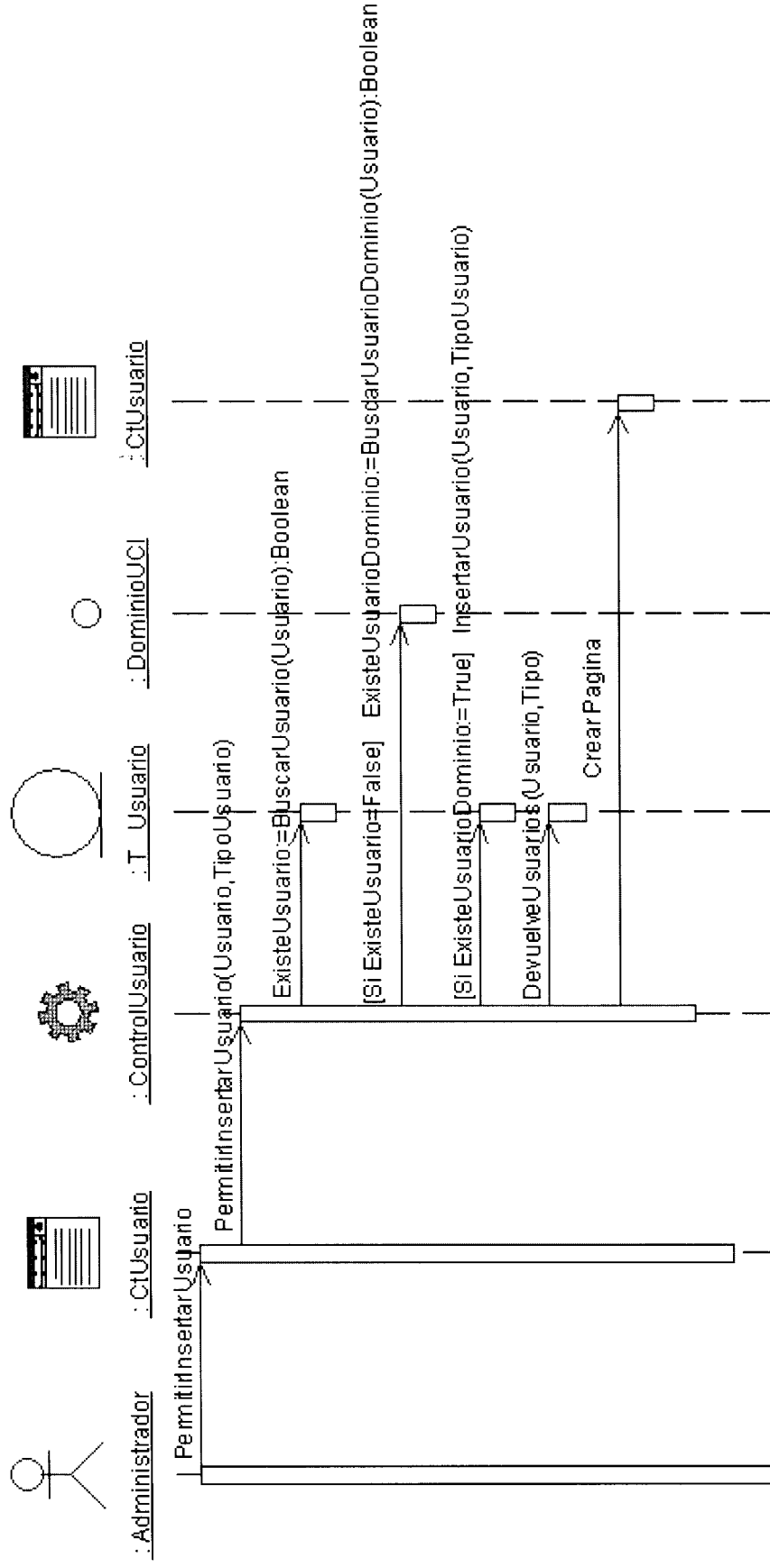
Autenticar usuario:



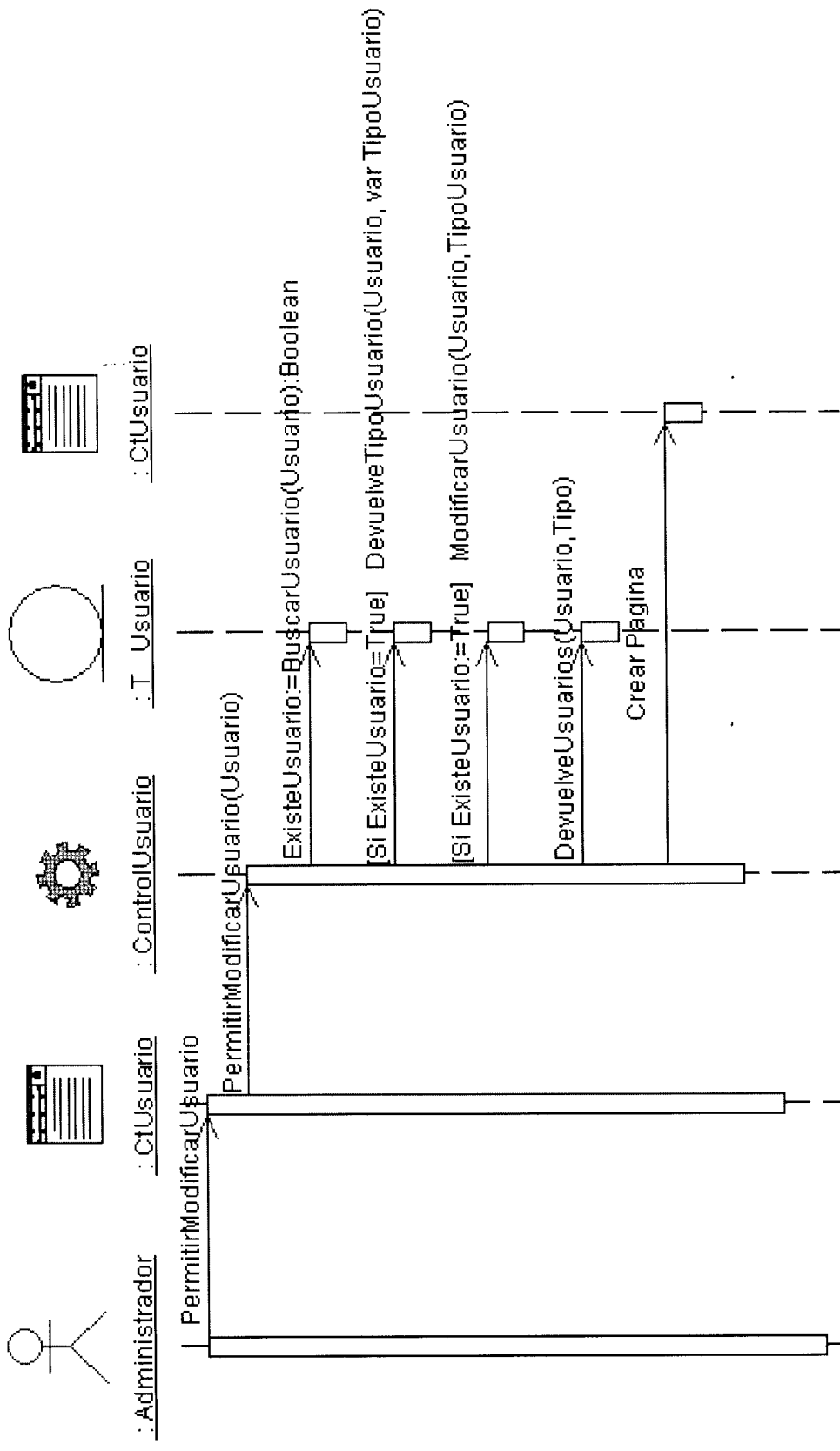
Eliminar usuario:



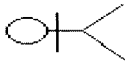
Insertar Usuario:



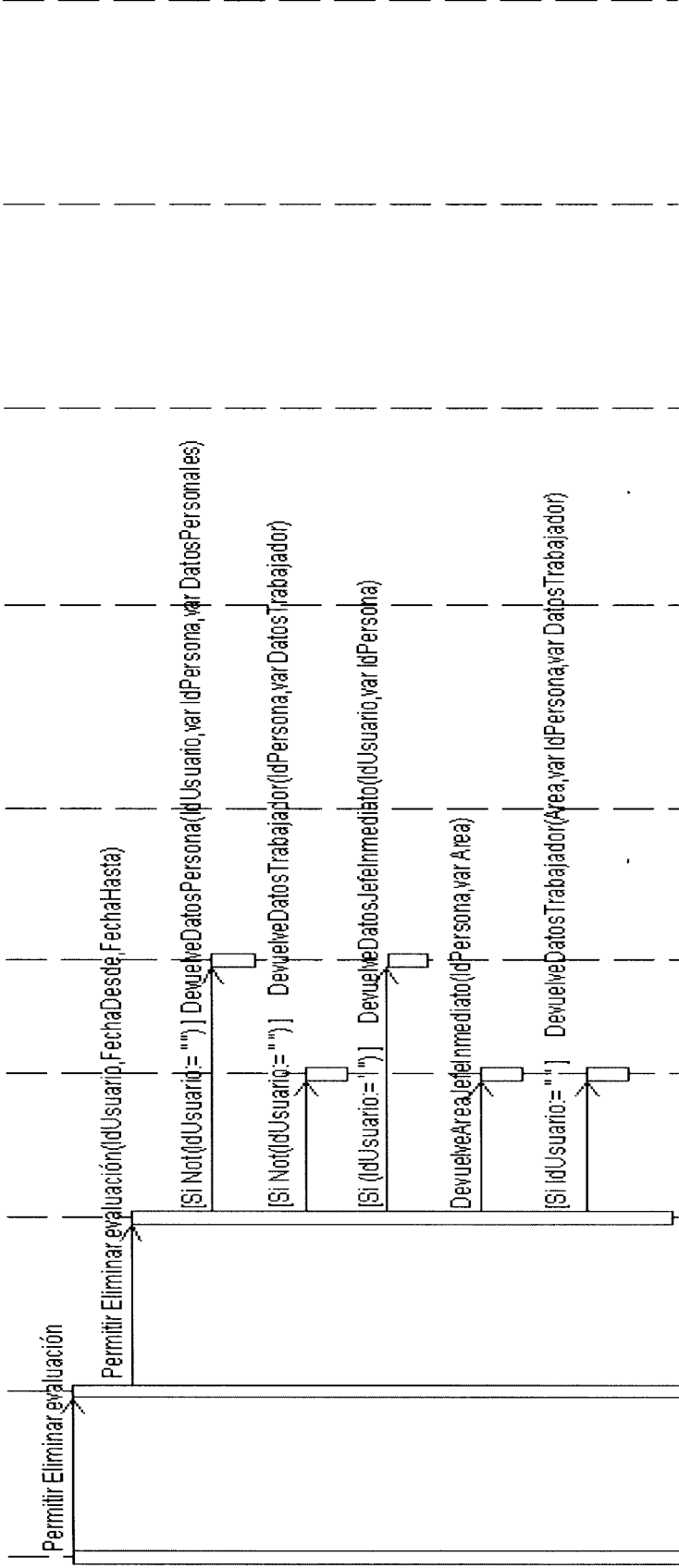
Modificar Usuario:



Eliminar Evaluación (a) continua.



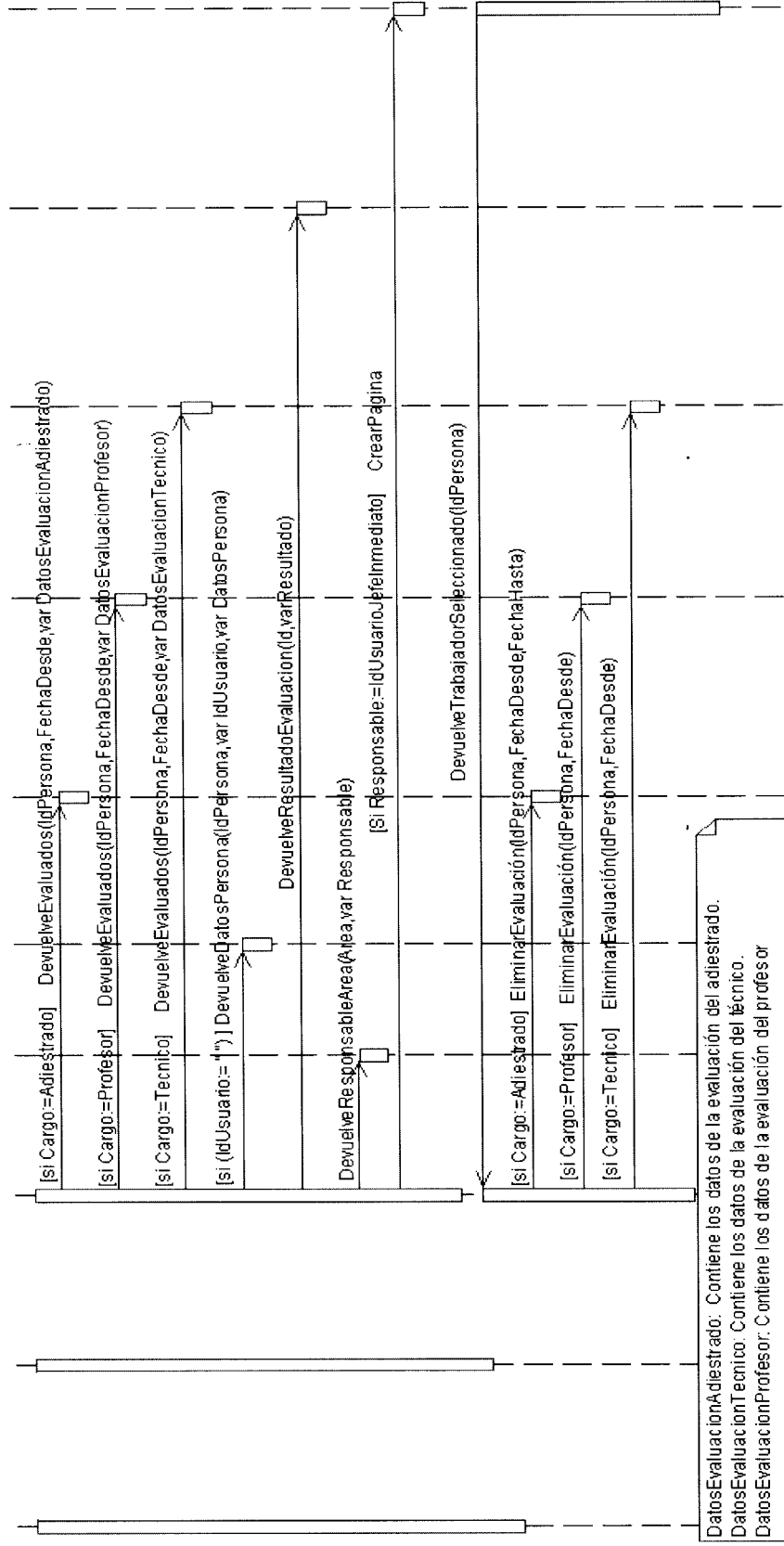
..Jefe Inmediato : C\EliminarEvaluacion : ControlEvaluacion : Trabajadores : Persona : T_EvaluacionAfiliado : T_EvaluacionProfesor : T_EvaluacionTecnico : T_ResultadoEvaluacion : C\EliminarEvaluacion



Eliminar Evaluación (b).

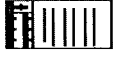
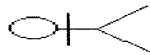


..Jefe inmediato : CtEliminarEvaluacion : ControlEvaluacion : Persona .. Trabajadores : Persona .. I_EvaluacionAdiestrado : I_EvaluacionProfesor : I_EvaluacionTecnico : I_ResultadoEvaluacion : CtEliminarEvaluacion

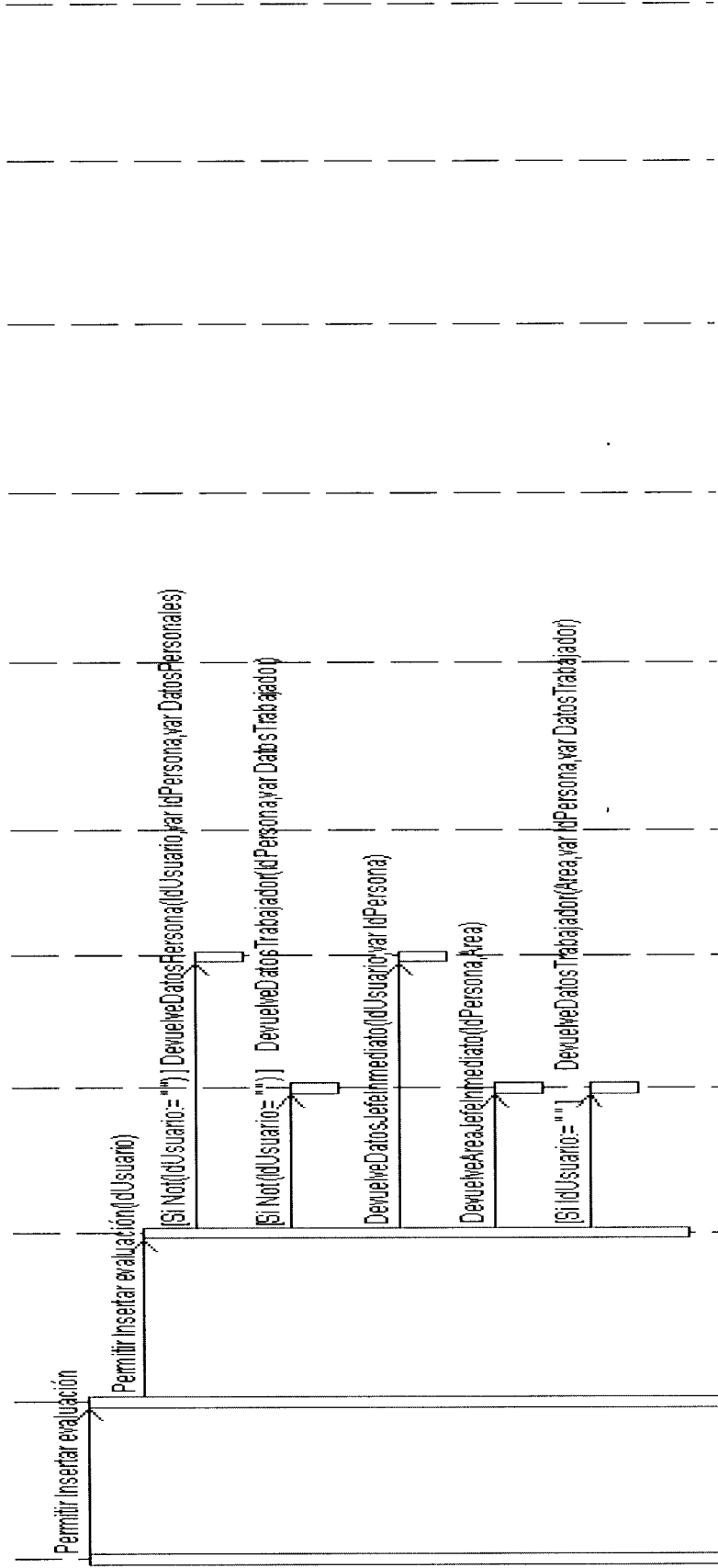


DatosEvaluacionAdiestrado: Contiene los datos de la evaluación del adiestrado.
 DatosEvaluacionTecnico: Contiene los datos de la evaluación del técnico.
 DatosEvaluacionProfesor: Contiene los datos de la evaluación del profesor

Insertar evaluación (a) continua:



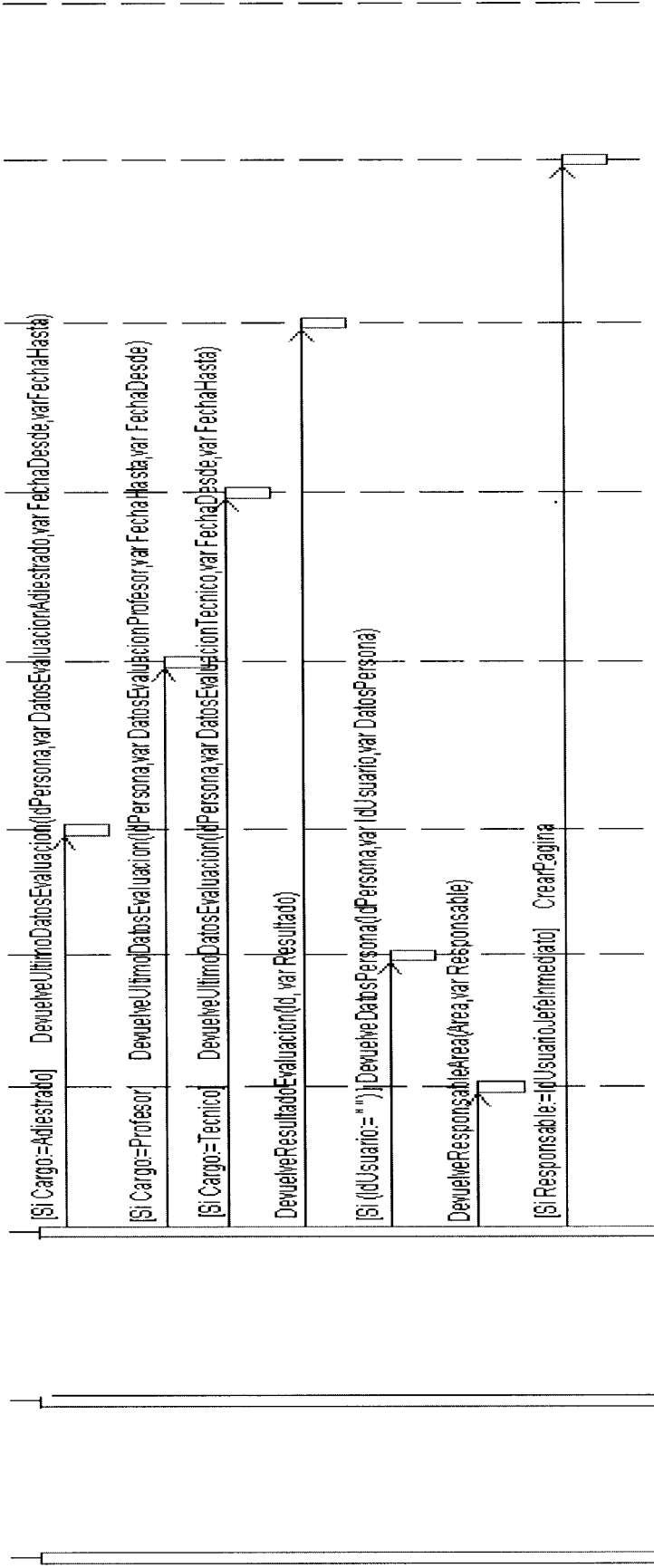
:Jefe inmediato :CInsertarEvaluacion :ControlEvaluacion :Trabajadores :Persona :TEvaluacionAdiestrado :TEvaluacionProfesor :TEvaluacionTecnico :TResultadoEvaluacion :CInsertarEvaluacion :CModeloEvaluacion



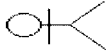
Insertar evaluación (b) continua:



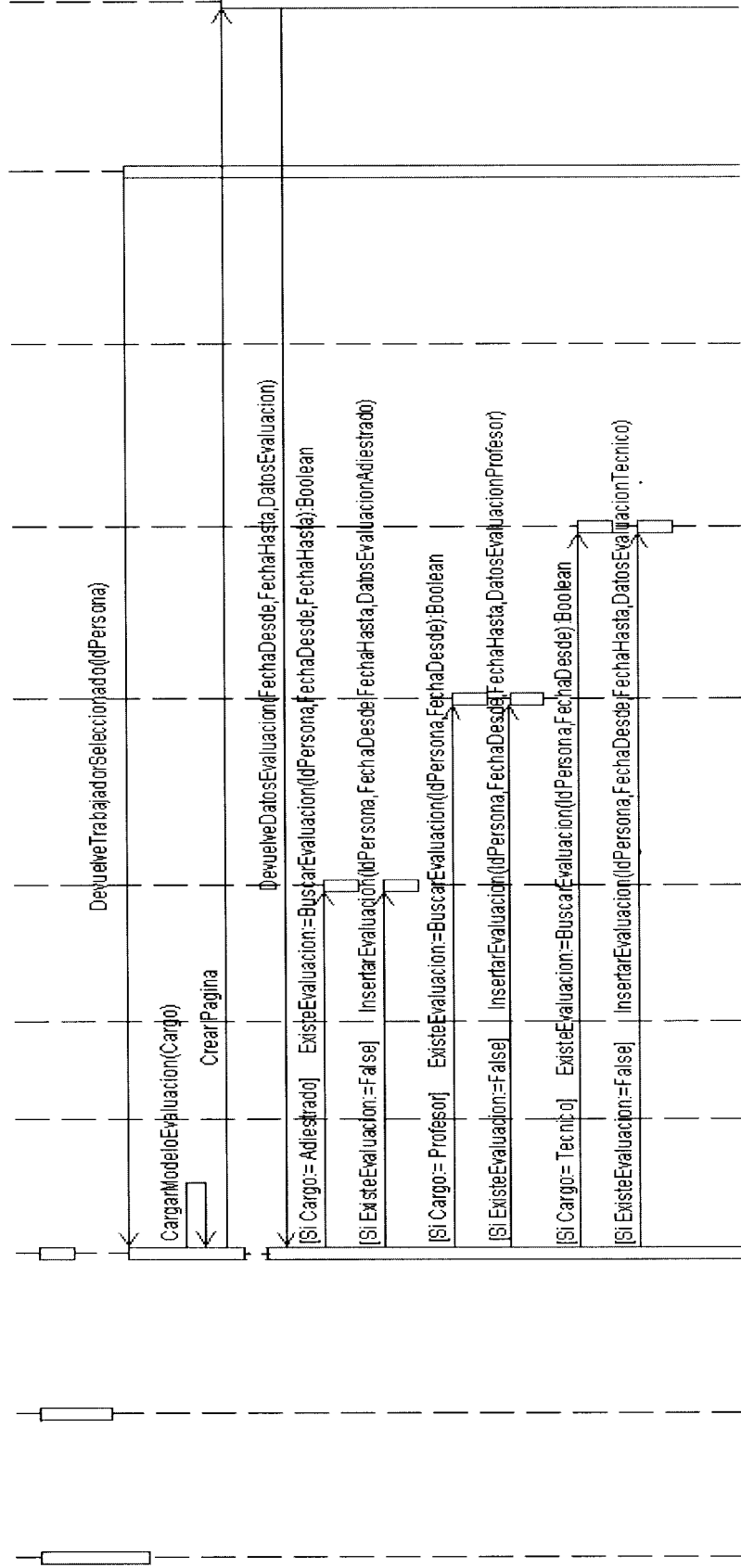
Jefe inmediato : ControlEvaluacion : Trabajadores : Persona : EvaluacionAdestrado : EvaluacionProfesor : EvaluacionTecnico : ResultadoEvaluacion : InsertarEvaluacion : ModeloEvaluacion



Insertar evaluación (c).



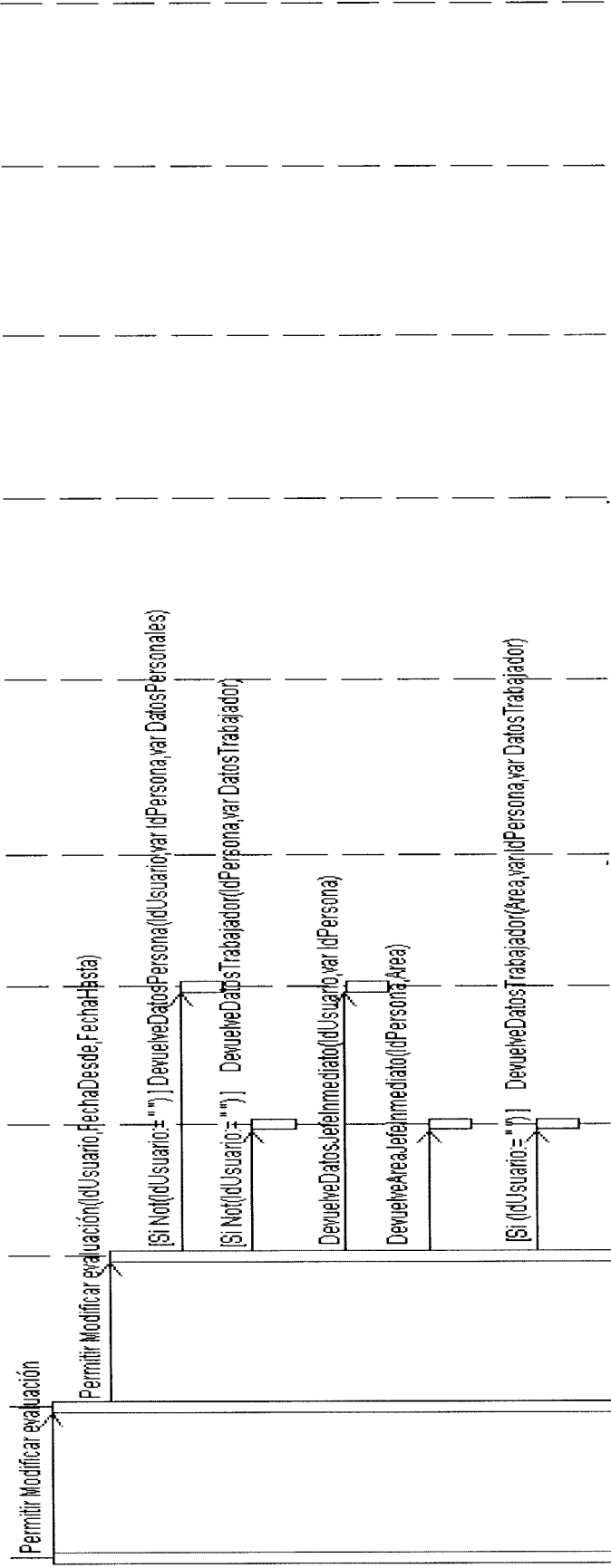
..Jefe inmediato : CiInsertarEvaluacion : ControlEvaluacion : Trabajadores : Persona : T. EvaluacionAdiestrado : T. EvaluacionProfesor : T. EvaluacionTecnico : T. ResultadoEvaluacion : CiInsertarEvaluacion : CiInsertarEvaluacion : CiModeloEvaluacion



Modificar evaluación (a) continua.



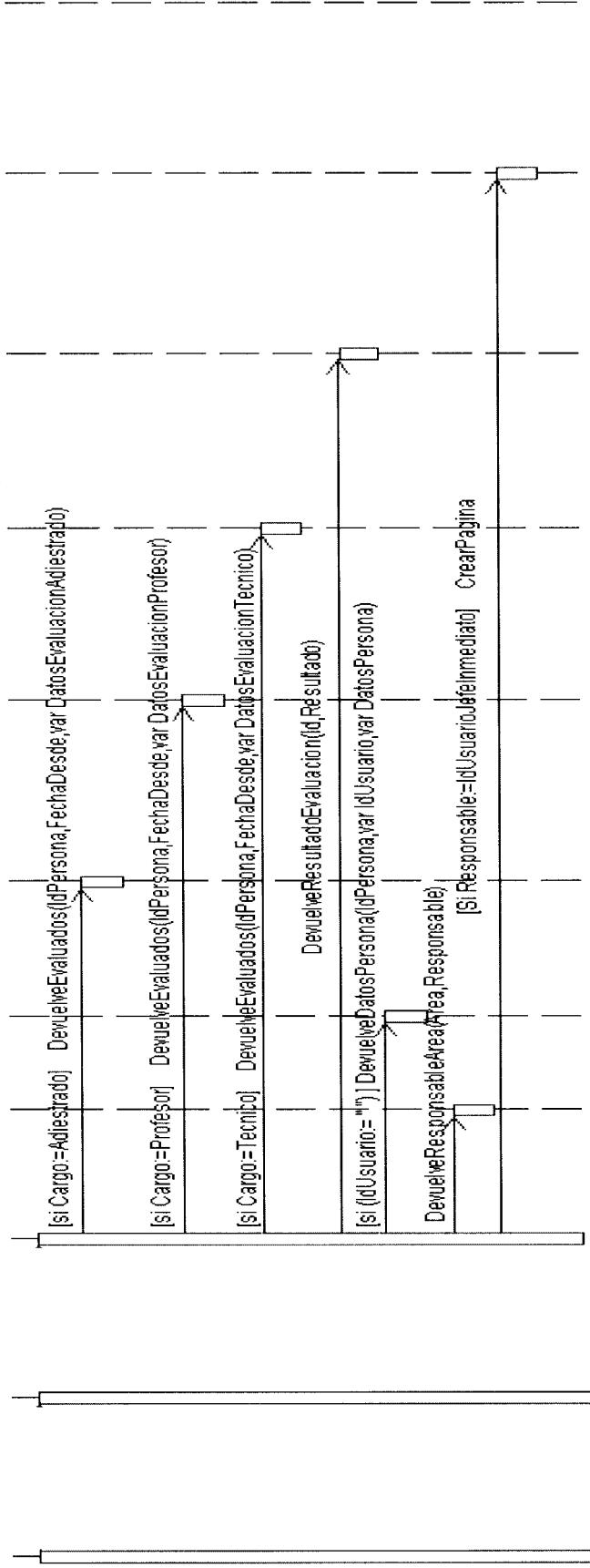
:Jefe inmediato : C:ModificarEvaluacion : ControlEvaluacion : Trabajadores : Persona : I: EvaluacionAdjestrado : I: EvaluacionProfesor : I: EvaluacionTecnico : I: ResultadoEvaluacion : C:ModificarEvaluacion : C:ModeloEvaluacion



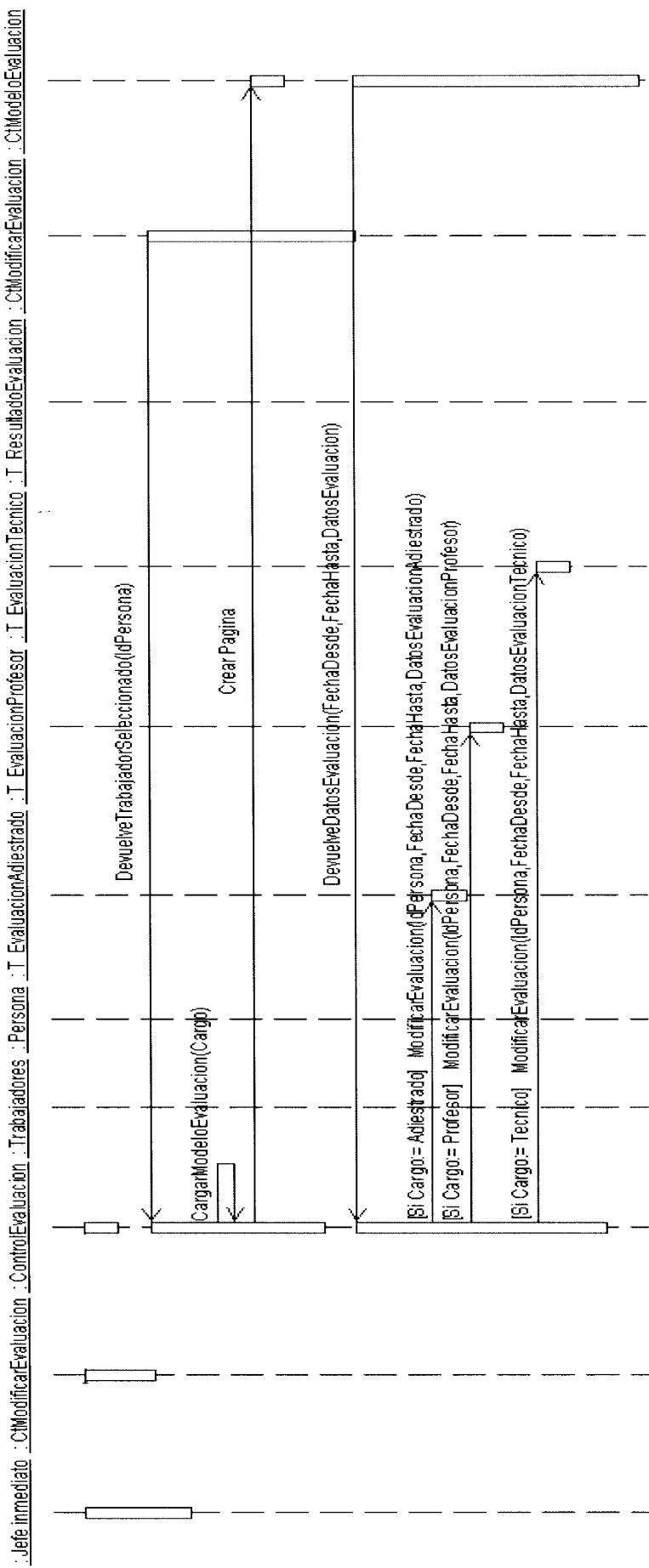
Modificar evaluación (b) continua.



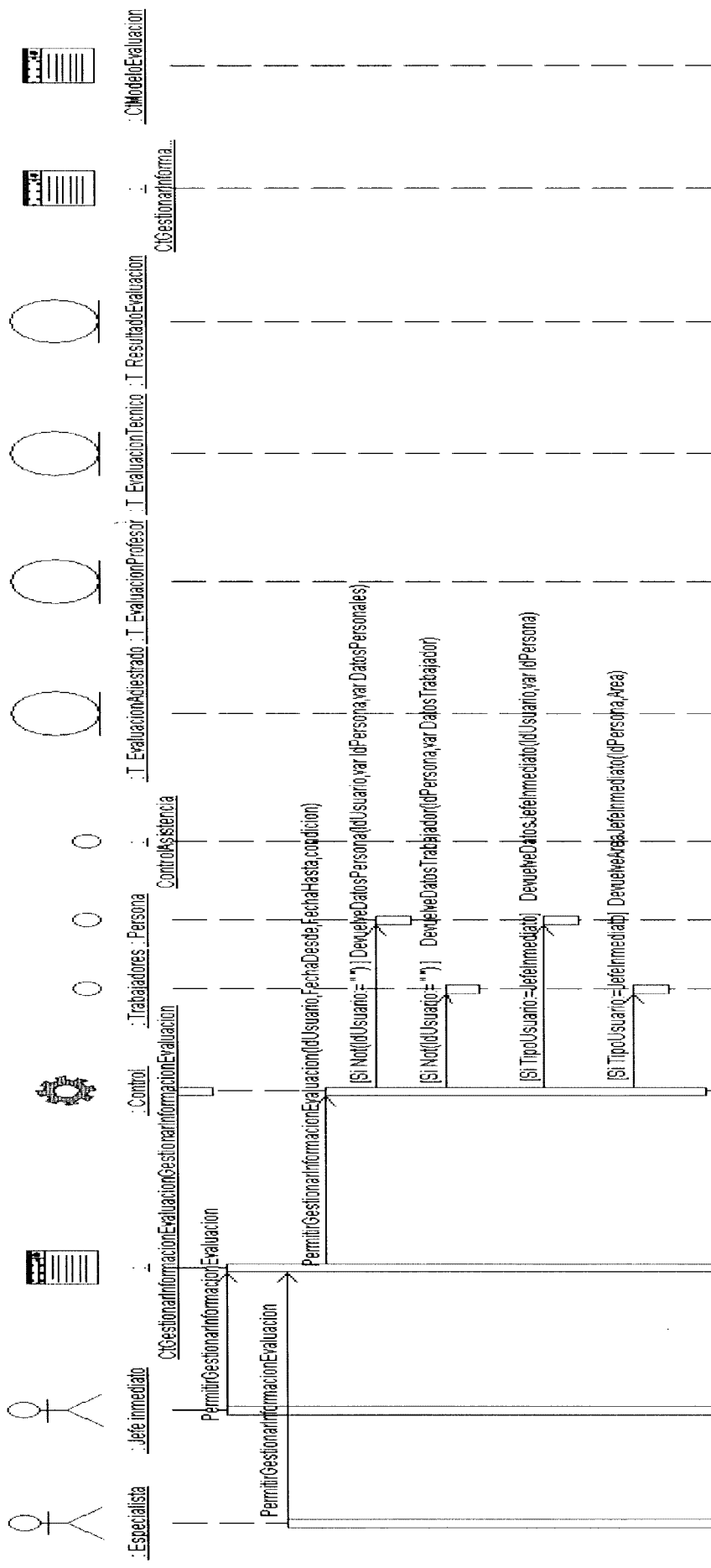
.:Jefe inmediato : CModificarEvaluacion : ControlEvaluacion : Trabajadores : Persona : T EvaluacionAdiestrado : T EvaluacionProfesor : T EvaluacionTecnico : T ResultadoEvaluacion : CModificarEvaluacion : CModificarEvaluacion : CModeloEvaluacion



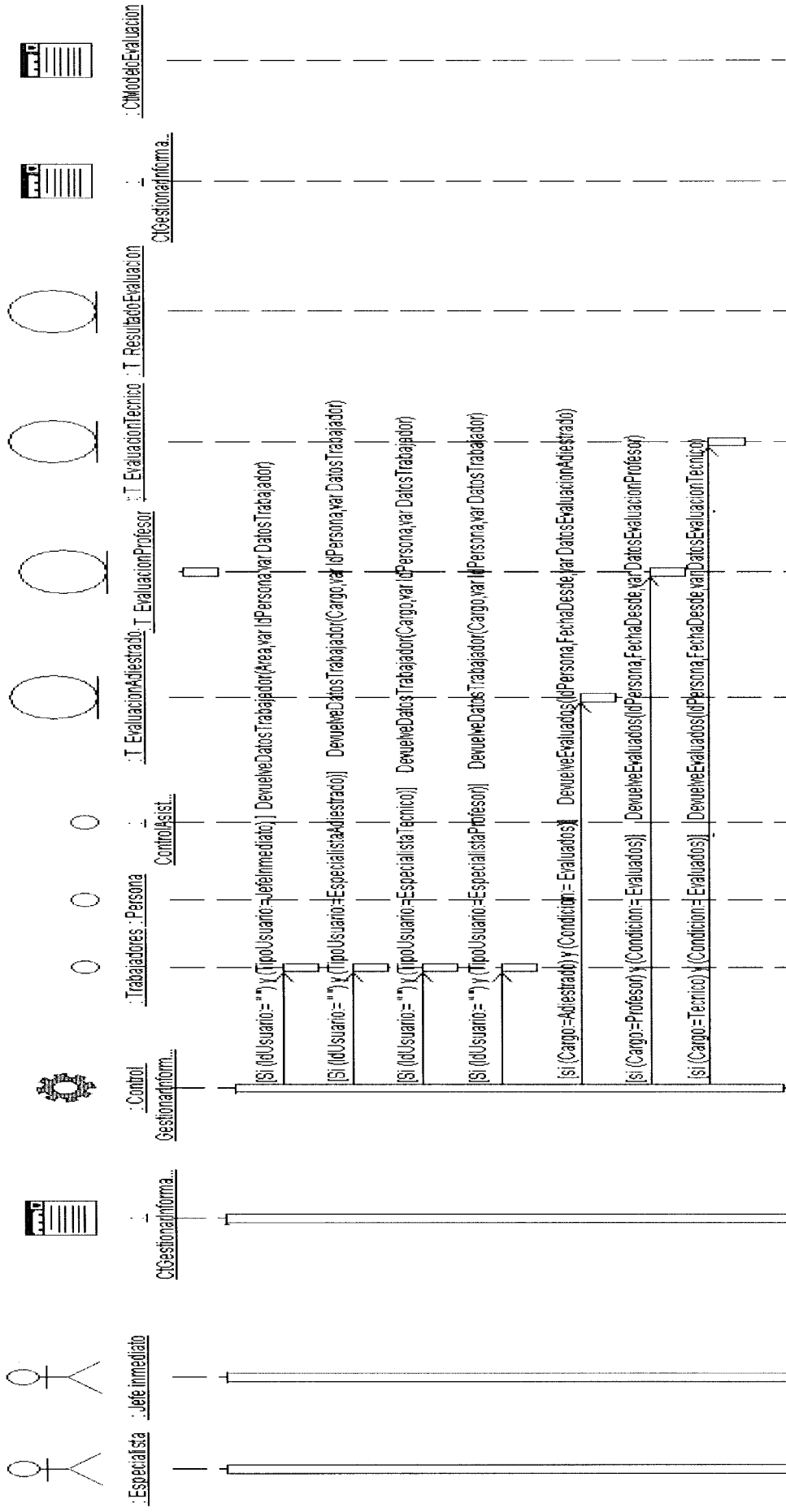
Modificar evaluación (c).



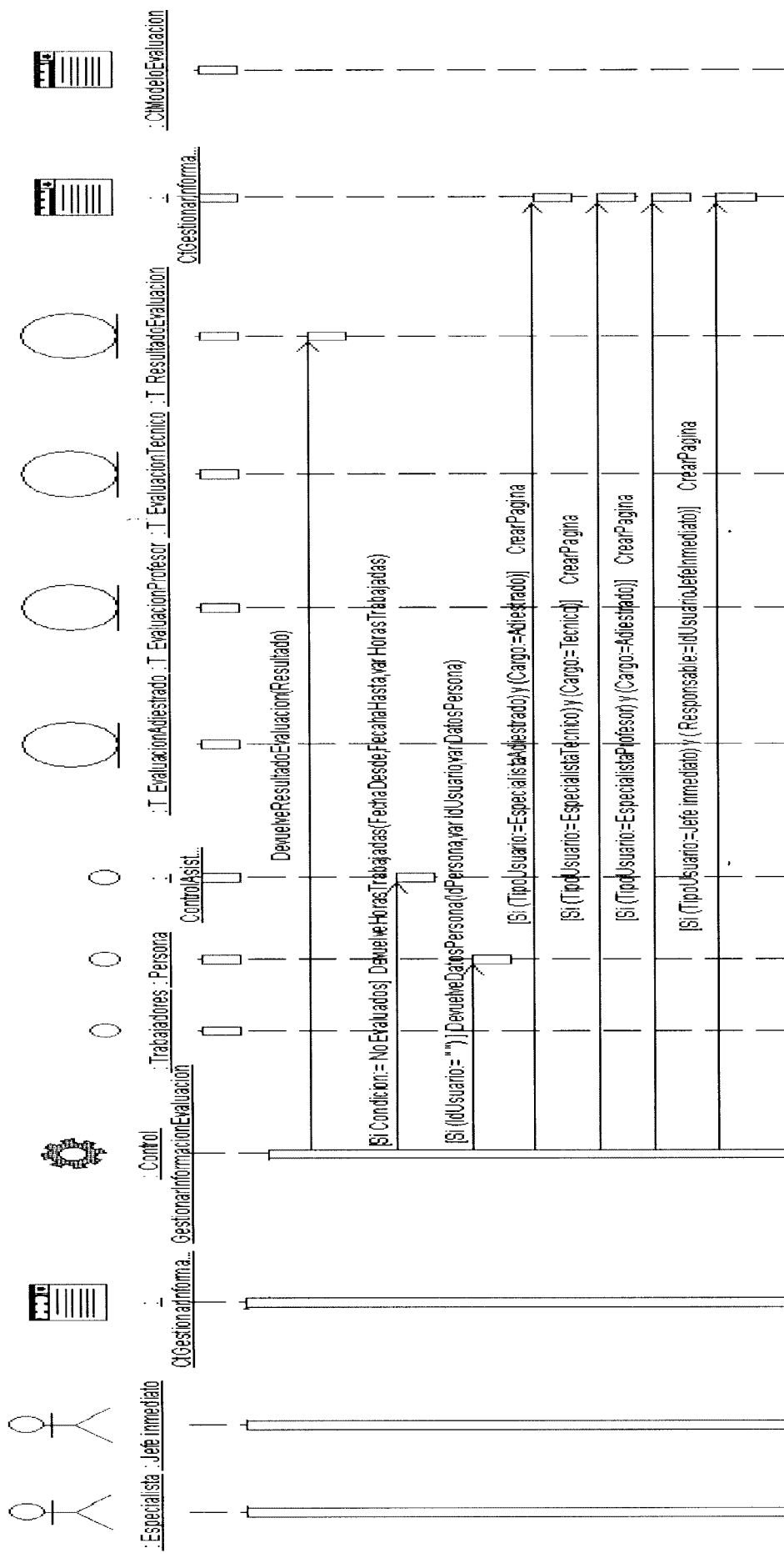
Gestionar Información Evaluación(a) continua.



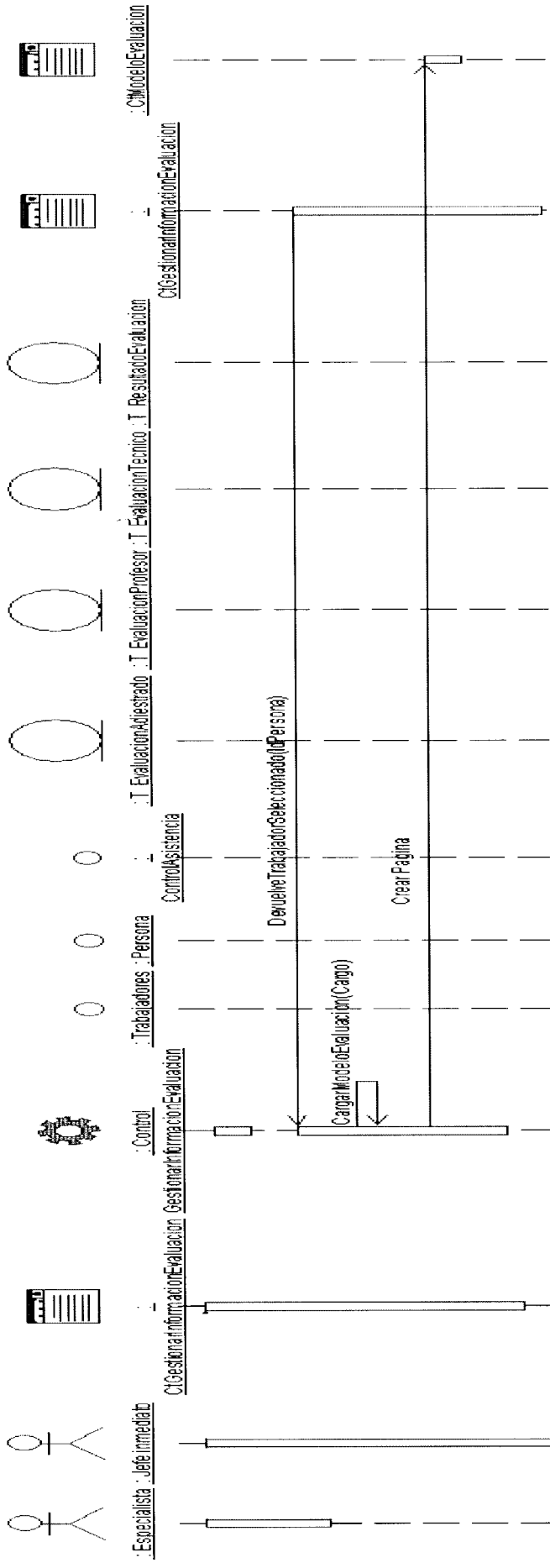
Gestionar Información Evaluación (b) continua.



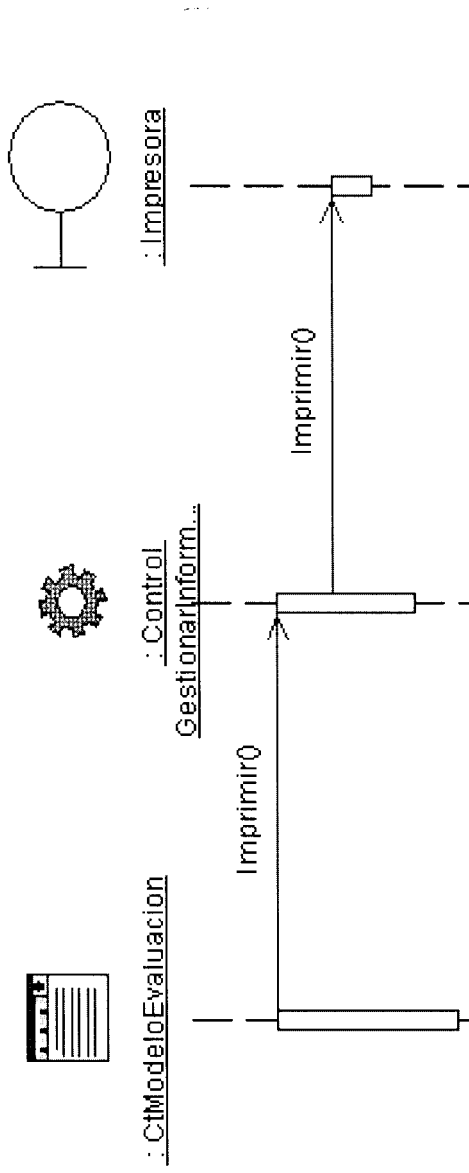
Gestionar Información Evaluación (c) continua.



Gestionar Información Evaluación (d).



Imprimir.



Glosario de Términos

A

Administración de un sistema: Es el rol que juega una o varias personas de controlar un sistema en cuanto a seguridad y configuración, además de chequear el buen funcionamiento del mismo.

D

Desempeño: Acción y efecto de desempeñar o desempeñarse.

E

Estructura cliente / servidor: Consiste en que los clientes piden que una tarea sea realizada y el servidor realiza dicha tarea y regresa la información al cliente a través de la red.

H

Herramienta Case: Ingeniería de sistemas asistida por ordenador (Computer-Aided Systems Engineering - CASE) es la aplicación de tecnología informática a las actividades, las técnicas y las metodologías propias de desarrollo de sistemas. Su objetivo es automatizar o apoyar una o más fases del ciclo de vida del desarrollo de sistemas.

HTTP: Protocolo usado para la transferencia de documentos WWW. Estas transferencias requieren un programa cliente http en un extremo de la comunicación y un servidor http en el otro.

I

Interfaz Web: Intranet: Red privada, desarrollada dentro de una compañía que utiliza el mismo software y provee de información similar que Internet, solo que es únicamente para el uso interno.

P

Parámetros de un sistema: Son un conjunto de variables que se utilizan para medir o determinar el comportamiento de un sistema, siendo influyentes en su dinamismo.

Protocolo FTP: Protocolo de transmisión de ficheros.

R

Red local: Se trata de una red de comunicación de datos geográficamente limitada, por ejemplo, una empresa.

Reporte: Informe detallado sobre alguna información, o sobre el estado de la información.