



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTADO DE CIENCIAS

**"SISTEMA (CHILAM) PARA LA APLICACIÓN DE EVALUACIONES
"EVALPROSOFT"**

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN

P R E S E N T A:

MAURICIO MORGADO CASTILLO

Tutora: DRA. HANNA OKTABA



2006



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

FACULTAD DE CIENCIAS



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

División de Estudios Profesionales

ACT. MAURICIO AGUILAR GONZÁLEZ
Jefe de la División de Estudios Profesionales
Facultad de Ciencias
P r e s e n t e .

Por este medio hacemos de su conocimiento que hemos revisado el trabajo escrito titulado:

"Sistema (Chilam) para la aplicación de evaluaciones "EvalProSoft"

realizado por **Morgado Castillo Mauricio**

con número de cuenta **09612995-9**, quien cubrió los créditos de la licenciatura en **Ciencias de la Computación**. Dicho trabajo cuenta con nuestro voto aprobatorio.

Tutor(a)

Propietario

Dra.

Hanna Oktaba

Propietario

M. en C.

María Guadalupe Elena Ibarguengoitia González

Propietario

Ing.

Luz María Castañeda de León

Suplente

M. en C.

Claudia Alquicira Esquivel

Suplente

M. en C.

José de Jesús Galaviz Casas

Atentamente
"POR MI RAZA HABLARÁ EL ESPÍRITU"
Ciudad Universitaria, D.F., a 30 de agosto del 2006.
**EL COORDINADOR DEL COMITÉ DE TITULACIÓN
DE LA LICENCIATURA EN CIENCIAS DE LA COMPUTACIÓN**

M. EN C. JOSÉ DE JESÚS GALAVIZ CASAS

FACULTAD DE CIENCIAS

Señor sinodal: antes de firmar este documento, solicite al estudiante que le muestre la versión digital de su trabajo y verifique que la misma incluya todas las observaciones y correcciones que usted hizo sobre el mismo.

MATEMÁTICAS

A mis padres Aide y Ezequiel, que me han hecho lo que soy
y me han enseñado lo mejor de ellos.

A mis hermana Berenice que me apoyo
cuando mas lo necesite.

A mi hermana Angélica que me
ha enseñado a ser feliz.

A Mario García que me ha enseñado
lo mejor de mi y de la carrera.

A mis amigos de DGSCA que han
estado conmigo.

A mis amigos de la facultad que me
ayudaron en la carrera.

A mis profesores que me
enseñaron tanto.

A todos las personas que me ayudaron
a realizar este trabajo.

Índice

1.- INTRODUCCIÓN.....	2
2.- CONTEXTO DEL PROBLEMA.....	3
2.1.- MOPROSOFT.....	3
2.2.- EVALPROSOFT.....	4
3.- REQUERIMIENTOS.....	9
3.1.- REQUERIMIENTOS FUNCIONALES.....	9
3.2.- REQUERIMIENTOS NO FUNCIONALES.....	9
3.3.- CASOS DE USO.....	10
4.- DISEÑO DEL SISTEMA.....	16
4.1.- DIAGRAMA DE PAQUETES.....	18
4.2.- DIAGRAMA DE CLASES.....	18
4.3.- DIAGRAMA DE COLABORACIÓN.....	28
4.4.- DIAGRAMAS DE SECUENCIA.....	32
4.5.- DIAGRAMAS DE COMPONENTES.....	49
5.- DETALLE DE LAS CAPAS DE LA APLICACIÓN.....	52
6.- TECNOLOGÍAS UTILIZADAS EN EL DESARROLLO.....	68
7.- PRUEBAS DEL SISTEMA.....	70
8.- CONCLUSIONES.....	72
9.- REFERENCIAS.....	73

1.- INTRODUCCIÓN

Chilam es un programa que sirve como herramienta para el desarrollo del proceso de evaluación EvalProSoft, ya que sintetiza las acciones que se llevan a cabo, como es la recopilación de datos de la empresa y los datos del cuestionario de evaluación. Actualmente los evaluadores capturan la información en plantillas de documentos utilizando procesadores de texto, lo que facilita muchas de las tareas de los evaluadores, como es la generación de los documentos durante la ejecución de la evaluación, capturar la información necesaria etc., pero al mismo tiempo implica nuevos problemas, como es el difícil manejo de la información, ya que no puede ser referenciada muy específicamente. Uno de los principales problemas de conservar la información radica en que no puede ser seccionada de manera sencilla, y a pesar de las ventajas de un procesador de texto no es una herramienta que este diseñada para ayudar al proceso de evaluación de procesos de desarrollo de software.

Chilam esta desarrollado con el lenguaje de programación java ya que este lenguaje posee una buena abstracción de objetos, los programas desarrollados con este lenguaje pueden correr en diversas plataformas, existen bibliotecas en la red para diversos propósitos escritas en este lenguaje.

Chilam provee los mecanismos para poder gestionar la información que se requiere durante el proceso de evaluación de manera organizada y segura. La información se almacena en documentos XML (eXtended Markup Language) que seccionan la información del proceso de evaluación permitiendo manipular la información que se desee sin necesidad de contar con programas especiales para mostrar el contenido. Además, XML facilita el cambio de formato de la información mediante el uso de tecnologías que explotan la estructura bien definida de los documentos como es XPath, XQuery, XSL ó permite almacenar la información en una base de datos no relacional como eXist. De este modo se almacena la información que se requiere, pero al mismo tiempo se prepara la información para que sea utilizada como se necesite.

Chilam facilita el proceso de evaluación, ya que una vez que se ha calificado el cuestionario de evaluación se tendría que estar navegando exhaustivamente el documento de Word para poder calificar el cuestionario usando la plantilla. Con Chilam, basta con indicar al sistema que califique el cuestionario para que en un sólo barrido de la información ingresada por el usuario se obtengan las calificaciones y el tiempo que se invirtió en la aplicación de los cuestionarios, permitiendo disminuir y contabilizar el rendimiento del equipo de evaluación así como el tiempo de la empresa empleado para evaluarse.

A partir de la información capturada por el usuario, Chilam es capaz de proveer mecanismos de seguridad para cifrar y autorizar el manejo de la información, ya que este sistema utiliza la contraseña que le es requerida al evaluador competente para cifrar la información que se va recabando durante el proceso de evaluación. Provee el mecanismo para que la corroboración de la información recopilada, sea autorizada por el evaluador competente durante la aplicación del cuestionario de evaluación y provee el mecanismo para que un cambio en los acuerdos de la evaluación tenga que ser autorizado por el evaluador competente, el facilitador y el representante de la empresa, procurando hacer un respaldo de la información que se tenía antes de aplicar el cambio, lo que asegura que siempre se contará con la información segura.

La arquitectura de Chilam está basada en el modelo de 3 capas y la codificación de las clases que son necesarias en el sistema van de acuerdo con este modelo, por lo que está diseñado para que pueda ser mejorado en versiones posteriores de manera localizada y sencilla, ya que todas las clases del sistema cuentan con la documentación propia que permite conocer la descripción de cada método y variable que sea importante en la clase.

Chilam forma parte de un proyecto conformado por 2 secciones llamando Chilam – Balam. Balam por otra parte, es un portal web que permite difundir información importante sobre MoProSoft, agendar evaluaciones etc. En particular Balam gestiona los resultados de las evaluaciones aplicadas a las empresas, estos resultados son proporcionados por Chilam.

2.- CONTEXTO DEL PROBLEMA

2.1.- MoProSoft (Modelo de procesos para la industria de software)

MoProSoft es un modelo de procesos para la industria nacional que desarrolla software, fomenta la estandarización de su operación mediante la incorporación de prácticas en gestión e ingeniería de software que regulan, documentan y mejoran la calidad de los productos. La adopción del modelo permite elevar la capacidad de las organizaciones que desarrollan o mantienen software para ofrecer servicios con calidad lo que les permite alcanzar niveles internacionales de competitividad. Este modelo es también aplicable en áreas internas de desarrollo de software de las empresas de diversos giros¹.

MoProSoft consta de los siguientes procesos:

Número	Abreviatura	Nombre del proceso
1	GN	Gestión de N egocios
2	GPR	Gestión de PR ocesos
3	GR	Gestión de R ecursos
4	RHAT	R ecursos H umanos y A mbiente de T rabajo
5	BSI	B ienes, S ervicios e I nfraestructura
6	CO	C onocimiento de la O rganización
7	GPY	Gestión de ProY ectos
8	APE	A ministración de ProY ectos E specíficos
9	DM	D esarrollo y M antenimiento

Tabla 1.1
Procesos que conforman el modelo de desarrollo de MoProSoft

Nivel	Capacidad del proceso
1	Realizado
2	Administrado
3	Establecido
4	Predecible
5	Optimizado

Tabla 1.2
Niveles de capacidades que puede contener cada proceso a evaluar.

Características de MoProSoft

- Es específico para el desarrollo y mantenimiento de software.
- Es sencillo de entender y adoptar.
- Facilita el cumplimiento de los requisitos de otros modelos como ISO 9000:2000, CMM y CMMI.
- Se enfoca a procesos.
- Se le considera práctico en su aplicación, principalmente en organizaciones pequeñas, con bajos niveles de madurez.
- Comprende un documento de menos de 200 páginas que, al compararlo con otros modelos y estándares, lo hace bastante práctico.
- Resulta acorde con la estructura de las organizaciones mexicanas de la industria de software.
- Está orientado a mejorar los procesos, para contribuir a los objetivos de negocio, y no simplemente ser un marco de referencia o certificación.
- Tiene un bajo costo, tanto para su adopción como para su evaluación.

MoProSoft es un modelo integrado donde las salidas de un proceso están claramente dirigidas como entradas a otros; las prácticas de planeación, seguimiento y evaluación se incluyen en todos los procesos de gestión y administración; por su parte los objetivos, los indicadores, las mediciones y las metas cuantitativas fueron incorporados de manera congruente en todos los procesos; las verificaciones, validaciones y pruebas están incluidas de manera explícita dentro de las actividades de los procesos; y existe una base de conocimientos que resguarda todos los documentos y productos generados.

Cada proceso puede contener a su vez 5 niveles de capacidades, un nivel (Tabla 1.2) está conformado por lo menos por un atributo de proceso y puede contener un producto de trabajo específico al nivel.

Una vez que la empresa que desarrolla software ha decidido adoptar MoProSoft como un modelo de procesos para mejorar la calidad de sus productos y ha logrado incorporar los procesos, se encuentra en condiciones de poder ser evaluada para medir la efectividad y capacidad de sus procesos.

2.2.- EvalProSoft

EvalProSoft es el método de evaluación para conocer el estado actual de las prácticas de la organización con respecto a MoProSoft.

El propósito del método de evaluación de procesos EvalProSoft para la industria de software es otorgar a la organización solicitante un perfil del nivel de capacidad de los procesos implantados en la organización y un nivel de madurez de capacidades, lo que a grandes rasgos permite a las empresas identificar las tareas que no se han cumplido en su totalidad, así como diseñar la incorporación de las nuevas tareas y procesos a implementar para continuar con la mejora del proceso de desarrollo del software.

La continua implantación de MoProSoft para organizar la producción del software permite a las empresas mejorar la calidad de sus productos, así como mantener un control y seguimiento de los productos ya desarrollados a través de evaluaciones, que permitan a la empresa identificar áreas de oportunidad y fortalezas.

Para poder llevar a cabo este proceso, EvalProSoft define roles y responsabilidades específicos para las personas involucradas (Tabla 1.4).

Rol	Responsabilidad
Promotor	<p><i>El Promotor es responsable de:</i></p> <ul style="list-style-type: none"> ■ verificar que el responsable de la evaluación es un Evaluador Competente; ■ asegurar la disponibilidad de los recursos humanos y materiales para conducir la evaluación; ■ acordar con el Evaluador Competente el Acuerdo de Evaluación; ■ solicitar al Organismo Rector el Paquete de Evaluación; ■ participar en la reunión de inicio de la evaluación; ■ estar disponible durante la evaluación para resolver problemas o conflictos que surjan durante la evaluación; ■ asistir a la reunión de la presentación de resultados; ■ recibir el Reporte de Resultados; ■ responder la Encuesta de Satisfacción.
Evaluador Competente	<p><i>El Evaluador Competente es responsable de:</i></p> <ul style="list-style-type: none"> ■ acordar con el Promotor el Acuerdo de Evaluación; ■ confirmar el compromiso del Promotor para realizar la evaluación; ■ elaborar el Plan de Evaluación con la colaboración del Representante de la Organización; ■ asegurar que la evaluación se conduce de acuerdo con los requisitos del método de evaluación EvalProSoft; ■ asegurar que los Participantes de la Evaluación están informados sobre el propósito, el alcance y el método de la evaluación; ■ asegurar que los miembros del Equipo de Evaluación tienen los conocimientos y las habilidades apropiadas para sus roles y para usar las herramientas seleccionadas para apoyar la evaluación; ■ asegurar que los miembros del Equipo de Evaluación tienen acceso a una guía documentada apropiada que contenga la información de como desempeñar las actividades definidas en la evaluación; ■ entregar el Reporte de Resultados al Promotor; ■ realizar las actividades de cierre de la Evaluación. En esta actividad se elabora el Reporte Estadístico, el cual se entrega al Organismo Rector; ■ aplicar y entregar las Encuestas de Satisfacción.
Representante de la Organización*	<p><i>El Representante de la Organización es responsable de:</i></p> <ul style="list-style-type: none"> ■ representar a la organización dentro del Equipo de Evaluación; ■ colaborar en la elaboración del Plan de Evaluación. <p><i>*Las responsabilidades del Representante de la Organización y del Facilitador pueden ser realizadas por una misma persona</i></p>
Facilitador	<p><i>El Facilitador de la organización es responsable de:</i></p> <ul style="list-style-type: none"> ■ asegurar la disponibilidad de la información, infraestructura, logística, y participación de los miembros de la organización
Equipo de Evaluación	<p><i>El Equipo de Evaluación es responsable de:</i></p> <ul style="list-style-type: none"> ■ realizar la evaluación de acuerdo con el método de evaluación EvalProSoft; ■ comprometerse a no realizar ninguna actividad en paralelo con la ejecución de la evaluación; ■ responder la Encuesta de Satisfacción, excepto el Evaluador Competente.
Participante de la Evaluación	<p><i>El Participante de la Evaluación es responsable de:</i></p> <ul style="list-style-type: none"> ■ asistir a las entrevistas y reuniones planeadas en la evaluación; ■ proporcionar la información verídica, tanto oral como documental, solicitada por el Equipo de Evaluación.

Tabla 1.3
Relación rol - responsabilidad.

EvalProSoft permite rastrear cada una de las tareas de los 9 procesos mostrados en la tabla 1.1, para ello es necesario seguir una organización que permita tanto a la empresa que es evaluada como al equipo evaluador identificar y evaluar cada una de las tareas, así como brindar mayor información a la empresa evaluada para que mejore los procesos definidos en el alcance de la evaluación. El Acuerdo de confidencialidad es el documento que firman todos los involucrados en la evaluación y que define:

- 1) Alcance de la evaluación.
- 2) Organigrama de la organización.
- 3) Unidades administrativas a evaluar.
- 4) Responsables por rol.
- 5) Restricciones.
- 6) Procedimiento de Control de Cambios.
- 7) Acuerdo de confidencialidad.

El alcance de la evaluación es una sección del Acuerdo de confidencialidad que detalla el número de procesos y niveles de capacidades que se tomarán en cuenta para la evaluación.

Cada una de estas secciones del documento están definidas por EvalProSoft.

Para llevar a cabo la evaluación se requiere de un equipo de evaluación que se encarga de recabar la información para la planeación, ejecución y entrega de resultados de la evaluación. Este equipo estará dirigido por un evaluador competente que se encarga de orientar los esfuerzos del equipo y es responsable de la evaluación ante el Organismo verificador.

Una vez que la empresa decidió evaluarse se acuerda el siguiente flujo de trabajo:

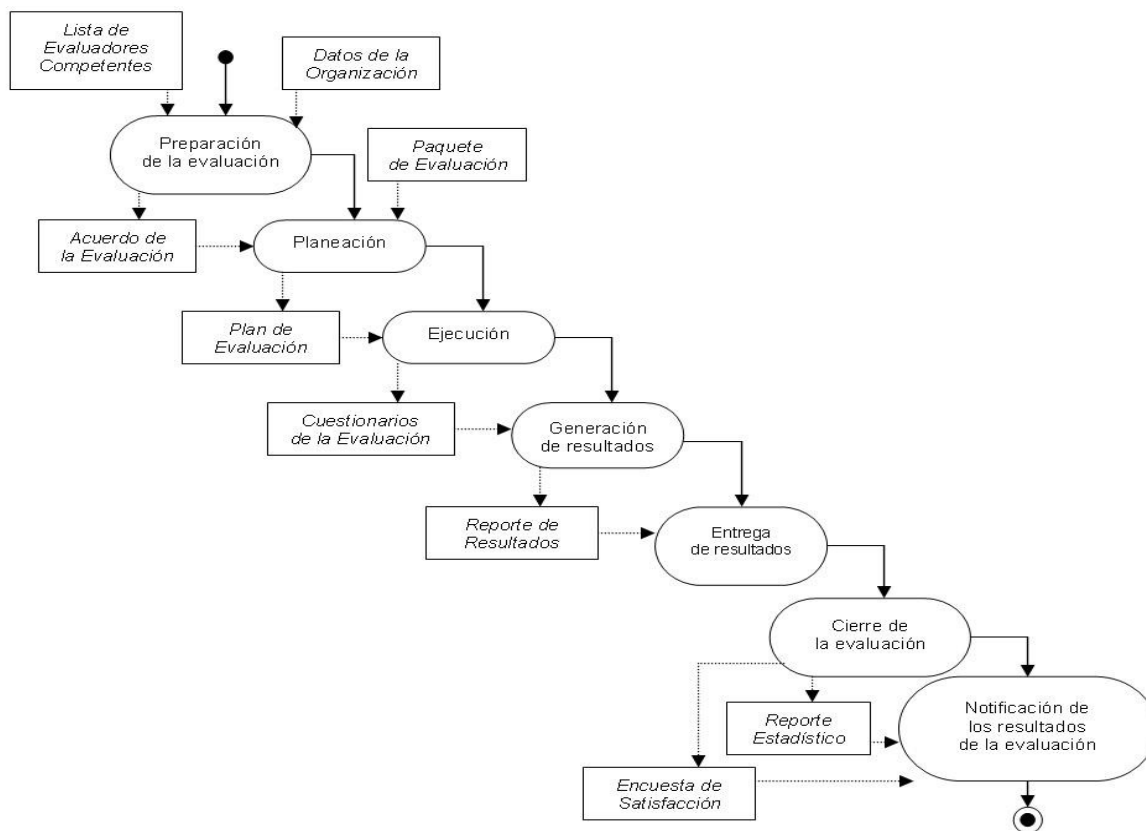


Figura 2.1
Diagrama de flujo del proceso de evaluación.

Cada etapa mostrada en la figura 1.2 requiere información específica para el proceso definido por EvalProSoft y como resultado de la ejecución de la fase se obtiene información necesaria para la siguiente etapa.

Durante la actividad de Preparación, el Promotor selecciona a un Evaluador Competente de la Lista de Evaluadores Competentes proporcionada por el Organismo Verificador. Con base en los Datos de la Organización, el Evaluador Competente en conjunto con el Promotor acuerdan, elaboran y firman el Acuerdo de la Evaluación. Finalmente, el Promotor solicita el Paquete de Evaluación al Organismo Verificador.

Durante la actividad de Planeación, el Evaluador Competente confirma el compromiso con el Promotor para realizar la Evaluación, identifica los proyectos a evaluar y a los participantes en la Evaluación, elabora el Plan de Evaluación y lo valida con el Promotor.

Durante la actividad de Ejecución, el Evaluador Competente prepara al Equipo de Evaluación y preside la reunión de inicio. El Equipo de Evaluación, para cada proyecto a evaluar, revisa la documentación del proyecto, prepara y realiza las entrevistas con el Responsable de la Administración del Proyecto Específico y con su equipo de trabajo.

Adicionalmente, el Equipo de Evaluación, para cada proceso de Alta Dirección y Gestión, revisa la documentación, prepara y realiza las entrevistas con cada uno de los responsables de estos procesos. La información recopilada se registra como evidencia documental y oral en los Cuestionarios de la Evaluación. Finalmente, se consolida y se corrobora esta información.

Durante la actividad de Generación de Resultados, el Equipo de Evaluación obtiene la tabla de perfiles de calificaciones de atributos, a partir de la cual genera el perfil del nivel de capacidad de los procesos implantados y el nivel de madurez de capacidades. Con base en éstos, elabora el Reporte de Resultados.

Durante la actividad de Entrega de Resultados, el Evaluador Competente presenta a la organización los resultados obtenidos y entrega el Reporte de Resultados al Promotor.

Durante la actividad de Cierre de la Evaluación, el Evaluador Competente genera el Reporte Estadístico, aplica las Encuestas de Satisfacción y los envía al Organismo Verificador. Entrega a la organización la información y los productos generados durante la evaluación.

Durante la actividad de Notificación de los Resultados de la evaluación, el Organismo Verificador revisa y verifica el reporte y encuestas recibidos. Registra y notifica los resultados de la evaluación.

Para evaluar MoProSoft, se ha diseñado el cuestionario de evaluación que contiene los procesos de MoProSoft que la empresa ha decidido evaluar (cabe señalar que cada proceso es evaluado hasta el nivel que se acuerda pero cada nivel es creciente, es decir, si se decide evaluar el nivel 3, por robustez del modelo se tendrán que evaluar el nivel 1 y el nivel 2). Cada tarea (producto de trabajo esperado o práctica base esperada) especificada en cada proceso tiene asociada una calificación que en conjunto definen la calificación del atributo de proceso o el producto de trabajo específico. El conjunto de las calificaciones de los atributos de proceso y los atributos de trabajo específico conforman la calificación del nivel y la calificación de un proceso está definida por las calificaciones del nivel.

Ejecución de la evaluación

Para evaluar un proceso hace uso de una serie de preguntas guía, información recopilada durante la entrevista, prácticas base esperadas y revisión de productos de trabajo esperados.

Para evaluar un producto de trabajo específico al nivel se hace uso de una serie de preguntas guía, información recopilada durante la entrevista y productos de trabajo esperados.

A cada una de las tareas definidas por los procesos del cuestionario de evaluación, le es asignada una calificación que mas adelante es utilizada para calificar el cuestionario de evaluación y obtener el nivel de capacidad de los procesos. La calificación que le puede ser asignada es:

Calificación	Significado	Valor numérico
N	No alcanzado	0
P	Parcialmente	1
A	Ampliamente	2
C	Completamente	3

Tabla 1.4
Calificaciones asignables.

EvalProSoft especifica como se debe de calificar cada proceso de la evaluación, para que se puedan obtener el perfil de la empresa evaluada.

En caso de que la evaluación requiera ser modificada es necesario llevar el procedimiento de control de cambios, procedimiento definido por EvalProSoft que permite adecuar las evaluaciones según las necesidades lo requieran. Este procedimiento permite alterar los datos iniciales de la evaluación como es el cuestionario de evaluación, los participantes de la evaluación, las fechas de término etc.

Para que este procedimiento se pueda llevar a cabo, es necesario contar con la aprobación del representante de la empresa, el evaluador competente y el promotor de la evaluación.

El objetivo de Chilam es contar con un sistema que sintetice las acciones que se llevan a cabo, como es la recopilación de datos de la empresa y los datos del cuestionario de evaluación que se generan durante la evaluación para reducir el tiempo y costo invertido durante la evaluación.

3.- REQUERIMIENTOS

El primer paso para desarrollar un sistema es la especificación de los requerimientos de software, es decir, se describe detalladamente el software a ser escrito procurando establecer en la medida de lo posible cada una de las tareas que serán implementadas en el sistema con la mayor precisión.

La herramienta para el proceso de aplicación de evaluaciones EvalProSoft, esta conformada por dos grandes componentes.

El primer componente denominado Chilam se encargará de facilitar la aplicación de los cuestionarios de evaluación a las empresas que así lo soliciten.

Las principales características de Chilam serán la captura y edición de la información de las fases más importantes de la evaluación, la información almacenada deberá permanecer segura y se generará los resultados de la evaluación conforme a lo especificado por EvalProSoft.

El segundo componente denominado Balam será un portal web que procurará la explotación de los datos recabados durante las evaluaciones y será desarrollado posteriormente.

El objetivo de estas 2 herramientas es disminuir el costo que toma llevar a cabo una evaluación.

Los requerimientos descritos a continuación fueron obtenidos mediante el análisis de las tareas que realizan los evaluadores durante la ejecución de la evaluación, la especificación del proceso de evaluación y las necesidades detectadas por los evaluadores.

Los requerimientos no funcionales fueron definidos en el orden en que la ejecución de la evaluación se lleva a cabo.

3.1 Requerimientos no funcionales

Requerimientos de software de Chilam

1. Máquina con [java jre](#) (java runtime environment) versión 1.5.X

Requerimientos de hardware de Chilam

1. Procesador mayor a 1Ghz.
2. Memoria de 512 Mb.
3. Espacio para almacenar al menos 500 Mb.

Capacidades no funcionales de Chilam:

- 1.- Programa ejecutable bajo la plataforma de windows.
- 2.- Programa desarrollado con el lenguaje java.
- 3.- La interfaz (IDE) independiente (Stand alone) que será capaz de correr sin necesidad de una conexión de red.
- 4.- Programa contará con un instalador que automatice el proceso de configuración.
- 5.- Conservar la información segura.

3.2 Requerimientos funcionales

- 1.- Capturar la información inicial.
 - 1.1.- Capturar los datos de la organización.
 - 1.2.- Capturar el acuerdo de evaluación.
 - 1.3.- Generar el acuerdo de evaluación en formato pdf.
 - 1.4.- Generar un documento que resuma la información guardada en esta sección en formato pdf.
- 2.- Capturar y editar el cuestionario de evaluación acordado.
 - 2.1.- Editar la información de los procesos.
 - 2.1.1.- Manejar sesiones de entrevistas por cada proceso, las cuales deben de estar conformadas por:
 - Evaluador que entrevista.
 - Participante de la evaluación que es entrevistado.

- Fecha y hora en que inicia la entrevista.
 - Fecha y hora en que termina la entrevista.
- 2.1.2.- Corroborar la información recopilada así como indicar al sistema cuando se puede corroborar la información recopilada, Esta corroboración solo podrá ser ejecutada por el evaluador competente (rol definido en el proceso de evaluación por EvalProSoft)
- 2.1.3.- Acceder a la información recopilada corroborada (evidencia) para ser asignada a las prácticas base y utilizada como referencia en la sección de "hallazgos".
- 3.- Calificar el cuestionario de evaluación.
- 4.- Visualizar los resultados de la evaluación.
- 5.- Generar el reporte para la empresa evaluada y el reporte para el Organismo verificador.
- 5.1.- Anexar el tiempo estimado empleado en las entrevistas y las tareas no satisfechas en su totalidad en el reporte de resultados.
- 6.- Proveer el procedimiento para el control de cambios.

3.3.- Casos de uso

El diagrama de casos de uso representa la forma en como un usuario (Actor) interactúa con el sistema en desarrollo, ya que representa la funcionalidad que ofrece el sistema.

En el sistema Chilam el usuario puede realizar seis tareas grandes, que describen el comportamiento general del sistema. Al mismo tiempo que los casos de uso describen como el usuario y Chilam se coordinarán para lograr las tareas (Figura 3.1).

Para cada uno de los seis requerimientos funcionales del sistema especificados anteriormente, se realiza un diagrama general de caso de uso para detallar las subtareas que se tienen que realizar para poder llevar a cabo cada funcionalidad, así como los actores que intervienen en el proceso.

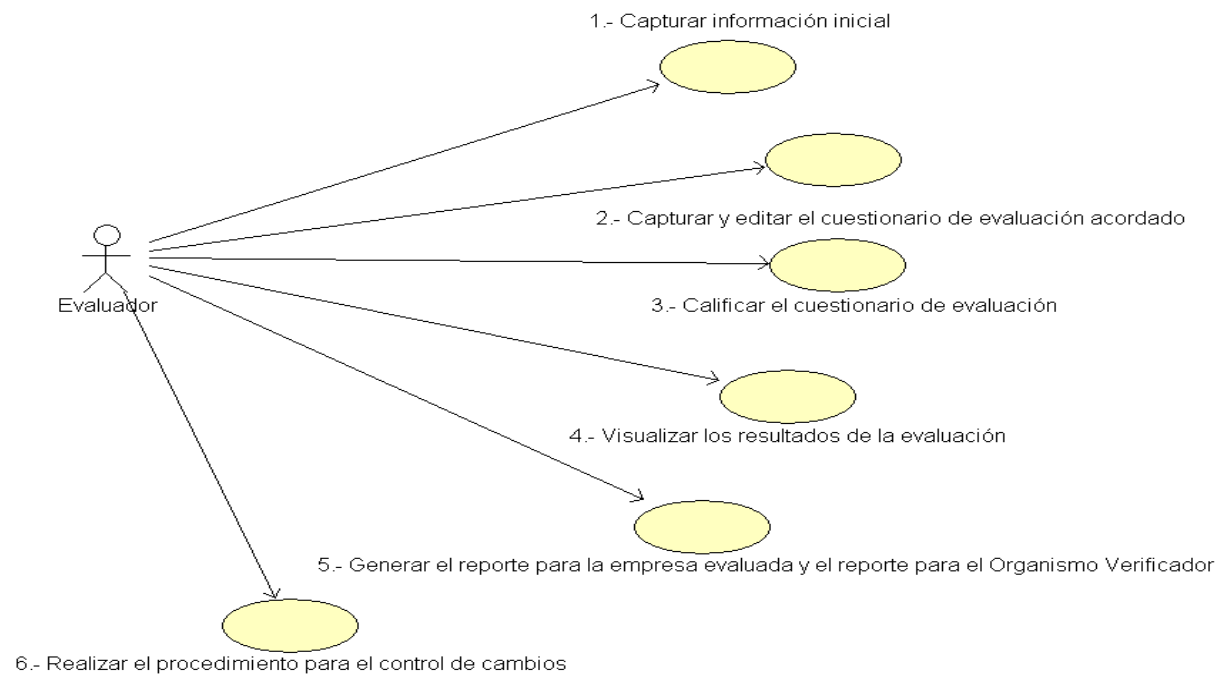


Figura 3.1
Diagrama general de los casos de uso resultantes de la especificación de requerimientos.

Caso de uso 1 (Capturar información inicial)

La información inicial que se requiere para poder comenzar con el proceso de evaluación está conformada por:

1.- Los datos de la organización.

Esta información esta conformada por:

- 1) Datos propios de la empresa que será evaluada.
- 2) Datos de los participantes en la evaluación.
- 3) Fechas que durará el proceso de evaluación.
- 4) Documentos de la organización que son requeridos por el proceso.
- 5) Datos del facilitador, representante, evaluador, promotor y gestor.
- 6) Datos del acuerdo de evaluación.
- 7) Datos del equipo de evaluadores.
- 8) Relación con MoProSoft

2.- Los datos del acuerdo de evaluación

La empresa y el equipo de evaluación acuerdan el número de procesos que se tomarán en cuenta durante la evaluación así como el nivel de los procesos seleccionados. Esta información conforma el acuerdo de evaluación, información vital para el proceso de evaluación.

Con la información capturada de los puntos 1 y 2, se tiene que imprimir por duplicado el documento que firmarán cada uno de los involucrados en la evaluación y que sirve para formalizar el acuerdo de la evaluación, por lo tanto este documento es escrito a partir de la plantilla especificada por EvalProSoft anexando la información particular de la empresa y los individuos que intervienen en la evaluación en formato pdf.

El acuerdo de evaluación forma parte del acuerdo de confidencialidad que es generado formato pdf, ya que este formato permite restringir la lectura mediante la petición de una contraseña.

Para contar con un respaldo de la información que describe las condiciones iniciales del proceso de la evaluación, es necesario generar un respaldo en formato pdf, este documento recopila la información capturada mediante la interfaz, es decir, no está basado en una plantilla.

Precondiciones	Post condiciones
Ninguna	Los datos de la organización y el acuerdo de evaluación seleccionados y escritos en documentos pdf.

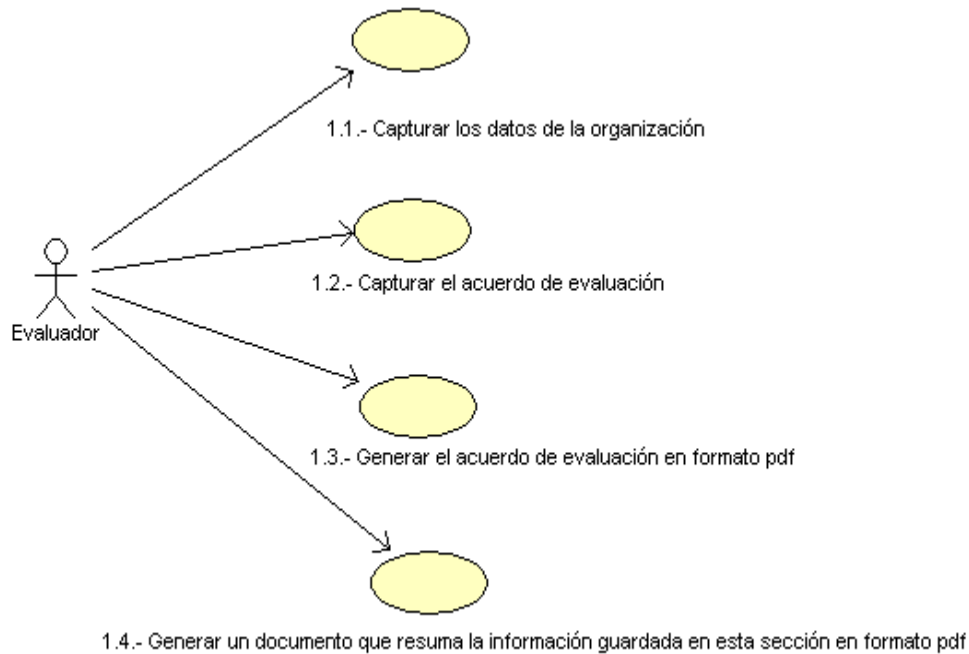


Figura 3.2
Diagrama del caso de uso 1 (Capturar la información general).

Caso de uso 2 (Capturar y editar el cuestionario de evaluación acordado)

Una vez que se sabe el alcance de la evaluación se procede a capturar la información que especifica cada uno de los procesos seleccionados. Por cada uno de los procesos es necesario que el evaluador indique a quien está entrevistando para contabilizar el tiempo y las personas involucradas en el mismo.

Por otro lado el evaluador competente es el encargado de corroborar la información recopilada de cada proceso, ya que la información recopilada que ha sido corroborada se emplea como evidencia que apoya a las prácticas base esperadas, las cuales deben de contar con al menos 2 evidencias asignadas para que su calificación pueda ser diferente de "N".

Las evidencias pueden ser utilizadas en cualquier proceso del cuestionario de evaluación sin importar en que proceso hayan sido capturadas y corroboradas, ya que también son aprovechadas para encontrar hallazgos y fortalezas de los procesos evaluados en la empresa, datos que son utilizados mas adelante en la generación de los resultados del proceso de evaluación.

Ésta sección del proceso de evaluación es una de las mas importantes y amplias en el sistema ya que el cuestionario de evaluación puede resultar ser muy extenso en cuanto a información se refiere.

Precondiciones	Post condiciones
Los datos de la organización y el acuerdo de evaluación capturados y con datos correctos.	El cuestionario de evaluación que contiene la información de la empresa evaluada capturado.

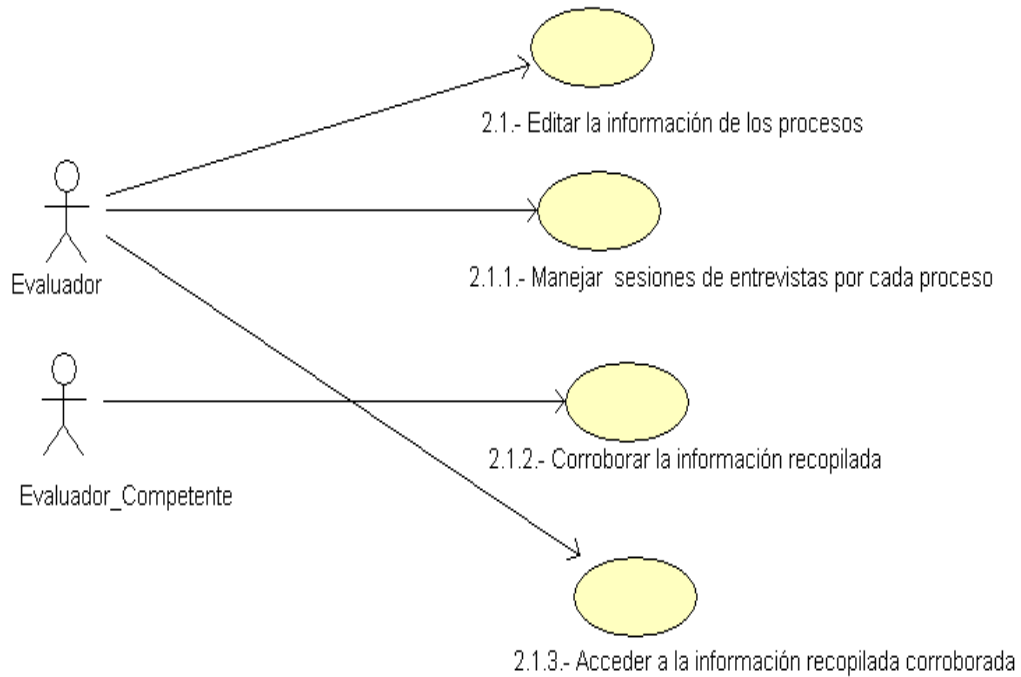


Figura 3.3
Diagrama del caso de uso 2 (Capturar y editar el cuestionario de evaluación)

Caso de uso 3 (Calificar el cuestionario de evaluación)

Con el cuestionario calificado por el evaluador y todas las sesiones cerradas, se califica el cuestionario de evaluación siguiendo el método para ejecutar la calificación de cada uno de los atributos de un proceso, que en suma conforman la calificación del propio proceso. El método de calificación del proceso es especificado por EvalProSoft y el sistema solo implementa este procedimiento.

Precondiciones	Post condiciones
Existe el cuestionario de evaluación que contiene la información de la empresa evaluada.	El cuestionario de evaluación que contiene la información de la empresa evaluada calificado.



Figura 3.3
Diagrama del caso de uso 3 (Calificar el cuestionario de evaluación)

Caso de uso 4 (Visualizar los resultados de la evaluación)

El sistema se encarga de obtener el tiempo estimado que se ha invertido en cada una de las entrevistas (sesiones) que fueron necesarias por cada proceso del cuestionario de evaluación, así como la obtención de las tareas que no fueron satisfechas, es decir, las tareas cuya calificación no es "C" (Completamente), y que sirven para brindar a la empresa evaluada mayor información acerca del estado de los procesos de desarrollo de software, así como la inversión de tiempo que fue requerida para la aplicación del cuestionario.

Con el cuestionario de la evaluación calificado se puede observar un resumen de los resultados de la evaluación, que incluye una sección donde se puede ver la gráfica en la que se observa la comparación entre el nivel objetivo, nivel deseado y el nivel obtenido por cada proceso, tareas no alcanzadas en su totalidad por la empresa y el tiempo invertido en las sesiones por cada proceso evaluado, información que es útil para obtener una idea del estado de los procesos que la empresa implanta.

Precondiciones	Post condiciones
1.- El cuestionario de evaluación que contiene la información de la empresa evaluada calificado. 2.- Los datos de la organización capturados 3.- El acuerdo de evaluación capturado	Se obtiene el resumen y gráficas de los resultados de la evaluación.

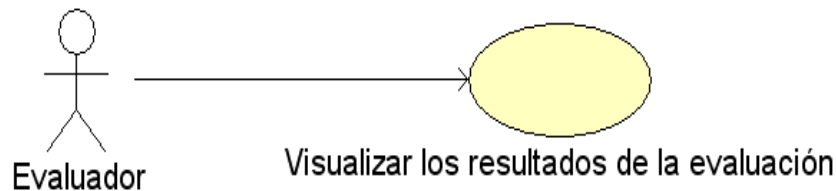


Figura 3.4
Diagrama del caso de uso 4 (Visualizar resultados de la evaluación)

Caso de uso 5 (Generar el reporte para la empresa evaluada y el reporte para el Organismo verificador)

La generación de los reportes de la empresa es la última funcionalidad del sistema, ya que estos reportes en resumen, proporcionan la información suficiente para poder saber el perfil del nivel de capacidad de los procesos implantados en la organización y un nivel de madurez de capacidades. El contenido de cada uno de los reportes está especificado por EvalProSoft (En estos reportes se anexa el tiempo estimado empleado en las entrevistas, este tiempo estimado se refiere a la suma del tiempo empleado, pero especificado hasta minutos, es decir la suma del tiempo invertido no es exacta).

Precondiciones	Post condiciones
1.- El cuestionario de evaluación que contiene la información de la empresa evaluada calificado. 2.- Los datos de la organización capturados. 3.- El acuerdo de evaluación capturado.	Se tienen los reportes: 1.- Reporte para la empresa generado. 2.- Reporte de los resultados de la empresa generado para el Organismo Verificador.

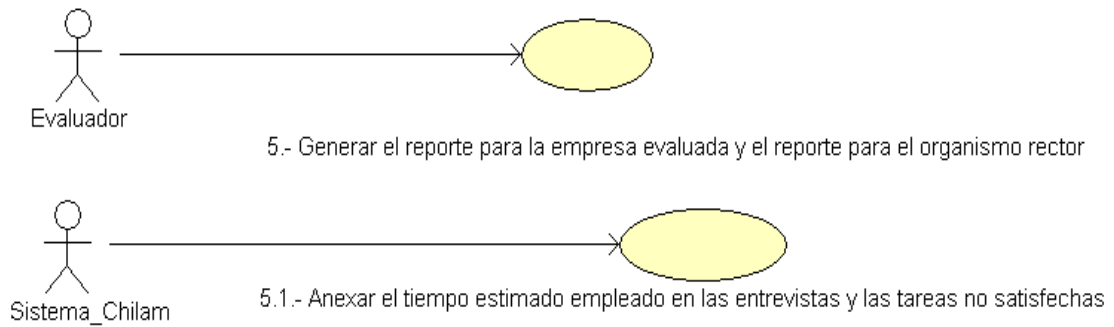


Figura 3.5
Diagrama del caso de uso 5 (Generar el reporte para la empresa evaluada y el reporte para el Organismo verificador)

Caso de uso 6 (Realizar el procedimiento para el control de cambios)

El procedimiento de control de cambios permite a los involucrados en la evaluación cambiar los términos del acuerdo de confidencialidad para adaptarlos a nuevas necesidades que sean requeridas, se espera que este procedimiento tenga que ser empleado pocas veces durante la aplicación de las evaluaciones, no obstante, es necesario realizar una serie de acciones que permitan contar con un respaldo de la información que se poseía antes de continuar con las nuevas disposiciones.

Precondiciones	Post condiciones
1.- Los datos de la organización capturados 2.- El acuerdo de evaluación capturado.	1.- Los datos de la organización adecuados. 2.- El acuerdo de evaluación adecuado. 3.- El cuestionario de evaluación adecuado.

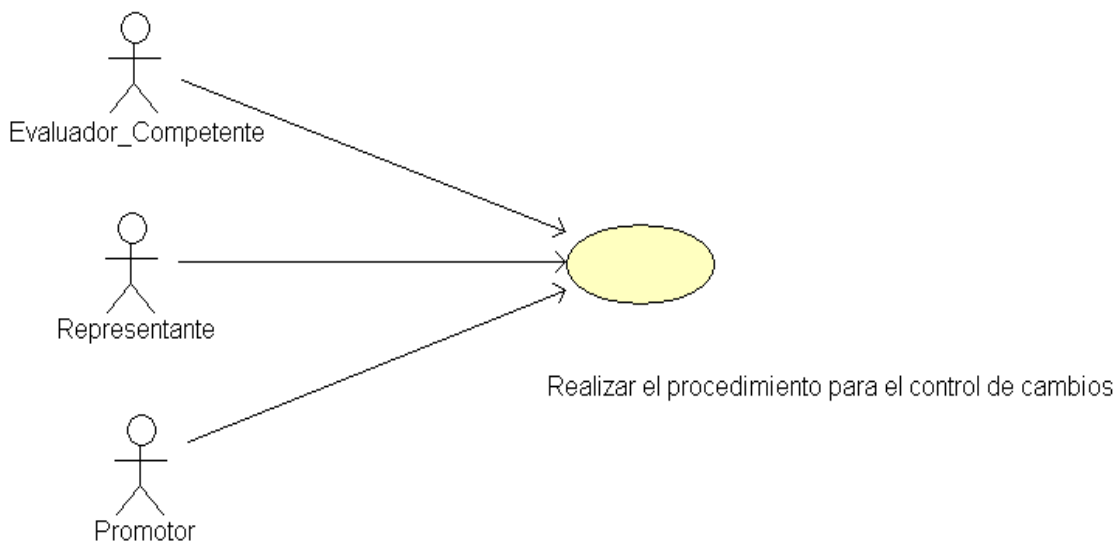


Figura 3.5
Diagrama del caso de uso 6 (Realizar el procedimiento para el control de cambios)

4.- DISEÑO DEL SISTEMA

El proceso de diseño es un conjunto de pasos repetitivos que al diseñador permiten describir los aspectos del sistema por construir. Debe ser una guía que puedan leer y entender los que construyan el código, los que prueban y mantienen el software. El diseño debe proporcionar una idea completa de lo que es el software, enfocando los dominios de datos, funcionalidades y comportamiento desde el punto de vista de la implementación.

Chilam está basado en el diseño de 3 capas (Figura 4.1).

La primera capa es de almacenamiento, que es la capa donde se encuentran los componentes del sistema que se cargan y almacenan la información desde 3 tipos de fuentes. La primera fuente son archivos en formato [XML](#) (**eXtensible Markup Language**). La segunda fuente es mediante el acceso a una base de datos utilizando el proyecto [SQLite](#), que permite acceder a la información almacenada en un archivo mediante sentencias escritas bajo el estándar SQL. La tercer fuente es [PDF](#), este formato de archivo es utilizado para generar documentos que contienen información específica, como es el acuerdo de evaluación, el reporte de resultados de la evaluación que va dirigido a la empresa evaluada etc., cabe señalar que en este formato solo se escriben los documentos.

La segunda capa del sistema, es la capa lógica de la aplicación, en esta capa se encuentran los componentes del sistema que se encargan de almacenar la información concerniente al cuestionario de evaluación, los datos de la organización evaluada y el alcance de la evaluación, información que denominaremos "dominio del problema"; los componentes que validan califican y extraen datos específicos con la información del dominio del problema; los componentes que proporcionan funcionalidades útiles para varias clases en diferentes capas y los componentes que permiten el rastreo y manejo de los casos excepcionales de la interfaz.

La tercer capa, es la capa de presentación, en esta capa se encuentran los componentes del sistema que se encargan de presentar las ventanas de la interfaz, ventanas que procuran guiar al usuario a través del proceso de la evaluación en el sistema, verificar que la información que proporciona el usuario es correcta, verificar que la información que conforma el proyecto es correcta, es decir, verificar que las precondiciones de ciertas secciones de información que forma parte de la evaluación se cumplan, entre otros aspectos.

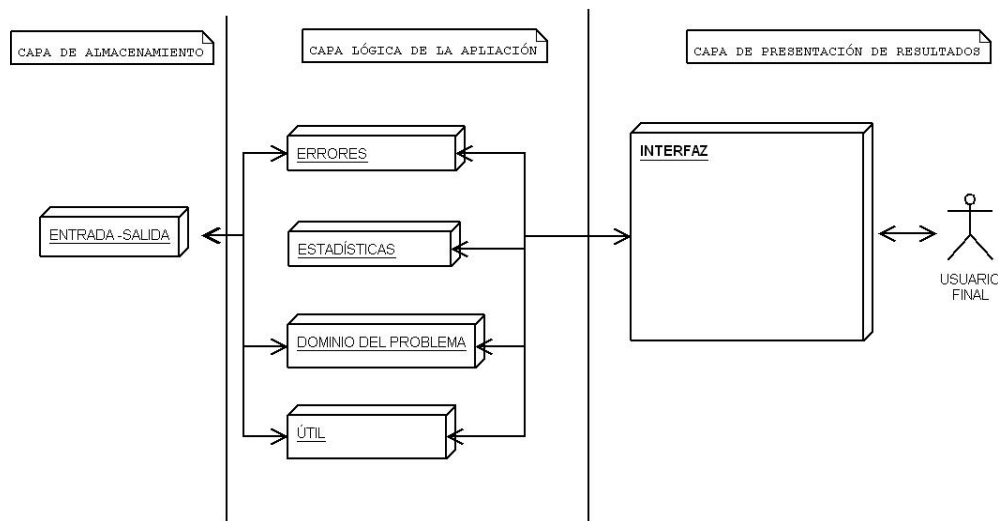


Figura 4.1
Diagrama del modelo de 3 capas de la arquitectura de Chilam

4.1- Diagrama de paquetes

Cualquier sistema se debe dividir en unidades más pequeñas, de modo que las personas puedan trabajar con una cantidad de información limitada a la vez, Al dividir el trabajo entre un equipo de personas el trabajo no interfieran con el trabajo de los otros.

Un paquete es una parte del diseño del sistema. Los paquetes ofrecen un mecanismo general para la organización de los subsistemas agrupando elementos de modelado. Pero para ser funcional, la asignación debe seguir un cierto principio funcional, una implementación relacionada y punto de vista común.

Los paquetes de Chilam están diseñados y organizados siguiendo el modelo de 3 capas que sigue la arquitectura del sistema, algunos paquetes están compuestos por otros paquetes que ayudan a dividir las funcionalidades para obtener mayor modularidad. Para obtener un mejor control del sistema cada una de las 3 capas de la arquitectura del sistema está conformada por paquetes, estos paquetes poseen una organización jerárquica independiente, la cual permite agrupar las clases que proveen las funcionalidades antes descritas. Además de proporcionar una modularidad al sistema para que los futuros cambios o adecuaciones sean locales a ciertos paquetes o posiblemente tan solo en algunas funciones de las clases. (Figura 4.2)

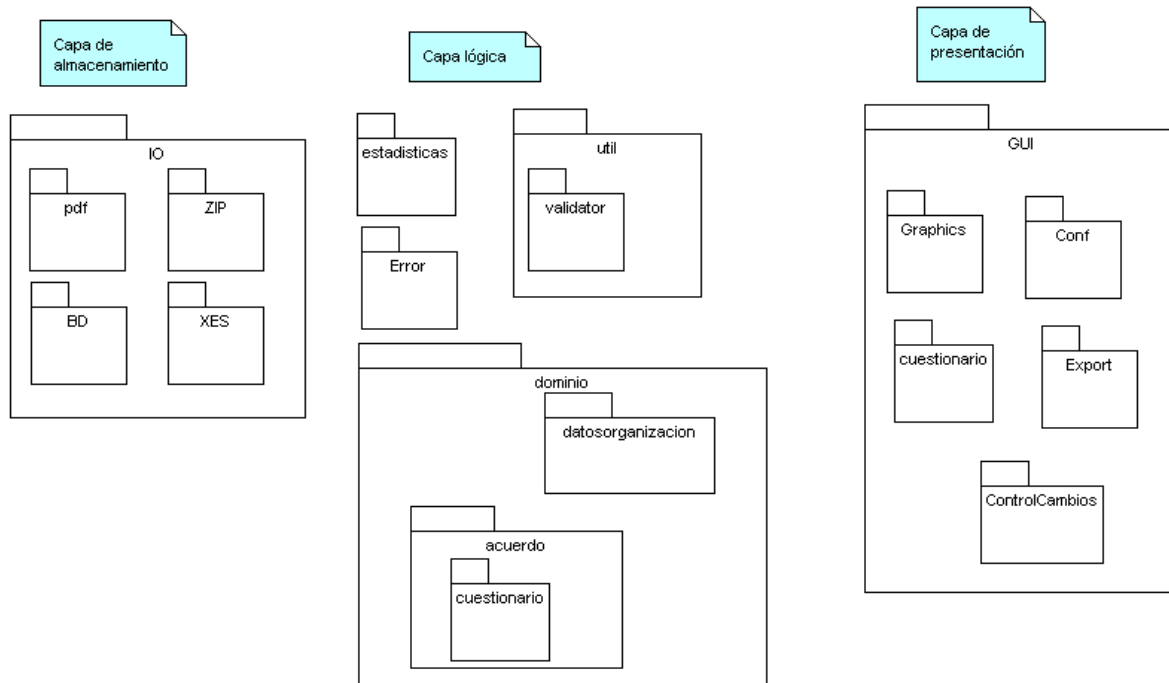


Figura 4.2

Diagrama de los paquetes que componen Chilam divididos por capa.

4.2.- Diagrama de clases

El Diagrama de Clases es el diagrama principal para el análisis y diseño. Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia. La definición de clase incluye definiciones para atributos y operaciones. El diagrama de clases aporta información para establecer las clases, objetos, atributos y operaciones.

A continuación se presentan los diagramas de clases de los paquetes que proporcionan la solución a los requerimientos planteados. Los diagramas están ordenados en las capas y paquetes de la arquitectura.

Estos diagramas contienen los atributos y métodos de las clases, en estos diagramas no se tomaron en cuenta clases privadas auxiliares en secciones muy específicas de los objetos que no tienen mayor relevancia en el diseño. Sin embargo se puede consultar la documentación de cada uno de los atributos y métodos de las clases en la página del proyecto que se incorporará cuando la segunda parte del proyecto (Balam) sea construida, así como se pueden observar los comentarios de las clases no referenciadas en los diagramas.

Clases contenidas en el paquete IO

Las clases del paquete IO permiten al sistema cargar y almacenar la información requerida desde distintas fuentes.

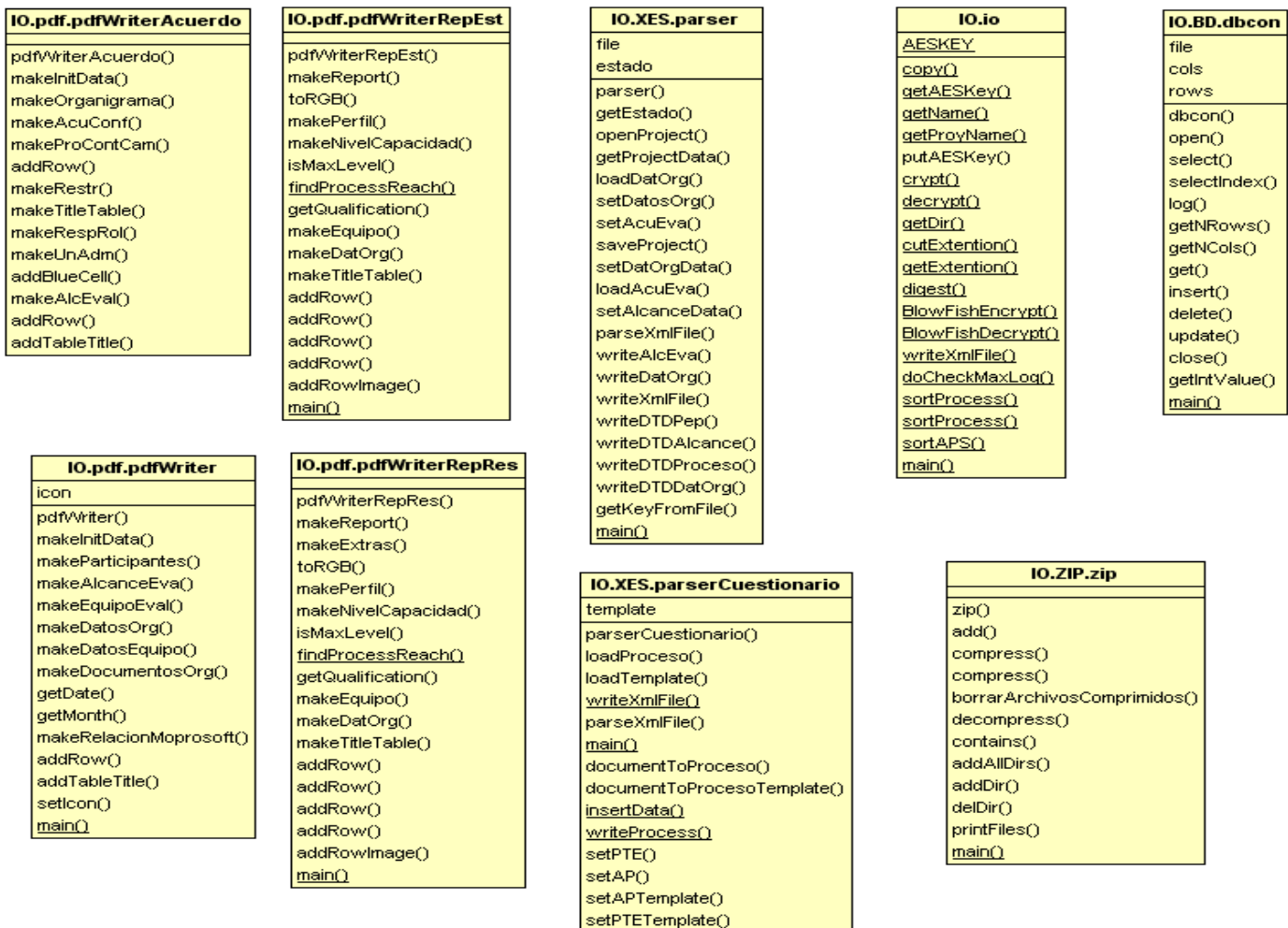


Figura 4.3
Clases en el paquete IO.

Clases contenidas en el paquete estadísticas

La clase "calificador" implementa el método de calificación especificado por EvalProSoft para obtener los resultados de la evaluación.

calificador
<u>AP_PTE_VIEW</u>
<u>LEVEL_VIEW</u>
<u>calificador()</u>
<u>isReady()</u>
<u>doProcess()</u>
<u>setLevelReached()</u>
<u>sortArrays()</u>
<u>isSort()</u>
<u>setLevelQualification()</u>
<u>translate()</u>
<u>setAPQualification()</u>
<u>setPTEQualification()</u>
<u>getCalificacion()</u>
<u>main()</u>

Figura 4.4
Clases del paquete estadísticas.

Clases contenidas en el paquete Error

Las clases del paquete "Error", gestionan los casos excepcionales identificados por el sistema y permiten llevar un rastreo más detallado de los mismos.

Error.Error	Error.Log
idmod	file
numError	<u>Log()</u>
info	<u>getDate()</u>
path	<u>getTime()</u>
errorFile	<u>getDayName()</u>
javaE	<u>getMonthName()</u>
show	<u>log()</u>
<u>Error()</u>	<u>errorLog()</u>
<u>Error()</u>	<u>main()</u>
<u>setJavaException()</u>	
<u>show()</u>	
<u>setModule()</u>	
<u>setModule()</u>	
<u>setError()</u>	
<u>setError()</u>	
<u>addInfo()</u>	
<u>addPath()</u>	
<u>toLog()</u>	
<u>stackTraceToString()</u>	
<u>getModuleName()</u>	
<u>main()</u>	

Error.XMLbundle
file
TAG
key
message
<u>setFile()</u>
<u>getMessage()</u>
<u>startDocument()</u>
<u>endDocument()</u>
<u>startElement()</u>
<u>endElement()</u>
<u>characters()</u>
<u>main()</u>

Figura 4.5
Clases del paquete Error.

Clases contenidas en el paquete util

Las clases del paquete "util" implementan funcionalidades que son empleadas por diversas clases en el sistema, sin importar el paquete en el que han sido agrupadas.

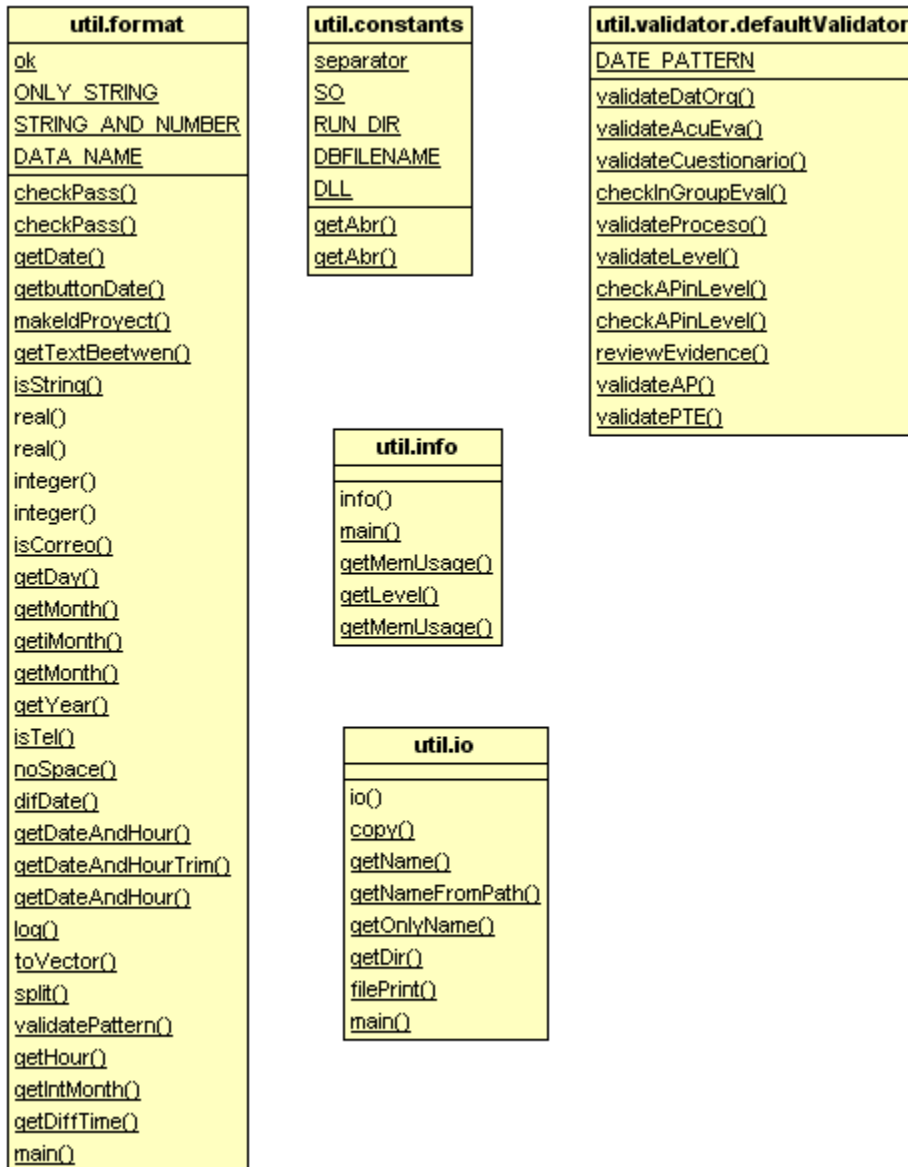


Figura 4.6
Clases del paquete "util".



Figura 4.7-b
Clases del dominio del problema

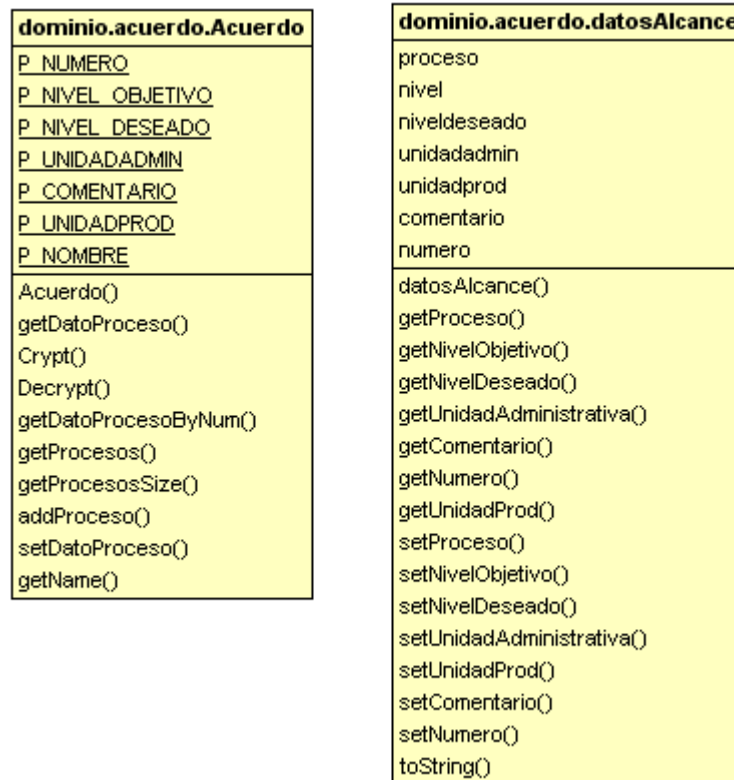


Figura 4.7-c
Clases del dominio del problema

3.5.6.- Clases contenidas en el paquete GUI

Las clases del paquete GUI presentan la interfaz gráfica al usuario y procuran implementar una lógica de navegación segura en el sistema.

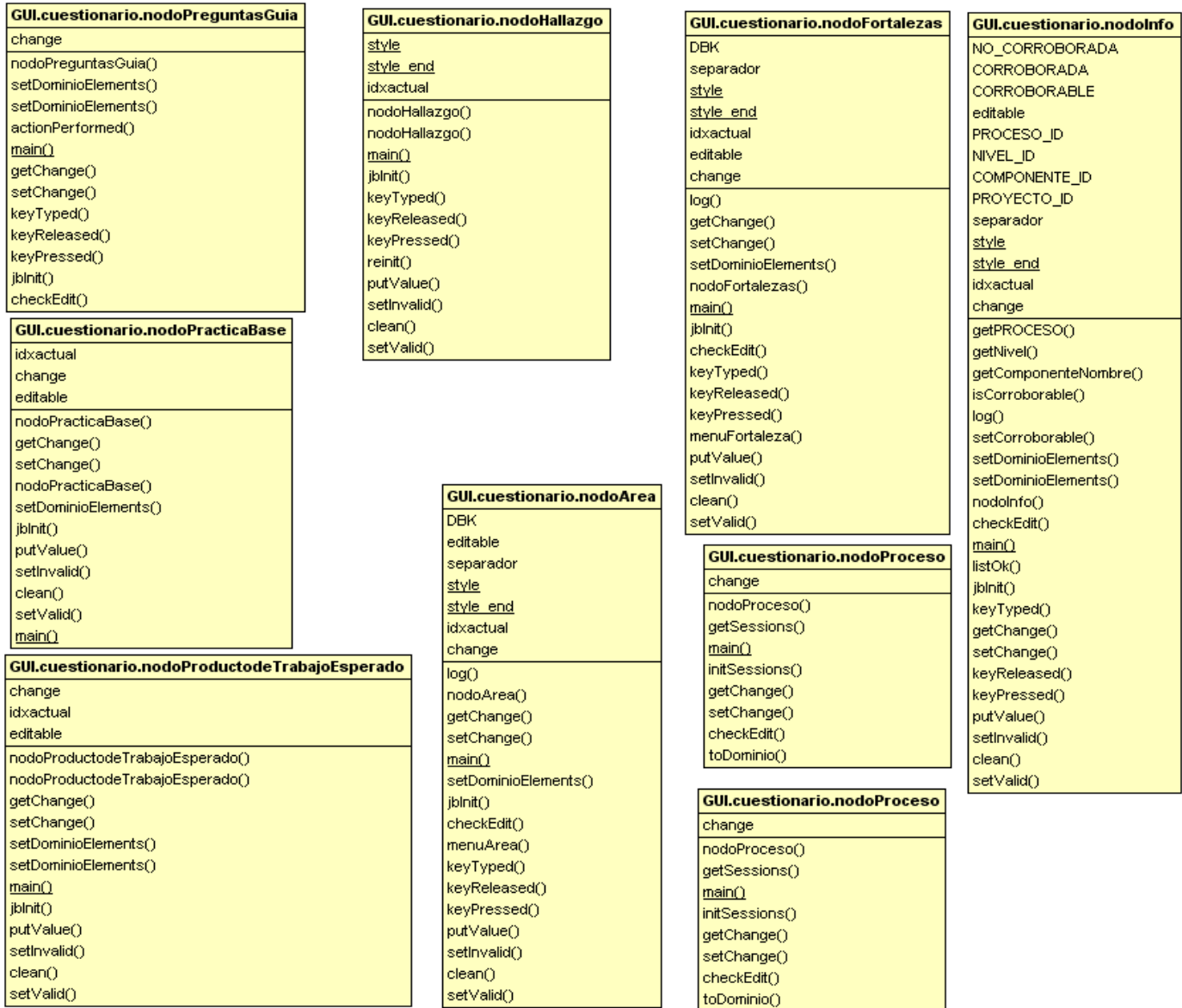


Figura 4.8-a
Clases del paquete "GUI".

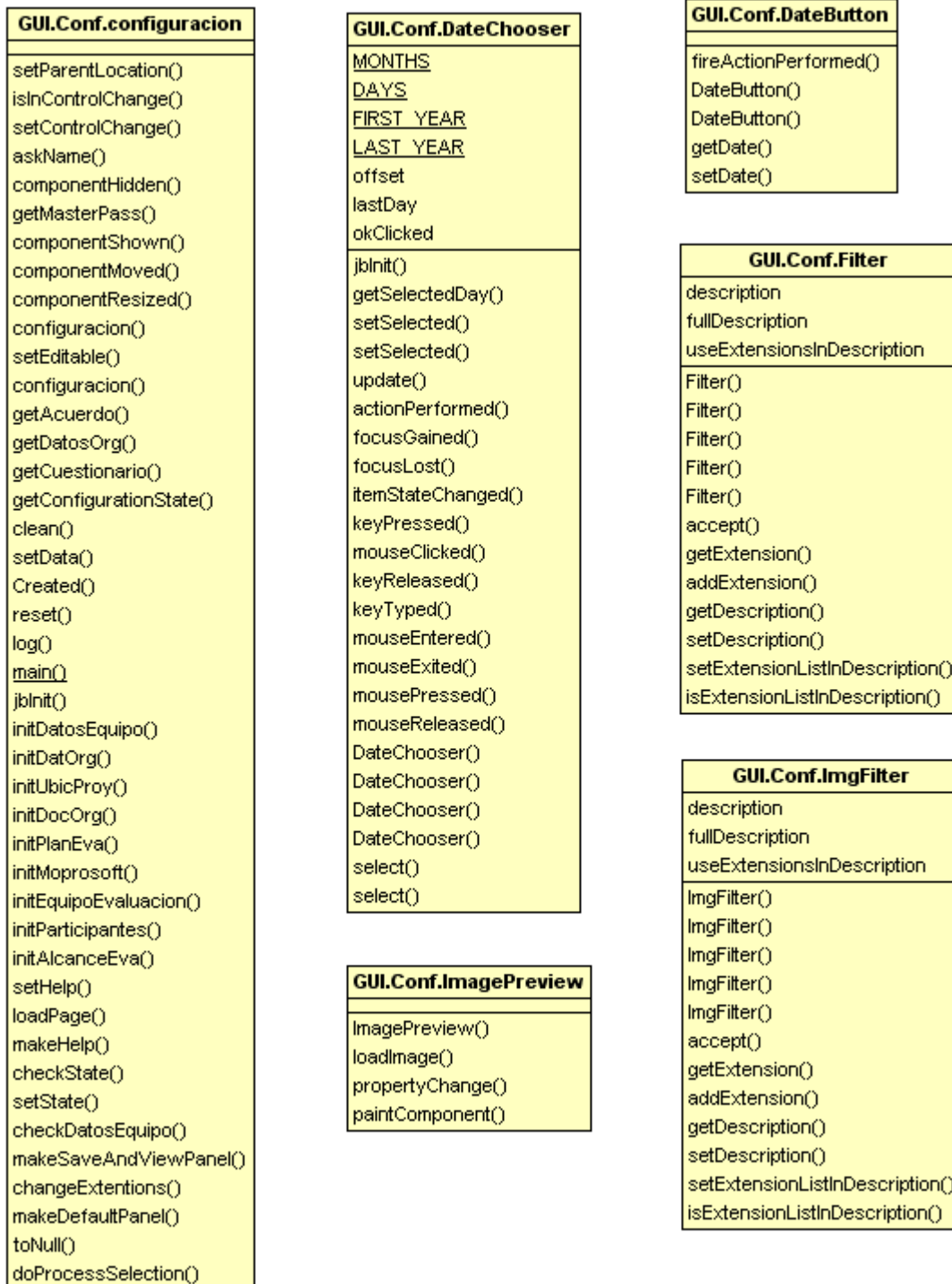


Figura 4.8-b
Clases del paquete GUI

GUI.cuestionario.DialogoCuestionario
editable
DBK
corroborar
DialogoCuestionario()
log()
isEditable()
main()
Resize()
cleanScreen()
jbInit()
actions()
toDominio()
getChange()
setAPDom()
getAPChange()
setPTEDom()
getPTEChange()
makeHelp()
loadPage()
makeTree()
setAP()
setPTE()
valueChanged()
checkEdit()
isInPanel()
desplegable()
setPanelVisible()
setCorroborable()

GUI.cuestionario.evidencias
DEBUG
insert
evidencias()
main()
fill()
getSelected()
getSelectedAsString()
Actions()
jbInit()

GUI.cuestionario.IconCellRenderer
m_selected
IconCellRenderer()
getTreeCellRendererComponent()
paint()

GUI.cuestionario.data
setValue()
getData()
getDataAsInt()
getDataAsDouble()
getDataAsBoolean()
toString()

GUI.cuestionario.IconData
label
data
name
idproc
setProc()
getData()
getProc()
setName()
getName()
IconData()
hasData()
getLabel()
setData()
getData()
getDataAsDouble()
getDataAsInt()
getDataAsBoolean()
IconData()
getIcon()
getOpenIcon()
getObject()
toString()

GUI.cuestionario.Nodo
toolTipText
Nodo()
getToolTipText()
getUserObjec()

GUI.cuestionario.msg
msg()
msg()
msg()
jbInit()
setText()
main()

GUI.cuestionario.passwordPane
ok
correctPass
hlp
passwordPane()
passwordPane()
passwordPane()
getPass()
setAyuda()
reset()
doPanel()
getResult()
actionPerformed()
main()

GUI.cuestionario.ItemChooser
DBK
change
ItemChooser()
setData()
jbInit()
isChoosed()
isContained()
getChange()
setChange()
Actions()
log()
getSeleccionados()
listeners()
swap()
main()

Figura 4.8-c
Clases del paquete GUI

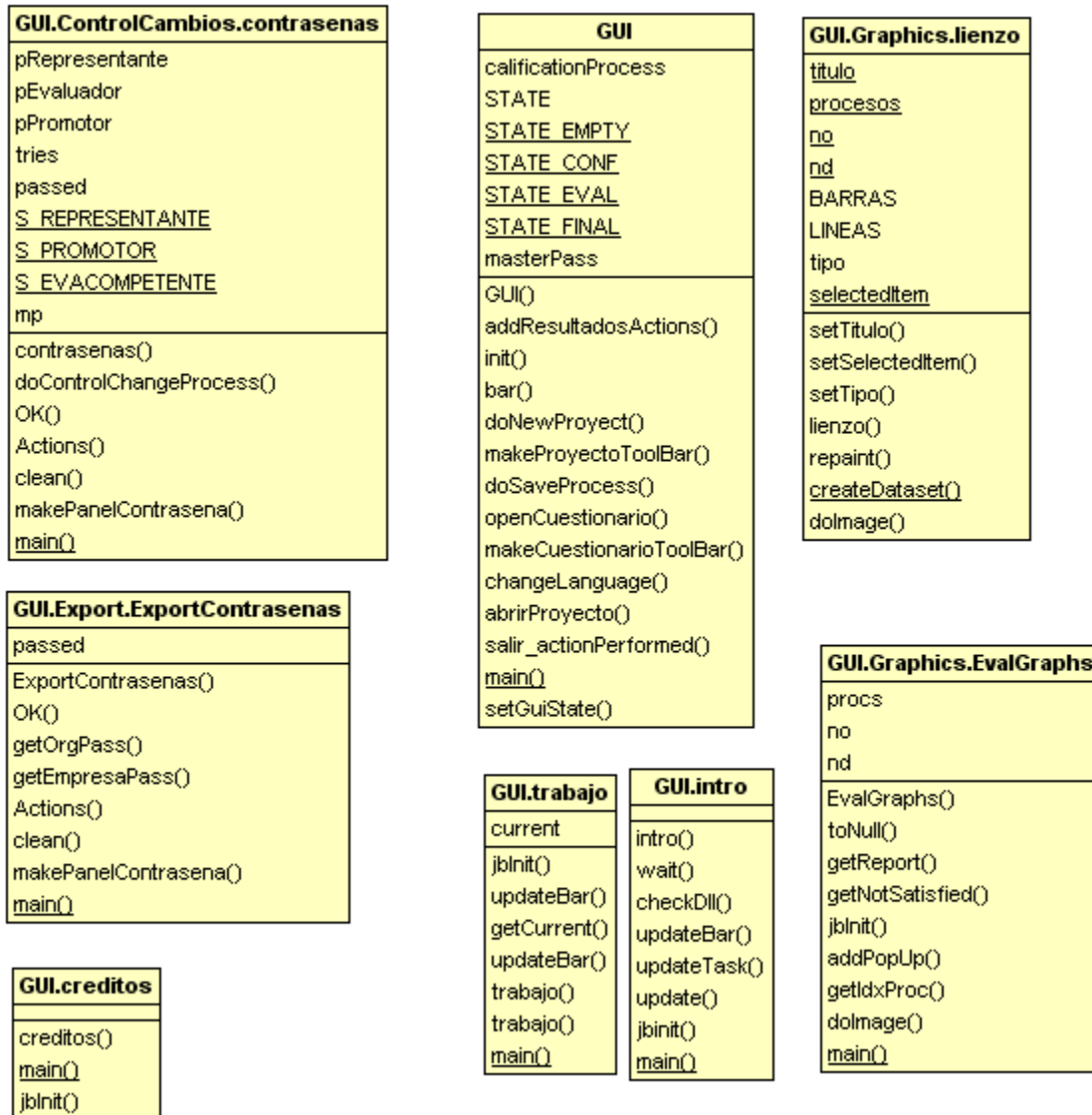


Figura 4.8-d
Clases del paquete GUI

Diagrama de Clases que muestra la relación que existe entre las clases siguiendo el modelo de 3 capas del sistema.

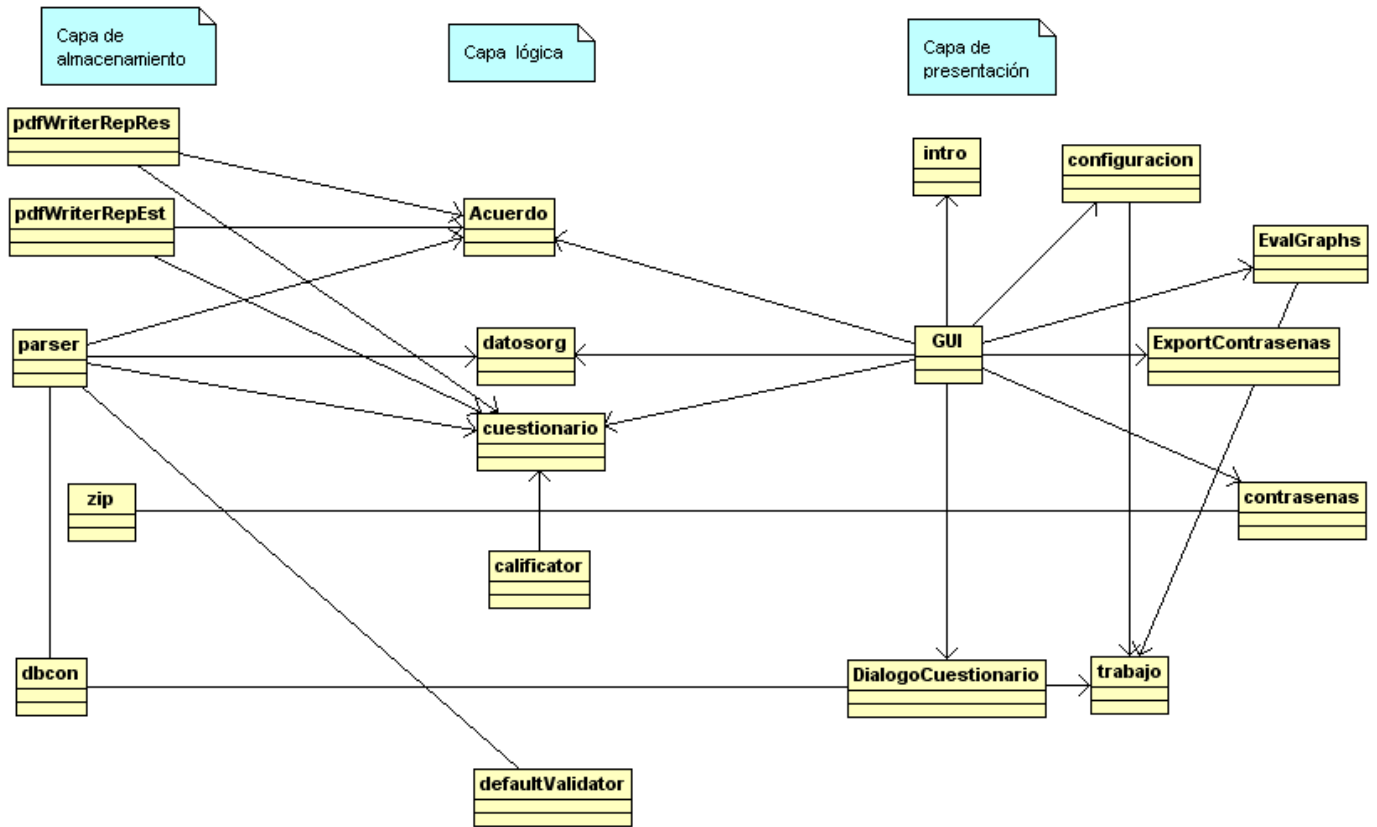


Figura 4.9
Diagrama de clases del sistema.

4.3.- DIAGRAMAS DE COLABORACIÓN

Un diagrama de colaboración es una forma de representar interacción entre objetos, el diagrama de colaboración proporciona una forma de ver el escenario en un orden temporal ya que se puede apreciar el orden en que ocurren las interacciones entre los objetos.

Para proporcionar una solución a los requerimientos planteados en la sección 3.2 se requiere de la colaboración de las clases contenidas en los paquetes, para proporcionar una idea más clara acerca de las tareas que realiza Chilam, a continuación se muestran los diagramas de colaboración.

Diagrama de colaboración para el caso de uso 1 (Capturar información inicial)

El evaluador acciona el evento "doNewProyect" que le indica a la interfaz que se desea generar un proyecto nuevo, la interfaz construye un objeto de la clase "configuracion", clase que se encarga a partir de este punto de guiar al evaluador en los pasos necesarios para capturar la información básica y generar una evaluación.

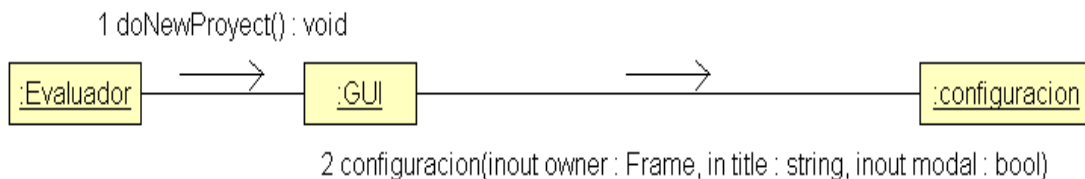


Figura 4.10

Diagrama de colaboración para el caso de uso 2(Capturar y editar el cuestionario de evaluación)

El evaluador acciona el evento "openCuestionario" que le indica a la interfaz que se desea mostrar el cuestionario de evaluación, la interfaz construye un objeto de la clase "DialogoCuestionario", el cual se encarga de comenzar a construir cada uno de los objetos que conforman las secciones de cada proceso para mostrar el árbol de navegación del cuestionario de evaluación. Cada una de las clases que son construidas reciben como parámetros la sección de la información del cuestionario que gestionan, es decir, cada objeto verifica que el proceso de captura y edición de la sección de la información que gestiona es correcta y procura que la información se conserve íntegra.

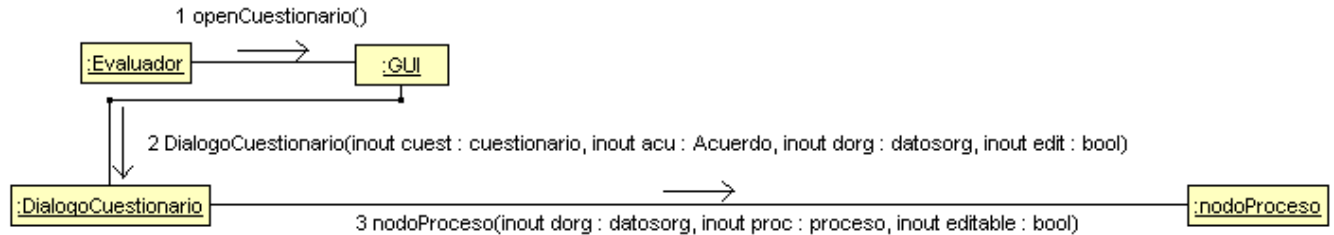


Figura 4.11-a
Diagrama de colaboración de las clases que intervienen en el caso de uso 2.

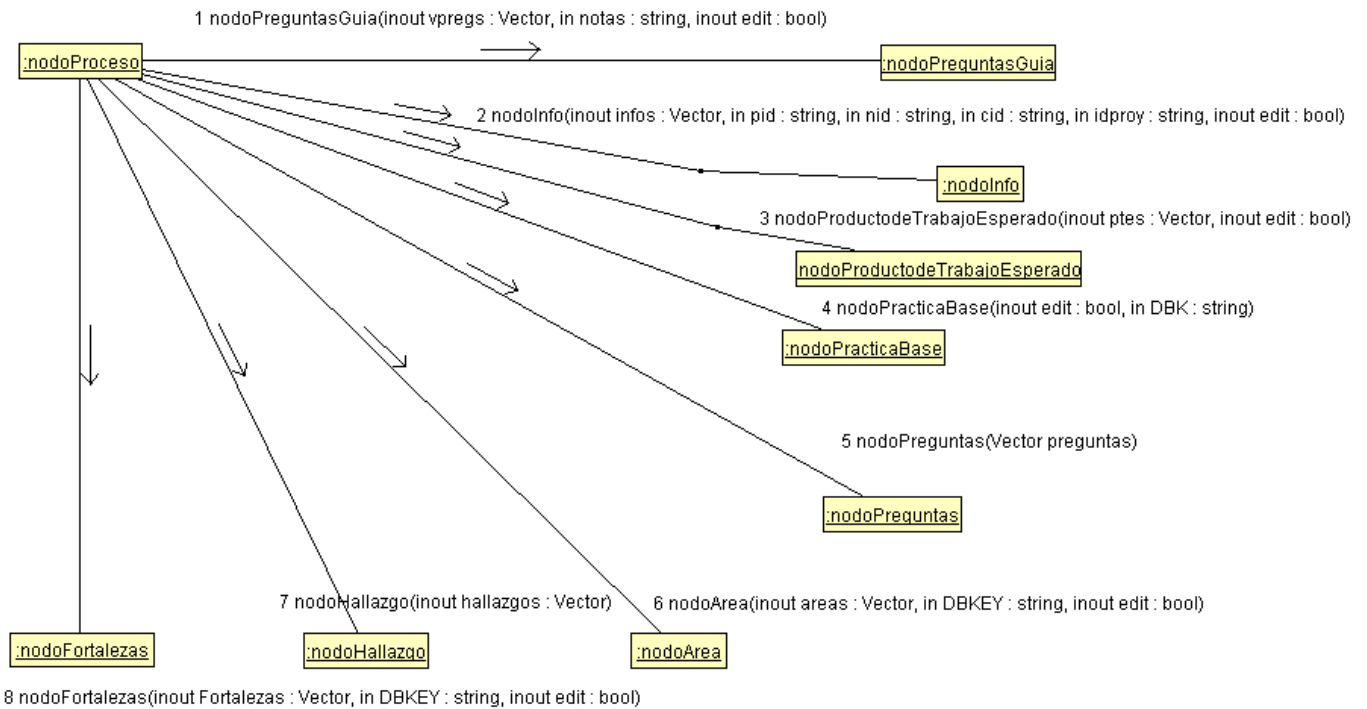


Figura 4.11-b
Diagrama de colaboración de las clases que intervienen en el caso de uso 2.

Diagrama de colaboración para el caso de uso 3(Calificar el cuestionario de evaluación)

El evaluador acciona el evento "calificar" que le indica a la interfaz que se desea calificar el cuestionario de evaluación, la interfaz extrae la información que se encuentra en cada una de las secciones de los componentes de la interfaz, que permite editar cada proceso para conformar el cuestionario de evaluación. Cuando el cuestionario de evaluación está conformado la interfaz valida por medio de la clase "calificador" que el cuestionario cumpla con los requerimientos mínimos para poder ser calificado, y en caso de cumplir con los requerimientos, la interfaz califica el cuestionario por medio de la clase "calificador".

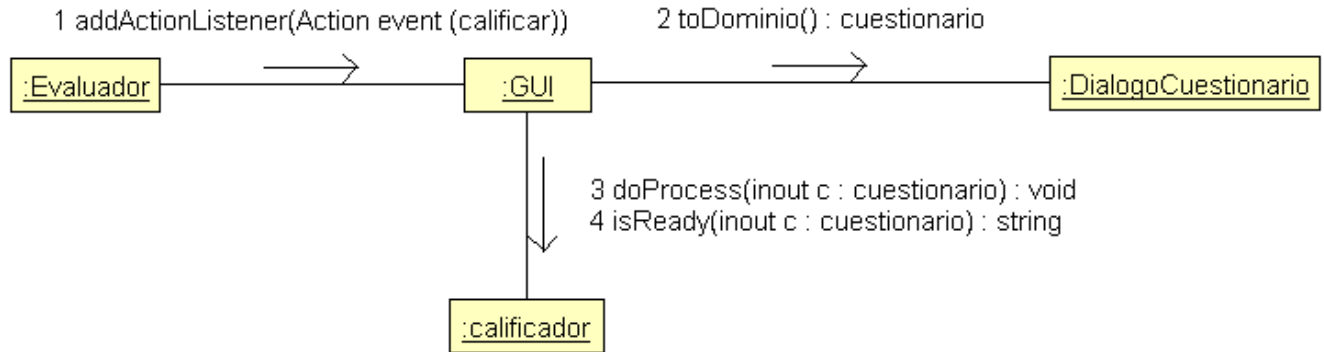


Figura 4.12
Diagrama de colaboración de las clases que intervienen para efectuar la calificación de un cuestionario.

Diagrama de colaboración para el caso de uso 4(Visualizar los resultados de la evaluación)

El evaluador acciona el evento "Ver resultados" que le indica a la interfaz que se desean observar los resultados de la evaluación, la interfaz extrae la información referente al tiempo invertido durante la evaluación y el reporte de las tareas no satisfechas del cuestionario de evaluación, para que éstas sean mostradas en la interfaz.

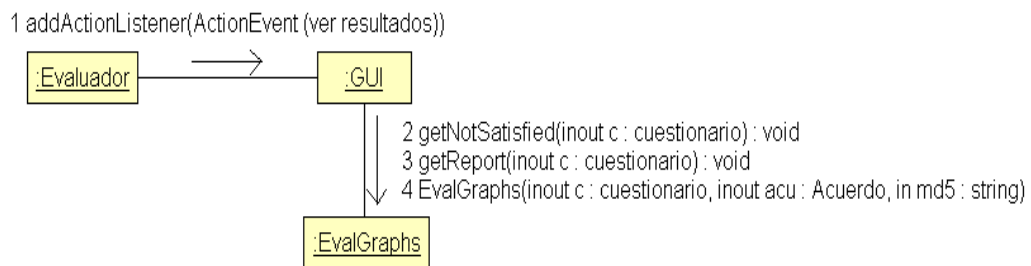


Figura 4.13
Diagrama de colaboración de las clases que intervienen para mostrar los resultados de la evaluación.

Diagrama de colaboración para el caso de uso 5(Generar el reporte para la empresa evaluada y para el Organismo Verificador)

El evaluador acciona el evento "Generar reportes" que le indica a la interfaz que se desean generar los reportes, la interfaz muestra el diálogo para que se establezcan las contraseñas de cada uno de los reportes que se van a generar mediante la creación de un objeto de la clase " exportContrasenas", la interfaz extrae cada uno de las contraseñas del diálogo y crea el objeto de la clase "pdfWriterRepEst", objeto al cual le proporciona la información necesaria y utiliza el método específico que genera el reporte estadístico, posteriormente crea un objeto de la clase "pdfWriterRepRes" al cual le proporciona la información necesaria y utiliza el método específico que genera el reporte de resultados.

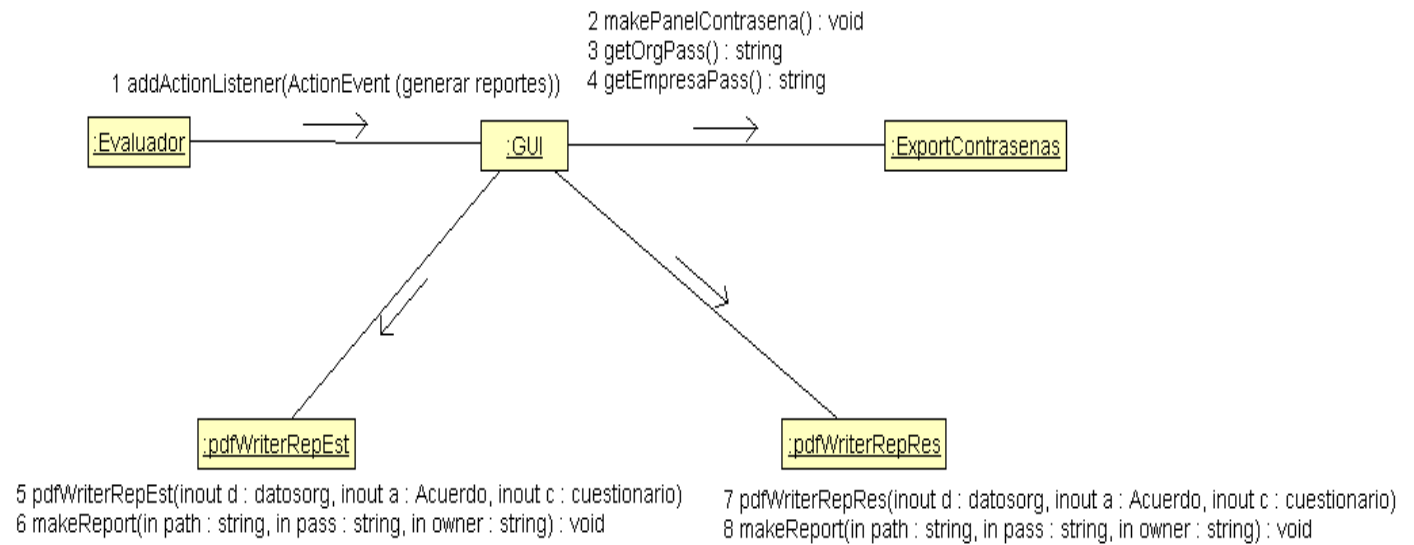


Figura 4.14
Diagrama de colaboración de las clases que intervienen en la generación de los reportes de resultados de la evaluación.

Diagrama de colaboración para el caso de uso 6(Realizar el procedimiento para el proceso de control de cambios)

El evaluador acciona el evento "Control de cambios" que le indica a la interfaz que se desea accionar el proceso de control de cambios, la interfaz muestra el dialogo para que se establezcan las contraseñas del representante, promotor y evaluador, con el fin de autenticar que el control de cambios es autorizado. Con la seguridad de que el control de cambios es autorizado se realiza el respaldo de la información actual y se muestra la ventana de configuración para que el usuario capture las adecuaciones a la evaluación.

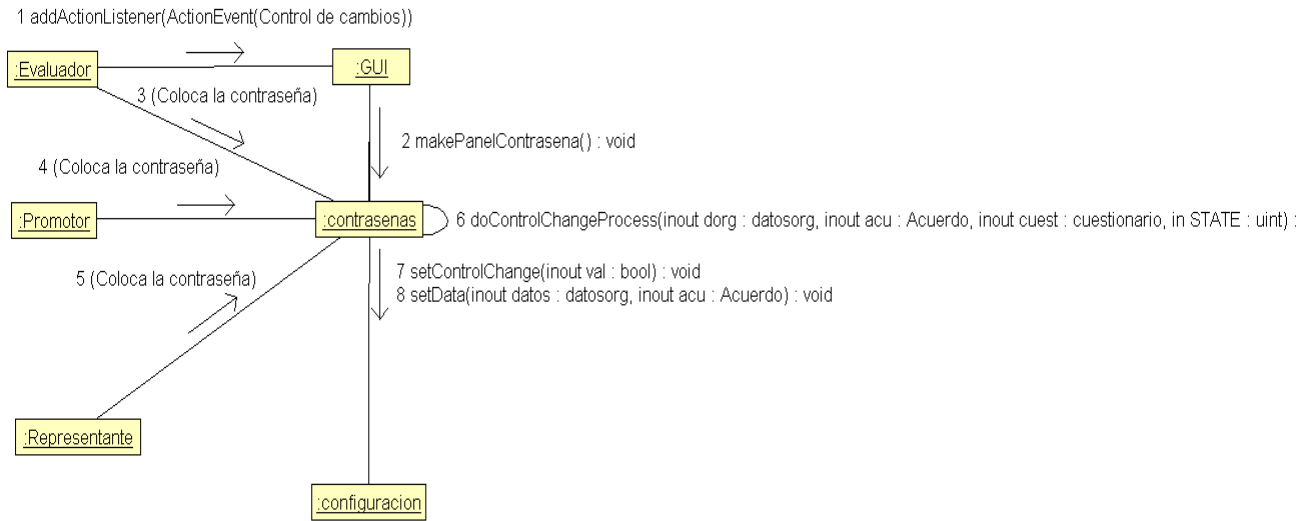


Figura 4.15
Diagrama de colaboración de las clases que intervienen en el proceso de control de cambios

4.4.- DIAGRAMAS DE SECUENCIA

Diagrama que muestra las interacciones entre los objetos organizadas en una secuencia temporal. En particular muestra los objetos participantes en la interacción y la secuencia de mensajes intercambiados. Dentro del conjunto de mensajes representados dispuestos en una secuencia temporal. Cada rol en la secuencia se muestra como una línea de vida, es decir, una línea vertical que representa el rol durante cierto plazo de tiempo, con la interacción completa. Los mensajes se muestran como flechas entre líneas de vida. Un diagrama de secuencia puede mostrar un escenario, es decir, una historia individual de transacción. Un uso de un diagrama de secuencia es mostrar la secuencia del comportamiento de un caso de uso.

Los diagramas de secuencia exponen con más detalle las acciones que se llevan a cabo entre el usuario final y las clases que conforman el sistema para satisfacer los requerimientos de cada caso de uso, además especifican los casos excepcionales que son identificados en el proceso.

Diagrama de Secuencia para el caso de uso 1.1 y 1.2 (Capturar los datos de la organización, capturar el acuerdo de evaluación)

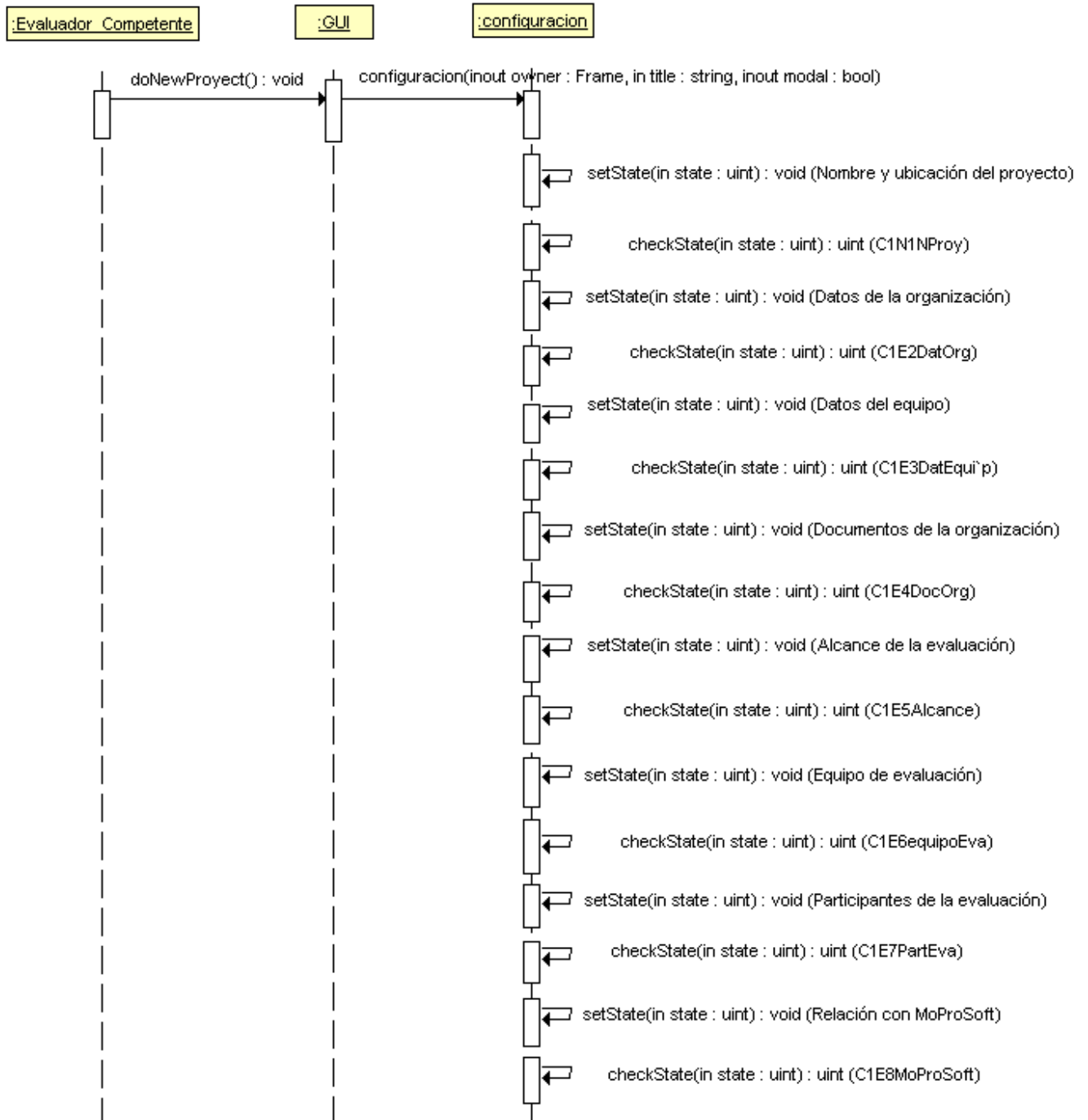


Figura 4.16
Diagrama de secuencia de las clases que intervienen para generar un nuevo proyecto

Flujo

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Emite el evento Nuevo Proyecto	2	Despliega la ventana de configuración	
3	Emite el evento (1.- Nombre y ubicación del proyecto)	4	El Sistema muestra los campos de captura para los datos requeridos y valida la información ingresada.	C1E1NProy
5	Emite el evento (2.- Datos de la organización)	6	El Sistema muestra los campos de captura para los datos requeridos y valida la información ingresada.	C1E2DatOrg
7	Emite el evento (3.-Datos del equipo)	8	El Sistema muestra los campos de captura para los datos requeridos y valida la información ingresada.	C1E3DatEquip
9	Emite el evento (4.-Documentos de la organización)	10	El Sistema muestra los campos para seleccionar los documentos y verificar su existencia.	C1E4DocOrg
11	Emite el evento (5.-Alcance de la evaluación)	12	El Sistema muestra los campos de captura para los datos requeridos y valida la información ingresada.	C1E5Alcance
13	Emite el evento (6.-Equipo de evaluación)	14	El Sistema muestra los campos de captura para los datos requeridos y valida la información ingresada.	C1E6EquipoEva
15	Emite el evento (7.-Participantes de la evaluación)	16	El Sistema muestra los campos de captura para los datos requeridos y valida la información ingresada.	C1E7PartEva
17	Emite el evento (8.- Relación con MoProSoft)	18	El Sistema muestra los campos de captura para los datos requeridos.	C1E8MoProSoft

Errores

id	Causa del error	Acción
C1E1NProy	Información invalida	Muestra los campos que se encuentran mal llenados, muestra el mensaje de error en la parte inferior de la ventana y manda un mensaje de volver a escribir la información.
C1E2DatOrg	Información invalida	Muestra los campos que se encuentran mal llenados, muestra el mensaje de error en la parte inferior de la ventana y manda un mensaje de volver a escribir la información.
C1E3DatEquip	Información invalida	Muestra los campos que se encuentran mal llenados, muestra el mensaje de error en la parte inferior de la ventana y manda un mensaje de volver a escribir la información.
C1E4DocOrg	Documentos inexistentes	Muestra un icono que indica que el documento no esta seleccionado correctamente.
C1E5Alcance	Información invalida	Muestra los campos que se encuentran mal llenados, muestra el mensaje de error en la parte inferior de la ventana y manda un mensaje de volver a escribir la información.
C1E6EquipoEva	Información invalida	Muestra los campos que se encuentran mal llenados, muestra el mensaje de error en la parte inferior de la ventana y manda un mensaje de volver a escribir la información.
C1E7PartEva	Información invalida	Muestra los campos que se encuentran mal llenados, muestra el mensaje de error en la parte inferior de la ventana y manda un mensaje de volver a escribir la información.
C1E8MoProSoft	Información invalida	Muestra los campos que se encuentran mal llenados, muestra el mensaje de error en la parte inferior de la ventana y manda un mensaje de volver a escribir la información.

Diagrama de Secuencia para el caso de uso 1.3 (Generar el acuerdo de evaluación en formato pdf)

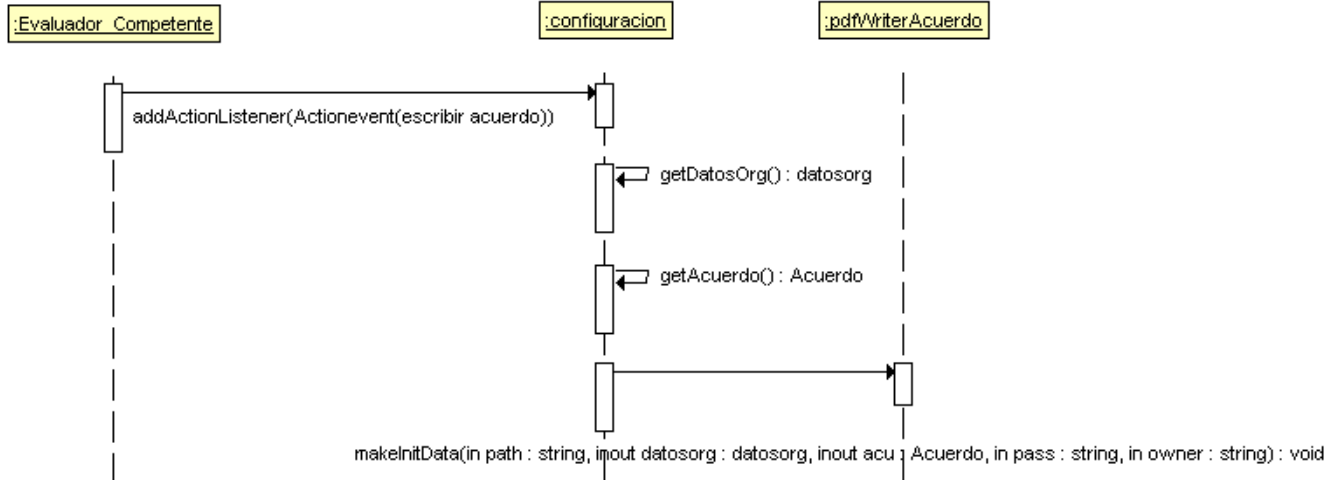


Figura 4.17

Diagrama de secuencia de las clases que intervienen en el proceso de creación del Acuerdo de confidencialidad

Flujo

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Emite el evento para escribir el acuerdo de evaluación	2	El sistema recorre los campos donde el usuario ha capturado información y la almacena.	
		3	El sistema recorre los campos donde el usuario ha capturado los datos del acuerdo de la evaluación almacena la información.	
		4	El sistema crea el acuerdo de la evaluación.	

Diagrama de Secuencia para el caso de uso 1.4 (Generar un documento que resume la información guardada en esta sección en formato pdf).

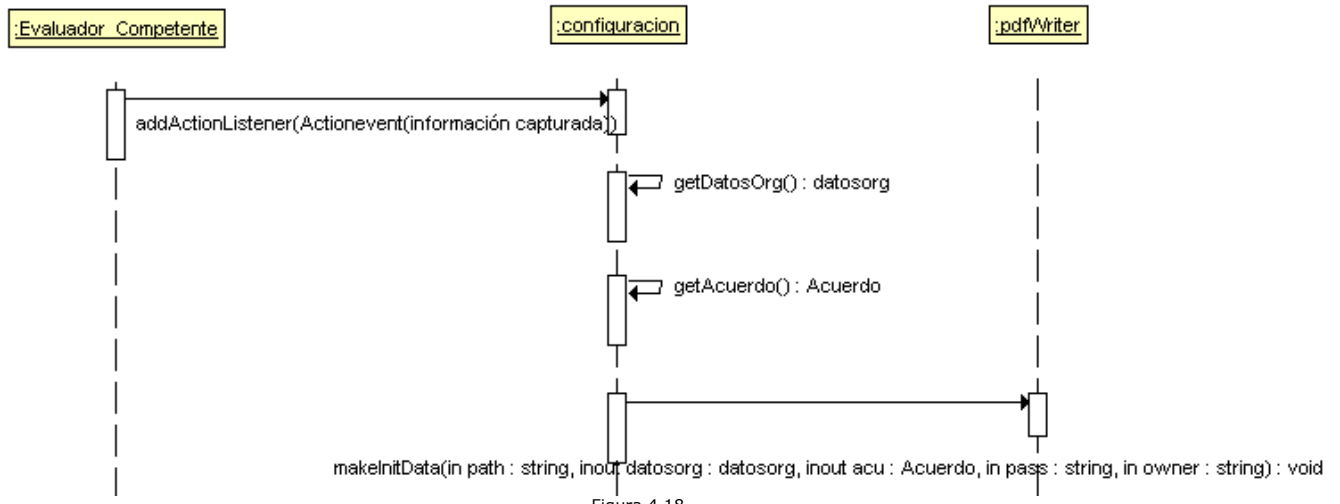


Figura 4.18

Diagrama de secuencia de las clases que intervienen en el proceso de creación del documento que resume la información contenida en la configuración.

Flujo

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Emite el evento para escribir la información que ha capturado en esta sección.	2	El sistema recorre los campos donde el usuario ha capturado información y la almacena.	
		3	El sistema recorre los campos donde el usuario ha capturado los datos del acuerdo de la evaluación y los almacena.	
		4	El sistema crea el documento que resume la información guardada en esta sección..	

Diagrama de Secuencia para el caso de uso 2.1 (Editar la información de los procesos).

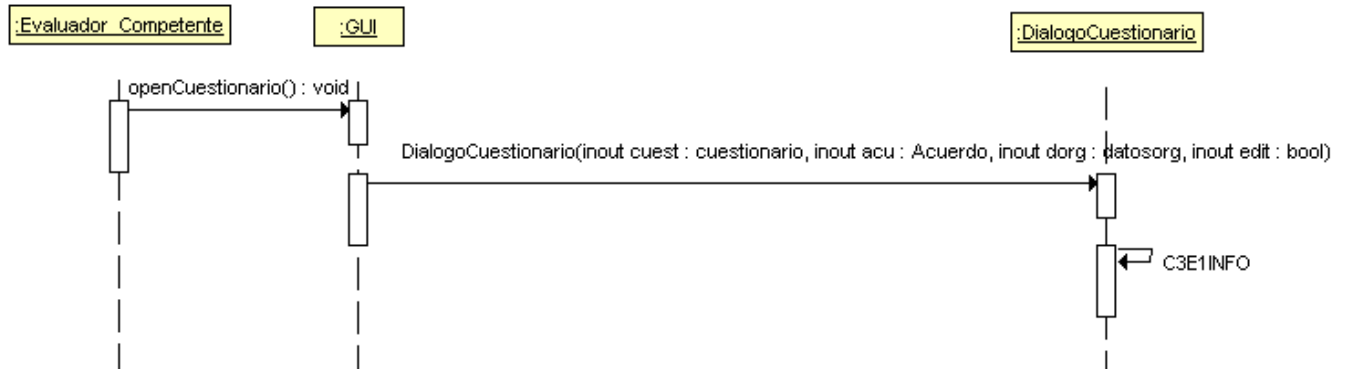


Figura 4.19 Diagrama de secuencia de las clases que intervienen para editar la información de los procesos

Flujo

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Emite el evento para editar el cuestionario de evaluación.	2	El sistema muestra el cuestionario de evaluación..	
		3	El dialogo presenta el cuestionario de evaluación y permite al usuario navegar, capturar y editar el cuestionario.	C3E1INFO

Error

id	Causa del error	Acción
C3E1INFO	Información invalida	Envía al usuario un mensaje donde le indica la razón por la cual la información es incorrecta y procura dejar el estado de la interfaz en un estado correcto.

Diagrama de Secuencia para el caso de uso 2.1.1 (Manejar sesiones de entrevistas por cada proceso).

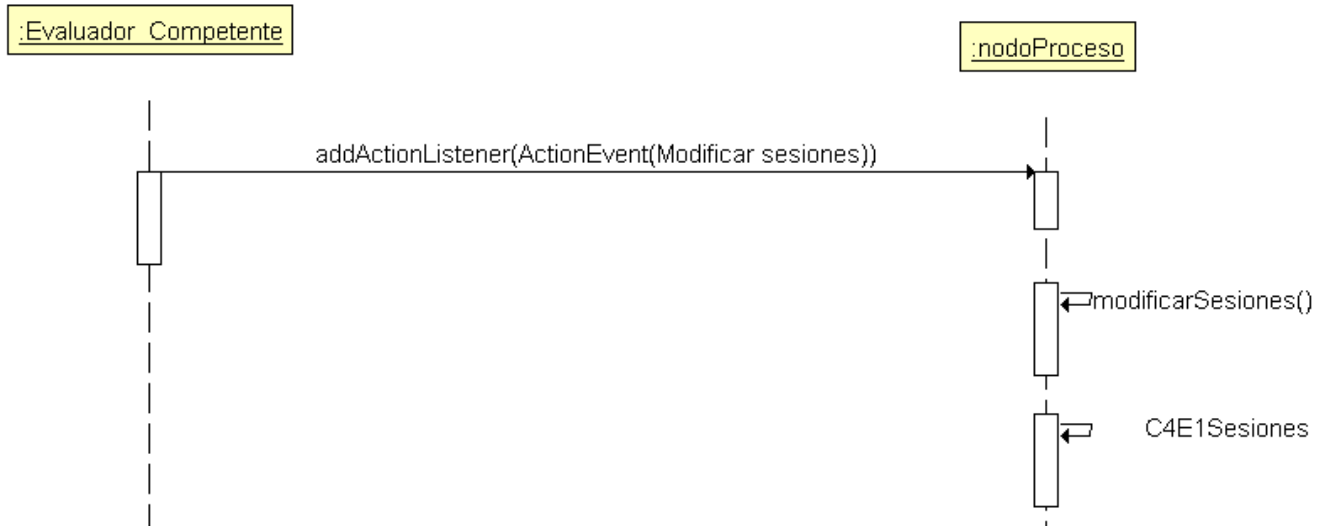


Figura 4.20 Diagrama de secuencia de las clases que intervienen para el manejo de las sesiones de los procesos.

Flujo

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Emite el evento para agregar o editar las sesiones del proceso seleccionado.	2	El sistema valida la modificación que el usuario desea hacer.	C4E1Sesiones

Error

id	Causa del error	Acción
C4E1Sesiones	Información invalida	Envía al usuario un mensaje donde le indica la razón por la cual la información es incorrecta y procura dejar el estado de la interfaz en un estado correcto.

Diagrama de Secuencia para el caso de uso 2.1.2 (Corroborar información recopilada)

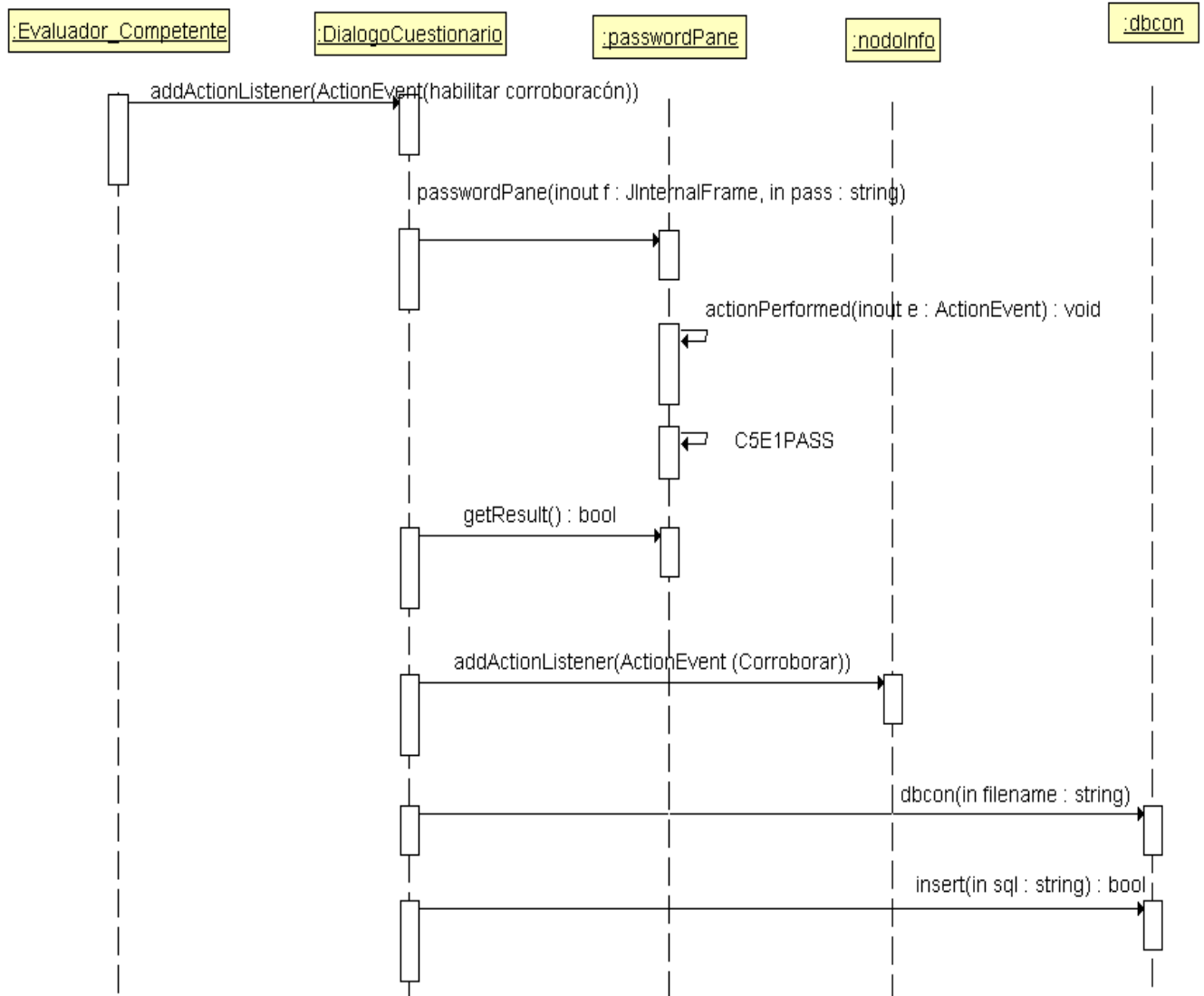


Figura 4.21
Diagrama de secuencia de las clases que intervienen para la corroboración de la información recopilada

Flujo

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Emite el evento para habilitar el dialogo para corroborar la información recopilada.	2	El sistema captura la contraseña del evaluador competente por validar.	
		3	El sistema valida la contraseña proporcionada por el usuario.	C5E1Corr
		4	El sistema realiza la autenticación del usuario.	
		5	Si el usuario es autenticado, la interfaz es habilitada para corroborar y el usuario corrobora la información recopilada.	
		6	La información recopilada es insertada en la base de datos.	

Error

id	Causa del error	Acción
C5E1Corr	Información invalida	Envía al usuario un mensaje donde le indica que la contraseña proporcionada es incorrecta y no habilita la interfaz para poder corroborar la información recopilada.

Diagrama de Secuencia para el caso de uso 2.1.3 (Acceder a la información corroborada)

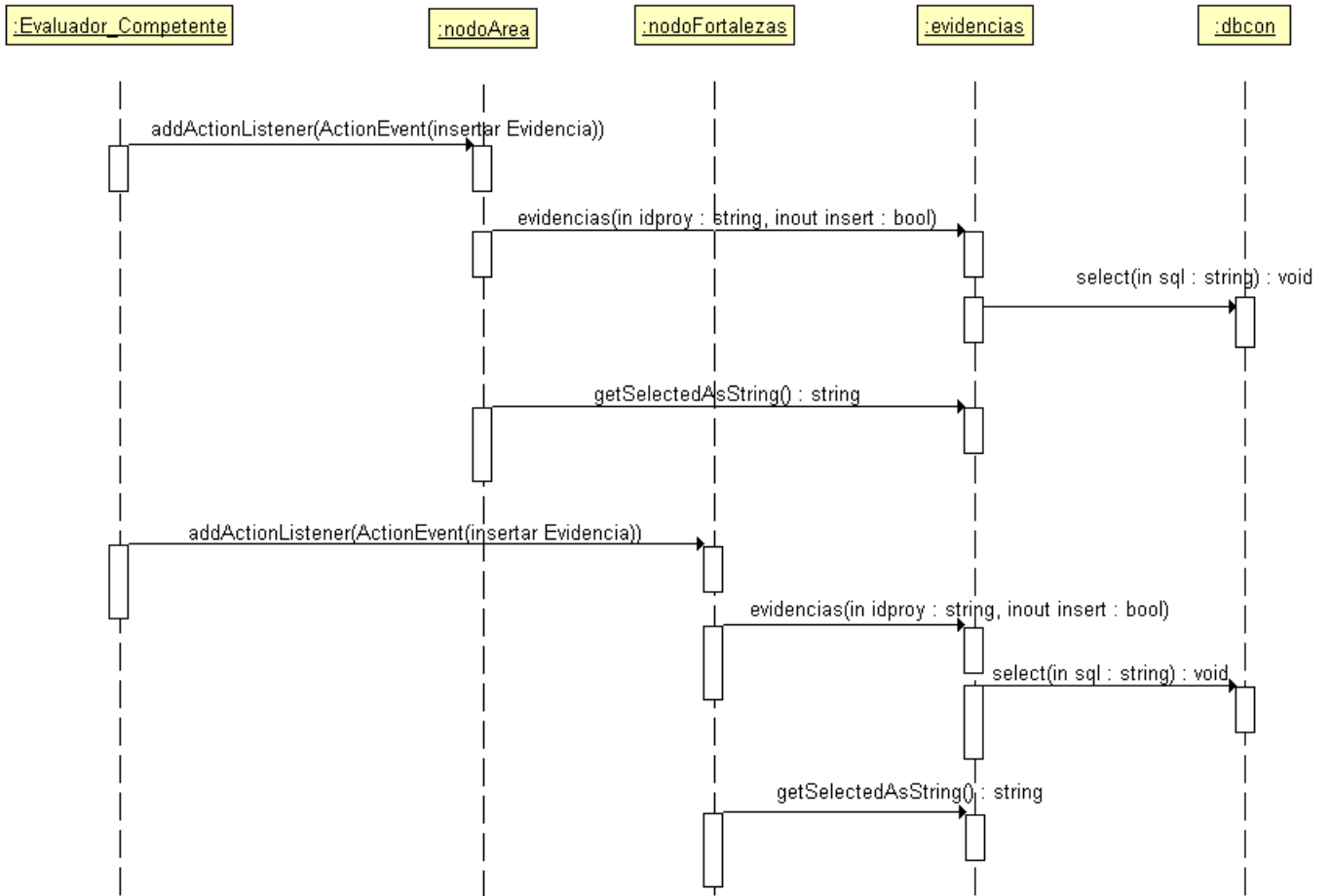


Figura 4.22
Diagrama de secuencia de las clases que intervienen para acceder a la información recopilada

Flujo

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Emite el evento insertar una evidencia en las áreas o fortalezas del proceso.	2	El sistema carga las evidencias almacenadas.	
		3	El sistema muestra las evidencias almacenadas hasta el momento para ser mostradas en la interfaz.	
	El usuario selecciona la evidencia que desea y envía el evento "insertar evidencia".	4	El sistema copia el texto de la evidencia seleccionada por el usuario en el área o fortaleza del proceso.	

Diagrama de Secuencia para el caso de uso 3 (Calificar el cuestionario de evaluación)

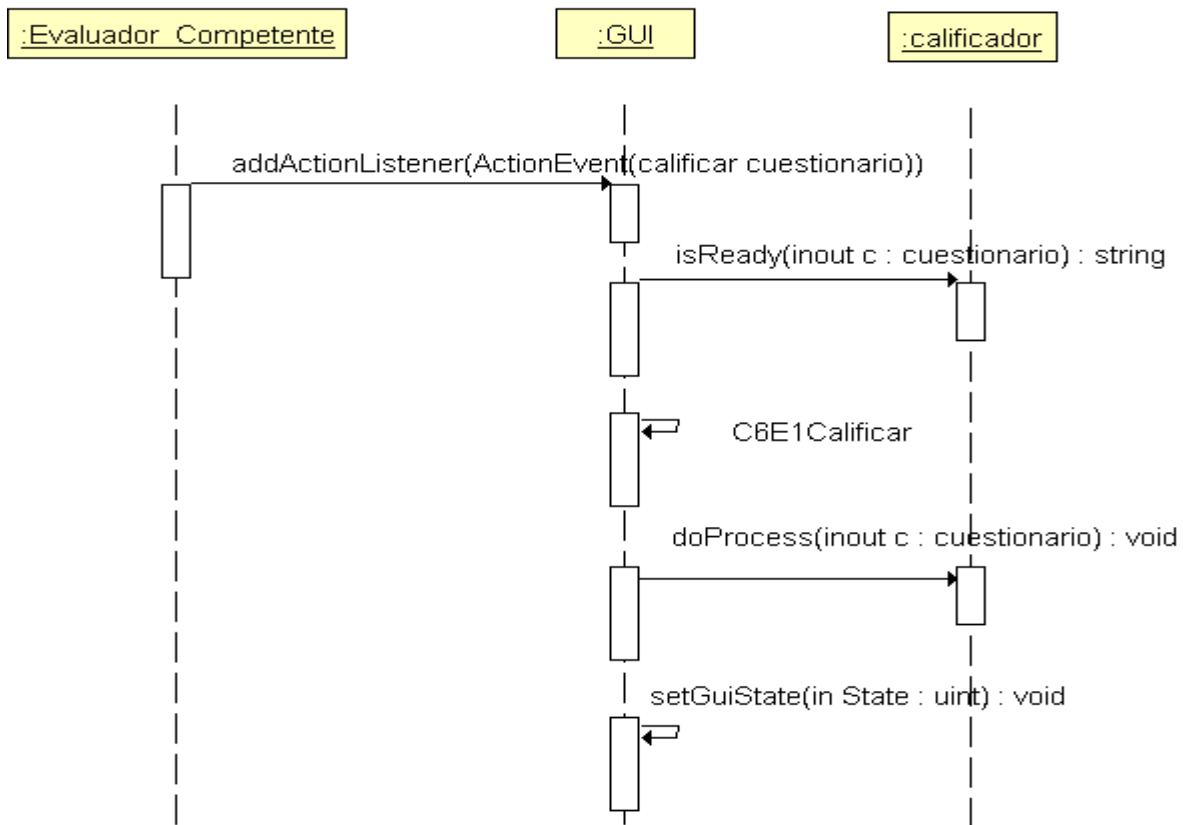


Figura 4.22
Diagrama de secuencia de las clases que intervienen para calificar el cuestionario de evaluación

Flujo

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Emite el evento "calificar cuestionario".	2	El sistema emplea inspecciona el cuestionario de evaluación	C6E1Calificar
		3	En caso de que el cuestionario este preparado para ser calificado, el sistema califica el cuestionario de evaluación.	
		4	El sistema se coloca en el estado "final", que indica que el proceso de calificación del cuestionario ha sido terminado.	

Error

id	Causa del error	Acción
C6E1Calificar	Información invalida	Envía al usuario un mensaje donde le indica que procesos no contienen la información mínima para poder ser calificados.

Diagrama de Secuencia para el caso de uso 4 (Visualizar los resultados de la evaluación)

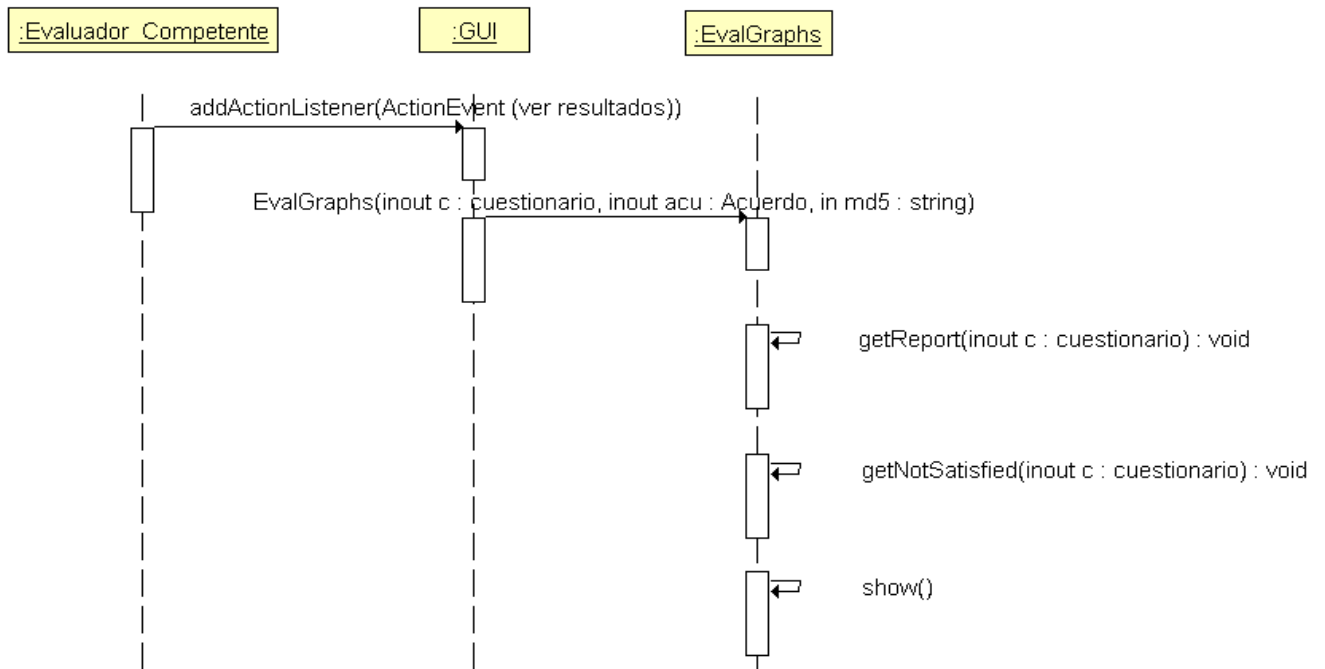


Figura 4.23

Diagrama de secuencia de las clases que intervienen en la visualización de los resultados

Flujo

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Emite el evento “ver resultados”.	2	El sistema construye habilita el componente para mostrar los resultados.	
		3	El componente extrae del cuestionario de evaluación el reporte de calificaciones.	
		4	El componente extrae del cuestionario el reporte de las tareas no satisfechas.	
		5	El componente muestra los resultados en la interfaz.	

Diagrama de Secuencia para el caso de uso 5 y 5.1 (Generar reporte de la empresa evaluada y el reporte para el Organismo verificador, anexas el tiempo estimado empleado en las entrevistas y las tareas no satisfechas)

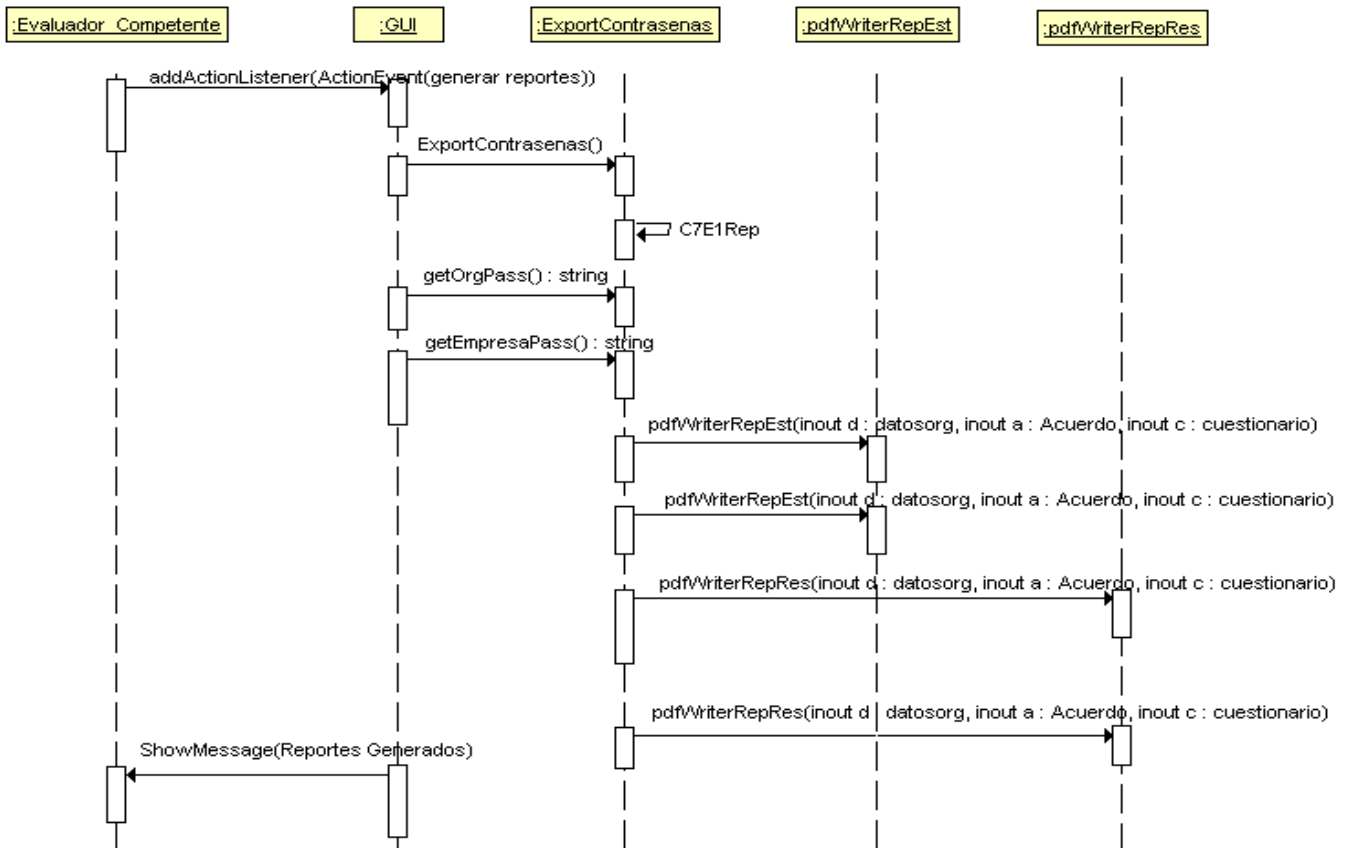


Figura 4.24

Diagrama de secuencia de las clases que intervienen para la generar los reportes de la evaluación.

Flujo

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Emite el evento “generar reportes”.	2	El sistema muestra el componente para capturar las contraseñas.	
		3	El componente captura la contraseña de cada reporte por generar.	C7E1Rep
		4	Si las contraseñas son correctas, el sistema solicita las claves para los reportes por generar	
		5	El sistema proporciona la información al componente para generar el reporte estadístico.	
		6	El componente genera el reporte estadístico.	
		7	El sistema proporciona la información al componente para generar el reporte de resultados.	
		8	El componente genera el reporte de resultados.	

Error

id	Causa del error	Acción
C7E1Rep	Información invalida	Envía al usuario un mensaje donde le indica que contraseña no es correcta.

Diagrama de Secuencia para el caso de uso 6 (Realizar el procedimiento para el control de cambios)

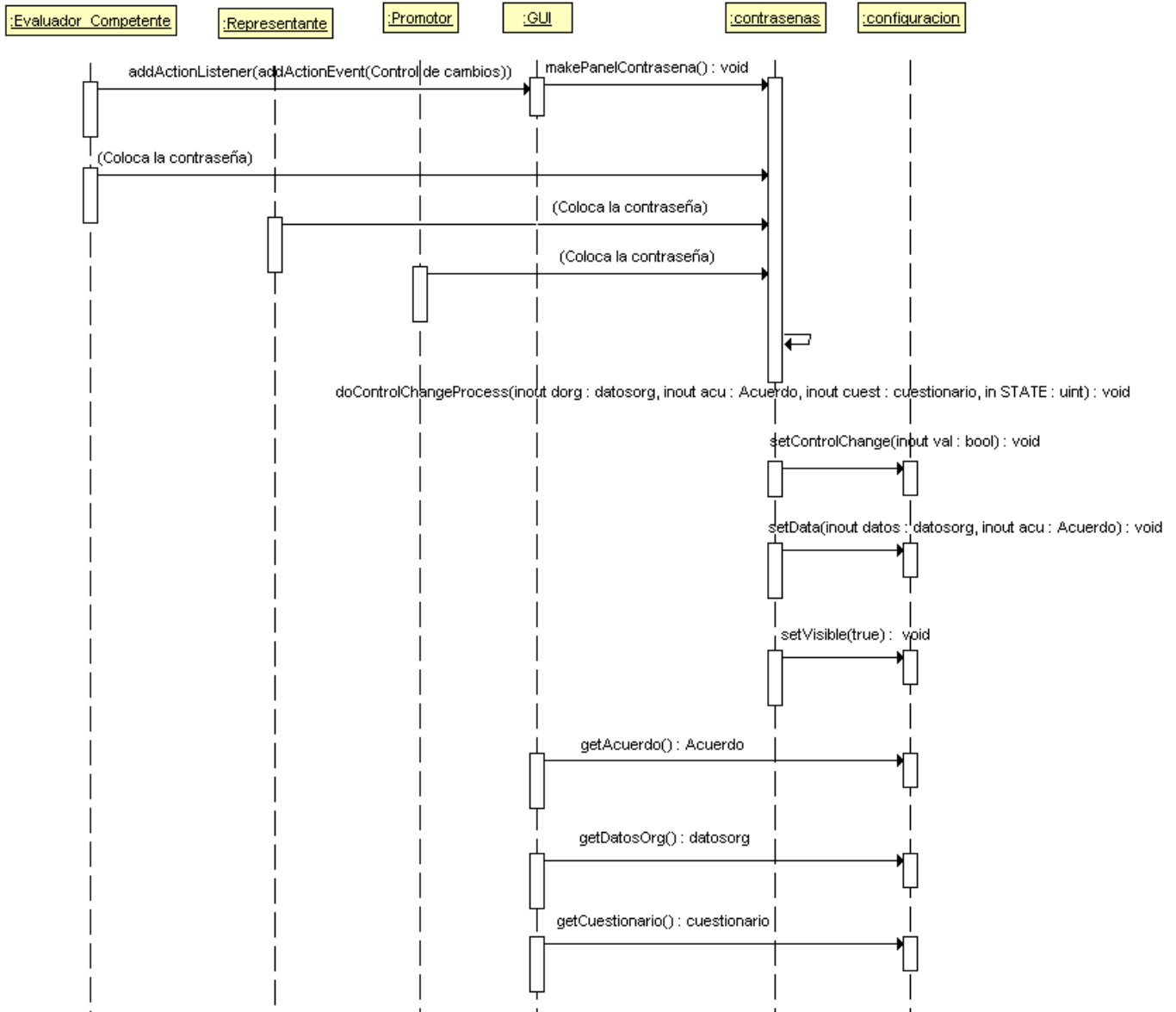


Figura 4.25
Diagrama de secuencia de las clases que intervienen en el proceso de control de cambios

Flujo

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	El evaluador competente emite el evento "Control de cambios".	2	El sistema construye el componente para capturar las contraseñas para ejecutar el control de cambios.	
3	El evaluador competente digita su contraseña	4	El componente captura la contraseña del evaluador competente.	C8E1CC
5	El representante digita su contraseña	6	El componente captura la contraseña del representante.	C8E1CC
7	El promotor digita su contraseña	8	El componente captura la contraseña del promotor.	C8E1CC
		9	Si las contraseñas son correctas, el componente realiza el proceso de control de cambios y regresa el control al sistema.	
		10	El sistema se coloca en el estado que indica que se acaba de realizar el proceso de control de cambios.	
		11	El sistema le proporciona al componente indicado los datos que se tienen actualmente en el sistema para que puedan ser modificados.	
		12	El sistema obtiene los nuevos datos modificados.	

Error

id	Causa del error	Acción
C8E1CC	Contraseña invalida	Envía al usuario un mensaje donde le indica que contraseña no es correcta.

4.5.- DIAGRAMAS DE COMPONENTES

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente, etc. Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente.

Diagrama general de componentes del código fuente

El sistema está dividido en 6 grandes componentes que muestran como está organizado físicamente el código fuente de las clases de java que componen Chilam, las clases se encuentran en los mismos componentes siguiendo un orden funcional con el que estas clases fueron creadas.

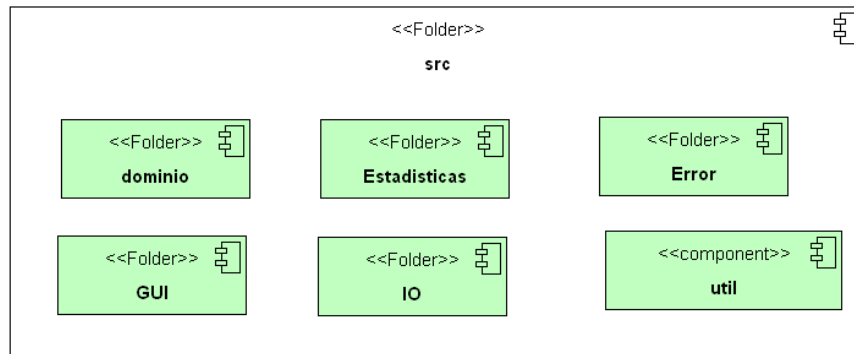


Figura 4.26
Diagrama de componentes del código fuente de Chilam.

Diagrama de componentes del componente "dominio"

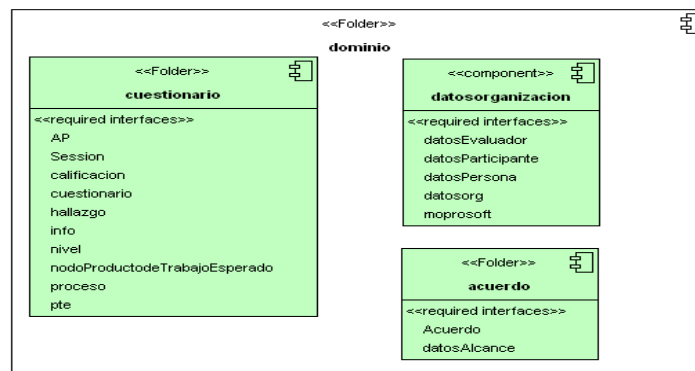


Figura 4.27
El paquete "dominio" contiene los componentes que gestionan la información elemental que maneja Chilam.

Diagrama de componentes del componente "estadísticas"

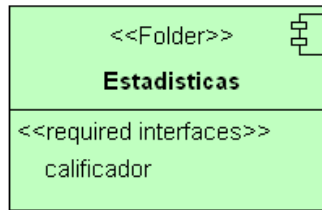


Figura 4.28
El paquete "estadísticas" contiene los componentes que permiten llevar a cabo la calificación del cuestionario de evaluación.

Diagrama de componentes del componente "Error"

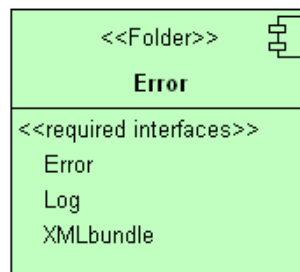


Figura 4.29
El paquete "Error" contiene los componentes que gestionan los casos excepcionales que ocurren en el sistema.

Diagrama de componentes del componente "GUI"

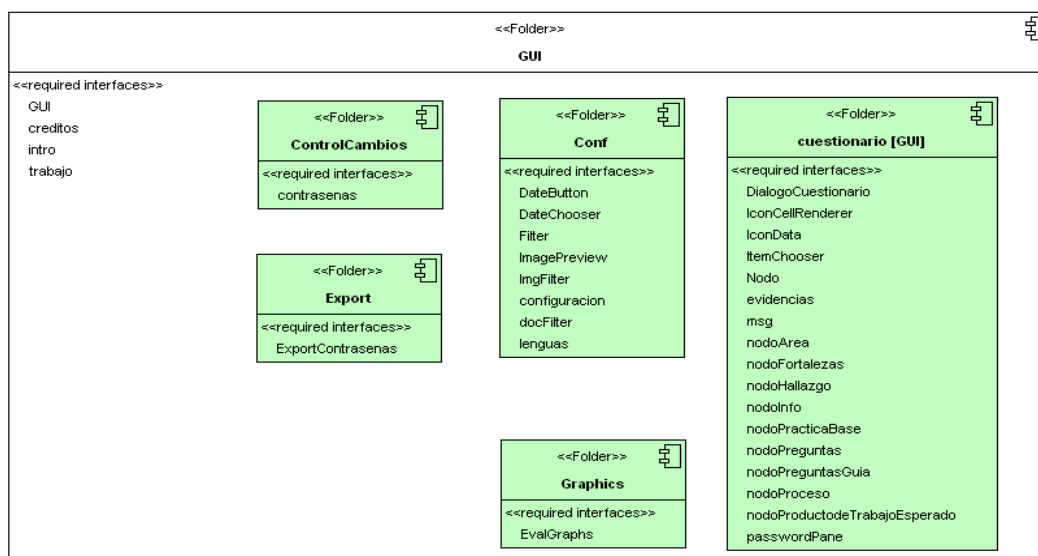


Figura 4.30
El paquete "GUI" contiene los componentes que interactúan Directamente con el usuario para llevar a cabo la evaluación.

Diagrama de componentes del componente "util"

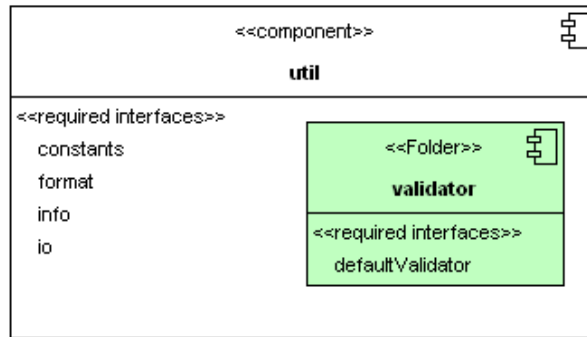


Figura 4.31
El paquete "util" contiene los componentes que proveen funcionalidades generales para el sistema.

Diagrama de componentes del componente "IO"

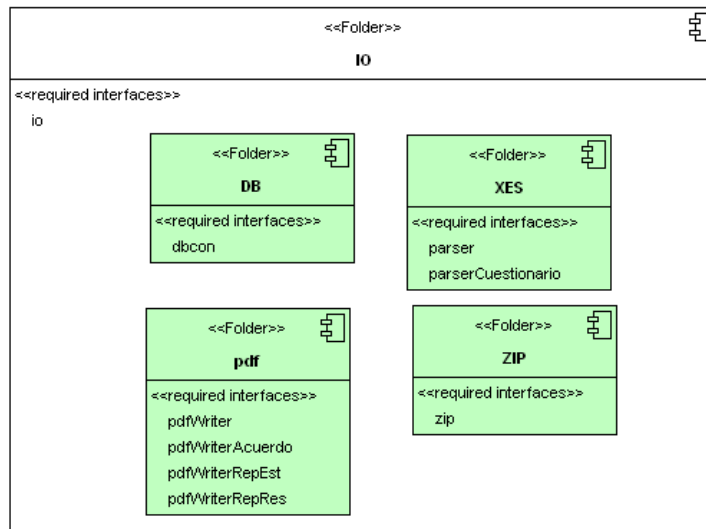


Figura 4.32
El paquete "IO" contiene los componentes que cargan y almacenan la información requerida por el sistema.

5.- DETALLE DE LAS CAPAS DE LA APLICACIÓN

El Modelo de n-capas, es un método utilizado en Ingeniería de software, para dividir la aplicación en capas. Cuando $n = 3$, tenemos un modelo de 3 capas, donde cada nivel tiene un tipo de abstracción diferente al de los demás, aunque cada capa tiene comunicación con las demás.

5.1 DETALLE DE LA CAPA DE ALMACENAMIENTO

En esta capa se lleva a cabo la carga y almacenamiento de toda la información que requiere Chilam, para poder auxiliar a los evaluadores durante el proceso de evaluación de la empresa.

Implementación

Para implementar ésta capa se usa una clase que llama a las demás, la cual es la encargada de leer y escribir los proyectos (la descripción de la estructura de un proyecto que utiliza el sistema es especificada mas adelante) mediante el cual se lleva el control de las evaluaciones. Esta clase lee y escribe archivos en formato XML. El proceso de lectura de un proyecto del sistema se lleva a cabo de la siguiente manera.

La clase "parser" lee y valida con base en un DTD (Document Type Definition), el cual se escribe justo antes de su lectura para evitar que se pueda editar este documento antes de abrirlo con Chilam. Cada documento XML contiene 2 encabezados fundamentales para llevar a cabo la validación:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE acuerdo SYSTEM "alcance.dtd">
```

La primer línea indica la versión de XML que se está usando y la codificación de los caracteres que se usa en el contenido del documento. La segunda es una instrucción de proceso que indica la locación del documento "alcance.dtd", el cual contiene la definición de la estructura del documento xml que se desea validar y la etiqueta principal del documento, que debe ser la primera etiqueta que debe de contener el archivo xml y cuya estructura está definida precisamente en el DTD.

Un documento DTD puede definir la estructura de un archivo xml, es decir, este documento puede definir el nombre, orden, atributos, valor de los atributos y la cardinalidad de cada etiqueta que es utilizado en el documento xml.

Este tipo de validaciones se llevan a cabo para validar que el documento contenga la información necesaria para el proceso que se desea implementar.

Nombre del DTD	Archivo xml que está definido por el DTD	Descripción del contenido del archivo xml
alcande.dtd	Proyecto.ae.xml	Este archivo contiene la información del alcance de la evaluación.
pep.dtd	Proyecto.pep.chi	Este archivo contiene el estado del proyecto y el nombre de los archivos que forman parte del proyecto.
proceso1.dtd	Proyecto.abreviación del proceso.xml	Este archivo contiene las tareas del proceso definidas por EvalProSoft.
datorg.dtd	Proyecto.do.xml	Este archivo contiene los datos de la organización

Tabla 5.1
Archivos DTD que definen la estructura de los archivos XML
Que almacenan la información del proyecto.

Al validar el documento con un DTD se asegura que la información que se requiere está contenida, pero no asegura que la información es valida, es decir, asegura la existencia de la información, pero no asegura que la información es correcta.

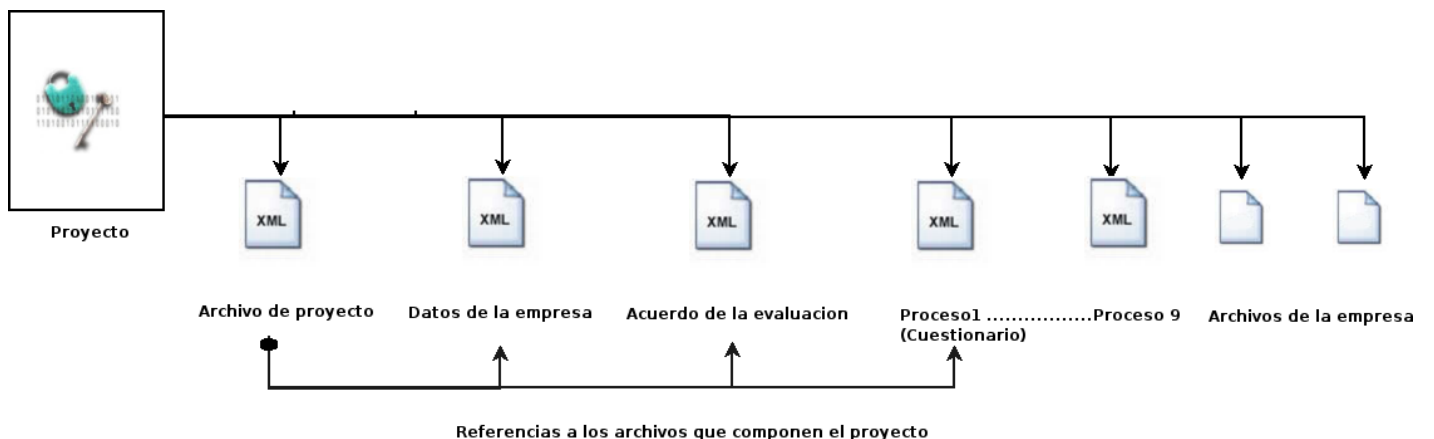
Una vez que la información ha sido leída y validada se vacía el contenido de los objetos DOM que se obtienen al leer un archivo XML a los objetos del dominio del problema propios de la clase "parser", y se eliminan los objetos DOM, ya que estos no son útiles después de este punto de la lectura del proyecto. Cada objeto del dominio del problema posee la capacidad de cifrar y descifrar la información que éste contiene. Esta capacidad es utilizada una vez que se ha pasado la información del árbol DOM a los objetos del dominio.

Una vez teniendo el texto claro, ya que el DTD no valida el contenido de los objetos , se utiliza la clase "defaultValidator" del paquete "util", clase que tiene a su cargo la validación de los datos contenidos en las clases del dominio del problema para asegurar que la información es correcta.

En el caso de los procesos del cuestionario de evaluación se tiene una característica especial, cada información corroborada del proceso debe de ser ingresada a la base de datos, razón por la cual cuando se termina el proceso de validación se ingresa toda la información en la tabla de base de datos para que posteriormente ésta pueda ser referenciada desde cualquier punto de la interfaz.

Estructura de proyectos del sistema

Un proyecto del sistema (Figura 5.2) está conformado por varios archivos XML y archivos requeridos a la empresa como es el organigrama de la empresa, entre otros. Para dar una organización a estos archivos se diseñó la estructura organizacional de un proyecto de la siguiente manera:



Referencias a los archivos que componen el proyecto
 Figura 5.2
 Conformación de los archivos que forman parte de un proyecto.

Todos los archivos que conforman el proyecto contendrán la información cifrada por motivos de seguridad. El tipo y algoritmo de cifrado de los archivos con formato XML se describe mas adelante. Estos archivos quedan almacenados dentro de una carpeta generada con el nombre que el usuario ha decidido.

Todos los documentos contendrán como nombre, el nombre del proyecto seguido de un punto, seguido de la abreviación para saber el contenido del archivo mediante el nombre (tabla 5.3), seguido de un punto seguido del formato en que se encuentra la información(XML).

Abreviatura	Descripción de la abreviación
pep	Proyecto EvalProSoft
do	Datos de la organización
ae	Alcance de la evaluación
GN	Gestión de Negocios
GPR	Gestión de PRocesos
GR	Gestión de Recursos
RHAT	Recursos Humanos y Ambiente de Trabajo
BSI	Bienes, Servicios e Infraestructura
CO	Conocimiento de la Organización
GPY	Gestión de ProYectos
APE	Administración de Proyectos Específicos
DM	Desarrollo y Mantenimiento

Tabla 5.3
Abreviatura asociada con cada proceso.

Durante el inicio de la evaluación el proceso de evaluación requiere de varios archivos que debe de proporcionar (tabla 5.4) la empresa evaluada. Algunos no son imprescindibles otros por el contrario, si lo son. Estos archivos son referenciados en el archivo que contiene los datos de la organización, copiados, cifrados y almacenados en la carpeta del proyecto. Estos archivos son renombrados de tal modo que al nombre del archivo original se le concatena la extensión sfh (secure file Handler, proyecto que es descrito mas adelante).

Archivo	Indispensable	Tipo de archivo soportado	
Organigrama de la organización	Si	Portable Network Graphics	.png
		Joint Photographic Experts Group	.jpeg
		Graphics Interchange Format	.gif
		Bitmap	.bmp
Inventario de proyectos que se revisarán durante la evaluación	No	Cualquier tipo de documento que pueda almacenar texto.	
Plan de mejora de la organización	No	Cualquier tipo de documento que pueda almacenar texto.	

Tabla 5.4
Información proporcionada por la organización.

Por ejemplo, un proyecto puede estar conformado por los archivos nombrados a continuación:

1. Proyecto.do.xml (Archivo que contiene los datos de la organización)
2. Proyecto.ae.xml (Archivo que contiene el alcance de la evaluación)
3. Proyecto.pep.xml (Archivo que contiene los datos del proyecto EvalProSoft)
4. Proyecto.GN.xml (Archivo que contiene el proceso Gestión de Negocios)
5. diagrama.png.sfh (Archivo cifrado que contiene el diagrama de la evaluación)
6. proyectos.doc.sfh (Archivo que contiene el inventario de proyectos que serán evaluados)

Archivo de proyecto:

Este archivo contiene las referencias a los documentos que se requieren, el acuerdo de la evaluación, el cuestionario de evaluación.

Para este archivo se asignará la extensión “.pep.xml”, (Proyecto de EvalProSoft).

Datos de la empresa:

Este archivo contiene los datos de la empresa, así como los archivos que le son requeridos para la evaluación (tabla 5.4).

Para este archivo se asignará la extensión “.do.xml”, (Datos de la organización).

Acuerdo de la evaluación:

Este archivo contiene los procesos, niveles que se desean alcanzar, niveles objetivo, comentarios, unidad de producción, unidad administrativa y el nombre del archivo que contiene el proceso.

Para este archivo se asignará la extensión “.ae.xml”, (Acuerdo de evaluación).

Conección a Base de datos

Durante el proceso de calificación del cuestionario de evaluación, se requiere que la información recopilada que ha sido corroborada pueda ser referida desde cualquier proceso, en particular, desde cualquier práctica base y desde cualquier sección de los hallazgos (Fortalezas ó áreas de oportunidad); esta información es vital para poder asignar una calificación en las prácticas base, razón por la cual es necesario proveer un mecanismo que permita a esta información ser referida de manera independiente del proceso donde fue ingresada y corroborada pero también se pueda saber la procedencia de la evidencia.

Otro aspecto importante de las evidencias, es el hecho que puede ser información muy grande y el recorrer el árbol que presenta la interfaz cada momento tomaría un costo importante en el rendimiento del sistema, así que es necesario proporcionar un método de acceso eficiente a esta información.

Para proporcionar una solución a estos aspectos se utiliza el proyecto SQLite (proyecto descrito más adelante), que a grandes rasgos permite manejar la información almacenada en un archivo como se accede a una base de datos relacional.

Las evidencias que son corroboradas en la interfaz son ingresadas a una base de datos (Figura5.5), que permite acceder a todas estas evidencias de manera independiente y optimizando el manejo de las mismas. Esta base de datos almacena la evidencia pero también asocia la procedencia de la evidencia lo cual permite acceder a la base de datos y ejecutar consultas que permitan ordenar y referenciar esta información.

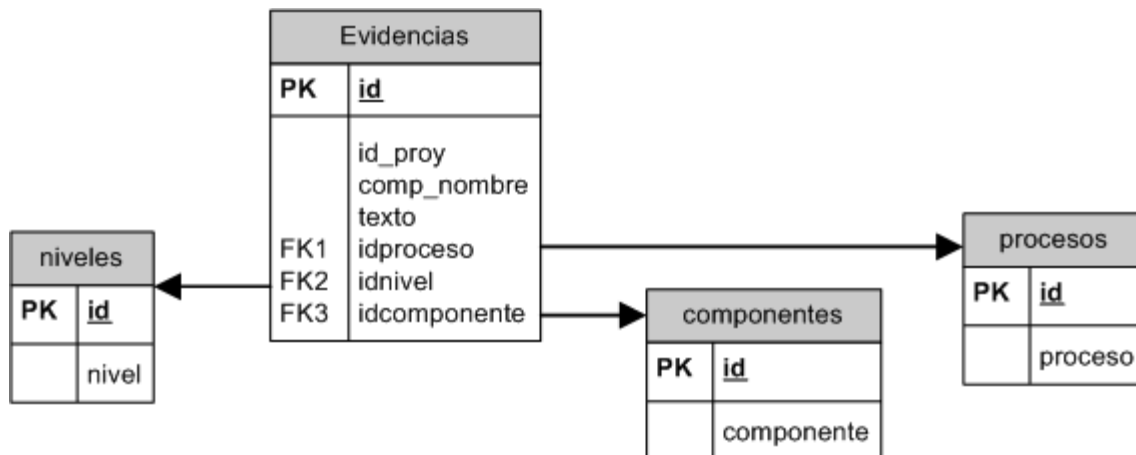


Figura 5.5
Diagrama de tablas de la base de datos implementada
Para el manejo de las evidencias de los procesos.

Para poder acceder a esta base de datos, se diseñó la clase “dbcon” que se encarga de conectarse a la base de datos del sistema. La base de datos reside en un archivo llamado “ev3.db” y esta clase se encarga de ejecutar las consultas a la base de datos. Las consultas escritas son un conjunto de sentencias en SQL reducida, es decir, este proyecto no soporta todo el conjunto de instrucciones del estándar SQL92, razón por la cual, el proyecto provee una documentación donde se describe el conjunto de instrucciones que si soporta.

De este modo se logra separar de manera independiente las evidencias y a su vez asocia las evidencias al proceso, nivel y componente en el cual esta evidencia fue encontrada. De esta forma la interfaz gráfica provee las funcionalidades necesarias para que el usuario pueda referenciar a las evidencias desde cualquier punto de la interfaz.

Cabe señalar que este proyecto no está diseñado para soportar grandes cantidades de información, razón por la cual se implementa el siguiente mecanismo para dar mantenimiento a la base de datos:

- 1) Cuando se abre un proyecto nuevo se genera la llave de la base de datos correspondiente.
- 2) Se borran todas las evidencias relacionadas con el proyecto que pudiesen existir en la base.
- 3) Se cargan las evidencias obtenidas a partir de los documentos XML leídos.
- 4) Cuando el proyecto es cerrado las evidencias en la base de datos asociadas con su llave se borran.

La llave de la base de datos está conformada por el nombre del evaluador competente, el nombre de la empresa evaluada y la fecha de inicio de la evaluación, por lo que la llave es única, sin importar las evaluaciones que se lleven a cabo.

Estructura de una llave de base de datos:

Nombre_Empresa_Nombre_evaluador_competente_Fecha_inicio.

Ejemplo:

Softmake_ArturoJimenez_11/05/2006

Cabe señalar que todo el acceso a la base de datos es realizado por completo por la clase "dbcon".

Compresión de archivos

La compresión de archivos es utilizada cuando se tiene que ejecutar el proceso de control de cambios (descrito más adelante). Los archivos son listados por la clase y posteriormente cuando la lista se tiene por completo se comprimen con el formato zip.

La clase "zip" provee los métodos para poder comprimir y descomprimir archivos con formato zip aunque en el sistema sólo se requiere la compresión de archivos.

Generación de documentos PDF.

Para el sistema se decidió generar documentos con formato PDF para manejar información diversa, en particular en 2 fases:

1.- Cuando se está capturando la información de la organización, se requiere de la escritura de los "datos recabados" y "el acuerdo de evaluación" que será firmado por los involucrados en la evaluación.

2.- Cuando el proceso de calificación del cuestionario de evaluación ha concluido, se deben de generar los reportes con los resultados de la evaluación para el Organismo verificador y para la empresa evaluada.

Para proveer estas funcionalidades el paquete "pdf" provee las clases que llevan a cabo estas tareas, proporcionando a su vez un mecanismo de seguridad, ya que los documentos quedan condicionados a una contraseña que es requerida previamente para poder ser abiertos.

De este modo queda resguardada la información contenida en los documentos.

Además de estos paquetes existe la clase "io" que provee los mecanismos para poder copiar documentos de un lugar a otro, escribir texto a documentos entre otras funciones generales que involucran una salida de datos así como el método base para cifrar o descifrar cadenas de texto. Esta clase es utilizada por varias clases para diversos propósitos.

DETALLE DE LA CAPA LÓGICA DE LA APLICACIÓN

Esta capa del sistema está diseñada para gestionar el comportamiento global del sistema, entre las tareas que se llevan a cabo en esta capa es la gestión de la información que conforma la evaluación, la gestión de errores etc.

Manejo de errores

Cuando una sección del sistema detecta algo que no es correcto, ésta debe informar del evento a la sección más alta del sistema que es la interfaz para que pueda llevar a cabo la manipulación del mismo, esto se logra mediante la gestión de errores.

Cuando un componente del sistema detecta un error debe de indicar que error ha ocurrido, qué módulo detecta el error y si este error está asociado con una descripción que permita al usuario saber la causa del mismo.

Para satisfacer estas necesidades la clase "Error" proporciona los medios para lograrlo.

Cuando el componente detecta el error crea un objeto de la clase "Error", al cual se le indica:

- 1) Número de error detectado.
- 2) Módulo emisor del error.
- 3) Una bandera que indica si la descripción del error debe de ser mostrado al usuario final.
- 4) Información extra que proporcione mayor información acerca del estado del componente en el momento en que ocurre el error.
- 5) Clase y función en que el error es lanzado.
- 6) Cadena de errores que java genera al momento de detectarse el error en caso de que se posea.

La clase "Error" extiende a la clase "Exception", por lo cual es posible manejar esta clase como una excepción de java. Esto significa que se puede lanzar el objeto creado con la información del error a través de los métodos que intervienen en el proceso cuando el error ocurre. Cuando una clase detecta que puede ocurrir un error esta especifica en el encabezado del método que lanza una excepción del tipo "Error", lo que significa que el método en el que la misma es utilizada debe de atrapar la excepción. Al momento de atrapar la excepción ésta coloca el nombre de la clase y la firma del método donde se atrapó la excepción lanzada con la siguiente sintaxis:

->[Nombre de la clase].firma del método y lanza la excepción para que la información de la excepción se envíe hasta la interfaz gráfica la cual atrapa esta excepción y le indica a la misma que gestione el tipo de error detectado.

Para lograr una gestión mas robusta del manejo de errores la clase error identifica el módulo emisor del error y busca su descripción en el catálogo de errores que corresponde al módulo emisor; con esta información envía el mensaje al usuario si es que así indica el mismo y posteriormente se escribe esta información junto con la cadena que manda java, la información complementaria proporcionada en el momento en que se genero la excepción y la serie de funciones que intervinieron en el proceso al archivo log.

Archivo log

Este archivo de texto se utiliza para escribir el reporte de errores que son detectados por Chilam y sirve para resguardar la mayor cantidad de información posible que permita a los desarrolladores un mejor control del sistema. Este archivo va guardando cada una de las excepciones que ocurren durante un día, ya que el archivo se genera en la carpeta "logs", donde Chilam es instalado y es nombrado con la fecha del día en que ocurre el error, seguido de la extensión "log".

Por ejemplo:

"9-abril-2006.log".

Cada mensaje enviado a este archivo es precedido por un separador en el que se indica el nombre del día, la fecha y hora en que fue enviado el mensaje y el carácter de este, es decir si es un mensaje de error o es una cadena enviada al archivo por el desarrollador.

Por ejemplo:

```
*-----Domingo 9-Abril-2006 11:32:141 AM-----Error
*-----Domingo 9-Abril-2006 11:32:141 AM-----Cadena de Debug
```

La primera línea ejemplifica el encabezado de un mensaje que es enviado al detectarse un error y la segunda ejemplifica el encabezado de un mensaje específicamente enviado por el desarrollador; de esta forma se asegura la permanencia de la información que permite al desarrollador hacerse de mayor información durante el desarrollo y posteriormente en el mantenimiento del software.

Catálogo de errores

Cada módulo que interviene en el sistema tiene asociado un archivo con formato xml que asocia el número de error con su descripción (tabla 5.6). Cada módulo puede llevar un catálogo de errores independiente ya que la clase "Error" puede localizar su catálogo gracias al número de módulo que le es indicado cuando ésta es construida. Los catálogos de error de los módulos se encuentran dentro de la carpeta "ecatálogo" donde Chilam queda instalado.

Nombre del archivo	Nombre del módulo
ifaz.xml	Interfaz gráfica
estad.xml	Estadísticas
errores.xml	Manejador de errores
seg.xml	Dominio del problema
es.xml	Entrada - salida
validator.xml	Validador

Tabla 5.6
Nombre de los catálogos de errores para cada componente del sistema.

Dominio del problema

Esta sección corresponde a la base del sistema ya que asocia la información requerida durante el proceso de evaluación con clases que permiten manejar y presentar esta información en el sistema.

La información requerida por el proceso de evaluación se dividió en 3 secciones.

- 1) Datos de la organización
- 2) Cuestionario de evaluación
- 3) Alcance de la evaluación

Datos de la organización

Para el manejo de esta información se desarrolló el objeto "datosorg" y es aquí donde se guardan el nombre de la empresa por ser evaluada, la información de los involucrados durante el proceso de evaluación, los nombres de los documentos de la empresa evaluada que son utilizados en la evaluación, la relación con MoProSoft etc.

Cuestionario de evaluación

Para el manejo de esta información se generó el objeto "cuestionario" que está conformado por una lista de objetos de la clase "proceso", cada proceso posee un diseño arbóreo, ya que el documento donde se describe la información involucrada de cada proceso del cuestionario de evaluación especificado por EvalProSoft así lo muestra.

La estructura de la clase "proceso" es reflejo del documento a excepción que esta clase posee una lista de sesiones que son necesarias para el proceso de edición del cuestionario de evaluación (Figura 5.7).

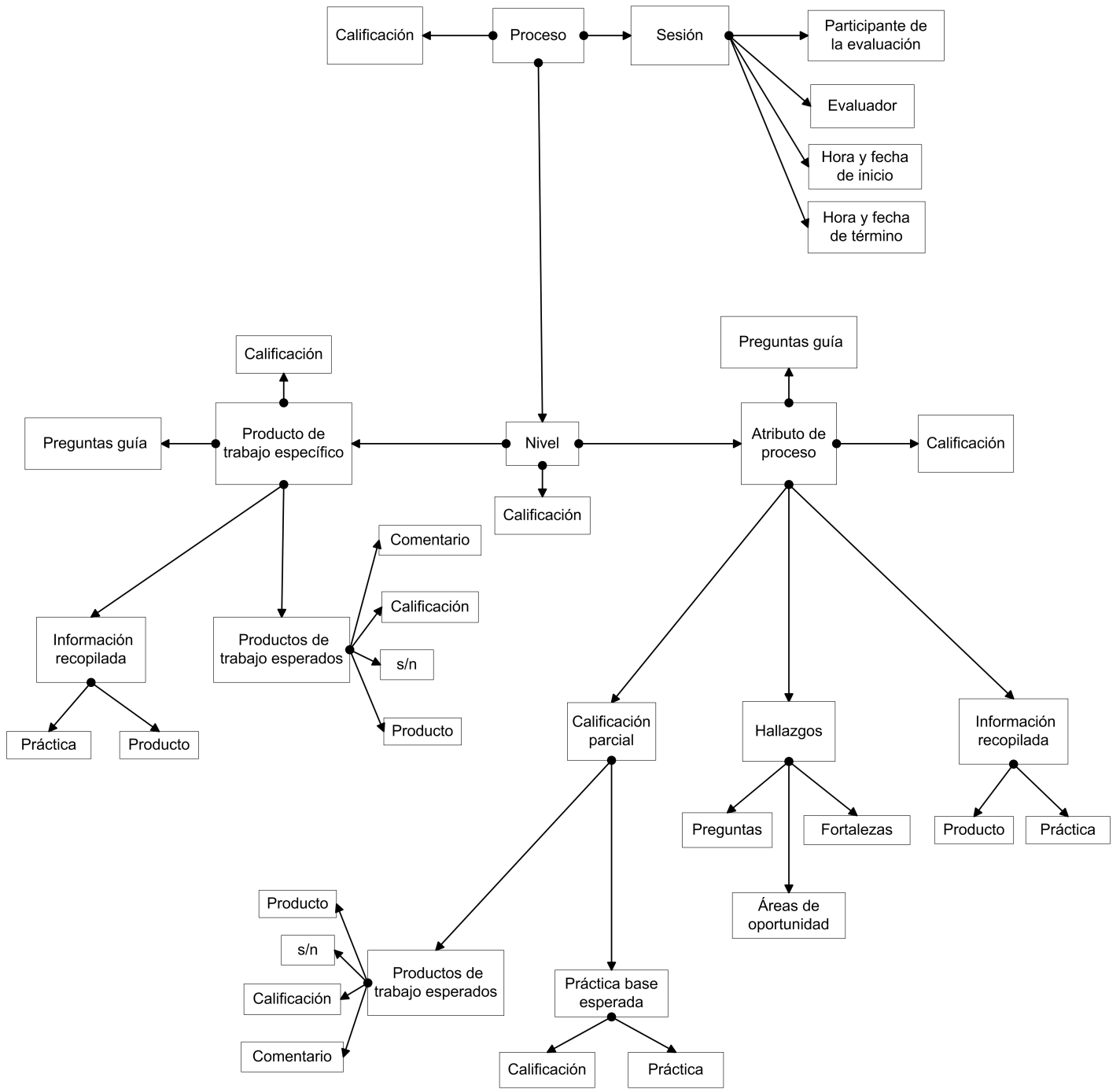


Figura 5.7
Estructura lógica de la información de cada proceso.

Alcance de la evaluación

Esta sección del dominio almacena el proceso a ser evaluado, el nivel deseado, el nivel objetivo, la unidad administrativa, la unidad de producción, comentarios y el nombre del archivo del proceso.

Estas 3 clases conforman la base del sistema ya que es aquí donde se almacena la información vital del proceso de evaluación y por estas razones fueron diseñadas de manera modular, es decir, cada una de ellas está conformada por otras clases que almacenan secciones independientes de información. De este modo el cambio en la

información afecta específicamente a pequeñas clases, de tal modo que los cambios son localizables y no afectan significativamente a las clases que hacen uso de ellas, lo que hace a este sistema escalable.

Cada clase implementa los métodos que permiten colocar y obtener las secciones de información que se requieran y son utilizadas en las 3 capas para mostrar, editar y conformar los documentos que son requeridos, como es el caso de los reportes estadísticos.

Cada clase que requiera hacer uso de las clases del dominio del problema, puede tener una instancia para realizar las funcionalidades que se requieran, pero la información final es almacenada en las instancias principales, las cuales quedan resguardadas en la interfaz gráfica. Cada clase contenida en este paquete del dominio tiene la capacidad de cifrar y descifrar su contenido, para ello hace uso de una llave que le es indicada cuando se ejecuta esta funcionalidad. El algoritmo que se usa para realizar el cifrado y descifrado está definido en otra clase, de tal modo que si se desea cambiar el algoritmo de cifrado, basta con implementarlo y usarlo, permitiendo independizar el método de cifrado y descifrado de las clases, ya que no es necesario realizar el cambio de algoritmo por cada clase, lo que brinda al sistema independencia de funcionalidades y permite la modificación de manera localizada y sencilla.

Calificación del cuestionario de evaluación

Una vez que el cuestionario ha sido calificado por el usuario, se procede a aplicar la calificación que no se explica en este documento pero que se encuentra definida por EvalProSoft. Esta clase implementa un recorrido del árbol que contiene la información del cuestionario para obtener estas calificaciones y calificar cada proceso, de tal modo que cada atributo de proceso, producto de trabajo específico, nivel y proceso quedan con una calificación asentada al final del proceso. Este proceso deja a la información del cuestionario en condiciones para conformar los reportes estadísticos y obtener resultados concretos de la evaluación.

Antes de comenzar el proceso de calificación del cuestionario de evaluación se deben de haber cerrado todas las sesiones asignadas a cada proceso, razón por la cual se verifica que las sesiones sean cerradas antes de proceder con la calificación del cuestionario de evaluación.

Métodos útiles para el funcionamiento del sistema

Esta sección contiene todos los métodos que son utilizados por distintas clases en diferentes paquetes del sistema, como son: los métodos para verificar números telefónicos, nombres, contraseñas, correos electrónicos, sistema operativo en que está corriendo la aplicación, etc.

Estos métodos son definidos en diversas clases pero todos ellos son estáticos, es decir, no es necesario declarar el objeto para acceder al método en específico.

Dentro de los objetos de esta sección se encuentra el objeto que se encarga de validar los datos descifrados después de haber realizado una lectura del proyecto. Este objeto se encarga de verificar que todos los campos ingresados de la lectura de los archivos sean coherentes, como es el caso de los nombres de los participantes, evaluadores, fechas de inicio y de término etc.

Estas interfaces fueron diseñadas para proporcionar al usuario final la mayor guía posible, tratando de evitar que el usuario realice operaciones para los cuales no se tienen las precondiciones satisfechas. Estos puntos se reflejan en todas las ventanas gráficas mostradas en el sistema.

DETALLE DE LA CAPA DE PRESENTACIÓN

Esta sección del sistema es la encargada de facilitar el proceso de evaluación, en ella se encuentran todos los objetos que muestran diálogos de interacción con el usuario. En esta capa es donde los objetos principales del dominio del problema son almacenados y los errores son gestionados.

La interfaz está diseñada para que funcione como un ambiente de trabajo, donde todas las ventanas están contenidas en un marco de trabajo y toda la información es desplegada dentro del mismo.

La ventana principal (figura 5.8) se encarga de brindar las funcionalidades del sistema y provee los componentes gráficos para que el usuario pueda abrir proyectos, guardar proyectos, llevar a cabo el control de cambios,

desplegar el cuestionario de evaluación, calificar el cuestionario de evaluación, ver los resultados del cuestionario de evaluación y generar los reportes finales.

La resolución mínima requerida para desplegar las interfaces en esta versión es de 1024 x 768, puesto que la información que se requiere desplegar en el cuestionario de evaluación y la recopilación de los datos mínimos para generar un proyecto, son numerosos y por lo tanto, la cantidad de componentes gráficos que se requieren tienen que ser suficientes. Esta capa está dividida en tres grandes componentes que se muestran de acuerdo con el proceso de evaluación.

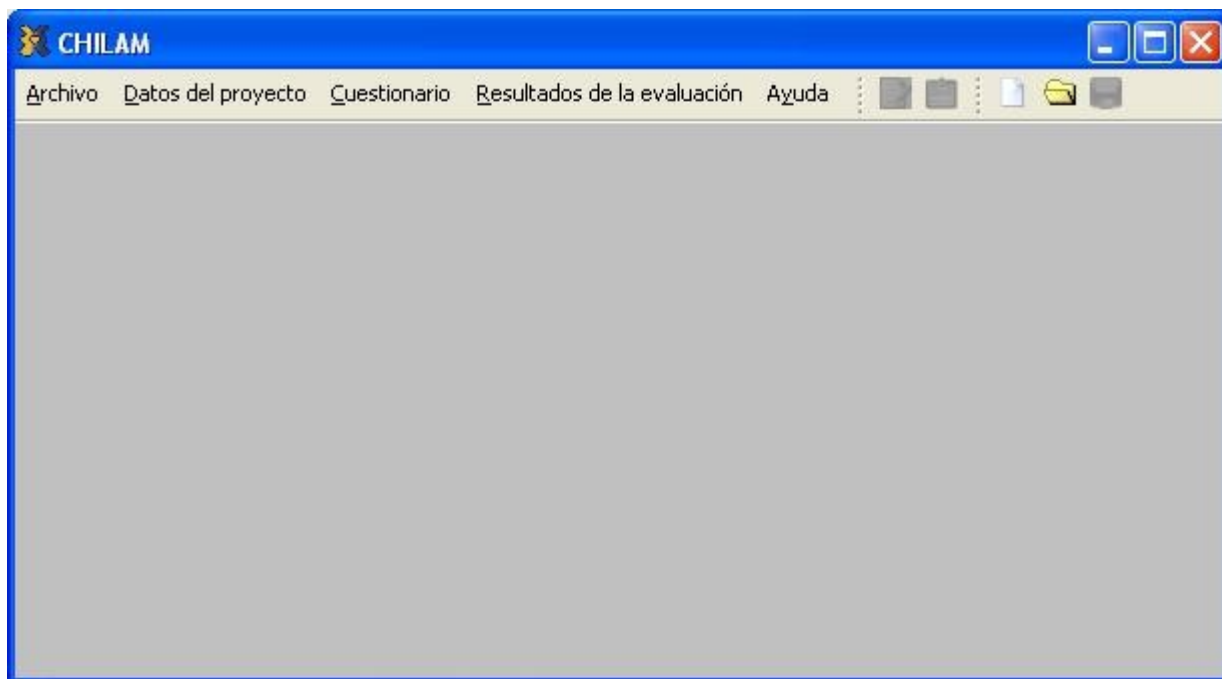


Tabla 5.8
Ventana principal de la interfaz gráfica.

1.- Configuración del proceso de evaluación

En esta ventana (Figura 5.9) se muestran los componentes necesarios para capturar la información, necesaria para poder comenzar con la evaluación de la empresa, es necesario que toda la información este completamente capturada para poder seguir adelante con la generación del "Acuerdo de evaluación", documento que sintetiza la información que ha sido recopilada y es firmado por los participantes del proceso de evaluación. El diálogo de configuración está dividido en 8 secciones que comprenden:

Ubicación y contraseña

Esta sección del documento captura el lugar del proyecto donde se generarán los archivos del proyecto, como son los archivos XML, los pdf's y se almacenarán los archivos necesarios al ocurrir un control de cambios, todos estos archivos quedan depositados en una carpeta con el nombre del proyecto que el usuario eligió.

Datos de la organización

Esta sección captura el nombre de la empresa evaluada, el período que dura la evaluación, la versión de MoProSoft que se utiliza, el tipo de propósito de la evaluación, las observaciones y restricciones de la evaluación.

Datos del equipo

Esta sección captura toda la información necesaria del representante de la empresa evaluada, el facilitador, el evaluador competente, el promotor y el gestor.

Estos son los roles más representativos durante el proceso de evaluación, ya que en el momento en que ocurre un cambio en los acuerdos para la evaluación es necesario llevar a cabo el proceso de control de cambios, para el cual se solicitan las contraseñas del representante, evaluador competente y promotor. De estos 3 roles, el rol del evaluador competente es el más importante, ya que la contraseña del evaluador competente es necesaria para la apertura de proyectos, ver los resultados de la evaluación, activar el cuestionario para la corroboración de la información recopilada y el proceso de control de cambios.

Documentos de la organización

Esta sección captura la ruta local de los 3 documentos que son proporcionados por la empresa evaluada.

Alcance de la evaluación

Esta sección captura los datos de los procesos que serán evaluados, la información de estos procesos es básica para poder generar el cuestionario de evaluación ya que con base al nivel objetivo del proceso por evaluar se genera dinámicamente la interfaz del cuestionario de evaluación.

Equipo de evaluación

Esta sección captura los datos de los evaluadores que participarán en el proceso; nombres que son utilizados en la interfaz del cuestionario de evaluación para poder seleccionar al entrevistador en la ventana que muestra el cuestionario de evaluación.

Participantes de la evaluación

Esta sección captura los datos de las personas que laboran en la empresa evaluada que participarán en la evaluación, estos nombres son utilizados en la interfaz del cuestionario de evaluación para poder seleccionar el entrevistado en la ventana que muestra el cuestionario de evaluación.

Relación con MoProSoft

Esta sección captura la relación de los procesos realizados por la empresa con los procesos definidos por MoProSoft.

Cada una de las personas que son listadas en diferentes secciones de la interfaz poseen una clave generada por la interfaz. En el caso del evaluador competente, se requiere el número de acreditación que proporciona el Organismo verificador.

La información contenida en las 8 secciones antes mencionadas es indispensable para que el "Acuerdo de evaluación" sea generado en formato *pdf*, ya que parte de la información de las 8 secciones es utilizada como texto de este documento y la generación de este documento, es requisito indispensable para que este diálogo concluya su objetivo en el sistema.

Toda la información antes mencionada es editable solo en esta ventana cuando se genera un nuevo proyecto, y para poder modificar el contenido de esta, es necesario llevar a cabo el proceso de control de cambios.

Una vez que se ha aceptado que la información contenida en esta venta es correcta, se avanza en el proceso de evaluación de la empresa, pero si se desea mostrar la información contenida, aparece la ventana pero la información que muestra no es editable, razón por la cual es muy importante que se asegure que la información es correcta.

Una de las características de este diálogo, es la posibilidad de leer una ayuda sintética acerca de la sección que se está mostrando y que aclara las restricciones que posee cada campo mostrado.

Método de selección de procesos para el cuestionario de evaluación

Cuando el usuario confirma que la información capturada en la ventana es correcta, se copian y cifran los documentos proporcionados por la empresa en la carpeta de trabajo, se leen los archivos predefinidos de los 9 procesos que especifica MoProSoft y una vez cargados, se secciona cada proceso dependiendo del nivel objetivo especificado.

Archivos de proceso predefinidos

Por cada uno de los 9 procesos definidos por MoProSoft, se cuenta con un archivo en formato XML que contiene la misma información con todas las calificaciones asentadas en "N". Estos archivos sirven como base para los cuestionarios de evaluación que se generan con el uso del sistema.

Estos 9 archivos están almacenados en la carpeta "templates" que se encuentra en el directorio donde Chilam es instalado.



Tabla 5.9
Diálogo que permite capturar la información inicial de la evaluación

2.- Cuestionario de evaluación

Esta sección es una de las más importantes del sistema, ya que es aquí donde se captura la información de los procesos que forman parte del cuestionario de evaluación, cuya calificación permitirá formar el perfil de la empresa evaluada.

La ventana cuenta con un diseño similar a la ventana de la sección anterior (Figura 5.9), del lado izquierdo se muestran los componentes que proporcionan la navegación a través de las secciones y del lado derecho se muestra la información que corresponde a la sección donde el usuario ingresa la información del cuestionario. En esta ventana se muestran del lado izquierdo cada uno de los procesos seleccionados para la evaluación, cada proceso es un árbol que se puede navegar para capturar la información que este seleccionada, de esta forma el usuario sabe en que sección del proceso se encuentra (Figura 5.10).

Cada sección del proceso muestra los componentes necesarios para su edición cuando es seleccionado en el árbol, de tal forma que cada uno de los componentes mostrados en el árbol es independiente a cualquier otro, y mas aún, el componente mostrado en la parte derecha de la interfaz puede ser incorporado en otras interfaces con facilidad.

Cada uno de estos paneles tiene la capacidad de gestionar su propio comportamiento. El comportamiento implementado está basado en los requerimientos especificados por EvalProSoft, de este modo, por ejemplo, una práctica base no puede recibir una calificación distinta a "N" si no cuenta con al menos 2 evidencias asociadas.

Como se mencionó con anterioridad el manejo de las evidencias es crucial para la calificación de prácticas base esperadas, ya que es aquí donde se puede hacer referencia de cualquier evidencia del cuestionario sin importar el componente en que se encuentre el usuario, razón por la cual, se muestra el botón que accede a la base de datos de las evidencias, y cuya ventana muestra todas las evidencias capturadas por el usuario ordenadas por el proceso, nivel en que se encuentran y por último de manera alfabética.

Uno de los requisitos para el manejo de evidencias, es que se habilitará la interfaz para que solo el evaluador competente pudiese corroborar la información recopilada. Por este motivo se proporciona el botón que permite al evaluador competente habilitar ó deshabilitar la capacidad de corroboración en interfaz, ya que una información corroborada no se puede editar posteriormente.

Para habilitar la corroboración de la información recopilada la interfaz muestra un diálogo que captura la contraseña del evaluador competente, en caso de que ésta sea correcta la interfaz se habilita y permite la corroboración. Por otro lado, si la interfaz está habilitada para corroborar, basta con presionar el mismo botón para deshabilitar la interfaz y esta no permita la corroboración de la información recopilada.

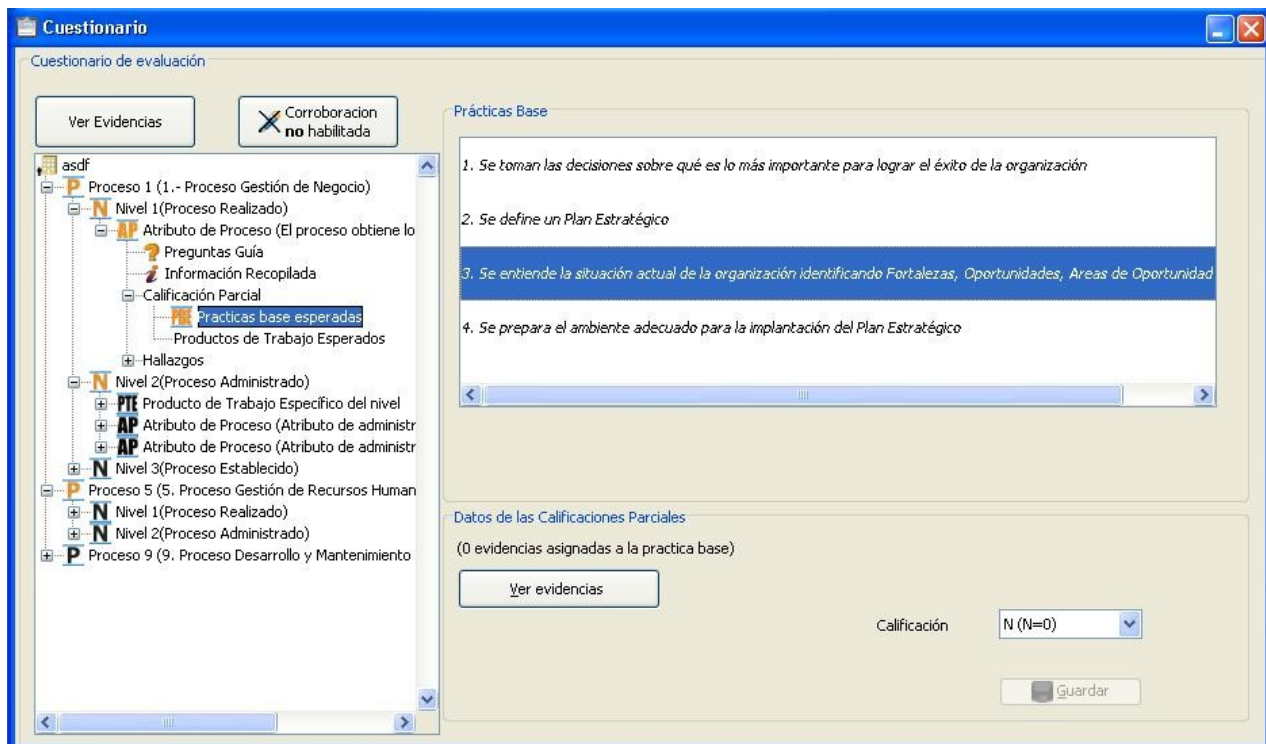


Tabla 5.10
Diálogo que permite capturar/editar el cuestionario de evaluación.

3.- Resultados de la evaluación

3.1 Resultados de la evaluación

Cuando el cuestionario de evaluación ha sido calificado, se pueden ver los resultados de evaluación (Figura 5.11). La pestaña "Sección de gráficas", permite mostrar la gráfica que relaciona el nivel objetivo y el nivel deseado de cada uno de los procesos o de todos los procesos del cuestionario de evaluación, Éste componente posee varias funcionalidades, como la capacidad de guardar la gráfica mostrada en un archivo de imagen.

La pestaña "Calificaciones", muestra las calificaciones obtenidas de manera puntual por cada una de las tareas del proceso, así como el tiempo aproximado invertido por sesión del proceso. Información que permite estimar con mayor precisión la duración de las evaluaciones que se aplican a las empresas.

La pestaña "Tareas no alcanzadas", muestra la lista de las tareas que obtuvieron una calificación distinta a "C", lista que permite a la empresa identificar con precisión los rubros por mejorar para la siguiente evaluación.

Cada una de las secciones antes mencionadas, se incluyen en los reportes generados para proporcionar mayor información a la empresa evaluada y al Organismo verificador.

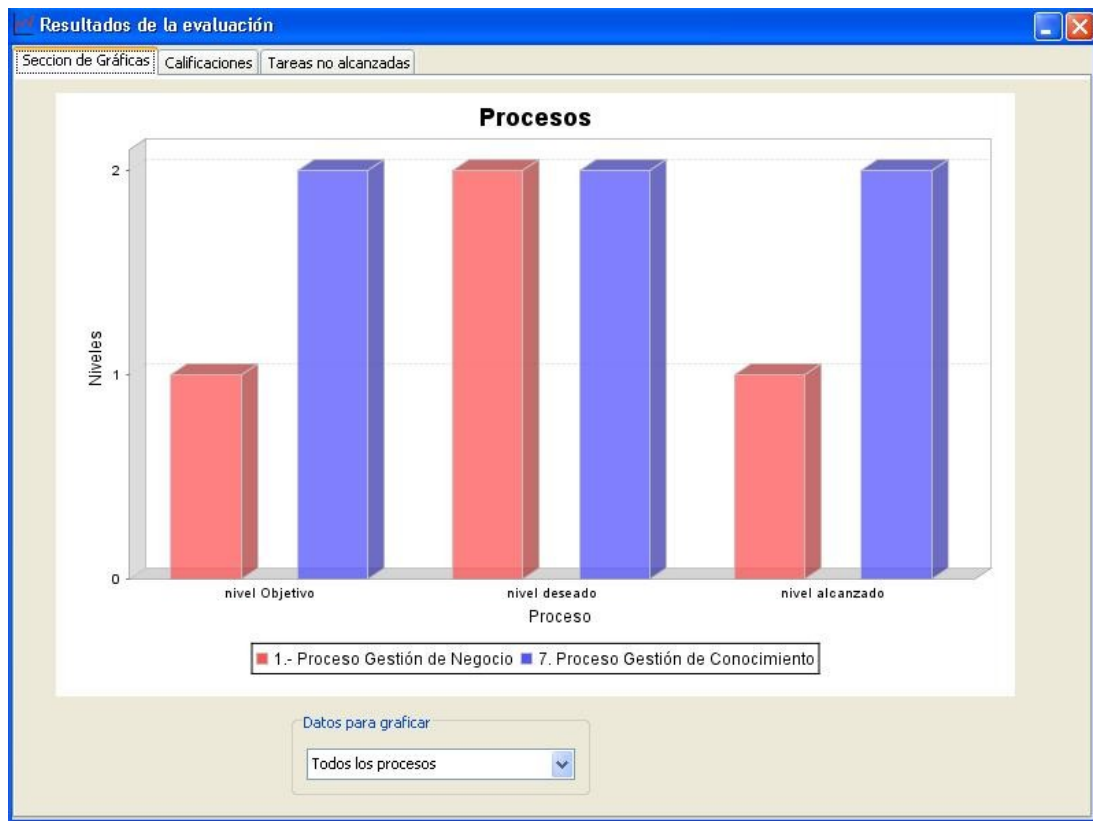


Tabla 5.11
Ventana que muestra los resultados de la evaluación.

3.2 Generación de paquetes

Para la generación de los reportes al Organismo verificador y la empresa evaluada, se requiere de una contraseña que restrinja la lectura del documento, con el fin de proporcionar mayor confidencialidad (Figura 5.12).

Con las contraseñas capturadas se generan los reportes dentro de la carpeta de trabajo del proyecto en formato pdf.

Tabla 5.12

Diálogo que permite capturar las contraseñas para generar los reportes.

Control de cambios

Esta sección de la interfaz se utiliza cuando es necesario modificar la información capturada durante la configuración del proyecto, razón por la cual se requiere la contraseña de la mesa de control de cambios, conformada por el evaluador competente, el promotor y el representante (Figura 5.13).

El procedimiento de control de cambios en el sistema, consiste en guardar el estado actual de la información en la interfaz, se crea un directorio llamado "versiones", se copian los archivos cifrados proporcionados por la empresa al directorio versiones, se copian los archivos XML al directorio versiones y se genera un solo archivo comprimido que contiene todos los archivos antes mencionados. Por último se muestra la ventana de configuración habilitada para editar la información, **es importante observar que no importa el avance que se tenía en la aplicación del cuestionario de evaluación, toda la información es perdida y se comienza todo el proceso desde el inicio.**

Tabla 5.13

Diálogo que permite capturar las contraseñas necesarias para llevar el proceso de control de cambios.

Estados de la interfaz

Los estados de la interfaz permiten conocer las acciones que ha realizado el usuario a través del sistema, lo que permite recordar en que sección del sistema se debe presentar la interfaz cuando el proyecto es abierto. Esta variable puede cambiar de estado según las acciones del usuario, los estados de la interfaz se determinaron a partir de la información proporcionada por EvalProSoft, de tal modo que se diseñaron los estados de la interfaz basados en el proceso que se sigue al aplicar una evaluación.

- 1.-El proyecto ha sido configurado, es decir, se ha ingresado toda la información de los datos de la empresa, se ha especificado el alcance de la evaluación, se ha ingresado la información acerca del equipo de evaluación etc.
- 2.- La evaluación se está llevando a cabo y el cuestionario no ha sido completado aún.
- 3.- El cuestionario ha sido terminado y se ha calificado.
- 4.- Se ha generado el paquete de evaluación del proyecto (el paquete que se queda en la empresa así como el paquete que va dirigido al Organismo Verificador).

6.- TECNOLOGÍAS UTILIZADAS EN EL DESARROLLO

1.- iText

iText es una biblioteca que permite generar archivos del pdf. Las clases del iText son muy útiles para la gente que necesita generar documentos que no se deseen alterar continuamente, esta biblioteca permite crear archivos que contienen texto, listas, las tablas, imágenes, etc.

Esta biblioteca está bajo la licencia GPL y posee una cantidad importante de documentación que ejemplifica su uso, la razón por la cual se incorporó esta biblioteca, es que permite asignar una contraseña de apertura para el documento generado que en comparación con otras bibliotecas, no es posible.

2.- SQLite

SQLite es una biblioteca pequeña escrita en C, que permite emular un sistema manejador de base de datos por medio de un archivo que no requiere ningún tipo de instalación.

Si se requiere utilizar este proyecto en conjunto con java es necesario anexar en el ambiente de ejecución una biblioteca dinámica, para que el menejador (driver) que es proporcionado por el proyecto pueda manejar las transacciones.

Esta biblioteca está bajo la licencia GPL y posee una cantidad de documentación mínima, la razón por la cual se incorporó al desarrollo fue para brindar el manejo independiente de las evidencias que es fundamental.

3.- XYLayout

Esta es una biblioteca desarrollada por Borland que permite acomodar los componentes gráficos de una manera mas sencilla, razón por la cual se incorporó al sistema.

4.- Secure File Handler

Secure File Handler es una biblioteca escrita en Java para cifrar o descifrar un archivo usando una contraseña.

Se incorporó al sistema para facilitar el cifrado de los documentos de la empresa. Este proyecto no posee documentación y está liberado bajo la licencia GNU.

5.- JFreeChart

JFreeChart es una biblioteca para graficar datos, posee varias modalidades ya que puede ser utilizado como graficador vía servlets, incorporado a Swing vía paneles o utilizado para generar las imágenes como un pojo (*plain old Java object*), este proyecto está liberado bajo la licencia LGPL de GNU.

La documentación que posee está referenciada por otras páginas que dan una información sencilla del proyecto, ya que su documentación es vendida.

6.- Inno Setup

Inno Setup es un generador de instaladores para programas que se ejecutarán sobre Windows, este desarrollo posee una licencia que se especifica en la página inicial del sitio y que permite utilizarlo para propósitos comerciales, este proyecto es incorporado para generar el instalador que configurará de manera automática Chilam en la computadora donde correrá. Este proyecto posee una documentación basta que permite incorporar diversas tareas.

7.- Launch4j

Launch4j es una herramienta para envolver las bibliotecas de java distribuidas como "jars" en ejecutables ligeros para Windows. El ejecutable se puede configurar para buscar cierta versión de JRE, y es posible fijar opciones para el ambiente de ejecución, como el tamaño inicial/máximo de memoria por utilizar.

Este proyecto es utilizado para envolver el "jar" generado para el proyecto y transformarlo en un ejecutable para windows, el proyecto cuenta con la información suficiente y está liberado bajo la licencia GPL.

8.- Doxygen

Doxygen es un sistema para generar la documentación de clases escritas en C++, C, Java, Python, IDL entre otros. Este proyecto es utilizado para extraer la documentación de las clases de java, ya que permite exportar la documentación en diversos formatos. Este proyecto está liberado bajo la licencia GPL.

7.- PRUEBAS DEL SISTEMA

Durante esta fase, el sistema se emplea de manera experimental para asegurarse que el software no tenga fallas, es decir, que funciona de acuerdo con las especificaciones y en la forma en que los usuarios esperan que lo haga. Se alimentan como entradas conjuntos de datos de prueba para su procesamiento y después se examinan los resultados. En ocasiones se permite que varios usuarios utilicen el sistema, para que los analistas observen si tratan de emplearlo en formas no previstas, antes de que la organización implante el sistema y dependa de él.

Cada una de las clases que conforman Chilam contiene un método llamando "main", método con el cual se hicieron pruebas unitarias para examinar el comportamiento en los casos correctos y los casos incorrectos, con el objetivo de encontrar la mayor cantidad de errores que pudiesen provenir de la clase. En el caso de las clases que integran la interfaz gráfica del sistema, se probó además de ello, diferentes maneras de conformar la interfaz y su comportamiento con el usuario final, con la finalidad de evitar que el usuario pueda accionar eventos que lleven a la interfaz a un estado erróneo ó que aumenten el tiempo de aprendizaje del uso de la interfaz.

Pruebas de integración

Las pruebas de integración tienen por objetivo verificar el funcionamiento conjunto de dos o mas clases que cumplen con el comportamiento requerido.

Para poder efectuar las pruebas de integración se optó por seguir un orden de integración siguiendo el proceso natural en que el usuario final usaría la interfaz, por lo que el orden escogido fue:

- 1) Integración de las clases del dominio en conjunto.
- 2) Integración de las clases pertenecientes a la interfaz gráfica, correspondientes a las secciones que modifican la información inicial y la sección del cuestionario de evaluación.
- 3) Integración de las clases que cargan y almacenan la información

Una vez hecha la integración se verificó el buen funcionamiento de estos módulos y se integraron las siguientes clases:

4. Integración de las clases que validan y califican el cuestionario de evaluación.
5. Integración de las clases que muestran los resultados de la evaluación.
6. Integración de las clases que generan los reportes de los resultados de la evaluación.
7. Integración de las clases que gestionan el proceso de control de cambios.

Pruebas del sistema

El objetivo de las pruebas de verificación, es buscar discrepancias entre los requerimientos y la ejecución del software. Para realizar las pruebas del sistema se generó un documento donde se registra la prueba realizada, el resultado de la prueba, la sugerencia del cambio a realizar, la fecha y comentarios de la ejecución del cambio.

Este documento sirve como registro de los casos en que la prueba arrojó un resultado no esperado o erróneo. Este documento permitió el trabajo paralelo entre la ejecución de las pruebas y las adecuaciones, por lo que se pudo afinar con mas detalle el funcionamiento de la interfaz.

Estas pruebas fueron enfocadas en 3 sentidos.

- 1) Tratar de buscar que la interfaz permita al usuario llegar a un estado incorrecto.
- 2) Verificar que la información almacenada en el archivo fuese diferente a la información guardada ó que la información se mostrara en texto claro.
- 3) Provocar un error fatal en el sistema.

Las pruebas funcionales se hicieron para estar seguros que el sistema funcione como se espera y fue diseñado. Estas pruebas se llevaron a cabo creando un grupo de casos de prueba con datos ficticios. Las pruebas incluyeron la validación de campos, reglas de negocio y la gestión de los datos fuera del sistema.

El documento de las pruebas del sistema lista los errores del sistema encontrados y no se anexa en este, pero puede ser consultado en los documentos que vienen con esta tesis.

Como parte de las pruebas realizadas a Chilam, se realizaron pruebas para medir la cantidad de memoria requerida y la velocidad en que el sistema respondía, dando como resultado un cambio en la programación para mejorar el rendimiento del sistema, esto se logró mediante la búsqueda de recomendaciones para disminuir las operaciones necesarias y realizar la misma tarea.

8.- CONCLUSIONES

Con la experiencia obtenida después de desarrollar este trabajo observe que el desarrollo de nuevas herramientas que apoyen la implantación de metodologías de software son variadas, pero actualmente no existen muchas herramientas que permitan apoyar la evaluación de los procesos para calificar el nivel de estas implantaciones, por lo cual, este desarrollo permite empezar a fomentar este tipo de herramientas.

Dada la escasez de herramientas orientadas para el apoyo de procesos de evaluación, el diseño de la interfaz es mas difícil, dado que no hay un marco de referencia donde se pueda obtener información acerca de buenas prácticas o referencia de estándares diseñados de este tipo de interfaces. Razón por la cual fue necesario apoyar la generación de la interfaz en el conocimiento del desarrollador, pero el creación de la interfaz va de la mano con la cantidad de información que el desarrollador posea acerca del proceso de evaluación.

En este sentido, las oportunidades de innovar son propicias, ya que es en este tipo de casos donde la cantidad de información que se posee, se ve utilizada al 100%. Pero mas allá de la falta de información o de las oportunidades, es muy importante seguir un proceso para el desarrollo del software que permita llevar un control de calidad y provea la información necesaria para que el sistema generado sea competitivo. La falta de ello se ve reflejada en una inversión de tiempo considerable, de hecho se invierte mucho mas tiempo del que se tenía considerado invertir, ya que la falta de diseño o la falta de especificación suficiente de los requerimientos, conlleva a solucionar continuamente nuevas problemáticas que no habían sido consideradas.

Una de las prácticas que permitió al sistema adaptarse a los nuevos requerimientos fue su diseño, ya que la modularidad de la implementación de las clases permitió la incorporación de los nuevos requerimientos, sin que este provocara cambios grandes o cambios en cascada, razón por la cual, incorporar un método de desarrollo de software es importante que procure invertir mayor esfuerzo en el diseño, ya que de este diseño, dependerá la modularidad, el escalamiento y rendimiento del sistema.

Un punto importante a resaltar en los desarrollos, es la incorporación de proyectos de software libre, ya que estos permiten añadir funcionalidades al sistema que no podrían ser incorporadas en un tiempo razonable, además de proveer un nuevo marco de referencia que puede ser utilizado para mejorar las prácticas de desarrollo o cambiar la filosofía de algunas clases.

Es importante destacar que la experiencia y el número de personas involucradas en el desarrollo de un sistema, siempre debe de guardar un equilibrio con la magnitud del proyecto a desarrollar. Ya que de lo contrario el tiempo de desarrollo se ve incrementado de manera sustancial por las cargas y responsabilidades, y dado que no existe un medio con el cual se pueda calcular el número de personas que requiere cada proyecto de manera exacta. Esta decisión depende de la experiencia del director del equipo de desarrollo.

Por último, lo mas importante de llevar a cabo una tesis es la experiencia y conocimiento que ésta brinda, por lo que el desarrollo de herramientas como ésta, terminan por reafirmar con gran relevancia muchas de las enseñanzas de los profesores que imparten clase en la facultad de Ciencias, por lo que personalmente esta tesis me permitió mejorar mis prácticas empleadas para el desarrollo de software y obtener más experiencia.

9.- REFERENCIAS

<http://www.lowagie.com/iText/>
<http://www.sqlite.org/>
<http://sourceforge.net/projects/sfilehandler>
<http://www.jfree.org/jfreechart/index.php>
<http://www.jrsoftware.org/isinfo.php>
<http://launch4j.sourceforge.net/>
<http://www.stack.nl/~dimitri/doxygen/>
<http://examples.oreilly.com/jswing2/code/>
<http://www.dcc.uchile.cl/~psalinas/uml/>
<http://www.clikear.com/manuales/uml/>
<http://www.java2s.com/Code/Java/CatalogJava.htm>
<http://javaalmanac.com/>
<http://forum.java.sun.com/index.jspa>
<http://www.w3.org/DOM/>
<http://tools.devchannel.org/>

Citas textuales

1. Diagrama de control de flujo tomado de la Guía para participantes de la evaluación que forma parte de la documentación de EvalProSoft, página 9.
2. Ibi dem página 6
3. Texto basado en las definiciones de las siguientes direcciones electrónicas:
<http://www.creangel.com/uml/casouso.php>
<http://www.dcc.uchile.cl/~psalinas/uml/casouso.html>