UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE INGENIERÍA INDUSTRIAL Y DE SISTEMAS

MODELO DE SEGURIDAD PARA PREVENIR EL USO INDEBIDO DE COMPONENTES DE LA CAPA DE REGLAS DE NEGOCIO EN SISTEMAS DESARROLLADOS CON TECNOLOGÍA MULTI-TIER

Informe de Suficiencia

Para optar el Título Profesional de:

Ingeniero de Sistemas

Carlos Omar Calero Cueva

Lima - Perú

2006

MODELO DE SEGURIDAD PARA PREVENIR EL USO INDEBIDO DE COMPONENTES DE LA CAPA DE REGLA DE NEGOCIOS EN SISTEMAS DESARROLLADOS CON TECNOLOGÍA MULTI-TIER

Aplicado al: Proyecto de Reingeniería de los Sistemas de Información Minera, actualmente en ejecución en la empresa Minera Aurífera Peruana S.A., dedicada a la explotación de oro.

DEDICATORIA:

A mis padres, Violeta y Carlos, por ser ejemplo de esfuerzo y dedicación, fuente inagotable de valores, y mi mayor apoyo en los momentos más difíciles. A mi tía Elsa, por su ilimitado amor de madre y por su ayuda permanente e incondicional.

ÍNDICE

DESCRIPTORES TEMÁTICOS

Modelo de Seguridad basado en claves

Seguridad Informática

Desarrollo multicapas

Estándares de desarrollo

Tecnología de Componentes

Niveles de desarrollo

Seguridad del código fuente

Gestión de versiones de código

Actualización automática de aplicaciones

Reingeniería de Sistemas de Información Minera

RESUMEN EJECUTIVO

Además de la significativa reducción de costos de mantenimiento, y los beneficios de centralizar la seguridad de nuestros procesos, el esquema de desarrollo de sistemas basado en capas también nos permite gestionar la escalabilidad de los mismos. Esto es, pasar a plataformas tecnológicas superiores con menores costos de desarrollo.

No obstante los beneficios de las tecnologías emergentes, existen siempre riesgos subyacentes. Los esquemas de seguridad convencionales protegen la información corporativa de la intrusión de entes externos; sin embargo, con frecuencia, omiten considerar la posibilidad de que el "intruso" ya esté dentro de la organización. El objetivo del presente trabajo es mostrar una nueva alternativa para cubrir este tipo de vacíos de seguridad y preparar a nuestros sistemas de información para soportar ataques de intrusos internos y externos a la organización.

El modelo de seguridad expuesto está basado en claves y fue desarrollado en el marco del Proyecto de Reingeniería de los sistemas de información minera de la empresa.

INTRODUCCIÓN

El aumento de la información y la necesidad de procesar, reprocesar y explotar mayores volúmenes de datos trae consigo, para los usuarios de sistemas de información, una mayor necesidad de adquirir conocimientos sobre el manejo de herramientas que les permitan automatizar, por si mismos, algunos procesos manuales propios de su que hacer laboral. Si bien tener usuarios con conocimientos de programación, enriquece y facilita en gran medida la comunicación con los analistas de sistemas, también propicia una situación riesgosa para nuestros datos.

Como sucede en Minera Aurífera Peruana, la necesidad de contar con *personal administrativo* que sepa hacer "*de todo un poco*" es una realidad en algunas de empresas del sector minero. A la luz de los costos de alimentación, hospedaje, medicación y trasporte a la unidad minera, las empresas mineras consideran más económico trabajar y rotar, en algunas áreas, pequeños grupos de trabajadores que conozcan de todo, en lugar de invertir en procesos de reclutamiento y capacitación especializada.

En Minera Aurífera Peruana, dada las limitaciones de disponibilidad de personal, los requerimientos de procesamiento de datos y el mantenimiento de sistemas fueron desatendidos. Como resultado, el personal administrativo que tenía la responsabilidad de elaborar informes mensuales comentados, gráficos estadísticos y reportes, fue autocapacitándose en programación para desarrollar sus propias mini-aplicaciones que les permitiría automatizar, en alguna medida, su trabajo. Esta situación, aunada a las deficiencias del antiguo esquema de seguridad de los datos, puso en peligro la seguridad de la información corporativa.

A mediados del año 2000, durante un primer proceso de auditoría, el área de sistemas en la unidad minera constató que la modificación de datos no autorizada. Algunos usuarios de los antiguos sistemas DOS, habían modificado irregular y sistemáticamente los datos, sin hacer uso de los sistemas informáticos y sin dejar rastro, causando su pérdida e inconsistencia.

Dentro del proceso de ejecución del Proyecto de Reingeniería de los Sistemas de Información Minera, se tomó muy en cuenta éste tipo de problemas, porque aunque la plataforma y la tecnología a utilizar serían completamente diferentes, los modos de explotar las deficiencias de seguridad aun subyacían.

El esquema desarrollado en el presente trabajo, logró cubrir los vacíos de seguridad informática que existieron por varios años en la empresa, eliminando completamente intrusiones, y accesos irregulares a la información corporativa. Podemos decir ahora, que nuestros sistemas de información proveen información oportuna y confiable para la toma de decisiones.

OBJETIVOS GENERALES

Mostrar los beneficios logrados para la empresa gracias a la aplicación creativa de los conocimientos adquiridos en la universidad y en el mercado laboral.

- Evidenciar con claridad que el origen de los problemas de seguridad de la información es el resultado de la convergencia de problemas de procedimiento, comunicación, gestión de recursos, eficiencia y, por último, problemas técnicos.

OBJETIVOS ESPECÍFICOS

Describir el desarrollo de un modelo de seguridad que, a un costo razonable, permita satisfacer los niveles de seguridad de datos exigidos por las empresas.

Evidenciar que los ataques de intrusión en los datos, no solo tienen su origen en el exterior de las organizaciones, sino que estos también se producen desde el interior.

CAPÍTULO I

ANTECEDENTES

1.1 RESEÑA

La historia de esta empresa es un ejemplo típico del esfuerzo decidido de un empresario peruano que alcanzó el éxito gracias a su constancia y optimismo. El mérito empresarial es mayor aún, si consideramos que la empresa se consolidó durante la década del 80, aquella en la que gran numero de minas tuvieron que ser paralizadas al haber dejado de ser rentables como consecuencia de la crisis económica que entonces soporto la minería nacional y el país en general.

1.2 DIAGNOSTICO ESTRATÉGICO

Minera Aurífera Peruana

Objetivo

"Contribuir a la sostenibilidad económica del Perú a través de la explotación efectiva de sus recursos auríferos, expandiendo la producción

minera en términos de eficiencia, para lograr el desarrollo de una minería peruana altamente competitiva".

Misión

"Alcanzar un nivel de minería eficiente, que impulse económicamente al país; cumpliendo siempre con las normas de seguridad y protección del medio ambiente; reduciendo el impacto e impulsando el desarrollo económico y cultural de las comunidades cercanas a los centros de producción".

Visión

"Ser una empresa consolidada nacional e internacionalmente en la explotación y exportación de oro, convirtiéndose en el principal contribuyente del crecimiento económico del país".

1.2.1 ANÁLISIS INTERNO

Fortalezas

- Rentabilidad alta y sostenida.
- Estados Financieros favorables.
- Alto nivel de productividad por unidad de capital.
- Nivel estratégico calificado, con amplia experiencia en el rubro.
- Posee un buen respaldo de reservas mineras.
- Tiene estabilidad económica sólida.

Debilidades

- Poca disposición por inversiones en TI.
- Organización vertical.
- No se realizan evaluaciones de proyectos de inversión, o su evaluación es superficial.
- Gestión de recursos humanos incipiente.
- Alta rotación de personal operativo. No existe gestión del conocimiento.

1.2.2 ANÁLISIS EXTERNO

Oportunidades

- Tendencia creciente del precio internacional del oro.
- Resultados alentadores en proyectos de exploración. Hallazgo de nuevas vetas.
- Reducción de precios de equipos tecnológicos para mejorar la comunicación con la unidad operativa.

Amenazas

- Entorno político-económico incierto.
- Impuesto a las sobreutilidades. Reducción de la rentabilidad.
- Incremento del gasto por pago a las contratas.

- Intervención del estado. Incremento del gasto por pago de planillas.
- Fuga de información confidencial de la empresa.

Análisis FO (Fortalezas-Oportunidades)

- Aprovechar la rentabilidad de la compañía para invertir en proyectos tecnológicos que nos permitan lograr ventajas competitivas. Es necesario mejorar los niveles de comunicación y coordinación entre la unidad minera y la sede central en Lima, reduciendo costos de tiempo y desplazamiento físico.

- Implementar un segmento de red inalámbrica para comunicar el centro de cómputo de sistemas-mina con las oficinas en los niveles de extracción más bajos.

- Incrementar la extracción de mineral en las zonas donde las reservas se estimen auspiciosas con el fin de exportar y aprovechar así el alto precio internacional del oro.

Análisis DO (Debilidades-Oportunidades)

- La falta de capacitación continua del personal podría hacer lento el proceso de aprendizaje en el uso de nuevos equipos TI, y por tanto frenar el incremento esperado de productividad. Debe realizarse un proceso de capacitación previo a la implementación de nuevos equipos TI.

- Gestionar el conocimiento para controlar la diversidad de costos en los que incurrimos como resultado de la continua rotación de personal.

Análisis FA (Fortaleza-Amenazas)

- Invertir en programas de capacitación de personal para reducir el volumen de la planilla conforme aumenta la productividad.

- Dado el protagonismo del sector minero en la economía nacional, negociar y establecer acuerdos con el estado en temas tributarios.

Análisis DA (Debilidades-Amenazas)

- Realizar reformas en la estructura organizacional de la compañía con el fin de agilizar el proceso de toma de decisiones. Las decisiones se encuentran demasiado centralizadas y esto le resta competitividad a la compañía.

- Es necesario tomar conciencia de la importancia de evaluar rigurosamente los proyectos de inversión para establecer su rentabilidad y el mejor modo de financiamiento. La ejecución no planificada de los proyectos representa sobrecostos que la empresa soporta con su rentabilidad. Esta situación no es sostenible en el tiempo.

1.2.3 MATRIZ FODA

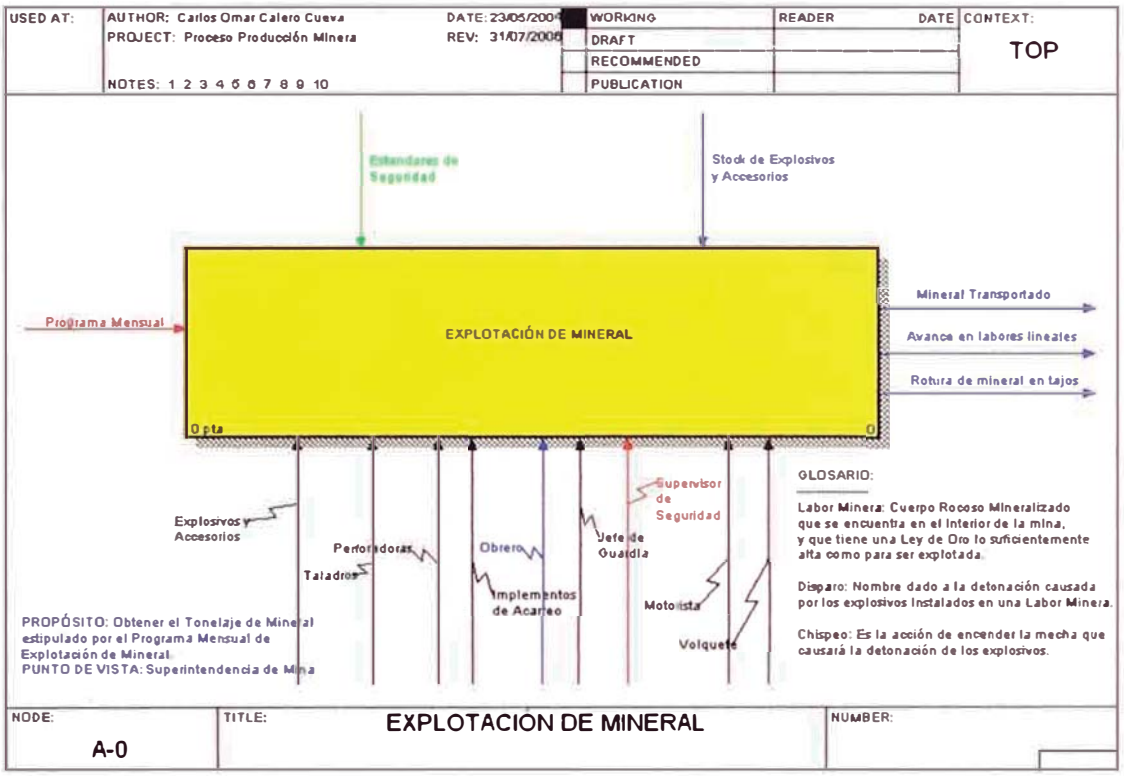
		Factores Internos	
		Oportunidades	Amenazas
Factores Externos	Fortalezas	***Potencialidades*** (FO) - Aprovechar el precio alto del oro incrementando la extracción en las vetas con mejor ley. Incrementar las inversiones en tecnologías de información y comunicaciones. -Incremento de productividad y reducción de costos gestionando el conocimiento y utilizando TIC.	***Riesgos*** (FA) - Invertir en programas de capacitación de personal para reducir el volumen de la planilla conforme aumenta la productividad. - Dado el protagonismo del sector minero en la economía nacional, negociar y establecer acuerdos con el estado en temas tributarios.
	Debilidades	***Desafios*** (DO) - Concientizar al nivel gerencial de la empresa acerca de la importancia de mantener capacitado al personal.	***Limitaciones*** (DA) - No se cuenta con los suficientes mecanismos de seguridad para proteger la confidencialidad e integridad de la información.

1.3 DIAGNÓSTICO FUNCIONAL

Proceso Principal: Explotación de Mineral

La Explotación de Mineral es el proceso principal de la empresa Minera Aurífera Peruana S.A. Este proceso, tiene un *Input* fundamental, conocido como: Programa Mensual de explotación de Mineral, el cual es generado por el área de Planeamiento, con participación de las áreas de Geología y Mina. El programa mensual se elabora utilizando la información de reservas almacenada por los Sistemas de Información Minera.

Proceso: Explotación de Mineral



La información contenida en el Programa Mensual consta de los siguientes componentes:

a. Listado de Labores: Relación de Labores Lineales y Tajos, en los cuales se realizarán actividades de:
   - Exploración
   - Preparación
   - Desarrollo
   - Explotación (solo sobre Tajos)

Los datos de la Labor incluyen: Su ubicación, Unidad Económica, División, Sección, Zona, Nivel, Block y Veta; sus características: Tipo de

Roca y Sección (área transversal). También se especifica el Método de Explotación a aplicar (Tipo de Labor), el Tipo de Equipo de Limpieza a utilizar en la labor, la o las Empresas Especializadas a las que se asigna la Labor, etc.

b. Ley promedio de oro: Resultado del proceso de *muestreo* realizado por el Área de Geología, y de la determinación de la Ley de Oro realizado por el Laboratorio.

c. Potencia de Veta: Resultado del proceso de Medición Mensual realizada por el Área de Ingeniería, dentro del subproceso de Medición Mensual de Labores.

d. Tonelaje de Mineral a extraer: Resultado del proceso de decisión realizada por Geología, Mina y Planeamiento, tomando en cuenta el resultado del subproceso de Cubicación de Reservas de Mineral realizado por el Área de Geología y que se encuentra registrado en los sistemas de información minera.

e. Observaciones: Son datos a tomar en cuenta con respecto a la labor. Generalmente se anotan necesidades identificadas para la labor, tales como tareas de sostenimiento, limpieza, etc.

Los *outputs* o salidas del proceso de Explotación de Mineral son:

1. ***Volumen de mineral transportado***: Es el volumen de mineral que se calcula con el peso específico y el peso registrado en la balanza electrónica al paso de los volquetes que lo transportan. El volumen de mineral está clasificado por tipo y se expresa en metros cúbicos.

2. ***Avance en labores lineales***: Es la distancia de avance lograda sobre labores lineales (cruceros, sub-niveles, chimeneas, etc.). Su medición se realiza a fin de mes y su valor se expresa en metros.

3. ***Rotura de Mineral en Tajos***: Es el volumen de mineral roto *In Situ*, producto de los ciclos de perforación y voladura. Se clasifica por tipo de mineral y se expresa en metros cúbicos. Su valor es siempre distinto al Volumen de mineral transportado, puesto que no todo lo que se rompe se transporta directamente a la planta, sino que se clasifica por tipo de mineral y normalmente quedan volúmenes de mineral en tránsito al interior de los socavones, generándose diferencias entre ambas lecturas.

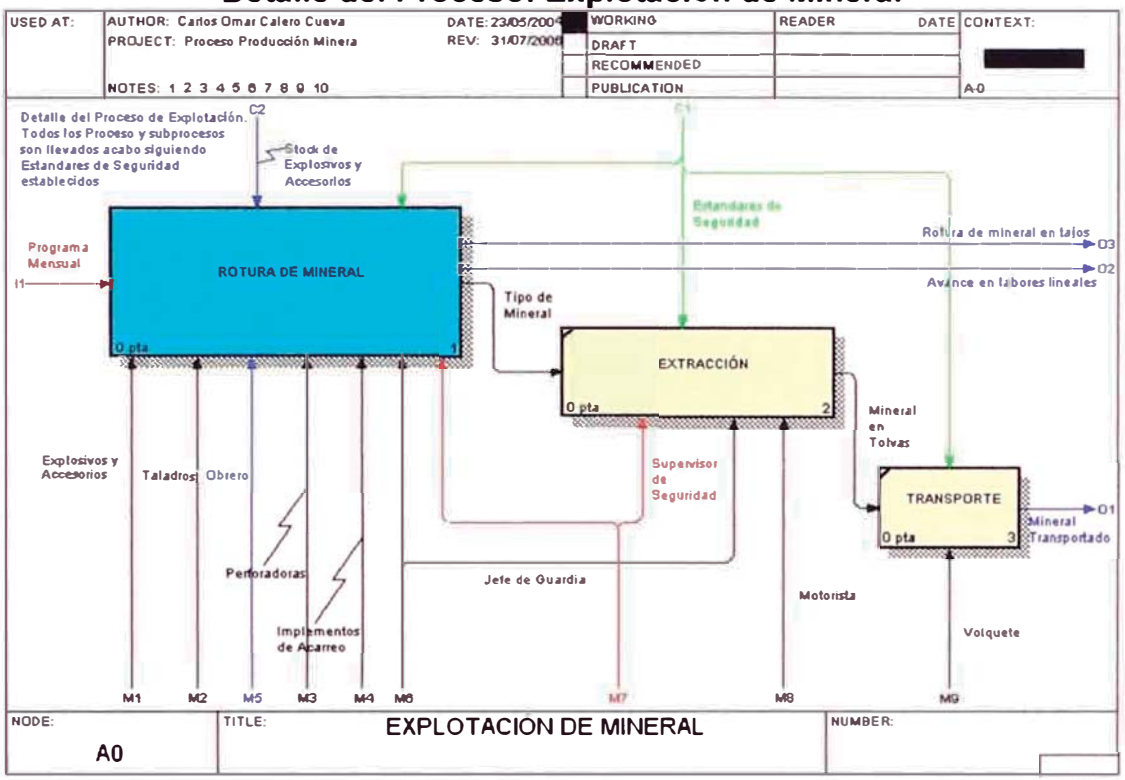
Dentro del Proceso de Explotación de Mineral, se tienen los siguientes subprocesos:

1. **Rotura de Mineral**: Se realiza mediante ciclos de perforación y voladura, perforando la roca para ubicar los explosivos que luego serán detonados para desprender el mineral de las paredes de la labor. La rotura de mineral se mide por avance (en metros) cuando se trata de Labores Lineales, y se mide por Tonelaje (en TM) cuando se trata de Tajos.

2. **Extracción de Mineral**: Consiste en extraer el mineral desprendido desde las labores mineras hacia las tolvas de mineral ubicadas en superficie.

3. **Transporte de Mineral**: Consiste en la carga, transporte y descarga de mineral, haciendo uso de volquetes. El destino final del mineral transportado es la Tolva de Gruesos ubicada en la Planta de Beneficio; punto de inicio del proceso de Beneficio o separación del oro.

Detalle del Proceso: Explotación de Mineral



Sub-Proceso: Rotura de Mineral

Consiste en desprender el mineral de las paredes de las labores haciendo uso de explosivos. La rotura de mineral se mide por avance (en metros) cuando se trata de Labores Lineales, y se mide por Tonelaje (en TM: Toneladas Métricas) cuando se trata de Tajos. El proceso de rotura de mineral para la Minería Subterránea es de alto riesgo, por lo que está permanentemente controlado y supervisado por el Programa de Seguridad Minera, a fin de que se cumpla con los estándares de seguridad exigidos por el Ministerio de Energía y Minas.

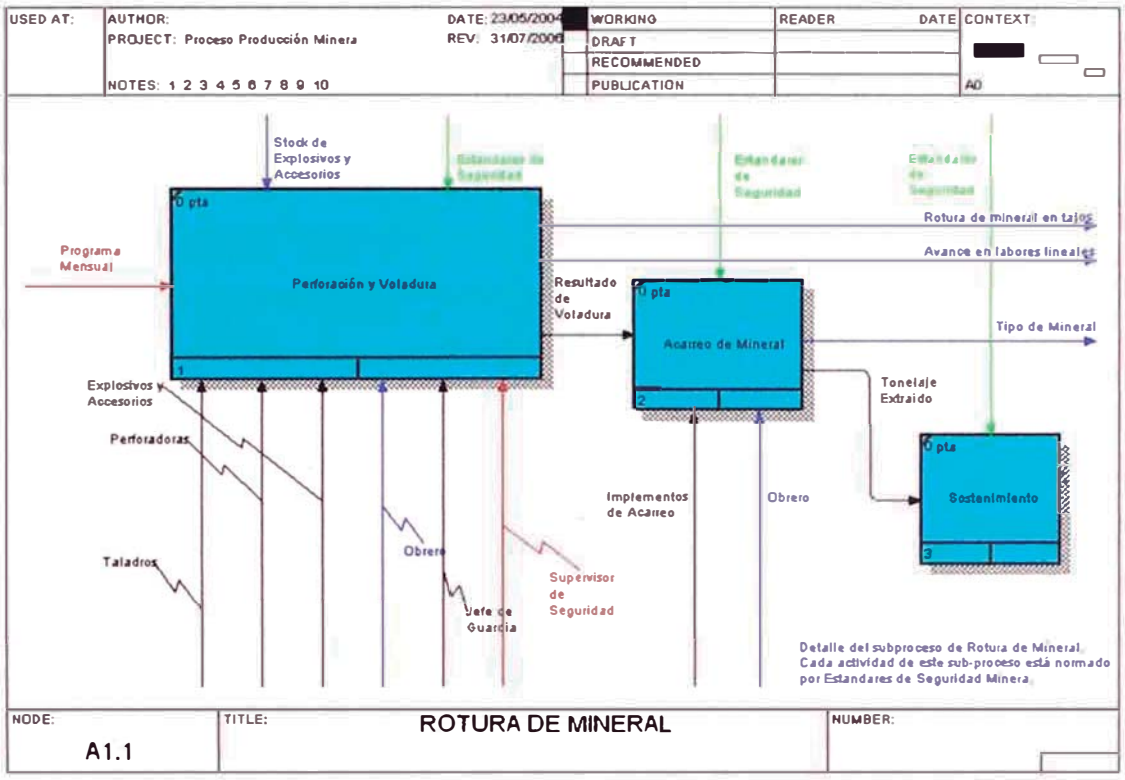
El Subproceso de Rotura de Mineral consta de las siguientes actividades:

a. Perforación y Voladura: Consiste en utilizar perforadoras y taladros para perforar orificios en la roca (paredes de las labores), los que serán, posteriormente, llenados con explosivos y detonados en forma sincronizada a fin de minimizar el impacto sobre la estructura de túneles ubicados en el interior de la mina. La acción de detonar los explosivos es comúnmente conocido como *"disparo"*.

b. Acarreo de Mineral: Es el recojo y transporte interno del mineral desprendido, producto del "*disparo*". El acarreo de mineral es un trabajo manual que los obreros realizan haciendo uso de palas y winches; y requieren de conocimiento y experiencia para determinar y clasificar el tipo de mineral que se ha obtenido (mineral de 1ra., de 2da, de 3ra. ó desmonte).

c. Sostenimiento: Son actividades realizadas para evitar que las labores colapsen. Todo disparo debe ser seguido por trabajos de sostenimiento para evitar derrumbes y/o desprendimiento de rocas. Estadísticamente, el desprendimiento de rocas es una da las causas principales de los accidentes de trabajo en el campo de la minería subterránea.

Detalle del Sub-Proceso: Rotura de Mineral



El *output* del Proceso de Explotación de Mineral: Tonelaje y Tipo de Mineral extraído, es uno de los *input* más importantes para el Proceso de Valorización, encargado de calcular el pago a las Empresas Especializadas asignadas a la explotación de mineral en dichas labores.

Proceso: Medición Mensual de Labores Mineras

Es el proceso que permite, a través de técnicas de medición y topografía, determinar los resultados reales que ha obtenido la operación minera. Este proceso es realizado por cuadrillas de topógrafos e ingenieros que se desplazan al interior de los socavones para medir los resultados.

Las mediciones se realizan sobre cada una de las labores trabajadas durante el mes. Todos los meses cada labor es asignada a una o más empresas especializadas. El objetivo es lograr que el trabajo realizado por estas empresas sobre sus labores asignadas, dé como resultado el cumplimiento del Programa Mensual de Explotación de Mineral establecido por el área de Planeamiento. La medición mensual permitirá, por tanto, verificar el nivel de cumplimiento de las metas. Los resultados que la medición deberá proporcionar son principalmente:

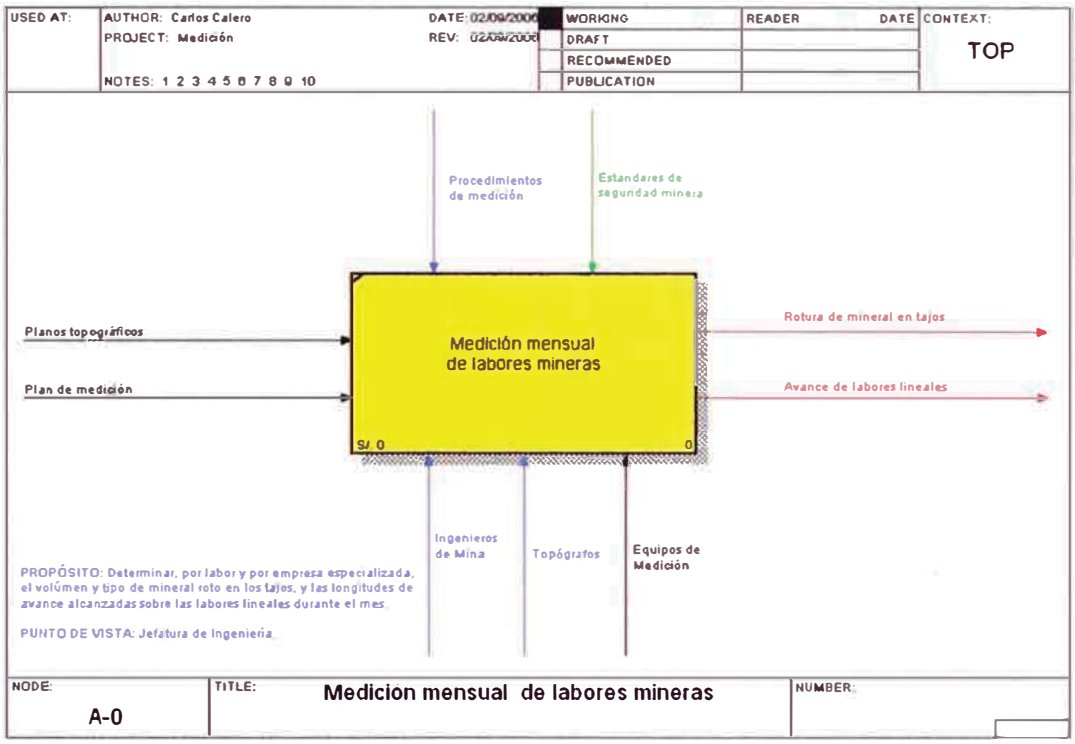
El tonelaje de mineral roto: Medición realizada sobre labores mineras denominadas TAJOS. Este tipo de labores son las que contienen en su interior concentraciones de oro, por lo que se requiere romper su estructura mediante secuencias de perforación y voladura. El tonelaje de mineral roto y la ley de oro del tajo permitirán determinar, posteriormente, la cantidad de oro fino que se obtendrá.

La longitud del avance: Medición realizada sobre labores mineras denominadas LINEALES. Este tipo de labores no proporcionan oro, sino que sirven fundamentalmente para establecer las vías de acceso y desplazamiento alrededor de los TAJOS.

Debido a que la explotación de mineral (perforación y voladura) es un proceso tercerizado mediante contratos con empresas especializadas, los

resultados de la medición mensual sirven para determinar el pago que deben recibir estas empresas.

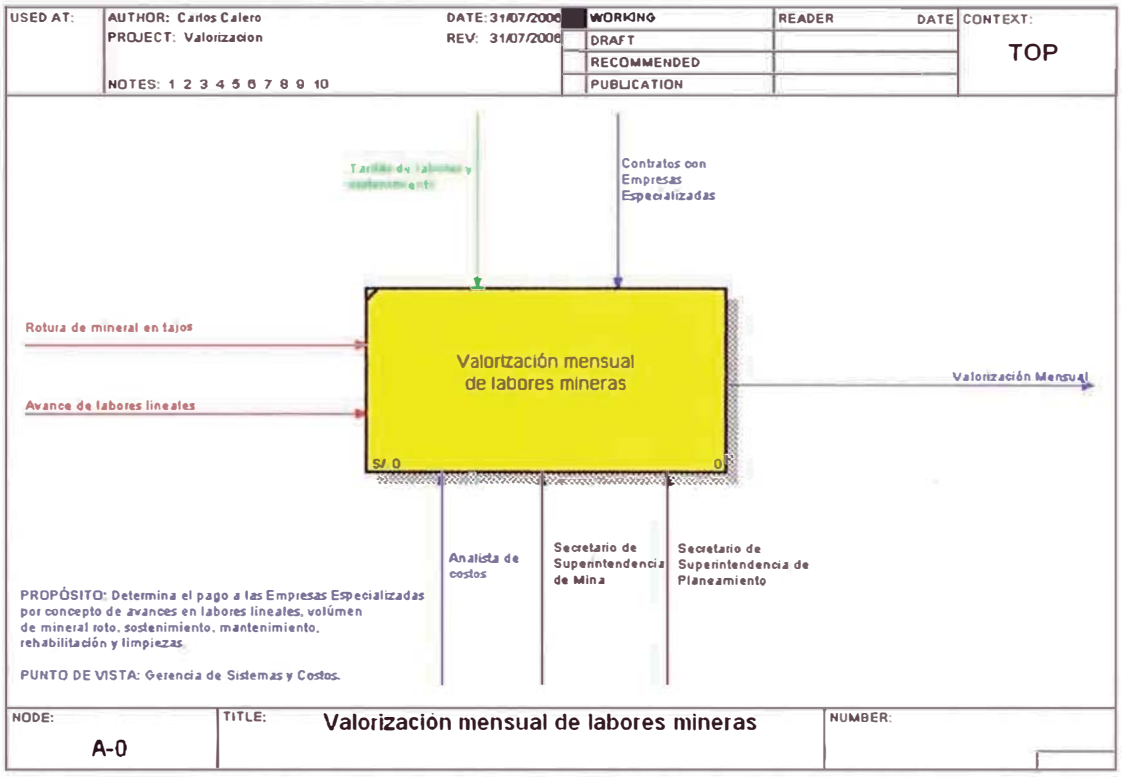
Proceso: Medición Mensual de Labores Mineras



Proceso: Valorización Mensual

Es uno de los procesos más importantes de la empresa. Colecta información de todos los sub-procesos productivos, pero se basa fundamentalmente en los datos proporcionados por el proceso de Medición Mensual de Labores Mineras.

Proceso: Valorización Mensual de Labores Mineras



Su resultado determina el pago a las empresas especializadas por distintos conceptos, entre los que tenemos:

1. Valorización de Labores Mineras: Es la valorización principal de la empresa. Mensualmente representa, en promedio, un millón y medio de dólares por pago a las empresas especializadas. La valorización se elabora en base a tarifas establecidas por contrato y según el nivel de avance o volumen de mineral roto en las labores.

2. Cálculo de incentivos: Son los incentivos, o bonos que se paga a las empresas especializadas por superar la meta de trabajo que mensualmente se establece.

3. Valorización de Enmaderado y Sostenimiento: Es el pago que se le hace a las empresas especializadas según las tarifas contratadas y de acuerdo a las tareas de enmaderado y sostenimiento que se hayan realizado durante el mes.

4. Valorización de Servicios Adicionales y de domingos y feriados: Son los pagos por concepto de días no laborables que se trabajaron durante el mes, así como por la realización de servicios tipificados como adicionales.

5. Amortización de Perforadoras: Son los pagos que se realizan a las contratas de acuerdo a la modalidad a la que se adquieren las perforadoras. La mayor parte de las perforadoras están en la modalidad de *leasing*.

6. Penalidades, Retenciones, Reintegro y otros: Las penalidades son descuentos que se realizan sobre el pago de las empresas especializadas por exceder los parámetros de consumo de explosivos. Las retenciones son reajustes; vale decir, montos que se dejan de pagar a las empresas especializadas debido a pagos en

exceso realizados en algún mes previo. De manera inversa, los reintegros son montos que se pagan para reajustar la omisión de algún pago en un mes previo.

El proceso de Valorización Mensual había sido, hasta entonces, un proceso semi-automatizado. El informe mensual de valorización se elaboraba tomando como base los datos de los antiguos sistemas de información de la empresa; datos cuya integridad estuvo completamente expuesta debido a los graves vacíos de seguridad que existieron en la plataforma tecnológica original.

El esquema de seguridad desarrollado en el presente trabajo, sirvió para evitar posibles fraudes en procesos como este. De hecho, a la Gerencia de Sistemas y a la Alta Dirección le quedaba claro, en este sentido, que la inversión en el nuevo esquema de seguridad estaba totalmente justificada, tomando en cuenta que este proceso representaba un costo promedio mensual de ocho millones de soles, y que cualquier cambio sistemático en los datos de base estaría significando grandes pérdidas económicas para la organización.

CAPÍTULO II

MARCO TEÓRICO

2.1 LA SEGURIDAD EN LA NUEVA ERA DE LA INFORMACIÓN

A medida que la Internet crecía en importancia, las aplicaciones informáticas se volvieron cada vez más interconectadas. Habían pasado aquellas épocas en que las computadoras eran, usualmente, islas de funcionalidad, con pequeña o ninguna conectividad; donde no importaba mucho la seguridad de las aplicaciones. Es así como, mientras las computadoras ejecutaban correctamente su aplicación informática, la mayoría de las personas olvidaron considerar el tema de seguridad de la información.

Ahora la situación es distinta. Con la Internet, están interconectados virtualmente todas las computadoras-servidores, computadoras personales, teléfonos celulares, dispositivos *de bolsillo*, entre otros. Aunque esto ofrezca

increíbles oportunidades de negocio y enormes mercados para los desarrolladores de software, también significa que los componentes interconectados pueden ser atacados.

Las aplicaciones que no fueron diseñadas para ejecutarse en entornos altamente interconectados frecuentemente originan sistemas susceptibles a ataques, pues sus creadores no las prepararon para funcionar en entornos expuestos a riesgos de intrusión. Un caso excepcionalmente riesgoso, desde el punto de vista de la seguridad de las aplicaciones, es la Internet. La World Wide Web es, a menudo, llamada *Wild Wild Web* debido a la hostilidad de los ataques que abundan en su entorno. Es fundamental entonces, que entornos interconectados, nuestras aplicaciones incluyan mecanismos que les permitan afrontar ataques informáticos.

La seguridad de la información no debe tomarse a la ligera. Decir que un entorno interconectado como la Internet es hostil, no es una exageración. Un demostración de la magnitud de la hostilidad que existe en la Internet se produjo el viernes 13 de Julio de 2001, cuando el Web Site operado por SANS Institute fue mutilada, reemplazándose su página principal con otra colocada por un atacante de Internet. La siguiente semana, SANS envió un e-mail a todos los subscriptores de su publicación SANS NewsBytes, con el siguiente comentario:

> "Este ha sido un asombroso recordatorio de cuan devastador puede ser un ataque de Internet. Todos los programas simples y configuraciones han sido revisados y, en muchos casos, rediseñados para que puedan operar en forma segura, no solo frente a los ataques de hoy en día, sino también de cara a los del nivel de amenaza que experimentaremos en un par de años. Algunos servicios pueden no estar disponibles por días."

Importancia de construir aplicaciones seguras

Debido al rápido desarrollo de la tecnología, nuestras aplicaciones deben estar preparadas para ejecutarse incluso en entornos o plataformas que aún no conocemos. Por tanto, cuando diseñamos la arquitectura del software, debemos asumir que este se ejecutará en el más hostil de los entornos.

Es importante recordar además que un sistema seguro es un sistema de calidad. El código que considera, desde un principio, mecanismos de seguridad en su diseño y construcción es más robusto que el código que considera la seguridad de manera tardía. Por ello, las aplicaciones seguras son más inmunes a las críticas, más atractivas para los usuarios, y menos caros de reparar y mejorar. Puesto que no se puede tener calidad sin seguridad, es fundamental hacer que el equipo de desarrollo sea conciente de la importancia de pensar siempre en los aspectos de seguridad del código.

Las Vulnerabilidades de Seguridad son costosas de corregir

Como todo cambio de ingeniería, las correcciones de seguridad son costosas cuando se incluyen tarde en el proceso de desarrollo. Es difícil determinar el costo de la corrección porque hay muchos intangibles involucrados, sin embargo el precio de hacerlo incluye los siguientes aspectos:

- El costo de la coordinación de la corrección. Alguien tiene que crear un plan para realizar completamente la corrección.
- El costo de los desarrolladores para encontrar el código vulnerable.
- El costo de los desarrolladores para corregir el código.
- El costo de los "testers" (examinadores) que deben probar la corrección.
- El costo de probar la configuración de la corrección.
- El costo de enviar la corrección al Web site o de instalarla en un Servidor de Componentes.
- El costo de escribir y mantener documentación.
- El costo de manejar malas relaciones públicas.
- El costo de pérdida de productividad de los desarrolladores. Es muy probable que todos aquellos desarrolladores involucrados en el proceso de corrección deberían estar, más bien, trabajando en la creación de nuevo código. Trabajar en la corrección de código, es tiempo perdido.

- El costo de pérdida de productividad de los usuarios. Ellos deberán ejecutar la aplicación corregida en un servidor de pruebas, para verificar que trabaja como fue planeado. Una vez más, el personal consume su tiempo en tareas menos productivas.
- Finalmente, el potencial costo por ingresos perdidos, generalmente, a causa de usuarios que deciden posponer o evitar el uso de nuestros sistemas por considerarlos inseguros o poco confiables.

Como podemos ver, el costo potencial de hacer una corrección de seguridad es claramente alto. Si consideramos los aspectos de seguridad desde el principio, mientras diseñamos y construimos el producto, podríamos estar ahorrando mucho dinero.

El esquema de seguridad debe estar presente sea cual sea la estrategia de desarrollo elegida, sin embargo es necesario tener especial cuidado con aplicaciones desarrolladas en esquemas multi-tier, debido a que en este tipo de esquemas, los componentes de la capa de regla de negocios pueden ejecutarse de forma remota.

2.1.1 Desarrollos de N-Capas.

El modelo multi-tier (n-capas) de informática distribuida ha emergido como la arquitectura predominante para la construcción de aplicaciones multiplataforma en la mayor parte de las empresas más destacadas del

mundo. Este cambio radical en los modelos de computación, desde los sistemas monolíticos basados en *mainframe* y los tradicionales sistemas cliente-servidor, hacia sistemas distribuidos multiplataforma altamente *modularizables*, representó simplemente la punta del iceberg de lo que estaba por llegar en el mundo del desarrollo de aplicaciones, tal y como se puso de manifiesto en las últimas tendencias de las grandes empresas de tecnología, como Sun con su estrategia Sun Tone, o Microsoft con DotNET (.Net).

2.1.2 Ventajas del Modelo

- Desarrollo paralelo de cada capa.
- Aplicaciones más robustas debido a la *modularidad* del código.
- Mayor sencillez del mantenimiento y soporte: Es más sencillo cambiar un componente que modificar una aplicación monolítica.
- Mayor flexibilidad: Se pueden añadir nuevos módulos para dotar al sistema de nueva funcionalidad.
- Escalabilidad: Una aplicación distribuida bien diseñada puede fácilmente incrementar el volumen de usuarios a los que atiende sin perder rendimiento; solo es necesario incrementar la potencia del hardware. El crecimiento es casi lineal y no es necesario añadir más código para conseguir esta escalabilidad.
- Menores requerimientos de potencia de hardware en las estaciones usuarias. El procesamiento de los datos se realiza en el servidor de

componentes, y es allí donde principalmente se centra la potencia de procesamiento del hardware.

- Como tecnología, las arquitecturas de n-capas proporcionan una gran cantidad de beneficios para las empresas que necesitan soluciones flexibles y fiables para resolver complejos problemas derivados de los cambios constantes.

En la actualidad el foco de la informática está basado en redes de computadoras; por ello, las ventajas del modelo de n-capas han hecho que esta arquitectura se posiciones como la piedra angular en los desarrollos de aplicaciones empresariales.

2.1.3 Adaptabilidad a las necesidades empresariales.

El desarrollo de aplicaciones en n-capas es un proceso iterativo de división del problema en piezas manejables denominadas componentes. Estos componentes, o *Componentes de Negocio* (Business Objects) son elementos de software basados, típicamente, en la abstracción de un objeto real, un evento o un proceso de negocio. Los componentes software individuales pueden formar parte y adaptarse tanto a estructuras independientes como a sistemas colaborativos.

El diseño de aplicaciones en n-capas es ideal para la creación de sistemas adaptables, donde cada componente puede ser utilizado y

reutilizado en nuevas combinaciones para satisfacer requisitos de negocio dinámicos. Esto permite a los desarrolladores y a las nuevas aplicaciones reutilizar componentes existentes que modelan lógica de negocio sobradamente probada. En un entorno tremendamente dinámico como el actual, utilizar aplicaciones basadas en diseños de n-capas le permite a las empresas proporcionar valor a sus clientes de manera ágil y adaptable, reduciendo además los costos de mantenimiento de sus sistemas.

2.1.4 Sistemas de negocio integrados.

Uno de los mayores retos que están afrontando hoy en día los negocios electrónicos (e-business), es la integración de las nuevas inversiones en tecnologías emergentes con las inversiones realizadas, hasta el momento, en desarrollo de software.

El marco de desarrollo en n-capas facilita la integración de nuevas y antiguas tecnologías, proporcionando soluciones que enlazan los procesos críticos de negocio conservando el valor de inversiones en tecnologías previas.

Las soluciones integradas de n-capas se pueden extender a lo largo de las aplicaciones de empresa y las de sus socios estratégicos, para

permitir transacciones y procesos de negocio cruzados entre compañías, a la vez que proporcionan un servicio de alto valor para sus clientes.

Con este tipo de desarrollo, el proceso de cambio tecnológico en las empresas puede ser gradual lo que facilita el manejo financiero de las inversiones en tecnología.

2.1.5 Desarrollo de aplicaciones basado en componentes.

El surgimiento de la tecnología de componentes distribuidos es la clave de las arquitecturas de n-capas. Estos sistemas de computación utilizan un número variable de componentes individuales que se comunican entre sí utilizando estándares predefinidos y estructuras de comunicación como:

- CORBA: Common Object Request Broker Architecture, del Object Management Group (OMG).
- DNA: Distributed interNet Architecture; de Microsoft (incluye COM/DCOM y COM+ además de MTS, MSMQ, etc.).

- EJB: Enterprise Java Beans, de Sun Microsystems.
- XML: eXtensible Markup Language, del World Wide Web Consortium (W3C).

Estas y otras tecnologías en rápida evolución proporcionan la infraestructura necesaria para operar en entornos complejos con múltiples plataformas, permitiéndole a las compañías ganar capacidad de computación distribuida.

2.1.6 Elementos de la tecnología.

Desde el punto de vista funcional, las aplicaciones informáticas realizan 3 tareas:

- Reciben *inputs* del operador de la aplicación.
- Almacenan los *inputs* como datos en un repositorio.
- Procesan los datos de acuerdo a las reglas de negocio de la empresa.

La arquitectura *multi-tier* permite que cada tarea sea gestionada individualmente por alguna de sus capas. El modelo de 3-capas, mostrado en la figura 1.0, es un caso particular de la arquitectura multi-tier. La descripción funcional de cada capa se detalla a continuación:

1. **Capa de Presentación:** Es la interfase de usuario o capa de presentación de datos. A través de esta capa, el usuario puede ingresar datos, visualizar resultados e interactuar con el sistema subyacente. En entornos Web, el llamado "browser" realiza estas

funciones; en aplicaciones de escritorio se realizan a través de las pantallas del sistema, mas conocidas como *formularios*.

2. Capa de Lógica y Procesos: Son los componentes que encapsulan la lógica o reglas de negocio de una organización. Estas reglas reflejan los procedimientos, requisitos y secuencia que debe seguir el procesamiento de la información corporativa. Pueden estar orientadas a tareas simples, o puede ser parte de una serie elaborada de tareas de un *business workflow*. En el caso particular de una aplicación web, esta capa está constituida por los componentes COM registrados como parte de una aplicación transaccional.

3. Capa de Datos: Es el repositorio donde se almacenan los datos. En esta capa, los datos colectados desde las aplicaciones distribuidas, quedan disponibles para ser consultados posteriormente. En general, la capa de datos está representada por gestores de base de datos estructurado, o por almacenes de datos no estructurados como los repositorios de correo electrónico.

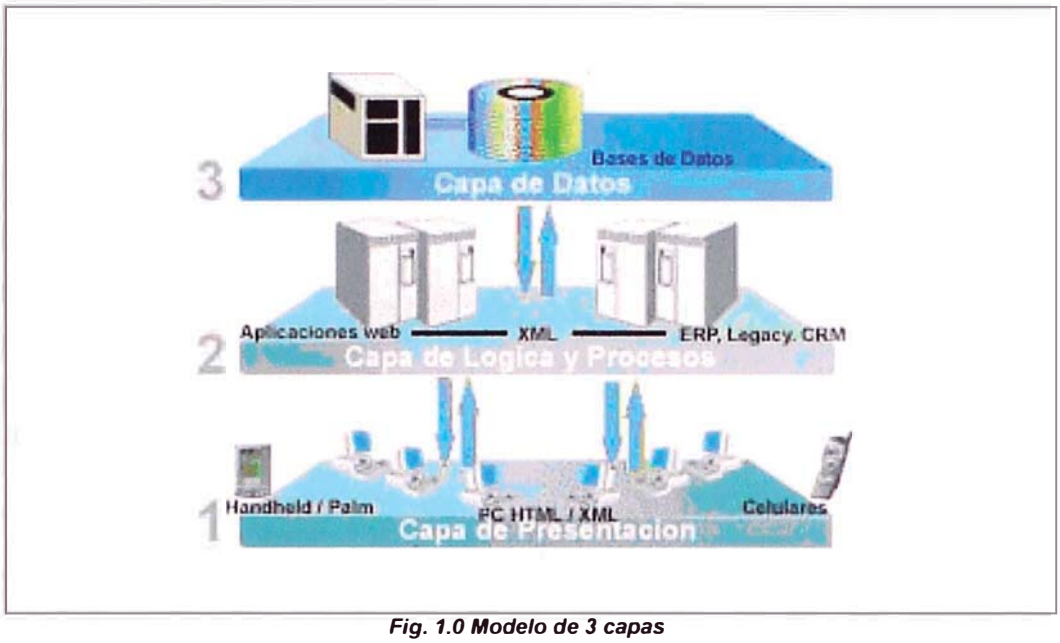


Fig. 1.0 Modelo de 3 capas

Asimismo, desde una perspectiva técnica podemos representar el diseño de software en tres niveles: Conceptual, Lógico y Físico. El nivel físico es el que revela con mayor detalle la interacción de los elementos software en una arquitectura de tres capas. Ver figura 2.0.

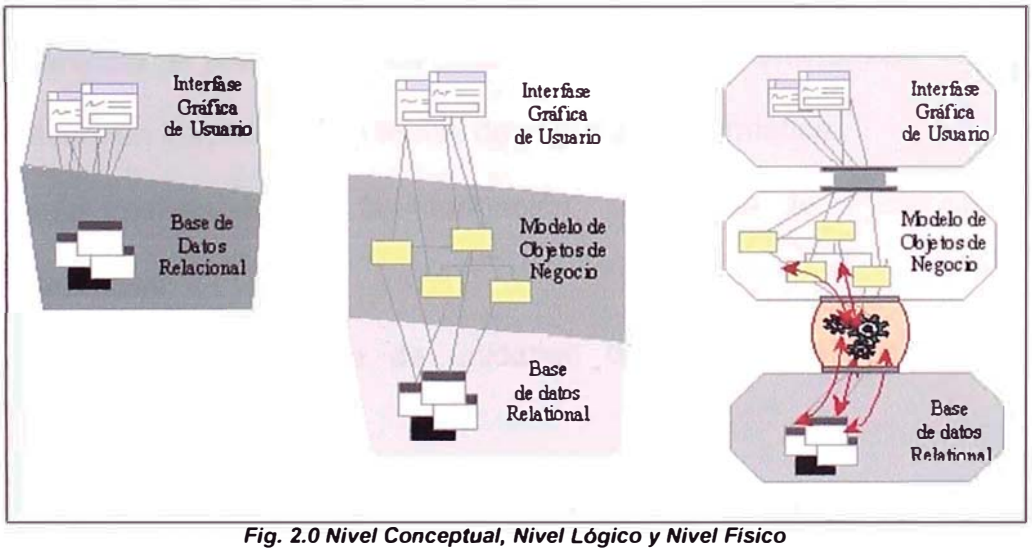


Fig. 2.0 Nivel Conceptual, Nivel Lógico y Nivel Físico

CAPÍTULO III

PROCESO DE TOMA DE DECISIONES

3.1 PLANTEAMIENTO DEL PROBLEMA

Los principales problemas identificados respecto a los sistemas de información de la empresa fueron:

- Repositorios de datos dispersos e inconsistentes.
- Inexistencia de auditoría de datos.
- Carencia de procedimientos para validaciones administrativas.
- Inversión insuficiente en temas de seguridad informática.
- Baja confiabilidad de la información. Evidencia de problemas de seguridad.
- Dependencia del Uso de unidades de Red. Unidades de Red Estáticas.
- Conocimiento altamente peligroso de Estructuras de Datos.
- Usuarios con conocimientos potencialmente peligrosos, de lenguajes de programación.

Repositorios de datos dispersos e inconsistentes.

Fueron el origen de que reportes acerca de lo mismo, no coincidan. Las áreas utilizaban distintas bases de datos, distintas tablas maestras e incluso distintos términos para referirse a lo mismo. El personal de las áreas operativas y administrativas, no compartía un "idioma" común.

Inexistencia de auditoría de datos.

La insuficiencia de personal de sistemas, el poco acercamiento de la jefatura al área de sistemas mina y la poca valoración que la alta dirección de la empresa le daba a las actividades de sistemas originaron que el aspecto de auditoria de datos fuera desatendido; debilidad que fuera el motivo de que las deficiencias de seguridad no se detectaran oportunamente.

Carencia de procedimientos para validaciones administrativas.

Los procedimientos administrativos necesarios para delinear los pasos a seguir para el procesamiento y consolidación de la información mensual no existían o no se cumplían de acuerdo a lo establecido. Asimismo, dadas las amplias distancias entre oficinas administrativas de la unidad minera, algunos procedimientos que si estaban establecidos no se cumplían debido a carencias de personal y de medios de comunicación como correo electrónico, anexos telefónicos y radios. La carencia de procedimientos y las dificultades para comunicarse facilitó que la modificación irregular de los datos no sea detectada durante la operación diaria.

Inversión insuficiente en temas de seguridad informática.

El nivel de inversiones en tecnología se enfocaba principalmente en la renovación de equipos para los usuarios de las áreas administrativas, desatendiendo aspectos de seguridad integral de la red corporativa. La poca valoración que la alta dirección le daba al área de sistemas y la ineficacia de los responsables para comunicar la importancia estratégica que tienen las inversiones en tecnología de información y comunicaciones, fue la causa principal de la desatención que la alta dirección le dio a estos aspectos. Era necesario comunicar eficazmente que renovar equipos no era suficiente. Con una plataforma tecnológica obsoleta y con grandes deficiencias de seguridad, las condiciones estaban dadas para que en cualquier momento los datos sean alterados.

Baja confiabilidad de la información. Evidencia de problemas de seguridad.

Tiempo atrás se habían observado algunos problemas de seguridad en cuanto a los permisos de acceso a los datos. El personal de sistemas había encontrado que algunas tablas de datos (archivos .dbf) habían sido modificadas irregularmente, desvirtuando su consistencia. Los datos en las tablas habían sido modificados haciendo uso de las conexiones de red que los usuarios de los sistemas tenían configuradas permanentemente en sus estaciones de trabajo. Pudo identificarse que los datos no tenían valores originados por transacciones (conjunto de acciones lógicamente relacionadas y ejecutadas sobre distintas tablas de base de datos), sino que habían sido hechas específicamente sobre ciertas tablas con el fin de

obtener resultados específicos en ciertos reportes. Sin embargo, a pesar de identificar qué se hizo, no fue posible identificar quién lo hizo, pues al ser una modificación hecha por fuera y no a través de los sistemas de información, los campos que guardaban los datos de auditoria (usuario, fecha y hora de creación / modificación) habían sido llenados arbitrariamente.

Dependencia del Uso de unidades de Red. Unidades de Red Estáticas.

Los usuarios tenían una configuración de unidades de red predefinida, es decir, que dependiendo del sistema o los sistemas que utilizaban, tenían creadas estáticamente unidades de red, que eran accesibles en cualquier momento, aún cuando no se estuvieran utilizando los sistemas de información.

La arquitectura de los sistemas de información minera, que implementaba ésta deficiente configuración, se describe a continuación:

1. La base de datos estaba almacenada en un servidor UNIX, en forma de tablas libres (.dbf), cuya restricción de acceso a ella estaba completamente controlada a través de permisos a nivel de carpetas y archivos para los usuarios de red. Los permisos en este servidor UNIX manejaban accesos de lectura / escritura.

2. La lista de usuarios con los que se hacia la asignación de permisos era la lista de usuarios del servidor UNIX, la cual estaba sincronizada a la lista de usuarios del Servidor Principal de Dominio (PDC) que era un Servidor con sistema Operativo Windows NT 4.0 y utilizaba una interfaz que mantenía este sincronismo entre las lista de usuarios de ambos Sistemas Operativos.

3. Todos los usuarios de los sistemas contaban con sistemas operativos Windos 98 o XP, desde los cuales levantaban los sistemas de información hechos en FoxPro DOS 2.6.

4. Todos los ejecutables de los Sistemas de Información Minera (.exe) estaban también centralizados en un mismo lugar en la red, ubicados en el mismo servidor UNÍX. Desde ésta ubicación central, los usuarios ejecutaban los sistemas en forma simultánea. Se había optado por esta forma de trabajo para evitar realizar instalaciones en las estaciones de trabajo de cada uno de los usuarios, cada vez que algún sistema de información era actualizado, producto de las tareas de mantenimiento.

Conocimiento altamente peligroso de Estructuras de Datos.

Los usuarios de los Sistemas de Información Minera, eran usuarios que durante su permanencia en la empresa habían adquirido conocimientos acerca de las estructuras de la tablas y de la forma como los datos se

almacenaban en ellas. Conocían, a partir de que tablas se generaban los reportes, cual era la estructura, y en que carpetas estaban ubicadas exactamente dentro del servidor UNÍX. Los usuarios desarrollaron este conocimiento debido a que requerían elaborar reportes adicionales, que los sistemas de información no les proveían; los procedimientos y el tiempo necesario para la aprobación e implementación de un nuevo módulo en los sistemas les hacían buscar una solución alterna para sus necesidades. La solución natural que los usuarios encontraron para sortear ésta situación, fue informarse e indagar, a través de las unidades de red, acerca de cuales eran las tablas que guardaban resultados, para que a partir de éstas, ellos mismos puedan crear sus propias *pequeñas aplicaciones* que obtuvieran datos de aquellas tablas y generaran los reportes que necesitaban.

Usuarios con conocimientos potencialmente peligrosos, de lenguajes de programación.

Los usuarios también se auto capacitaron en el uso de herramientas de desarrollo y lenguajes de programación. La organización también había adquirido algunos productos de software, que permitían cubrir parcialmente las necesidades de los usuarios; estos productos o aplicaciones visuales, trabajaban con bases de datos que se guardaban localmente en la estación de trabajo de los usuarios, las cuales igualmente eran accedidas a través de entornos de desarrollo como Visual Basic.

De esta manera, los usuarios no solo conocían el lenguaje Fox Pro para DOS, sino que también conocían entornos de desarrollo como Visual Basic, una herramienta que les brindaba suficiente capacidad para acceder y modificar los datos.

Es así como convergen éstas tres situaciones entre los usuarios de los sistemas de información: Conocimiento de las estructuras de las tablas, permisos para leer/escribir en estas, y conocimientos suficientes de programación como para realizar acciones sobre sus datos.

Antecedentes del problema.

Se habían evidenciado alteraciones en los resultados mensuales, en sistemas cuyo *output* era utilizado para realizar el pago mensual de servicios a las Empresas Especializadas (conocidas también como Contratas). Los datos en los que se había identificado alteraciones, hasta entonces, fueron los siguientes:

Base de Datos del Sistema de Planillas: Se encontró que el pago mensual a las empresas especializadas, experimentaba variaciones evidentemente anormales. El incremento del pago por concepto de tareas trabajadas sufría incrementos sin explicación aparente; posteriormente se identificó que se habían incrementado irregularmente la cantidad de tareas trabajadas por parte de los Colectivos o Grupos de Trabajo organizadas por las Empresas

Especializadas. Se requirieron muchas horas hombre para contrastar la información digitada contra la información en papel, finalmente se encontraron las enormes diferencias, sin embargo no fue posible identificar plenamente a los responsables, puesto que las modificaciones se habían realizado irregularmente y no a través del Módulo de Digitación de Tareo. El monto en exceso pagado a las contratas fue descontado en los meses siguientes.

Base de Datos del Sistema de Valorización: Una de las funciones del Sistema de Valorización es hacer el Cálculo de Incentivos. Los incentivos son pagos que hace la organización a las Empresas Especializadas a modo de premiación, por haber superado la meta mensual programada. El incentivo se paga en función del avance en las labores lineales, y el tonelaje de mineral extraído en los tajos. El *Input* para éste sistema, son los registros de medición mensual, los cuales son llenados por el departamento de ingeniería, con información proporcionada por sus topógrafos. En este caso también se verificaron variaciones irregulares en términos de costo, por lo que se realizó una revisión de datos que concluyó en que se había producido la alteración de los mismos.

3.2 ALTERNATIVAS DE SOLUCION

Independientemente de la alternativa de solución a seguir, la situación requirió que los problemas de base sean comunicados con claridad a la Gerencias de Sistemas y a la Alta Dirección. Cualquier alternativa de solución sería insuficientemente para la empresa si no se solucionaban los problemas de orden y comunicación existentes entre las áreas administrativas y de operación. Estos dos aspectos, aunados a los problemas técnicos de seguridad, habían propiciado la alteración de los datos corporativos.

Creación Dinámica de Unidades de Red

La principal deficiencia del esquema de seguridad utilizado, era mantener permanentemente unidades de red predeterminadas para los usuarios de los sistemas de información. Tener unidades de red permanentes, significaba que los usuarios podían acceder a los directorios y tablas de base de datos, en cualquier momento, haciendo uso del explorador de Windows. Esto les daba la capacidad de explorar estructuras y datos utilizando entornos de desarrollo como los de FoxPro o Visual FoxPro.

Esta facilidad de acceso podría reducirse en cierta medida, si los sistemas tuvieran la capacidad de conectar dinámicamente la unidad de red al inicio de su ejecución, y desconectarlas al momento de su finalización. Sin embargo, esta solución solo es útil si es implementada desde el principio y

se guarda estricta confidencialidad de la ubicación de los archivos de base de datos en la red. De otra manera, el usuario podría manualmente crear la conexión de red y obtener nuevamente el acceso a la base de datos. Una gran deficiencia de éste esquema era que el usuario estaría en la capacidad de averiguar la ubicación física de la base de datos (sea ejecutando el comando DOS adecuado o utilizando nuevamente el explorador de Windows) mientras estaba en ejecución alguno de los sistemas que creó dinámicamente la unidad de red.

Es importante señalar que ésta alternativa pudo haber sido suficiente si hubiera sido implementada desde un principio y se hubiera evitado generar indirectamente la necesidad de tener usuarios programadores en la red, ávidos de conocer ubicaciones y estructuras de tablas. Sin embargo, bajo éste esquema, el riesgo de accesos no autorizados no hubiera desaparecido ya que, como se mencionó, este es igualmente insuficiente cuando realmente existen usuarios con la firme intención de acceder irregularmente a la base de datos.

Restricción de Permisos de Instalación de entornos de Desarrollo.

El control de instalaciones de entornos de desarrollo en las estaciones de trabajo también disminuye el riesgo de modificaciones mal intencionadas a las tablas. Sin embargo, no elimina la posibilidad de destruir o corromper los archivos físicos de la base de datos. Por otra parte para restringir los permisos de instalación en las estaciones de trabajo, son necesarios

sistemas operativos específicos como Windows NT Workstation o Windows XP; sin embargo en la realidad la mayoría de usuarios de la empresa utilizaba, hasta entonces, el sistema operativo Windows 98.

Utilizar una nueva arquitectura para encapsular la Base de Datos y garantizar su acceso únicamente a través de los Sistemas de Información Minera.

La viabilidad de ésta alternativa surge originada por la existencia del Proyecto de Reingeniería de los Sistemas de Información Minera. Este proyecto consideraba hacer la migración de la base de datos FoxPro a la base de datos SQL Server. Con esto se tenía cubierto solo uno de los elementos de riesgo para la seguridad de la información. Por otro lado, según el Proyecto de Reingeniería, todos los Sistemas de Información serían cambiados a un entorno visual (Visual FoxPro), por lo tanto todas las aplicaciones debían ser nuevamente desarrolladas. El Esquema de desarrollo no estaba definido para éste proyecto, pero se tenia la intención de utilizar un modelo de 2 capas en la que se utilizaría SQL Pass-through como forma de acceso a la Base de datos. Sin embargo éste esquema no cubría las exigencias de seguridad exigidas por la empresa debido a que las cuentas de acceso a la base de datos serían accesibles para los desarrolladores de los sistemas. Por otro lado, el Proyecto de Reingeniería de los Sistemas de Información Minera, no había tomado en cuenta el aspecto de seguridad y control del código fuente.

Observadas las deficiencias del proyecto original, se planteó trabajar bajo un esquema de desarrollo en 3 capas, que separarían la base de datos, las reglas de negocio e interfaces de usuario, haciéndolos independientes desde el punto de vista funcional. Adicionalmente, se definió el esquema de trabajo para garantizar la seguridad del código fuente dentro de un entorno de programación colaborativo.

El uso de componentes, sin embargo, al igual que cualquier otra arquitectura de software, tenía deficiencias que había que tomar en cuenta. En este caso, los sistemas debían estar preparados para posibles ataques de usuarios programadores. Específicamente, el riesgo subyacente era: la ejecución remota, no autorizada, de procedimientos invocados por usuarios con conocimientos de programación.

Identificado el riesgo, se inició el diseño de los mecanismos de seguridad que se aplicarían a la Capa de Componentes, con el fin de garantizar su uso exclusivo a través de los sistemas de información minera y la protección de la confidencialidad de las Reglas de Negocio de la Empresa, cuyo desarrollo, a pedido de la Gerencia de Sistemas, debía ser asignado solo a personal de confianza.

Evidentemente, todo este nuevo planteamiento hizo necesario evaluar el beneficio marginal que significaría reformular el Proyecto de Reingeniería. La evaluación financiera del cambio requerido sobre el proyecto original,

tomando en cuenta los recursos adicionales necesarios y los beneficios esperados, se muestra en los anexos. Los resultados indicaron que la inversión adicional sería rentable y que se recuperaría en un período menor a 2 años.

3.3 METODOLOGÍA DE SOLUCIÓN

Encarados los problemas de procedimiento y comunicación que estaban detrás de los problemas de seguridad de los datos, se procedió a desarrollar una metodología que, desde el punto de vista técnico, alcanzara los niveles de seguridad de datos exigidos por los directivos de la empresa.

El modelo de seguridad desarrollado y expuesto en el presente trabajo, está compuesto por la aplicación de los siguientes componentes:

- El Esquema de desarrollo en 3 capas.
- El Esquema de Seguridad basado en Claves.
- Definición de Estándares de Programación orientados a la Seguridad.
- Distribución del equipo de desarrollo.
- Control de Código Fuente.

3.3.1 El Esquema de desarrollo en 3 capas.

La tecnología seleccionada para el desarrollo de los sistemas, basada en componentes, técnicamente proporciona niveles de seguridad muy

completos. Uno de los niveles de seguridad que más se acercaba a nuestros requerimientos era el Nivel de seguridad basado en roles. Este nivel de seguridad restringe el uso de componentes solo a aquellos usuarios miembros del rol autorizado, de tal forma que podíamos configurar nuestro Servidor de Componentes para que solo cierto grupo de usuarios utilice solo cierto grupo de componentes. Sin embargo, además de toda la carga de trabajo que el servidor hubiera tenido que realizar para autenticar a un usuario cada vez que éste instanciaba un componente (lo cual es muy frecuente en ésta tecnología), subsistía el peligro principal para nuestros sistemas: el uso indebido o irregular de los componentes por parte del personal autorizado, vale decir, por parte de usuarios-programadores con capacidad de acceder irregularmente a los datos.

3.3.2 El Esquema de Seguridad basado en Claves.

A la luz de las limitaciones de la tecnología frente al tipo particular de riesgo que no nos permitía cubrir con todos los aspectos de seguridad identificados; se decidió asignar recursos a la creación de una metodología flexible y complementaria que nos permitiera afrontar, además de los riesgos convencionales, el riesgo de tener usuarios autorizados con capacidades de programación y conocimiento de nuestras estructuras de datos.

La metodología, resultado de éste trabajo, está principalmente basada en el manejo de claves descentralizadas que permiten, a la vez, un mantenimiento flexible.

Antes de iniciar con la descripción de la metodología, es necesario explicar el significado de algunos términos a utilizar:

1. Clave de Creación: Secuencia encriptada de caracteres *case-sensitive* que se subdivide y almacena descentralizadamente. Permite la ***creación*** de una instancia operativa de componente.

2. Clave de Ejecución: Secuencia encriptada de caracteres *case-sensitive* que se subdivide y almacena descentralizadamente. Permite la ***ejecución*** de los métodos de una instancia operativa de componente.

3. Registro del Sistema Operativo: Repositorio de información del Sistema, donde se almacena información relacionada a la configuración de aplicaciones, parámetros funcionales inherentes al Sistema Operativo, entre otros.

4. Componente: Elemento software compilado, representado generalmente por un archivo de extensión .dll. Un componente es una definición de Clase, que contiene atributos y métodos que le dan funcionalidad. Estos atributos y métodos están implementados en el archivo de extensión .dll.

5. Instancia de Componente: Es una variable que *replica* la definición de un componente. Una instancia de componente tiene los atributos y la misma funcionalidad que su clase padre. Una instancia de componente es en si una instancia de clase y por lo tanto, es un objeto.

6. Componentes Proxy: Es un elemento software que realiza llamadas RPC a una definición de Componente ubicado en un Servidor remoto. Un Componente Proxy no tiene implementados ni atributos ni métodos; opera como vía de comunicación con una definición de Componente Remoto, logrando una funcionalidad igual a la de este.

7. Servidor de Componentes: Es un equipo físico que ofrece el Servicio de Administración de Componentes. El servicio de administración de componentes permite crear Aplicaciones e instalar definiciones de componentes (archivos .dll).

8. Servidor de Base de Datos: Es el Sistema Manejador de Base de Datos encargado de almacenar, administrar y asegurar la integridad y el acceso autorizado a nuestros datos.

9. Base de Datos Maestra: Es la base de datos principal de los sistemas; hace la función de Tabla de Localización de Bases de Datos, dado que almacena la dirección de Instancia y el Nombre de la Base de Datos,

según el año al que se desea consultar. Almacena también parámetros globales del Sistema Integrado de Información Minera.

10. Componentes Críticos: Son aquellos componentes remotos necesarios para la creación de Componentes No Críticos. Sus funciones básicas son:

    a. Proporcionar conectividad a los componentes No Críticos.
    b. Permitir el proceso de Validación de claves.

    Los Componentes Críticos son únicamente dos, y las definiciones de clase que las originan son:

    a. CDatabase
    b. CRegistry

11. Componentes NO Críticos: Son todos aquellos componentes que hacen uso de las Componentes Críticos. Los Componentes No Críticos obedecen a una Clave de Ejecución que valida la funcionalidad de sus métodos.

    Un componente inicializado con una Clave de Ejecución no válida, no poseerá funcionalidad. Los componentes No críticos están constituidos

por todos los demás componentes remotos originados por definiciones de clase diferentes de CRegistry y CDatabase.

3.3.3 Definición de Estándares de Programación Orientados a la Seguridad.

Esta metodología de seguridad, esta compuesta de dos partes fundamentales: Primero, la definición de la estrategia a seguir, y segundo de los lineamientos que hay que cumplir para garantizar que nuestra estrategia funcione adecuadamente.

Parte fundamental de los lineamientos que se deben cumplir para lograr el objetivo, es seguir adecuadamente los estándares de desarrollo establecidos. Esta metodología se apoya en el cumplimiento estricto de las definiciones de Estándares de Programación orientados a la seguridad.

Se establecieron como estándares de programación de código fuente, dos tipos de mecanismos:

1. Creación de Pseudoconstructores en cada componente (sea crítico o no crítico)
2. Validación claves en cada uno de los métodos de componente.

Creación de Pseudoconstructores

Los *pseudoconstructores* fueron creados para emular a los *contructores* propios de lenguajes como Java y Lenguaje C. Su finalidad es la de inicializar las propiedades o atributos del componente.

En éste caso, el constructor se encarga de ***establecer la conectividad*** del componente y de ***recepcionar la clave de ejecución*** proporcionada desde la aplicación cliente.

Validación de Claves

Se definió un conjunto estándar de líneas de código para validar la Clave de Ejecución enviada desde la aplicación cliente. Esta Clave de ejecución se compara contra la Clave Real de Ejecución obtenida por un Procedimiento de obtención de Claves que arma la clave completa a partir de dos secuencias obtenidas desde el Servidor de Componentes y desde el Servidor de Base de Datos. Ver Figura 3.0.

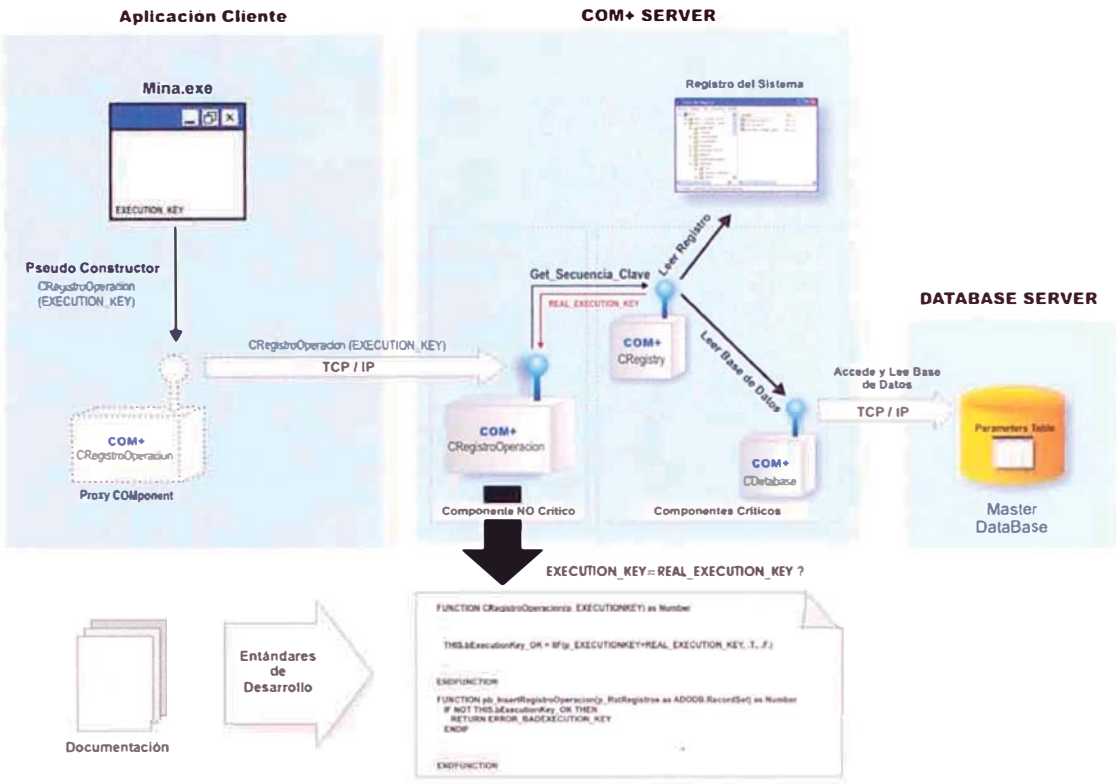


Fig. 3 Validación de Claves

3.3.4 Distribución del equipo de desarrollo.

Debido a experiencias previas y a la cultura organizacional de la empresa, la información es considerada un intangible muy valioso que no debe caer en manos de personas no autorizadas. La información acerca de Resultados de Exploraciones Mineras, Información Geológica, Leyes, Tonelajes de Mineral extraídos, etc. es información estratégica que debe mantenerse en reserva; más aún si se tiene a otras empresas mineras competidoras, explorando muy cerca de los denuncios de la empresa.

Por esta razón, la Gerencia de Sistemas y Costos consideró pertinente asignar el desarrollo de Componentes Remotos (Reglas de Negocio), exclusivamente, sólo al Personal de Confianza de la organización; mientras que, tanto los Componentes Locales como las Aplicaciones Cliente, debieron ser desarrollados por el Personal contratado.

Los gráficos mostrados a continuación muestran la subdivisión realizada sobre las actividades de desarrollo de software en el marco del Proyecto de Reingeniería de los Sistemas de Información Minera, tomando en cuenta la decisión de la Gerencia de Sistemas y Costos, respecto a la confidencialidad de las reglas de negocio de la empresa:

Fig. 4 Desarrollo de Componentes Críticos

(Se tiene acceso a la Clave de Creación del Componente CRegstry)

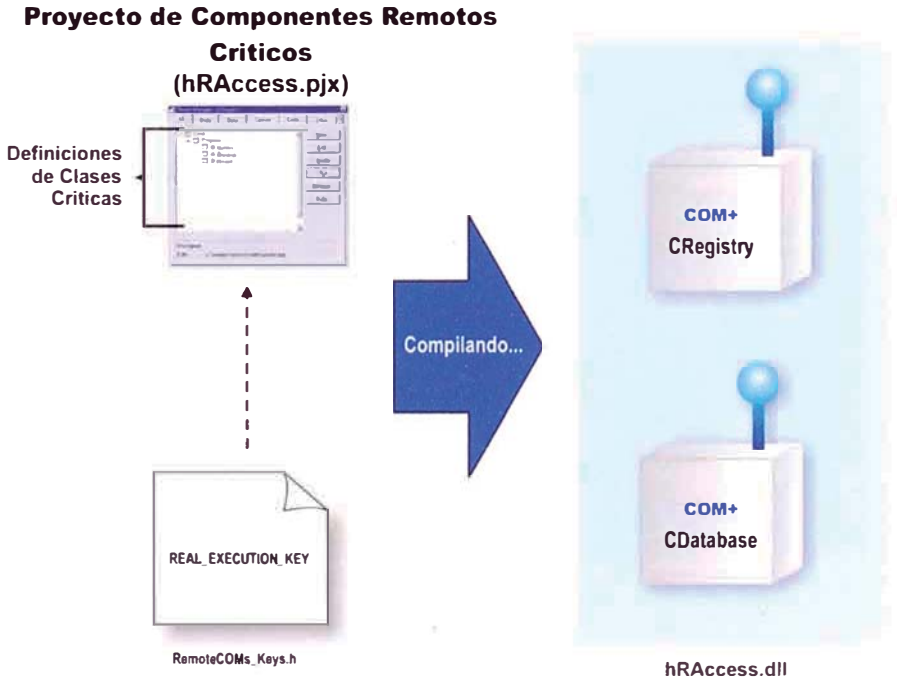


Fig. 5 Desarrollo de Componentes No Críticos

(Se tiene Acceso a la Clave de Creación del Componente CRegstry)

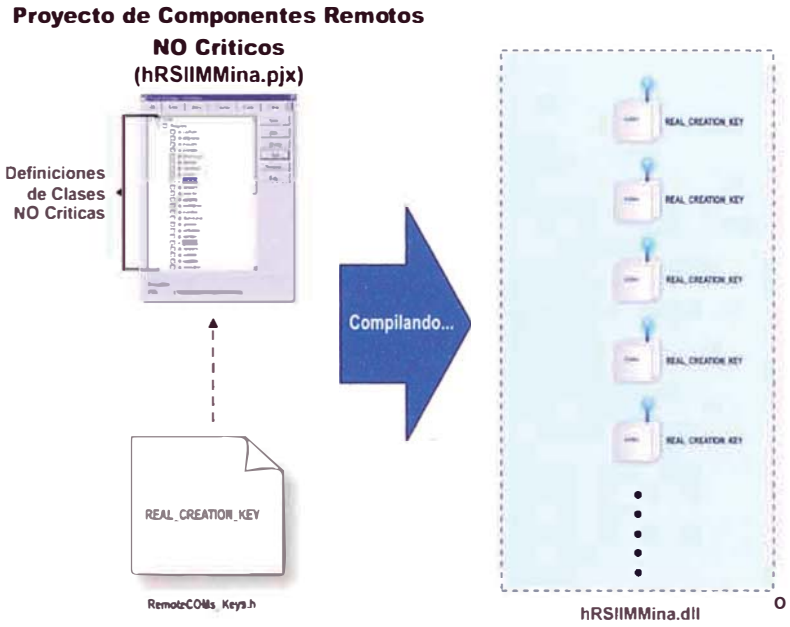


Fig. 6 Desarrollo de Componentes Locales

(Se tiene Acceso, solamente, a la Clave de Ejecución de los Componentes)

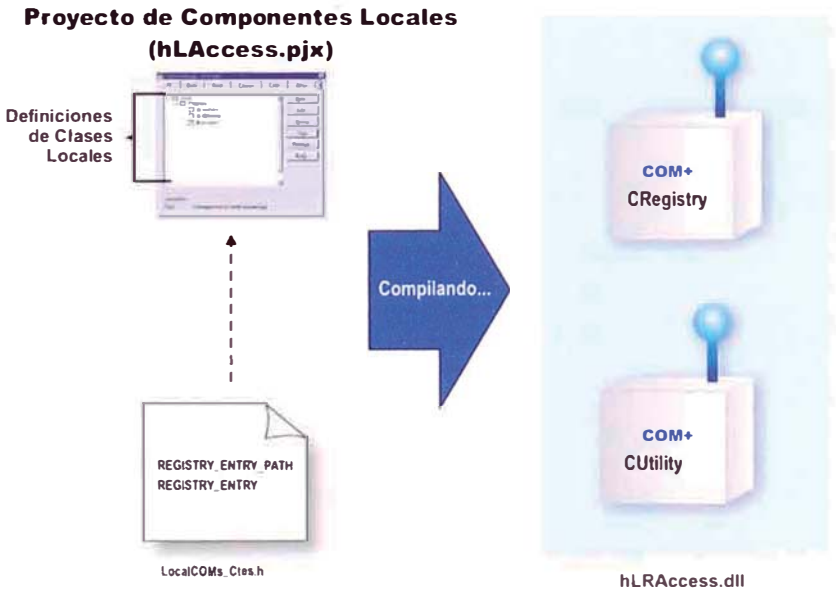


Fig. 7 Desarrollo de Aplicaciones Cliente

(Se tiene Acceso solo a la Clave de Ejecución de los Componente Remotos)

Las aplicaciones cliente se ubican adecuadamente, de forma que desde estas, sólo se puede conocer la Clave de Ejecución de los Componentes No Críticos. Al no conocerse mas que aquella clave, desde el nivel de aplicación cliente solo es posible crear instancias y utilizar componentes no Críticos, aislándose y protegiéndose a los componentes Críticos.

Interacción entre Componentes Críticos y No Críticos

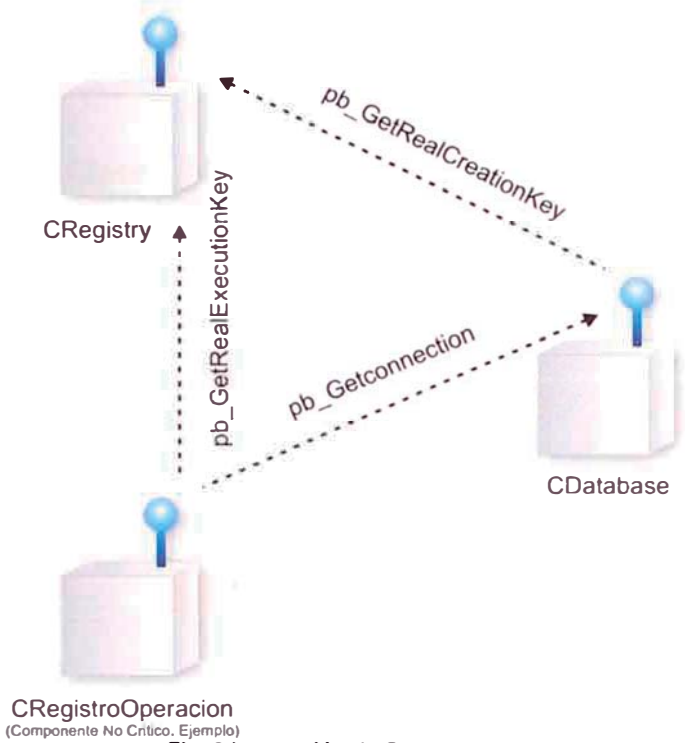


Fig. 8 Interacción de Componentes

Distribución del Equipo de Desarrollado

A su vez el desarrollo de Aplicaciones Cliente se restringió completamente solo al uso de instancias de componentes No Críticos. Los desarrolladores de aplicaciones cliente debían conocer, a lo más, la clave de ejecución de las Clases No Críticas. El personal designado para el desarrollo de ésta capa fue el personal contratado o personal que no tenía la categoría de empleado de confianza.

El desarrollo de las Clases Críticas (CRegistry y CDatabase) fue asignado a personal de confianza de la empresa; su definición se realizó por única vez y el código fuente se almacenó y aisló en forma segura. Las clases Críticas,

por su importancia y nivel de uso debían estar comprobadamente 100% libre de errores.

El desarrollo de Clases No Críticas también se asigno a personal de confianza de la organización. Toda clase No Critica hace uso de las Clases Críticas, obligatoriamente, para obtener Conectividad y Funcionalidad.

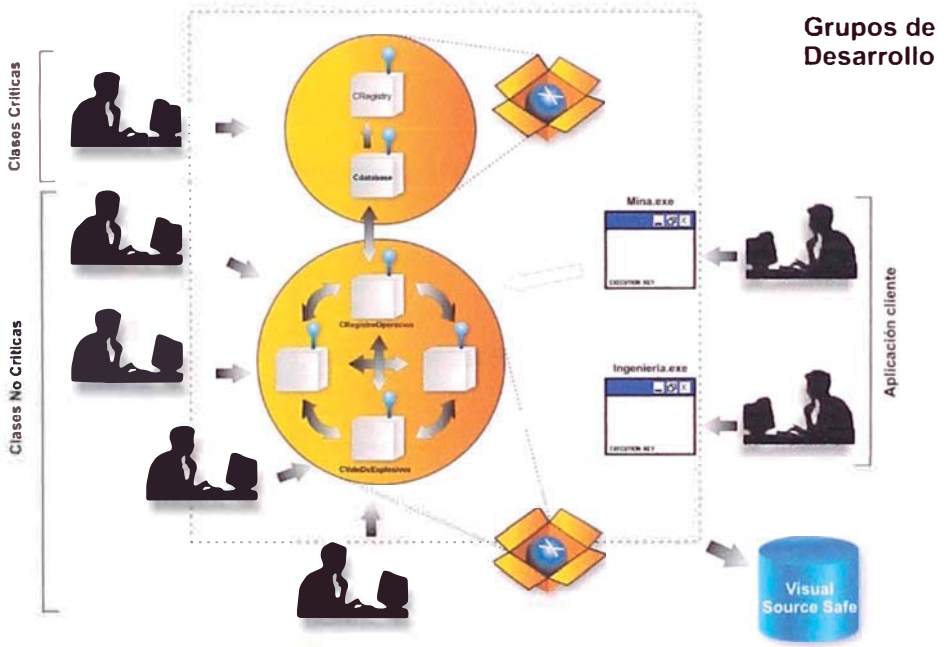


Fig. 9 Designación de Grupos de Desarrollo

3.3.5 Control de Código Fuente.

Para prevenir el acceso no autorizado al código fuente, tanto de componentes como de aplicaciones cliente, se adquirió una herramienta de

control de versiones de código fuente, proporcionada por Microsoft: *MS Visual SourceSafe®*.

Esta herramienta nos ha permitido guardar, desde entonces, un historial de versiones para cada uno de los objetos utilizados en el desarrollo. Para cada versión de algún documento o programa, se almacena un comentario, y el nombre del usuario modificador, lo cual hace posible rastrear la evolución de cada archivo.

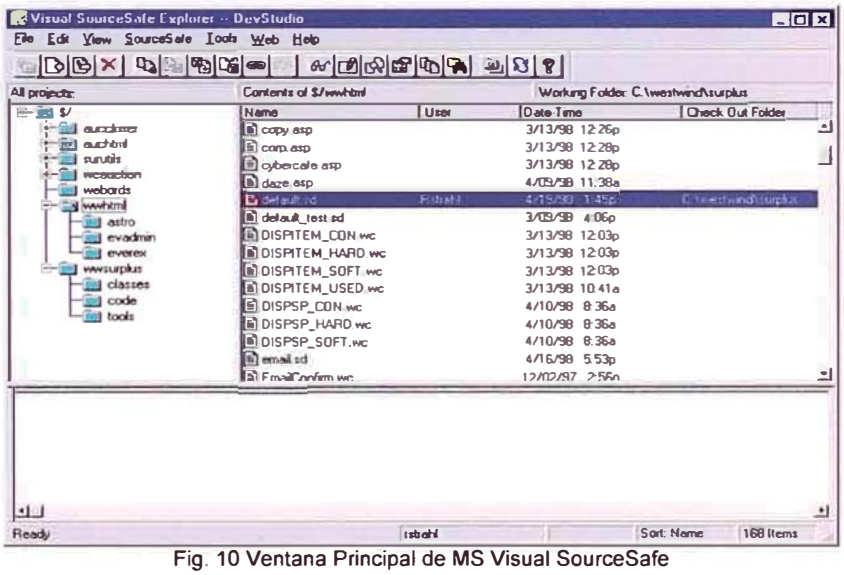


Fig. 10 Ventana Principal de MS Visual SourceSafe

Visual SourceSafe está integrada al entorno de desarrollo de Visual Fox Pro, lo que permite el control automático de versiones. Por otro lado, el nivel de acceso al código fuente es completamente configurable a través de un módulo de administración.

Esta herramienta nos facilita el desarrollo en paralelo de los sistemas, permitiéndonos además compartir código común, manteniendo los lineamientos de seguridad y permisos de acceso.

CAPITULO IV

DESCRIPCIÓN DE LA METODOLOGÍA

4.1 SEGURIDAD BASADA EN CLAVES

La búsqueda de un esquema seguro, para la utilización de componentes que permita encapsular las reglas de negocio de la empresa, concluyó en la definición de una metodología basada en claves.

La idea básica de éste esquema era validar la creación y el uso de los componentes a través de dos tipos de clave: La Clave de Creación y la Clave de Ejecución.

Definición de Clases

Una definición de Clase es aquel conjunto de líneas de código, que luego de ser compilado, da origen a un Componente COM.

Dentro de la metodología, se definieron dos tipos de clases: Las Clases Críticas y las Clases No Críticas.

La definición de las clases, incluye un mecanismo que asegura que su componente COM relacionado solo pueda ser creado y utilizado si las claves correspondientes son las correctas.

Este esquema divide las Clases contenidas en los Componentes en dos tipos:

Clases Críticas: Son aquellas que son necesarias para la creación de Clases No Críticas. Las funciones básicas de las clases críticas son: proporcionar conectividad a los componentes no críticos, y permitir el proceso de validación de claves.

CDataBase: Establece los inicios de sesión del servidor de base de datos, según el sistema al que se acceda. Proporciona la conectividad entre Componentes No Críticos y base de datos.

CRegistry: Permite el acceso al registro del Servidor de Componentes, a efectos de realizar el proceso de validación de claves.

Cada una de estas Clases obedece a una ***Clave de Creación*** particular que es la que validará la funcionalidad de sus métodos. Una instancia de clase inicializada con una Clave de Creación no válida, no poseerá funcionalidad.
La creación de los Componentes Críticos, se realiza por única vez, y es implementada en forma separada a la implementación de Componentes No Críticos.

Los desarrolladores de Componentes no críticos sólo pueden hacer uso de Componentes Críticos Compilados y no tienen acceso a su Código Fuente.

El Código Fuente de los Componentes Críticos se guarda separadamente y no es accesible por ningún desarrollador. No es modificable por ningún motivo, excepto por modificaciones en los inicios de sesión existentes en la base de datos.

La clase crítica CDatabase, que proporciona conectividad con la base de datos a las Clases no críticas, establece los parámetros de conexión y de inicio de sesión según el contexto desde el cual se instancia la Clase No Crítica.

Todo componente no Crítico hace uso de un Componente Crítico para proveerse de conectividad.

Clases No Críticas: Son todas aquellas clases que hacen uso de las Clases Críticas. Las Clases No Críticas obedecen a una ***Clave de Ejecución*** que validará la funcionalidad de sus métodos. Una instancia de clase inicializada con una Clave de Ejecución no válida, no poseerá funcionalidad.

La finalidad de proteger los Componentes mediante claves de Creación y Ejecución es mantener 2 niveles de utilización para éstos:

En el Primer Nivel, las Clases Críticas solo deben ser utilizadas desde dentro de Clases No Críticas. Las Clases Críticas se definen una única vez, para de ahí en adelante, ser utilizadas por las Clases No Críticas. Un ***Programador de Componentes*** solo necesita conocer la Clave de Creación de la Clase ***CRegistry.*** La Clave de Creación de la clase ***CDataBase*** y la Clave de Ejecución que gobierna a todos los componentes se guardan *encriptadas* en el Registro de Sistema del Servidor de Componentes y en la Base de Datos Maestra, y son transparentes para el programador.

En el segundo nivel está el ***Programador de Aplicaciones Cliente*** quien solo debe utilizar Clases No Críticas. Para esto, el programador deberá conocer la *Clave de Ejecución* de éstas clases, las cuales estarán guardadas como parte del Proyecto de Aplicación Cliente. Solo existe una Clave de Ejecución para todas las Clases No Críticas.

Observación: El Programador de Aplicaciones Cliente también podría instanciar Clases Críticas, sin embargo para que estos objetos tengan funcionalidad, el programador tendría que conocer las Claves de Creación (que no están a su disposición en el proyecto de desarrollo).

Las claves reales, necesarias para Crear y Utilizar componentes, desde la Capa de Regla de Negocios, son las siguientes:

Categoría de Componente	Componente	Tipo de Calve Requerida	Ubicación
Críticos	CRegistry	Creación	Archivo de Claves Remotas (remotecoms_keys.h)
	CDatabase	Creación	Archivo de Claves Remotas (remotecoms_keys.h)
No Críticos	CRegistroOperacion	Ejecución	Archivo de Claves Remotas (remotecoms_keys.h)

Las claves reales, necesarias para Crear y Utilizar componentes, desde la Capa de Presentación (aplicación cliente), son las siguientes:

Categoría de Componente	Componente	Tipo de Calve Requerida	Ubicación
Críticos	CRegistry	Creación	No accesible
	CDatabase	Creación	No Accesible
No Críticos	CRegistroOperacion	Ejecución	Archivo de Claves Locales (siim_constants.h)

La figura 11 muestra la estrategia utilizada para guardar y validar las claves. El almacenamiento de las Claves de Creación y Ejecución es descentralizado y *encriptado*. Cada clave esta compuesta por dos secuencias de datos, la primera de las cuales se almacena en una entrada del Registro de Sistema, en el Servidor de Componentes; la segunda se almacena en la Base de datos maestra que contiene información para el direccionamiento de todos los sistemas hacia las bases de datos operacionales, de acuerdo al año y la empresa de la que se trate.

Teniendo en cuenta la idea de que no hay esquema totalmente seguro, se decidió almacenar las claves de manera descentralizada en dos servidores distintos: El servidor de componentes (COM+ Server) y el Servidor de Base de Datos (Database Server); éste esquema incrementa la dificultad de obtener las claves, dado que es necesario vulnerar dos servidores en lugar de uno, para obtenerlas.

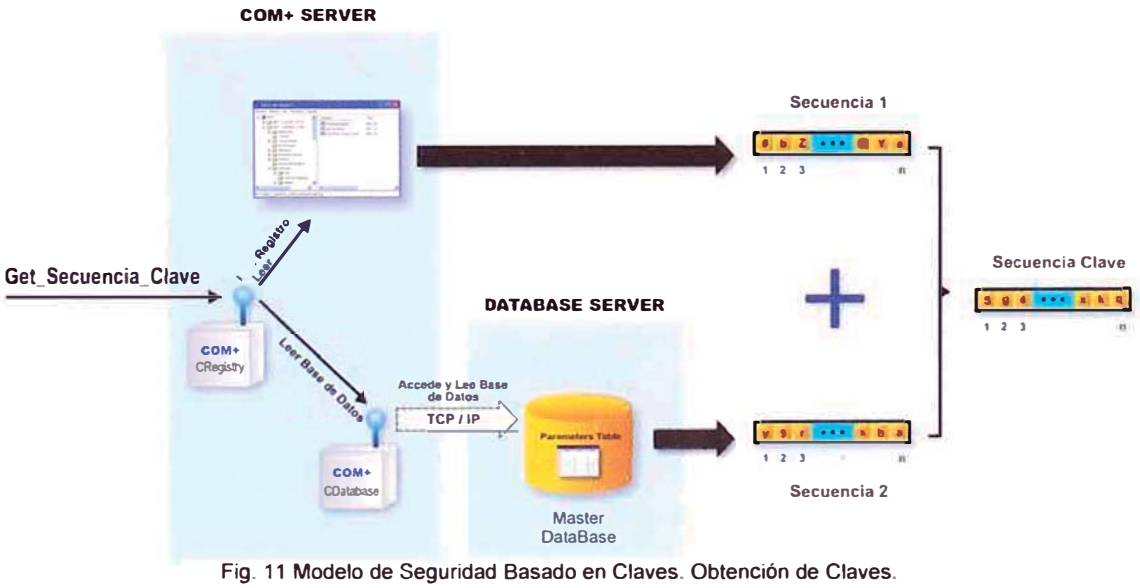


Fig. 11 Modelo de Seguridad Basado en Claves. Obtención de Claves.

La lectura de Claves se lleva a cabo haciendo uso, justamente, de los Componentes Criticos originados por: CRegistry y CDatabase, de ahí el calificativo.

Dada su función de obtención de claves, el desarrollo de Componentes Críticos debe ser separado y diferenciado, del desarrollo de todos los demás componentes remotos.

Almacenamiento y Obtención de Claves

Cada una de las claves, la de Creación y la de Ejecución, se componen de un par de secuencias de caracteres, cada una de estas secuencias se ubican, de forma encriptada, en dos repositorios distintos:

a. En el Registro del Servidor de Componentes: Se crean entradas, con nombres y rutas estandarizadas y predefinidas, en el Registro del Servidor de Componentes. Estas entradas almacenan, en forma encriptada, la primera parte de las secuencias que componen las Claves de Creación y Ejecución. Las entradas del registro están protegidas por los niveles de seguridad del Sistema Operativo.

El acceso al registro se lleva a cabo a través del Componente Crítico originado por la definición de clase CRegistry, el cual debe ser instalado en el Servidor de Componentes por el mismo usuario que crea las entradas en el Registro.

b. En la Base de Datos Maestra: La tabla de parámetros del Sistema Integrado, ubicada en la Base de Datos Maestra, almacena en forma *encriptada* la segunda parte de las secuencias que componen las Claves de Creación y Ejecución.

El acceso a la base de datos maestra se realiza a través del Componente Crítico originado por la definición de clase CDatabase.

4.1.1 Mantenimiento de Claves

Las claves reales de creación y ejecución, al estar compuestas cada una por dos secuencias de caracteres almacenadas, separadamente, en el Registro del Servidor de Componentes y en la Base de Datos, duplican la dificultad de ser vulneradas por algún atacante informático. El proceso de mantenimiento de claves pude requerir la *recompilación* de componentes críticos, no críticos y/o aplicaciones cliente, dependiendo de la clave que se vaya a modificar.

- **Mantenimiento de Clave de Creación**

Las secuencias de la Clave de Creación pueden ser cambiadas en cualquier momento; sea en el Registro del Servidor de Componentes, en la base de datos maestra, o en ambos lugares. Esto no afectará a las Aplicaciones Cliente; sin embargo, será necesario *recompilar* las clases Críticas y No Críticas.

- **Mantenimiento de Clave de Ejecución**

Si se cambian las secuencias de la Clave de Ejecución en el Registro del Servidor de Componentes o en la base de datos maestra, las Aplicaciones Cliente mostrarán un mensaje avisando que la Clave de Ejecución con la que fue compilada no es la correcta y la aplicación cliente perderá su funcionalidad. Para reactivar la funcionalidad de las aplicaciones cliente, dentro del proceso de mantenimiento de la clave de ejecución, tenemos dos alternativas:

1. Compilamos nuevamente las Aplicaciones Cliente con la Nueva Clave de Ejecución y las distribuimos a los usuarios.

2. En la estación de trabajo de cada usuario, ejecutamos una *entrada de registro* para almacenar la nueva Clave de Ejecución en el Registro Local del sistema, desde donde la Aplicación Cliente leerá alternativamente la clave de Ejecución, luego de haber comprobado que la Clave de ejecución con la que fue compilada ya no es válida.

4.1.2 Actualización automática de versiones y claves

Uno de los riesgos que había que afrontar era el cambio intempestivo de Claves. Teóricamente, un cambio de claves en el registro del Servidor de Componentes haría que absolutamente todos los sistemas perdieran su

funcionalidad. Es por esto que se aplicó un esquema de actualización automática de versiones que permitiera mantener todos los sistemas actualizados tanto con los cambios funcionales, como con los cambios de clave. La figura siguiente muestra el funcionamiento de éste esquema:

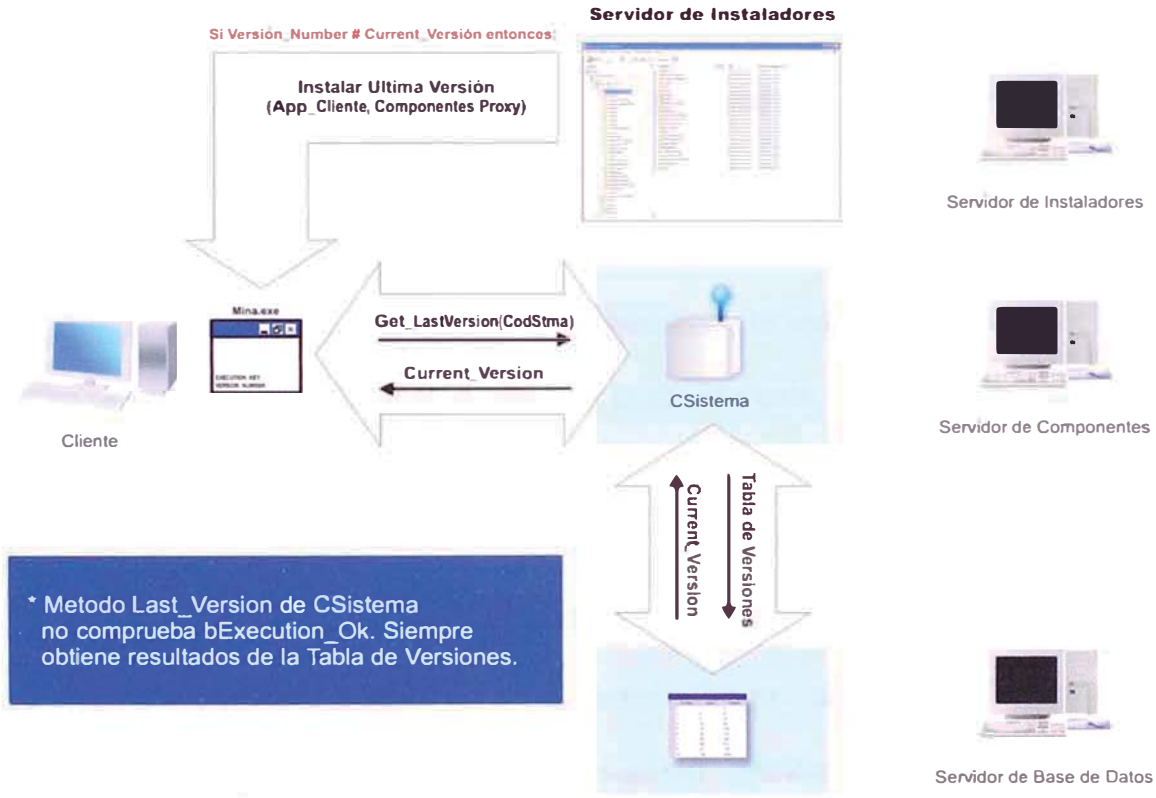


Fig. 12 Esquema de Actualización Automática de Versiones y Claves

La versión y el código de clave son almacenados en el registro del Servidor de Componentes y, de modo compilado, en la Aplicación Cliente. Ambos, código de clave y versión, son comparados cada vez que se intenta activar el sistema. Si se encuentra alguna diferencia entre estos códigos, se inicia

automáticamente el proceso de instalación de la última versión de los sistemas, conteniendo las nuevas claves a utilizar.

4.1.3 Esquema alterno a la actualización automática de versiones y claves

En éste esquema, si no existiera coincidencia entre la clave compilada en la aplicación cliente y las secuencias clave almacenadas en el Servidor de Componentes y en la base de datos maestra, la aplicación cliente buscará una clave válida en el registro local de Windows, la cual, reemplazará a la clave con la que fue compilada. Una vez realizado esto, se vuelve a realizar la comparación.

La nueva clave válida ubicada en el registro local de Windows es colocada allí a través de una *entrada de registro* aplicado en la estación de trabajo cliente. La figura 13, mostrada a continuación, grafica la secuencia lógica que sigue este procedimiento:

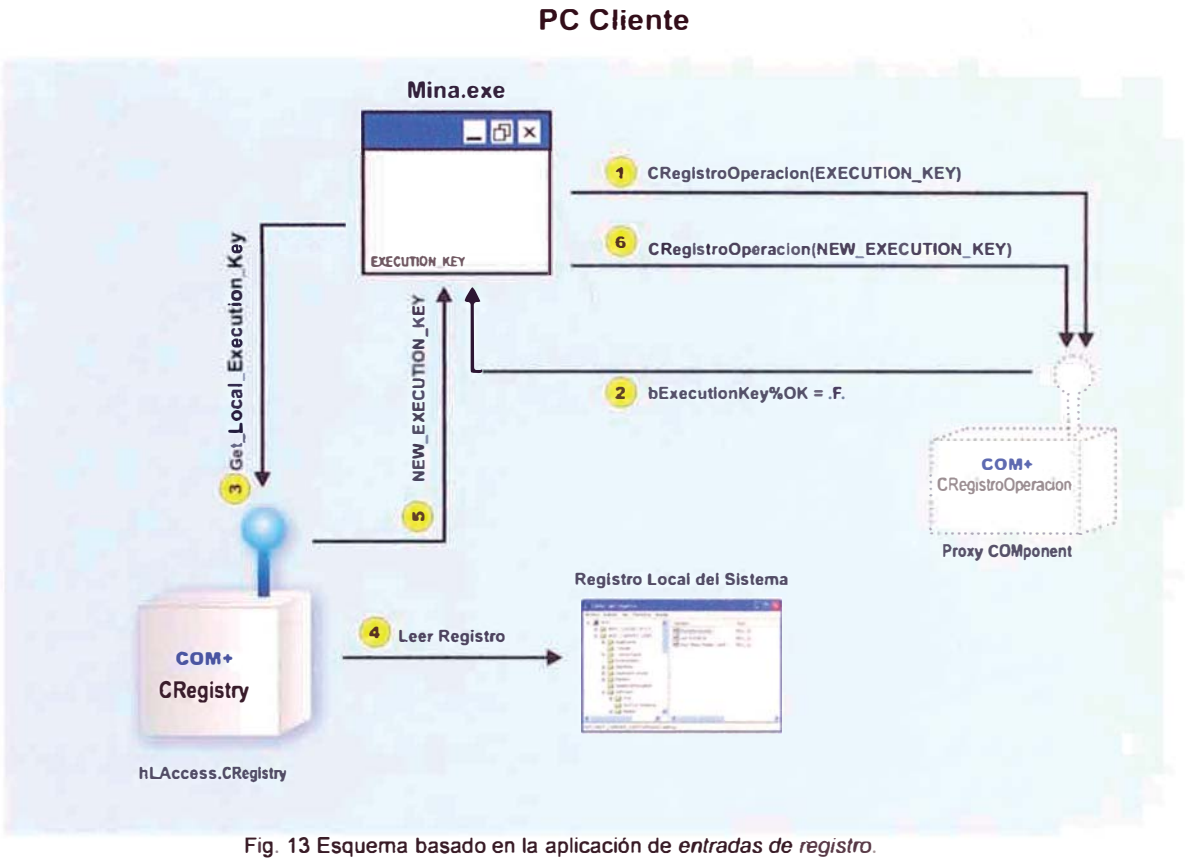


Fig. 13 Esquema basado en la aplicación de *entradas de registro*.

Es necesario señalar, sin embargo, que esta solución alterna es riesgosa puesto que las claves son ubicadas temporalmente en la estación de trabajo del usuario, pudiendo ésta ser accedida si el usuario decide realizar la revisión del registro. Esta es una de las desventajas que tiene frente a la actualización automática de versiones.

CAPITULO V

DECISIONES, ESTRATEGIAS Y RESULTADOS

5.1 TOMA DE DECISIONES

La Gerencia de Sistemas y Costos, de acuerdo a la evaluación financiera realizada, decidió optar por la alternativa de: *Utilizar una nueva arquitectura para encapsular la Base de Datos y garantizar su acceso únicamente a través de los Sistemas de Información Minera.* Las decisiones subsiguientes, derivaron de la decisión central y fueron seleccionadas de entre las alternativas disponibles:

Especificación de procedimientos.

Los procedimientos no definidos fueron elaborados por el área de sistemas en coordinación con las Gerencias administrativas. La Alta Dirección tomó participación activa de este proceso debido a que los procedimientos bosquejan las reglas de negocio de la empresa. Parte principal de los procedimientos se enfocó en los procesos de cierre de mes, en los procesos de corrección autorizada de datos y en la gestión de la información histórica.

Mejoramiento de los medios de comunicación.

La carencia de recursos de comunicación había sido una de las causas que permitió el "*incumplimiento justificado*" de los procedimientos implícitamente establecidos. La empresa decidió entonces autorizar un recurso que no se tenía hasta entonces en la unidad minera: el uso de correo electrónico interno para el personal administrativo. Asimismo, se decidió incrementar el número de anexos y radios para mejorar la comunicación, tanto al interior de la unidad minera, como la comunicación que se establecía diariamente con la sede en Lima.

Selección de la Tecnología a utilizar en el desarrollo de los nuevos sistemas.

Se decidió implementar una arquitectura distribuida, que permitiera independizar nuestra base de datos, las reglas de negocio y la presentación de los datos. Para esto se utilizó el modelo de desarrollo en tres capas, implementando el mecanismo de seguridad basado en claves para cubrir vulnerabilidades previamente identificadas en esta tecnología. Es necesario mencionar en este punto, que el nivel de vulnerabilidad de la tecnología de tres capas, es exactamente la misma, independientemente del lenguaje de programación que se utilice para desarrollarla.

Selección de Software.

La herramienta seleccionada para el desarrollo de componentes y aplicaciones cliente fue Visual FoxPro 8.0. Esta decisión se tomó, por ser una herramienta confiable y principalmente, porque la organización es socio

estratégico de Microsoft® y cuenta con las licencias originales de ésta herramienta de desarrollo desde la versión 2.6 para DOS.

Por otra parte, el manejador de base de datos seleccionado fue SQL Server de Microsoft®, por ser el más adecuado para gestionar el volumen de información que generaba la empresa. Los nuevos servidores adquiridos para renovar la red de datos de la empresa incluyeron, sin costo adicional, licencias del último sistema operativo de red lanzado por Microsoft®.

Adquisición de Hardware para procesamiento.

Se adquirieron equipos adicionales para ser utilizados como servidores de componentes y de base de datos.

Incremento del ancho de banda en la Wave LAN.

Para la comunicación entre el área de Sistemas Mina y los centros de digitación apartados, se decidió aumentar el ancho de banda original, de 10Mbps a 64Mbps. Se realizó la adquisición de una nueva antena, sin embargo durante las pruebas de operación se comprobó la perdida de algunos paquetes de datos, por lo que fue finalmente la Wave LAN fue configurada para operar a 54Mbps evitando la pérdida de datos.

Contratación de personal externo.

Se decidió, también, considerar dentro del presupuesto reajustado, la contratación de nuevos desarrolladores, a la vez que el tiempo estimado

para la culminación del proyecto tuvo que ser recalculado y ampliado. Se decidió que las tareas de desarrollo de componentes serían realizadas solo por el personal de confianza de la empresa, mientras que las aplicaciones cliente serían desarrolladas por el personal contratado.

Políticas de mantenimiento de claves.

Se decidió establecer un cronograma de mantenimiento de claves a cargo del administrador de base de datos, el administrador de redes y el administrador del código fuente. Se decidió también que terminada cada etapa de desarrollo, las claves utilizadas fueran completamente renovadas antes de ponerlas en el ambiente de producción.

Implementación de Políticas de Auditoría de datos.

Se estableció que los procesos críticos, al interior de las reglas de negocio en los sistemas, mantuvieran un registro de las acciones, usuarios y direcciones de red que activaban dichos procesos, de forma que pudiera hacerse un seguimiento efectivo en caso se evidenciaran problemas. Asimismo, se establecieron programas de auditoría interna de datos para comprobar periódicamente la seguridad y confiabilidad de los datos.

Se tomaron estas decisiones de acuerdo al análisis de riesgo mostrado en el cuadro siguiente:

Cuadro de Riesgos y Alternativas de Acción

Riesgo	Alternativas	Efecto	Selección
Alteración de datos a través de conexiones de red estáticas	Creación de conexiones de red, dinámicas.	Evitar tener acceso permanente a los datos; sin embargo no evita la alteración de éstos durante la ejecución de aplicación que genera la conexión de red dinámica.	No
	Cambiar a manejador de base de datos relacional.	El acceso a los datos sólo es posible previa autenticación con el Manejador de Base de Datos.	Si
Utilización indebida de componentes a través de entornos de desarrollo.	Restringir permisos para instalación de entornos de desarrollo desde el S.O Windows XP.	Impide la instalación de entornos de desarrollo; sin embargo: a. No todos los usuarios contaban con Windows XP, S.O. que permitía esta configuración. b. Algunos usuarios requerían la utilización de macros dentro de la herramienta MS Office, para lo cual necesitaban necesariamente tener habilitado su entorno de desarrollo.	No
	Aplicar un diseño de seguridad basado en Claves, que evite la necesidad de deshabilitar entornos de desarrollo como el de MS Office, necesario para algunos usuarios.	Permite la utilización de los componentes, solo desde dentro de las aplicaciones o sistemas de información. Todo uso de componentes a través de entornos de desarrollo, requiere de las claves de Creación y Ejecución.	Si
Detención de la operación de los sistemas por Cambio de Clave	Utilizar el esquema que implementa la actualización automática de versiones.	Evitaría la detención de la operación de todos los sistemas de información. La instalación de los sistemas en las estaciones de trabajo se mantendría constantemente actualizada con la última versión y por lo tanto con el último cambio en las claves.	Si
	Utilizar la alternativa que usa "*entradas de registro*" para que las aplicaciones clientes operen con la nueva clave.	Los sistemas al no encontrar coincidencia entre sus claves y las claves ubicadas en el servidor de componentes, hacen una búsqueda de la nueva clave en el registro local de Windows, la cual es nuevamente comparada con las claves ubicadas en el servidor de componentes. El riesgo de ésta alternativa es alto dado que las claves de creación y ejecución se estarían ubicando temporalmente en la estación de trabajo del usuario y podría ser accesible por un usuario programador que se tome el trabajo de buscarlas.	No
Acceso NO autorizado al Código fuente de Componentes Críticos, No Críticos y Aplicaciones Cliente	Utilizar una herramienta de Control de Versiones y acceso al código fuente.	Todo el Código fuente permanece centralizado en un solo repositorio. El permiso que tendrá cada desarrollador es configurable para cada Proyecto de Componentes dentro de su entorno de desarrollo y desde el entorno de Visual SourceSafe. Cada desarrollador solo puede ver lo que debe y necesita ver; esto garantiza el conocimiento restringido de las claves en función de las tareas de desarrollo que se le han designado (desarrollo de Componentes Criticos, desarrollo de componentes No Criticos ó desarrollo de Aplicaciones Cliente).	Si
Conocimiento de secuencias de clave, a través de los desarrolladores.	Cambio de clave antes de la puesta a producción de los sistemas de información.	Las clave solo deben ser conocidas por los responsables de la implantación de los sistemas. Las claves utilizadas durante el desarrollo deberán ser completamente diferentes a la clave utilizada en los sistemas en producción.	Si

5.2 ESTRATEGIAS ADOPTADAS

Habiendo expuesto, ante la Gerencia de Sistemas y la Alta Dirección, los problemas de orden y comunicación que subyacían a los problemas de seguridad de la información; y luego de aprobarse e implementarse la iniciativa de mejorar los medios de comunicación y definir los procedimientos administrativos y operativos (reglas de negocio) en coordinación con las gerencias y la Alta Dirección; se continuó con el desarrollo del Modelo de Seguridad basado en claves.

a. Definición de Estándares de Trabajo: Previa a la fase de desarrollo se establecieron los lineamientos de orden y nomenclatura a seguir.

   Plantillas de Código: Líneas de Código de comprobación de Claves de Creación y Ejecución. La clave de Ejecución es comprobada en cada uno de los métodos de componente.

   Librerías de Clase: Se establecieron los lineamientos de utilización de la herencia en las clases visuales, comunicación mediante parámetros, etc.

   Componentes Comunes: Se identificaron las definiciones de clase comunes a todos los sistemas, se crearon plantillas de código para el mantenimiento de tablas maestras.

Componentes de Sistema: Se separaron las definiciones de clase inherentes a cada sistema en particular.

b. Instalación de Sistemas Operativos más seguros y configurables respecto a permisos de instalación de programas: Se utilizó el sistema operativo Windows XP Professional para restringir la Instalación de programas y entornos de desarrollo no autorizados.

Alcance de los mecanismos de Seguridad en el Modelo:

1. Etapa de Desarrollo:

    a. Control de Código Fuente: Cuentas de usuario y permisos en proyectos SourceSafe.
    b. Distribución de las tareas de desarrollo según la categorización del personal.

2. Etapa de Producción:

    a. Aplicación Cliente:
        i. Control de acceso por Módulos.
        ii. Control de acciones sobre cada Módulo.
        iii. Control de alcance de Visualización.

b. Componentes:

   i. Control de acceso por Roles.

   ii. Control de acceso por Claves de Creación y Ejecución.

c. Base de Datos:

   i. Control de acceso a objetos de Base de Datos: Por inicios de sesión y función se controló el acceso a bases de datos, tablas, procedimientos almacenados, etc.

   ii. Control de acciones campos de tabla: Control sobre sentencias *Select, Insert, Update y Delete*.

5.3 EVALUACIÓN DE RESULTADOS

Resultados Reales

Hasta el momento se han obtenido resultados satisfactorios en términos de:

- Tiempo de respuesta: El tiempo de respuesta se ha reducido a la décima parte gracias al servidor de componentes dedicado a administrar las transacciones, y a la ampliación del ancho de banda.

- Facilidad de Mantenimiento: Se prevé que el tiempo de mantenimiento por módulo se reducirá en aproximadamente 30%,

gracias a la estandarización de código y el orden establecido durante la etapa de desarrollo.

Utilización más eficiente del ancho de banda de la Wave LAN: La red inalámbrica establecida entre la oficina de sistemas-mina y las zonas altas de la unidad minera posee ahora una velocidad de 54Mbps, la cual es aprovechada más eficientemente gracias a la arquitectura distribuida de los sistemas. El uso más eficiente del ancho de banda ha influido también en el tiempo de respuesta.

Es necesario destacar en éste punto que, luego de la ampliación del ancho de banda, el tiempo de respuesta de los sistemas antiguos con arquitectura centralizada se redujo a su sexta parte: procesos que anteriormente demoraban 30 minutos en concluir, ahora lo hacían en 5 minutos. En su conjunto, la ampliación del ancho de banda y el uso de arquitectura distribuida redujeron el tiempo original de respuesta a su décima parte.

Por tanto, fue evidente que, en términos de velocidad de respuesta, el ancho de banda había constituido uno de los "cuellos de botella" más importantes en la comunicación con los usuarios ubicados en las zonas alejadas del departamento Sistemas-Mina.

Dificultades y Problemas encontrados

Curva de Aprendizaje: Esta fue una de las dificultades particulares que se presentaron. Dada la condición de tener que desarrollar componentes utilizando el lenguaje de programación Visual FoxPro, una combinación nueva en el mercado peruano de desarrollo de sistemas, todo el equipo de desarrollo de componentes requirió pasar por la etapa de capacitación.

La sintaxis y estructura de componentes creados con Visual FoxPro era diferente a la utilizada en Visual Basic; lenguaje de programación muy popular entre los desarrolladores. En este sentido, la experiencia de estos últimos, había perdido su peso específico. La creación de estándares y plantillas de código en Visual FoxPro fue el Factor Crítico que permitió superar ésta dificultad.

Es muy probable que la organización sea la primera empresa peruana en desarrollar, y poner en producción, Sistemas de Información cuyo CORE está 100% constituido por componentes seguros, creados con el lenguaje de programación Visual FoxPro.

Prestaciones de la Herramienta de Desarrollo: El entorno de desarrollo de Visual FoxPro hasta la versión 8.0 ha mejorado mucho, sin embargo, aún no ha llegado a tener el mismo nivel de prestaciones que el entorno de desarrollo de Visual Basic, por ejemplo. La creación de componentes

requiere de algunas prestaciones que faciliten al desarrollador trabajar con más de una instancia del entorno a la vez, para realizar la depuración de los componentes y aplicación cliente en forma simultánea. Este tipo de prestaciones reduce mucho el tiempo de desarrollo utilizado para la creación y depuración de componentes. Esta dificultad, sin embargo, fue compensada por la estandarización del código fuente.

En sus últimas versiones, Visual Foxpro ha mejorado mucho el manejo de errores con respecto a sus versiones anteriores y con respecto a otros lenguajes de programación. En sus versiones más recientes ha implementado el uso de las sentencias Try... Catch... Finally para el control de excepciones; copiando el esquema utilizado originalmente por lenguajes de programación como Java. Dentro del Modelo de Seguridad aplicado, más específicamente en las plantillas de código de componentes predefinidas, se establece el uso intensivo de estas sentencias con el fin de obtener también sistemas robustos, mayor estabilidad y mejor control de errores.

Restricciones

Tiempos Cortos: Tiempo de desarrollo cortos considerando que los sistemas habían que hacerse desde cero.

- Versión actualizada del Servidor de Componentes: Se detecto un BUG en el funcionamiento de los componentes creados en Visual Foxpro y el Sistema Operativo del Servidor de Componentes

(Windows 2003). Este problema es solucionado si se bajan las actualizaciones correspondientes desde Internet.

Lenguaje de Programación: La creación de componentes generalmente es realizada usando lenguajes de programación como Visual Basic. Existen amplia disponibilidad de desarrolladores e información al respecto de éste tema; sin embargo el proyecto de reingeniería debía realizarse al 100% utilizando el lenguaje de programación de Visual FoxPro, puesto que la organización es propietaria de la Licencia, la cual es renovada periódicamente. Esta restricción provocó algunos retrasos, causados por la Curva de Aprendizaje (capacitación a los analistas), prestaciones limitadas de la herramienta de desarrollo, poca disponibilidad de información sobre el tema y acceso restringido a Internet.

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

El origen de los problemas de seguridad, no son problemas exclusivamente técnicos. La realidad nos ha demostrado una vez más que son producto de la convergencia de la falta de procedimientos, falta de comunicación, ineficacia en la gestión de recursos, ineficiencia y, finalmente, carencias técnicas.

Las amenazas a la seguridad de los datos no solo se encuentran fuera de la organización. Podemos cerrar las puertas para que el intruso no ingrese; pero él ya podría estar adentro.

6.2 RECOMENDACIONES

Una vez implementado el esquema de seguridad basado en claves y con el fin de garantizar la continuidad de la seguridad de los datos, se recomienda lo siguiente:

Programación del Mantenimiento de Claves

Implementar un programa de mantenimiento periódico de claves. Las claves deben mantenerse en estricta reserva y ser periódicamente cambiadas. Estas deben incluir, no solo a las claves de creación y ejecución al interior del software, sino también a las claves de las cuentas de usuario en general.

- *Control de entornos de desarrollo*

  Desinstalar o desactivas entornos de desarrollo en aquellas estaciones de trabajo que no estén autorizadas a tenerlos. Siempre existirá el riesgo de usar, desde esos entornos, componentes de uso libre creados por terceros como los proporcionados por Microsoft Corporation ®. Estos componentes permiten el acceso a todo tipo de repositorios de datos (conectividad abierta), incluyendo SQL Server.

Protección de información crítica de red

Mantener en reserva la información relativa a componentes críticos de la red corporativa; estos son: nombres y números IP de controladores

de dominio, de servidores de componente, de routers, entre otros. Asimismo, es recomendable que las estructuras de las bases de dato sean de conocimiento exclusivo del personal de sistemas o de quienes le dan mantenimiento.

GLOSARIO DE TÉRMINOS

Archivos .dbf: Archivos utilizados por FoxPro y Visual FoxPro para almacenar datos. Son tablas que pertenecen al motor de base de datos nativo de estos lenguajes de programación.

Block: Lugar geométrico en un plano geológico que delimita un área correspondiente a una veta.

Browser: Navegador. Expresión utilizada generalmente para denotar el software de navegación utilizado para obtener información desde la Internet.

Business Workflow: Denominación de la secuencia de actividades que se realizan al interior de una organización.

Case-sensitive: Que diferencia mayúsculas de minúsculas.

COM: Microsoft® Component Object Model. Arquitectura de software creada por Microsoft Corporation.

Desencriptar: Decodificar un mensaje encriptado para que sea legible en condiciones normales.

Encriptar: Codificar un mensaje para que sea ilegible en condiciones normales.

Entrada de Registro: Archivo con extensión .reg que al ejecutarse almacena una secuencia de caracteres en una ubicación específica del registro local del sistema en el cual se ejecuta.

Ley de Oro: Es la cantidad de gramos de oro contenida en una tonelada de material extraído en la operación minera.

Mainframe: Ordenador central.

Modularizable: Que puede ser dividido en dos o más módulos.

Multi-tier: Multicapa. Denominación utilizada para referirse a una estructura con dos o mas capas.

Negocio Electrónico: Es la utilización de tecnologías de información y comunicaciones, con fines empresariales.

PDC: Primary Domain Controller. Control Principal de Dominio. Equipo gestor de la red y de los usuarios que a ella pueden acceder.

RPC: Remote Procedure Calls. Llamadas a Procedimiento Remoto.

SANS Institute: SysAdmin, Audit, Network and Security Institute. Es la institución más grande del mundo en certificación y entrenamiento para seguridad de información (http://www.sans.org).

Sistema Colaborativo: Sistema tecnológico mediante el cual un conjunto de individuos realizan actividades con la finalidad de lograr un objetivo común. Se basa en el desarrollo de conocimiento compartido, la aceleración de los flujos de información y la coordinación de los recursos por parte de todos los individuos para reducir costos y tiempo.

Sobrecostos: Costos que pueden evitarse.

SQL Pass-through: Tecnología que permite enviar sentencias nativas de SQL, directamente al servidor de base de datos.

Sun Tone: Programa de Sun Microsystems, dirigido a certificar íntegramente el conjunto de infraestructuras, operaciones y personal de los proveedores de servicios.

Tercerizar: Contratar los servicios de un tercero para que realice parte de un proceso que entes realizábamos.

Winche: Herramienta similar a rastrillo, utilizado para acopiar material mineralizado en sectores específicos del interior de los socavones.

BIBLIOGRAFIA

Mary Kirtland (1998). *Designing Component-Based Applications*. Book & CD-ROM edition. Microsoft Press.

Michael Howard y David LeBlanc (2003). *Writing Secure Code*. 2nd. edition. Microsoft Press.

Kenneth C. Laudon, Jane P. Laudon (2003). *Management Information Systems*. 8ª.edición. Prentice Hall

Richard Brealey, Stewart Myers (2002). *Principios de Finanzas Corporativas*. 7ª.edición. McGraw Hill.

ANEXOS

Anexo 1.1.

Evaluación Financiera de la reformulación del Proyecto de Reingeniería de los Sistemas de información Minera.

El presupuesto original del proyecto se incrementó para cubrir los nuevos requerimientos, producto de la aplicación del modelo de Seguridad.

El incremento del presupuesto correspondió a los siguientes elementos:

- Adquisición de Licencias de MS SQL Server 2000.
- Adquisición de Licencias de MS Visual SouceSafe.
- Adquisición de Servidor de Componentes.
- Adquisición de Servidor de Base de Datos.
- Ampliación del Ancho de Banda de la Wave LAN.
- Contratacion de 3 desarrolladores adicionales para tareas de desarrollo y mantenimiento de sistemas.

El ahorro en costos y beneficios adicionales por la aplicación el Modelo de Seguridad es:

- Valorización Mensual con datos confiables.
- Ahorro por correcto costeo de consumo de materiales.
- Ahorro en Horas-Hombre para tareas de mantenimiento de los sistemas.
- Ahorro en costos por utilización de Horas – Hombre dedicados a la comprobación de la consistencia de los datos.
- Toma de decisiones oportunas, con menor grado de incertidumbre.
- Eliminación del riesgo de fuga de información.

Se ha considerado un horizonte de 24 meses en la evaluación financiera puesto que es el tiempo que está dispuesta a esperar la Gerencia de la empresa para ver redituada la inversión.

El ahorro en costos se percibe una vez implementados los principales sistemas de información que participan en el proceso de valorización mensual. Se estima que los sistemas de información principales estarán en producción a finales del primer año del proyecto, por lo que se han considerados los beneficios a partir del segundo año.
Se estima que la distorsión en los resultados de la valorización y en el cálculo de costos por consumo de materiales alcanza el 0.06%, afectando negativamente a la empresa. La valorizacion mensual por concepto de pago a las empresas especializadas es de 8 millones de dólares americanos, en promedio, mientras que el costo mensual por consumo de materiales es de 2 millones 150 mil dólares americanos.

En promedio, las utilidades anuales de la empresa alcanzan los 8 millones 500 mil dólares americanos. Se estima que el costo de oportunidad anual de la empresa por la no disponibilidad de información real y oportuna es del 0.04% sobre las utilidades anuales. Igualmente se estima que el riesgo permanente de fuga de información equivale a un costo anual de 3% sobre las utilidades anuales.

Se estima que la tasa de cambio promedio durante el horizonte analizado será de 3.20 nuevos soles por dólar americano.

Para determinar la tasa de descuento de los flujos, se está tomando en cuenta la metodología del Costo Promedio Ponderado de Capital (CPPC) de acuerdo a las siguientes consideraciones:

- La política de dividendos de la empresa está definida y no cambia durante el horizonte de tiempo analizado.
- El riesgo del proyecto es semejante al riesgo de la empresa.
- La relación Deuda de Largo Plazo sobre Capital, permanece constante para la empresa durante el periodo analizado (12 meses).

La relación Deuda / (Capital + Deuda) se considera en un nivel del 60%, y el costo explícito de la deuda determinado por el promedio ponderado de todas las deudas de largo plazo de la empresa, alcanza el 14% anual por lo que la tasa efectiva mensual será de 1.098%. Asimismo, se sabe que la tasa impositiva para el sector minero es del 30%.

El Costo Promedio Ponderado de Capital, que es la tasa a la que se descontarán los flujos, está dado por la siguiente expresión:

$$CPPC = \%D * Ki * (1 - T) + \%C * Ke$$

Donde:
%D: Relación Deuda / (Capital + Deuda de la empresa).
Ki: Costo Explícito de la deuda.
T: Tasa Impositiva del sector
%C: Relación Capital / (Capital + Deuda de la empresa), por lo que:

$$\%C = 1 - D\%$$

Ke: Costo de Oportunidad del Accionista

Anexo 1.2.

Evaluación Financiera de la reformulación del Proyecto de Reingeniería de los Sistemas de información Minera (continuación).

El costo de Oportunidad del accionista viene dado por la siguiente expresión:

$$Ke = (1 + Ki) * (1 + R) - 1$$

Donde:
Ke: Costo de oportunidad del accionista
Ki: Costo Explicito de la deuda.
R: Prima por riesgo requerida por el accionista

La prima por riesgo requerida por el accionista es el porcentaje adicional que solicita este, por ser quien cobra después de los acreedores en una eventual crisis financiera. En este caso los accionistas requieren una prima del 5% mensual sobre el costo explicito de la deuda. Reemplazando valores en la expresión anterior, tenemos:

$$Ke = (1 + 1.098\%) * (1 + 5\%) - 1 = 6.15\%$$

Luego, reemplazando valores en la expresión del Costo promedio Ponderado de Capital, tenemos:

$$CPPC = 60\% * 1.098\% * (1 - 30\%) + 40\% * 6.15\% = 2.92\%$$

El CPPC es la tasa con que se descontarán los flujos estimados producidos por la inversión requerida para implementar las decisiones tomadas.

Los valores utilizados en la evaluación marginal del proyecto expandido se muestran en el Anexo 1.3. El detalle de los flujos estimados puede verse en los anexos 2 y 3. El valor actual neto del Flujo de Caja Económico, descontado a la tasa de CPPC=2.92% (mensual), es de 10,123 nuevos soles, lo que equivale a 3,164 dólares americanos aproximadamente.

Es necesario resaltar que, el valor actual neto obtenido de los flujos, es el excedente que se obtiene sobre la rentabilidad esperada de los accionistas. Es decir, es el plus que obtiene la empresa, además la utilidad exigida por la inversión.

El horizonte de análisis de solo dos años, ha sido suficiente para determinar que dentro de este periodo la inversión se recuperará. Sin embargo, es necesario notar que los beneficios por ahorro en el cálculo de las valorizaciones mensuales, costos de mantenimiento, costos de oportunidad, entre otros, son beneficios que se mantendrán más allá del horizonte de tiempo utilizado; por lo tanto, los beneficios se convierten en una perpetuidad que incrementa mucho más el valor actual neto de los beneficios. El flujo de la perpetuidad de beneficios no ha sido calculado puesto que solo basta con determinar que la inversión en tecnología se recupera dentro del periodo de tiempo exigido por los accionistas. Por lo tanto, el resultado de la estimación indica que el proyecto es rentable y debería ejecutarse.

La tasa de retorno mensual es de 0.73% y el tiempo de retorno es de 23 meses aproximadamente. La Tasa interna de retorno es de 3.45% mensual, la cual supera al Costo Promedio Ponderado de Capital (2.92%) confirmando que el proyecto es rentable. El resumen de estos resultados, pueden encontrarse en el Anexo N° 4.

Anexo 1.3.

Información relevante para el Flujo Económico de la reformulación del Proyecto de Reingeniería de los Sistemas de Información Minera.

Inversiones

Licencias de MS SQL Server	3,800	dólares
Licencias de MS SourceSafe	3,000	dólares
Equipo Servidor de Componentes	2,800	dólares
Equipo Servidor de Base de Datos	3,500	dólares
Ampliación de ancho de banda de la Wave LAN	5,000	dólares

Egresos Operativos

Pago mensual a analistas programadores (desarrollo)	12,000	soles
Pago mensual a analistas programadores (mantenimiento)	12,000	soles
Costo mensual de mantenimiento de quipos	4,500	soles

Parámetros de Análisis

Margen de error inicial en la valorización	0.06%	mensual
Valorización promedio mensual	8,000,000	mensual
Consumo de Materiales	2,150,000	
Diferencia en costos por valorización	4,800	
Diferencia en costos por consumo de materiales	1,290	
Tasa de ahorro en mantenimiento de sistema	30%	
Ahorro por mantenimiento de Sistemas	3,600	Pago por Horas-Hombre (analistas) utilizadas en tareas de mantenimiento de los sistemas.
Ahorro en costos de revisión de consistencia de datos	7,500	Mensual.
Utilidades promedio	8,500,000	Anual.
Tasa de Costo de Oportunidad	0.04%	
Tasa de Riesgo de Fuga de Información	3.00%	
Tasa de Depreciación de equipos informáticos	20.00%	anual
Costo de Oportunidad	283	Retraso en entrega de valorización. Imprecisión de resultados. Distribución contable no real, estados financieros no reales, toma de decisiones con alto nivel de incertidumbre.
Costo por fuga de Información	21,250	Garantía de disponibilidad permanente y oportuna de la información.

Tasa Impositiva	30%	
Costo explícito de deuda (Ki)	1.098%	real
Prima por Riesgo	5%	
Costo de Oportunidad del Accionista (Ke)	6.15%	
%Deuda	60%	
%Capital	40%	
Tipo de Cambio promedio (S/. / US$)	3.20	
TEA	14.00%	
TEM	1.098%	
Ahorro por diferencia en costos de valorización	4,800	
Pago mensual a analistas programadores (mantenimiento)	12,000	4 analistas de sistemas utilizando el 60% de su disponibilidad en tareas de mantenimiento.
Ahorro por mantenimiento de Sistemas	3,600	los analistas 40% menos tiempo en tareas de mantenimiento quedando tiempo disponible para otras tareas.

Anexo 2.

FLUJO ECONÓMICO (AÑO 1): Implementación del Modelo de Seguridad para el Sistemas Integrado de Información Minera

	AÑO 1												
	0	1	2	3	4	5	6	7	8	9	10	11	12
Flujo de Inversiones (S/.)													
Licencias de MS SQL Server	(12,160)												
Licencias de MS SourceSafe	(9,600)												
Equipo Servidor de Componentes	(8,960)												
Equipo Servidor de Base de Datos	(11,200)												
Ampliación de ancho de banda de la Wave LAN	(16,000)												
TOTAL FLUJO DE INVERSIONES	(57,920)	0	0	0	0	0	0	0	0	0	0	0	0
Flujo de Operaciones (S/.)													
Egresos													
Pago mensual a analistas programadores (desarrollo)		(12,000)	(12,000)	(12,000)	(12,000)	(12,000)	(12,000)	(12,000)	(12,000)	(12,000)	(12,000)	(12,000)	(12,000)
Pago mensual a analistas programadores (mantenimiento)													
Costo mensual de mantenimiento de quipos		(4,500)	(4,500)	(4,500)	(4,500)	(4,500)	(4,500)	(4,500)	(4,500)	(4,500)	(4,500)	(4,500)	(4,500)
Depreciación													(4,032)
Ingresos													
Por ahorro en Costos de Valorización													
Por ahorro en Costeo de consumo de materiales													
Por ahorro en mantenimiento de Sistemas													
Por ahorro en costos de revisión de consistencia de datos													
Por mejores decisiones (Información oportuna y precisa: Costo de Oportunidad)													
Por eliminación de fuga de información													
Utilidad antes de Impuestos	0	(16,500)	(16,500)	(16,500)	(16,500)	(16,500)	(16,500)	(16,500)	(16,500)	(16,500)	(16,500)	(16,500)	(20,532)
- Impuestos	0	(4,950)	(4,950)	(4,950)	(4,950)	(4,950)	(4,950)	(4,950)	(4,950)	(4,950)	(4,950)	(4,950)	(6,160)
Fondos generados	0	(11,550)	(11,550)	(11,550)	(11,550)	(11,550)	(11,550)	(11,550)	(11,550)	(11,550)	(11,550)	(11,550)	(14,372)
+ Depreciación	0	4,500	4,500	4,500	4,500	4,500	4,500	4,500	4,500	4,500	4,500	4,500	4,500
TOTAL FLUJO DE OPERACIONES	0	(7,050)	(7,050)	(7,050)	(7,050)	(7,050)	(7,050)	(7,050)	(7,050)	(7,050)	(7,050)	(7,050)	(9,872)
FLUJO DE CAJA ECONOMICO	(57,920)	(7,050)	(7,050)	(7,050)	(7,050)	(7,050)	(7,050)	(7,050)	(7,050)	(7,050)	(7,050)	(7,050)	(9,872)

Anexo 3.

FLUJO ECONÓMICO (AÑO 2): Implementación del Modelo de Seguridad para el Sistemas Integrado de Información Minera

	AÑO 2											
	1	2	3	4	5	6	7	8	9	10	11	12
Flujo de Inversiones (S/.)												
Licencias de MS SQL Server												
Licencias de MS SourceSafe												
Equipo Servidor de Componentes												
Equipo Servidor de Base de Datos												
Ampliación de ancho de banda de la Wave LAN												
TOTAL FLUJO DE INVERSIONES	0	0	0	0	0	0	0	0	0	0	0	0
Flujo de Operaciones (S/.)												
Egresos												
Pago mensual a analistas programadores (desarrollo)												
Pago mensual a analistas programadores (mantenimiento)	(12,000)	(12,000)	(12,000)	(12,000)	(12,000)	(12,000)	(12,000)	(12,000)	(12,000)	(12,000)	(12,000)	(12,000)
Costo mensual de mantenimiento de quipos	(4,500)	(4,500)	(4,500)	(4,500)	(4,500)	(4,500)	(4,500)	(4,500)	(4,500)	(4,500)	(4,500)	(4,500)
Depreciación												(4,032)
Ingresos												
Por ahorro en Costos de Valorización	4,800	4,800	4,800	4,800	4,800	4,800	4,800	4,800	4,800	4,800	4,800	4,800
Por ahorro en Costeo de consumo de materiales	1,290	1,290	1,290	1,290	1,290	1,290	1,290	1,290	1,290	1,290	1,290	1,290
Por ahorro en mantenimiento de Sistemas	3,600	3,600	3,600	3,600	3,600	3,600	3,600	3,600	3,600	3,600	3,600	3,600
Por ahorro en costos de revisión de consistencia de datos	7,500	7,500	7,500	7,500	7,500	7,500	7,500	7,500	7,500	7,500	7,500	7,500
Por mejores decisiones (Información oportuna y precisa: Costo de Oportunidad)	283	283	283	283	283	283	283	283	283	283	283	283
Por eliminación de fuga de información	21,250	21,250	21,250	21,250	21,250	21,250	21,250	21,250	21,250	21,250	21,250	21,250
Utilidad antes de Impuestos	22,223	22,223	22,223	22,223	22,223	22,223	22,223	22,223	22,223	22,223	22,223	18,191
- Impuestos	6,667	6,667	6,667	6,667	6,667	6,667	6,667	6,667	6,667	6,667	6,667	5,457
Fondos generados	15,556	15,556	15,556	15,556	15,556	15,556	15,556	15,556	15,556	15,556	15,556	12,734
+ Depreciación	4,500	4,500	4,500	4,500	4,500	4,500	4,500	4,500	4,500	4,500	4,500	4,500
TOTAL FLUJO DE OPERACIONES	20,056	20,056	20,056	20,056	20,056	20,056	20,056	20,056	20,056	20,056	20,056	17,234
FLUJO DE CAJA ECONOMICO	20,056	20,056	20,056	20,056	20,056	20,056	20,056	20,056	20,056	20,056	20,056	17,234

Anexo 4.

Tasa de Descuento del Flujo Económico

Método: Costo Promedio Ponderado del Capital (CPPC)

CPPC = %D x Ki x (1-T) + %C x Ke =	**2.92%**	mensual
VANE (S/.)	**10,123**	
VANE (US$)	**3,164**	

Rentabilidad del Proyecto

Tasa de Retorno Total (%)	**17.48%**	
Tasa de Retorno Mensual (%)	**0.73%**	
Retorno en meses	**23m / 3d**	
TIR (mensual)	**3.45%**	mensual