

Instituto Tecnológico de Costa Rica
Sede San Carlos

Escuela Ingeniería en Computación



NortheK Software S.A.



Módulo de Incentivos y Modulación de CMI

Informe Final del Proyecto de Graduación para optar por
el grado de Bachiller en Ingeniería en Computación

Maikol Porras Sánchez.

200214931
mporras@northeK.com

Ciudad Quesada Junio, 2006

Resumen Ejecutivo

Este resumen presenta los aspectos más importantes y los productos obtenidos en el desarrollo de la práctica de especialidad realizada en la empresa Northek Software con el fin de optar por el grado de Bachiller en Ingeniería en Computación en el Instituto Tecnológico de Costa Rica el primer semestre del año 2006.

Lo primero que presenta en el informe es el contexto del proyecto en el cual se presenta la información de la empresa y algunos de los antecedentes del proyecto los cuales son de vital importancia para la comprensión del mismo.

Después de analizar el contexto se dará la descripción del problema el cual consiste en dos partes, la primera de ellas es realizar un módulo para el cálculo de incentivos para el sistema CMISOFT el cual es uno de los principales productos de Northek Software. La segunda parte es modular el sistema CMISOFT ya que este fue desarrollado a la medida y se deseaba brindar a otros clientes potenciales la posibilidad de utilizarlo y por ende se requiere poder adaptarlo al máximo a las necesidades que estos presentan.

Luego se definirán aspectos de importancia como el análisis de riesgos, los objetivos y alcances del proyecto, los cuales ayudaran a comprender más en que consiste y lo que se pretende obtener del proyecto.

Mas adelante en el documento se mostrara la implementación de la solución, en la cual se detallara la forma en la que se desarrollo el proyecto, los pasos que se siguieron y el producto obtenido, esta sección es de gran interés ya que con ella se visualizara de alguna forma el producto final obtenido.

Y por ultimo se darán algunas conclusiones de en cuanto al desarrollo de la practica y algunos aspectos y vivencias que se tuvieron mientras se realizo.

Tabla de Contenido

| | |
|--|------------|
| RESUMEN EJECUTIVO | II |
| TABLA DE CONTENIDO | III |
| ÍNDICE DE FIGURAS | IV |
| CONTEXTO DEL PROYECTO | 1 |
| INFORMACIÓN DE LA EMPRESA | 1 |
| ANTECEDENTES DEL PROYECTO | 2 |
| DESCRIPCIÓN DEL PROBLEMA | 5 |
| DESCRIPCIÓN DE LA SOLUCIÓN | 6 |
| STACKEOLDERS | 8 |
| NECESIDADES Y EXPECTATIVAS | 8 |
| PERSPECTIVAS, SUPUESTOS Y DEPENDENCIAS | 9 |
| REQUERIMIENTOS NO FUNCIONALES | 10 |
| CARACTERÍSTICAS GENERALES | 10 |
| DESCRIPCIÓN DE RIESGOS | 11 |
| OBJETIVOS DEL SISTEMA | 13 |
| OBJETIVO GENERAL: | 13 |
| OBJETIVOS ESPECÍFICOS: | 13 |
| ALCANCES DEL SISTEMA | 14 |
| IMPLEMENTACIÓN DE LA SOLUCIÓN | 16 |
| DISEÑO DE CLASES | 16 |
| <i>Diagrama de clases</i> | 16 |
| <i>Clase Incentive:</i> | 17 |
| Descripción: | 17 |
| Atributos: | 17 |
| Constructores: | 18 |
| Descripción de Métodos: | 19 |
| <i>Clase IncentiveEmployee:</i> | 24 |
| Descripción: | 24 |
| Atributos: | 24 |
| Constructores: | 25 |
| Descripción de Métodos: | 26 |
| <i>Clase IncentiveDB</i> | 30 |
| Descripción: | 30 |
| Constructores: | 30 |
| Descripción de Métodos: | 31 |
| INTERFACES DE USUARIO | 36 |
| DISEÑO DE BASE DE DATOS | 39 |
| RESEÑA SOBRE CMISOFT | 40 |
| MODULARIZACIÓN CMISOFT | 43 |
| RECOMENDACIONES | 53 |
| CONCLUSIONES Y COMENTARIOS | 54 |
| ANEXO #1 | 56 |

Índice de Figuras

| | |
|---|----|
| FIGURA. 1. ORGANIGRAMA DE NORTHEK ----- | 2 |
| FIGURA. 2. ARQUITECTURA CMISOFT ----- | 3 |
| FIGURA. 3. DIAGRAMA DE CLASES ----- | 16 |
| FIGURA. 4. CLASE INCENTIVE ----- | 17 |
| FIGURA. 5. CLASE INCENTIVEEMPLOYEE ----- | 24 |
| FIGURA. 6. CLASE INCENTIVEDB ----- | 30 |
| FIGURA. 7. INTERFAZ DEL INDEX DE INCENTIVOS ----- | 36 |
| FIGURA. 8. INTERFAZ VER INFORMACIÓN DE UN INCENTIVO ----- | 36 |
| FIGURA. 9. INTERFAZ DE CREAR UN NUEVO INCENTIVO ----- | 37 |
| FIGURA. 10. INTERFAZ DEL ETITAR UN INCENTIVO ----- | 37 |
| FIGURA. 11. INTERFAZ DEL CALCULAR Y ASIGNAR UN INCENTIVO ----- | 38 |
| FIGURA. 12. DISEÑO DE LA BASE DE DATOS ----- | 39 |
| FIGURA. 13. MODELO CONCEPTUAL ACTUAL CMISOFT ----- | 43 |
| FIGURA. 14. INTERFAZ CREAR INDICADOR ----- | 44 |
| FIGURA. 15. INTERFAZ INSERTAR EMPLEADO ----- | 44 |
| FIGURA. 16. INTERFAZ SECCIÓN GENERAL DE CMISOFT ----- | 45 |
| FIGURA. 17. INTERFAZ ADMINISTRACIÓN ----- | 46 |
| FIGURA. 18. INTERFAZ DE TABLEROS DE CONTROL ----- | 46 |
| FIGURA. 19. INTERFAZ SECCIÓN GENERAL CMISOFT ----- | 46 |
| FIGURA. 20. INTERFAZ SECCIÓN ESTRUCTURA ORGANIZACIONAL ----- | 47 |
| FIGURA. 21. INTERFAZ GENERACIÓN DE REPORTES ----- | 48 |
| FIGURA. 22. MODULO CONCEPTUAL CMISOFT MODULARIZADO ----- | 49 |
| FIGURA. 23. DIAGRAMA DE DEPENDENCIAS CMISOFT MODULARIZADO ----- | 49 |
| FIGURA. 24. INTERFAZ CMIVG ----- | 52 |
| FIGURA. 25. DISEÑO DE LA BD MODULARIZADO ----- | 56 |

Contexto del Proyecto

Información de la Empresa

Northek Software es una empresa nacional que se dedica al desarrollo de software en múltiples plataformas, y con un enfoque hacia el software empresarial.

Se encuentra ubicada en la zona norte del país y cuenta con más de tres años de experiencia en la implementación de soluciones integrales de Software, que van desde aplicaciones de escritorio hasta sistemas Web.

La Misión de Northek Software S.A. es brindar soluciones integrales en el área de tecnologías de información y comunicación, prioritariamente a las organizaciones de la Región Huetar Norte, garantizando calidad y un excelente servicio al cliente, fundamentados en la innovación tecnológica constante y el aprovechamiento del recurso humano local.

Entre sus principales productos están:

Northek Note: un sistema completo de administración de documentos y clientes para Notarios Públicos.

Northek Help Desk: es una solución integral que se adapta a las necesidades de su empresa y que brinda una administración y organización de casos de forma ordenada, rápida y eficiente.

CMISOFT: sobre este producto se basa la práctica de especialidad, es un sistema de cuadro de mando integral, que permite el monitoreo constante sobre la actividad en las empresas.

Northek es una empresa pequeña que tiene la siguiente estructura organizacional (ver Figura.1):

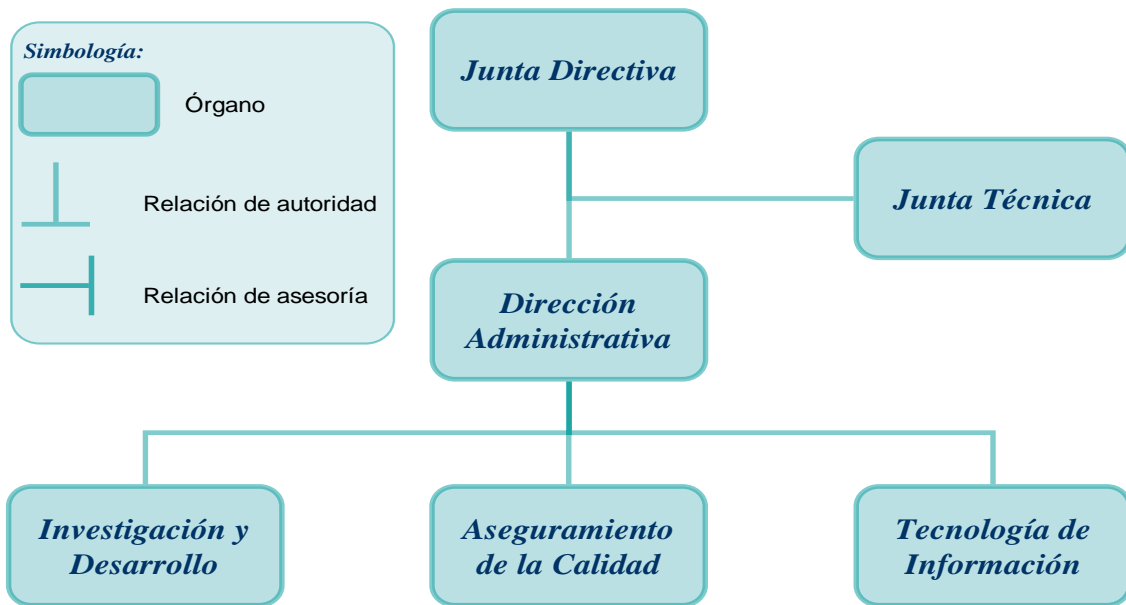


FIGURA. 1. ORGANIGRAMA DE NORTHEK

Antecedentes del proyecto

El proyecto CMISOFT, fue realizado por la empresa Northek Software S.A., con el fin de brindar un programa de Cuadro de Mando Integral a una de las empresas más importantes de la zona norte.

El sistema CMISOFT le permite a la empresa manejar muchos aspectos de la organización estructural y de la estrategia ya que posibilita el uso de indicadores para lograr un monitoreo integral de la empresa.

En si CMISOFT es un sistema muy amplio, y posee entre otras las siguientes características:

- Permite la administración de indicadores que son la base de todo el sistema. Tienen una métrica definida y distintos atributos que lo relacionan con un elemento de la estructura de la empresa.
- Posibilita el uso de tableros de control, los cuales están asociados con indicadores que permitir un monitoreo de cada departamento o unidad de la empresa.

- Maneja aspectos a nivel estratégico como lo son planes de acción, factores críticos de éxito, áreas y metas estratégicas, lo cual ayuda a la toma de decisiones de la alta jerarquía de la empresa.
- Por otro lado también maneja la información de la estructura organizacional de la empresa, entre ellas unidades, empleados, facultades y periodos.
- Brinda también la posibilidad realizar varios tipos de evaluaciones a los empleados, y crear encuestas para realizar dichas evaluaciones.
- Además da la posibilidad de realizar diferentes tipos de reportes.

La descripción de este proyecto es muy amplia, además de estos detalles, existen más aspectos que realiza CMISOFT, y que se dan conforme interactúan los elementos citados anteriormente.

El sistema esta hecho sobre la plataforma J2EE y posee una arquitectura de tres capas como se muestra en la Figura.2:

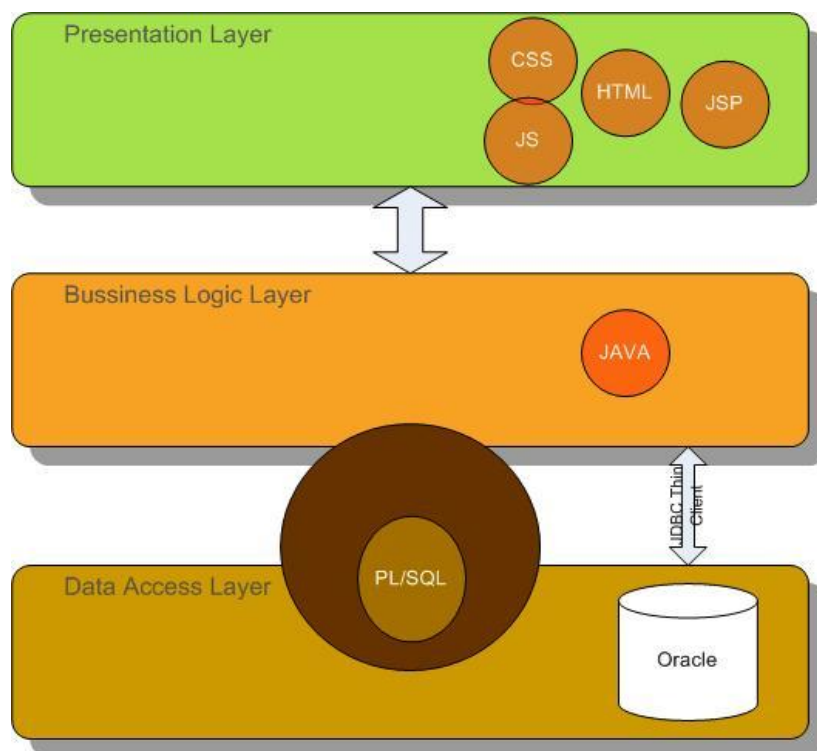


FIGURA. 2. ARQUITECTURA CMISOFT

Data Access Layer (DAL): Esta capa la componen el Gestor de Base de Datos y los objetos encargados de proveer una conexión segura, estable y eficiente con el gestor así como las acciones o procedimientos de acceso propios de cada objeto del sistema. Estas clases fueron programadas en Java y en su mayoría cada procedimiento responde a un procedimiento de PL/SQL definido en la capa intermedia entre el DAL y el BLL. Estos procedimientos se encargan de manipular las acciones sobre el gestor de base de datos, así como la captura de excepciones y verificaciones necesarias en lo que a manipulación de datos se refiere; otros se encargan de ejecutar procesos de mucha carga sobre el gestor. Estos procedimientos forman parte integra de la base de datos de CMISOFT.

Bussines Logic Layer (BLL): Todos los objetos de esta capa son los encargados de dar vida a la aplicación implementando las tareas y procedimientos necesarios para la correcta interpretación y presentación de los datos. Todos estos objetos fueron programados en Java.

Presentation Layer (PL): Esta ultima capa esta relacionada directamente con lo que el usuario interactúa. Dada algunas limitaciones que poseen las aplicaciones Web se dio a la tarea de implementar algunas herramientas y técnicas en esta capa para dar al usuario la sensación de una aplicación de escritorio. Esta capa esta desarrollada completamente en HTML, utilizando scriptlets de JSP, algunos componentes y procedimientos en Javascript y CSS para dar a la aplicación un look and feel estandarizado y profesional.

En cada capa se intento utilizar las últimas tendencias y estándares de programación existentes a nivel mundial.

Por último, los lenguajes que se utilizaron para el desarrollo fueron Java Server Pages para la capa de presentación, Java para la lógica y PL/SQL para la de base de datos.

Descripción del Problema

El proyecto consiste en desarrollar un módulo de incentivos para CMISOFT, con el cual se pretende simplificar el proceso de selección de los empleados que optan por un incentivo anual definido por la empresa así como el cálculo del monto del incentivo correspondiente a cada uno de ellos.

Este módulo de incentivos se dividirá en dos secciones:

- Administración de los incentivos: esto significa poder ingresar, editar y borrar incentivos, entre los datos que debe contener cada incentivo están: un nombre, una descripción, un conjunto de reglas (las cuales limitan quien lo recibe) y el periodo para el cual aplica el incentivo.
- Seleccionar los empleados que cumplan con las reglas y dar la opción al usuario de calcular el monto a otorgar a cada empleado siguiendo una formula específica. A todo esto se suma la generación de reportes, para saber que incentivos fueron otorgados, a quien fueron otorgados y el monto que se le dio a cada empleado.

Por otra parte el objetivo final del software CMISOFT, aunque sus inicios hayan sido como un software a la medida, es funcionar como un producto producible en masa y modularizado completamente, permitiendo la inclusión o exclusión de módulos funcionales. Además el sistema CMISOFT debe contar con un crecimiento y adaptación eficientes a nuevos mercados y clientes, permitiendo su puesta en marcha con cualquier combinación del total de módulos con los que cuente el sistema. Este esquema permitirá entonces la posibilidad de desarrollos paralelos de componentes y módulos que puedan ser integrados a clientes que cuenten con la aplicación y a nuevos pedidos que denominaremos productos a la medida.

Según lo anterior es necesario realizar la modulación de CMI, esta modulación se da con el fin de poder brindar soluciones adaptables para diferentes empresas. En este momento se tienen presentes tres tipos de empresas: pequeñas, medianas y grandes.

Cuando se termine la modulación, se pretende desarrollar un sitio Web para la demostración del producto, esto para brindarle a las empresas un lugar en el cual puedan probar el sistema y entender los beneficios del mismo.

El objetivo del proyecto como se mencionó anteriormente es lograr que el sistema CMISOFT se pueda utilizar en componentes, para ello se pretende realizar un análisis de el comportamiento del sistema y de esta manera poder conocer cual seria la forma más fácil de dividir CMISOFT y en cuantas módulos. Este análisis abarca el funcionamiento lógico, la base de datos y el manejo de interfaz.

Descripción de la Solución

Este proyecto se divide en varias partes, todas diferentes, pero con un mismo objetivo expandir el sistema CMISOFT, debido a ello se planteado una forma en la cual se deberá desarrollar el proyecto para aprovechar al máximo el tiempo, facilitar y maximizar el esfuerzo de los involucrados.

Para la primera parte del proyecto que es el módulo de incentivos, se pretende buscar la forma en la que se pueda desarrollar evitando las dependencias en el sistema, ya que se sabe que se tiene que modular y lo que se busca es no duplicar el trabajo. Por lo tanto se debe trabajar con el cuidado de no descentralizar la función del módulo para después no tener que unirla. Además se debe trabajar según la arquitectura en la que esta desarrollado el proyecto, esta es de tres capas y sobre la plataforma J2EE. La cual se explicó anteriormente.

El módulo de incentivos es lógicamente muy simple, por lo cual mientras se tenga el correcto enfoque, no se presentarán problemas importantes con el desarrollo del mismo.

Con respecto a la parte técnica lo que se pretende es diseñar primero las tablas de la base de datos y buscar la relación con otras partes del sistema, una vez terminado se continuará con el diseño de la interfaz de usuario, esto para tener todo listo para trabajar con la lógica del sistema. De esta manera es que se pretende desarrollar el módulo.

Por para realizar la modularización del CMISOFT, lo que se pretende primero es realizar una análisis del sistema actual, para poder detectar la dependencias del sistema

y de esta manera poder elaborar un planteamiento para la separación de módulos y funciones. Esta separación abarca todo el sistema, base de datos, paquetes de base de datos, clases e interfaz.

Una vez lista la propuesta de modularización se continuara con independizar los módulos, reubicando funciones y creando otras (en caso de ser necesario).

Para finalizar la parte de la modularización de CMISOFT se buscara la manera de crear algún tipo de generador de versiones, esto para facilitar el trabajo a la hora de hacer el sistema a la medida de acuerdo a los requerimientos del cliente.

Y la ultima parte que se hará es el sitio Web en el cual se brindara información del sistema y se dará la opción de la probar el sistema en línea. Para llevar esto a cabo primero se plantearan las secciones que deberá contener el sitio para luego llevar a cabo el desarrollo.

Stakeholders

| Nombre Completo | Unidad | Responsabilidades |
|--------------------|--------|--|
| Maikol Porras | R&D | <ul style="list-style-type: none"> - Desarrollo y Diseño del módulo de Incentivos. - Desarrollo y Diseño sitio Web. - Investigar acerca de la forma más idónea de modular el sistema. - Llevar a cabo la modularización de CMISOFT. |
| Juan Daniel Alfaro | R&D | <ul style="list-style-type: none"> - Colaborar en la resolución de dudas en cuanto al comportamiento del sistema y problemas técnicos. - Participar en el desarrollo (en caso de que surja algún problema importante). |
| Mario Rodríguez | QA | - Funciones básicas de Aseguramiento de la calidad |
| Alejandro Alfaro | QA | - Funciones básicas de Aseguramiento de la calidad |

Necesidades y Expectativas

El proyecto surge de acuerdo a una necesidad la cual es la creación del módulo de incentivos y una oportunidad que es poder brindar el sistema CMISOFT a múltiples compañías que necesiten un sistema de cuadro de mando integral.

La necesidad del módulo de incentivos se da al tener que satisfacer las necesidades del cliente, ya que los incentivos van ligados al rendimiento del empleado y el sistema controla este rendimiento por medio de evaluaciones, en este aspecto fue donde el cliente vio la gran oportunidad de implementar un componente en el cual pueda agilizar y de cierto modo automatizar la entrega de los incentivos a sus empleados.

En cuanto a la oportunidad se refleja en el interés de ciertas empresas en el sistema, teniendo la oportunidad de promover el uso del sistema se decidió modularlo con el fin de poder hacer el sistema flexible y poder adaptarlo a las necesidades del cliente y poder brindarle exactamente lo que necesita y lo que le sirve a su empresa de acuerdo a su tamaño.

Las expectativas que se tienen son muchas, primero esperar que el módulo funcione correctamente, lograr la satisfacción del cliente y facilitar y agilizar la labor de asignación de incentivos.

Y segundo que es la mas importante, lograr manejar un sistema que permita la adaptación a cualquier tipo de empresa brindándole componentes funcionales y que cumplan con los requerimientos del cliente sin brindarle mas o menos de lo que realmente necesita.

Perspectivas, supuestos y dependencias

Perspectivas

- Aumentar la funcionalidad del sistema CMISOFT.
- Proveer al sistema de un módulo para el cálculo de incentivos basado en evaluaciones.
- Brindar un sistema adaptable a cualquier tipo de empresa. Y a la vez ofrecer al cliente la oportunidad de elegir cuales componentes desea para una versión específica del sistema.
- Dar mayor proyección a la aplicación con el fin de encontrar nuevos clientes interesados en el producto.

Supuestos

- Buen funcionamiento de la sección de evaluaciones del sistema (módulo incentivos).
- Apoyo técnico sobre el funcionamiento de la aplicación (modularización).
- Herramientas necesarias para el desarrollo del proyecto.
- Información y requisitos sobre el cálculo de los incentivos (módulo incentivos).

Dependencias

- El departamento de desarrollo deberá brindar una capacitación sobre el uso y funcionamiento del sistema CMISOFT.

Requerimientos no funcionales

- Facilitar el cálculo de incentivos en la empresa, tratando de automatizar al máximo el procedimiento de cálculo y asignación de incentivos.
- Hacer que el sistema sea flexible y adaptable a cualquier tipo de versión que se desee crear, esto para dar un producto a la medida.
- Dar a conocer el sistema CMISOFT mostrando su capacidad y los aportes que un sistema de este tipo puede ofrecer a una empresa.
- El módulo deberá proveer un sistema de control de usuario con el que se pueda verificar si la persona tiene acceso o no a alguna de las funciones del módulo.
- El módulo brindará un manejo de excepciones y de mensajes completo e integrado con el manejador de Bases de Datos. Presenta mensajes de fácil comprensión tanto para el usuario experto como el de poco conocimiento técnico.
- Deberá existir un empleado administrador del sistema con todos los permisos.

Características Generales

Llevar el control de los incentivos de las compañías así como la asignación de los mismos y de esta manera poder realizar reportes sobre los incentivos otorgados.

Facilidad a la hora de hacer una versión del sistema CMISOFT de acuerdo a los deseos del cliente y las características de la empresa, además agilizar el mantenimiento del sistema ya que no se tendrán versiones por cada empresa que lo quiera sino que se manejará como un conjunto de componentes.

Presentar y promocionar el sistema CMISOFT por medio de un sitio Web en el cual se manejará información de la Northek y del sistema, y permitirá la utilización del mismo en línea.

Descripción de Riesgos

Estos son los riesgos asociados con el módulo de incentivos:

Nombre: Problemas con el manejo de nuevas herramientas.

Categoría: Tecnológicos.

Causa: Poco conocimiento de las herramientas a utilizar.

Impacto: Alto.

Probabilidad de Ocurrencia: 40%

Exposición: 40%

Estrategia de Evasión: Estudio de las herramientas a utilizar antes de comenzar el desarrollo del proyecto.

Estrategia de Contingencia: Reunirse con una persona que tenga conocimientos en las herramientas a utilizar y pedir asesoría a las mismas.

Nombre: Falta de información para calcular incentivos.

Categoría: Personas.

Causa: Falta de comunicación con el cliente para que explique la manera de calcular el incentivo.

Impacto: Alto.

Probabilidad de Ocurrencia: 15%

Exposición: 15%

Estrategia de Evasión: Consultar con la persona indicada antes de realizar el trabajo sobre la fórmula.

Estrategia de Contingencia: la misma de evasión ya que es la única opción viable.

Nombre: Problemas de entendimiento con el cliente.

Categoría: Personas.

Causa: Poca comunicación con el cliente.

Impacto: Alto.

Probabilidad de Ocurrencia: 20%

Exposición: 20%

Estrategia de Evasión: Estar en contacto continuo con el cliente.

Estrategia de Contingencia: Buscar la manera de reunirse rápidamente con el cliente para definir dudas y hacer sugerencias.

Los riesgos asociados con la modularización de CMISOFT son:

Nombre: No entender el funcionamiento del sistema.

Categoría: Tecnológicos.

Causa: Poco conocimiento del sistema a modularizar.

Impacto: Alto.

Probabilidad de Ocurrencia: 30%

Exposición: 30%

Estrategia de Evasión: Investigar en que consiste y como funciona el sistema.

Estrategia de Contingencia: Reunirse con una persona que tenga conocimientos del sistema y pedir asesoría a las mismas.

Nombre: Que partes del sistema que no se puedan independizar.

Categoría: Tecnológicos.

Causa: Interrelaciones que no pueden cambiarse entre varias partes del sistema.

Impacto: Alto.

Probabilidad de Ocurrencia: 20%

Exposición: 20%

Estrategia de Evasión: Un buen estudio sobre los módulos que debería tener el sistema.

Estrategia de Contingencia: Tratar de componer los componentes que tienen estas dependencias en un mismo módulo.

Nombre: Atraso en el cronograma.

Categoría: Tecnológicos y Personas.

Causa: Que se presente alguno de los problemas que se mencionaron anteriormente.

Impacto: Alto.

Probabilidad de Ocurrencia: 20%

Exposición: 20%

Estrategia de Evasión: Llevar un control adecuado del tiempo y del avance del proyecto.

Estrategia de Contingencia: Buscar apoyo ya sea técnico o informativo con alguna persona capacitada para resolver problemas y llevar al día el cronograma.

Objetivos del Sistema

Objetivo General:

Expandir el sistema CMISOFT modularizando el sistema para poder desarrollar diferentes versiones del sistema sin la necesidad de realizar cambios en el mismo y agregando una nueva característica como lo es el módulo de incentivos.

Objetivos Específicos:

1. Desarrollar un módulo que permita una fácil administración y una rápida asignación de los incentivos que otorga la compañía.
2. Modular el sistema CMISOFT para que este pueda implantarse en distintas empresas y que se adapte a las necesidades de estas.
3. Un sitio Web en el cual se pueda mostrar CMISOFT, para explicar su funcionamiento y su capacidad, en el cual también se brindara la posibilidad de que las empresas prueben el sistema.

Alcances del Sistema

1. El módulo de incentivos debe contar con las siguientes características:
 - 1.1. Permitir al usuario el ingreso, la edición y el borrado de incentivos en el sistema las opciones de edición e ingreso contarán con los campos de nombre, descripción, fechas de inicio y fin. También contará con campos para definir las reglas que consisten en un periodo de tiempo y una nota, por último tendrán varias checks para activar o desactivar el incentivo así como varios para elegir las evaluaciones que se utilizarán para calcular el incentivo.
 - 1.2. Mostrar al usuario, los empleados que cumplen con los requisitos para recibir el incentivo en una lista, de tal manera que pueda elegir alguno y poder calcular el incentivo.
 - 1.3. Permitir que el usuario pueda calcular el incentivo para cada empleado y asignarle el monto calculado, esto mediante una serie de campos que representan una fórmula, algunos de los cuales los carga el sistema cuando se elige al usuario y otros que el encargado completará.
 - 1.4. Mostrar al usuario en pantalla la lista de empleados a los que se les ha asignado el incentivo y el monto que se les otorga.
 - 1.5. Permitir que el usuario puede revocar un incentivo asignado, de la lista de empleados con incentivo asignado.
 - 1.6. Generar reportes de los incentivos que contengan el incentivo y los empleados que fueron beneficiados con el así como el monto otorgado.

2. La Modulación de CMISOFT, debe permitir la implantación del sistema en diferentes empresas de acuerdo a las necesidades de las mismas. Se pretende crear una serie de repositorios de los cuales poder generar una versión del sistema ya sea de manera automática o manual.

3. Sitio Web debe poseer las siguientes características:
 - 3.1. Sección de Información Dinámica: en la cual se dará a conocer características de la empresa y del sistema.
 - 3.2. Sección informativa: en la cual se brindará información técnica del sistema.
 - 3.3. Sección contactos
 - 3.4. Sección casos de éxito

3.5. Sección para empresas: en esta sección se mostrara que módulos deberá tener cada empresa dependiendo de su tamaño y la labor que realice.

3.5.1. CMISOFT Básico – Pequeña Empresa

3.5.2. CMISOFT Profesional - Mediana Empresa

3.5.3. CMISOFT Empresarial – Grande Empresa

3.6. Sección prueba en línea: en esta sección se dará la posibilidad a los visitantes de la pagina poder utilizar el sistema.

Implementación de la Solución

Diseño de Clases

Diagrama de clases

En la siguiente imagen (Figura.3) podemos observar el diagrama de clases para el módulo de incentivos de la aplicación CMISOFT.

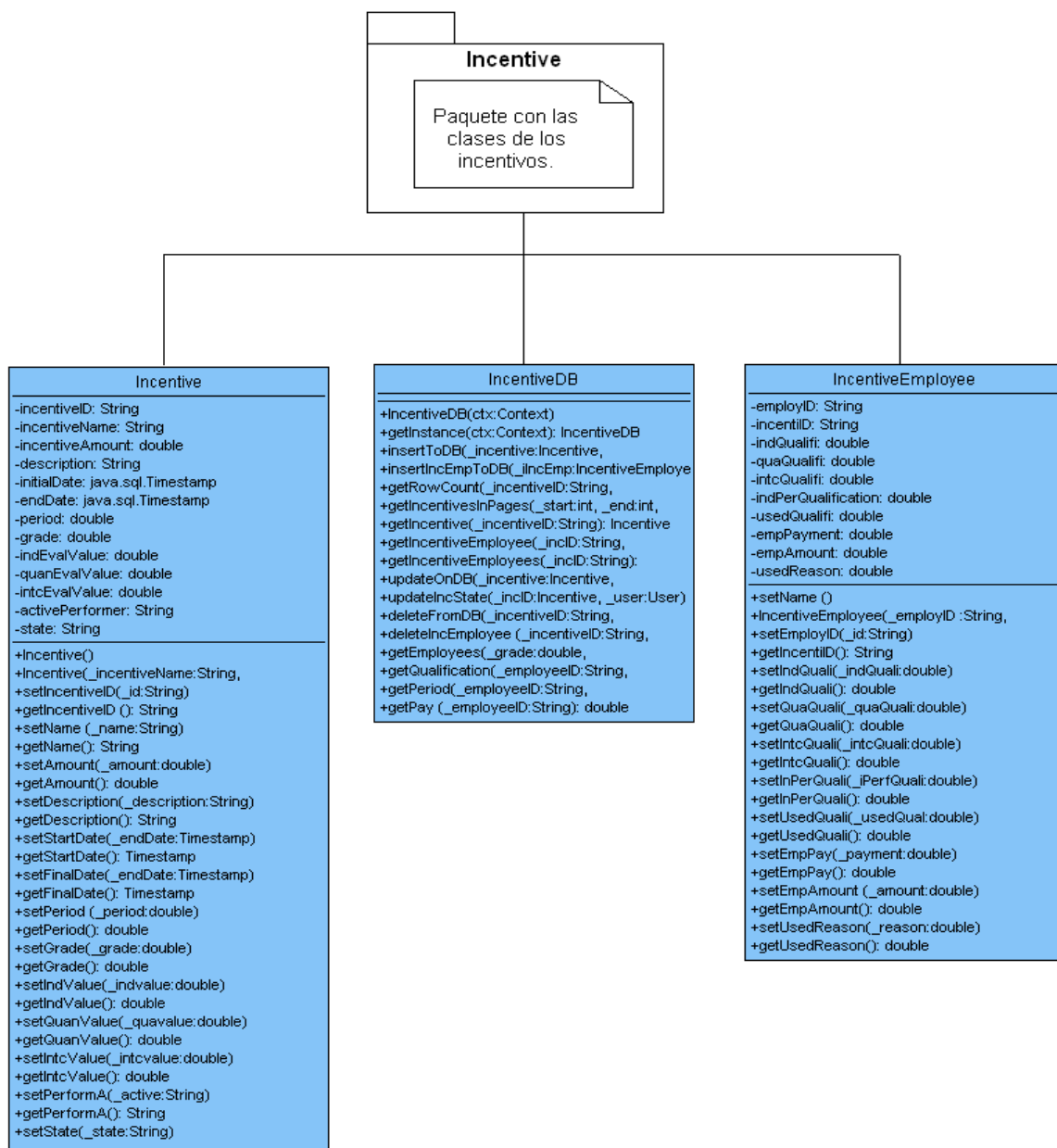


FIGURA. 3. DIAGRAMA DE CLASES

Class Incentive:

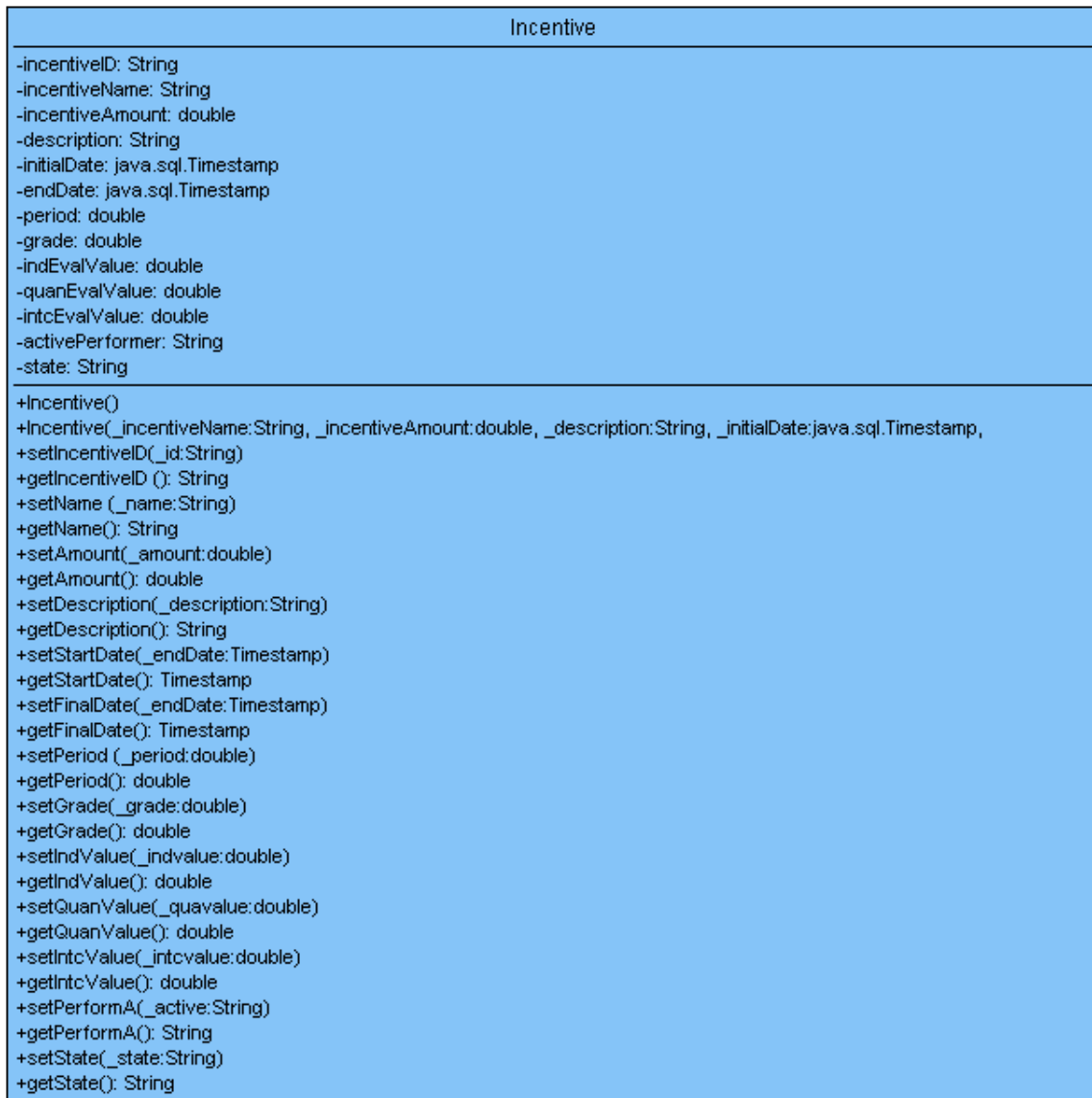


FIGURA. 4. CLASE INCENTIVE

Descripción:

Esta clase (Figura.4) se encarga de crear el objeto que representa un incentivo del módulo de incentivos.

Atributos:

- incentiveID: Es el identificador único del incentivo, el cual se utilizara para identificarlo en la BD.

- incentiveName: Nombre asignado al incentivo.
- incentiveAmount: Presupuesto asignado al incentivo.
- Description: Breve descripción del incentivo.
- initialDate: Fecha de inicio en al cual se tomara el consideración el incentivo.
- endDate: Fecha de fin en al cual se tomara el consideración el incentivo.
- Period: Periodo de tiempo laborado requerido para que un empleado pueda optar por el incentivo.
- Grade: Nota mínima requerida en las evaluaciones, para que un empleado sea elegible para recibir el incentivo.
- indEvalValue: Valor porcentual asignado a las evaluaciones individuales.
- quanEvalValue: Valor porcentual asignado a las evaluaciones cuantitativas
- intcEvalValue: Valor porcentual asignado a las evaluaciones de cliente interno
- activePerformer: Indica si se utilizara la evaluación de desempeño.
- State: Estado del incentivo (si ya fue otorgado o no).

Constructores:

| | |
|-------------------------------------|--|
| Nombre del Constructor: | <code>public Incentive()</code> |
| Descripción del Constructor: | Constructor nulo de la clase incentive. Crea una instancia del objeto incentive con valores por defecto o nulos. |

| | |
|-------------------------------------|--|
| Nombre del Constructor: | <pre>public Incentive(String _incentiveName, double _incentiveAmount, String _description, java.sql.Timestamp _initialDate, java.sql.Timestamp _endDate, double _period, double grade, double _indEvalValue, double _quanEvalValue, double _intcEvalValue , String _activePerfor, String _state)</pre> |
| Descripción del Constructor: | Crea una instancia del objeto Incentive con los valores enviados por parámetro. |
| Parámetro: | Descripción |
| <code>_incentiveName</code> | Nombre de incentivo |
| <code>_incentiveAmount</code> | Monto del incentivo |
| <code>_description</code> | Descripción del incentivo |
| <code>_initialDate</code> | Fecha inicio del periodo del incentivo |

| | |
|----------------|--|
| _endDate | Fecha fin del periodo del incentivo |
| _period | Periodo requerido para recibir el incentivo |
| _grade | Nota mínima requerida en la(s) prueba(s) |
| _indEvalValue | Valor porcentual asignado a la evaluación individual |
| _quanEvalValue | Valor porcentual asignado a la evaluación cuantitativa |
| _intcEvalValue | Valor porcentual asignado a la evaluación de cliente interno |
| _activePerfor | Indica si se utilizara la nota de desempeño |
| _state | Indica si se puede asignar el incentivo |

Descripción de Métodos:

| | |
|--------------------------------|---|
| Nombre del Método: | <i>public void setIncentiveID(String _id)</i> |
| Descripción del Método: | Asigna el código de identificación del incentivo. |
| Parámetro: | Descripción |
| _id | Código de identificación del incentivo. |

| | |
|--------------------------------|--|
| Nombre del Método: | <i>public String getIncentiveID()</i> |
| Descripción del Método: | Obtiene el código de identificación del incentivo. |
| Returns: | Código de identificación del incentivo. |

| | |
|--------------------------------|--|
| Nombre del Método: | <i>public void setName(String _name)</i> |
| Descripción del Método: | Asigna el nombre del incentivo. |
| Parámetro: | Descripción |
| _name | Nombre del incentivo. |

| | |
|--------------------------------|----------------------------------|
| Nombre del Método: | <i>public String getName()</i> |
| Descripción del Método: | Obtiene el nombre del incentivo. |
| Returns: | Nombre del incentivo. |

| | |
|--------------------------------|--|
| Nombre del Método: | <i>public void setAmount(double _amount)</i> |
| Descripción del Método: | Asigna el monto del incentivo. |
| Parámetro: | Descripción |
| _amount | Monto del incentivo. |

| | |
|--------------------------------|----------------------------------|
| Nombre del Método: | <i>public double getAmount()</i> |
| Descripción del Método: | Obtiene el monto del incentivo. |
| Returns: | Monto del incentivo. |

| | |
|--------------------------------|--|
| Nombre del Método: | <i>public void setDescription(String _description)</i> |
| Descripción del Método: | Asigna una descripción del incentivo. |
| Parámetro: | Descripción |
| _description | Descripción del incentivo. |

| | |
|--------------------------------|---------------------------------------|
| Nombre del Método: | <i>public String getDescription()</i> |
| Descripción del Método: | Obtiene la descripción del incentivo. |
| Returns: | Descripción del incentivo. |

| | |
|--------------------------------|---|
| Nombre del Método: | <i>public void setStartDate(Timestamp _initDate)</i> |
| Descripción del Método: | Asigna la fecha de inicio para la aplicación del incentivo. |
| Parámetro: | Descripción |
| _initDate | Fecha de inicio. |

| | |
|--------------------------------|--|
| Nombre del Método: | <i>public Timestamp getStartDate(){</i> |
| Descripción del Método: | Obtiene la fecha de inicio para la aplicación del incentivo. |
| Returns: | Fecha de inicio. |

| | |
|--------------------------------|---|
| Nombre del Método: | <i>public void setFinalDate(Timestamp _endDate)</i> |
| Descripción del Método: | Asigna la fecha fin para la aplicación del incentivo. |
| Parámetro: | Descripción |
| _endDate | Fecha Final. |

| | |
|--------------------------------|--|
| Nombre del Método: | <i>public Timestamp getFinalDate()</i> |
| Descripción del Método: | Obtiene la fecha fin para la aplicación del incentivo. |
| Returns: | Fecha Final. |

| | |
|--------------------------------|--|
| Nombre del Método: | <i>public void setPeriod(double _period){}</i> |
| Descripción del Método: | Asigna el periodo requerido para obtener el incentivo. |
| Parámetro: | Descripción |
| _period | Periodo requerido. |

| | |
|--------------------------------|---|
| Nombre del Método: | <i>public double getPeriod()</i> |
| Descripción del Método: | Obtiene el periodo requerido para obtener el incentivo. |
| Returns: | Periodo requerido. |

| | |
|--------------------------------|--|
| Nombre del Método: | <i>public void setGrade(double _grade)</i> |
| Descripción del Método: | Asigna la nota mínima para obtener el incentivo. |
| Parámetro: | Descripción |
| _grade | Nota mínima. |

| | |
|--------------------------------|---|
| Nombre del Método: | <i>public double getGrade()</i> |
| Descripción del Método: | Obtiene la nota mínima para obtener el incentivo. |
| Returns: | Nota mínima. |

| | |
|--------------------------------|---|
| Nombre del Método: | <i>public void setIndValue(double _indValue)</i> |
| Descripción del Método: | Asigna el valor representativo de la evaluación individual. |
| Parámetro: | Descripción |
| _indValue | Valor representativo de la evaluación individual. |

| | |
|--------------------------------|--|
| Nombre del Método: | <i>public double getIndValue()</i> |
| Descripción del Método: | Obtiene el valor representativo de la evaluación individual. |
| Returns: | Valor representativo de la evaluación individual. |

| | |
|--------------------------------|---|
| Nombre del Método: | <i>public void setQuanValue(double _quaValue){</i> |
| Descripción del Método: | Asigna el valor representativo de la evaluación cuantitativa. |
| Parámetro: | Descripción |
| _quaValue | Valor representativo de la evaluación cuantitativa. |

| | |
|--------------------------------|--|
| Nombre del Método: | <i>public double getQuanValue()</i> |
| Descripción del Método: | Obtiene el valor representativo de la evaluación cuantitativa. |
| Returns: | Valor representativo de la evaluación cuantitativa. |

| | |
|--------------------------------|---|
| Nombre del Método: | <i>public void setIntcValue(double _intcValue)</i> |
| Descripción del Método: | Asigna el valor representativo de la evaluación de cliente interno. |
| Parámetro: | Descripción |
| _intcValue | Valor representativo de la evaluación de cliente interno. |

| | |
|--------------------------------|--|
| Nombre del Método: | <i>public double getIntcValue()</i> |
| Descripción del Método: | Obtiene el valor representativo de la evaluación de cliente interno. |
| Returns: | Valor representativo de la evaluación de cliente interno. |

| | |
|--------------------------------|---|
| Nombre del Método: | <i>public void setPerformA(String _active)</i> |
| Descripción del Método: | Asigna el valor para determinar si se usara la evaluación de desempeño. |
| Parámetro: | Descripción |
| _active | Valor para determinar si se usara la evaluación de desempeño. |

| | |
|--------------------------------|--|
| Nombre del Método: | <i>public String getPerformA()</i> |
| Descripción del Método: | Obtiene el valor para determinar si se usara la evaluación de desempeño. |
| Returns: | Valor para determinar si se usara la evaluación de desempeño. |

| | |
|--------------------------------|--|
| Nombre del Método: | <i>public void setState(String _state)</i> |
| Descripción del Método: | Asigna el estado del incentivo. |
| Parámetro: | Descripción |
| _state | Estado del incentivo. |

| | |
|--------------------------------|----------------------------------|
| Nombre del Método: | <i>public String getState()</i> |
| Descripción del Método: | Obtiene el estado del incentivo. |
| Returns: | Estado del incentivo. |

Clase IncentiveEmployee:

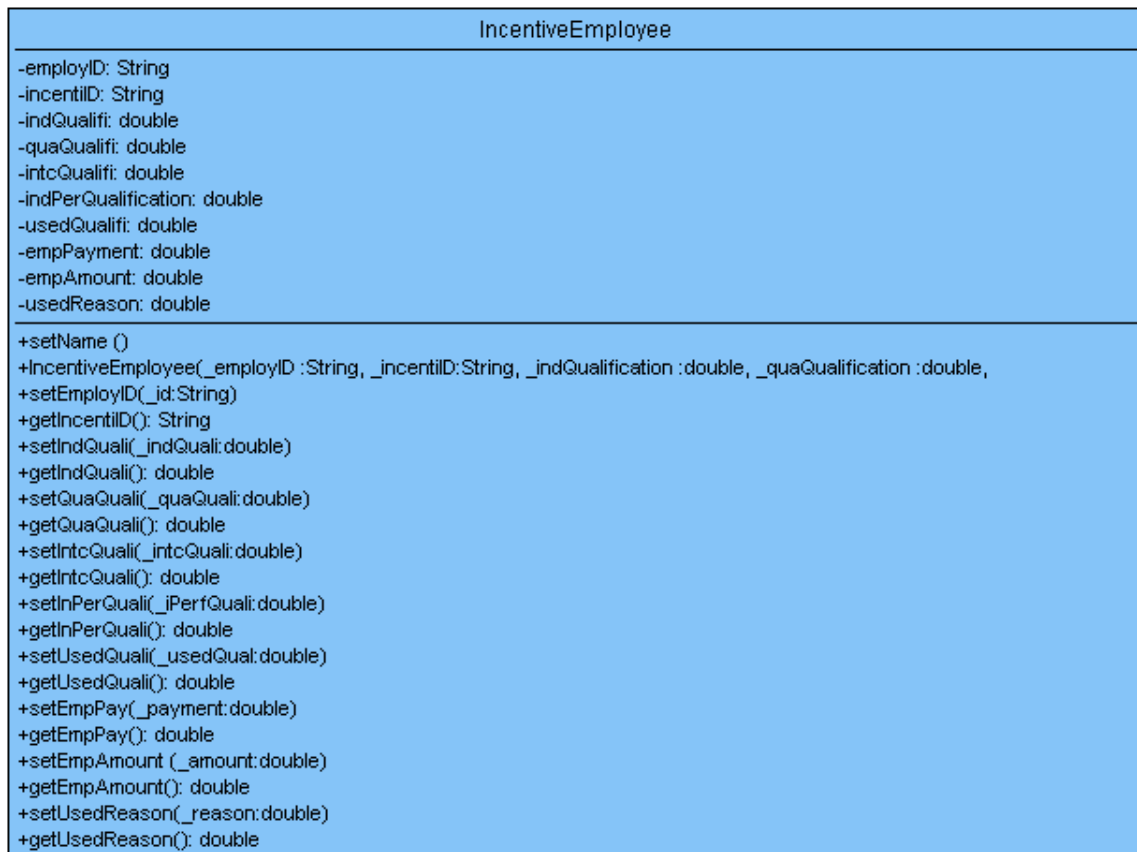


FIGURA. 5. CLASE INCENTIVEEMPLOYEE

Descripción:

Esta clase (Figura.5) se encarga de crear el objeto que representa un incentivo otorgado a un empleado. Contiene toda la información necesaria sobre el incentivo otorgado a un empleado y las características con las cuales se le adjudico y calculo el incentivo a ese empleado.

Atributos:

- Employed: Identificador del empleado.
- incentiID: Identificador del incentivo.
- indQualification: Calificación obtenida en la evaluación Individual obtenida.
- quaQualification: Calificación obtenida en la evaluación Cuantitativa.
- intcQualification: Calificación obtenida en la evaluación de Cliente Interno.

- **indPerQualification:** Calificación obtenida en la evaluación de desempeño individual.
- **usedQualification:** Calificación utilizada en la formula.
- **empPayment:** Salario del empleado.
- **empAmount:** Monto del incentivo entregado al empleado.
- **usedReason:** Razón salarial utilizada en el cálculo del incentivo.

Constructores:

| | |
|-------------------------------------|--|
| Nombre del Constructor: | <code>public IncentiveEmployee()</code> |
| Descripción del Constructor: | Constructor nulo de la clase IncentiveEmployee. Crea una instancia del objeto IncentiveEmployee con valores por defecto o nulos. |

| | |
|-------------------------------------|--|
| Nombre del Constructor: | <code>public IncentiveEmployee(String _employID, String _incentiID, double _indQualifi, double _quaQualifi, double _intcQualifi, double _indPerQualifi, double _unitPerQualifi, double _usedQualifi, double _empPayment, double _empAmount, double _usedReason)</code> |
| Descripción del Constructor: | Crea una instancia del objeto IncentiveEmployee con los valores enviados por parámetro |
| Parámetro: | Descripción |
| <code>_employID</code> | Identificador del empleado. |
| <code>_incentiID</code> | Identificador del incentivo asignado. |
| <code>_indQualifi</code> | Calificación de la evaluación individual. |
| <code>_quaQualifi</code> | Calificación de la evaluación cuantitativa. |
| <code>_intcQualifi</code> | Calificación de la evaluación de cliente interno. |
| <code>_indPerQualifi</code> | Calificación de la evaluación de desempeño. |
| <code>_usedQualifi</code> | Calificación utilizada para calcular en incentivo. |
| <code>_empPayment</code> | Salario del empleado utilizado para calcular el incentivo. |
| <code>_empAmount</code> | Monto del incentivo asignado al empleado. |
| <code>_usedReason</code> | Razón salarial utilizada en el calculo del incentivo. |

Descripción de Métodos:

| | |
|--------------------------------|--|
| Nombre del Método: | <i>public void setEmployID(String _id)</i> |
| Descripción del Método: | Asigna el código de identificación del empleado. |
| Parámetro: | Descripción |
| _id | Código de identificación del empleado. |

| | |
|--------------------------------|---|
| Nombre del Método: | <i>public String getEmployID()</i> |
| Descripción del Método: | Obtiene el código de identificación del empleado. |
| Returns: | Código de identificación del empleado. |

| | |
|--------------------------------|---|
| Nombre del Método: | <i>public void setIncentiID(String _id)</i> |
| Descripción del Método: | Asigna el código de identificación del incentivo. |
| Parámetro: | Descripción |
| _id | Código de identificación del incentivo. |

| | |
|--------------------------------|--|
| Nombre del Método: | <i>public String getIncentiID()</i> |
| Descripción del Método: | Obtiene el código de identificación del incentivo. |
| Returns: | Código de identificación del incentivo. |

| | |
|--------------------------------|---|
| Nombre del Método: | <i>public void setIndQuali(double _indQuali)</i> |
| Descripción del Método: | Asigna la calificación de la evaluación individual. |
| Parámetro: | Descripción |
| _indQuali | Calificación de la evaluación individual. |

| | |
|--------------------------------|--|
| Nombre del Método: | <i>public double getIndQuali()</i> |
| Descripción del Método: | Obtiene la calificación de la evaluación individual. |
| Returns: | Calificación de la evaluación individual. |

| | |
|--------------------------------|---|
| Nombre del Método: | <i>public void setQuaQuali(double _quaQuali)</i> |
| Descripción del Método: | Asigna la calificación de la evaluación cuantitativa. |
| Parámetro: | Descripción |
| _quaQuali | Calificación de la evaluación cuantitativa. |

| | |
|--------------------------------|--|
| Nombre del Método: | <i>public double getQuaQuali(){</i> |
| Descripción del Método: | Obtiene la calificación de la evaluación cuantitativa. |
| Returns: | Calificación de la evaluación cuantitativa. |

| | |
|--------------------------------|---|
| Nombre del Método: | <i>public void setIntcQuali(double _intcQuali)</i> |
| Descripción del Método: | Asigna la calificación de la evaluación de cliente interno. |
| Parámetro: | Descripción |
| _intcQuali | Calificación de la evaluación de cliente interno. |

| | |
|--------------------------------|--|
| Nombre del Método: | <i>public double getIntcQuali()</i> |
| Descripción del Método: | Obtiene la calificación de la evaluación de cliente interno. |
| Returns: | Calificación de la evaluación de cliente interno. |

| | |
|--------------------------------|--|
| Nombre del Método: | <i>public void setInPerQuali(double _iPerfQuali)</i> |
| Descripción del Método: | Asigna la calificación de la evaluación de desempeño individual. |
| Parámetro: | Descripción |
| _iPerfQuali | Calificación de la evaluación de desempeño individual. |

| | |
|--------------------------------|---|
| Nombre del Método: | <i>public double getInPerQuali()</i> |
| Descripción del Método: | Obtiene la calificación de la evaluación de desempeño individual. |
| Returns: | Calificación de la evaluación de desempeño individual. |

| | |
|--------------------------------|---|
| Nombre del Método: | <i>public void setUsedQuali(double _usedQual)</i> |
| Descripción del Método: | Asigna la calificación utilizada en la fórmula. |
| Parámetro: | Descripción |
| _usedQual | Calificación utilizada en la fórmula. |

| | |
|--------------------------------|--|
| Nombre del Método: | <i>public double getUsedQuali()</i> |
| Descripción del Método: | Obtiene la calificación utilizada en la fórmula. |
| Returns: | Calificación utilizada en la fórmula. |

| | |
|--------------------------------|---|
| Nombre del Método: | <i>public void setEmpPay(double _payment)</i> |
| Descripción del Método: | Asigna el salario del empleado usado en la fórmula. |
| Parámetro: | Descripción |
| _payment | Salario del empleado usado en la fórmula. |

| | |
|--------------------------------|--|
| Nombre del Método: | <i>public double getEmpPay()</i> |
| Descripción del Método: | Obtiene el salario del empleado usado en la fórmula. |
| Returns: | Salario del empleado usado en la fórmula. |

| | |
|--------------------------------|---|
| Nombre del Método: | <i>public void setEmpAmount(double _amount)</i> |
| Descripción del Método: | Asigna el monto del incentivo dado al empleado. |
| Parámetro: | Descripción |
| _amount | Monto del incentivo dado al empleado. |

| | |
|--------------------------------|--|
| Nombre del Método: | <i>public double getEmpAmount()</i> |
| Descripción del Método: | Obtiene el monto del incentivo dado al empleado. |
| Returns: | Monto del incentivo dado al empleado. |

| | |
|--------------------------------|---|
| Nombre del Método: | <i>public void setUsedReason(double _reason)</i> |
| Descripción del Método: | Asigna la razón salarial utilizada en la fórmula. |
| Parámetro: | Descripción |
| _reason | Razón salarial utilizada en la fórmula |

| | |
|--------------------------------|--|
| Nombre del Método: | <i>public double getUsedReason()</i> |
| Descripción del Método: | Obtiene la razón salarial utilizada en la fórmula. |
| Returns: | Razón salarial utilizada en la fórmula |

Clase IncentiveDB

| IncentiveDB |
|---|
| <pre> +IncentiveDB(ctx:Context) +getInstance(ctx:Context): IncentiveDB +insertToDB(_incentive:Incentive, _user:User): String +insertIncEmpToDB(_incEmp:IncentiveEmployee, _iUser:User) +getRowCount(_incentiveID:String, _name:String, _initDate:java.sql.Timestamp, _endDate:java.sql.Timestamp): double +getIncentivesInPages(_start:int, _end:int, _incentiveID:String, _name:String, _initDate:java.sql.Timestamp, +getIncentive(_incentiveID:String): Incentive +getIncentiveEmployee(_incID:String, _empID:String): IncentiveEmployee +getIncentiveEmployees(_incID:String): ArrayList +updateOnDB(_incentive:Incentive, _user:User) +updateIncState(_incID:Incentive, _user:User) +deleteFromDB(_incentiveID:String, _user:User) +deleteIncEmployee (_incentiveID:String, _employeeID:String) +getEmployees(_grade:double, _initDate:java.sql.Timestamp, _endDate:java.sql.Timestamp): ArrayList +getQualification(_employeeID:String, _initDate:java.sql.Timestamp, _endDate:java.sql.Timestamp, _quaType:int): ArrayList +getPeriod(_employeeID:String, _initDate:java.sql.Timestamp, _endDate:java.sql.Timestamp): int +getPay (_employeeID:String): double </pre> |

FIGURA. 6. CLASE INCENTIVEDB

Descripción:

Esta clase (Figura.6) se encarga ejecutar todas las instrucciones correspondientes a la inserción, modificación y borrado de la información en la base de datos y que se corresponden al módulo de incentivos.

Constructores:

| | |
|-------------------------------------|--|
| Nombre del Constructor: | <i>public IncentiveDB(Context ctx)</i> |
| Descripción del Constructor: | Constructor de la clase. |
| Parámetro: | Descripción |
| _ctx | Contexto de la aplicación. |

| | |
|-------------------------------------|---|
| Nombre del Constructor: | <i>public static IncentiveDB getInstance(Context ctx)</i> |
| Descripción del Constructor: | Contexto de la aplicación que realiza la llamada. |
| Parámetro: | Descripción |
| _ctx | Identificador del empleado. |
| Returns: | La instancia del objeto. |

Descripción de Métodos:

| | |
|--------------------------------|---|
| Nombre del Método: | <i>public String insertToDB(Incentive _incentive, User _user) throws NorthekException</i> |
| Descripción del Método: | Inserta en la base de datos un nuevo incentivo. |
| Parámetro: | Descripción |
| _incentive | Objeto incentive a insertar en la base de datos. |
| _user | Usuario actual del sistema. |
| Returns: | Código de identificación asignado al incentivo. |
| Throws: | NorthekException: Si sucedió alguna excepción durante el proceso. |

| | |
|--------------------------------|---|
| Nombre del Método: | <i>public void insertIncEmpToDB(IncentiveEmployee _iIncEmp, User _iUser) throws NorthekException</i> |
| Descripción del Método: | Inserta en la base de datos la información de un incentivo otorgado a un empleado. |
| Parámetro: | Descripción |
| _iIncEmp | Objeto IncentiveEmployee a insertar en la base de datos. |
| _iUser | Usuario actual del sistema. |
| Throws: | NorthekException: Si sucedió alguna excepción durante el proceso. |

| | |
|--------------------------------|--|
| Nombre del Método: | <i>public double getRowCount(String _incentiveID, String _name, java.sql.Timestamp _initDate, java.sql.Timestamp _endDate) throws NorthekException</i> |
| Descripción del Método: | Obtiene el número total de incentivos en el sistema. |
| Parámetro: | Descripción |
| _incentiveID | Identificador del incentivo. |
| _name | Nombre del incentivo. |
| _initDate | Fecha de inicio del incentivo. |
| _endDate | Fecha de fin del incentivo. |
| Returns: | El total de incentivos. |
| Throws: | NorthekException: Si sucedió alguna excepción durante el proceso. |

| | |
|--------------------------------|---|
| Nombre del Método: | <i>public ArrayList<NtkResultSet> getIncentivesInPages(int _start, int _end, String _incentiveID, String _name, java.sql.Timestamp _initDate, java.sql.Timestamp _endDate) throws NortheKException</i> |
| Descripción del Método: | Obtener los incentivos de la base de datos por página. |
| Parámetro: | Descripción |
| _start | Inicio de la paginación. |
| _end | Fin de la paginación. |
| _incentiveID | Identificador del incentivo. |
| _name | Nombre del incentivos. |
| _initDate | Fecha de inicio del incentivo. |
| _endDate | Fecha de fin del incentivo. |
| Returns: | ArrayList con los incentivos por página. |
| Throws: | NortheKException: Si sucedió alguna excepción durante el proceso. |

| | |
|--------------------------------|---|
| Nombre del Método: | <i>public Incentive getIncentive(String _incentiveID) throws NortheKException</i> |
| Descripción del Método: | Obtiene un incentivo de la base de datos. |
| Parámetro: | Descripción |
| _incentiveID | Identificador del incentivo. |
| Returns: | Objeto Incentivo específico. |
| Throws: | NortheKException: Si sucedió alguna excepción durante el proceso. |

| | |
|--------------------------------|---|
| Nombre del Método: | <i>public IncentiveEmployee getIncentiveEmployee(String _incID, String _empID) throws NortheKException</i> |
| Descripción del Método: | Obtiene la información un incentivo otorgado a un empleado. |
| Parámetro: | Descripción |
| _incID | Identificador del incentivo. |
| _empID | Identificador del empleado. |
| Returns: | Objeto IncentiveEmployee con la información solicitada. |
| Throws: | NortheKException: Si sucedió alguna excepción durante el proceso. |

| | |
|--------------------------------|---|
| Nombre del Método: | <i>Public ArrayList<NtkResultSet> getIncentiveEmployees (String _incID) throws NorthekException</i> |
| Descripción del Método: | Obtiene toda la información sobre los montos de un incentivo otorgados a los empleados. |
| Parámetro: | Descripción |
| _incID | Identificador del incentivo. |
| Returns: | ArrayList con la información sobre la asignación del incentivo. |
| Throws: | NorthekException: Si sucedió alguna excepción durante el proceso. |

| | |
|--------------------------------|---|
| Nombre del Método: | <i>public void updateOnDB(Incentive _incentive, User _user) throws NorthekException</i> |
| Descripción del Método: | Actualiza un Objeto Incentive en la base de datos. |
| Parámetro: | Descripción |
| incentive | Identificador del incentivo. |
| _user | Usuario del sistema. |
| Throws: | NorthekException: Si sucedió alguna excepción durante el proceso. |

| | |
|--------------------------------|--|
| Nombre del Método: | <i>public void updateIncState (Incentive _incentive, User _user) throws NorthekException</i> |
| Descripción del Método: | Actualiza el estado del incentivo en la base de datos. |
| Parámetro: | Descripción |
| incentive | Identificador del incentivo. |
| _user | Usuario del sistema. |
| Throws: | NorthekException: Si sucedió alguna excepción durante el proceso. |

| | |
|--------------------------------|---|
| Nombre del Método: | <i>public void deleteFromDB (Incentive _incentive, User _user) throws NorthekException</i> |
| Descripción del Método: | Elimina el objeto de la base de datos que corresponda al identificador enviado por parámetro. |
| Parámetro: | Descripción |
| _incentive | Identificador del incentivo. |
| _user | Usuario del sistema. |
| Throws: | NorthekException: Si sucedió alguna excepción durante el proceso. |

| | |
|--------------------------------|--|
| Nombre del Método: | <i>public void deleteIncEmployee (String _incentiveID, String _employeeID) throws NortheKException</i> |
| Descripción del Método: | Elimina el detalle de un incentivo otorgado a un empleado. |
| Parámetro: | Descripción |
| _incentiveID | Identificador del incentivo. |
| _employeeID | Identificador del empleado. |
| Throws: | NortheKException: Si sucedió alguna excepción durante el proceso. |

| | |
|--------------------------------|---|
| Nombre del Método: | <i>public ArrayList<NtkResultSet> getEmployees(double _grade, java.sql.Timestamp _initDate, java.sql.Timestamp _endDate) throws NortheKException</i> |
| Descripción del Método: | Obtiene los empleados elegibles para un incentivo de la base de datos. |
| Parámetro: | Descripción |
| _grade | Nota requerida para optar por el incentivo. |
| _initDate | Fecha de inicio del incentivo. |
| _endDate | Fecha de fin del incentivo. |
| Returns: | ArrayList con los empleados que cumplen con los requisitos. |
| Throws: | NortheKException: Si sucedió alguna excepción durante el proceso. |

| | |
|--------------------------------|--|
| Nombre del Método: | <i>public ArrayList<NtkResultSet> getQualification(String _employeeID, java.sql.Timestamp _initDate, java.sql.Timestamp _endDate, int _quaType) throws NortheKException</i> |
| Descripción del Método: | Obtiene las notas de las evaluaciones de un empleado en específico. |
| Parámetro: | Descripción |
| _employeeID | Identificador del empleado. |
| _initDate | Fecha de inicio del incentivo. |
| _endDate | Fecha de fin del incentivo. |
| _quaType | Tipo de calificación a la cual pertenece la nota. |
| Returns: | ArrayList con las notas de la evaluación. |
| Throws: | NortheKException: Si sucedió alguna excepción durante el proceso. |

| | |
|--------------------------------|--|
| Nombre del Método: | <i>public int getPeriod(String _employeeID, java.sql.Timestamp _initDate, java.sql.Timestamp _endDate) throws NorthekException</i> |
| Descripción del Método: | Obtiene el periodo laborado de un empleado basado en el periodo del incentivo. |
| Parámetro: | Descripción |
| _employeeID | Identificador del empleado. |
| _initDate | Fecha de inicio del incentivo. |
| _endDate | Fecha de fin del incentivo. |
| Returns: | Periodo laborado del empleado. |
| Throws: | NorthekException: Si sucedió alguna excepción durante el proceso. |

| | |
|--------------------------------|--|
| Nombre del Método: | <i>public double getPay (String _employeeID) throws NorthekException</i> |
| Descripción del Método: | Obtiene el salario de un empleado. |
| Parámetro: | Descripción |
| _employeeID | Identificador del empleado. |
| Returns: | Salario del empleado. |
| Throws: | NorthekException: Si sucedió alguna excepción durante el proceso. |

Interfaces de Usuario

Las interfaces de usuarios diseñadas hasta el momento para el módulo de incentivos son:



Manejo de Incentivos

Administración general de los incentivos del sistema.

Filtrar Incentivos

| Código | Nombre | Monto | Día de Inicio | Día de Finalización |
|--------|-------------------------|-----------|-------------------|---------------------|
| inc-1 | Incentivo de excelencia | 6,000,000 | 1 de mayo de 2006 | 31 de mayo de 2006 |

FIGURA. 7. INTERFAZ DEL INDEX DE INCENTIVOS

En esta sección (Figura.7) se muestran los incentivos que tiene el sistema, y se brinda la posibilidad de ver los datos mas significativos de los mismos como lo son el código, el nombre, el monto y las fechas de inicio y fin para el periodo de vigencia del incentivo. Además brinda las opciones de poder ver toda le información del incentivo (sin la posibilidad de editarlos), la de calcular y asignar el incentivo, la de editarlo y también la de eliminarlo del sistema y por ultimo en la parte inferior cuenta con un botón el cual permite crear un nuevo incentivo.

La sección de ver un incentivo (Figura.8) se ve de la siguiente forma:



incentivos

Administración general de los incentivos del sistema.

Datos Generales del Incentivo [inc-1]

Características del Incentivo

Nombre:

Monto:

Descripción:

Requisitos

Periodo minimo laborado:

Nota minima:

Evaluaciones Utilizadas

Evaluaciones usadas y su porcentaje representativo

Individual: %

Cuantitativa: %

Cliente Interno: %

Se utiliza solo la nota de Desempeño

Periodo

Periodo para el que aplicara el incentivo.

Fecha Inicio:

Fecha Fin:

FIGURA. 8. INTERFAZ VER INFORMACIÓN DE UN INCENTIVO

Tiene la opción de ver todos los datos del incentivo que se explicaran a continuación en el apartado de crear un incentivo.

En esta sección se crea el incentivo (Figura.9) el cual debe de contener la siguiente información. Como referencia del incentivo se tiene un nombre, un monto, una descripción y fechas de inicio y fin. Por otro lado contiene requisitos los cuales son la nota mínima ponderada que debe de tener un empleado para optar por el incentivo y el periodo mínimo de tiempo laborado. Y finalmente las evaluaciones a tomar en cuenta y el porcentaje que representaran en la ponderación de notas.



Manejo de Incentivos

Administración de los incentivos de la empresa.

Complete los datos solicitados para crear un nuevo incentivo.

Datos Generales

Características del Incentivo

Nombre: *
Incentivo de excelencia

Monto: *
\$6000000

Descripción: *
Este incentivo se otorga a todos los empleados que.....

Evaluaciones a Considerar

Evaluaciones que se tomarán en cuenta para otorgar el incentivo y su porcentaje representativo

Individual 75 %

Cuantitativa 25 %

Cliente Interno 0 %

NOTA: Si no selecciona ninguna se utilizará la nota de desempeño como referencia.

Requisitos

Periodo mínimo laborado: *
1

Nota mínima: *
90

Periodo

Periodo para el cual se aplicará el incentivo.

Fecha Inicio: *
01/05/2006

Fecha Fin: *
31/05/2006

FIGURA. 9. INTERFAZ DE CREAR UN NUEVO INCENTIVO

La parte de editar el incentivo es exactamente igual al de crear con las diferencia editar contiene el identificador del incentivo en la parte superior como se muestra en la siguiente imagen (Figura.10):



Incentivos

Administración general de los incentivos del sistema.

Datos Generales del Incentivo [inc-1]

Características del Incentivo

Nombre: *
Incentivo de excelencia

Monto: *
\$6000000

Descripción: *
Este incentivo se otorga a todos los empleados que.....

Evaluaciones Utilizadas

Evaluaciones usadas y su porcentaje representativo

Individual: 75.0 %

Cuantitativa 25.0 %

Cliente Interno 0.0 %

NOTA: Si no selecciona ninguna se utilizará la nota de desempeño como referencia.

Requisitos

Periodo mínimo laborado: *
1.0

Nota mínima: *
90.0

Periodo

Periodo para el que aplicará el incentivo.

Fecha Inicio: *
01/05/2006

Fecha Fin: *
31/05/2006

FIGURA. 10. INTERFAZ DEL ETITAR UN INCENTIVO

Y por ultimo esta la sección de calcular y asignar un incentivo.



Calcular y Asignar Incentivos

Administración general del calculo y asignación de Incentivos.

Seleccione a un empleado y calcule su incentivo para agregarlo a la lista de elegidos.

Asignación del Incentivo[inc-1] Monto: 6,000,000

Empleados Elegibles

| Código | Nombre |
|-----------|----------------------|
| 2-308-006 | GUIDO ARCE RODRIGUEZ |

Lista de empleados que cumplen con los requisitos para optar por el incentivo.

Calcular Incentivo

Nombre: *GUIDO ARCE RODRIGUEZ

Código: * 2-308-006

| R | S | C | MI / Me |
|-----|-------|-----|---------|
| 0.9 | ***** | 1.0 | 0.8 |

Monto: * 72000

R: Razón salarial.
S: Salario base del empleado.
C: Calificación obtenida.
MI: Meses laborados / Me: Meses evaluados.

Empleados con el incentivo calculado

| Código | Nombre | Monto Incentivo |
|-----------|-------------------------|-----------------|
| 2-581-959 | YANINA SABORIO GONZALEZ | 21,000 |

Lista de los empleados a los cuales les asigno un incentivo.

Total incentivos asignados: **21,000**

FIGURA. 11. INTERFAZ DEL CALCULAR Y ASIGNAR UN INCENTIVO

Aquí el sistema muestra (véase Figura.11) al lado izquierdo la lista de empleados que cumplen con los requisitos, el usuario selecciona un empleado y al hacerlo se carga la información del empleado en el centro de la pagina en la cual el usuario completara algunos de los datos para calcular la el incentivo utilizando la fórmula, una vez calculado podrá ingresar al empleado en la lista que se encuentra a la derecha la cual es la que contiene todos los empleados a los que se les a asignado un incentivo.

El usuario puede eliminar empleado de la lista de la empleados con incentivos a signados si lo desea y volver a calcular el incentivo variando datos de la fórmula. Al finalizar de asignar los incentivos el usuario puede salir de la sección y seguir calculando los incentivos en otra ocasión o cerrar el incentivo lo que indica que ese incentivo no se podrá asignar a otros empleados.

Diseño de Base de Datos

El diseño de la base de datos es pequeño, únicamente consta de dos tablas las cuales se muestran a continuación en la Figura.12:

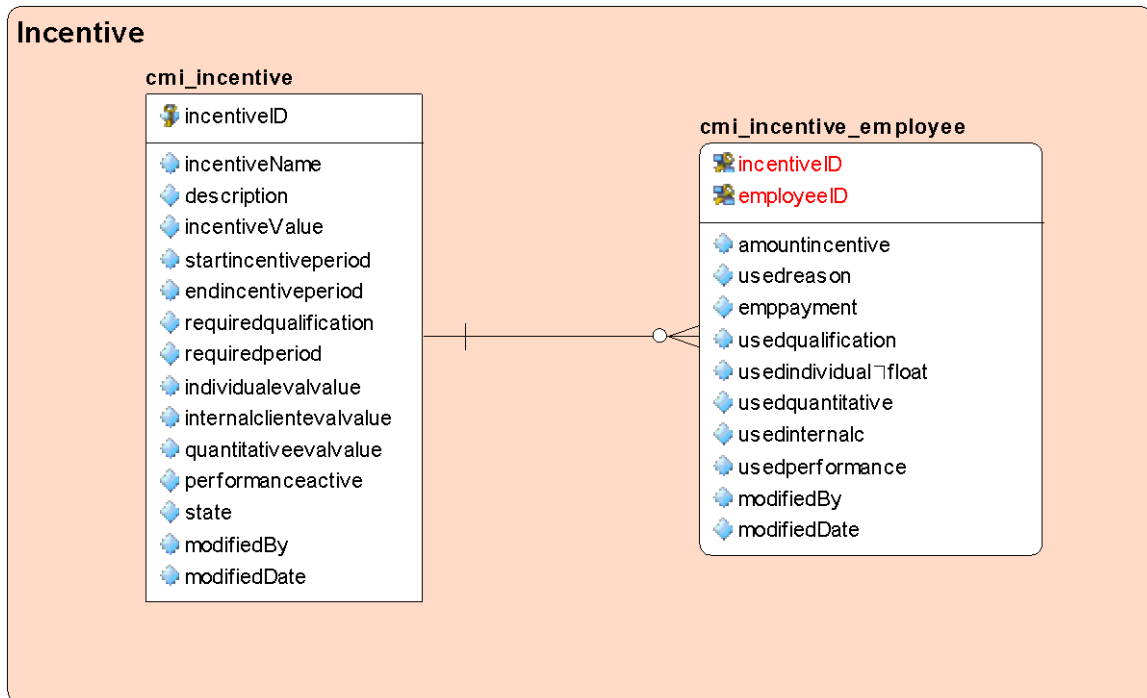


FIGURA. 12. DISEÑO DE LA BASE DE DATOS

Estas dos tablas esta relacionas con otras del sistema pero lo extenso del mismo de decidió no mostrarlas en este documento.

Reseña sobre CMISOFT

Antes de entrar con lo que consiste la modularización del sistema veremos de manera resumida y rápida en que consiste el sistema CMISOFT y algunos conceptos básicos para el entendimiento del mismo.

CMISOFT es un programa de Cuadro de mando integral (CMI), el cual permite a las empresas por medio de indicadores llevar a cabo un plan estratégico para determinar el estado y el funcionamiento de la compañía.

El núcleo de CMISOFT lo componen los indicadores, un indicador lo podemos definir como la evolución de un parámetro a lo largo del tiempo. Una representación gráfica permite visualizarlos e interpretarlos fácilmente. En el caso del CMISOFT, los indicadores miden la evolución de los parámetros asociados a los objetivos estratégicos.

Los indicadores se les asocia muchos elementos del sistema entre ellos un administrador, el cual es el único que puede modificar los datos del mismo, también una serie de entidades (las cuales lo observan (*Visualizador*), lo alimentan (*Evaluador*) o dependen del valor del indicador para una evaluación (*Calificado*)), además están relacionados con otros indicadores, alertas, FCE (factores críticos de éxito) y riesgos. Estos tres últimos los veremos mas adelante.

También el sistema cuenta con tableros de control que son básicamente indicadores agrupados por un criterio en específico, el cual queda a criterio del usuario y permite ver el estado de los indicadores asociados a ese criterio. Por ejemplo se puede crear un tablero que contenga los indicadores de la unidad de recursos humanos del sistema, entonces el usuario que creó ese tablero al ingresar podrá ver en que estado están los indicadores y verificar la información de los mismos sin la necesidad de tener que ir a buscarlo entre todos los indicadores del sistema.

Dejando de lado los indicadores entraremos ahora a la sección de estructura organizacional, la cual es en la que se administran los empleados, las unidades, las facultades y las empresas. Estos criterios no hace falta explicarlos ya que son bastante

claros, lo único que comentaremos es que a los empleados se les asocia con una facultate específica, al empleado se le asocia con una unidad, las unidades son asociadas a las empresas y también se asocian entre ellas esto con el fin de poder evaluar a nivel de cliente interno.

También se maneja en CMISOFT lo que son áreas y metas estratégicas, perspectivas y FCE, todos estos elementos se utilizan para definir y marcar la pauta a nivel estratégico como bien se indica en los nombres, se fijan metas y objetivos a seguir con una misión definida y con la cual se crean los FCE que están asociados a los indicadores. No vamos a profundizar mucho en el tema ya que todos estos elementos son de planeación estratégica y no se centran mucho en el funcionamiento del sistema.

Ahora entramos a lo que refiere a los riesgos, los cuales son características, circunstancias, o rasgos del ambiente del pueden alterar creando efectos adversos en el proyecto o en los indicadores afectando el buen desempeño o la calidad de estos, estos también se manejan de acuerdo a una matriz de riesgo que es una representación de los valores de Impacto contra la probabilidad en la que un riesgo puede evaluarse. La matriz esta compuesta por valores de probabilidad, valores de impacto, categorías y valores.

Además en el sistema se manejan las acciones que son tareas a realizar de acuerdo al estado de un indicador o si no se llega a cumplir por ejemplo la meta estratégica que fue asignada al mismo, las acciones también permiten llevar planes de acción que especifican que se debe de llevar a cabo para mantener o mejorar el estado de uno o más indicadores de una unidad empresarial específica.

Por otro lado tenemos las alertas que son objetos del sistema que pretenden ser un monitor sobre el comportamiento de los indicadores a los que se relacione. Cuando se define una alerta se define por si sola y luego esta alerta puede ser relacionada a uno o muchos indicadores.

Ahora entramos en la sección de evaluaciones la cual se utiliza para calificar el desempeño de un empleado o unidad en un periodo laborado en específico. Existen cuatro tipos de evaluaciones que son:

- La Individual: esta es la que aplica a cada empleado y es evaluado por varios entes que pueden ser jefes, compañeros o subalternos de el funcionario a evaluar.
- La cuantitativa: es la que califica ya sea a una unidad o empleado basándose en el valor actual de los indicadores asociados a ella.
- La de cliente interno: cuyo fin es de calificar el servicio de una unidad a otra.
- Y por ultimo la de desempeño que es un ponderado de todas las anteriores se puede realizar tanto a un empleado como a una unidad y el porcentaje representativo de cada evaluación lo elige el usuario al crear la evaluación.

Para finalizar se cuenta con la sección de incentivos la cual si se contempla en este documento, la de reportes y la de administraron la cual consta de configurar ciertos aspectos del sistema, así como llevar el control de los cambios ellos en una bitácora también crear usuarios y perfiles de usuarios para que utilizan el sistema.

En conclusión CMISOFT es un sistema completo el cual se pretende modularizar para adaptarlo a al necesidad de diversos clientes y brindar una solución integral en cuanto a plantación estratégica del sistema.

Modularización CMISOFT

A continuación se brindara un estudio sobre el diseño de CMISOFT, para conocer la posibilidad de modular el sistema, con el fin de poder brindar soluciones a los clientes de acuerdo a las características de su empresa.

El trabajo consiste en un análisis actual del sistema, para detectar las relaciones entre las partes del programa y conocer detalladamente el funcionamiento del mismo.

Luego se realizara una propuesta, la cual mostrara los posibles módulos en los que se puede dividir el sistema y los requisitos que debe de cumplir CMISOFT, para poder agregar alguno de los módulos, mejor dicho cuales módulos tienen que existir para utilizar algún otro.

El sistema en estos momentos esta conformado como se muestra en la Figura.13:

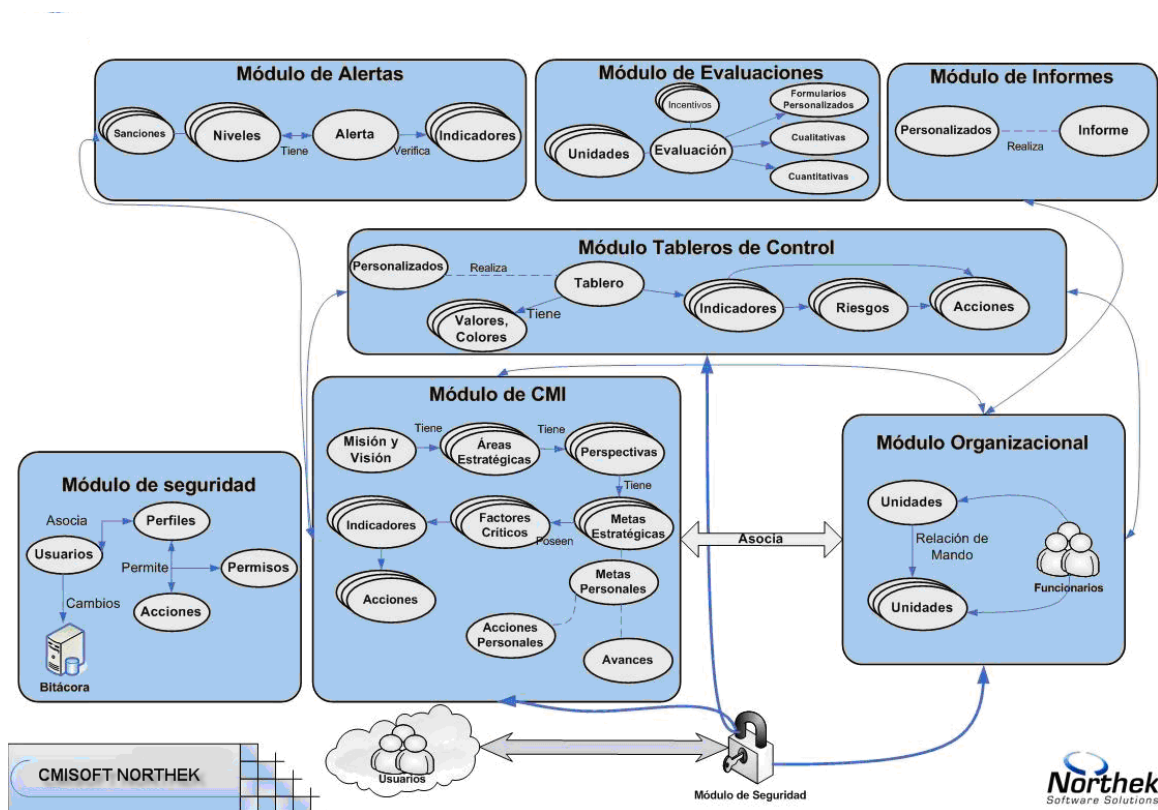


FIGURA. 13. MODELO CONCEPTUAL ACTUAL CMISOFT

Según como esta diseñado el sistema, los siguientes aspectos son partes imprescindibles de CMISOFT:

- Los indicadores.
- La estructura organizacional.
- Los FCE (Factores Críticos de Éxito).
- La seguridad y administración.

El primer punto a analizar son los indicadores que son de suma importancia para el sistema. Los indicadores son fundamentales en un CMI, por lo tanto deben de estar en el core del sistema, estos están relacionados con los empleados y los FCE (Figura.14), he aquí las primeras dependencias, de las cuales hablaremos a continuación.

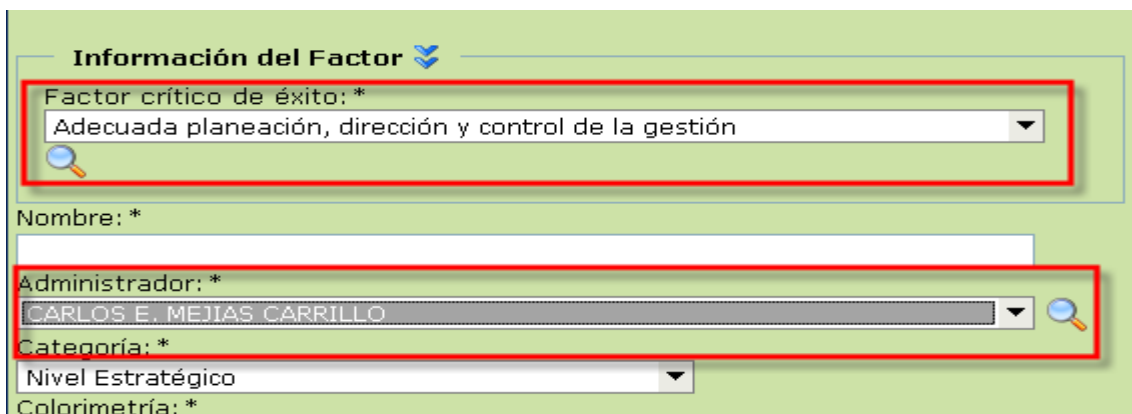


FIGURA. 14. INTERFAZ CREAR INDICADOR

Los empleados son muy utilizados en el sistema porque están relacionados con una gran cantidad de funciones, incluyendo la creación de indicadores como dijimos antes, ya que para poder crear un indicador es necesario asignarle un empleado que sea el administrador, por lo tanto también son obligatorios en el sistema. Vale la pena mencionar que los empleados necesitan una facultad para poder crearse como se muestra en la Figura.15, por lo tanto si se quieren empleados se necesitan también facultades.

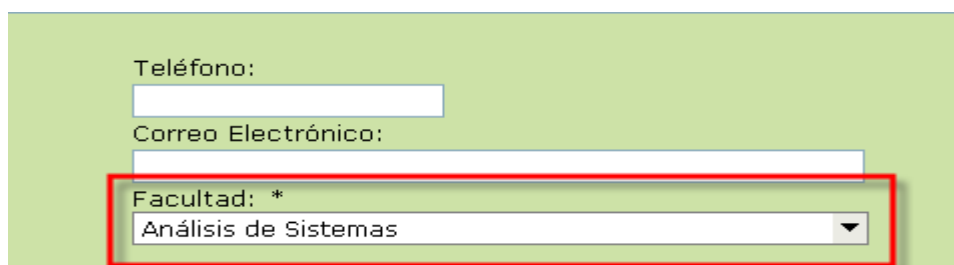


FIGURA. 15. INTERFAZ INSERTAR EMPLEADO

Por otro lado tenemos los FCE, los cuales al igual que los empleados son requisitos para crear los indicadores, un aspecto importantes aquí es que los FCE están relacionas con las perspectivas y con las áreas y metas estratégicas (Figura.16), por lo tanto también son imprescindibles para que CMISOFT funcione.



FIGURA. 16. INTERFAZ SECCIÓN GENERAL DE CMISOFT

El último requerimiento para una versión básica del sistema son las secciones de seguridad y administración, porque la función de estas dentro del sistema es indispensable.

También tenemos aspectos que se pueden manejar por aparte y de los cuales no dependen los elementos básicos del sistema, estos son: las acciones, los riesgos, los incentivos, las evaluaciones, los reportes y las alertas.

Cambiando el tema, uno de los puntos a destacar es que, a la hora de realizar una versión del sistema se tiene que cambiar partes del código para que aparezcan los tabs de algunas de las opciones que se manejaran por separado, algunos ejemplos de esta situación se mencionan a continuación.

En la parte de administración se manejan aspectos como la matriz de riesgos, los feriados etc, esto se muestra a en la Figura.17, ahora supongamos un caso en el que un cliente pida la versión mas básica, esta versión debe de tener obligatoriamente la parte de administración (ya que es la de seguridad), pero no tiene riesgos, por lo tanto se debe de eliminar la sección de matriz de riesgo del sistema.

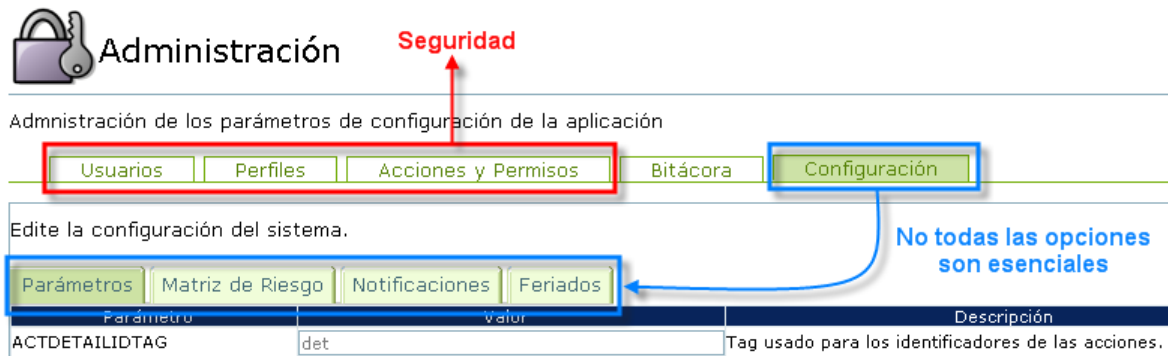


FIGURA. 17. INTERFAZ ADMINISTRACIÓN

Algo parecido se tendrá que hacer con el tablero de control, en esta sección se manejan además de los tableros, las acciones y riesgos del usuario como se indica en la Figura.18, en dado caso si la versión que el cliente no disponga de los módulos de acciones y riesgos, se tiene que actualizar la parte en el sistema, para que no despliegue las opciones de riesgos y acciones.

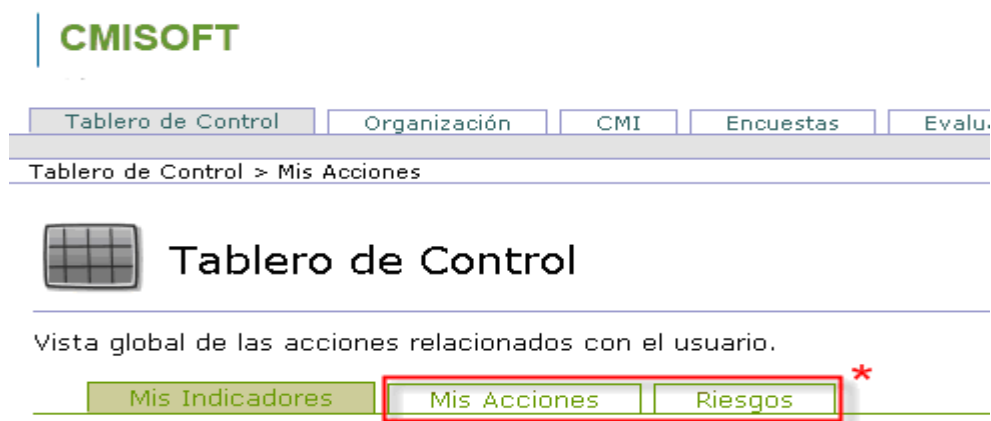


FIGURA. 18. INTERFAZ DE TABLEROS DE CONTROL

Por otro lado tenemos la sección de CMI, se tienen que ocultar los tabs de acciones, de riesgos y de alertas, o mostrar solo unas dependiendo de la elección del cliente (en la Figura.19 se señala por colores las opciones que podría contener la versión).

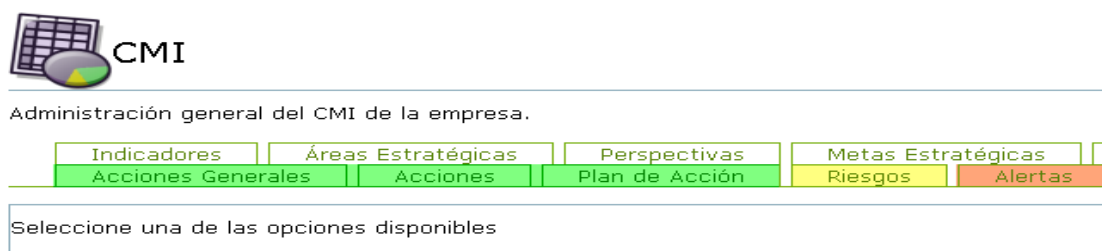


FIGURA. 19. INTERFAZ SECCIÓN GENERAL CMISOFT

Esto se da también en la parte de estructura organizacional (véase la Figura.20), ya que no siempre se manejarán unidades y empresas. Vale la pena destacar que el uso de las unidades por razones de estabilidad del sistema (evitando cambiar mucho el código fuente original) es preferible manejarlos en el core del sistema modularizado.

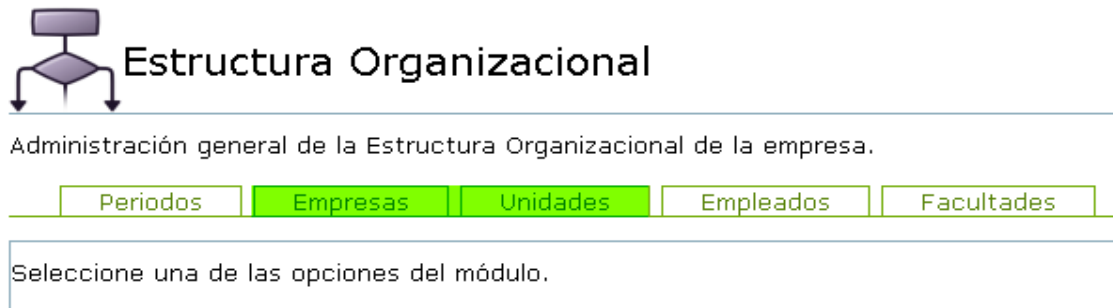


FIGURA. 20. INTERFAZ SECCIÓN ESTRUCTURA ORGANIZACIONAL

Para resolver estos problemas se creará una aplicación, capaz de tomar los archivos JSP que conforman la página web y modificar su código para que cree los tabs necesarios de acuerdo a los módulos elegidos.

Otro aspecto a considerar son los reportes, aquí la situación que se presenta es la generación de los datos que contendrá la tabla de reportes, ya que los posibles reportes a generar dependen de los módulos que use el sistema, por lo tanto se debe cargar la tabla de reportes de acuerdo a los módulos que conformen la versión de CMISOFT que el cliente desea.

Por ejemplo, si la versión no usa evaluaciones, en la lista de reportes no debería de existir la opción de generar reporte de alguna evaluación, la lista de reportes la podemos observar el Figura.21.

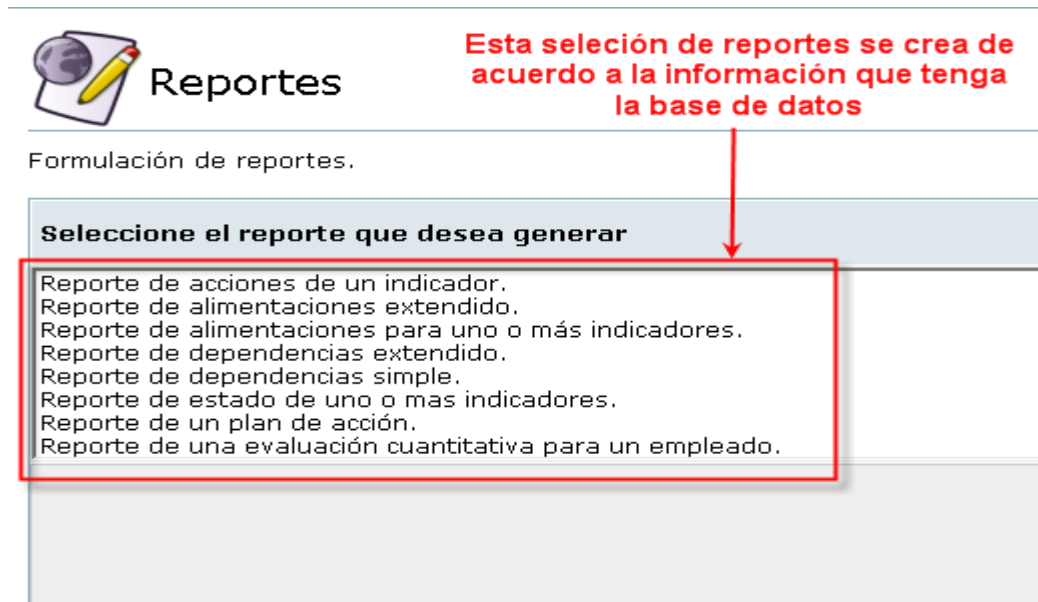


FIGURA. 21. INTERFAZ GENERACIÓN DE REPORTES

Además de lo anterior que se basa mas que todo en dependencias y presentación, se verifico el sistema por debajo y se analizó el diseño de la base de datos, para verificar que esta no presentara ningún problema a la hora de modular el sistema.

El mayor problema que se noto fue en las clases ColorValue e Indicator, las cuales usan parámetros como riesgos y alertas en su, por lo cual se deben modificar estas clases de acuerdo a los módulos que conformen la versión del sistema.

Según lo visto en el análisis anterior la propuesta que se tiene para la modulación de CMI consiste en 8 módulos, los cuales son:

- CMI (core).
- Estructura Organizacional (que será solo para las Empresas).
- Reportes.
- Riesgos.
- Alertas.
- Acciones.
- Evaluaciones.
- Incentivos.

Estos módulos se representan en el siguiente esquema (Figura.22):

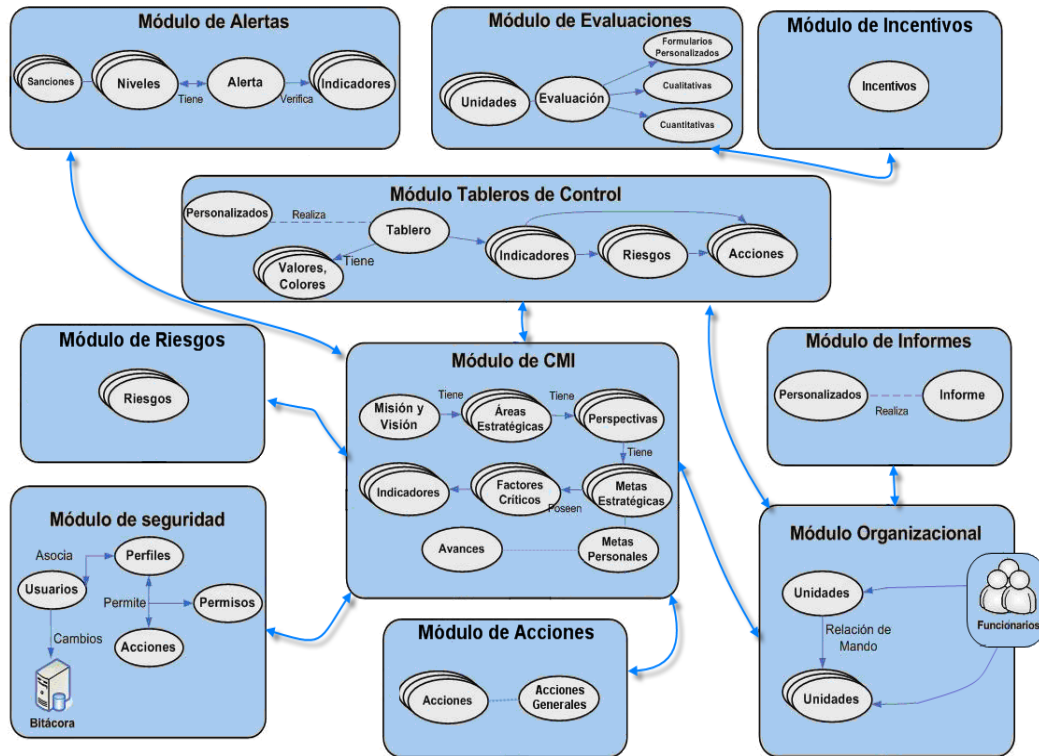


FIGURA. 22. MODULO CONCEPTUAL CMISOFT MODULARIZADO

Hay que tomar en cuenta que para unir algunos módulos es necesario que otros existan, en la Figura.23 se muestra las dependencias y que módulos que deben existir para agregar algún otro:

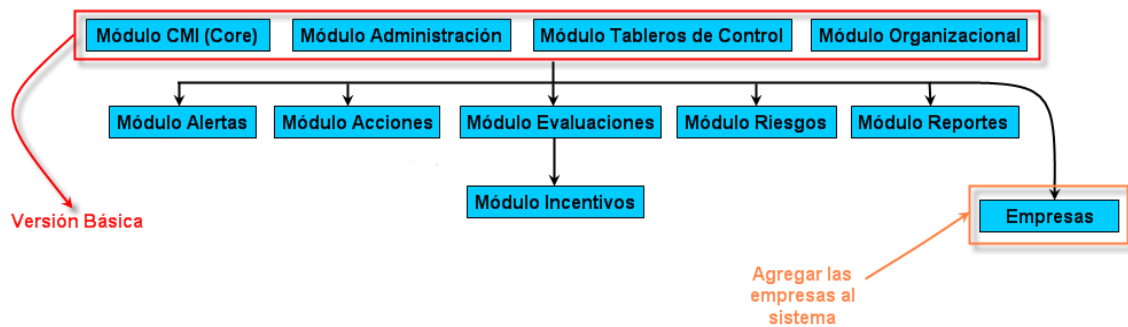


FIGURA. 23. DIAGRAMA DE DEPENDENCIAS CMISOFT MODULARIZADO

Basados la información anterior se llevo a cabo la modularización del sistema que se realizo según una serie de pasos para poder generar un producto final que cumpliera las expectativas de la empresa.

Lo primero que se hizo fue dividir la base de datos de acuerdo con los módulos en los cuales se dividirá el sistema, como dato adjunto (anexo #1, Figura.25) se encuentra el diseño de la base de datos en el cual se puede observar que las tablas tiene diferentes colores, esto es con el fin de ubicar cada una de ellas dentro de un módulo en particular y de esta manera poder visualizar cuales de ellas son necesarias para un módulo en específico.

Luego de dividir la base de datos se continuo con la reubicación de los procedimientos almacenados en los paquetes de la base de datos, en este punto se dieron algunos problemas con procedimientos que no se podían mover por razones especiales lo que llevo a tener que buscar soluciones alternativas como manejar las consultas en cadenas de caracteres para que el compilador de base de datos no diera error a la hora de compilar una función que hiciera referencia a una tabla que no existía en el sistema(estos porque el módulo que la utilizaba no estaba incluido en la versión).

Una vez concluido esto se cambiaron las llamadas en los paquetes del sistema para que hicieran referencia a la función correcta en la BD y se modificaron algunos que al igual que lo anterior generaban excepciones al un encontrar intentar leer algún dato que no existía en el sistema.

Por ultimo se hizo una aplicación que generara las versiones, a este generador de versiones se le agregaron múltiples funciones con el fin de facilitar al máximo la verificación de la versión y la implantación de la misma. A continuación realizaremos una breve descripción del generador de versiones.

El CMIVG es una aplicación en la cual se eligen dos directorios uno origen (en el cual esta el repositorio del proyecto modularizado) y el destino en el cual quedara la versión de CMISOFT. Una de la primeras funciones adicionales del generador se da a la hora de que el usuario no disponga del repositorio del sistema modularizado, si este es el caso tiene la posibilidad de indicárselo al generador y este le hará checkout en la carpeta

origen al repositorio del sistema modularizado ubicado en el servidor de la empresa. Vale la pena mencionar que el sistema utiliza método de autenticación por lo tanto CMIVG permite al usuario dar la dirección del repositorio y escribir su nombre de usuario y contraseña.

Una vez seleccionado las carpetas de origen y destino es usuario elegirá los módulos que desea para esa versión. Una vez hecho esto el sistema brinda una nueva opción, la cual es la de generar un build de la versión. Este build es un archivo comprimido que contiene el proyecto listo para la implantación y que se genera después de un proceso largo, que se explicará a continuación mientras se describe la generación de una versión.

Cuando se genera un versión lo primero que hace el sistema es copiar todos los archivos (necesarios para la versión) a la carpeta destino según corresponda, si es necesario hace un checkout para obtener el proyecto modulado como se menciona anteriormente, una vez hecho esto genera algunos archivos necesarios como los scripts instaladores de la base de datos, ya que varían de versión en versión, luego edita las clases y archivos que lo necesiten (esto se hace marcando los archivos a editar con ciertas etiquetas las cuales sirven para decirle al generador que esa sección del código puede no ser necesaria en la versión entonces debe de editar ese archivo para que no se generen problemas a la hora de compilar i ejecutar la aplicación), el generador sabe cuales archivos editar según le indique un archivo de configuración el contiene el nombre y la dirección de dichos archivos. Después elimina los archivos que no sean necesarios. Aquí termina el proceso, se genero una versión pero si el cliente opto por generar un build el proceso se hace mas largo, lo que hace el sistema una vez concluida la “movilización de archivos” es ejecutar una secuencia de instrucciones JAVA-ANT que se encargan de compilar la versión, para verificar que no tenga errores en el código, luego ejecuta los unit test para comprobar que las funciones y procedimientos trabajen adecuadamente y por ultimo genere el build con toda la información de la versión generada, vale la pena destacar que para que los units test se realicen correctamente se debe de tener instalados una serie de datos específicos en la base de datos, por lo tanto el generador antes de comenzar la operación del build limpia la base de datos e instala esta información especifica, haciendo un paréntesis en el procesos aclararemos que CMIVG también permite al usuario configurar una serie de aspectos a la hora de generar el build, como la base de datos a utilizar, las autenticación de la misma(usuario y contraseña), además de otros

detalles como especificar a quienes se desea que llegue el correo de notificación del build que indica si este se realizo correctamente o si se dio algún fallo. Volviendo al proceso de generación del build, una vez terminado se limpia de nuevo la BD y por ultimo se eliminan algunos datos temporales que se crean para lleva a cabo todo el proceso y se copia en la carpeta destino un archivo comprimido que contiene el build realizado.

Esta es la presentación de CMIVG (Figura.24).

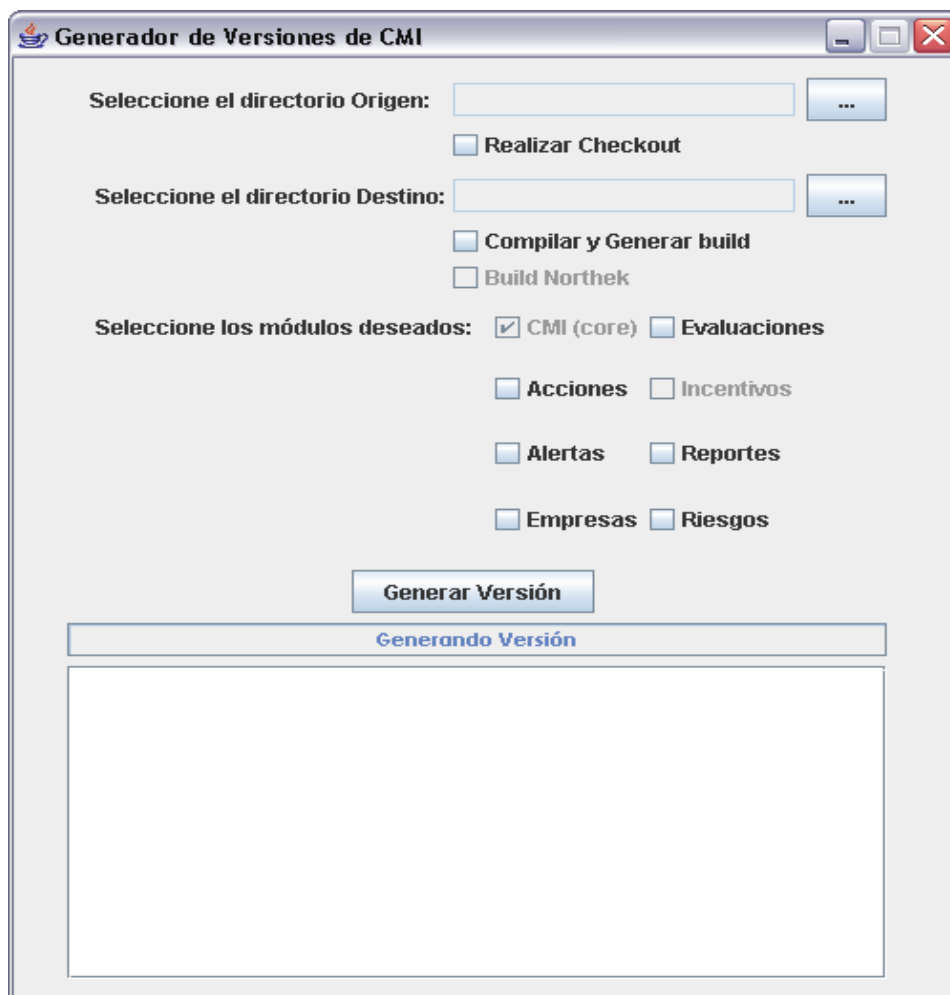


FIGURA. 24. INTERFAZ CMIVG

Recomendaciones

En este documento se a mencionado que el sistema CMISOFT es una aplicación muy mejorada ya que se a trabajado en ella por mucho tiempo cumpliendo criterios de exigencia muy grandes, por lo tanto no hay muchos aspectos que cambiar de ella, lo único que si recomendaría a futuro son dos cosas, la primera es dar un poco más de flexibilidad a la hora de manejar la formula para calcular incentivos, ya que no todas las empresas realizan los cálculos de la misma manera y en el sistema se dejo como predeterminado manejarlo de la misma manera que lo hace la empresa para la cual fue realizado. Y por ultimo algo que también mejoraría en mucho la aceptación e implantación del sistema es realizar una pequeña aplicación capaz de poder instalar el sistema y que permita al cliente crear una configuración inicial, ya que actualmente para poder implantarse se tiene que visitar al cliente y correr los scripts con actualizaciones, montar el sistema en el servidor y ponerlo a correr entre otras cosas, algo que muchas veces se torna complicado y no se debe dejarlo a responsabilidad del cliente, pero si se tuviera la aplicación seria mas sencillo hacer el proceso ya que seria como un wizard que el cliente podrá ejecutar para instalar el sistema y con el cual también se podría obtener actualizaciones cuando estas estén disponibles, por otro lado tampoco se puede dejar de lado el beneficio que traería a la empresa ya que estar enviando personal a implantar el sistema cuando el cliente se encuentra lejos representaría una perdida de recursos, la que se puede evitar.

Conclusiones y Comentarios

En si la practica se realizó de manera normal, el único objetivo que no se cumplió fue el de realizar el sitio Web ya que no se tenían listos algunos aspectos importantes para el desarrollo del mismo, entre los cuales se puede mencionar la falta de un servidor adecuado para montar el proyecto (el cual es una aplicación Web) debido a que el sistema trabaja sobre una base de datos Oracle y el servidor con el que se contaba no soportaba este tipo de base de datos. Por otro lado se tenían dudas en cuanto a la función que se le iba a dar al sitio, entre las opciones que se manejaban estaban si el sitio únicamente se utilizaría para mostrar al cliente la forma en la cual se usaba el sistema o si también se iba a dar la posibilidad de poder usar el sistema en línea, algo que sin duda alguna requiere de mucho trabajo y no se había estipulado manejarlo de esa manera.

Como se mencionó anteriormente no se realizó el sitio web, por tal razón se decidió dar prioridad al generador de versiones. En secciones anteriores se describió el funcionamiento del mismo y se explicaron muchas de las funciones que realiza de las cuales la mayoría no estaban especificadas al inicio del proyecto. El generador se mejoro tanto de llegar al punto de automatizar totalmente algunos procesos que anteriormente se hacían de forma manual y ayudó a que muchas de las personas que trabajan en la empresa que no estaban familiarizadas con algunas de las funciones como la de generar builds lo pudieran hacer sin ningún problema y de una forma rápida, lo cual facilitaba el trabajo, ya que por ejemplo un empleado del departamento de QA ya no tiene que esperar a que un funcionario de desarrollo genere el build para poder probarlo, ya que ahora él genera la versión que desea probar y de una vez es generado el build correspondiente.

En cuanto a la experiencia obtenida en el desarrollo de la práctica fue mucha, ya que no solo se participó en el proyecto de la práctica sino también en varios otros proyectos pequeños que se realizaron de los cuales se aprendió mucho acerca de diferentes metodologías y herramientas de trabajo.

Uno de los aspectos importantes fue tomar parte de la reuniones con el cliente ya que son de vital importancia en el desarrollo de un proyecto y en ocasiones estas reuniones se tornan complicadas ya que no se da un buen entendimiento entre ambas partes, lo cual ayuda a tomar experiencia para situaciones futuras y de esta manera saber como enfrentarlas.

También se aprendió mucho de nuevas tecnologías, como fue el desarrollo en JSP con una base de datos Oracle, herramientas que nunca antes había usado y son muy utilizadas hoy en día y lo mas importante trabajar en una aplicación desarrollada en la plataforma J2EE, la cual es altamente solicitada para el desarrollo de aplicación empresariales (ya que hacia estas están orientadas). Además son herramientas muy útiles y flexibles en las cuales hay mucha demanda y son base de grandes aplicaciones.

Otro aspecto a mencionar es el convivió con otros empleados y el trabajo en equipo, aspectos de mucha importancia en el desarrollo de sistemas, además de una serie de obligaciones y practicas de trabajo que no se conocían, como el uso de controladores de versiones para poder mantener estable un proyecto y controlar todos los cambios que cada persona realice, también la presentación de informes semanales para monitorear el desarrollo y el estado del proyecto y por ultimo la utilización de Project Server para controlar el trabajo realizado día con día.

Finalmente se experimento laborar en un ambiente totalmente diferente al de la universidad y bajo presiones totalmente distintas lo cual ayuda a adaptarse a un nuevo estilo de vida el cual nos acompañara de ahora en adelante.

Anexo #1



FIGURA. 25. DISEÑO DE LA BD MODULARIZADO