

4. Sistema Evaluador de CMM Niveles 2 y 3

4.1 Introducción

Una vez definido lo que significa y comprende la métrica de calidad de CMM, así como las características de un sistema experto, podemos empezar a establecer las etapas para la realización del sistema. Como sabemos, para la creación de cualquier sistema es necesario llevar a cabo un proceso como lo establece la Ingeniería de Software. Este proceso consta de algunas actividades a realizar para que el proceso siga su curso en forma normal sin que haya inconsistencias a lo largo de su concepción.

Primero definiremos el análisis del sistema, que establece el conocimiento que tenemos acerca de lo que el sistema debe hacer. Éste especifica también alcances, restricciones del mismo y los requerimientos como hardware, software, etc.

Después hablaremos del diseño del sistema que es donde a partir de la información obtenida en el análisis, haremos los esquemas que servirán para realizar la implementación del programa. Aquí hablaremos de las estructuras de datos, la arquitectura de software, el algoritmo a usar y las interfaces.

Por último hablaremos de la implementación que siguiendo el proceso de software, solo será poner en código la información y esquemas creados con anterioridad.

4.2 Análisis del Sistema

En el Capítulo 1 ya hablamos sobre la necesidad del sistema a desarrollar así como el objetivo principal y los objetivos específicos del mismo. A grandes rasgos podemos definir el sistema de la siguiente forma: el usuario del sistema debe contestar una serie de preguntas que ubicarán a su empresa en cierto nivel de CMM y en base a esto recibirá una retroalimentación.

Teniendo esto en mente podemos decir que el primer paso es obtener la información para crear los cuestionarios. Esta información está conformada por las especificaciones de los niveles dos y tres del modelo CMM, por textos referentes al tema y por casos prácticos en la industria. Se busca unir casos teóricos y prácticos porque muchas veces no siempre lo que se presenta en la teoría refleja la realidad en la práctica. Las especificaciones de CMM son una serie de lineamientos en donde se detalla *que* hay que hacer para alcanzar cierto nivel de madurez, mas no explica el *como*. Muchos textos posteriores tratan de llenar estos huecos de información de distintas formas tomando varios enfoques, pero ninguno logra recopilar esta información. Es por eso que estos cuestionarios además de presentar la información de CMM en forma práctica, hace también una recopilación de textos relacionados a la aplicación de la misma. Con esto podemos decir que contamos con una base de información robusta.

Una vez recopilada la información el siguiente paso es la creación de los cuestionarios del sistema. La forma más eficaz de estructurar estos cuestionarios es siguiendo las distintas prácticas clave (*key practices o KPA*) en las áreas de proceso, ya que esa es la forma en que está estructurada la información de CMM. De esta manera las personas que ya están familiarizadas con dicha métrica, entenderán el seguimiento del cuestionario y las personas que no conocían la métrica, podrán hacerlo a través de las preguntas.

Es necesario agregar que cada una de las prácticas clave (todas, desde Nivel 2 hasta Nivel 5) contiene dentro de sí lo que se conoce como características en común (*common features*) es decir, campos en los que se divide cada KPA [Humphrey 89]. Los primeros cuatro campos están enfocados a definir los pasos necesarios para el proceso de mejora dentro de la compañía, mientras el quinto señala las acciones necesarias para mantener este proceso y hacerlo en forma natural. Estos campos son:

- **Compromisos a Realizar.** Las prácticas clave de este tipo demuestran el intento de la organización por hacer que la misma sea una práctica común a la hora del desarrollo. Esta práctica generalmente es una política organizacional firmada por un alto directivo de la empresa.
- **Habilidades a Desarrollar.** Éstas prácticas aseguran que los recursos para llevar a cabo las demás prácticas (como dinero y tiempo) estén disponibles y que las condiciones como el entrenamiento, han sido satisfechas.
- **Métricas y Análisis.** Las prácticas aquí incluidas aseguran que el estado de cada práctica clave es conocido por la organización en forma cuantitativa.

- **Verificación de la Implementación.** En éstas prácticas normalmente se solicita que se realice una revisión regular hecha por la administración y generalmente también por el equipo de garantía de calidad de software. Con esto se busca que la práctica llevada a cabo tenga el efecto deseado y para saber si es necesario que la administración lleve a cabo alguna acción para resolver problemas de implementación.
- **Actividades a Realizar.** Las actividades a realizar sugieren acciones al equipo técnico o a los administradores para llevar a cabo el seguimiento o el entrenamiento dentro de cada área de proceso. Sin estas actividades, no habría nada que institucionalizar por parte de las demás características comunes.

Las prácticas clave que conforman el nivel 2, como se dijo en un capítulo anterior son: Requerimientos de Administración, Planeación del Proyecto de Software, Seguimiento del Proyecto de Software, Garantía de Calidad de Software, Administración de la Configuración de Software y Administración de Contratistas externos de Software. Cada uno de ellos, así como las preguntas asociadas a cada uno, tanto de nivel dos como de nivel tres estarán anexadas a este documento en el Apéndice A.

En los cuestionarios del nivel 2 buscaremos que mediante la aplicación de todas las prácticas, se logra el objetivo principal del mismo: que el proceso de software exitoso sea repetible. Esto significa que no sea resultado del trabajo extenuante por parte de solo algunas personas o simplemente cosa de suerte.

Aquí se observará un control de métricas y de proyectos anteriores. Generalmente los errores cometidos son observados y documentados para no volver a ser cometidos además de que se ha establecido un proceso sólido en base al cual se llevarán a cabo las mejoras.

Las prácticas clave que conforman el nivel 3 son: Enfoque de Proceso de la Organización, Definición de Procesos de la Organización, Programa de Entrenamiento, Administración de la Integración de Software, Ingeniería del Producto de Software, Coordinación entre Grupos y Revisiones por Pares.

En el nivel 3, conocido como el proceso definido, se busca que una vez que el proceso de software es repetible, las mejores prácticas de proyectos anteriores sean adoptadas para su uso por toda la organización. De esta forma, la organización debe encontrar la forma de comunicar y compartir las mejores prácticas. Éstas, según los lineamientos de CMM, se llevan a cabo gracias a dos prácticas a nivel organización: el proceso de desarrollo de software y el proceso estándar de software de la organización.

En el primero se establece la forma, los tiempos y los encargados del proceso de desarrollo. Este proceso se enfoca en el ciclo de vida y es llevado a cabo por el equipo de ingeniería de software. El segundo es el proceso ideal o modelo del cual parte la planeación de todos los proyectos de la compañía haciendo solo pequeñas modificaciones para adaptarlo a las condiciones específicas de cada uno.

Durante el desarrollo de los cuestionarios se podrá observar que ninguno de éstos es un proceso restrictivo, al contrario, ahora los proyectos tienen caminos explícitos para adaptar el conocimiento acumulado que ha sido encontrado en los proyectos anteriores.

Una vez establecida la forma en que se llevará a cabo la creación de los cuestionarios, podemos ocuparnos del sistema en sí. Los puntos más importantes son la forma de almacenamiento de información, la interfaz del sistema, los requerimientos del sistema y los alcances y las limitaciones del mismo.

Gracias a las características del producto como un sistema evaluador a través de cuestionarios, una de las partes principales del sistema es la necesidad de administración de sesiones para el usuario. Esto significa que cada usuario del sistema contará con una clave y una contraseña para poder guardar el estado de su sesión para siguientes ocasiones. Esto es importante debido a los siguientes factores:

- Un usuario debe poder realizar una parte de los cuestionarios y regresar en otra ocasión al sistema y continuar donde se había quedado ya que los cuestionarios abarcan mucha información y sería muy agotador contestarlos de una sola vez.
- La sesión debe permitir al usuario regresar a la pregunta en que se encontraba ya que si se inicia nuevamente la sesión mucho tiempo después de que fue creada, el usuario no recordará en que parte de la evaluación se encontraba.

- La retroalimentación debe de estar disponible para el usuario aún cuando éste ya haya terminado de resolver los cuestionarios para poder ser consultada cuando éste así lo desee.

Debido a ésta información podemos inferir que el sistema requiere de una forma de almacenamiento para poder cumplir las características mencionadas anteriormente. Sobre la forma de almacenamiento existen varias posibilidades y cada una tiene ventajas y desventajas. Las posibilidades son:

- **Uso de archivos.** El usar archivos tiene como ventaja que son una forma segura de almacenamiento ya que siempre su almacenamiento es en disco, además de que si se codifica no es peligroso que se conozca la información guardada en el mismo. Otra es que no requiere de grandes recursos para poder llevarse a cabo. Las desventajas de usar un archivo es que a fin de cuentas es una forma de almacenamiento local y así, si alguien inició su sesión en una computadora, solo podrá continuarla en ésta misma o si no tendrá que iniciar un nuevo proceso desde cualquier otra máquina.
- **Uso de Cookies.** Las cookies son pequeñas unidades de almacenamiento de información que son guardadas por un navegador dentro de la máquina donde se está abriendo el sistema. Tanto sus ventajas como sus desventajas son muy similares a las del uso de un archivo. La única diferencia ante éstos es su estructura y que un cookie puede tener un tiempo de vida asignado (o no), después del cual se destruirá a si mismo.
- **Uso de Base de Datos.** El uso de una base de datos tiene como principal ventaja que no es una forma de almacenamiento local, es decir, que cuando uno desee puede continuar

su sesión desde cualquier equipo sin ningún problema. De esta característica se deriva su mayor desventaja, que la base de datos necesita un servidor en *internet* para mantener la base de datos disponible para su acceso desde cualquier parte.

Una vez analizadas estas características concluí que la mejor opción de almacenamiento era hacer uso de la base de datos ya que permite gran flexibilidad en cuanto al uso de distintos equipos. Cabe agregar que hacer uso de cookies también es una opción recomendable a la hora de navegar a través de las páginas a usar ya que esto permite la multisesión, a diferencia de una base de datos que lo volvería más difícil.

Como se ha comentado en algunas ocasiones en este capítulo, la forma de realizar el sistema se pensó desde un principio a través de HTML. Esto se decidió así debido a que realizar un programa solamente en JAVA o algún otro lenguaje de programación hubiera significado tener que instalarlo cada vez que se utilizara, además que las interfaces pueden ser menos amigables usando swing en comparación con HTML. La amigabilidad es una característica importante en sistemas de este tipo. Otra de las razones por las que HTML es buena opción es su utilización a través de internet ya que esto permite que cualquier persona tenga acceso al sistema sin tener que instalarlo.

Siendo que se va a usar HTML para la interfaz del sistema, entonces la parte de bajo nivel del mismo queda reducida a las siguientes opciones de implementación:

- **Servlets.** Un servlet es un programa en Java que corre como parte de un servicio de red, normalmente un servidor de HTTP, que responde a servicios solicitados por el cliente.

El uso más común de un servlet es extender el servicio de un servidor para generar contenido dinámico en una página. Son la contraparte de los Applets, que también son hechos en Java aunque corren dentro de una máquina virtual del navegador de la parte del cliente. Los servlets son más flexibles que los scripts CGI y al ser escritos en Java, también son más portables [SUN 03a].

- **JavaServer Pages.** O comúnmente conocidos como JSP, cumplen la misma función básica de los servlets que es generar contenido dinámico en una página. La separación entre la interfaz y la codificación del contenido permite a los diseñadores de las páginas cambiar el resultado visual sin tener que volver a escribir el código. La diferencia entre los JSP y los servlets en forma sencilla es que los servlets son código en Java con HTML incrustado mientras los JSP son una base de HTML con código Java incrustado. Ambos pueden interactuar con páginas hechas 100% de HTML [SUN 03b].
- **JavaScript.** Inicialmente un script hecho por y para Netscape, JavaScript está íntimamente ligado a los navegadores de internet. Aun a pesar de su nombre, tiene poca relación con Java ya que su sintaxis es más parecida a C. Su funcionalidad es limitada a realizar formularios mejorados, manejos simples de bases de datos y mejoras a la navegación. Debido a que inicialmente fue pensado solo para trabajar en Netscape, es difícil escribir un código que trabaje de la misma forma en Netscape Navigator y en Internet Explorer [Netscape 98].

Haciendo un análisis de éstas opciones podemos decir que JavaScript tiene una ventaja sobre las otras opciones y ésta es que no necesita de un servidor para correr, ya que reside directamente en el código de HTML. La desventaja es que es más difícil que haga un

manejo eficiente de base de datos además que es un poco limitado y su código es muy encapsulado. Entre los servlets y los JSPs, la decisión depende de qué tanto se requiere la utilización de código Java o HTML en la presentación de la página. Debido a que el sistema va a usar bastante código, se decidió usar servlets en lugar de JSP.

Por último, hablando de los alcances y limitaciones del sistema, uno de los más importantes es el alcance de los cuestionarios. Como el nombre del sistema lo indica, éstos solo abarcan los niveles 2 y 3. Esto se decidió por dos razones primordiales, una es los clientes a los que está enfocado y otra debido a las características de la métrica de CMM.

La primera razón en otras palabras es que el cuestionario fue pensado para empresas dentro del país y teniendo en cuenta que hay muy pocas empresas nivel 2 y aún menos de nivel 3, en realidad podemos decir que con estos cuestionarios abarcamos casi todos los casos de empresas mexicanas. La segunda razón es que debido a las características de las prácticas de CMM, es muy difícil definir con un cuestionario las prácticas de nivel 4 y 5, ya que son más subjetivas y el cuestionario tiende a generalizar los procesos por lo que difícilmente abarcaría las prácticas reales de los niveles superiores.

4.3 Diseño del Sistema

Para el análisis ya se cuenta con los cuestionarios realizados, así como la retroalimentación que se presentará al usuario al final de los mismos. Acerca de los cuestionarios, tenemos un

promedio de 20 preguntas por área de proceso (o práctica clave), lo que significa si son 13 áreas de proceso, tenemos un poco más de 260 preguntas en total.

Debido a que las preguntas no se van a presentar necesariamente en orden gracias a las rutas que se van a trazar, lo más conveniente es almacenarlas en la base de datos. De esta forma, los servlets irán presentando en pantalla la pregunta correspondiente, según la construcción de la base de conocimientos.

Para la organización de la información todo se va a llevar a cabo en tablas de la base de datos. La descripción de éstas tablas así como su diseño se encuentran incluidas en el Apéndice D. Las tablas contempladas son:

- Tablas que contendrán las preguntas y la retroalimentación, es decir dos para las preguntas, una por cada nivel y dos para la retroalimentación. Cada pregunta tendrá un identificador único.
- Tablas que se encarguen del proceso de sesión. Este proceso se distribuirá en dos tablas distintas, una donde el usuario tendrá su clave y contraseña y otra, la de sesión, donde se guardará la información referente a las preguntas.

En cuanto a la sesión, se tiene contemplado que ésta guarde el lugar exacto donde se quedó el usuario la última vez que ingresó al sistema, su historial, los cuestionarios que ya contestó y las respuestas que dio a cada pregunta. Para realizar todas estas funciones es

necesario tener una estructura lógica que lo permita, además de un proceso de estos datos por parte de la máquina de inferencia, de la que se hablará mas adelante.

La estructura que se ha pensado es una cadena de números binarios en donde se represente una pregunta por cada número. Así si el usuario respondió que sí en la pregunta número diez, entonces habrá un número uno en la posición diez de la cadena. Cuando sea creada una nueva sesión, la cadena contendrá todas las respuestas en cero, es decir, todas las respuestas serán negativas. Esto será debido a que inicialmente el usuario no ha contestado nada, por lo que se encuentra en el nivel más bajo de la evaluación.

Con ésta cadena, ya hemos satisfecho el problema de mantener las respuestas que ha presentado el usuario a las preguntas. Ahora debemos resolver el problema de donde se quedó el usuario la última vez. Para saber esto lo más sencillo es mantener un campo en la base de datos que así lo señale, es decir un campo que contenga el índice de la última respuesta integrada a la cadena de sesión. Así cuando se cargue la sesión de nuevo, el sistema sabrá en que pregunta se quedó el usuario y podrá continuar desde ahí.

Por último para saber que cuestionarios ha contestado el usuario se necesita crear otro campo donde se guarde esta información. Este campo puede ser una cadena donde va a tener tres estados: uno donde indique que no se ha contestado ese cuestionario, otro donde indique que cuestionario se está resolviendo en ese momento y otro donde se vea que ya terminó de contestar un cuestionario.

Teniendo estas estructuras podremos saber cuando un usuario ya resolvió cierto cuestionario y no permitir que lo vuelva a resolver por accidente. Con esto evitamos que si un usuario inicia el proceso de evaluación y lo continúa varios meses después, no conteste de nuevo información que ya ingresó anteriormente.

Para la creación de la máquina de inferencia se estudiaron diversas formas que ya se explicaron en el Capítulo 3 sobre los sistemas expertos. Debido a la estructura de los cuestionarios lo más recomendable es realizarla a través de condicionales. Así, cuando alguna pregunta que determine un camino u otro dependiendo de la respuesta, simplemente se evalúa con la máquina de inferencia. Al solo haber dos posibles respuestas, no es necesario usar otra estructura para establecer esto.

Otro elemento de la máquina de inferencia es determinar si una pregunta es crítica o no para ubicar al usuario en uno de los niveles de CMM. Esto también se decidirá a la hora que el usuario responda la pregunta y este resultado sea evaluado según la importancia de la misma.

Ahora es necesario definir cuales serán los elementos que va a contener el sistema. La parte principal será el cuestionario. En esta sección el usuario podrá elegir primero el nivel que va a evaluar y el área de proceso. Una vez elegido esto se presentará al usuario la serie de preguntas a valorar. En el momento que quiera, el usuario podrá ver como va el avance de retroalimentación así como su ubicación frente al nivel 2 o 3.

También se permitirá al usuario volver a evaluar algún área de proceso contestada anteriormente. Esta opción será aparte para que, como se había dicho antes, no lo haga en forma inconsciente.

Por último el usuario tendrá la posibilidad de consultar un glosario de términos donde se explicarán los conceptos usados a lo largo de las preguntas del cuestionario. Este glosario tendrá un buscador para facilitar la búsqueda al usuario.

En el Apéndice D se encuentran los esquemas donde se explica la estructura interna del sistema y en los que se basa la implementación.

4.4 Implementación del Sistema

Para la implementación del sistema, se decidió crear un sistema integral en el cual se encontraran dos sistemas muy similares, el sistema Evaluador de CMM y el sistema Asesor de CMMI, tesis realizada por Juan Manuel Ulibarri Penichet. Debido a esta razón, a partir de la fase actual del sistema los procesos se van a realizar para ambos sistemas como un conjunto ya que la interfaz va a ser común entre los dos. [Ulibarri 04]

4.4.1 Base de Datos

Lo primero que hubo que decidir para realizar la implementación fue la base de datos a usar. Las opciones fueron:

- **Microsoft Access.** Ésta es una base de datos relacional, creada para correr sobre Microsoft Windows. Los datos son guardados en forma de tablas, que consisten de cierto número de registros que a su vez contienen cierto número de campos. Access permite al usuario crear formularios y reportes, además de facilitar el uso de ligas entre tablas que comparten un campo y de filtrar registros de acuerdo a cierto criterio de búsqueda.
- **Oracle DB.** Los drivers de JDBC de Oracle son compatibles 100% con Java. Basados en la especificación internacional de SQL, Oracle ha creado un sistema de base de datos robusto tanto para su uso como cliente como servidor, además de ser compatible con cualquier plataforma.
- **MySQL.** Esta es la base de datos más popular del mundo en cuanto a *opensource* se refiere. A pesar de ser gratuito, MySQL es confiable, robusto y muy rápido. Ideal para un manejo medio de datos, es muy amigable para acoplarse a los requerimientos de cada usuario.

Analizando las características de cada una de las bases de datos, podemos observar que Oracle y Access tienen un costo de licencia, mientras MySQL no. MySQL y Access no son multiplataforma como Oracle, aunque MySQL existe en versiones para todas las plataformas. En resumen, creemos que la mejor opción es MySQL ya que soporta una buena cantidad de datos, además que es gratuita.

Las bases de datos que se van a necesitar son básicamente cuatro: la de preguntas, la de respuestas, la de sesión y la de usuario. Las de preguntas y respuestas, como su nombre lo indica, contienen las preguntas a presentar en pantalla y la retroalimentación a éstas. Cada pregunta esta relacionada a un identificador único que relaciona ambas tablas entre sí, además de relacionarlas con la tabla de sesión.

La tabla de sesión contiene un identificador único que es la clave del usuario. Los demás campos que contiene esta tabla son la sesión, la pregunta actual y el estado de los KPA. Esta tabla depende completamente de la tabla de usuario ya que fueron creadas de forma que existe una indexación común entre ellas mediante su llave principal (la característica innodb). Esto lo que significa es que cuando se modifica la tabla de usuario, se modifica la tabla de sesión automáticamente.

La tabla de usuario contiene la clave del usuario y su contraseña. Como se mencionó anteriormente, la tabla de sesión depende de esta así que por ejemplo, si se borra un registro con la clave de usuario 1234, el manejador de base de datos busca ese mismo registro en la tabla de sesión y lo borra automáticamente. Lo que no sucede es que al dar de alta un registro en la tabla usuario éste se de de alta automáticamente en la tabla de sesión, pero esto se va a hacer a través del servlet. La relación entre las tablas se encuentra en el Apéndice D.

4.4.2 Sistema Experto

Para la implementación del sistema, primero que nada hubo que generar la máquina de inferencia que haría del producto un sistema experto. Se preguntó a varios especialistas en cuanto al tema y se hicieron consultas en diversos textos.

Después de hacer un análisis, se observó que la mejor opción de implementación para la máquina sería a través de condicionales. Un sistema CASE no hubiera sido una opción adecuada ya que desde la definición del mismo, es una especie de esquema prefabricado a partir del cual se hacen correcciones. Si se hubiera elegido esta opción, se hubiera necesitado hacer muchos cambios al mismo para que trabajara como lo requería el sistema.

La opción de los condicionales sobre otras se debió a que una vez establecida la forma de representación de las preguntas, el proceso se redujo a una representación de resultados binarios. La estructura de los condicionales (if-then-else) permite ir de un nodo a otro decidiendo que ruta tomar y definiendo la secuencia de las preguntas.

Así existen dos métodos que definen la siguiente pregunta a presentar en pantalla. El primero determina los límites de las áreas de proceso y el incremento que se debe hacer, de esta forma el método regresa el siguiente número o -1 si se ha llegado al final de esa área de proceso.

El segundo método es donde se encuentran las dependencias entre preguntas mediante condicionales. Este método recibe el número de la pregunta actual y busca su relación con

otras preguntas. El resultado de la pregunta llama al primer método y el número resultante es llamado recursivamente hasta encontrar un número que no dependa del resultado de la pregunta inicial.

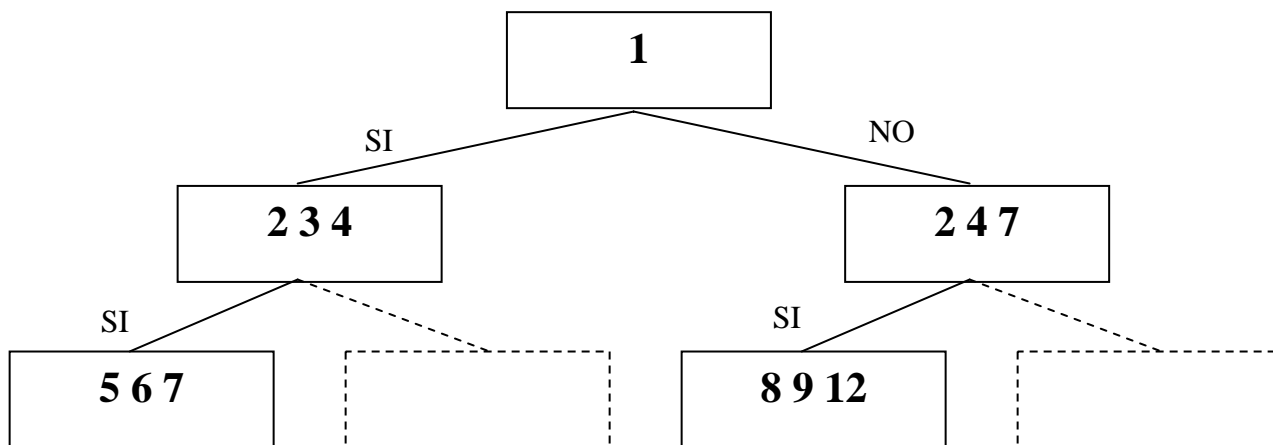


Figura 4.1

Como podemos observar en la figura 4.1, dependiendo de la respuesta en la pregunta uno, se presentan diferentes preguntas en la siguiente pantalla. Para el tercer nivel del árbol existen más posibilidades ya que son tres preguntas las que se presentan y por lo tanto 8 posibles respuestas por nodo.

4.4.3 Interfaz

La implementación de la interfaz fue una de las primeras cosas que se decidieron desde antes de iniciar el análisis del sistema. Ya se mencionó en las fases anteriores de desarrollo, pero se buscó realizar el sistema a través de HTML y servlets ya que esto le daba fácil acceso desde cualquier lugar a través de internet, permitía jugar más con la interfaz que por

ejemplo con java swing, aprovechaba el hecho de que casi toda la gente ha usado anteriormente un navegador, etc.

El uso de alguna forma de manipulación de datos era necesaria ya que se necesitaba alguien que funcionara como una capa de servicios entre el HTML y MySQL. Esta opción fueron los servlets ya que ofrecen la robustez de Java y no se meten directamente con el código de HTML a diferencia de los JSP.

El proceso general del sistema es primero que el usuario aporte datos al HTML, después los datos son enviados a algún servlet predefinido en la página cuando el usuario hace un submit. Aquí es donde se hace el procesamiento de datos una vez que estos llegan al servlet. Esto incluye accesos a la base de datos para actualizar la información, ya sea manipulada o no. Una vez terminado esto, se usa una función de Java que imprima en el siguiente HTML un buffer de información (ya estructurada también como HTML) y esto da inicio al ciclo de nuevo.

Un ejemplo de lo anterior es a la hora de presentar las preguntas en pantalla. Primero el usuario contesta las preguntas actuales. Cuando este manda la información al servlet, lo primero que se hace es direccionar la información a variables de los servlets. Luego el sistema verifica mediante las cookies que usuario es el que envía la información y accesa a la sesión de este. Mediante la máquina de inferencia y otras clases que sirven de apoyo se generan las preguntas siguientes, se busca su contenido en la tabla de preguntas y se van agregando al buffer, ya con formato HTML. Una vez terminado esto, se actualiza la base de

datos de sesión y por último se envía el buffer para la creación de la nueva página de preguntas.

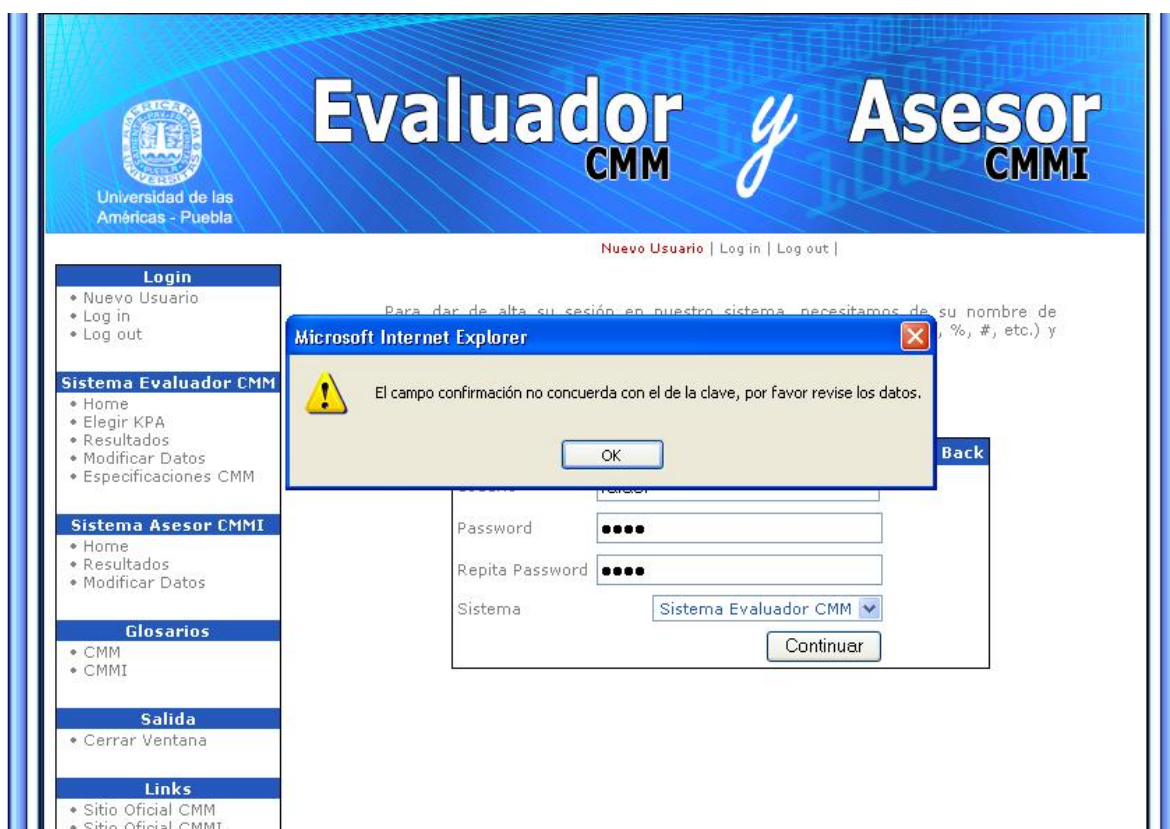


Figura 4.2

Hay ciertas validaciones entre campos y comparaciones que se hacen a través de Javascript, como se puede ver en la figura 4.2. Esto fue muy útil gracias a que para llevar a cabo estos procesos no es necesario interactuar con la base de datos por lo que en varios casos, hacer uso de los scripts ahorraron la creación de servlets. Otro punto a favor de los scripts es que aunque los accesos a la base de datos son muy rápidos, si validamos la información a través de ellos no se necesita ningún intermediario por lo que es aún más rápida la manipulación de información.

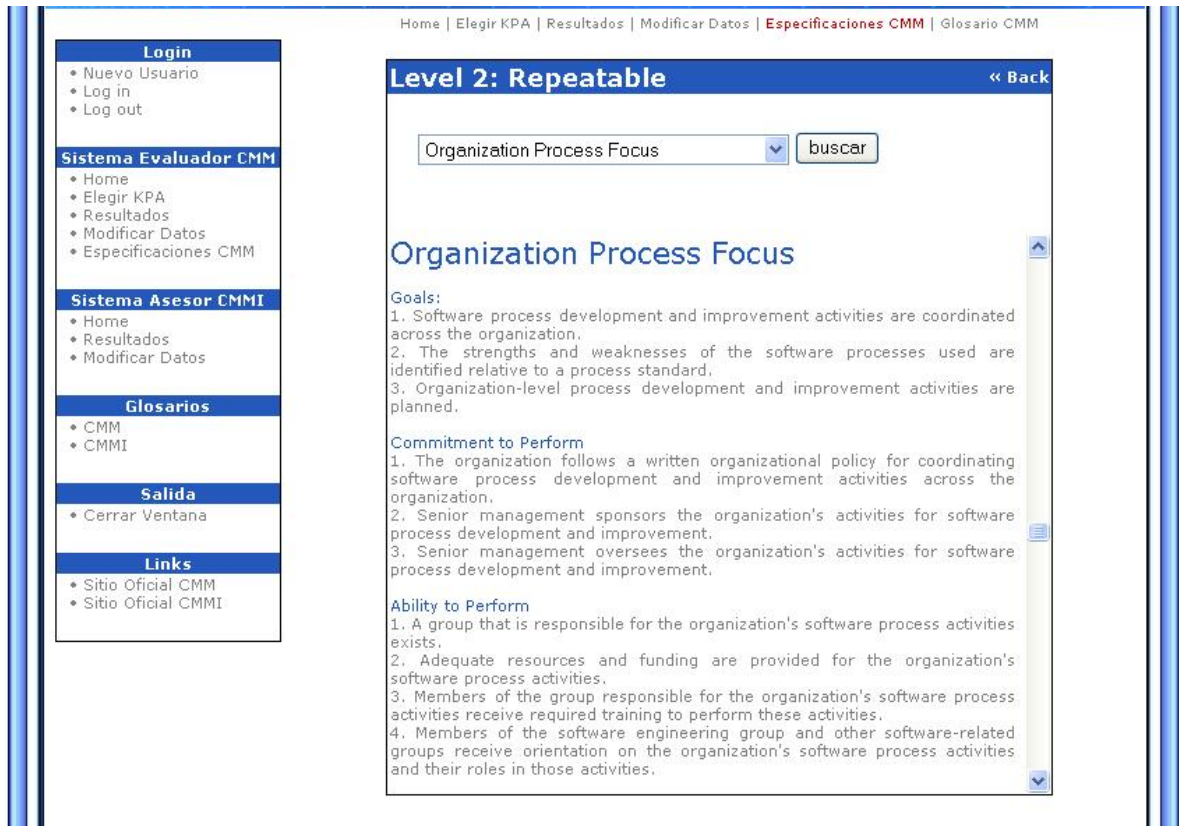


Figura 4.3

Para la creación de las páginas se usó un editor de HTML. En las páginas debido al tipo de sistema que se va a realizar, todo se hizo a través de formularios y en algunos casos frames e iframes, como se ve en la figura 4.3. Para la manipulación de imágenes se usó el Adobe Photoshop y para los marcos y cosas que son iguales en todas las páginas se hicieron referencias a un stylesheet, que permite hacer cambios y que éstos se reflejen en todas las páginas.

Para poder mantener la comunicación entre las páginas de HTML y los servlets fue necesario montar un servidor que soportara el uso de servlets y de HTML. Se buscaron distintas opciones pero en realidad, Yakarta Tomcat fue pensado desde un principio como el servidor a ser usado. Esto es debido a que es el servidor gratuito más robusto que hay

actualmente. Existen otros servidores más seguros pero su costo es muy elevado y eso saldría de las expectativas del sistema.

Para compilar los servlets, se usó una opción de compilación que proporciona el Tomcat, que añade las librerías que el servidor contiene a las opciones de compilación. El compilador obviamente necesita de un ambiente JDK instalado.

4.4.4 Sesión

Para implementar la sesión se buscó una forma de representación que tuviera varias características como por ejemplo que fuese sencilla de guardar, fácil de modificar y que no contuviera ambigüedad en sus datos.

Como se mencionó en la sección anterior, al solo haber dos tipos de respuestas los resultados pueden ser binarios (verdadero y falso). Esto es muy conveniente ya que su representación es más fácil. Otro factor a considerar es la extensión de los cuestionarios. Si juntamos las preguntas de las trece áreas de proceso, encontramos que asciende a un total de 268. Con esto encontramos que se necesita una secuencia de 268 caracteres que tengan dos representaciones distintas.

Después de analizar la situación concluimos que lo mejor era usar una cadena de caracteres que aceptara valores de 0 y 1. Se decidieron caracteres en lugar de dígitos ya que primero, no queremos que se manipulen los datos entre sí, segundo es que puede haber varios ceros

al principio y si se manejara como dígito esto no sería posible (ceros a la izquierda son nulos) y tercero y más importante es que un dígito de 268 posiciones ocuparía demasiada memoria y sería imposible de representar.

Otro problema con el que nos topamos es que el campo varchar que representa las cadenas en la base de datos, no alcanza más de 255 caracteres por lo que sería también insuficiente. Debido a esto tuvimos que usar el tipo de dato blob que si acepta este número de caracteres, además de ser compatible con la clase String de Java.

Los demás campos que se encuentran en la tabla de sesión, que son la última pregunta y el estado de los KPA, sirven para determinar en forma exacta el estado en el que se quedó la sesión del usuario. Esto fue necesario ya que si solo tuviéramos la cadena de caracteres del campo sesión, solamente tendríamos una cadena de ceros y unos pero no sabríamos que preguntas ya han sido respondidas (ya que el usuario puede elegir que área de proceso desea evaluar, sin que esto signifique que lo tenga que hacer en secuencia) o en que pregunta se quedó para que a la siguiente continúe contestando esta.

Para determinar esto, cada vez que el usuario contesta una pregunta, se guarda el identificador de esta pregunta también. De esta forma podemos saber cual fue la última pregunta contestada y presentar la pregunta exacta en que el usuario se quedó. El estado de los KPA se determina con otra cadena de caracteres. Existe un carácter por cada KPA o área de proceso. Cada carácter tiene 3 posibles estados. El primero es no contestado, el segundo terminado y el último en proceso. De esta forma sabemos exactamente cuantas preguntas faltan y que áreas de proceso ya han sido terminadas.

4.4.5 Cookies

Las cookies, también ya mencionadas en la parte de análisis, son una parte importante del funcionamiento del sistema. El uso de cookies se debió a que la forma de implementar este sistema es a través de páginas de HTML, cuya relación entre sí son los servlets. Aún a pesar de esto, cada página de HTML es independiente de otra por lo que si en una página sabemos quien es el usuario actual, a la siguiente no tenemos ni idea.

Una forma para saber esto es enviando la clave del usuario a través de los servlets a la base de datos, cosa poco útil ya que habría mucho tráfico innecesario en la base de datos. Otras formas incluyen el uso del objeto sesión dentro del servlet, el envío de variables a través del url, etc.

La forma que creemos es la más adecuada es el uso de los cookies. Cuando un cookie es creado, se asigna un valor a un objeto cookie relacionado con un nombre y éste se mantiene en la computadora del usuario. De ésta forma cuando la página solicita la información que tiene cierto cookie, lo que hace es extraer el arreglo de cookies que tiene el usuario en su computadora, busca el cookie a través de su nombre, obtiene la información y si necesita actualizarla, sobrescribe en el cookie anterior.

Otro detalle importante de los cookies es su tiempo de vida. En este caso, el cookie se mantiene con vida hasta que el usuario cierra su navegador. De esta forma no se guarda basura en la computadora del usuario.

El cookie permite que el sistema sea multiusuario ya que como el HTML depende del usuario en forma local y de la información que este guarda de sí mismo, entonces varios usuarios pueden acceder a la base de datos con distintas claves de acceso.

4.4.6 Resultados de los cuestionarios

Los resultados de los cuestionarios son la parte que le regresa algo al cliente, por lo que sin la correcta implementación de estos, el sistema en general no tendría caso. Estos resultados pueden ser consultados en cualquier momento, una vez que el usuario ha empezado a contestar los cuestionarios. Para llevar a cabo este proceso existe un servlet que administra varias clases que sirven para obtener esta información.

Este servlet lo que hace es que una vez identificado el usuario, obtiene su sesión hasta ese momento. Esta información la envía a una clase que se encarga de crear una cadena de caracteres de resultados a partir de la de preguntas. Con estos resultados, se busca en la base de datos de respuestas las recomendaciones a mostrar en pantalla. Cabe mencionar que esta cadena generada no se guarda en la base de datos, es decir, se genera cada vez que se solicitan resultados.

Home | Elegir KPA | **Resultados** | Modificar Datos | Especificaciones CMM | Glosario CMM

El resultado de las preguntas anteriores es:

Respuestas « Back

Le faltan 110 prácticas para alcanzar el Nivel 2 de CMM

Le faltan 101 prácticas para alcanzar el Nivel 3 de CMM

Resultados de Requerimientos de Administración:

Debe existir un equipo de ingeniería de software que lleve a la práctica el ciclo de vida del proyecto, que administre los requerimientos del proyecto y que siga el plan de desarrollo de software. Es el que tiene el contacto principal con la parte de software del proyecto.

Resultados de Planeación del Proyecto de Software:

Un equipo encargado de los estimados de software es necesario ya que solo basados en los estimados suficientes es posible generar un software de datos

Figura 4.4

Además de las recomendaciones, en los resultados se muestra que tan cerca está un usuario de alcanzar un nivel de CMM (figura 4.4). Para realizar este cálculo se manda la cadena de respuestas a una clase que contiene la relación de preguntas críticas y no críticas. Esto significa que si una pregunta es crítica, la respuesta que el usuario haya dado a esa pregunta influye en el resultado final, mientras que si es no crítica solamente influye en las recomendaciones más no en el resultado.

El resultado de la clase es otra cadena de caracteres que igual que la de respuestas no se guarda en la base de datos. Esta se usa para obtener una cantidad que significa que tan lejos está ese usuario de alcanzar el siguiente nivel.

4.4.7 Modificar Datos

Modificar datos significa que un usuario puede cambiar la respuesta que había asignado anteriormente a una pregunta solamente cuando ha terminado de contestar algún área de proceso. Lo interesante sobre la implementación de esta opción es que hace una manipulación de información distinta las anteriores.

Primero que nada cuando se presentan preguntas a modificar, éstas muestran los resultados que se les había asignado anteriormente. Esto significa que si el usuario había respondido que sí a la pregunta 6 de Requerimientos de Administración, cuando esta pregunta se presente en pantalla para modificación presentará como respuesta predeterminada “sí”. Otra cosa es que modifica los resultados que no se presentaron en las rutas anteriores. Esto es más claro en la figura 4.5.

Aquí lo que se muestra es que inicialmente un usuario respondió que sí a la pregunta 1 y esto hizo que tomara la ruta normal, contestando todas las preguntas. Una vez que termina esta área de proceso, decide modificar sus datos. Si ahora en la pregunta 1 la responde negativamente, esto significa que muchas de las preguntas que ya había contestado anteriormente, ahora deben ser contestadas en forma negativa automáticamente. Los números rojos muestran las respuestas anteriores mientras que los azules son los cambios hechos durante las modificaciones.

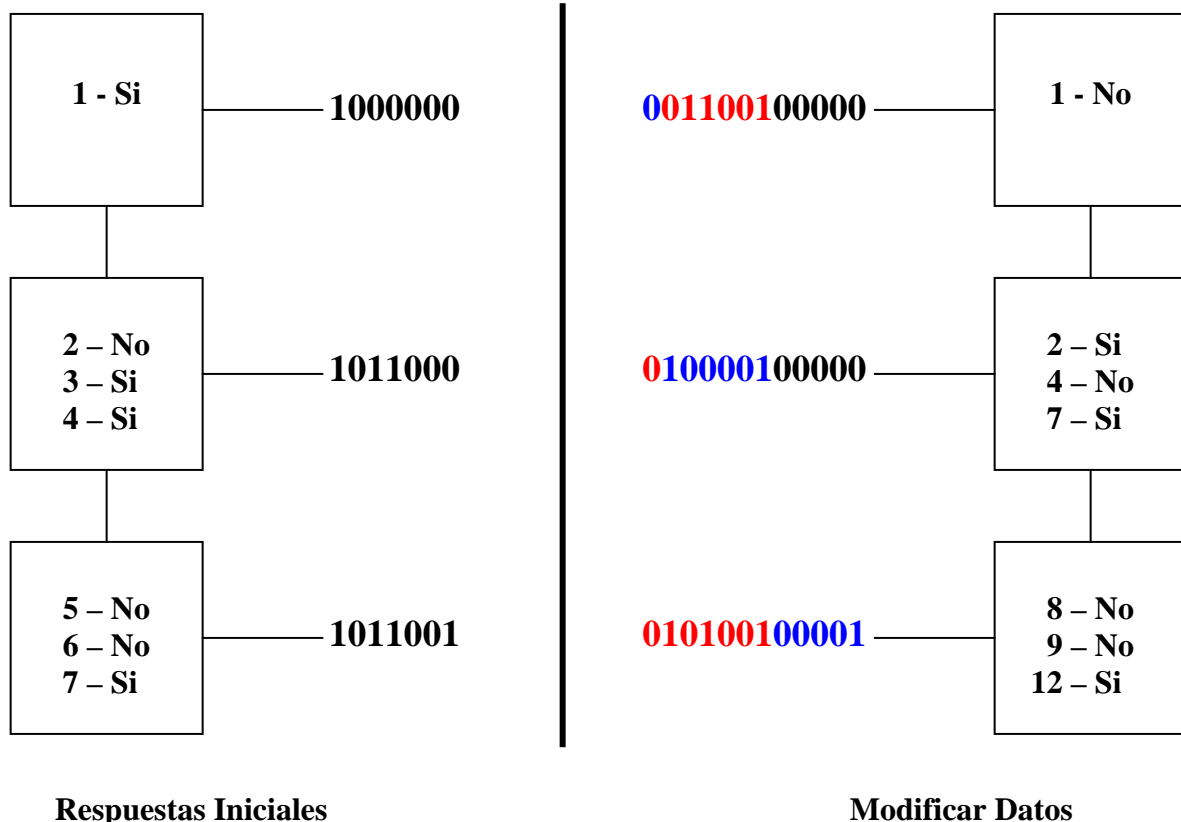


Figura 4.5

4.4.8 Otras Implementaciones

De los demás detalles de implementación, solo cabe mencionar que tanto el servidor de base de datos como el servidor de internet estarán corriendo en una cuenta de la universidad, usada especialmente para este propósito. La cuenta que se usa forma parte de los proyectos de Ingeniería de Software del departamento y su nombre es msqroot. Esta decisión fue hecha principalmente para que este disponible todo el tiempo que sea necesario.