

Universidad de las Ciencias Informáticas

Facultad 15



**Título: Sistema informático para la identificación
de las competencias asociadas a los
profesiogramas o perfiles de cargo.**

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas.

Autor(es): Maray Corrales Guerrero.
Yoiner Barallobre Reyes.

Tutor: Ing. Janet Rodríguez Febles.

Junio de 2010

DECLARACIÓN DE AUTORÍA

Declaramos ser los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año_____.

Yoiner Barallobre Reyes (Autor)

Maray Corrales Guerrero (Autor)

Janet Rodríguez Febles (Tutor)

AGRADECIMIENTOS

Agradecemos especialmente a Janet Rodríguez Febles, como tutora de la tesis, por toda la ayuda y el apoyo brindado, tanto humano como profesional.

Un especial agradecimiento a Uriser Barbón Suárez, gracias por todo el tiempo y el apoyo brindado. A otras personas como Javier Ruiz Durán, José Manuel Barceló y Maikel Yulier Sañudo, por su granito de arena puesto en el desarrollo de este trabajo.

Agradecer a la Universidad de las Ciencias Informáticas (UCI) y profesores de toda la vida, por todo el conocimiento brindado, sin el cual, no seríamos hoy lo que somos.

A nuestros familiares, amigos y compañeros de estudio, tanto los que están aquí, como los no están presentes.

DEDICATORIA

A nuestros familiares y amigos de toda la vida, tanto los viejos como los que conocimos en el transcurso de estos cinco años en la UCI. Para ustedes es este trabajo.

RESUMEN

En nuestro país, con el objetivo de aumentar la producción y la calidad de los servicios de las empresas y organizaciones, se ha decidido adoptar un Sistema de Gestión Integrada de Capital Humano (SGICH), para ello, la norma cubana propone un modelo formado por 9 procesos fundamentales, cuyo núcleo lo constituyen las competencias laborales. En el mismo se define el Comité de Competencias como el grupo responsable para identificar y proponer a la alta dirección, las competencias claves o distintivas de la organización.

Un elemento fundamental para la elaboración de los perfiles de cargo es la identificación de las competencias, con este propósito el sistema de identificación de las competencias asociadas a los profesiogramas o perfiles de cargo (SISCOM) está concebido como una aplicación web, que automatiza el método Delphi por Rondas, para obtener las competencias que componen un perfil de cargo mediante el criterio de expertos.

SISCOM es una aplicación desarrollada en el lenguaje PHP, guiado por la metodología ágil SXP, se usó el CodeIgniter como framework de desarrollo, se utilizó PostgreSQL como servidor gestor de base de datos y como servidor web Apache, entre otras herramientas.

Palabras Claves: Profesiogramas, Perfiles de cargos, Competencias laborales, Norma Cubana.

TABLA DE CONTENIDOS

| | |
|--|------------|
| DECLARACIÓN DE AUTORÍA..... | I |
| AGRADECIMIENTOS..... | II |
| DEDICATORIA..... | III |
| RESUMEN..... | IV |
| INTRODUCCIÓN..... | 1 |
| CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA..... | 5 |
| 1.1. Introducción..... | 5 |
| 1.2. La gestión de Capital Humano por Competencias..... | 5 |
| 1.2.1. Conceptualización de las competencias..... | 5 |
| 1.2.2. Clasificación de las competencias..... | 6 |
| 1.2.3. Perfiles de Cargo..... | 7 |
| 1.2.3.1. Perfiles por Competencias..... | 7 |
| 1.3. Métodos para la generación de los perfiles por competencias..... | 8 |
| 1.4. Sistemas informáticos para la gestión del capital humano en el mundo..... | 13 |
| 1.5. Tendencias actuales para el desarrollo de software..... | 15 |
| 1.6. Metodologías de desarrollo de software..... | 16 |
| 1.7. Lenguajes de modelado..... | 22 |
| 1.8. Herramientas CASE (Computer Aided Software Engineering)..... | 25 |
| 1.9. Lenguajes de programación..... | 28 |
| 1.10. Framework de desarrollo..... | 29 |
| 1.11. Entorno de desarrollo integrado (IDE)..... | 31 |

| | |
|--|-----------|
| 1.12. Sistemas gestores de bases de datos (SGBD)..... | 33 |
| 1.13. Servidores web..... | 34 |
| 1.14. Enfoque de la arquitectura de software para las metodologías ágiles..... | 35 |
| 1.15. Patrones de arquitectura..... | 36 |
| 1.15.1. Patrón Modelo – Vista – Controlador..... | 36 |
| 1.16. Patrones de diseño..... | 37 |
| 1.16.1. Patrones GRASP | 37 |
| 1.16.2. Patrones GOF..... | 38 |
| 1.17. Conclusiones..... | 39 |
| CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA | 40 |
| 2.1. Introducción..... | 40 |
| 2.2. Concepción del sistema..... | 40 |
| 2.2.1. Descripción del producto..... | 40 |
| 2.2.2. Planificación del proyecto por roles..... | 41 |
| 2.3. Modelo de Historia de Usuario del negocio..... | 41 |
| 2.3.1. Actores del negocio..... | 41 |
| 2.3.2. Trabajadores del negocio..... | 42 |
| 2.3.3. Diagrama de procesos del negocio..... | 42 |
| 2.4. Lista de Reserva del Producto (LRP)..... | 42 |
| 2.5. Historia de Usuario..... | 43 |
| 2.5.1. Tareas de ingeniería | 43 |
| 2.6. Modelo de datos..... | 43 |
| 2.7. Plan de release | 43 |

| | |
|--|-----------|
| 2.8. Conclusiones..... | 44 |
| CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA..... | 45 |
| 3.1. Introducción..... | 45 |
| 3.2. Pruebas de unidad..... | 45 |
| 3.2.1. Cálculo de la complejidad ciclomática..... | 46 |
| 3.3. Pruebas de aceptación..... | 50 |
| 3.4. Pruebas de rendimiento..... | 50 |
| 3.4.1. Pruebas de Stress..... | 50 |
| 3.5. Conclusiones..... | 52 |
| CONCLUSIONES..... | 54 |
| RECOMENDACIONES..... | 55 |
| BIBLIOGRAFÍA..... | 56 |
| ANEXOS..... | 58 |
| Anexo #1: Prototipos de interfaz de usuarios..... | 58 |
| GLOSARIO..... | 63 |

INTRODUCCIÓN

La administración de personal, surgió como resultado de la necesidad de la sociedad de trabajar en grupo. En la actualidad se le ha dado un nuevo enfoque a este tema, dando paso a la formación de la gestión de capital humano. Este nuevo concepto asume un gran cúmulo de actividades relacionadas con la organización laboral en su interacción con las personas, convirtiéndose en la esencia de la dirección o gestión empresarial.

La gestión por competencia es una herramienta que se le ha vinculado a la gestión de capital humano, integrándose a varios de sus procesos fundamentales, como son: la selección e integración, la capacitación, el desarrollo y evaluación de desempeño. Definir cada una de ellas y la importancia que poseen para el desempeño en un cargo, permitirá una mejor interpretación de las mismas en la toma de decisiones, logrando una mejor armonía entre la productividad y los conocimientos.

Hacia 1970 se comienza a ver con mayor auge el tema de las competencias laborales. Las grandes empresas y organizaciones hacían una selección del personal mediante pruebas de inteligencia y exámenes donde se medía los conocimientos que poseía un individuo para desempeñar una tarea determinada, aquellas personas que obtuvieran mayor puntuación eran consideradas las más apropiadas para conseguir el puesto y por lo tanto empleadas.

En la década de los 80 Inglaterra crea el Consejo Nacional para las Calificaciones Profesionales, cuyo propósito era establecer y certificar el sistema de calificaciones profesionales de ese país. A partir de entonces en varios países de Europa al igual que en Estados Unidos, Canadá y Australia se comienzan a crear sistemas muy parecidos. Un poco más tarde en la década de los 90 en México se crea un sistema normalizado de competencia laboral.

Cuba no se ha mantenido al margen de los cambios que han ocurrido en el mundo en cuanto a este tema. Entre los logros de la revolución cubana se encuentra el nivel alcanzado en la preparación (calificación y competencia) de los recursos humanos, aunque todavía existen problemas como:

- Falta de procedimientos para la identificación y validación de las competencias.
- Utilización de métodos para la creación de perfiles de cargo que no contemplan la asignación de pesos a las competencias.
- Falta de automatización de procesos vinculados a esta actividad.

Con el propósito de erradicar estos problemas, se concibió un Sistema de Gestión Integrada de Capital Humano (SGICH) para ser adoptado en las empresas cubanas. Este modelo está formado por 9 procesos fundamentales, cuyo núcleo lo constituyen las competencias laborales. En el mismo se define el Comité de Competencias como el grupo responsable para identificar y proponer a la alta dirección, las competencias claves o distintivas de la organización.

La creación de los perfiles de cargo por competencias en las empresas, es un elemento fundamental para la implantación del SGICH, esta actividad demanda tiempo y en ocasiones se requiere la presencia de especialistas o expertos que pudieran encontrarse en lugares distantes.

Debido a la situación expuesta anteriormente se plantea el siguiente **problema a resolver**: ¿Cómo facilitar el proceso de identificación de las competencias asociadas a los perfiles de cargos, mediante la elaboración de un sistema informático?

El **objeto de estudio** del presente trabajo se enmarca en los sistemas informáticos para la gestión del capital humano por competencias y el **campo de acción** lo constituye el sistema informático para la identificación de las competencias asociadas a los profesiogramas.

El **objetivo general** de la investigación es: Desarrollar un sistema informático para la identificación de las competencias asociadas a los profesiogramas o perfiles de cargo.

Del objetivo general se derivan los siguientes **objetivos específicos**:

- Elaborar el marco teórico sobre las tendencias de la gestión del capital humano por competencias, los sistemas informáticos, las tecnologías, metodología y herramientas.
- Realizar los artefactos correspondientes a las distintas fases de la metodología a utilizar.
- Implementar el sistema informático para la identificación de las competencias asociadas a los profesiogramas o perfiles de cargo.
- Probar el sistema informático para la identificación de las competencias asociadas a los profesiogramas o perfiles de cargo.

Idea a defender: Si se implementa el sistema informático para la identificación de las competencias asociadas a los profesiogramas o perfiles de cargo, se facilitará, el proceso de identificación de las

competencias asociadas a los profesiogramas o perfiles de cargos, a partir de cuestionarios aplicados a expertos.

Los **métodos utilizados** para la investigación son.

Métodos Teóricos:

- **Histórico lógico:** Utilizado para analizar las principales tendencias de las tecnologías, metodología, herramientas y sistemas informáticos vinculados a la gestión de los recursos humanos por competencias.
- **Sistémico:** Utilizado para estudiar el objeto mediante la determinación de sus componentes, teniendo presente las relaciones que se establecen entre ellos.

El documento se encuentra estructurado en tres capítulos:

CAPITULO I: “FUNDAMENTACIÓN TEÓRICA”. Brinda al lector el estado del arte relacionado con los sistemas informáticos vinculados a la formación del personal. Justificándose las tendencias, tecnologías, metodologías y herramientas que fueron utilizadas para el desarrollo del mismo.

CAPITULO II: “PROPUESTA DEL SISTEMA”. Muestra las principales características del sistema. En este capítulo también se lleva a cabo el desarrollo ágil del sistema informático para la identificación de las competencias asociadas a los profesiogramas o perfiles de cargo, explicando toda la dinámica del proyecto mediante los artefactos generados por la metodología utilizada.

CAPITULO III: “VALIDACIÓN DE LA PROPUESTA”. Este capítulo muestra los casos de pruebas o test de aceptación a las que fue sometida la aplicación en cada una de las iteraciones. Se exponen los resultados obtenidos al aplicarle métricas de calidad al sistema y se muestran las funcionalidades alcanzadas en el período de desarrollo ágil.

Los fundamentos teóricos de este trabajo, tienen como antecedente el estudio realizado por Armando Cuesta¹ el que sirve de apoyo para entender el concepto de competencias laborales desde un enfoque estructural y dinámico. Se utilizó también para la conceptualización de las competencias el trabajo

¹ Psicólogo graduado en la Universidad de La Habana (1973), Master en Organización del Trabajo (1979) y Doctor en Ciencias Económicas (1984). Desde 1976, Profesor de la Facultad de Ingeniería Industrial del Instituto Superior Politécnico José Antonio Echeverría (ISPJAE)

valorativo sobre este tema realizado por Maura González², permitiendo profundizar en el punto de vista psicológico y persono-lógico de la definición de competencia. Los aportes legales de la concepción del sistema informático para la identificación de las competencias asociadas a los profesiogramas o perfiles de cargo, fueron tomados del modelo cubano para la adopción de un SGICH en las empresas, vigente en la Norma Cubana 3000-3002. Para la clasificación de las competencias se tuvo en cuenta las investigaciones realizadas por Fernando Vargas³ y Bunk⁴.

² Dra. en Psicología Viviana *González Maura*

³ **Consultor Cinterfor/OIT** fvargas@cinterfor.org.uy

OIT. Oficina Regional para América Latina. Estrategias de productividad y competitividad. Recursos Humanos y empleo en los 90.

⁴ Destacado teórico Norteamericano.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1. Introducción.

En este capítulo se aborda el concepto de la gestión del capital humano, haciéndose una aproximación al tema de las competencias. Se analizan las tendencias actuales de las tecnologías para el desarrollo de software además, se ofrece una panorámica, de las herramientas que se utilizarán en la implementación del sistema informático para la identificación de las competencias asociadas a los profesiogramas o perfiles de cargo, conjuntamente a la arquitectura que soporta el mismo y las metodologías a usar en el ciclo de vida del proyecto.

1.2. La gestión de Capital Humano por Competencias.

“Actividades coordinadas para dirigir y controlar una organización con un enfoque basado en las competencias laborales y la capacidad de aprendizaje de los trabajadores. Su objetivo es una organización de calidad y la disposición del colectivo integrado para el logro de los objetivos de la organización.” (Norma_Cubana, 2007)

Utilizando como referencia la norma cubana para la adopción del SGICH, en la que se basa esta investigación, con el objetivo de realizar un acercamiento a los principales conceptos necesarios para entender el funcionamiento del sistema informático propuesto, se puede concluir que la gestión de capital humano por competencia consiste en el conjunto de actividades que se realizan dentro de una organización o empresa para determinar el conjunto de competencias de sus trabajadores, garantizando así la calidad adecuada de cada uno de sus integrantes logrando el aumento de la productividad y el buen funcionamiento de la misma.

1.2.1. Conceptualización de las competencias

Existen diversos conceptos relacionados con el término competencia según diferentes autores; a continuación se muestran algunos de estos.

Maura González conceptualiza el término competencia como: “configuración psicológica compleja que integra en su estructura y funcionamiento formaciones, motivacionales, cognitivas y recursos personológicos que se manifiestan en la calidad de la actuación profesional del sujeto, y que garantizan un desempeño profesional responsable y eficiente”. (González, 2002)

El Dr. Armando Cuesta Santos define competencias como: “características subyacentes en las personas, asociadas a la experiencia, que como tendencia están causalmente relacionadas con actuaciones exitosas en un puesto de trabajo, contextualizado en determinada cultura organizacional”. (Cuestas, 2005)

La Norma Cubana para la adopción de un SGICH reconoce como concepto de competencia a: “ El Conjunto sinérgico de conocimientos, habilidades, experiencias, sentimientos, actitudes, motivaciones, características personales y valores, basado en la idoneidad demostrada, asociado a un desempeño superior del trabajador y la organización (...)” (Norma_Cubana, 2007)

En lo visto anteriormente se puede observar que cada experto tiene opinión diferente sobre el término competencia, aunque todos coinciden en que son características propias de los individuos que le permiten desempeñarse satisfactoriamente en un puesto. De acuerdo al estudio realizado podemos llegar a un concepto de competencia que se ajuste a los objetivos de la investigación y podemos decir que las competencias son las características subyacentes de cada persona, que están relacionadas con una correcta actuación en su puesto de trabajo y que pueden basarse en la motivación, en los rasgos de carácter, en el concepto de sí mismo, en actitudes o valores, en una variedad de conocimientos, capacidades o conductas. En definitiva, se trata de cualquier característica individual que se pueda medir de modo fiable y cuya relación con la actuación en el puesto de trabajo sea demostrable.

1.2.2. Clasificación de las competencias.

Las clasificaciones de las Competencias, ha sido igualmente estudiada y definida según el criterio de varios expertos. A continuación se muestra una selección de las definiciones más comunes.

Fernando Vargas Z. distingue tres definiciones de competencia laboral.

- Básica: que incluye aquellos comportamientos que se adquieren en la educación básica como (Matemática, Física, etc.).
- Genérica: La que abarca los desempeños comunes a diferentes ocupaciones.
- Específica: lo define como comportamientos asociados a los conocimientos técnicos de una ocupación particular. (Vargas, 2004)

G. P. Bunk plantea que existen cuatro clasificaciones.

- Técnica: Es dominio experto de tareas, conocimiento y destrezas en el trabajo.
- Metodológica: Se define a través de la capacidad de reacción a emergencias y soluciones.
- Social: la define como la colaboración proactiva con otros en su grupo y la comunicación efectiva.
- Participativa: Se establece de acuerdo a la relación con la organización, toma de decisiones y la disposición a aceptar nuevas responsabilidades. (G.P.Bunk, 1994)

El sistema informático para la identificación de las competencias asociadas a los profesiogramas o perfiles de cargo, utilizando como base las clasificaciones expuestas anteriormente, permitirá al responsable del comité de competencia seleccionar que tipo de competencias desea identificar.

1.2.3. Perfiles de Cargo.

Un perfil de cargo es un documento que enmarca o define un conjunto de requisitos que se estima debe tener una persona para desempeñarse adecuadamente en un cargo determinado. Permite conocer y explicar las características que deben tener tanto los candidatos que postulan a un cargo, como aquellos que se encuentran desempeñando una labor determinada. Con este documento se optimizan las aptitudes de cada uno de los funcionarios que desarrollan las labores, capacitando a aquellos que presenten dificultades con algunos de las exigencias estipuladas, así quedarán en el puesto de trabajo solo aquellas personas que cumplan con las condiciones que requiere la empresa. (Gobierno de Chile, 2006)

El sistema informático para la identificación de las competencias asociadas a los profesiogramas o perfiles de cargo, dará soporte a la confección de un tipo específico de perfiles de cargos, que son los perfiles por competencia, en ellos quedan recogidas las competencias idóneas que debe tener un individuo para el desempeño exitoso de un puesto en una empresa u organización.

1.2.3.1. Perfiles por Competencias.

El perfil por competencias se define como un listado de competencias esenciales para el desarrollo de un puesto, en términos de; conocimientos, habilidades y conductas observables. Los actuales perfiles por competencias se centran más bien en las responsabilidades y actuaciones que deberá tener el

individuo para desempeñar exitosamente su trabajo, bajo una situación contextualizada. (Cuestas, 1999).

Utilizando como base la investigación realizada se concluye que: El profesiograma o perfil por competencias, incluye la determinación de qué personas y en qué medida, de entre varios candidatos, cumple con las características que corresponden a los requerimientos de un puesto de trabajo, indica qué competencias y en qué grado son importantes para desarrollar adecuadamente las funciones y tareas propias del puesto.

1.3. Métodos para la generación de los perfiles por competencias.

Existen distintos métodos para recopilar la información necesaria para llevar a cabo la confección de un perfil, los más utilizados son:

Entrevistas.

Se entrevista a personas que ocupen o hayan ocupado el cargo de forma exitosa, a fin de obtener información relevante respecto del contexto en que se desempeña, objetivos, funciones y actividades. En el caso especial de las entrevistas basadas en competencias debe estar encaminada a determinar las características personales, físicas, profesionales, conductuales y evaluar la adecuación o no del candidato al puesto de trabajo.

Sus objetivos son:

- Obtener información inicial sobre el candidato.
- Explorar su trayectoria profesional y personal, así como sus competencias conductuales específicas para el puesto.
- Explorar el área motivacional del candidato y su posible ajuste socio-afectivo al equipo de trabajo.
- Aportar información sobre el puesto para el que se le selecciona, para que el candidato evalúe con profundidad su interés en él.
- Motivar y alentar al candidato para que continúe en el proceso de selección hasta el final.

Durante la entrevista es de vital importancia elaborar un reporte donde se anota toda la información que ofrece el entrevistado. Luego de finalizada la misma se debe actualizar este reporte agregando los

aspectos que conforman la valoración sobre el candidato acerca de: Presentación, Expresión, Contacto, Personalidad entre otros.

Esta técnica, aunque es importante para el proceso de selección de personal, no constituye un método a tener en cuenta en la confección del sistema informático para la identificación de las competencias asociadas a los profesiogramas o perfiles de cargo, porque este sistema está concebido para dar soporte a distancia, lo que implica que este método no se ajuste al problema a resolver.

Questionarios.

Los cuestionarios son un conjunto de interrogantes realizadas a una o varias personas con el objetivo de recopilar información, los mismos pueden contener preguntas abiertas o cerradas, las primeras pueden llegar a ser más difíciles de tabular, pero muestran datos más diversos y enriquecedores. Por el contrario, las preguntas cerradas limitan las respuestas de las personas encuestadas, no obstante, entregan información sistematizada. Este método cuando se vincula a la recopilación de información para elaborar un perfil por competencia, está encaminado a realizar preguntas sobre el puesto en cuestión como son las funciones a desempeñar, competencias requeridas y el peso o grado de importancia que los encuestados otorgan a una u otra competencia.

El sistema informático para identificación de competencias asociadas a los profesiogramas o perfiles de cargo, como parte de la solución que brinda, permite la confección de cuestionarios, esta es una actividad clave del proceso de definición de las competencias necesarias para un puesto específico.

Encuestas.

Es una técnica estructurada que utiliza la comunicación, para recolectar información directamente de los integrantes de una muestra, mediante la aplicación de un cuestionario. Las encuestas pueden verse como una herramienta para la determinación de perfiles por competencia, independientemente de las formas diversas en que puedan ser aplicadas:

Encuesta Personal: Se realiza localizando a los participantes, que en el caso particular, de la identificación de competencias para la elaboración de un perfil de cargo, suelen ser expertos que forman parte del proceso, luego se debe lograr su cooperación para el llenado del cuestionario previamente definido con preguntas redactadas lo más claramente posible, para permitir una correcta interpretación. Por último, se registran los resultados en la forma precisa.

Encuesta en Línea: Son encuestas mediante las cuales, el participante o encuestado llena un cuestionario personalizado localizado remotamente en un servidor de Internet, al que accede mediante un hipervínculo, ventana de navegador o registro. Otra manera de realizar este tipo de encuestas, es mediante del correo electrónico.

El sistema informático para identificación de competencias asociadas a los profesiogramas o perfiles de cargo, utiliza la técnica de las encuestas en línea, para resolver el problema que representa la ubicación geográfica en que pueden encontrarse los expertos que participen en el proceso.

Observación Directa.

Se supervisa directamente la rutina del ocupante del cargo, también puede realizarse de una forma indirecta observando videos de las personas en sus puestos de trabajo. Su objetivo fundamental es identificar y caracterizar los roles efectivamente desempeñados en la situación de trabajo, constatando las características y cualidades requeridas para el ejercicio eficiente del mismo, así como verificar las condiciones ambientales que intervienen en el desarrollo del individuo en la práctica.

Por lo expuesto anteriormente se puede concluir que la observación directa es una herramienta válida en el proceso de construcción del perfil por competencias, pero evidentemente no puede ser el único mecanismo o proceso para dicha elaboración. En muchas ocasiones la información referente a la personalidad del individuo, como por ejemplo, grado de autonomía en la tarea, grado de conocimiento, entre otras, deben ser preguntados personalmente. Esta técnica no es definitiva por sí sola, se hace necesario que se complemente con otros métodos, como la entrevista.

Tormenta de ideas.

La esencia de este método radica en la exposición ideas libres y espontáneas, se debe evitar críticas y ataques entre los participantes. Se emplea para la recolección rápida de ideas, sin tener en cuenta la calidad de éstas, ni su factibilidad, sólo su cantidad, la validez se decide en un paso posterior. Este método requiere ser bien explicado y dejar claramente definidas el papel que juega cada contribuyente del proceso. Todos los miembros deben hacer sus aportes, exponiendo sus ideas libremente, sin criticar, las conclusiones deben sacarse por la mayoría, es preciso buscar sistemáticamente combinaciones o asociaciones entre las ideas enunciadas.

Esta técnica tiene como desventaja que la exposición de las ideas no se entienda con claridad, usualmente esta actividad se convierte en una tarea extenuante. Otro factor a tener en cuenta es que en este método influyen las características personales de los participantes, como pueden ser, la facilidad de expresión, la disposición al debate, entre otras, lo que trae consigo que las ideas planteadas por personas de un carácter extrovertido y con mayor actitud para liderazgo sean adoptadas aunque no constituyan las más acertadas.

Delphi por Rondas.

Este método es el más usado, debido que su funcionamiento está basado en criterios emitidos por expertos que no tienen que estar necesariamente cerca. El primer paso consiste en la creación de un grupo de expertos, el cual es aprobado por la alta dirección de la organización y se efectúa un proceso de entrenamiento en gestión por competencias. Vale aclarar que el método consta de 1...m rondas, hasta alcanzar las competencias necesarias para desarrollar con eficiencia un puesto en particular.

Luego se pasa a la primera ronda, donde cada experto (E) debe responder en una hoja a la pregunta: ¿Cuáles son las competencias que deben conformar el contenido del puesto X?

Los Expertos relacionan todas las competencias, y después reducen el listado erradicando repeticiones o similitudes.

En la **segunda ronda**, se le entrega por separado a cada experto una hoja de papel donde son mostrados los resultados de la ronda 1. Se le pregunta a los expertos: ¿Está usted de acuerdo en que esas son verdaderamente las competencias para ese puesto? Con las que no esté de acuerdo márkuelas con N.

Una vez recogida las respuestas de todos los expertos, es determinado el nivel de concordancia a través de la expresión:

$$C_c = (1 - V_n / V_t) * 100$$

Donde,

Cc: Coeficiente de Concordancia expresado en porcentaje.

Vn: Cantidad de Expertos en contra del criterio predominante.

Vt: Cantidad total de expertos.

Si resulta $C_c \geq 60\%$ se considera aceptable la concordancia. Las C que obtuvieron valores $C_c < 60\%$ se eliminaron por baja concordancia o poco consenso entre los E.

En la **tercera ronda** la pregunta es: ¿Qué ponderación o peso usted daría a cada una de las C, con el objetivo de ordenarlas atendiendo a su importancia en el desempeño de máximo éxito?

Se le da una escala del 1 a n (el número de competencias que resultaron en concordancia), donde el 1 representa la más importante y n la menos importante. Se insiste en que no deben ocurrir iguales ponderaciones a una misma competencia pues se reduciría el poder de discriminación.

Recogidas las respuestas se ordenan las ponderaciones de acuerdo con el valor de la sumatoria de la fila indicada por R_j (sumatoria de ponderaciones por fila). La de menor R_j será la competencia más importante. Si no se cumple que todas las C_c ≥ 60%, hay que pasar a una **cuarta ronda** donde se pregunta:

¿Está de acuerdo con sus ponderaciones y el orden obtenido? Reflexione detenidamente. Puede modificar o mantener sus ponderaciones. Después se procede a repetir idénticos cálculos que en la **tercera ronda**.

Para el procesamiento estadístico implícito en la realización del método Delphi, se utiliza el coeficiente de concordancia Kendall W. (Cuesta Santos, 2000). Este estadígrafo W, por su fundamentación matemática supera al coeficiente de concordancia (C_c). El valor del coeficiente W oscila entre [0...1] y ofrece el nivel de concordancia entre los expertos. Mientras más se aproxime este valor a 1 mayor será la concordancia. Si no se alcanza la concordancia deseada, se recomienda realizar nuevas rondas.

La fórmula matemática para la obtención del coeficiente W se muestra a continuación:

$$W = \frac{S}{1/12 K^2 (N^3 - N) - K \sum T}$$

Donde:

S: suma de los cuadrados de las desviaciones observadas de la media de R_j; su cálculo es efectuado mediante la expresión:

$$S = \sum (R_j - \sum R_j / N)^2$$

K: número de jueces (expertos).

N: número de factores ordenados (competencias).

T: factor de corrección cuando existen observaciones ligadas, obteniéndose así:

$$T = \sum (t^3 - t) / 12$$

t: número de observaciones de un grupo ligado por un rango dado.

Después del análisis de cada uno de los métodos estudiados para la elaboración de perfiles por competencia, se selecciona el método Delphi por Rondas como base para la implementación del sistema informático para la identificación de las competencias asociadas a los profesiogramas o perfiles de cargo, ya que este método es mucho más confiable y completo que el resto de las técnicas vistas anteriormente y en mucho de los casos las incluye como parte de su proceso de realización.

1.4. Sistemas informáticos para la gestión del capital humano en el mundo.

En la actualidad existen en el mundo diversos sistemas con el objetivo de gestionar los recursos humanos. Conformando el objeto de estudio del presente trabajo aquellas aplicaciones que automaticen procesos como: gestión de personal por competencia y registro del desempeño personal. Entre ellas se destacan las siguientes aplicaciones:

RHPlan.

Es una aplicación desktop para la planificación de recursos humanos, con esta herramienta las empresas pueden asignar y monitorizar los recursos humanos durante el desarrollo de software, Contratar y calificar efectivamente a sus profesionales, además permite definir sus propios elementos de competencia (habilidades, conocimientos, experiencias), llevar a cabo la gestión de personal por competencia. (André Ampuero, y otros, 2010)

Sifor.

Esta herramienta se puede encontrar en dos variantes la primera para pequeñas y medianas empresas (Pyme) y otra para empresas de gran tamaño. Sifor permite reducir los costos y aumentar la productividad de los responsables de la gestión del personal, obtener un exhaustivo catálogo con más de 2500 competencias clasificadas entorno a las funcionalidades desarrolladas en la organización y las características actitudinales de las personas, enlazar competencias con planes de acción y obtener potentes informes gráficos que permiten analizar el sistema potencial de la organización. (Neuland Agüero, et al., 2008) y (IMH, 2007)

Cezanne Software.

Provee soluciones avanzadas de Administración de Capital Humano ayudando a las organizaciones a mejorar, gestionar, recompensar y retener sus recursos más importantes: las personas. Sus productos

incluyen aplicaciones para la gestión del rendimiento del empleado, planes de sucesión y carreras, formación y desarrollo, gestión de las personas, selección, análisis salarial, planificación retributiva, revisión salarial, encuestas y diseño de organigramas. (Ibérica, 2007)

Se puede concluir que los sistemas caracterizados anteriormente, aunque poseen potencialidades para la gestión de proyectos y gestión del trabajo en equipo, no constituyen una solución para las empresas cubanas debido a que estas herramientas son propietarias. Lo que va en contra de la actual política adoptada en el país para el uso de software, a raíz de las condiciones impuestas a cuba a través del bloqueo norteamericano. Lo que hace necesario que las empresas utilicen herramientas libres, debido a la imposibilidad de obtener productos propietarios de forma legal.

TEAMSOFT.

Este software fue una propuesta llevada a cabo en el Instituto Superior Politécnico “José Antonio Echeverría” (CUJAE) para dar solución a la falta de una herramienta que sirviera de apoyo a la asignación del personal a los equipos de desarrollo de software, el mismo que posee una interfaz flexible y amigable. El objetivo de este sistema es controlar el desempeño de los miembros de un equipo en la elaboración de un software, aplicando las prácticas de *Personal Software Process* (PSP) y *Team Software Process* (TSP). Para ello a partir de la definición de un proyecto, los roles, las tareas, las actividades, las competencias necesarias y las métricas se puede asignar el personal adecuado y crear las bases para evaluar el desempeño individual y colectivo en el desarrollo del proyecto. El sistema cuenta con dos módulos; uno para el control del desempeño de los equipos que aplica prácticas de TSP y otro para el control del desempeño individual de cada miembro que constituye una ampliación del *Personal Process Dashboard* (herramienta que implementa las prácticas de PSP). (André Ampuero & Baldoquín de la Peña, 2010)

La herramienta antes mencionada posee grandes potencialidades para la gestión de proyectos y gestión del trabajo en equipo, además es un sistema libre. Pero tiene como inconveniente que solo contempla las empresas destinadas al desarrollo de software, por lo que se puede concluir que esta herramienta no se ajusta a las necesidades de la mayoría de las empresas cubanas.

Teniendo en cuenta el análisis previo, se propone la implementación de una aplicación (SISCOM) que permita identificar las competencias asociadas a los profesiogramas o perfiles de cargo y que pueda contribuir a la implantación del SGICH propuesto por la norma cubana.

1.5. Tendencias actuales para el desarrollo de software.

En la actualidad constituye una tarea de primer orden para los desarrolladores de software, conocer las modernas tendencias de las tecnológicas relacionadas con industria informática. Es notable el incremento de usuarios que utilizan software libre, por las ventajas que representa para organizaciones de pocos recursos o que por una u otra razón se encuentran limitadas a la obtención de un conjunto de técnicas de punta, debido a la brecha digital que existe hoy en el mundo. En Cuba se ha adoptado como política para la producción de software, la migración hacia el software libre como una alternativa para enfrentarnos al reto que representa el bloqueo económico impuesto a Cuba por más de 50 años. Un ejemplo donde se aplica hoy esta solución, lo constituye la UCI, donde la utilización de esta tecnología está presente en los proyectos productivos, las instalaciones docentes y facultades que la conforman.

Por estas razones, la gran mayoría de los proyectos informáticos que se llevan a cabo en gran parte del mundo están encaminados a la realización de aplicaciones Web, no solo por los bajos costos que implica su desarrollo, sino también por el alto nivel de flexibilidad y agilidad con que se pueden alcanzar los objetivos trazados por el equipo de desarrollo. Este tipo de técnica facilita el acceso universal a estos servicios sin preocuparse por el tipo de hardware, software, infraestructura de red, origen, ubicación geográfica y capacidades de los usuarios que deseen utilizar el sistema, además permite que la aplicación sea escrita una vez y mostrada a todos los clientes a través del acceso a un servidor. Esto brinda la posibilidad de que las actualizaciones del producto sólo se hagan en el servidor, siendo transparentes para los usuarios, impidiendo que sean mostrados posibles errores de actualización provocados por usuarios de poca práctica. Por las ventajas que ofrece esta nueva tendencia, se ha quedado atrás la realización de aplicaciones de escritorio para Windows, Mac OS X⁵, GNU/Linux⁶, y otros sistemas operativos. Teniendo en cuenta lo expuesto anteriormente se concluye que la mejor vía para la realización del sistema informático para la identificación de las competencias asociadas a los profesiogramas o perfiles de cargo, es la implementación de una aplicación web. Para ello, en las secciones posteriores se ilustran los principales conceptos, tecnologías y herramientas a utilizar en la construcción de la solución propuesta.

⁵ [sistema operativo](#) desarrollado y comercializado por [Apple Inc.](#) que ha sido incluido en su gama de computadoras Macintosh desde [2002](#)

⁶ sistema operativo [libre](#), desarrollado por más de mil voluntarios alrededor del mundo, que colaboran a través de Internet.

1.6. Metodologías de desarrollo de software.

Las mismas permiten llevar a cabo un proceso de desarrollo de software disciplinado con la finalidad de hacerlo más predecible y eficiente, logrando alcanzar productos que cumplan con los requerimientos impuestos por los clientes, en el tiempo establecido con el menor costo y la calidad requerida. Existen diversas metodologías para el desarrollo de software que se clasifican en dos grandes grupos: las robustas o tradicionales y las ágiles.

Las metodologías robustas o tradicionales, se caracterizan por estar centradas específicamente en el control del proceso, lo que las hace efectivas y necesarias en proyectos de gran tamaño que requieren una elevada cantidad de personas para su desarrollo con grandes volúmenes de información. Entre ellas se pueden mencionar: *Rational Unified Process* (RUP) actualmente conocido como *Unified Process* (UP) y *Microsoft Solutions Framework* (MSF). Este tipo de metodologías, no ofrecen una buena solución para proyectos donde el entorno es volátil y los requisitos resulten ser muy variables. Además, invierten un gran costo con relación al tiempo y esfuerzo en la planificación y el desarrollo del proyecto, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas a utilizar.

Las metodologías ágiles dedican poco tiempo a la planificación, invirtiendo la mayor parte del tiempo en el desarrollo, las pruebas y rectificación, con el objetivo de conseguir el resultado adecuado acorde a las necesidades del cliente. Se encuentran orientadas a proyectos pequeños que se realizan en corto plazo, donde el cliente es parte del desarrollo, el equipo de trabajo es reducido y el ambiente es inestable. Entre ellas se pueden mencionar: *Crystal Methodologies*, *Dynamic Systems Development Method* (DSDM), *Adaptive Software Development* (ASD), *Feature-Driven Development* (FDD), *Scrum* y *Extreme Programming* (XP). (Ver Tabla 1.1)

| Metodologías ágiles. | Metodologías tradicional o robustas. |
|---|---|
| Basadas en heurísticas provenientes de prácticas de producción de código. | Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo. |
| Especialmente preparados para cambios durante el proyecto. | Cierta resistencia a los cambios. |
| Impuestas internamente (por el equipo de desarrollo). | Impuestas externamente. |
| Proceso menos controlado, con pocos principios. | Proceso mucho más controlado, con numerosas políticas/normas. |
| No existe contrato tradicional o al menos es bastante flexible. | Existe un contrato prefijado. |
| El cliente es parte del equipo de desarrollo. | El cliente interactúa con el equipo de desarrollo mediante reuniones. |
| Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio. | Grupos grandes y posiblemente distribuidos. |
| Pocos artefactos. | Más artefactos. |
| Pocos roles. | Más roles. |
| Menos énfasis en la arquitectura del software. | La arquitectura del software es esencial y se expresa mediante modelos. |

Tabla 1. Diferencias entre metodologías ágiles y no ágiles. (Canós, Letelier, & Penadés, 2002)

El Sistema informático para la identificación de las competencias asociadas a los profesiogramas o perfiles de cargo, consta de dos integrantes en el equipo de desarrollo, con un tiempo limitado para la obtención del producto, con la presencia del cliente durante el Ciclo de Vida del software, con

requisitos variables y con un entorno de trabajo volátil. Estas razones permiten identificar en las metodologías ágiles, el camino adecuado que guíe todo el proceso de desarrollo ya que estas se consideran, ideales para pequeños equipos de desarrollo, evitando el exceso de robustez en las documentaciones y adaptándose perfectamente a lidiar con todo tipo de cambios enfrentados a lo largo del proceso de producción.

Aunque todas las metodologías ágiles cumplen con las características expuestas anteriormente, cada una cuenta con sus propias especificidades que serán objeto de análisis, con la finalidad de seleccionar la que más se ajuste al proyecto.

Crystal Methodologies.

Conjunto de metodologías para el desarrollo de software que se caracterizan por estar centradas en las personas que componen el equipo (de ellas depende el éxito del proyecto) y la reducción al máximo del número de artefactos producidos. El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas. La familia *Crystal* dispone un código de color para marcar la complejidad de una metodología, cuanto más oscuro un color, más “pesado” es el método. Cuanto más crítico es un sistema, más rigor se requiere. Estos métodos no prescriben las prácticas de desarrollo, las herramientas o los productos que pueden usarse, pudiendo combinarse con otros métodos como *Scrum*, *XP* y *Microsoft Solutions Framework*. Una de las variantes más usada es *Crystal Clear (CC)*, la cual acentúa el proceso como un conjunto de ciclos anidados (Díaz, 2006). La generación de una documentación extensa, es una de las desventajas que presenta esta metodología, pues en su desarrollo se obtienen 20 artefactos, lo que hace muy largo el ciclo de vida del proyecto.

Dynamic Systems Development Method (DSDM).

Establece el marco para desarrollar un proceso de producción de software. Sus principales características son: es un proceso iterativo e incremental, en que el equipo de desarrollo y el usuario trabajan juntos. Propone cinco fases: estudio viabilidad, estudio del negocio, modelado funcional, diseño y construcción, y finalmente implementación. Durante estas fases, se producen un gran número de artefactos, lo que hace más compleja la utilización de esta metodología.

Adaptive Software Development (ASD).

Sus principales características son: iterativo, orientado a los componentes software más que a las tareas y tolerante a los cambios. El ciclo de vida que propone tiene tres fases esenciales: especulación, colaboración y aprendizaje. En la primera de ellas se inicia el proyecto y se planifican las características del software; en la segunda desarrollan las características y finalmente en la tercera se revisa su calidad, y se entrega al cliente. La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo. ASD jerarquiza más a los componentes que a las tareas, lo que implica que esté más dirigida a la calidad, por tal motivo no brinda una guía para el desarrollo de software, sino que está enfocada a adaptación de una empresa a los cambios.

SCRUM.

Propone una forma de gestionar proyectos de software. Como método, enfatiza valores y prácticas de gestión de manera que puede ser considerado como un conjunto de patrones organizacionales, sin pronunciarse sobre requerimientos, implementación y demás cuestiones técnicas define un proceso empírico, iterativo e incremental de desarrollo que se fundamenta en la aceptación de la naturaleza caótica del desarrollo de software, y la utilización de prácticas tendientes a manejar el riesgo a niveles aceptables. El ciclo de vida de SCRUM consta de 4 fases: Pre-Juego, Pre-Juego o Montaje (Staging), Juego o desarrollo, Pos-Juego o Liberación. (Kniberg, 2007) y (Díaz, 2006)

Xtreme Programing (XP).

Es una metodología ligera de desarrollo que incorpora un conjunto de buenas prácticas conocidas en programación, las cuales tienen como propósito:

- Aplicar un modelo de espiral evolutivo donde el cliente es parte activa de proceso de desarrollo, logrando que los programadores estén en constante intercambio con ellos, para satisfacer sus requisitos y responder rápidamente a los cambios solicitados.
- Aplicar la programación de pares promoviendo la despersonalización del código, con el objetivo de obtener un código limpio y legible, sin el temor de reprogramar si es necesario.

XP se distingue por la claridad de su codificación y su diseño, lo que hace que esta metodología se base en la simplicidad. Su ciclo de vida es iterativo en el que se incluyen seis fases fundamentales: Exploración, Planificación de la Entrega, Iteraciones, Producción, Mantenimiento, Muerte del Proyecto. (Kniberg, 2007) y (Díaz, 2006)

Sin embargo, ninguno de los artefactos generados en esta metodología, tienen en cuenta la gestión de proyecto a la hora de desarrollar un software, por lo que se puede afirmar que XP por sí sola no cuenta con elementos que favorezcan el desarrollo de esta actividad durante su utilización.

Scrum Xtreme Programing SXP.

Esta metodología brinda una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control del mismo.

Esta metodología consta de 4 fases principales en las que se realizan numerosas actividades de donde se generan artefactos para documentar todo el proceso. A continuación se especifican tanto las actividades y artefactos generados en todo el ciclo de vida de esta metodología.

Por cada fase se generan las siguientes actividades y artefactos:

1. Planificación-Definición: Se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.

Actividades:

- Escribir la visión (Concepción del sistema), presupuesto y reserva del producto con estimaciones iniciales.

Artefactos:

- Concepción del sistema.
- Historia de usuario.
- Lista de reserva del producto (LRP).
- Lista de riesgos.
- Modelo de diseño.
- Modelo de historia de usuario del negocio.

2. Desarrollo: Se realiza la implementación del sistema hasta que esté listo para ser entregado.

Actividades:

- Reunión de planificación de la iteración.
- Definir la reserva de la iteración.
- Reuniones de coordinación.

- Revisión de la iteración.
- Realización de las pruebas de aceptación del sistema.

Artefactos:

- Caso de prueba de aceptación.
- Cronograma de producción.
- Estándar de código.
- Plan de release.
- Tarea de ingeniería.

3. Entrega: Puesta en marcha.

Actividades:

- Documentación.
- Entrega.
- Marketing.

Artefactos:

- Manual de desarrollo.
- Manual de identidad.
- Manual de usuario.

4. Mantenimiento: Donde se realiza el soporte para el cliente.

Actividades:

- Soporte.

Artefactos:

- Gestión de cambio.

Cada uno de estos artefactos es desarrollado y supervisado por los roles enumerados a continuación:

1. Líder del Proyecto.
2. Gerente.
3. Especialistas.
4. Consultor.
5. Cliente.

6. Miembros del Proyecto.
7. Programadores.
8. Analista.
9. Diseñadores.
10. Encargado de Pruebas.
11. Arquitecto.

Las entregas son frecuentes, y existe una refactorización continua, lo que permite mejorar el diseño cada vez que se le añade una nueva funcionalidad. SXP está especialmente indicada para proyectos de pequeños equipos de trabajo, rápido cambio de requisitos o requisitos imprecisos, muy cambiantes, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. Esta metodología ayuda a que trabajen todos juntos, en la misma dirección, con un objetivo claro, permitiendo además seguir de forma clara el avance de las tareas a realizar, de forma que los jefes pueden ver día a día cómo progresa el trabajo. (Kniberg, 2007) y (Díaz, 2006)

Teniendo en cuenta el estudio realizado y las características del equipo de desarrollo del presente trabajo de diploma, se ha escogido como metodología a utilizar SXP, que fue creada en la UCI, como resultado del proyecto SXP, de la facultad 10, asesorada por el grupo Unicornio y aprobada por CALISOFT (Romero, 2008). Esta metodología no es más que la combinación de SCRUM y XP, reuniendo sus mejores prácticas. Las mismas pueden integrarse de manera fructífera teniendo en cuenta que SCRUM se enfoca en la organización y gestión de proyecto, mientras que XP se centra más en las prácticas de programación, por lo que se complementan, brindando la mejor opción para el desarrollo del sistema informático para la identificación de las competencias asociadas a los profesiogramas o perfiles de cargo.

1.7. Lenguajes de modelado.

El uso de los lenguajes de modelado brinda al equipo de desarrollo de un software una guía para llevar a cabo con organización y calidad la realización del producto, facilitado la visualización desde diferentes puntos de vista las características y funcionalidades del sistema que va a construirse. Además, los modelos alcanzados con un alto nivel de abstracción pueden ayudar a una mejor comunicación con el cliente, logrando que estos participen de forma activa en el proceso de concepción del producto, para cumplir satisfactoriamente las necesidades y expectativas de los mismos.

Existen varios lenguajes que permiten realizar varias actividades claves de la ingeniería de software como representar los procesos, definir, organizar, visualizar los elementos que configuran la arquitectura de una aplicación entre otras. Entre los más usados se encuentran Método para la Modelación Funcional de Procesos (IDEF0), Notación de Modelado de Procesos de Negocio (BPMN), Lenguaje Unificado de Modelado (UML) entre otros.

Método para la modelación funcional de procesos (IDEF0).

Estándar para el desarrollo estructurado de gráficos representativos de procesos, provee técnicas de modelados simples y formales que facilitan la descripción, el análisis y la evaluación de distintos puntos de vista de una organización. También permite producir un “Modelo Funcional” o representación estructurada, de procesos (funciones o actividades) del sistema a modelar. IDEF0 en su forma original incluye una definición de lenguaje gráfico del modelo (sintaxis y semántica) y una descripción de una metodología extensa para modelos en desarrollo.

De manera general IDEF0 tiene las siguientes características:

1. Técnica genérica que permite modelar gráficamente procesos a cualquier nivel de detalle.
2. Comprensivo y expresivo, capaz de representar gráficamente reglas y una amplia variedad de negocios.
3. Lenguaje simple y coherente.
4. Permite una mejor comunicación entre los analistas de sistemas, los desarrolladores y los usuarios por la facilidad de aprender sus distintas representaciones y por su énfasis en la exposición jerárquica del detalle.
5. Puede ser generado por una variedad de herramientas de modelado gráfico de computadora.
6. Facilita el análisis para identificar puntos de mejora.

Notación de Modelado de Procesos de Negocio (BPMN).

Nuevo estándar para la modelación de procesos de negocio, su principal objetivo es proveer una notación fácil de entender por parte de todos los involucrados e interesados del negocio como los analistas de negocio, los desarrolladores técnicos y los gerentes y administradores del negocio. En síntesis: BPMN tiene la finalidad de servir como lenguaje común para cerrar la brecha de comunicación que frecuentemente se presenta entre el diseño de los procesos de negocio y su implementación. Los objetos de flujo tienen tres elementos centrales: evento, actividad y decisión; estos se conectan en un

diagrama para crear el esqueleto básico de la estructura de un proceso de negocio. BPMN en su versión actual predefine sólo tres tipos de artefactos: objeto de datos, grupo y comentario.

Para el estudio y el correcto entendimiento del negocio que conlleva a la realización de esta tesis, se propone la utilización de esta notación para el modelado de procesos de negocio (BPMN), la cual vinculándose otro lenguaje gráfico como UML y con herramientas para el modelado, que serán estudiadas a profundidad más adelante en el presente trabajo, establece una buena práctica para el modelado de flujos de procesos y servicios web.

Lenguaje Unificado de Modelado (UML).

Lenguaje gráfico para el modelado de procesos que permite detallar la arquitectura, los objetos, los datos y elementos del ciclo de vida de una aplicación, así como aspectos más relacionados con el diseño de componentes incluyendo su construcción y despliegue. Además, permite realizar especificaciones estáticas y dinámicas de un sistema orientado a objetos. Los modelos estáticos incluyen la definición de atributos, operaciones, interfaces, clases y las relaciones que se establecen entre ellas, como pueden ser la herencia, asociación, dependencia, etc. Mientras que los modelos dinámicos por su parte pueden representarse mediante los diagramas de secuencia y colaboración que UML facilita. En casos más complejos, este lenguaje de modelado también proporciona mecanismos para el mejor entendimiento de las características particulares del sistema a modela como la representación de máquinas de estado. Por último UML incluye una notación para representar el agrupamiento del diseño lógico por medio de componentes, el despliegue y ubicación de esos componentes en nodos dentro de una arquitectura distribuida. (Rumbaugh, Jacobson, & Booch, 2002)

Por todas las características expuestas anteriormente y teniendo presente que UML ayuda a una construcción eficiente de aplicaciones de alta calidad y a un menor coste. Se selecciona como lenguaje gráfico, para junto BPMN poder llevar a cabo tanto el modelado de los procesos de negocio, como la definición y visualización de todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar la documentación requerida para el correcto desarrollo del sistema informático para la identificación de las competencias asociadas a los profesiogramas o perfiles de cargo.

1.8. Herramientas CASE (Computer Aided Software Engineering).

En la actualidad existen un gran número de herramientas que facilitan el proceso de elaboración de sistemas informáticos. Las herramientas CASE forman parte de las nuevas tecnologías de punta en la industria de software, las mismas constituyen aplicaciones que favorecen el desarrollo, reduciendo el esfuerzo, el costo y el tiempo. Entre las más utilizadas en todo el mundo, se encuentran: *Enterprise Architect* (EA), *Rational Rose* y *Visual Paradigm* (VP). Cada una de estas herramientas cuenta con sus propias particularidades lo que las hace divergir entre ellas, por esta razón es de vital importancia realizar la valoración de cada una de ella, para poder seleccionar la que más se ajusta a las características tanto del equipo de trabajo, como las del sistema a desarrollar.

Enterprise Architect (EA).

Es una herramienta comprensible de diseño y análisis que utiliza UML, cubriendo el desarrollo de software desde el paso de los requerimientos a través de las etapas del análisis, diseño, pruebas y mantenimiento. Es una herramienta multi-usuario, basada en Windows, diseñada para ayudar a construir software robusto y fácil de mantener. Ofrece salida de documentación flexible y de alta calidad. Ayuda a administrar la complejidad con herramientas para rastrear las dependencias, brinda soporte para modelos muy grandes, control de versiones con proveedores CVS o SCC, líneas base por cada punto del tiempo, la utilidad de comparar para seguir los cambios del modelo, interfaz intuitiva y de alto rendimiento con vista de proyecto como un "explorador". Provee una generación poderosa de documentos y herramientas de reporte con un editor de plantilla completo WYSIWYG. Ayuda a visualizar sus aplicaciones, soportando ingeniería inversa de un amplio rango de lenguajes de desarrollo de software y esquemas de repositorios de base de datos. El control de versiones es muy lento, por lo menos con subversion (SVN), y es complicado hacer combinaciones entre dos importaciones con elementos en común. Además es una herramienta de licencia propietaria o privativa. El Software privativo, es aquel cuya copia, redistribución o modificación está en alguna medida prohibido por su propietario. Para realizar alguna de estas operaciones es necesario solicitar un permiso o pagar, lo que la hace a esta herramienta difícil de obtener por llegar alcanzar altos precios de mercado. Enterprise Architect es recomendada para empresas de gran tamaño que puedan financiar los gastos de la compra de licencia de software para el desarrollo de sus productos.

Rational Rose.

Es una herramienta de modelación visual basada en UML. Presenta plataforma independiente que ayuda a la comunicación entre los miembros de equipo, a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas. Utiliza la notación estándar en la arquitectura de software, la cual permite a los integrantes del equipo de desarrollo visualizar el sistema completo, utilizando un lenguaje común. Además, los diseñadores pueden modelar sus componentes e interfaces en forma individual, y luego unirlos con otros componentes del proyecto. Poderosa en el modelado de análisis y diseño de sistemas basados en objetos, cubriendo todo el ciclo de vida de un proyecto.

Las principales características de Rational Rose:

1. No es gratuito, se debe hacer un previo pago para poder adquirir el producto.
2. La ingeniería de código (directa e inversa) es posible para ANSI C++, Visual C++, Visual Basic 6, Java, J2EE/EJB, CORBA, Ada 83, Ada 95, Bases de datos: DB2, Oracle, SQL 92, SQL Server, Sybase, Aplicaciones WEB.
3. Solamente Ingeniería reversa para COM.
4. Rational Rose habilita asistentes para crear clases y provee plantillas de código que pueden aumentar significativamente la cantidad de código fuente generado. Adicionalmente, se pueden aplicar los patrones de diseño, Rational Rose ha provisto 20 de los patrones de diseño GOF para Java.
5. Admite la integración con otras herramientas de desarrollo (IDE).

Visual Paradigm (VP).

Es una herramienta de modelado, soporta UML y permite modelar todos los diagramas necesarios para representar gráficamente el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue.

Entre las características que posee se encuentran:

1. Soporte de UML versión 2.1.
2. Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento.
3. Modelado colaborativo con CVS y SVN.
4. Interoperabilidad con modelos UML2 (metamodelos UML 2.x para plataforma Eclipse) a través de XML.
5. Ingeniería de ida y vuelta.

6. Ingeniería inversa - Código a modelo, código a diagrama.
7. Ingeniería inversa Java, C++, Esquemas XML, XML, NET exe/dll, CORBA IDL.
8. Generación de código - Modelo a código, diagrama a código.
9. Editor de Detalles de Casos de Uso - Entorno todo-en-uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.
10. Diagramas EJB - Visualización de sistemas EJB.
11. Generación de código y despliegue de EJB - Generación de beans para el desarrollo y despliegue de aplicaciones.
12. Diagramas de flujo de datos.
13. Soporte ORM - Generación de objetos Java desde la base de datos.
14. Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos.
15. Ingeniería inversa de bases de datos - Desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación.
16. Generador de informes.
17. Distribución automática de diagramas - Reorganización de las figuras y conectores de los diagramas UML.
18. Importación y exportación de ficheros XML.
19. Integración con Visio - Dibujo de diagramas UML con plantillas (stencils) de Microsoft Visio.
20. Editor de figuras.

Visual Paradigm (VP) es la herramienta CASE a utilizar en el presente trabajo, teniendo en cuenta las características enumeradas anteriormente, que se adecuan a los lenguajes de modelado seleccionados para el desarrollo del sistema informático para la identificación de las competencias asociadas a los profesiogramas o perfiles de cargo. Es válido señalar que VP es una herramienta profesional, su licencia es comercial, es decir, que la empresa que desarrolla este software, lo hace con fines lucrativos, cuyo lucro se logra con la distribución del mismo. La licencia del VP se encuentra en la UCI, por lo que no existe impedimento alguno para su utilización. Es fácil de instalar, de actualizar y compatible entre ediciones, también proporciona abundantes tutoriales, demostraciones interactivas y proyectos de UML.

1.9. Lenguajes de programación.

Se define a un lenguaje de programación como un conjunto de símbolos, reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Se utilizan para controlar el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión y para la comunicación entre los humanos con los ordenadores.

En la actualidad existen muchos lenguajes de programación, pero la mayoría presentan altos precios, por lo que no resulta prudente utilizarlos, atendiendo a la filosofía de desarrollo de nuestro país, que es utilizar el software libre, es por eso que solo se van a analizar tres lenguajes de programación libres Java, PHP, y *Python*. Otro aspecto a tener en cuenta para el análisis de los mismos, es que en nuestra Universidad, los dos primeros presentan un porcentaje similar de proyectos que los usan para desarrollar, y el último no está a la misma altura.

Hypertext Preprocessor (PHP).

Fue creado originalmente por Rasmus Lerdorf en 1994. PHP es un lenguaje de programación, diseñado originalmente para la creación de páginas web dinámicas. Es un lenguaje libre. Es usado principalmente en interpretación del lado del servidor pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+. La mayor parte de su sintaxis es similar a C, Java y Perl. En nuestra universidad existe más de un 10 % de proyectos que utilizan este lenguaje para programar. Su presencia, en el impresionante número de 10 millones de servidores lo ha llevado a estar muy por encima de cualquier otro lenguaje script. Es un lenguaje específicamente diseñado para realizar aplicaciones web, mientras que otros lenguajes son adaptaciones de lenguajes preexistentes, no pensados para la web. Mantiene un bajo consumo de recursos de máquina. Presenta gran seguridad, muy poca probabilidad de corromper los datos. Dispone de muchísimos más recursos que ASP. Las tareas fundamentales que puede realizar directamente el lenguaje son definidas en el mismo lenguaje como funciones. Presenta una filosofía totalmente diferente y, con un espíritu más generoso, es progresivamente construido por colaboradores desinteresados que implementan nuevas funciones en nuevas versiones del lenguaje. (Ciberaula, 2006) (Group, 2001-2009)

Java.

Es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principio de los años 90's. Es muy robusto, muchos lo recomiendan para proyectos grandes. Presenta una buena

forma de expresar los algoritmos en sus sintaxis, estas son muy parecidas al C++. La eliminación de punteros, lo ha hecho mucho más seguro. Se considera que es el lenguaje mejor definido, pues ha logrado erradicar la ambigüedad. Se dice es compatible con los sistemas de base de datos más usados mundialmente. Debido a que los programas desarrollados en java, no corren en el lenguaje de máquina de la PC, sino en uno denominado "*Byte Code*", que es más bien un pseudo-lenguaje, presenta un lenguaje intérprete llamado Máquina Virtual de Java, que se encarga de comunicar el de la PC con el Byte Code, por esta funcionalidad se consume buena parte de la memoria RAM del ordenador, por lo que las aplicaciones hechas en java son más lentas que las de otros lenguajes. (Microsystems, 1994-2010)

Python.

Es un lenguaje de programación creado por Guido van Rossum a principios de los años 90. Es un lenguaje similar a Perl, pero con una sintaxis muy limpia y que favorece un código legible. Se trata de un lenguaje interpretado o de script, con tipado dinámico, es decir, no es necesario declarar el tipo de dato que va a contener una determinada variable. Es orientado a objetos. Es un lenguaje multiplataforma. La sintaxis de Python es sencilla y muy cercana al lenguaje natural. Debido a que es un lenguaje interpretado, necesita consumir más recursos del ordenador que otros lenguajes. Por esta razón se concluye que no es un lenguaje factible para la implementación del sistema informático propuesto.

Se puede concluir que el lenguaje de programación a utilizar para la implementación de el sistema informático para la identificación de las competencias asociadas a los profesiogramas o perfiles de cargo es PHP, ya que desde su creación fue pensado para la realización de aplicaciones web, además es un lenguaje libre y tiene como ventaja sobre otros lenguajes como Java y Python su escaso consumo de recursos del ordenador, pues no requiere un intérprete para comunicarse con el lenguaje máquina.

1.10. Framework de desarrollo.

Los framework, se pueden considerar como una aplicación genérica y configurable, que constan de un conjunto de componentes (clases de java script, archivos XML y descriptores) que evitan la reinención de la rueda, aspecto muy importante que disminuye, costo, esfuerzos y recursos. Ofrecen una mejor organización y estructura a los proyectos.

En la actualidad existen muchos framework, pero este trabajo va a centrarse en el análisis de solo dos de ellos, el *Symfony* (usado en varios proyectos de la UCI) y el *CodeIgniter*, no tan conocido, pero sí muy prometedor. Se dice que ambos son un tanto parecidos e incluso, especialistas afirman que el *CodeIgniter* es el hermano menor del *Symfony*, pero tienen características específicas que los diferencian. A continuación se muestran las principales características de cada uno de ellos.

CodeIgniter.

Es un framework sencillo a la hora de configurar y de usar, está destinado para proyectos pequeños. Es compatible tanto con PHP 4 y PHP 5, esta característica lo hace superior a otros framework. Posee un gran soporte para múltiples bases de datos, plantillas, validaciones. Funciona en servidores dedicados o compartidos, y es multiplataforma (Windows, Mac, o Linux). Posee un solo archivo de configuración. No presenta líneas de comando. Sus librerías pueden ser acopladas con facilidad. Usa el Modelo Vista Controlador, como estructura de programación. La filosofía para los que trabajan con este framework es: “desarrollar aplicaciones lo más rápidamente posible”. (Ellislab, 2006-2009)

Symfony.

Es un framework más robusto, y por sus propiedades, es mejor utilizarlo para proyectos grandes, ejemplo de esto es el proyecto de Fiscalía General de la República, de la facultad 15 de la UCI. Es un poco difícil de manejar con servidores compartidos y solo es compatible en su totalidad con plataforma Linux. Cuenta con muchos archivos para su configuración (más de 6). Presenta una amplia línea de comandos con muchísimas posibilidades. Consume más memoria RAM, debido a su robustez. Tiene dependencia de librerías internas. Usa también el MVC. Permite encapsular el código para minimizarlo. (Eguiluz, 2010)

Debido a las características de la aplicación que se va a desarrollar y su sencillez, es mejor utilizar el *CodeIgniter*, ya que va en correspondencia a lo antes expuesto, es decir, se usará un framework pequeño en la implementación de una aplicación pequeña. Además para la capa de presentación se propone usar la librería *EXTjs*, que no es más que una librería java script open-source de alto rendimiento, que sirve para la creación y desarrollo de aplicaciones web dinámica. Provee interfaces

gráficas de usuario que brindan experiencias parecidas o iguales a las que se tienen con aplicaciones de escritorio. Permite la utilización de componentes predefinidos. Es extensible para la gran mayoría de los navegadores, evitando el tedioso problema de validar el código para cada uno de estos. La comunicación asíncrona permite la comunicación con el servidor sin necesidad de pedirle una acción al usuario, dando la libertad de cargar la información sin que el usuario lo note. Por otra parte, para la capa de acceso a datos, se va a usar Doctrine que no es más que un ORM (*Object Relations Mappers*). Que se usa para PHP 5.2 y posterior a esta. Dentro de sus principales ventajas está el uso del lenguaje DQL (*Doctrine Query Language*), que está basado en el lenguaje de consultas de Hibernate. Cuenta con estándares y patrones de diseño que lo hace una herramienta muy útil a la hora de implementar. Por tener un mapeador incluido, permite la reutilización de código, lo que facilita que se puedan llamar métodos de distintas partes de la aplicación e incluso de otros sistemas. Admite el encapsulamiento de la lógica de los datos y se pueden implementar mecanismos de seguridad para proteger la aplicación de ataques como SQL Injections.

1.11. Entorno de desarrollo integrado (IDE).

Los entornos de desarrollo integrado, son programas informáticos, compuestos por varias herramientas de programación, y destinados a los programadores. Estos programas están compuestos por un editor de código, un compilador, un depurador y un constructor de interfaz gráfica.

El equipo de desarrollo, teniendo en cuenta la política a seguir por el país, y a universidad de usar aplicaciones libres, propone solamente analizar a los IDE libres, a continuación se muestran dos de estos:

Eclipse PHP

Es un IDE libre, multiplataforma y de licencia open Source. Fue creado por IBM, como sucesor de la familia VisualAge. La base arquitectónica para Eclipse es la Plataforma de cliente enriquecido, todo lo contrario de cliente ligero (RCP) y está constituido por los siguientes componentes:

- Plataforma principal - inicio de Eclipse y ejecución de plugin.
- OSGi - una plataforma para el desarrollo estándar.
- El Standard Widget Toolkit (SWT) - Un widget toolkit portable.
- JFace – manejador de texto y de archivos, incluye editores de texto.
- El Workbench de Eclipse – posee vistas, editores, perspectivas y asistentes.

Dentro de las características que presenta para PHP están:

1. Auto-completado y coloreado de funciones PHP nativas.
2. *Outline* (muestra la estructura en métodos, propiedades de nuestra clase)
3. Inserta bloques PHPDoc.
4. Debugging.
5. Autoformato de documento.
6. Explorador de proyecto
7. Refactoring
8. Explorador de base de datos SQL.

Aptana Studio

Es un IDE de desarrollo gratuito, y de código libre. Presenta soporte para las plataformas más conocidas, Windows, Linux y Mac OS. Posee soporte Ajax, PHP, Ruby on Rails y Adobe Air. Está basado en el conocido entorno de desarrollo Eclipse. Dispone de una herramienta para conexión por FTP para hacer *upload* o *download* de archivos, así como la sincronización de nuestros contenidos en local con los del servidor remoto. Su instalador es de solo 85 Mb. Entre sus características se pueden mencionar las siguientes:

1. Ayudas visuales para la escritura de scripts en diversos lenguajes, como coloreado y auto escritura del código, ayudas contextuales de referencia a medida que se escribe, etc.
2. Visualización de errores de sintaxis a medida que se escribe.
3. Debug en Firefox o Internet Explorer.
4. Librerías de funciones en Java script populares en Ajax/Java script para utilizar en los proyectos.
5. Ejemplos ya creados para empezar a conocer las posibilidades de desarrollo rápidamente.
6. Extensible a partir de plugins que puede crear Aptana u otras empresas y herramientas para estar al tanto de cualquier nuevo añadido.
7. Extensible por Java script.
8. Permite escribir scripts para realizar acciones y macros.
9. Los Snippets permiten insertar fragmentos de texto que se utilizan muy a menudo.

NetBeans.

Herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación. Existe además un

número importante de módulos para extender el IDE NetBeans. El IDE NetBeans es un producto libre y gratuito sin restricciones de uso. Por estas razones constituye el IDE de desarrollo a utilizar en la implementación del sistema propuesto.

1.12. Sistemas gestores de bases de datos (SGBD).

Son aplicaciones informáticas que sirven de interfaz entre las bases de datos, los usuarios y las aplicaciones que la utilizan. Permite la administración de los datos en una o varias bases de datos. El objetivo fundamental de un SGBD consiste en suministrar al usuario las herramientas que le permitan manipular, en términos abstractos, los datos, sin importar la forma en que estos estén almacenados en el servidor. Por estas razones, deben garantizar la fiabilidad, seguridad y confidencialidad de los datos. A continuación, algunos de los SGBD más usados en la actualidad:

Microsoft SQL Server.

Potente gestor de base de datos, desarrollado por la Microsoft. Está basado en el lenguaje Transact-SQL y específicamente en Sybase IQ. Su última versión salió en el año 2008. Dentro de sus principales características están:

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Además permite administrar información de otros servidores de datos.
- Es un software privativo, y su licencia tiene altos costos.

Oracle.

Potente gestor de bases de datos, de última generación, esto quiere decir que está orientado para acceso remoto y redes. Es multiplataforma. Está orientado para grandes bases de datos y mucho flujo de información. Por su potencia y altos precios, las grandes empresas y multinacionales, son su principales clientes. Es uno de los gestores más completos. Hoy en día se encuentra sufriendo la competencia de SQL y de otros gestores con licencia libre como PostgreSQL.

PostgreSQL.

Este gestor es considerado por muchos como el mejor y más avanzado gestor que existe en la actualidad. Presenta una estabilidad y un prestigio legendario, desde su creación en la Universidad de

Berkeley. Posee licencia BSD. Una de sus principales características es la alta concurrencia. Esto permite que mientras se realizan cambios en una tabla, otros procesos accedan a la misma tabla sin la necesidad de bloqueos, además de que cada usuario tiene visión de la última modificación. Presenta un amplio número de enlaces con lenguajes de programación como son C, C++, Java, Perl, PHP, Python y otros. (PostgreSQL Global Development, 1996 – 2008)

Teniendo en cuenta lo expuesto anteriormente, se concluye que el gestor de base de datos a utilizar en la implementación del sistema informático propuesto es PostgreSQL, ya que además de ser considerado un potente gestor, es el único que posee licencia en la UCI.

1.13. Servidores web.

Son programas que son diseñados para transferir continuamente hipertextos, páginas web o paginas HTML (Hipertext Markup Language). Implementa el protocolo HTTP (Hipertext transfer Protocol, perteneciente a la capa de aplicación del modelo OSI). Se ejecuta continuamente en una computadora, esperando peticiones por parte de los clientes, y responde a las peticiones adecuadamente.

Dentro de los servidores web más reconocidos están:

Apache.

Apache es el servidor web más conocido del mundo, su configurabilidad, robustez y estabilidad hacen que cada vez millones de personas reiteren su confianza sobre este programa. Un 70 % de los servidores webs del mundo son Apache. Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal. Apache es una tecnología gratuita open-source. Actualmente existen muchos módulos para Apache que son adaptables a este, y están disponibles para su instalación cuando sean necesarios. Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurarlo para que ejecute un determinado script cuando ocurra un error en concreto. (Ciberaula, 2006)

AOLServer.

Es un servidor HTTP de tipo multihebra, basado en TCL, que incluye muchas facilidades de uso orientadas a entornos de gran escala y a sitios web con contenido dinámico. Hay que destacar que todos los dominios y servidores de AOL, que son más de 200, soportan miles de usuarios simultáneos y millones de conexiones funcionan con AOLServer. Gracias a su integración con OpenACS, es un software de gestión de contenidos de código libre muy potente, desarrollado por una empresa llamada

ArsDigita y liberado bajo licencia GPL. Es un tanto deficiente para dar apoyo a proveedores de servicios de internet que requieren de miles de páginas estáticas. Todo su desarrollo usa el mismo lenguaje, por lo que solo requiere de las funciones internas. Fue diseñado con la intención de contar con una excelente conexión a servidores de bases de datos.

Para el desarrollo de la aplicación se decide utilizar a Apache como servidor web, ya que es un software libre; altamente configurable de diseño modular que permite ampliar sus capacidades. Hoy en día cuenta con miles de módulos, y cualquiera que tenga experiencia en C o en Perl, puede implementar un módulo agregándole las funcionalidades que necesite. Es compatible con lenguajes como Perl, Java, C, y sobre todo con PHP, que es el lenguaje que se va a utilizar para la implementación del sistema. Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor, posibilitando configurarlo para que ejecute un determinado script cuando ocurra un error concreto.

1.14. Enfoque de la arquitectura de software para las metodologías ágiles.

En las metodologías ágiles, no se hace demasiado énfasis en la definición temprana de una arquitectura de software para el sistema, la misma se va definiendo y mejorando a lo largo del proyecto. Teniendo en cuenta las características de estas metodologías es necesario que la arquitectura no sea demasiado compleja, si es preciso un cambio y el mismo no se ajusta a la arquitectura entonces esta se cambia.

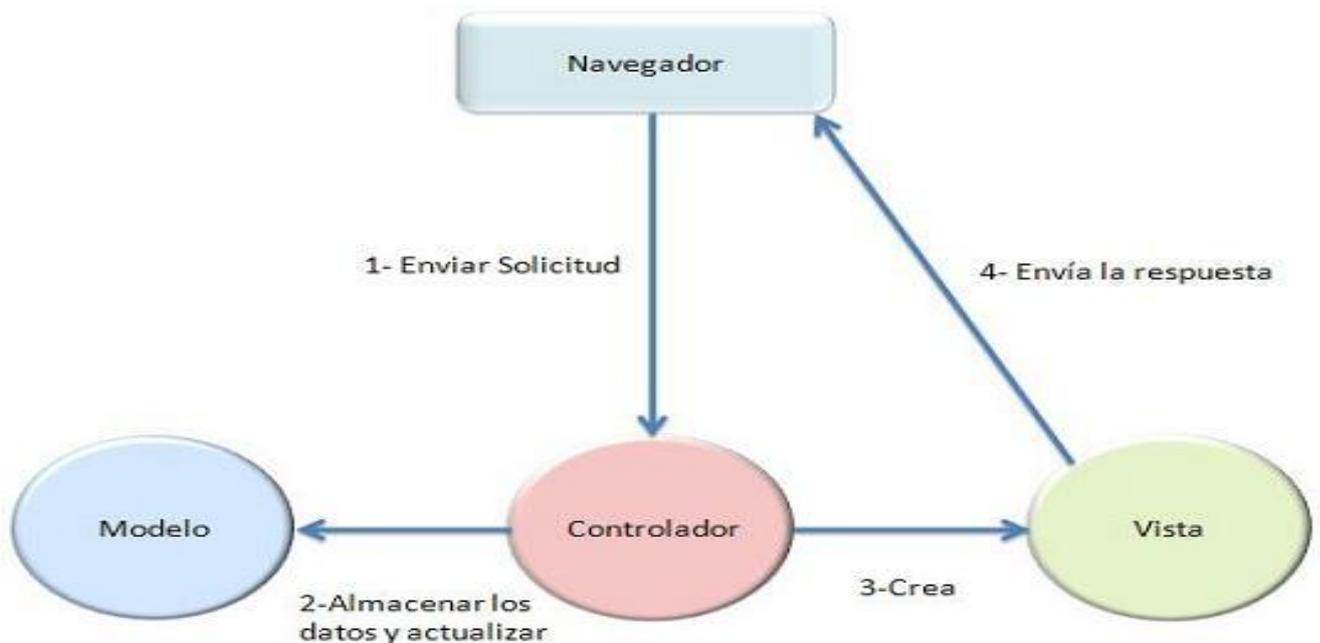
En el caso específico de XP la arquitectura se asume de forma evolutiva y los posibles inconvenientes que se generarían por no contar con ella explícitamente en el comienzo del proyecto se resuelven con la existencia de una metáfora. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema (conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema, ayudando a la nomenclatura de clases y métodos del sistema). (Canós, y otros, 2002)

En esta metodología, la descripción de la arquitectura se concibe a través del documento de la arquitectura, donde se resume toda la información referente a la utilización de los diferentes patrones tanto de diseño como los patrones arquitectónicos, al igual que las herramientas, lenguajes de programación, IDE y framework utilizados.

1.15. Patrones de arquitectura.

Para llevar a cabo la implementación del sistema propuesto, es necesario el uso de patrones de desarrollo, entre ellos se encuentran los patrones de arquitectura, que brindan una solución genérica a un problema que se presenta en un contexto específico y de manera recurrente. Como ya se ha expuesto anteriormente, en el desarrollo de la aplicación se utilizará el *framework* CodeIgniter, el cual está basado en un patrón clásico del diseño web conocido como arquitectura Modelo – Vista – Controlador (MVC). El mismo permite realizar la programación multicapa, separando en tres componentes distintos los datos de una aplicación, la interfaz del usuario y la lógica de control.

Por las razones enunciadas anteriormente el sistema que se implementará en el presente trabajo tendrá un estilo arquitectónico Modelo Vista Controlador, como se muestra en la siguiente figura:



Representación gráfica del modelo.

1.15.1. Patrón Modelo – Vista – Controlador.

Este modelo fue creado con el propósito de minimizar el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. Su particularidad radica en

que el modelo, la vista y el controlador se tratan de manera separada, lo que garantiza que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas.

Las partes que componen al MVC:

El Modelo: Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.

La Vista: Maneja la presentación visual de los datos representados por el Modelo. Interactúa con el Modelo a través de una referencia al mismo.

El Controlador: Proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza un cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo.

Este patrón convierte una aplicación en un paquete mantenible, modular y de desarrollo rápido. La modularidad y el diseño independiente permiten a los desarrolladores y diseñadores hacer cambios en alguna parte de la aplicación sin afectar a los demás.

1.16. Patrones de diseño.

Estos patrones brindan soluciones a problemas específicos que aparecen de forma frecuente en el diseño orientado a objeto. Además hacen más fácil la reutilización exitosa de la arquitectura y los diseños. Para el diseño del sistema propuesto se tuvieron en cuenta una serie de patrones, que se describen a continuación.

1.16.1. Patrones GRASP

Los patrones GRASP, forman parte de los patrones de diseños para el desarrollo de software. Describen los principios fundamentales del diseño de objetos para la asignación de responsabilidades los que se verán durante el proceso de desarrollo del sistema para la definición de las competencias asociadas a los profesiogramas o perfiles de cargo, serán.

Experto: Este patrón es utilizado ya que permite asignar una responsabilidad al experto en información: el módulo que cuenta con la información necesaria para cumplir la responsabilidad. Ofrece facilidad de entender, mantener y manipular. Permite la reutilización. Para hacer uso del mismo se tuvo en cuenta que la responsabilidad de hacer una labor debe ser de la clase que tiene o puede tener datos involucrados (atributos) a esa labor. Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada.

Creador: Propone que una instancia de un objeto tiene que ser creada por el objeto que tiene la información para ello. Solución que es factible para el diseño del sistema puesto que si se asignan bien el diseño puede soportar un bajo acoplamiento, mayor claridad, encapsulación y reutilización.

Bajo Acoplamiento: Plantea la asignación de responsabilidades de manera que el acoplamiento permanezca bajo, es decir, de forma tal que las clases se comuniquen con el menor número de clases que sea posible. Permitiendo tener la mínima conexión entre los módulos de esta aplicación.

Alta cohesión: Cada elemento de nuestro diseño debe hacer una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable, una clase con baja cohesión hace muchas cosas no relacionadas o hace demasiado trabajo. El uso de este patrón permite asignar a las clases responsabilidades que trabajen sobre una misma área de la aplicación y que no tengan mucha complejidad.

Controlador: El manejo de este patrón facilita establecer la creación de clases que se asocian con operaciones del sistema como respuestas a los eventos del mismo, tales como mensajes, métodos y relaciones entre ellos.

1.16.2. Patrones GOF.

Un patrón de diseño es una descripción de clases y objetos que se comunican entre sí, adaptada para resolver un problema general de diseño en un contexto particular. La utilización de estos patrones facilitan la localización de los objetos que formarán el sistema, Proporcionan la determinación de la granularidad adecuada, también especifican interfaces para las clases e implementaciones. Garantizan un fácil aprendizaje y una buena comunicación entre programadores y diseñadores.

Para la implementación del sistema se utiliza el framework CodeIgniter, el cual implementa una serie de patrones GOF.

Singleton (Instancia única): Garantiza que una clase tenga una única instancia y proporciona un mecanismo de acceso global a dicha instancia. Este patrón proporciona una serie de ventajas entre las que se pueden mencionar. El acceso a la instancia única está más controlado, además permite refinamientos en las operaciones y en la representación, mediante la especialización por herencia.

Decorator (Envoltorio): Se encarga de añadir dinámicamente funcionalidad a un objeto. Esto da la posibilidad de no tener que crear sucesivas clases que hereden de la primera incorporando la nueva funcionalidad, sino otras que la implementan y se asocian a la primera.

Command (Comando): Este patrón encapsula una petición en un objeto permitiendo con ello, parametrizar a los clientes con diferentes peticiones y almacenar peticiones para deshacerlas en caso necesario. Garantiza que se pueda definir una dependencia entre un objeto y un conjunto de ellos, de modo que los cambios en el primero se vean reflejados en los otros. (Gamma, y otros, 1995) y (Prieto, 2008)

1.17. Conclusiones.

En este capítulo se vieron los principales conceptos relacionados con la gestión del capital humano por competencias. A partir del estudio de las herramientas existentes se decidió utilizar SXP como metodología de desarrollo, UML y BPMN como lenguajes de modelado; como herramienta CASE se seleccionó al Visual Paradigm, para la programación de la aplicación se eligió el lenguaje PHP, también se propone CodeIgniter como framework de desarrollo, utilizando la librería EXTjs para la capa de presentación y el ORM Doctrine para la capa de acceso a datos. Para la gestión de la base de datos se utilizará el servidor PostgreSQL y como servidor web Apache.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1. Introducción.

En este capítulo se realiza la descripción del sistema informático para la identificación de las competencias asociadas a los profesiogramas o perfiles de cargo, para ello se describe el proceso del negocio, a través de los diagramas del modelo de negocio. Se especifican los requisitos funcionales y no funcionales con los que cuenta el sistema y se muestra el modelo conceptual del mismo. Se presentan algunas planillas necesarias para poder desarrollar con la metodología ágil seleccionada. Se explica toda la dinámica del proyecto en forma de historias de usuarios, prototipos de interfaz de usuario y algunos modelos auxiliares.

2.2. Concepción del sistema.

En la planilla de concepción del sistema que propone la metodología SXP en la fase de planeación se muestra la visión y descripción general del producto a implementar, recoge los diferentes roles que intervendrán en el desarrollo del software, así como las tecnologías usadas. (Peñalver Romero, y otros, 2009)

2.2.1. Descripción del producto.

En nuestro país con el objetivo de aumentar la producción y la calidad de los servicios de las empresas y organizaciones, se ha decidido adoptar un SGICH, para ello la norma cubana propone un modelo formado por 9 procesos fundamentales, cuyo núcleo lo constituyen las competencias laborales. En el mismo se define el Comité de Competencias como el grupo responsable para identificar y proponer a la alta dirección, las competencias claves o distintivas de la organización.

Un elemento fundamental para la elaboración de los perfiles de cargo es la identificación de las competencias, esta actividad requiere de una táctica que permita obtener las cualidades necesarias para ocupar un puesto determinado en una empresa, para ello el sistema propuesto implementa un conjunto de pasos y métodos que garantizan dar cumplimiento a esta tarea de una manera realmente eficaz. El sistema informático para la identificación de las competencias asociadas a los profesiogramas o perfiles de cargos, está concebido como una aplicación web, que automatiza el método Delphi por Rondas, con el objetivo de obtener las competencias que componen el perfil de cargo mediante el criterio de expertos.

2.2.2. Planificación del proyecto por roles.

| Rol | Miembro |
|-------------------------------|--|
| Gerente (Management) | Yoiner Barallobre Reyes |
| Cliente (Customer) | Janet Rodríguez Febles |
| Programadores (Programmers) | Yoiner Barallobre Reyes Maray Corrales Guerrero |
| Analista (Analyst) | Maray Corrales Guerrero |
| Diseñadores (Designers) | Yoiner Barallobre Reyes |
| Encargado de Pruebas (Tester) | Maray Corrales Guerrero |
| Arquitecto (Architect) | Yoiner Barallobre Reyes |

2.3. Modelo de Historia de Usuario del negocio.

En la planilla de modelo de historia de usuario del negocio que propone la metodología SXP en la fase de planeación se especifican todos los actores y trabajadores del negocio, realizando una descripción simple de cada uno de ellos. Además se ilustra el diagrama de historias de usuario del negocio.

2.3.1. Actores del negocio.

| Actores | Descripción |
|----------------|---|
| Organizaciones | Empresas cubanas que como parte de la implantación del SGICH, necesitan identificar, las competencias asociadas a los cargos. |

2.3.2. Trabajadores del negocio.

| Trabajadores | Descripción |
|----------------------------------|--|
| Empresas cubanas | Interesadas en identificar las competencias asociadas a los cargos. |
| Alta dirección de la empresa. | Responsable de la aprobación de las competencias de los cargos definidas por el Comité de Competencias. |
| Jefe del comité de competencias. | Responsable máximo del proceso de identificación de las competencias y la selección de los expertos. |
| Especialistas. | Personal identificado inicialmente como posibles expertos a participar en el proceso de identificación de competencias. |
| Expertos. | Especialistas que cumplen con los requisitos necesarios para participar en el proceso de identificación de las competencias. |

2.3.3. Diagrama de procesos del negocio.

En este epígrafe se ilustran el diagrama de cada uno de los procesos que conforman el negocio, además se hace una descripción detallada de cada uno de ellos. (Ver artefactos)

2.4. Lista de Reserva del Producto (LRP).

En la planilla LRP que propone la metodología SXP en la fase de planeación se define el trabajo que se va a realizar en el proyecto, priorizando cada uno de los requisitos funcionales y no funcionales del sistema. Es importante tener en cuenta que cuando un proyecto comienza es muy difícil tener claro todos los requerimientos sobre el producto, por ello esta lista puede crecer y modificarse a medida que se obtiene más conocimiento acerca del producto y del cliente, para así poder garantizar que el

producto definido al terminar la lista es el más correcto, útil y competitivo posible. (Peñalver Romero, y otros, 2009)(Ver artefactos)

2.5. Historia de Usuario.

En la plantilla de historia de usuario que propone la metodología SXP, en la fase de planeación se recoge la descripción de cada uno de los requisitos funcionales presentes en la LRP, debe recogerse la estimación del tiempo en que será desarrollado dicha funcionalidad y el nombre del miembro del equipo de trabajo que la implementará (Ver Anexo # 1), además de contar con una imagen de cada una de las interfaces relacionadas con dicha historia de usuario. (Ver artefactos)

2.5.1. Tareas de ingeniería.

En la plantilla de tareas de ingeniería que propone la metodología SXP, en la fase de desarrollo se recogen las tareas correspondientes a cada una de las historias de usuario a realizar para llevar a cabo el desarrollo del sistema. (Ver Anexo # 1) También se realizó el prototipo de interfaz de usuario para cada una de estas plantillas. (Ver artefactos)

2.6. Modelo de datos.

A continuación se presenta la concepción del modelo de datos empleado para dar comienzo a la implementación del sistema para la identificación de las competencias asociadas a los profesiogramas o perfiles de cargo. Para la realización del mismo se tuvieron en cuenta las reglas de normalización y para facilitar su diseño se utilizaron nomencladores, quedando un modelo de 9 clases persistentes y otras 9 que son producto de relaciones de mucho a mucho, o simplemente nomencladores. Para generar el modelo se aplicó ingeniería inversa utilizando la herramienta Visual Paradigm. Como gestor de base de datos se utilizó PostgreSQL en su versión 8.3. (Ver artefactos)

2.7. Plan de release.

En el plan de release, que propone la metodología SXP para la parte de desarrollo, se definen las iteraciones a realizar para lograr las entregas intermedias y la entrega final del producto. Se tiene como entrada la relación de Historias de Usuario definidas previamente. Para colocar una historia en cada iteración se tiene en cuenta la prioridad que definió el cliente para dicha Historia. Cada iteración debe

tener asociado un tiempo, dado en semanas. Se puede concluir que el producto debe estar listo dentro de dos meses.

| Release | Historias de Usuarios | Tiempo estimado (Semanas) |
|---------|---------------------------|---------------------------|
| 1 | 1; 2; 3; 4 | 2 |
| 2 | 5; 6; 7; 8; 9; 10; 11; 12 | 4 |
| 3 | 13;14;15;16;17 | 2 |

2.8. Conclusiones.

En el presente capítulo se realizó la planificación del proyecto, dejando bien claro los roles establecidos para cada integrante del mismo. Se describieron todos los actores del negocio así como la modelación de los principales modelos de negocio con que cuenta el sistema. Se desarrolló la lista de reserva del producto, estableciendo los principales requisitos funcionales y no funcionales, y su importancia. Se establecieron las historias de usuario y las principales tareas de ingeniería asociadas a las mismas para la primera planificación y se priorizaron. Las iteraciones se planificaron teniendo en cuenta la prioridad para el negocio de cada historia de usuario. Se realizó el diseño de la base de datos del sistema, el cual permitió dar una mejor visión del proyecto.

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

3.1. Introducción.

Durante la elaboración de un software, suelen realizarse algunos errores, por esta razón se hace necesario acompañar este proceso de actividades y técnicas que garanticen la calidad del mismo. En este capítulo se realiza la validación de la solución propuesta, realizando varios tipos de pruebas que facilitan la identificación de las deficiencias presentes en el sistema y permiten comprobar la eficacia de las respuestas dadas a los distintos requisitos planteados por el cliente.

Los tipos de pruebas que se le realizan son: unidad, integración, aceptación y rendimiento los que serán descritos a continuación.

3.2. Pruebas de unidad.

Son pruebas realizadas a las unidades mínimas desarrolladas durante la implementación de un software, estas unidades pueden ser clases, métodos, propiedades, componentes, etc. Las pruebas realizadas a SISCO, están dirigidas a los diferentes módulos que lo conforman. Se prueban los caminos de control más importantes, con el fin de descubrir errores dentro del ámbito del módulo que afecten el buen funcionamiento del mismo, haciendo un uso intensivo de las técnicas de prueba de caja blanca, utilizando la herramienta Unit Testing que contiene el framework CodeIgniter. Se utilizaron métricas que facilitan una buena medición de dos factores fundamentales para probar que una aplicación es fiable y fácilmente mantenible.

Factor descriptivo: Identifica el código propenso a errores, difícil de entender, modificar y probar para ello se calcula la complejidad ciclomática.

Factor prescriptivo: Indica si se debe dividir un módulo altamente complejo en otros más sencillos se estima la complejidad esencial.

Las pruebas de caja blanca: Están orientadas al análisis exhaustivo del código que conforma la aplicación.

A continuación, utilizando esta técnica se analiza cada uno de los métodos críticos de cada módulo del sistema. Esta tarea requiere que se calcule la complejidad ciclomática, para obtener el número de

caminos linealmente independientes del código a examinar y así obtener la cantidad de casos de pruebas a llevar a cabo.

Complejidad ciclomática: Esta métrica desarrollada por Tomas McCabe en 1976, es una medida cuantitativa de la complejidad lógica de un programa. EL uso de esta métrica también permite estimar el riesgo que representa cada fragmento de código para un software, y brinda una idea de lo complicado que puede ser su mantenimiento, estos factores pueden medirse teniendo en cuenta la tabla propuesta por McCabe:

| Complejidad Ciclomática | Evaluación del Riesgo |
|-------------------------|--|
| 1-10 | Programa Simple, sin mucho riesgo |
| 11-20 | Más complejo, riesgo moderado |
| 21-50 | Complejo, Programa de alto riesgo |
| 50 | Programa no testeable, Muy alto riesgo |

3.2.1. Cálculo de la complejidad ciclomática.

Primeramente se debe obtener el diagrama de flujo o grafo del código en cuestión, luego se pasa al cálculo de la complejidad ciclomática utilizando alguno de los tres métodos que existen.

1. Método de predicados: consiste en adicionarle una unidad, a la sumatoria, de los nodos predicados que no son más que los formados por condiciones y por tanto de ellos se desprenden dos o más aristas.

$$V(G) = \sum \text{Nodos Predicados} + 1$$

2. Métodos de regiones: Consiste en sumar todas las regiones internas del grafo más la externa.

$$V(G) = \sum \text{Región}_1 + \text{Región}_2 + \text{Región}_{n\dots}$$

3. Método de aristas y nodos: Consiste en restarle al número de aristas, la cantidad de nodos y luego sumarle dos unidades.

$$V(G) = (\sum \text{Aristas} - \sum \text{Nodos}) + 2$$

Ejemplo cálculo de la complejidad ciclomática de un método:

```
public function adicionarUsuario ()
{
    $idempresa=$this->input->post ('idempresa');
    $nombre_completo=$this->input->post ('nombre_completo');
    $usuario=$this->input->post ('usuario');
    $contrasena=$this->input->post ('contrasena');
    $repcontrasena=$this->input->post ('repcontrasena');
    $idrol=$this->input->post ('idrol');

    $usuario_array=Usuario::obtenerUsuarioNombreUsuario ($usuario, $idempresa);

    //Para validar un solo Jefe de Comite de Competencia
    if ($idrol==3)
    {
        $cantidad_jefe=count (Usuario::obtenerJefeCompetencia ($idempresa));
        if ($cantidad_jefe!=0)
        {
            $info['codeMsg']=2;
            $info['msg']="Ya existe un Jefe de Comite de Competencias para esta Empresa";
            echo (json_encode ( $info));
            return;
        }
    }
}
```

```

    }
}
if(count ($usuario_array)==0)
{
    if($contrasena==$repcontrasena)
    {
        $user=new User();
        $user->nombrecompleto=$nombre_completo;
        $user->usuario=$usuario;
        $user->password=$contrasena;
        //$user->Usuario->idrol=$idrol;
        //$user->Usuario->idempresa=$idempresa;

        $user->save();

        $dat_usuario=new Usuario();
        //$dat_usuario->nombrecompleto=$nombre_completo;
        //$dat_usuario->usuario=$usuario;
        //$dat_usuario->password=$contrasena;
        $dat_usuario->iduser=$user->iduser;
        $dat_usuario->idrol=$idrol;
        $dat_usuario->idempresa=$idempresa;
        $dat_usuario->save();
        $info['codeMsg']=1;
        $info['msg']="Se ha adicionado satisfactoriamente";
        echo (json_encode ( $info));
    }
    {
        $info['codeMsg']=2;
        $info['msg']="La contraseña debe confirmarse correctamente";
        echo (json_encode ( $info));
    }
}
else
{
    $info['codeMsg']=2;
    $info['msg']="Ese usuario ya esta siendo utilizado por la Empresa.";
    echo (json_encode ( $info));
}
}

```

Complejidad ciclomática del método adicionarUsuario ().

| | | | |
|----------------|----------------------------|---------------------------------|--------------------------------------|
| | Método de predicados | Métodos de regiones | Método de aristas y nodos |
| | $V(G) = \sum \text{Nodos}$ | $V(G) = \sum \text{Región 1} +$ | $V(G) = (\sum \text{Aristas} - \sum$ |
| | Predicados + 1 | Región 2 + Región | Nodos) + 2 |
| | <u>Entonces...</u> | n... | <u>Entonces...</u> |
| $V(G) = 3 + 1$ | <u>Entonces...</u> | $V(G) = 13 - 11 + 2$ | |
| $V(G) = 4$ | $V(G) = 4$ | $V(G) = 4$ | |

Complejidad esencial:

Es una medida cuantitativa de la cantidad de construcciones no estructuradas que se encuentran en un módulo, entendiéndose por esto aquellas que no siguen la lógica normal de su uso.

La complejidad esencial siempre debe mantenerse en el intervalo entre 1 y el valor de complejidad ciclomática calculada para un método. Si fuese 1, significaría que el código del módulo en cuestión está completamente estructurado, pero si por el contrario es igual al valor de la complejidad ciclomática significaría que el código está completamente desestructurado.

$$1 \leq eV(G) \leq V(G)$$

El método de cálculo de la complejidad esencial, es calcular la complejidad ciclomática del grafo de flujo reducido.

Grafo Reducido: Es aquel grafo de flujo en el que los nodos correspondientes a instrucciones estructuradas son reducidos a 1 nodo. Las instrucciones estructuradas son principalmente los if y los if – else.

$$eV(G) = V(G')$$

Complejidad esencial del método adicionarUsuario ().

| | | | |
|-----------------|-----------------------------|----------------------------------|---------------------------------------|
| | Método de predicados | Métodos de regiones | Método de aristas y nodos |
| | $V(G') = \sum \text{Nodos}$ | $V(G') = \sum \text{Región 1} +$ | $V(G') = (\sum \text{Aristas} - \sum$ |
| | Predicados + 1 | Región 2 + Región | Nodos) + 2 |
| | <u>Entonces...</u> | n... | <u>Entonces...</u> |
| $V(G') = 0 + 1$ | <u>Entonces...</u> | $V(G) = -1 + 2$ | |

| | | | |
|--|--|-------------|------------|
| | | $V(G') = 1$ | $V(G) = 1$ |
|--|--|-------------|------------|

Después de haberle calculado la complejidad ciclomática y la complejidad esencial a todos los métodos críticos de la aplicación, se observa que los valores máximos obtenidos para los métodos que componen SISCO, es 6 para la complejidad ciclomática y 2 para la complejidad esencial, por lo que se puede concluir que los resultados han sido favorables, ya que el valor máximo recomendado por Tomas McCabe es 10 para la Complejidad Ciclométrica y 5 para la Complejidad Esencial.

3.3. Pruebas de aceptación.

Estas pruebas no son más que la comprobación de cada una de las funcionalidades del sistema, este tipo de pruebas están basado en el criterio de pruebas de caja negra. Se realiza un caso de prueba por cada historia de usuario identificada en la fase de planeación. Y el responsable de llevarlas a cabo es el cliente. Al aplicar estas pruebas, se pudo comprobar que la aplicación cumple con todos los requisitos funcionales identificados previamente. (Ver artefactos)

3.4. Pruebas de rendimiento.

La estabilidad y la disponibilidad son factores claves para SISCO, ya que está concebido como una aplicación Web, para prestar servicios a las empresas cubanas, las que requieren que el sistema sea capaz dar respuesta a las peticiones de los usuarios con un alto grado de rapidez.

Las pruebas de rendimiento pueden tener varios propósitos, para determinar cómo responde la aplicación a una determinada carga y para validar otros atributos de calidad como pueden ser la estabilidad, la fiabilidad o el uso de recursos entre otros.

Existen distintos tipos de Pruebas de Rendimiento que le ayudan a mejorar las capacidades de una aplicación. A continuación se muestra los resultados obtenido al aplicar un tipo de prueba de rendimiento al sistema obtenido.

3.4.1. Pruebas de Stress.

La herramienta que se utiliza para las pruebas de rendimiento es el software, JMeter. El mismo fue implementado en Java y cuyo propósito es probar el comportamiento funcional y medir el desempeño de una aplicación Web, esta es una herramienta open source y de fácil uso.

Utilizando las funcionalidades que brinda JMeter para realizar este tipo de test, se configuró con el siguiente caso de prueba:

Entrada de 10 usuarios al sistema:

1. Autenticación en el sistema lo que consta de entrar nombre de usuario y contraseña.
2. Desplegar la página de bienvenida con las opciones correspondientes al rol de cada uno de los usuarios autenticados.
3. Realizar las actividades correspondientes a cada rol.

Administrador

- a. Adiciona usuario
- b. Modifica usuario
- c. Elimina Usuario

Jefe comité de Competencia

- d. Crea cuestionario
- e. Revisa cuestionario

Experto

- f. Responder cuestionario

4. Salir del sistema.

Los usuarios se conectan y realizan las actividades descritas anteriormente, para lo cual tomaron se los tiempos de respuesta y registraron alguno de los datos estadísticos que brinda esta herramienta. Para encontrar el número ideal de usuarios que soporta la aplicación, se realizaron varias pruebas con un número de usuarios incremental. Hasta obtener que el caso óptimo es 900 y el caso crítico es de 950. En la tabla que se muestra a continuación se muestra el reporte que arroja JMeter para las pruebas realizadas.

| Actividades/ Respuestas | 1 | 2 | 3.a | 3.b | 3.c | 3.d | 3.e | 3.f | 4 |
|----------------------------|------|------|------|-----|-----|------|------|------|-----|
| Cantidad de repeticiones | 950 | 950 | 950 | 950 | 950 | 950 | 950 | 950 | 950 |
| Tiempo/milisegundos | 3949 | 1231 | 8 | 9 | 6 | 156 | 124 | 215 | 0 |
| Rendimiento /sec | 2.27 | 0.77 | 118 | 106 | 158 | 6.08 | 7.66 | 4.41 | 0 |
| % Error | 7.71 | 0.00 | 0.00 | 0.0 | 0.0 | 0.00 | 0.0 | 0.0 | 0.0 |

Se pudo observar que la respuesta crítica se detectó en la actividad de autenticación, ya que se mostró un por ciento de error de 7.71% lo que quiere decir que de los 950 usuarios de muestra a 73 se le mostró un mensaje de error. El tiempo promedio de conexión obtenido, fue de 0.66.

| Actividades/ Respuestas | 1 | 2 | 3.a | 3.b | 3.c | 3.d | 3.e | 3.f | 4 |
|----------------------------|------|------|------|------|------|------|------|------|------|
| Cantidad de repeticiones | 900 | 900 | 900 | 900 | 900 | 900 | 900 | 900 | 900 |
| Tiempo/milisegundos | 1291 | 1031 | 4 | 5 | 2 | 95 | 82 | 100 | 0 |
| Rendimiento /sec | 1.3 | 0.22 | 128 | 92 | 106 | 4.07 | 5.26 | 2.41 | 0 |
| % Error | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

Para el caso óptimo se utilizó una muestra de 900 y se obtuvieron muy buenos resultados ya que el por ciento de error de las respuestas del sistema para cada una de las actividades fue de 0.00% lo que quiere decir que no surgieron problemas de conexión. Por lo que se puede concluir que el sistema está listo para soportar una alta cantidad de usuarios, que es un factor crítico para la aplicación desarrollada.

3.5. Conclusiones.

En este capítulo, se realizaron diferentes tipos de pruebas como son, unidad, aceptación, y rendimiento, para ello se emplearon técnicas tanto de caja negra, como de caja blanca, para garantizar

la calidad de la aplicación, obteniendo resultados satisfactorios, lo que demuestra que SISCOM está listo para implantación.

CONCLUSIONES

- Con la presente investigación se dan por cumplidos los objetivos propuestos en el trabajo de diploma, pues se estudiaron las tendencias de la gestión del capital humano por competencias los sistemas informáticos, las tecnologías, metodologías y herramientas, quedando establecidas así las bases para la implementación del sistema informático para la identificación de competencias a los perfiles de cargo.
- Se realizaron los artefactos propuestos por la metodología utilizada, tales como la Lista de Reserva del Producto (LRP), el Plan de Release, las Historias de Usuario y otros.
- Se implementó el sistema SISCOM para identificar las competencias asociadas a los profesiogramas utilizando las herramientas:
 - Visual Paradigm como herramienta CASE.
 - CodeIgniter como framework de desarrollo.
 - EXTjs como librería para la capa de presentación.
 - ORM Doctrine para la capa de acceso a datos.
 - PostgreSQL como gestor de base de datos.
 - PHP como lenguaje de programación.
 - Apache como servidor web.
- Se realizó el proceso de validación del software a través de diferentes técnicas y métricas que permitieron detectar y corregir los errores de la fase de implementación garantizando un buen funcionamiento del producto final, que del mismo se puede decir que cumple con las especificaciones establecidas por el Sistema de Gestión del Capital Humano, propuesto por la Norma Cubana.

RECOMENDACIONES

- Agregar nuevas funcionalidades al sistema que contribuyan a mejorar el proceso de identificación de competencias.
- Extender la aplicación SISCOM a todas las empresas cubanas para elevar la productividad de las mismas.
- Brindar soporte al sistema desarrollado.

BIBLIOGRAFÍA

André Ampuero, Margarita and Baldoquín de la Peña, María Gulnara. 2010. Un sistema de soporte a la decisión para la asignación de recursos humanos a equipos de proyectos de software. Ciudad de la Habana : REVISTA INVESTIGACIÓN OPERACIONAL, 2010. Vol. 31, 1. 61-69.

Canós, José H, Letelier, Patricio and Penadés, María Carmen. 2002. *Métodologías Ágiles en el Desarrollo de Software*. Valencia : s.n., 2002.

Cuestas, Armando. 2005. *Tecnología de Gestión de Recursos Humanos*. Ciudad de la Habana : s.n., 2005. Segunda edición corregida y ampliada.

Díaz, María Irma. 2006. *La incertidumbre y la ingeniería de software*. 2006.

Eguiluz, Javier. 2010. *Symfony*. [Online] 2010. <http://www.symfony.es/>.

G.P.Bunk. 1994. *La transmisión de las competencias en la formación y perfeccionamiento profesionales de la RFA*. 1994.

Gamma, Erich, et al. 1995. *Design Patterns. Elements of Reusable Object-Oriented Software*. 1995.

González, Maura. 2002. ¿Qué significa ser un profesional competente? Reflexiones desde una perspectiva psicológica. *Revista Cubana de Educación Superior*. 2002, Vol. Volumen XXIII No. 1 .

Ibérica, Cezanne Software. 2007. CEZANNE SOFTWARE. [Online] 2007. [Cited: 03 15, 2010.] <http://www.cezannesw.com/es/>.

IMH, Instituto de máquina Herramienta. 2007. Gestión de personas por competencias. *IMH20urte*. [Online] 07 19, 2007. [Cited: 03 20, 2010.] <http://www.imh.es/publicaciones/ficheros/efcii.pdf>.

Neuland Agüero, Dennis, Muñoz Castillo, Vanesa Danae and Hernández Díaz, Lester Reinier. 2008. "TEAMSOFTE: Sistema para la gestión del trabajo en equipo en el desarrollo de proyectos de software". 2008.

Norma_Cubana. 2007. Norma Cubana 3000. *SISTEMA DE GESTIÓN INTEGRADA DE CAPITAL HUMANO*. La Habana : s.n., 2007. 3000-3002.

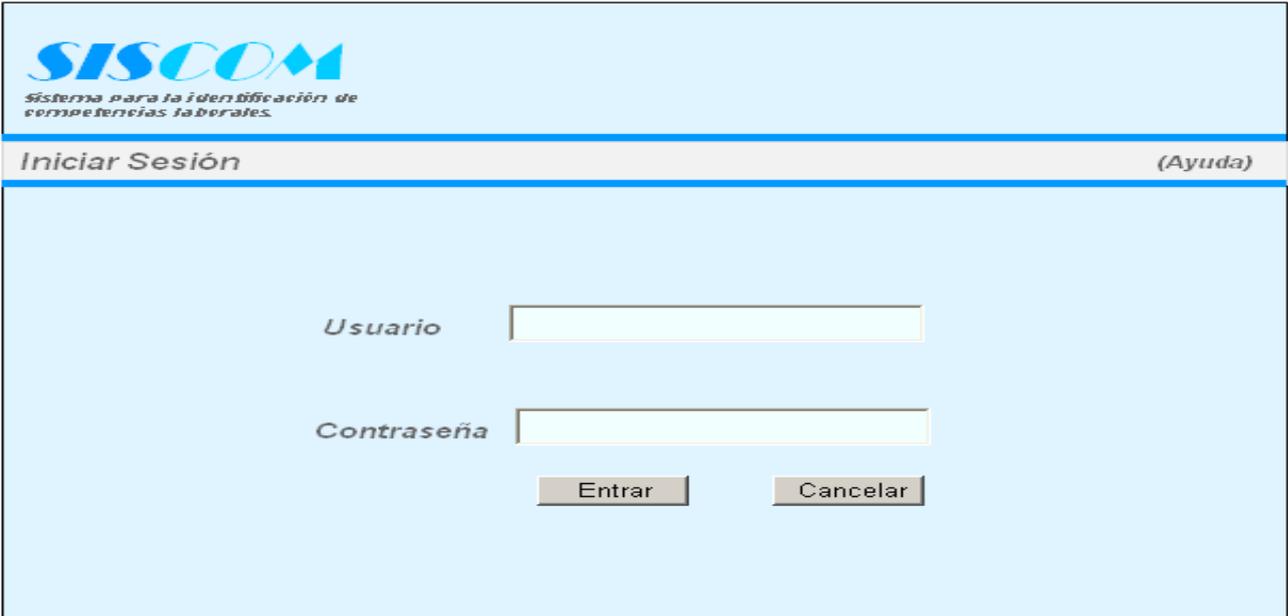
Peñalver Romero, Gladys Marsi, et al. 2009. SXP. Metodología ágil para proyectos de software libre. *Metodología ágil*. La Habana : s.n., 2009.

Prieto, Félix. 2008. Patrones de diseño. *Programación de Sistemas*. Valencia : s.n., 2008.

Vargas, Fernando. 2004. 40 preguntas sobre competencia laboral. *cinterfor*. [En línea] 2004. <http://www.cinterfor.org.uy/public/spanish/region/ampro/cinterfor/temas/complab/xxxx/esp/index.htm>.

ANEXOS

Anexo #1: Prototipos de interfaz de usuarios.



The image shows a login screen for the SISCOM system. At the top left, the logo "SISCOM" is displayed in blue, with the tagline "Sistema para la identificación de competencias laborales." below it. The page title "Iniciar Sesión" is centered at the top, and a link "(Ayuda)" is located at the top right. The main content area contains two input fields: "Usuario" and "Contraseña", each followed by a text box. Below these fields are two buttons: "Entrar" and "Cancelar".

2.1 Pantalla para entrar a la aplicación.

SISCOM
sistema informático para la identificación de competencias asociadas a los profesioqramas o perfiles de cargo.

Opciones de Administrador (Cancelar Proceso) (Salir)

| Empresa | Procesos | Estado |
|--|---------------|------------|
| <input checked="" type="checkbox"/> PESCATUM | Rol Analista | Activo |
| <input type="checkbox"/> | Tecnico ref. | Finalizado |
| <input type="checkbox"/> | Cocinero | Activo |
| <input type="checkbox"/> | Recepcionista | Finalizado |
| <input checked="" type="checkbox"/> UCI | Programador | Finalizado |
| <input type="checkbox"/> | Analista | Activo |
| <input type="checkbox"/> | | |

2.2 Pantalla para cancelar un proceso de una empresa determinada.

SISCOM
sistema informático para la identificación de competencias asociadas a los profesioqramas o perfiles de cargo.

Opciones de Administrador (Insertar Empresa) (Cancelar) (Salir)

Nombre de la empresa: *

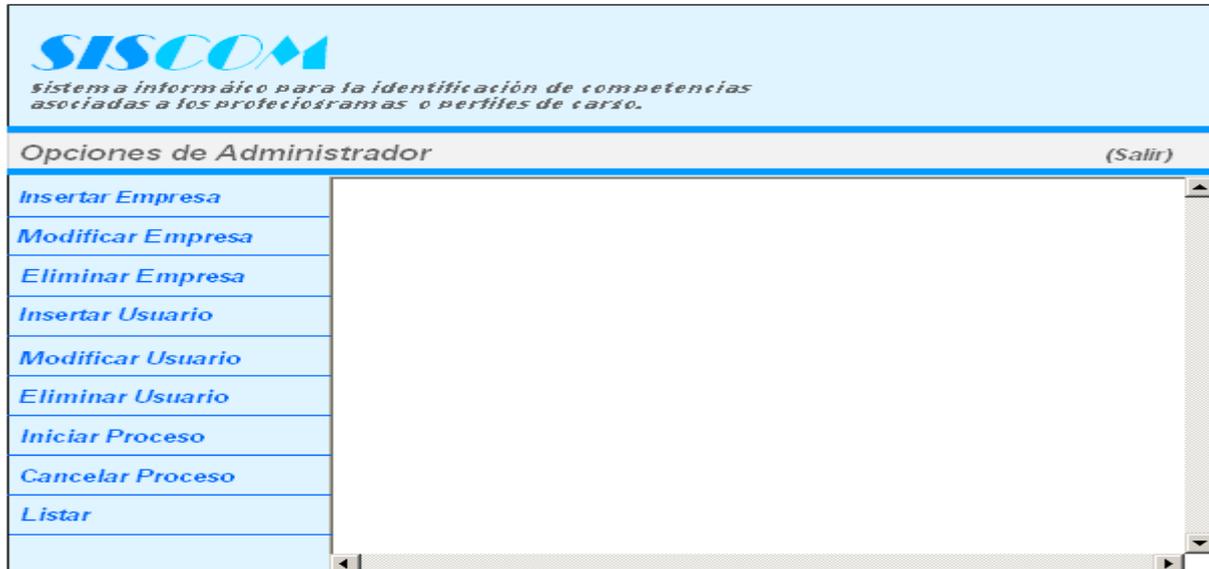
Dirección Particular: *

Ministerio: *

2.3 Pantalla para insertar una empresa nueva al sistema.

2.4 Pantalla para que el jefe de comité de competencia inserte un nuevo especialista.

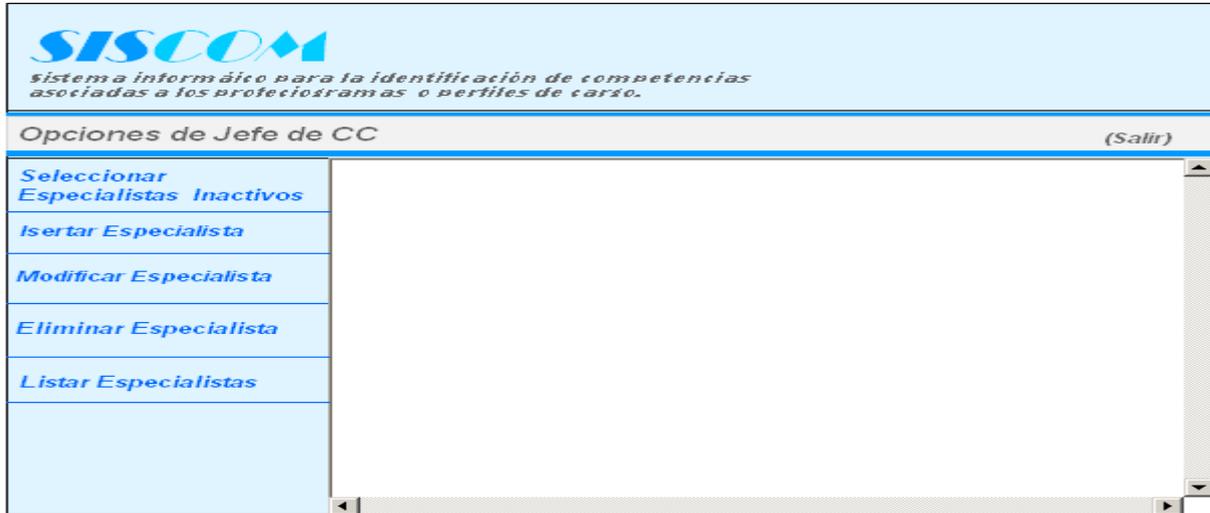
2.5 Pantalla para que los especialistas puedan responder los cuestionarios para seleccionarlos como expertos.



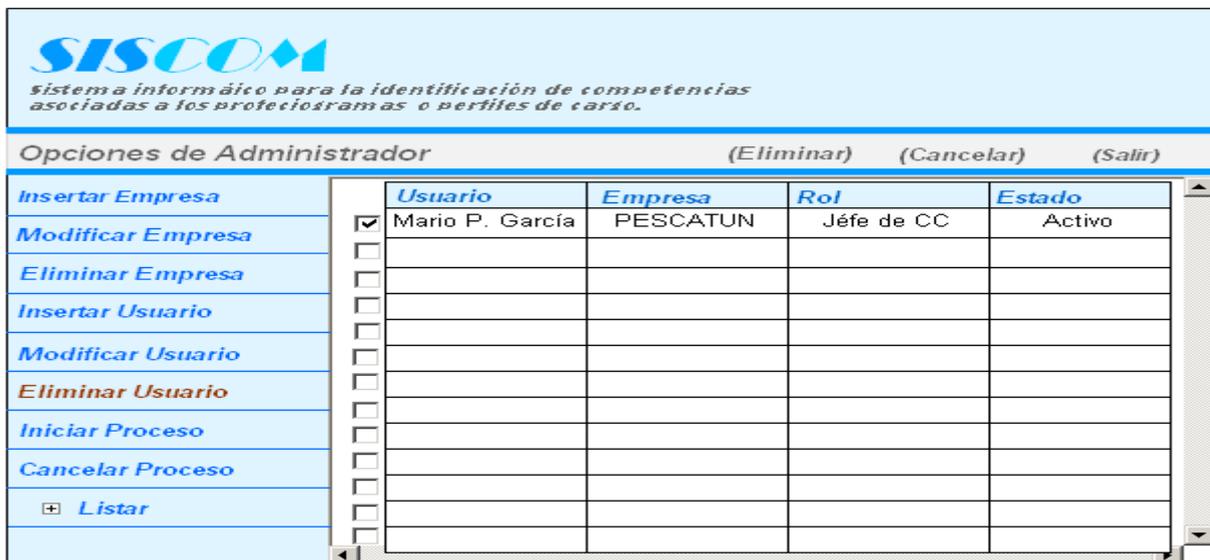
2.6 Pantalla con todas las opciones del administrador del sistema.



2.7 Pantalla con las opciones de la alta dirección de una empresa, con las opciones que esta tiene.



2.8 Pantalla para la gestión de los especialistas de la empresa, cuyo responsable es el jefe del comité de competencias.



2.9 Pantalla para eliminar un usuario del sistema.

GLOSARIO

- Gestión de Recursos Humanos (GRH), su objeto: las personas que trabajan portando el capital humano o la “capacidad de trabajo”