

Universidad de las Ciencias Informáticas

Facultad 3



Título: “Desarrollo del componente Competencias laborales del subsistema Capital Humano del Sistema Integral de Gestión Cedrux.”

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

Autor: Yaneisi Rodríguez Armas

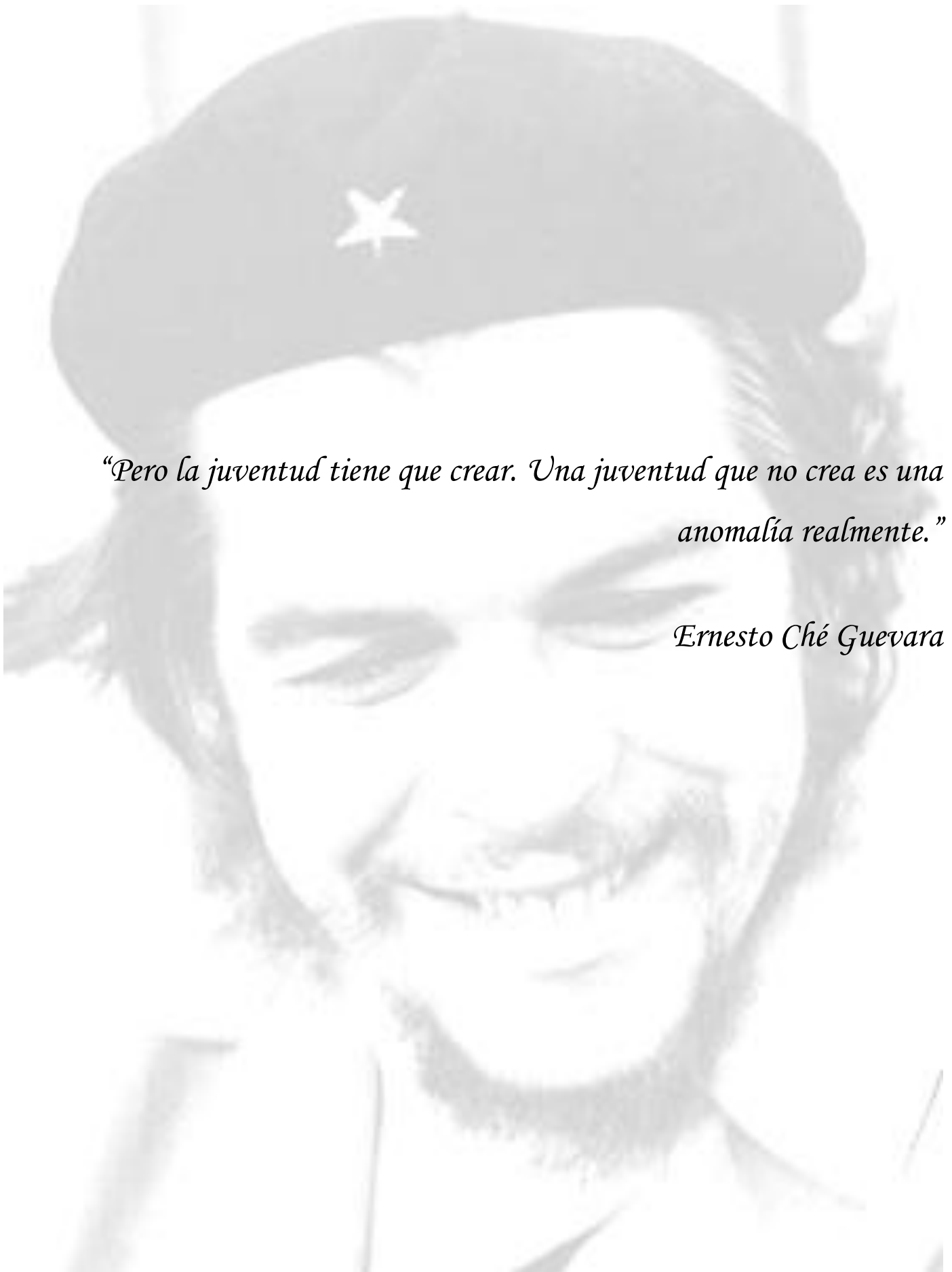
Tutor: Ing. Arnolis Salgueiro Arzuaga.

Co-tutor: Ing. Yoandry Pardo Escobar

Junio 2012

“Pero la juventud tiene que crear. Una juventud que no crea es una anomalía realmente.”

Ernesto Ché Guevara



DECLARACIÓN DE AUTORÍA

Declaro ser autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Yaneisi Rodríguez Armas
Autor

Ing. Arnolis Salgueiro Arzuaga
Tutor

Ing. Yoandry Pardo Escobar
Co-Tutor

DATOS DE CONTACTO

Datos de los tutores:

Ing. Arnolis Salgueiro Arzuaga

Ingeniero en Ciencias Informáticas. Graduado en la Universidad de las Ciencias Informáticas en el 2009. Actualmente se desempeña como jefe del proyecto Capital humano del Departamento Desarrollo de Productos, CEIGE, Facultad 3.

Correo: asalgueiro@uci.cu

Ing. Yoandry Pardo Escobar

Ingeniero en Ciencias Informáticas. Graduado en la Universidad de las Ciencias Informáticas en el 2010. Actualmente se desempeña como desarrollador de software de la línea Capital humano del Departamento Desarrollo de Productos, CEIGE, Facultad 3.

Correo: ypescobar@uci.cu

Dedicatoria

Esta tesis quiero dedicarla de forma muy especial a mis padres, por ser la razón de que haya llegado hasta aquí, por brindarme su apoyo en cada momento, por escucharme siempre, por su confianza, por haberme hecho la mujer que soy y por toda la felicidad que me han dado siempre.

A mi abuela por todo su cariño y por estar siempre tan orgullosa de mí.

A mi hermano que es lo más lindo que me han regalado mis padres, para que sepa que cuando se quiere se puede y para que le sirva de ejemplo y se haga un hombre de bien.

A Hazel por compartir conmigo todos estos años y darme los momentos más felices que viví aquí.

A mi bisabuela para que siempre sienta orgullo de mí.

A mi familia en general por su apoyo siempre y la confianza que han depositado en mí.

Agradecimientos

Quiero agradecerles a todas aquellas personas que estuvieron conmigo siempre estos cinco años.

Primeramente a mis padres por todo su apoyo, por su confianza, por sus consejos.

A mi papá por todos los viajes que tuvo que dar, por su tiempo y dedicación. Por la confianza que siempre depositó en mí.

A mi mamá por escucharme siempre, por darme apoyo en los momentos más difíciles, por ser mi amiga, mi consejera. Por traerme al mundo y por ser un ejemplo de mujer para mí.

A mi novio por estar a mi lado siempre, porque a pesar de que me dio algunos dolores de cabeza hizo mi vida estos cinco años más felices.

A Ariel por toda su ayuda, sin él no hubiera sido posible la realización de este trabajo.

A mis tutores Arnolis y Yoandry por estar dispuestos siempre cuando los necesité, por toda su ayuda y su tiempo.

A toda mi familia, por recibirme siempre con tanta alegría y estar pendientes siempre de mí.

A todas las muchachitas del apartamento, a mis compañeros de grupo y a todos aquellos que compartieron conmigo estos años de universidad que me hicieron pasar momentos inolvidables.

A toda la gente del proyecto, por brindarme su ayuda en los momentos que la necesité.

RESUMEN

Los sistemas de gestión son una solución robusta para aquellas empresas que buscan una solución universal a la centralización de su información. En la Universidad de las Ciencias Informáticas se desarrolla el Sistema Integral de Gestión Cedrux con particularidades de la economía cubana. Cedrux cuenta con un subsistema dedicado a la gestión del capital humano, el cual debería incluir un componente que tratara las Competencias laborales para facilitar la gestión del mismo, evitando deficiencias en procesos como los de Selección del personal, Evaluación y Capacitación y Desarrollo. El objetivo del presente trabajo es describir la implementación del componente Competencia laboral, para el cual se realizó un estudio de otros sistemas informáticos que incluyen este proceso, concluyendo que era necesario construir una solución propia. Las herramientas y tecnologías utilizadas para su desarrollo son las definidas por el Centro de Informatización de la Gestión Entidades adaptables a todas las entidades cubanas y cumpliendo con la política de migración a software libre.

El nuevo componente permitirá gestionar las competencias del personal y las que requiere una determinada estructura, ya sea un área, cargo o puesto de trabajo, logrando que se efectúe una correcta selección del personal, evaluación y capacitación y desarrollo.

Palabras claves: capital humano, Cedrux, competencia laboral, componente, gestión de capital humano.

ÍNDICE DE CONTENIDOS

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	4
1.1. Introducción.....	4
1.2. Conceptos generales.....	4
1.2.1. <i>Capital humano</i>	4
1.2.2. <i>Competencia laboral</i>	5
1.2.3. <i>Clasificación de las competencias</i>	5
1.2.4. <i>Gestión por competencias</i>	6
1.3. Sistemas informáticos similares.....	7
1.3.1. <i>Sistemas informáticos existentes en el mundo</i>	7
1.3.2. <i>Sistemas informáticos existentes en Cuba</i>	9
1.3.3. <i>Valoración de los sistemas estudiados</i>	10
1.4. Modelo de desarrollo de software	11
1.5. Tecnologías y herramientas para el desarrollo	12
1.5.2. <i>Lenguajes de programación</i>	15
1.5.3. <i>Gestor de base de datos</i>	16
1.5.4. <i>Navegador web</i>	16
1.5.5. <i>Servidor web</i>	17
1.5.6. <i>Herramientas para el modelado</i>	17
1.5.7. <i>IDE de desarrollo</i>	17
1.5.8. <i>Herramientas para el desarrollo colaborativo</i>	18
1.6. Normas y estándares de codificación	18
1.6.1. <i>Estándares de nomenclatura</i>	18
1.7. Conclusiones parciales.....	19
CAPÍTULO 2: PROPUESTA DE SOLUCIÓN.....	21
2.1. Introducción	21
2.2. Valoración del diseño propuesto por los analistas.....	21
2.2.1 <i>Requerimientos identificados</i>	21
2.3. Diagrama de clases	23
2.4. Modelo de datos	28

2.5. Implementación	30
2.5.1. Descripción del componente	30
2.5.2. Diagrama de componentes	30
2.5.3. Servicios utilizados.....	31
2.6. Interfaces de la aplicación.....	33
2.7. Diagrama de despliegue	37
2.8. Conclusiones parciales	39
CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN	40
3.1. Introducción	40
3.2. Métricas para la validación del diseño propuesto	40
3.2.1 Métrica Tamaño operacional de clase.....	41
3.2.2. Métrica Relaciones entre clases.....	44
3.3. Pruebas de software	48
3.4. Métodos de prueba	48
3.4.1. Pruebas de Caja blanca o Estructurales	48
3.4.2 Pruebas de Caja negra	54
3.5. Conclusiones parciales	58
CONCLUSIONES.....	60
RECOMENDACIONES	61
REFERENCIAS BIBLIOGRÁFICAS	62
GLOSARIO	69

ÍNDICE DE FIGURAS

Figura 1. Arquitectura de Sauxe	13
Figura 2. Diagrama de clases Gestionar clasificación de competencias.....	24
Figura 3. Diagrama de clases Gestionar competencias	25
Figura 4. Diagrama de clases Expediente de competencias de la persona.....	26
Figura 5. Diagrama de clases Profesiograma.....	28
Figura 6. Modelo de Datos de Competencia laboral.....	29
Figura 7. Diagrama de componentes.....	31
Figura 8. Interfaz Gestionar clasificación de competencia.	34
Figura 9. Interfaz Gestionar competencia.....	35
Figura 10. Interfaz Expediente de competencia de la persona.	36
Figura 11. Interfaz Profesiograma.	37
Figura 12. Diagrama de Red para PC Cliente con Disco Duro.	38
Figura 13. Diagrama de Red para PC Cliente sin Disco Duro.	38
Figura 14. Representación en % de los resultados de la evaluación de la métrica TOC en el atributo Responsabilidad.43	
Figura 15. Representación en % de los resultados de la evaluación de la métrica TOC en el atributo Complejidad de implementación.	44
Figura 16. Representación en % de los resultados de la evaluación de la métrica TOC en el atributo Reutilización.	44
Figura 17. Representación en % de los resultados de la evaluación de la métrica RC en el atributo Acoplamiento.	47
Figura 18. Representación en % de los resultados de la evaluación de la métrica RC en el atributo Complejidad de mantenimiento.	47
Figura 19. Representación en % de los resultados de la evaluación de la métrica RC en el atributo Reutilización.....	47
Figura 20. Representación en % de los resultados de la evaluación de la métrica RC en el atributo Cantidad de pruebas.	47
Figura 21. Grafo de flujo asociado al método cargarpuestotrabajo ().	51
Figura 22. Resultado de las Iteraciones	58
Figura 23. Interfaz Adicionar clasificación de competencia.	65
Figura 24. Interfaz Adicionar competencia.....	65

ÍNDICE DE TABLAS

Tabla 1. Descripción de los servicios brindados	32
Tabla 2. Descripción de los servicios que consume.....	32
Tabla 3. Tamaño operacional de clase (TOC).....	41
Tabla 4. Rango de valores para la evaluación técnica de los atributos de calidad relacionados con la métrica TOC.	42

Tabla 5. Resultados de la aplicación de la métrica TOC para cada clase del sistema.	42
Tabla 6. Relaciones entre clases (RC).	44
Tabla 7. Rango de valores de para la evaluación técnica de los atributos de calidad (Acoplamiento, Complejidad de Mantenimiento, Reutilización y Cantidad de Pruebas) relacionados con la métrica RC.	45
Tabla 8. Resultados de la aplicación de la métrica RC para cada clase del sistema.	46
Tabla 9. Caminos independientes del flujo.	51
Tabla 10. Caso de prueba para el camino básico 1.	52
Tabla 11. Caso de prueba para el camino básico 2.	52
Tabla 12. Caso de prueba para el camino básico 3.	53
Tabla 13. Posibles escenarios al adicionar una competencia.	55
Tabla 14. Resultado de las iteraciones de las pruebas de caja negra realizadas por el Grupo de Aseguramiento de la Calidad.	57

INTRODUCCIÓN

En la actualidad, debido a la complejidad y al ambiente competitivo que existe en el ámbito económico internacional de las empresas se requieren de herramientas que les proporcionen control y centralización de su información, con el fin de tomar las mejores decisiones para sus procesos y estrategias de negocios. Los sistemas de gestión son una solución robusta para aquellas empresas que buscan una solución universal a la centralización de su información.

En Cuba se ha identificado la necesidad de introducir un sistema informático que permita centralizar la gestión de las entidades presupuestadas del país. Sin embargo, las limitaciones técnico-económicas para adaptar sistemas de este tipo internacionalmente conocidos a las condiciones cubanas, impulsaron la idea de desarrollar un sistema de software de Planificación de Recursos Empresariales (ERP por sus siglas en inglés Enterprise Resource Planning) nacional, de esta manera y siguiendo los principios de independencia tecnológica surge Cedrux, un Sistema Integral de Gestión con particularidades de la economía cubana. Cedrux cuenta con un subsistema dedicado a la gestión del capital humano, el cual debería incluir un componente que tratara las Competencias laborales para facilitar la gestión del mismo, evitando deficiencias en procesos como los de Selección del personal, Evaluación y Capacitación y Desarrollo.

Las competencias laborales se han convertido en los últimos años en un marco de referencia para la organización y el desarrollo del personal en las organizaciones actuales, en las que el proceso de adaptarse y adelantarse a los cambios del entorno se presenta como un factor indispensable para alcanzar resultados positivos en el entorno empresarial. Así lo evidencian diversos países que han incorporado el concepto de Competencias laborales en sus modelos de Gestión del Capital humano (GCH).

Cuba, inmersa en una etapa de perfeccionamiento empresarial llevó a cabo un estudio que trataba a las competencias laborales como una fuente de ventaja competitiva, adoptando normas y resoluciones que introducen el enfoque de competencias en los procesos de capacitación profesional de los recursos humanos. A pesar de estos esfuerzos no se ha logrado automatizar estos procesos en la mayoría de las entidades, por lo que en algunas se continúa realizando manualmente y en otras se utilizan diferentes sistemas informáticos, en dependencia de las particularidades de cada entidad. Estos sistemas informáticos no cumplen con todas las políticas

vigentes en el país, entre las que se encuentran la reducción de importaciones y la migración hacia el software libre.

El hecho de que en la mayoría de las entidades las competencias laborales se realicen de forma manual dificulta la calidad y rapidez con que se desarrollan algunos procesos, ya que se necesita de mucho tiempo y coordinación para el traspaso de los documentos oficiales de un área a otra de la empresa y entre las mismas. Esto también provoca errores sistemáticos en los datos ofrecidos, ya sea por un mal control en las entidades o a la propia naturaleza humana. Otro problema que pudiera surgir con la realización manual del proceso competencias laborales serían las posibles manifestaciones de corrupción que se crearían en los niveles de dirección intermedia del sector empresarial, ya que muchas veces no se tiene en cuenta las capacidades de una persona para una determinada ocupación laboral, sino las relaciones personales. Debido a esto, dentro del subsistema Capital humano del Sistema Integral de Gestión Cedrux, procesos como la selección del personal y evaluación se ven afectados, al no contar con un componente que le proporcione las competencias laborales necesarias, tanto para una adecuada selección como para una correcta evaluación en función de las mismas, si no existe un servicio que brinde la posibilidad de mostrar las competencias de la persona y del puesto de trabajo, el proceso se verá limitado.

Por tanto, se plantea como **problema a resolver**:

La realización manual del proceso Competencias Laborales dificulta la gestión del capital humano en procesos vitales como los de Selección del personal, Evaluación y Capacitación y Desarrollo.

De esta forma queda definido como **objeto de estudio**: Sistemas de Gestión del Capital humano y el **campo de acción**: Sistemas de Gestión de competencias laborales.

Para darle solución al problema planteado, se tiene como **objetivo general** desarrollar el componente Competencias laborales para facilitar la gestión del capital humano evitando deficiencias en procesos vitales como los de Selección del personal, Evaluación y Capacitación y Desarrollo.

Los **objetivos específicos** son:

- ✓ Realizar un estudio de los requerimientos especificados en cuanto a la gestión de Competencias laborales de los trabajadores y su relación con subsistema de Capital humano del Sistema Integral de Gestión Cedrux.
- ✓ Desarrollar los requerimientos especificados para los procesos de Competencias laborales.

- ✓ Validar el resultado obtenido.

Con la finalidad de dar cumplimiento a los objetivos específicos se proponen las siguientes **tareas investigativas**:

- ✓ Caracterización de los procesos de Competencias laborales así como los relacionados con estos.
- ✓ Análisis de los sistemas existentes identificando sus características y deficiencias fundamentales.
- ✓ Justificación del uso de las tecnologías, lenguajes y herramientas propuestas para el desarrollo de la aplicación.
- ✓ Implementación de los requerimientos especificados para los procesos de Competencias laborales.
- ✓ Realización de pruebas al resultado obtenido utilizando prueba de Caja Negra y Caja Blanca.

Se tiene como **hipótesis**: Si se desarrolla el componente Competencia laboral disminuirán las dificultades para la gestión del capital humano en procesos vitales como los de Selección del personal, Evaluación y Capacitación y Desarrollo.

La estructura del trabajo de diploma será la siguiente:

CAPÍTULO 1: Fundamentación teórica: Describe el estado del arte de los sistemas que gestionan el capital humano, los principales conceptos relacionados, así como la fundamentación del uso de los lenguajes, herramientas y modelo de desarrollo.

CAPÍTULO 2: Propuesta de solución: Realiza una valoración crítica al diseño existente teniendo en cuenta su importancia para la implementación, propone un nuevo diseño y describe la implementación de la solución.

CAPÍTULO 3: Validación de la solución: Se muestran los resultados obtenidos de la validación del componente.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

En el presente capítulo se muestran algunas definiciones expuestas por diferentes autores sobre el concepto de Capital humano, Competencia laboral y la Gestión por competencia, así como su clasificación. Se hará mención de algunas tendencias actuales, tanto internacionales como nacionales, enfocadas a esta moderna forma de Gestión del Capital humano (gestión por competencias). Se explicará además el modelo de desarrollo realizado así como las herramientas y tecnologías que serán utilizadas para llevar a cabo la solución.

1.2. Conceptos generales

1.2.1. Capital humano

El capital humano comprende las capacidades para hacer trabajo, dadas por el conjunto de conocimientos, experiencias, habilidades, sentimientos, actitudes, motivaciones y valores, portados por las personas que trabajan. Comprende ciencia, economía y conciencia ética, como capacidades portadas por las personas de la organización laboral. (Santos, 2010)

El capital humano, según Tovar, se refiere al conocimiento útil para la empresa que poseen las personas y los equipos de la misma, así como a la capacidad para regenerarlo. Es decir, el capital humano es la parte del capital intelectual en que se recogen tanto las competencias actuales (conocimientos, habilidades y actitudes) como la capacidad de aprender y crear de las personas y equipos de trabajo que integran la organización. (Tovar, 2002)

Capital humano es conjunto de conocimientos, experiencias, habilidades, sentimientos, actitudes, motivaciones, valores y capacidad para hacer, portados por los trabajadores para crear más riquezas con eficiencia. Es, además, conciencia, ética, solidaridad, espíritu de sacrificio y heroísmo. (Norma Cubana 3000, 2007)

Como se puede observar existe variedad de definiciones sobre este tema, para el presente trabajo se tendrá en cuenta la definición aprobada por la NC 3000 que plantea que Capital humano implica no solo conocimientos, sino también experiencia, conciencia, habilidades, actitudes, motivaciones,

sentimientos, sacrificio y la capacidad de hacer por parte de los trabajadores cualquier tarea con eficiencia.

1.2.2. Competencia laboral

Según el Dr. Cuesta, las competencias “son habilidades, destrezas, aptitudes, actitudes, rasgos de personalidad, son estilos de dirección, formas de relaciones interpersonales, de experiencias que se adquieren durante toda la vida, tanto personal como laboral”. (Santos, 1999)

Competencia laboral es 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 de la organización, en correspondencias con las exigencias técnicas, productivas y de servicios. Es requerimiento esencial que esas competencias sean observables, medibles y que contribuyan al logro de los objetivos de la organización. (Norma Cubana 3000, 2007)

Las competencias laborales refieren a un nuevo concepto en la administración pública, que se define como la capacidad de una persona para desempeñar en diferentes contextos y con base en los requerimientos de calidad y resultados esperados en el sector público, las funciones inherentes a un empleo; capacidad que está determinada por los conocimientos, destrezas, habilidades, valores, actitudes y aptitudes que debe poseer y demostrar el empleado público.

Las competencias comunes, las comportamentales y las funcionales, además de los requisitos de estudio y experiencia, conforman las competencias laborales. (Vélez, 2005)

De manera general, siguiendo la Norma Cubana 3000, se puede decir que las competencias laborales son los conocimientos, características y habilidades que posee una persona para ejercer determinada ocupación, que deben ser medibles y que de estas depende el logro que pueda llegar a alcanzar una organización.

1.2.3. Clasificación de las competencias

Las competencias se pueden clasificar de diferentes formas, cada autor desde una perspectiva diferente de acuerdo a rasgos de personalidad, habilidades o conocimientos. Para el presente trabajo se tendrán en cuenta las definidas por el Dr. Armando Cuesta, el cual las clasifica en: (Santos, 1999)

- ✓ **Genéricas:** Estas pueden responder a cualquier puesto de trabajo o a la mayoría según su misión, objetivos y metas.
- ✓ **Específicas:** Estas se adecuan a puestos de trabajo, con objetivos de trabajo específicos y pueden cambiar de acuerdo a los escenarios y las estrategias que se traza la organización.
- ✓ **Compuestas:** Son aquellas en la que intervienen varias competencias, son más difíciles de dimensionar y dificultan el análisis de desempeño.
- ✓ **Simple:** Son las que se refieren a una sola competencia para evaluar, siendo fáciles de dimensionar y por lo tanto facilitan su evaluación.
- ✓ **Importantes:** Todas no tienen la misma importancia porque depende de las características del puesto de trabajo y de la misión, objetivos y estrategias del puesto y la organización.
- ✓ **Esenciales:** Debe poseer una persona independientemente del puesto que desempeña.

1.2.4. Gestión por competencias

La gestión por competencias es un modelo de gerenciamiento que permite evaluar las competencias específicas que requiere un puesto de trabajo de la persona que lo ejecuta, además, es una herramienta que permite flexibilizar la organización, ya que logra separar la organización del trabajo de la gestión de las personas, introduciendo a éstas como actores principales en los procesos de cambio de las empresas y finalmente, contribuir a crear ventajas competitivas de la organización. (Muñoz, et al., 2001)

Según la opinión del Dr. Armando Cuesta Santos, “el futuro de la gestión de competencias es sin dudas prometedor. Su complejidad exige rigor científico técnico para hacerla viable de modo sustentable cada vez con mayor eficacia y eficiencia. En correspondencia demanda profesionales bien preparados desde una óptica polivalente o multidisciplinaria. Ya no es novedad destacar que el recurso humano de las organizaciones determina la ventaja competitiva básica”. (Santos, 2010)

Realizándose en las empresas una eficiente GCH basándose en las competencias, los conocimientos y las habilidades de los trabajadores aumentarían en gran medida, creando así el desarrollo individual y organizacional, enfocándose en las necesidades de capacitación y apoyando a los sistemas de desarrollo del personal.

1.3. Sistemas informáticos similares

En la actualidad existen diferentes sistemas informáticos que integran y manejan los procesos de negocio de las empresas, dentro de los cuales se encuentra el proceso de competencia laboral, a continuación se describen algunos de ellos:

1.3.1. Sistemas informáticos existentes en el mundo

Microsoft Dynamics AX: es uno de los productos de software de ERP de Microsoft, perteneciente a la familia Microsoft Dynamics. Proporciona una base diseñada expresamente para cinco sectores, junto con funcionalidades de ERP completas y básicas para la gestión de finanzas, recursos humanos y operaciones. Microsoft Dynamics AX contiene un módulo dedicado a los recursos humanos que trata las competencias laborales como fuente fundamental para la contratación del personal en la empresa. (Microsoft Corporation, 2011)

Desventajas: Microsoft Dynamics AX es un software privativo adaptable solo para plataformas de Windows. Trata las competencias laborales únicamente para la contratación, no lo incluye para los procesos de evaluación y capacitación y desarrollo del personal. Incluye soporte para la gestión empresarial de operaciones en 36 países (funcionalidades específicas del país disponibles en un lanzamiento por fases), en los que no está incluida Cuba.

Oracle PeopleSoft: PeopleSoft es una plataforma conjunta de solución ERP para el área de producción que cubre todos los sectores industriales, destacándose en el área de Telecomunicaciones. Es un sistema líder en administración de recursos humanos. Oracle PeopleSoft cuenta con una solución de talento que se integra completamente con los principales procesos de recursos humanos. La gestión de talento de Oracle PeopleSoft cubre todos los módulos en cuanto a talento se refiere, entre los cuales se encuentran la planificación, el reclutamiento, el desarrollo profesional, la planificación, las revisiones de talento y la medición y presentación de informes. La gestión de talentos trata criterios de competencias, donde se tienen en cuenta las capacidades del personal y las experiencias en la vida profesional como ayuda para los procesos mencionados anteriormente. (Zoder, 2010)

Desventajas: Oracle PeopleSoft es un software privativo adaptable solo para plataformas de Windows. Incluye soporte para la gestión empresarial en algunos países en los que no está incluida Cuba.

TalentSoft: El mismo es una solución integrada que permite combinar la gestión de las evaluaciones, la formación y la remuneración. También permite al empleado involucrarse más en la empresa. Con esta herramienta los usuarios tendrán un marco de evaluación (últimos objetivos evaluados, objetivos para el próximo año y los intereses de formación del personal), que incorporará de manera automática las competencias requeridas para el puesto del empleado quienes estarán evaluados según todos estos elementos. Los administradores podrán analizar la evolución, la movilidad o las distancias entre las competencias del empleado y las definidas por el repositorio de empleos y competencias. A parte de las evaluaciones anuales, los empleados tendrán acceso a las ofertas de puestos dentro de la empresa y al repositorio de empleos y competencias. (Talentsoft, 2012)

Desventajas: SAGE France es un software privativo adaptable solo para plataformas de Windows. Las competencias ya están establecidas dentro de la herramienta, lo que no permite que se modifiquen o agreguen nuevas.

El ERP SAP (Sistemas, Aplicaciones y Productos): SAP contiene un paquete de recursos humano que incluye el proceso Gestión del talento que proporciona soporte al personal en todas las fases de su empleo, desde su contratación hasta su formación, desarrollo y retención. Permite definir un catálogo de competencias clave a lo largo de la arquitectura de su trabajo y cultivar la cantera de talentos. Permite a los empleados actualizar información personal y controlar procesos administrativos. De este modo, el departamento de recursos humanos puede centrarse en proyectos que generen valor en lugar de tener que centrarse en la administración. Aún con estas características el módulo competencias laborales de SAP no puede ser aprovechado en la mayoría de las empresas de Cuba debido a su alto costo, además de estar diseñado para economías totalmente diferentes. (Genix, et al., 2011)

Desventajas: Está enfocado a un sector específico, desarrollado para funcionar según las características de otro sector empresarial diferentes a los de Cuba. Es un software propietario, lo que

implica altos costos por conceptos de licencia, capacitación y soporte. Adaptable solo para plataformas de Windows.

1.3.2. Sistemas informáticos existentes en Cuba

Cuba no está ajena a todos los cambios que han acontecido en el transcurso de los años en las empresas, también se han tratado y estudiado las competencias laborales viendo el Capital humano y la Gestión del conocimiento como principales fuentes de ventaja competitiva. En marzo de 1999, el Ministerio de Trabajo y Seguridad Social (MTSS) adopta la Resolución 21/99, donde se regula y norma los procesos Capacitación Profesional de los Trabajadores en la empresas inmersas en el perfeccionamiento empresarial, introduciéndole de esta forma, de manera oficial en el país, el enfoque de competencias en los procesos de capacitación profesional de los recursos humanos.

El 22 de mayo de 2007 se llevó a cabo el lanzamiento de la familia de Normas Cubanas (NC) 3000, 3001, 3002 conocidas como Sistema de Gestión Integral de Capital humano. Estas normas cubanas son de buenas prácticas, no tienen un carácter jurídico y sirven de guía o patrón de orientación para que cada organización diseñe e implemente su propio Sistema de Gestión Integrada de Capital humano, están dirigidas fundamentalmente a la mejora continua y una de ella es además certificable para aquellas entidades laborales con la determinación de alcanzar y demostrar una gestión integrada de capital humano. El punto 4.2 (Requisitos vinculados a las competencias laborales) de la NC 3001 se refiere a las competencias laborales donde se establece, que debe existir un grupo encargado de identificar y proponer a la alta dirección las competencias distintivas tanto de la organización como de las diferentes actividades que en esta se realicen. La certificación del proceso se realiza por la propia organización, la que mediante documentos constituidos debe dejar plasmado el cumplimiento de las competencias establecidas.

A partir de estos preceptos diversas empresas, sobre todo las que se encontraban desarrollando el proceso de Perfeccionamiento empresarial, desde finales de la primera década del presente siglo iniciaron un proceso de adaptación de los métodos tradicionales de GCH a esta novedosa forma de gestión y algunas de ellas lograron certificarse, entre las que se encuentran la empresa TRAYCO (Transporte de la Construcción) de Pinar del Río, la Empresa de Ingeniería y Proyectos (EIPP) de La Habana, la Empresa Mecánica de Níquel de Holguín y la Empresa Comercializadora de Camagüey.

El sistema **RH-Expert** desarrollado por la empresa Datazucar de Azcuba es hoy el único producto oficial del perfil de la gestión del capital humano en Cuba ideado sobre normas específicas relativas a las competencias. Es un software que implementa criterios de la Gestión por competencias, pero que no abarca todos los procesos que esta incluye. Ofrece soluciones para gestionar lo concerniente a los expedientes laborales, la evaluación del desempeño, seguridad y salud, y capacitación. Hasta la fecha es un producto único con estas características y enfoque en el mercado nacional. (Nicola, 2011).

El sistema prevé un entorno del trabajo multiusuario, se desarrolla en Delphi7 y con un Sistema Gestor de Base de Datos SQL Server 2000. Está desarrollada solo para funcionar en plataformas de Windows.

En el 2005 el Centro de Información y Gestión Tecnológica (CIGET) de la Delegación Territorial del CITMA en Cienfuegos desarrolló el sistema **RH-CITMA**, con similares propósitos, pero que nunca llegó a utilizarse.

1.3.3. Valoración de los sistemas estudiados

Aún cuando en el ámbito internacional se encuentran en explotación varias aplicaciones que gestionan las competencias, se observó que las soluciones estudiadas brindan un trabajo incompleto, ya que algunas solo utilizan las competencias para la contratación y otras traen las competencias incluidas de forma estática en la herramienta, imposibilitando su gestión. Otra desventaja que presentan estos sistemas es que son soluciones propietarias, lo que implica que se realicen pagos por conceptos de licencias, capacitación y soporte, aspectos que no son factibles por las condiciones económicas de Cuba. Están desarrolladas para sistemas empresariales con características diferentes a las entidades cubanas e implementadas para funcionar solo en plataformas de Windows, lo cual atenta contra el proceso de migración hacia software libre en el que está enmarcada Cuba. En el ámbito nacional a pesar de existir un sistema que gestiona las competencias se pudo apreciar que a diferencia de Cedrux, RH-Expert es una aplicación ideada para propósitos concretos de la gestión del capital humano.

Finalmente se puede concluir que estos sistemas no son compatibles con el sistema integral de gestión Cedrux, por lo que ninguna de las herramientas estudiadas que gestionan las competencias

puede ser adaptada al mismo, aunque sí sirvieron de apoyo en la concepción de algunas interfaces y funcionalidades que por la complejidad del negocio no quedaban del todo claras en los documentos que se tuvieron en cuenta para el desarrollo del componente.

1.4. Modelo de desarrollo de software

Un modelo de desarrollo de software establece el orden en el que se realizarán las acciones en un proyecto, es una descripción simplificada de un proceso del software que presenta una visión de ese proceso. Estos modelos pueden incluir actividades que son parte de los procesos y productos de software y el papel de las personas involucradas en la ingeniería de software. Son de vital importancia ya que proporcionan estabilidad, control y organización.

El modelo de desarrollo de software propuesto describe la secuencia de actividades de alto nivel para la construcción y desarrollo de soluciones. Se logra con la combinación entre los modelos orientado a componentes e iterativo e incremental (Miniet, et al., 2009), el mismo presenta las siguientes características:

- ✓ **Centrado en la arquitectura:** La arquitectura determina la línea base, los elementos de software estructurales a partir de los elementos de la arquitectura de negocio. Interviene en la gestión de cambios y diseña la evolución e integración del producto. La arquitectura orienta las prioridades del desarrollo y resuelve las necesidades tecnológicas y de soporte para el desarrollo.
- ✓ **Orientado a componentes:** Las iteraciones son orientadas según la significación arquitectónica de los componentes, los mismos son abstracciones arquitectónicas de los procesos de negocio y requisitos asociados que modelan, el componente es la unidad de medición y ordenamiento de las iteraciones.
- ✓ **Iterativo e incremental:** Las iteraciones son planificadas y coordinadas con el equipo de arquitectura, los clientes y la alta gerencia. Cada iteración constituye el desarrollo de componentes, los cuales son integrados al término de la iteración, permitiendo de esta manera la evolución incremental del producto.
- ✓ **Ágil y adaptable al cambio:** El desarrollo de las partes formaliza solamente las características principales de la solución priorizando los talleres y las comunicaciones entre las

personas, los clientes y funcionales están involucrados en el proyecto y poseen parte de la responsabilidad del éxito del mismo y los cambios son conciliados semanalmente, discutidos y aprobados.

1.5. Tecnologías y herramientas para el desarrollo

El desarrollo de la solución se realizará utilizando el marco de trabajo Sauxe, implementado por el Departamento de Tecnología del Centro de Informatización de la Gestión de Entidades (CEIGE), el cual contiene un conjunto de componentes reutilizables que provee la estructura genérica y el comportamiento para una familia de abstracciones, logrando una mayor estandarización, flexibilidad, integración y agilidad en el proceso de desarrollo.

El primer elemento que debe regir una selección adecuada de alcance de un marco tecnológico radica en las restricciones de diseño que el mismo asume, basados en las tecnologías, la capacidad técnica del equipo de desarrollo, los intereses jurídicos mercantiles así como la infraestructura tanto de la organización productora como cliente de los productos que el mismo facilitará desarrollar. Contiene un conjunto de componentes reutilizables que provee la estructura genérica y el comportamiento para una familia de abstracciones, logrando una mayor estandarización, flexibilidad, integración y agilidad en el proceso de desarrollo. Cuenta con una arquitectura en capas como se muestra en la figura 1 reutilizando las siguientes tecnologías libres (Baryolo, 2008):



Figura 1. Arquitectura de Sauxe

1.5.1. Marco de trabajo

Las tecnologías libres con las que cuenta el marco de trabajo Sauxe son:

ExtJS 2.2: Es una librería Java Script código abierto de alto rendimiento para la creación y desarrollo de aplicaciones web dinámicas. Su potencia radica en la rica colección de componentes para el diseño de interfaces de usuarios del lado del cliente haciendo uso extensivo de Ajax¹. Permite la creación de aplicaciones complejas utilizando componentes predefinidos. Es extensible para la gran mayoría de los navegadores, evitando el tedioso problema de validar el código para cada uno de estos. Entre sus principales ventajas se encuentra el balance entre Cliente-Servidor, distribuyendo la carga de procesamiento en el último, y este al tener menor carga, maneja los clientes de manera más eficiente. La comunicación asíncrona permite el intercambio de información con el servidor sin necesidad de pedirle una acción al usuario, dando la libertad de cargar la información sin que este lo note. (Cross-Browser)

¹ Técnica de desarrollo web para crear aplicaciones interactivas.

Doctrine 1.2.1: Doctrine es un potente y completo sistema mapeador relacional de objetos (ORM) para PHP con un capa de abstracción de bases de datos (DBAL) incorporado que permite exportar una base de datos a sus clases correspondientes y viceversa, o sea, a partir de las clases creadas y siguiendo las especificaciones de ORM, generar las tablas de la base de datos. Se encuentra en la parte superior de una poderosa Capa de Abstracción de Base de Datos (DBAL). Una de sus principales características es la opción de escribir las consultas de base de datos en un objeto con una propiedad orientada al dialecto SQL llamada Doctrine Query Language (DQL), inspirada en Hibernate HQL. Esto proporciona a los desarrolladores una poderosa alternativa a SQL que mantiene la flexibilidad, sin necesidad de la duplicación de código innecesaria. (Cross-Browser)

Ventajas que facilitan tareas comunes y de mantenimiento:

- ✓ Reutilización: La principal ventaja que aporta un ORM es la reutilización permitiendo invocar a los métodos de un objeto de datos desde distintas partes de la aplicación e incluso desde diferentes aplicaciones.
- ✓ Encapsulación: La capa ORM encapsula la lógica de los datos pudiendo hacer cambios que afectan a toda la aplicación únicamente modificando una función.
- ✓ Portabilidad: Utilizar una capa de abstracción nos permite cambiar en mitad de un proyecto de una base de datos MySQL a una Oracle sin ningún tipo de complicación. Esto es debido a que no utiliza una sintaxis MySQL, Oracle o SQLite para acceder al modelo, sino una sintaxis propia del ORM utilizado que es capaz de traducir a diferentes tipos de bases de datos.
- ✓ Seguridad: Los ORM suelen implementar mecanismos de seguridad que protegen la aplicación de los ataques más comunes como SQL Injection.
- ✓ Mantenimiento del código: Gracias al correcto ordenamiento de la capa de datos, modificar y mantener el código es una tarea sencilla

Zend Framework 1.7: Se trata de un framework para desarrollo de aplicaciones web y servicios web con PHP, brinda soluciones para construir sitios web modernos, robustos y seguros. Además es Open Source (código abierto) y trabaja con PHP 5. A diferencia de CakePHP que trabaja con PHP 4 y PHP 5. Este framework está formado por una serie de métodos estáticos y componentes que

usarán estos métodos. Los componentes son varios y variados y aunque alguno es posible que no se use nunca, hay otras que puede que se usen hasta la saciedad, por ejemplo el componente para la BD. Entre los componentes de vital importancia se encuentran: Zend_Config para temas de configuración de aplicaciones web, Zend_Db para tratar con bases de datos, Zend_Search o Zend_Feed, entre otros. (Pedro Boda, et al., 2009)

1.5.2. Lenguajes de programación

Un lenguaje de programación permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos de hardware y software existentes. (Marin, 2008)

PHP 5.2.6: Es un lenguaje interpretado de propósito general ampliamente usado, diseñado especialmente para desarrollo web y que puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. (Rasmus Lerdorf, et al., 2006)

Entre las características que posee este lenguaje y que lo convierten en una potente herramienta están:

- ✓ Es un software de código abierto.
- ✓ Soporta muchas bases de datos entre las que se encuentran (MySQL y PostgreSQL).
- ✓ Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- ✓ Ofrece una solución simple y universal para las paginaciones dinámicas Web de fácil programación.

JavaScript 1.1: es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas Web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. Al contrario que Java, JavaScript no es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de Herencia, es más bien un lenguaje basado en prototipos, ya que las nuevas clases se

generan clonando las clases base (prototipos) y extendiendo su funcionalidad. Para interactuar con una página Web se provee al lenguaje JavaScript de una implementación del DOM. JavaScript es un lenguaje con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página Web y en programas más grandes, orientados a objetos mucho más complejos. Con JavaScript se puede crear diferentes efectos e interactuar con los usuarios. JavaScript es soportado por la mayoría de los navegadores como Internet Explorer, Netscape, Opera, Mozilla Firefox, entre otros. (2001)

1.5.3. Gestor de base de datos

PostgreSQL 8.3: PostgreSQL es un sistema gestor de base de datos objeto-relacional, bajo licencia BSD. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre. Es el sistema de gestión de bases de datos de código abierto más avanzado del mundo y en sus últimas versiones posee muchas características que solo se podían ver en productos comerciales de alto calibre.

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

Se ejecuta en casi todos los principales sistemas operativos: Linux, Unix, BSDs, Mac OS, Beos, Windows, etc. Documentación muy bien organizada, pública y libre, con comentarios de los propios usuarios.

Comunidades muy activas, varias comunidades en castellano. Soporte nativo para los lenguajes más populares del medio: PHP, C, C++, Perl, Python, etc. Soporte de todas las características de una base de datos profesional (triggers, store procedures, funciones, secuencias, relaciones, reglas, tipos de datos definidos por usuarios, vistas, vistas materializadas, etc.). Altamente adaptable a las necesidades del cliente.

1.5.4. Navegador web

Mozilla Firefox 2.17: Es un navegador de software libre. Entre sus características se encuentran que presenta una forma rápida y eficiente de navegar por la web, que le permite abrir varias páginas en una misma ventana mediante el empleo de pestañas separadas. Contiene un Plugin Firebug que se

utiliza para ver los errores del código. Firefox es un navegador multiplataforma y está disponible en varias versiones de Microsoft Windows, Mac OS X, GNU/Linux y algunos sistemas basados en Unix. Su código fuente es software libre, publicado bajo una triple licencia GPL/LGPL/MPL. (López, 2004)

1.5.5. Servidor web

Apache 2.2.9: Corre en una multitud de sistemas operativos, lo que lo hace prácticamente universal. Es una tecnología gratuita de código fuente abierta. Es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor. Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Tiene una alta configurabilidad en la creación y gestión de logs. (Ciberaula International Training, S.L., 2010)

1.5.6. Herramientas para el modelado

Visual Paradigm 6.4: Es una herramienta CASE de modelado que utiliza UML como lenguaje de modelado profesional y que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite realizar ingeniería tanto directa como inversa. La herramienta es colaborativa, es decir, soporta múltiples usuarios trabajando sobre el mismo proyecto; genera la documentación del proyecto automáticamente en varios formatos como HTML, PDF y permite control de versiones. Dentro de sus características principales se puede destacar su robustez, usabilidad y portabilidad. Para el desarrollo de este trabajo se escogió como herramienta CASE de modelado al Visual Paradigm porque tiene un coste favorable y realiza de una forma íntegra los planes de construcción del software. Permite ingeniería inversa, del modelo físico se puede llegar al modelo lógico. Genera de forma automática códigos desde diagramas, además brinda la posibilidad de generar documentación.

1.5.7. IDE de desarrollo

Los Entornos de Desarrollo Integrado, IDE por sus siglas en inglés (Integrated Development Environment), son un entorno de programación que ha sido empaquetado como un programa de aplicación, o sea, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones

existentes. El IDE seleccionado para el trabajo con los lenguajes de programación para la realización del sistema es:

NetBeans 6.9: Es una 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 extenderlo. El IDE NetBeans es un producto libre y gratuito sin restricciones de uso. (Oracle Corporation, 2011)

1.5.8. Herramientas para el desarrollo colaborativo

SVN (Subversión) 1.6.6: Es un software de sistema de control de versiones diseñado específicamente para reemplazar al popular CVS, el cual posee varias deficiencias. Es software libre bajo una licencia de tipo Apache/BSD y se le conoce también como SVN por ser ese el nombre de la herramienta de línea de comandos. Una característica importante de Subversión es que, a diferencia de CVS, los archivos versionados no tienen cada uno un número de revisión independiente. En cambio, todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en cierto punto del tiempo. (Osmosis Latina, 2005)

1.6. Normas y estándares de codificación

Los estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. Este apartado constituye una visión de la estandarización del código que se utilizó en el desarrollo de la aplicación desde un punto de vista arquitectónico. Los estándares de codificación en el marco del ERP permiten una mejor integración entre las líneas de producción, estableciendo un modelo de desarrollo que conlleva a lograr un código más legible y reutilizable, de tal forma que se pueda facilitar su mantenimiento a lo largo del tiempo. Entre las normas y estándares de codificación utilizados en la implementación de este producto se encuentran:

1.6.1. Estándares de nomenclatura

En el estándar de nomenclatura se define cómo serán nombradas las clases atendiendo entre otras cosas al tipo de clases que sean. Los nombres de las clases comienzan con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación

“PascalCasing”. Con solo leerlo se reconoce el propósito de la misma. Las clases controladoras después del nombre llevan la palabra: “Controller” y las clases de los modelos que están dentro de “Busines” llevan la palabra “Model”, las de los dominios reciben el nombre de las tablas de la base de datos y las que se encuentran dentro de los dominios bases comienzan con la palabra “Base” seguido del nombre de la tabla en la Base de Datos.

- ✓ Clases controladoras.

Ejemplo: GestionarcompetenciaController

- ✓ Clases de los modelos

Ejemplo: NomCompetenciaModel

- ✓ Domain (Dominio)

Ejemplo: NomClasificacionCompetencia

- ✓ Generated (Dominio bases)

Ejemplo: BaseNomCompetencia

El nombre a emplear para las funciones se escribe con la primera palabra en minúscula, en caso de que sea un nombre compuesto se empleará notación camelCasing, y con solo leerlo se reconoce el propósito de la misma.

En caso de ser una acción de la clase controladora se le pone el nombre y seguida la palabra: “Action”

Ejemplo: gestionarcompetenciaAction

1.7. Conclusiones parciales

En este capítulo se hizo mención de los conceptos necesarios para un mejor entendimiento de la investigación científica en cuestión. Se realizó un estudio de las tendencias actuales, tanto nacionales como internacionales, enfocadas a la Gestión de capital humano por competencias, concluyendo que:

- ✓ Estos sistemas no cumplen con todo lo establecido en las normas cubanas para la gestión del capital humano referente a las competencias laborales.
- ✓ Están desarrollados para condiciones diferentes a las de Cuba, enfocadas a un sector específico.

- ✓ Están implementados con programas que son propietarios atentando contra el proceso de migración hacia software libre en el que está enmarcado el país.

Después de haber realizado el estudio de los sistemas existente se determinó la necesidad de crear una solución propia que responda a las características de las entidades cubanas y cumpla con el paradigma de independencia tecnológica por la que aboga el país. Se realizó una descripción de las tecnologías, modelo de desarrollo y herramientas con las cuales se va a desarrollar la solución para lograr un mejor entendimiento de estas, demostrando que su utilización fortalece el paradigma de independencia tecnológica por el cual apuesta el país.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

2.1. Introducción

En el presente capítulo se propone modelar la solución, donde se tratan los elementos fundamentales incluidos en el desarrollo del componente. Para ello se comienza realizando un análisis de los requisitos funcionales, luego se muestra cómo se aprovechó la arquitectura y las posibilidades proporcionadas por el framework y las librerías utilizadas en la programación de la aplicación. Para una mayor comprensión de las funcionalidades de los componentes implementados se realizaron las descripciones de clases así como las operaciones utilizadas.

2.2. Valoración del diseño propuesto por los analistas

Teniendo en cuenta la importancia del diseño propuesto para la implementación del sistema se realizará una valoración crítica del mismo. Los artefactos entregados por los analistas fueron la Descripción de los Requisitos Funcionales, el diagrama entidad relación, los diagramas de clases del diseño y el diagrama de componentes. Una entrevista con los analistas arrojó que el diseño entregado no se ajustaba con las nuevas especificaciones, ya que los requerimientos entregados habían sufrido cambios, provocando ambigüedades en los diagramas, malas interpretaciones y falta de información para los desarrolladores. Además, como no se refleja con claridad la relación existente entre las clases, resulta poco probable conocer con exactitud la repercusión que pueda traer la modificación en una de ellas para el resto.

De forma general los diagramas de clases del diseño anterior no reflejaban el uso correcto de estereotipos web, carecían de relaciones entre clases esenciales para una correcta comprensión y desarrollo de la solución, así como funcionalidades que estas debían tener.

Al detectarse errores en dichos artefactos y por la importancia que representa poseer un diseño actualizado, se propone la modificación del mismo partiendo de los requerimientos entregados.

2.2.1 Requerimientos identificados

Los requerimientos funcionales identificados y descritos por los analistas de acuerdo a los objetivos planteados y que posibilitaron la identificación de las clases y las funcionalidades a implementar son los siguientes:

RF1 Gestionar clasificación de competencias.

RF 1.1 Adicionar clasificación.

RF 1.2 Modificar clasificación.

RF 1.3 Eliminar clasificación.

RF 1.4 Listar clasificaciones.

RF 2 Gestionar competencias.

RF 2.1 Adicionar competencia.

RF 2.2 Modificar competencia.

RF 2.3 Eliminar competencia.

RF 2.4 Listar competencias.

RF 3 Gestionar expediente de competencias de la persona.

RF 3.1 Adicionar competencia a la persona.

RF 3.2 Modificar competencia de la persona.

RF 3.3 Eliminar competencia de la persona.

RF 3.4 Listar competencias de la persona.

RF 4: Gestionar profesiograma.

RF 4.1 Adicionar competencia por estructura.

RF 4.2 Modificar competencia por estructura.

RF 4.3 Eliminar competencia por estructura.

RF 4.4 Listar competencias por estructura.

Los requerimientos funcionales establecen los comportamientos del sistema. Definen lo que el sistema tiene que hacer y los servicios que debe proporcionar al usuario. Para el desarrollo del componente Competencia laboral quedaron establecidos previo al análisis realizado 4 agrupaciones de requerimientos funcionales que permitirán una mejor comprensión para su posterior implementación. La primera agrupación de requerimientos “Gestionar clasificación de competencias” permite gestionar las clasificaciones de competencia definidos por la entidad, la agrupación “Gestionar competencias” se encarga de gestionar todas las competencias definidas por la empresa, este es utilizado posteriormente en el expediente de competencias de la persona y en el profesiograma para asignarle competencias a determinada área, cargo o puesto de trabajo. La

agrupación “Gestionar expediente de competencias de la persona” está relacionada con la gestión de las competencias de la persona y será utilizado para la selección del personal. Por otro lado la agrupación “Gestionar profesiograma” permite gestionar las competencias que requiere cierta área, cargo o puesto de trabajo de la entidad.

2.3. Diagrama de clases

Los diagramas de clases según la clasificación UML son diagramas de estructura estática donde la representación de los requerimientos se lleva a cabo a través de las clases del sistema y sus interrelaciones. Representan una abstracción del dominio de modo que es formalizado el análisis de conceptos y constituyen el pilar básico del modelado, mostrando en términos generales qué debe hacer el sistema.

En la figura 2 se muestra el diagrama de clases del escenario Gestionar clasificación de competencias, en el cual se le da solución a los requerimientos relacionados con la gestión de las clasificaciones de las competencias. Las clases `gestionarClasificacion.js` y `gestionarclasificacion.phtml` conforman la capa arquitectónica de presentación, la clase `gestionarClasificacionController` maneja la comunicación entre la vista y el modelo, las clases `Model` son las encargadas de la lógica del negocio y como encargada al acceso de los datos se encuentra `NomClclasificacioncomp` y la clase `Base` de la cual extiende la misma.

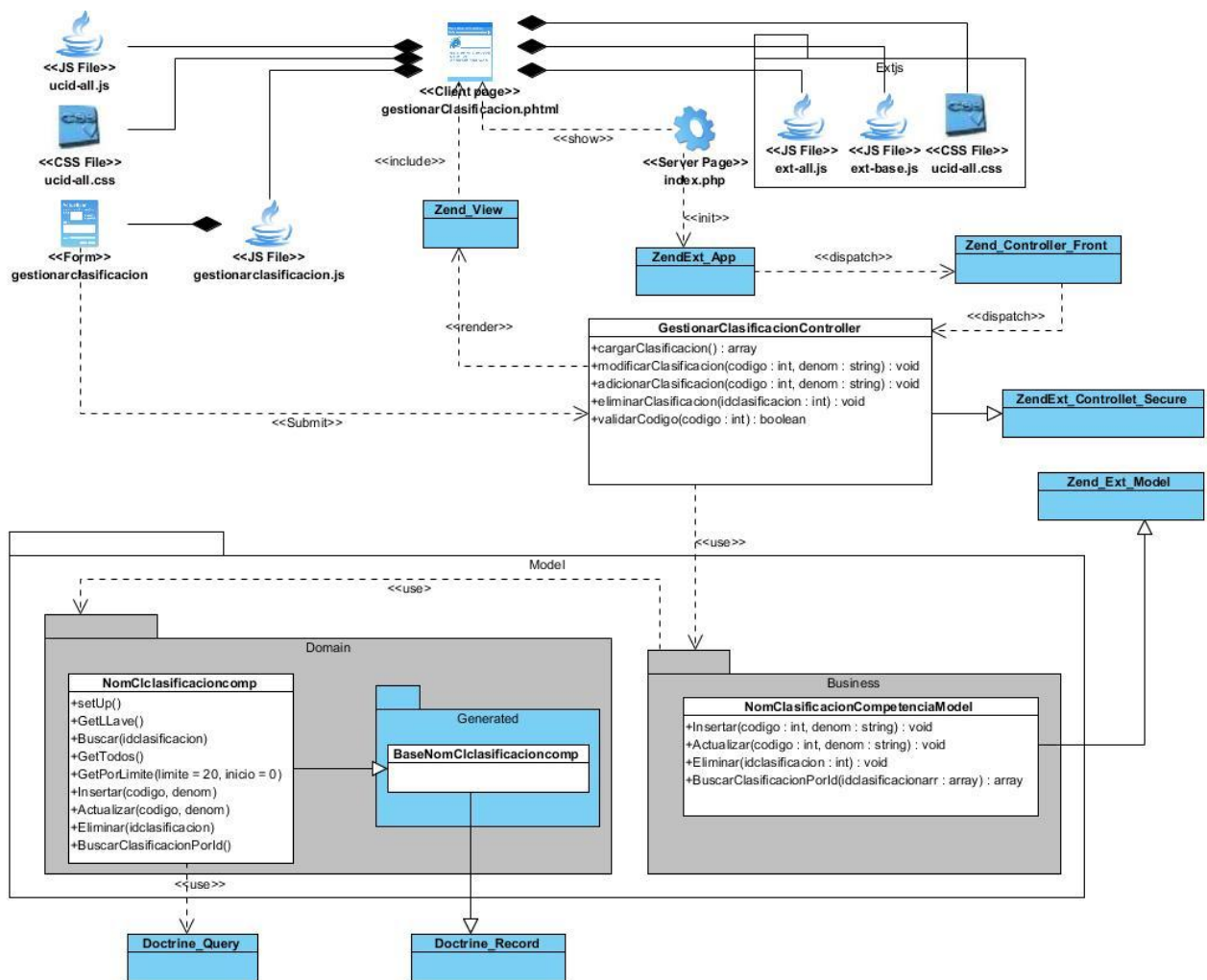


Figura 2. Diagrama de clases Gestionar clasificación de competencias

En la figura 3 se muestra el diagrama de clases del escenario Gestionar competencias, en el cual se le da solución a los requerimientos relacionados con la gestión de las competencias. Las clases gestionarNomCompetencia.js y gestionarCompetencia.phtml conforman la capa arquitectónica de presentación, la clase GestCompetenciaController maneja la comunicación entre la vista y el modelo, las clases Model son las encargadas de la lógica del negocio y como encargadas al acceso de los datos se encuentran NomCclasificacioncomp, DatClcompetencia y las clases Base de la cual extiende la misma.

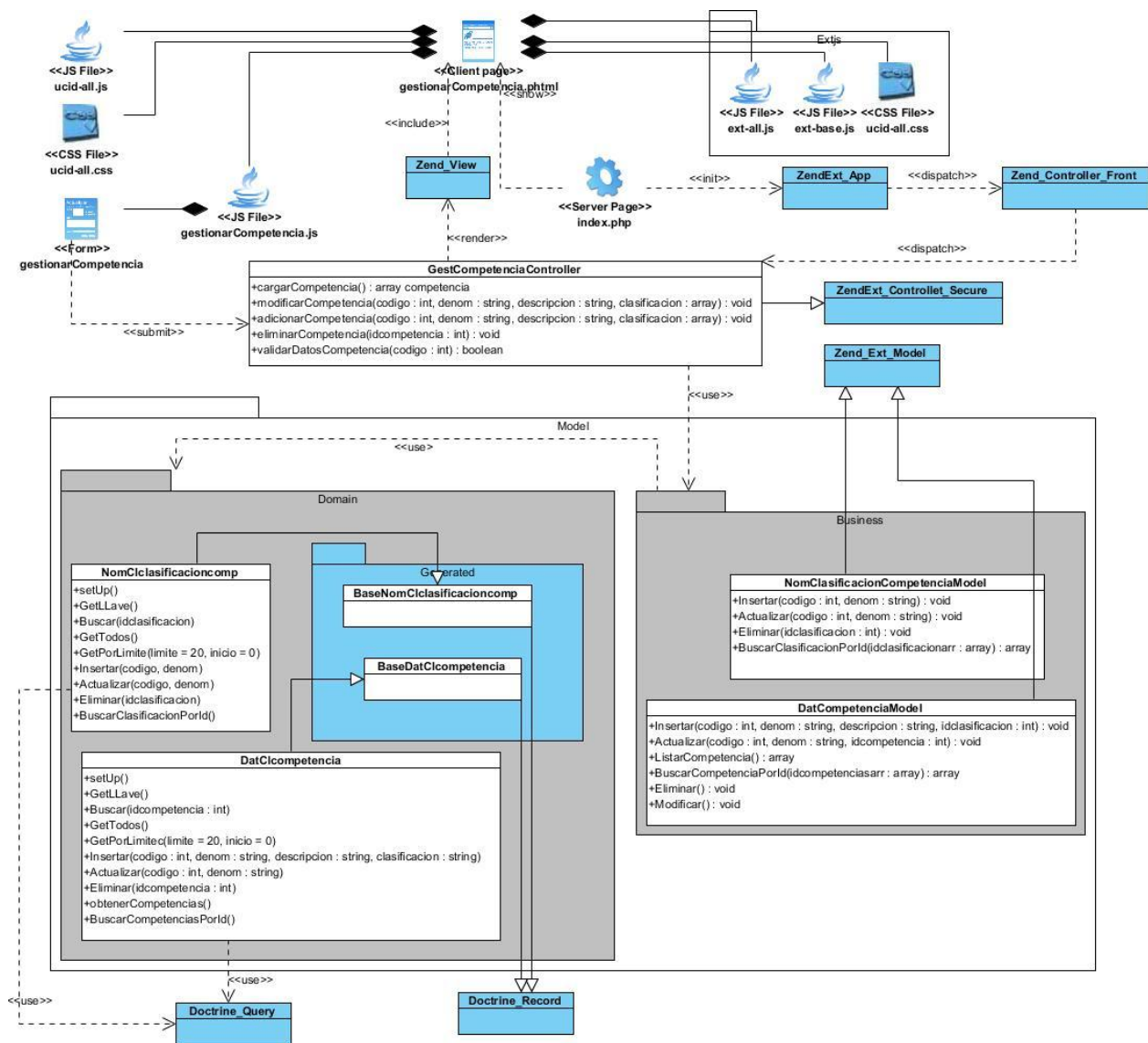


Figura 3. Diagrama de clases Gestionar competencias

En la figura 4 se muestra el diagrama de clases del escenario Expediente de competencias de la persona, en el cual se le da solución a los requerimientos relacionados con la gestión de las competencias de la persona. Las clases `gestionarExpediente.js` y `gestionarExpediente.phtml` conforman la capa arquitectónica de presentación, la clase `GestExpedienteCompetenciaController` maneja la comunicación entre la vista y el modelo, las clases `Model` son las encargadas de la lógica

del negocio y como encargadas al acceso de los datos se encuentran NomCclasiificacioncomp, DatCclcompetencia, DatClexpedientecompetencia y las clases Base de las cuales extienden las mismas.

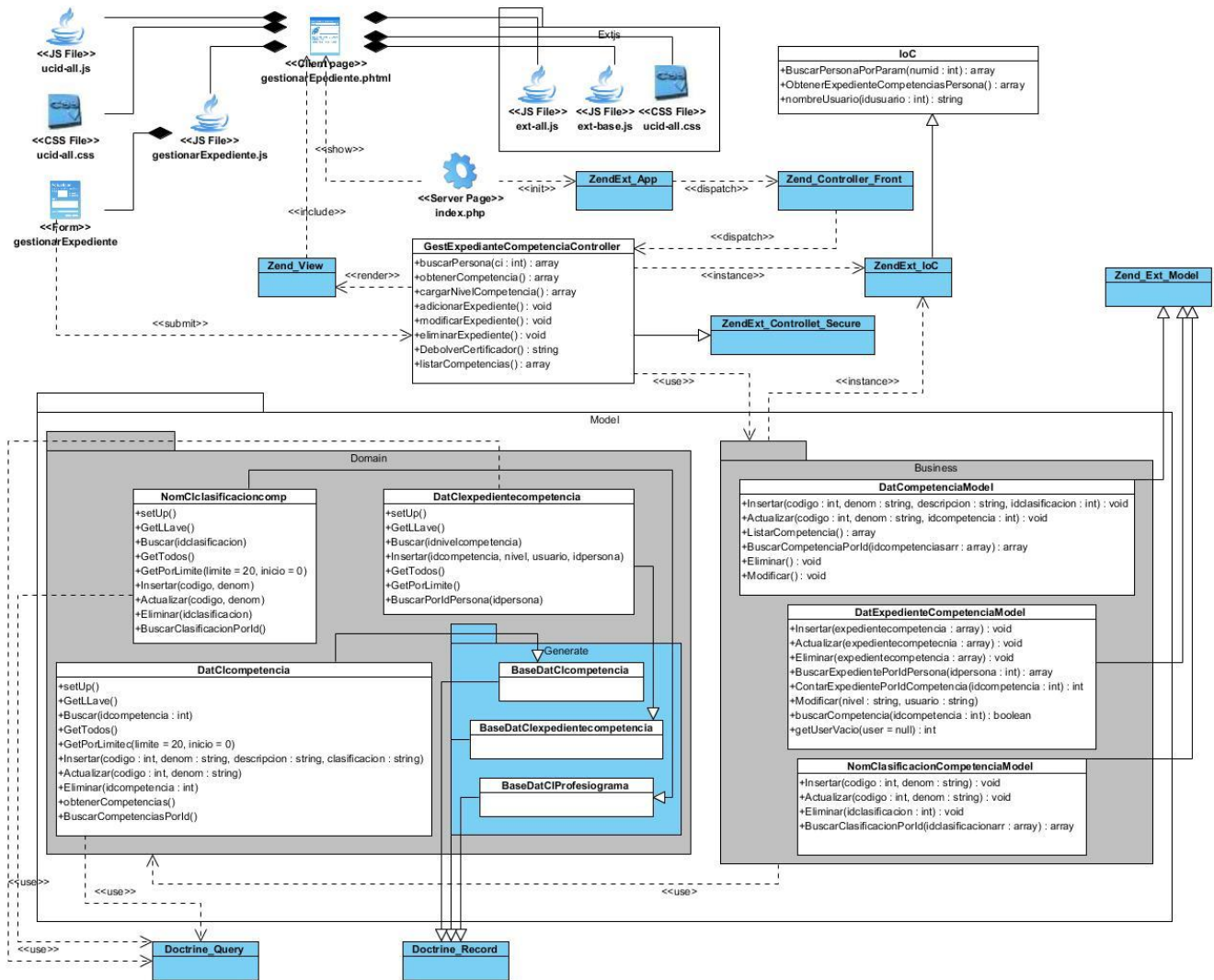


Figura 4. Diagrama de clases Expediente de competencias de la persona

En la figura 5 se muestra el diagrama de clases del diseño del escenario Gestionar profesigramas, en la cual se le da solución a los requisitos funcionales relacionados con la gestión del

profesiograma, se representan las principales clases, operaciones y relaciones que se necesitan para darle cumplimiento a estos requerimientos, donde la capa arquitectónica de presentación está formada por las clases `gestionarProfesiograma.js` y `gestionarProfesiograma.phtml`. La comunicación entre la vista y el modelo es realizada por la clase `GestProfesiogramaController`, las clases `Model` son las encargadas de la lógica del negocio, implementando funcionalidades que garantizan el cumplimiento de los requisitos identificados, encargadas del acceso a los datos se encuentran `NomClclasificacioncomp`, `DatClcompetencia`, `DatClProfesiograma` así como todas las clases `Base` de las cuales extienden.

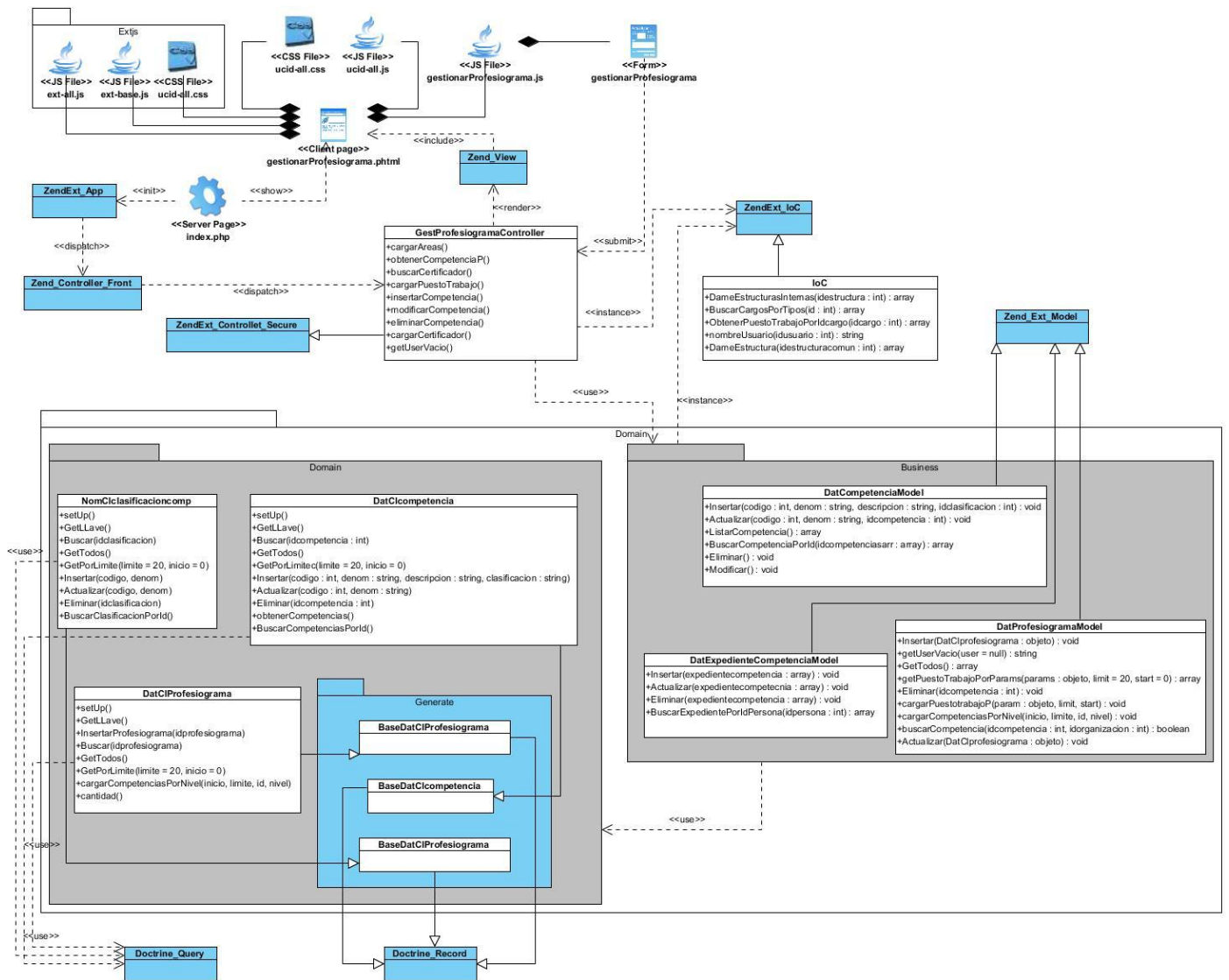


Figura 5. Diagrama de clases Profesigram

2.4. Modelo de datos

Un modelo de datos permite describir los elementos que intervienen en un problema dado y la forma en que se relacionan dichos elementos entre sí, permite además describir las estructuras de la base de datos, el tipo de dato que incluye la base y sus relaciones.

El modelo de datos propuesto cuenta con un total de 4 tablas, este cuenta con una relación externa que permitirá obtener datos de la persona que se encuentran en otro componente. En la Figura 6 se presenta el diagrama obtenido:

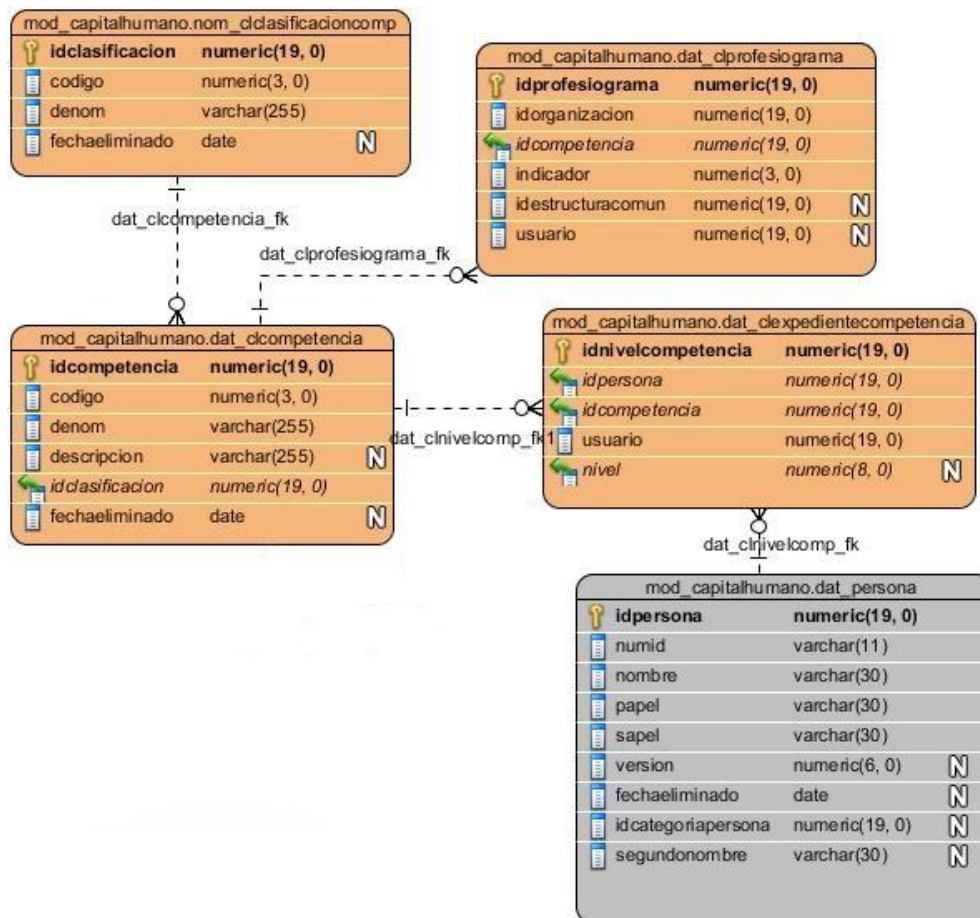


Figura 6. Modelo de Datos de Competencia laboral

A continuación se muestra la descripción de las tablas del modelo de datos:

nom_clclsificacioncomp: Es la encargada de registrar las clasificaciones de las competencias definidas por la entidad, la misma tiene relación con la tabla dat_clcompetencia.

dat_clcompetencia: En ella se registran los datos referentes a las competencias, la misma se relaciona con dat_clprofesiograma, dat_clexpedientecompetencia y nom_clclsificacioncomp.

dat_clprofesiograma: Esta tabla registra las competencias de las diferentes estructuras de la entidad y se relaciona con dat_clcompetencia.

dat_clexpedientecompetencia: En ella persisten las competencias que presentan cada una de las personas, esta se relaciona con dat_persona, dat_clcompetencia y nom_clclificacioncomp.

2.5. Implementación

2.5.1. Descripción del componente

El componente Competencia laboral es el encargado en el subsistema Capital humano de gestionar las competencias (creación, edición, procesamientos, etc) definidas por la entidad a través de un comité de competencias para luego certificarlas a los trabajadores. Existen dos procesos fundamentales que brindarán información a otros componentes, estos son:

- ✓ Profesiograma: contiene las competencias de las diferentes áreas, cargos y puestos de trabajo de la entidad, y es utilizado en los componentes de Selección y reclutamiento.
- ✓ Expediente de competencias de la persona: contiene las competencias de la persona, y es utilizado en el proceso de Selección.

2.5.2. Diagrama de componentes

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes, mostrando además las dependencias entre estos. Este tipo de diagrama contiene componentes, interfaces y sus relaciones, puede contener también paquetes que se utilizan para agrupar elementos del modelo. En la figura 7 se presenta el diagrama correspondiente al proceso Competencias laborales:

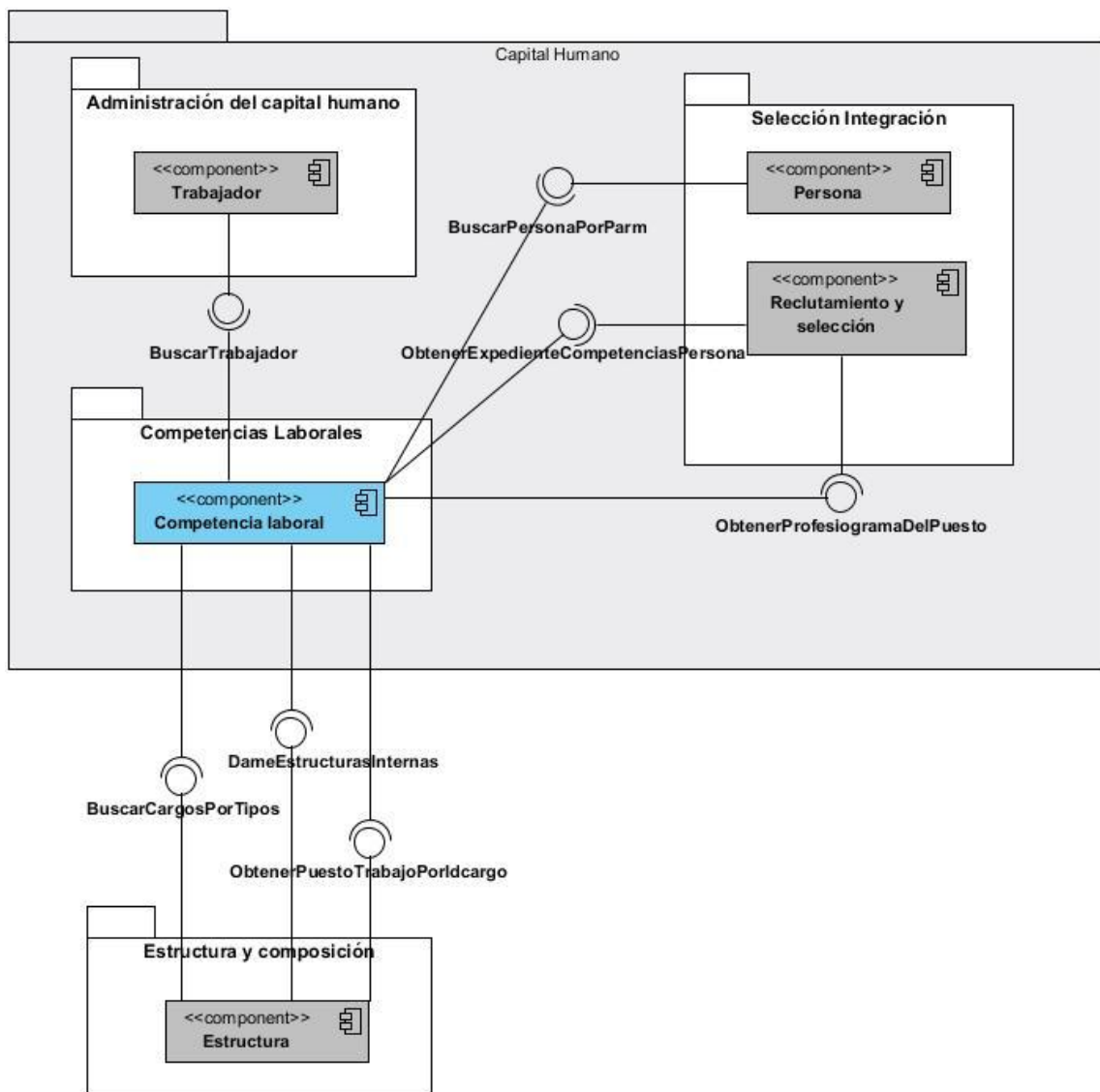


Figura 7. Diagrama de componentes

En el diagrama de componentes presentado anteriormente se puede observar la dependencia que existe entre el componente Competencias laborales y los demás componentes representados.

2.5.3. Servicios utilizados

La comunicación que se establece entre diferentes componentes y subsistemas de la aplicación se basa en el funcionamiento del componente llamado Inversión de Control (IoC). El IoC es donde se

publican los servicios que brinda cada uno de los componentes del Subsistema en cuanto a nombre de clases, funciones y tipo de resultado. La utilización de este se evidencia en el fichero .xml con igual nombre (ioc.xml) encargado de registrar las funcionalidades que ofrecen los métodos de las clases servicios de los componentes del sistema y especifica respuestas deseadas a sucesos o solicitudes de datos concretos, dejando que otro módulo o componente lleve a cabo las acciones de control que se requieran en el orden necesario y para el conjunto de sucesos que tengan que ocurrir. Existe un IoC interno para la integración entre componentes del subsistema y para la integración entre subsistemas un IoC externo.

Tabla 1. Descripción de los servicios brindados

Componente	Servicios que brinda	Componentes que lo utilizan	Descripción
Competencia Laboral	ObtenerExpedienteCompetenciasPersona	Reclutamiento y Selección	Este servicio le proporciona el expediente de competencias para que este a su vez lo utilice para seleccionar a la persona más competente e integrarla a determinado puesto de trabajo.
	ObtenerProfesiogramaDelPuesto	Reclutamiento y Selección	Este servicio le brinda el profesiograma del puesto para que este pueda conocer los tipos de competencias asignadas por cada área de trabajo.

Tabla 2. Descripción de los servicios que consume

Componente	Servicios que consume	Componentes que lo utilizan	Descripción
Persona	BuscarPersonaPorParam	Competencia laboral Prenómina	Este servicio brinda los datos de la persona para luego ser registrado en el

			Expediente de competencia de la persona.
Puesto Trabajo	de	BuscarCargosPorTipos	Competencia laboral Trabajadores
Puesto Trabajo	de	ObtenerPuestoTrabajoPorIdcargo	Competencia laboral Trabajadores
Estructura		DameEstructurasInternas	Competencia laboral Trabajadores
Estructura		DameEstructura	Competencia laboral
			Mediante este servicio se obtiene el listado de todos los cargos para luego utilizarlo en el Profesiograma.
			Mediante este servicio se obtiene el listado de todos los puestos de trabajo para luego utilizarlo en el Profesiograma.
			Este servicio devuelve un arreglo con los datos del área especificada por el ID de la estructura.
			Devuelve un arreglo con los datos de la estructura externa

2.6. Interfaces de la aplicación

La figura 8 muestra la interfaz Gestionar clasificación de competencia, la cual es la encargada de registrar las posibles clasificaciones de las competencias, la misma ofrece las opciones de Adicionar, Modificar, Listar o Eliminar las clasificaciones definidas.

Código	Denominación
1	Genericas
2	Especificas
3	Compuestas
4	Simples
5	Importantes
6	Esenciales

Figura 8. Interfaz Gestionar clasificación de competencia.

La figura 9 muestra la interfaz Gestionar competencia, la cual se encarga de gestionar las competencias definidas. Cada competencia tiene asociada una clasificación. Esta ofrece las opciones de Adicionar, Modificar, Listar o Eliminar las competencias.

Código	Denominación	Clasificación
96	Genio	Compuesta
23	lopiopioj	Esenciales
456	Conocimientos en Java	Específicas
101	12 grado	Esenciales
103	Conocimientos en Java	Simples
102	Ingeniero en Informatica	Importantes
101	Titulo 12 grado	Esenciales
10	Conocer sobre hardware	Simples

Figura 9. Interfaz Gestionar competencia.

En la figura 10 se muestra la interfaz Expediente de competencia de la persona, la cual es la encargada de registrar las competencias a las personas, la misma ofrece las opciones Buscar persona, Adicionar, Modificar, Listar o Eliminar las competencias de la persona.

Inicio Expedient... Registro ...

Expediente de competencia

Datos de la persona

Carnet de identidad: 85041327558 Nombre: Arnolis Primer apellido: Salgueiro Segundo apellido: Arzuaga Buscar

Competencia: PRUEBA Nivel de Competencia: Bajo Certificador: Admin| Adicionar Modificar Eliminar

Código	Denominación	Clasificación	Nivel de competencia
--------	--------------	---------------	----------------------

Página 1 de 1

Figura 10. Interfaz Expediente de competencia de la persona.

La figura 11 muestra la interfaz Profesiograma, en ella se asignan las competencias a cada una de las estructuras, que pueden ser área, cargo o puesto de trabajo. Brinda las opciones de Adicionar, Modificar, Listar o Eliminar las competencias por estructura.

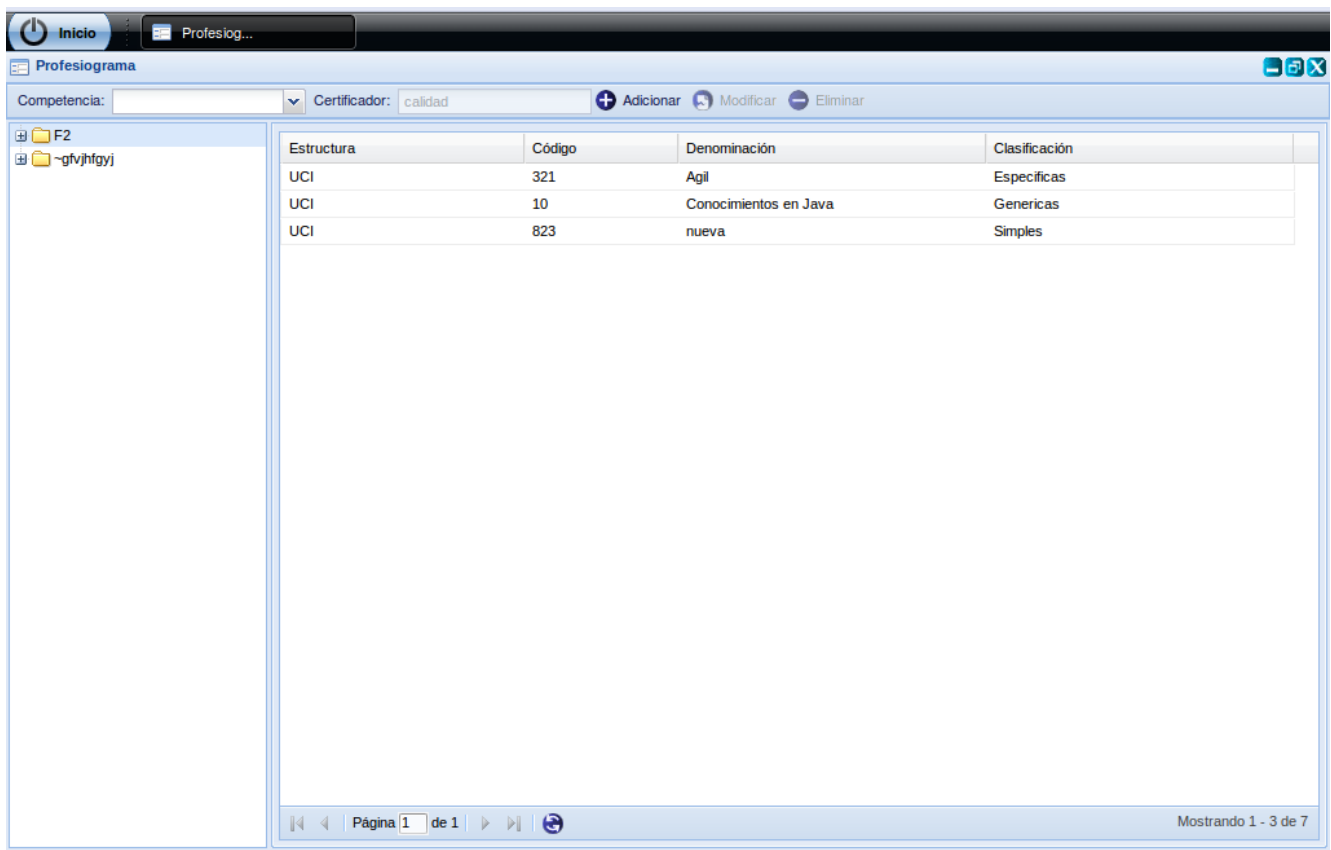


Figura 11. Interfaz Profesiograma.

ANEXO 1: Se muestran el resto de las interfaces relacionadas con Gestionar clasificación de competencia y Gestionar competencia.

2.7. Diagrama de despliegue

Los diagramas de despliegue representan a los nodos y sus relaciones. Los nodos son conectados por asociaciones de comunicación tales como enlaces de red, conexiones TCP/IP, microondas, etc. (Àlvaro, et al.)

Describen la topología del sistema, incluyendo la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos. El despliegue del componente Competencia laboral se rige por la especificación dada para el subsistema Capital humano, debido a que el mismo no permite ser implementado de forma independiente ya que depende de otros componentes.

Se definirá una arquitectura cliente-servidor detallada de la siguiente manera. Ver figura 12 y 13.

Cliente: Computador, la aplicación se ejecuta a través de un navegador internet instalado, en este caso se debe usar el Mozilla Firefox sobre cualquier sistema operativo (se sugiere GNU/Linux, en específico la distribución de Ubuntu para estaciones donde se conectarán dispositivos externos: Impresoras).

Servidor web: Radica la lógica de negocio de la aplicación. Servidor web Apache 2.0. Utilizando el lenguaje PHP 5.

Servidor de Base de datos: Servidor de Datos PostgreSQL 8.3

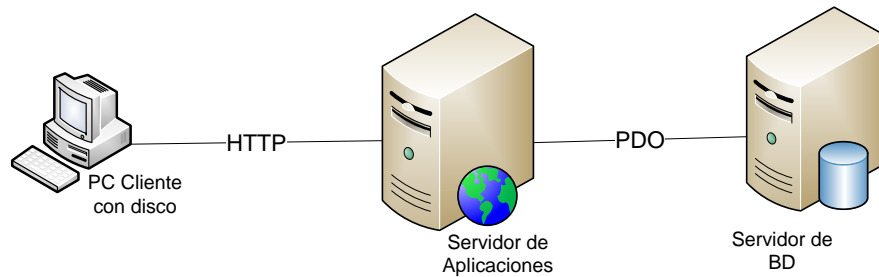


Figura 12. Diagrama de Red para PC Cliente con Disco Duro.

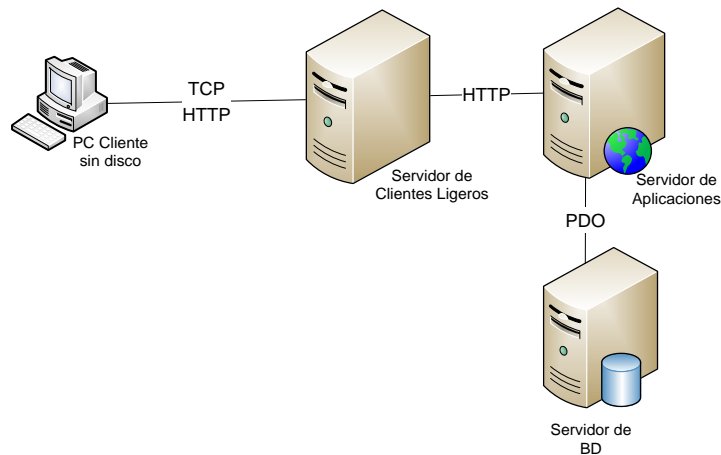


Figura 13. Diagrama de Red para PC Cliente sin Disco Duro.

2.8. Conclusiones parciales

En éste capítulo se abarcaron los aspectos más significativos del diseño e implementación de la solución propuesta. Partiendo del análisis de los artefactos entregados por los analistas del sistema se modificaron los diagramas de clases de diseño, que permitieron desarrollar de manera más eficiente la implementación del sistema, el diagrama de componentes que permitió una visión más clara para el desarrollo del mismo, y de las integraciones entre los componentes que están relacionados con el sistema, así como el diagrama de despliegue donde se mostraron los posibles escenarios en los cuales puede ser desplegado la aplicación y el modelo de datos encargado de representar las tablas, donde se pudo identificar sus propiedades y las relaciones entre ellas. De forma general se lograron implementar los requerimientos definidos por los analistas para los procesos de gestión de competencia laboral, pertenecientes al subsistema Capital humano del Sistema Integral de Gestión Cedrux. Se realizó la implementación del componente el cual se encuentra listo para ser validado.

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN

3.1. Introducción

El desarrollo del software implica una serie de actividades de producción en las que las posibilidades de que aparezcan fallos humanos son muy grandes. Los errores pueden presentarse debido a especificaciones erróneas e imperfectas de los requisitos, uso indebido de las estructuras de datos, errores al integrar módulos, entre otras causas. Dado a la imposibilidad humana de trabajar y comunicarse de forma perfecta, el desarrollo del software ha de ir acompañado de una actividad que garantice la calidad. Para dar solución a todos estos problemas surge un nuevo proceso dentro del ciclo de desarrollo del software: el proceso de pruebas. Las pruebas de software, son los procesos que permiten verificar y revelar la calidad de un producto de software, son utilizadas para identificar posibles fallos de implementación, calidad o usabilidad de un programa.

En este capítulo se abordarán temas como las pruebas realizadas al software, en específico las pruebas de caja blanca y caja negra, además se hace una valoración de las mismas según los resultados obtenidos.

3.2. Métricas para la validación del diseño propuesto

Las métricas de software son una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado. Permiten averiguar cuán bien están definidas las clases y el sistema, lo cual tiene un impacto directo en el mantenimiento del mismo, tanto por la comprensión de lo desarrollado como por la dificultad de modificarlo con éxito. Posibilitan al ingeniero evaluar el software al inicio del proceso, haciendo cambios que reducirán la complejidad y mejorarán la viabilidad, a largo plazo, del producto final. También permiten descubrir y corregir problemas potenciales antes de que se conviertan en defectos catastróficos. Estas métricas tienen como propósito entender y mejorar la calidad del producto, evaluar la efectividad del proceso y mejorar la calidad del trabajo llevado a cabo al nivel del proyecto (Pressman, 2005).

Una vez realizado el diseño del componente Ejecución, se dio paso a su validación, para esto se utilizaron las métricas orientadas a objetos, específicamente las orientadas a clases, ya que las

medidas y métricas para una clase individual, la jerarquía de clases y las colaboraciones de estas permiten medir la calidad del diseño propuesto (Pressman, 2005).

Las métricas empleadas para evaluar el diseño realizado fueron la métrica **Tamaño operacional de clase (TOC)** y la métrica **Relaciones entre clases (RC)**, ya que se ajustan para evaluar al diseño realizado, siendo las mismas de fácil utilización.

Estas métricas están diseñadas para evaluar los siguientes atributos de calidad:

Responsabilidad. Consiste en la responsabilidad asignada a una clase en un marco de modelado de un dominio o concepto, de la problemática propuesta.

Complejidad de implementación. Consiste en el grado de dificultad que tiene implementar un diseño de clases determinado.

Reutilización. Consiste en el grado de reutilización presente en una clase o estructura de clase, dentro de un diseño de software.

Acoplamiento. Consiste en el grado de dependencia o interconexión de una clase o estructura de clase, con otras, está muy ligada a la característica de Reutilización.

Complejidad de mantenimiento. Consiste en el grado de esfuerzo necesario a realizar para desarrollar un arreglo, una mejora o una rectificación de algún error de un diseño de software. Puede influir indirecta, pero fuertemente en la planificación del proyecto.

Cantidad de pruebas. Consiste en el número o el grado de esfuerzo para realizar las pruebas de calidad del producto (componente, módulo, clase, conjunto de clases) diseñado.

3.2.1 Métrica Tamaño operacional de clase

Tamaño operacional de clase (TOC): Está dado por el número de métodos asignados a una clase. A continuación se muestra una serie de tablas encaminadas a un mejor entendimiento de la utilización de esta métrica.

Tabla 3. Tamaño operacional de clase (TOC).

Atributo que afecta	Modo en que lo afecta
---------------------	-----------------------

Responsabilidad	Un aumento del TOC implica un aumento de la responsabilidad asignada a la clase.
Complejidad de implementación	Un aumento del TOC implica un aumento de la complejidad de implementación de la clase.
Reutilización	Un aumento del TOC implica una disminución del grado de reutilización de la clase.

Tabla 4. Rango de valores para la evaluación técnica de los atributos de calidad relacionados con la métrica TOC.

Atributo	Categoría	Criterio
Responsabilidad	Baja	\leq Promedio
	Media	Entre Promedio y $2 \times$ Promedio
	Alta	$> 2 \times$ Promedio
Complejidad de implementación	Baja	\leq Promedio
	Media	Entre Promedio y $2 \times$ Promedio
	Alta	$> 2 \times$ Promedio
Reutilización	Baja	$> 2 \times$ Promedio
	Media	Entre Promedio y $2 \times$ Promedio
	Alta	\leq Promedio

La tabla que se muestra a continuación ofrece las clases del sistema a las que se le aplicó la métrica y los resultados obtenidos para cada atributo evaluado.

Tabla 5. Resultados de la aplicación de la métrica TOC para cada clase del sistema.

No	Clase	Cantidad de	Responsabilidad	Complejidad	Reutilización
----	-------	-------------	-----------------	-------------	---------------

		Procedimientos			
1	GestionarclasificacionController	8	Baja	Baja	Alta
2	GestionarcompetenciaController	8	Baja	Baja	Alta
3	NivelcompetenciaController	9	Media	Media	Media
4	ProfesiogramaController	10	Media	Media	Media
5	DatClcompetenciaModel	8	Baja	Baja	Alta
6	DatClexpedientecompetenciaModel	7	Baja	Baja	Alta
7	DatClprofesiogramaModel	11	Media	Media	Media
8	NomClclasificacioncompModel	5	Baja	Baja	Alta
9	DatClcompetencia	9	Media	Media	Media
10	DatClexpedientecompetencia	7	Baja	Baja	Alta
11	DatClprofesiograma	8	Baja	Baja	Alta
12	NomClclasificacioncomp	9	Media	Media	Media

Las gráficas que se muestran a continuación muestran los resultados obtenidos para cada uno de los atributos medidos.

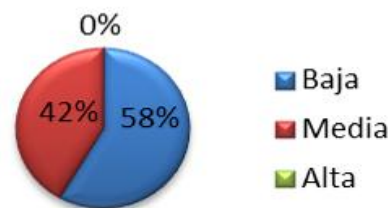


Figura 14. Representación en % de los resultados de la evaluación de la métrica TOC en el atributo Responsabilidad.

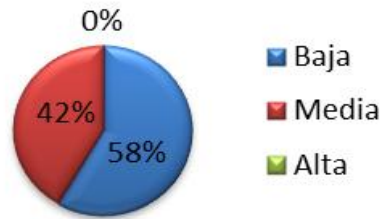


Figura 15. Representación en % de los resultados de la evaluación de la métrica TOC en el atributo Complejidad de implementación.

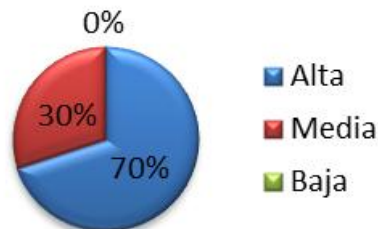


Figura 16. Representación en % de los resultados de la evaluación de la métrica TOC en el atributo Reutilización.

Al realizar un análisis de los resultados obtenidos al aplicarle esta métrica a las clases controladoras y a las del modelo se demuestra que la mayoría de estas clases se encuentran dentro de las categorías media y baja, quedando demostrado de esa forma la elevada reutilización, baja complejidad y responsabilidad que presenta el diseño propuesto, permitiendo una sencilla implementación y una amplia realización de pruebas con mayor facilidad, por lo que los resultados obtenidos según esta métrica son positivos.

3.2.2. Métrica Relaciones entre clases

Relaciones entre clases (RC): Está dado por el número de relaciones de uso de una clase con otra y evalúa los siguientes atributos de calidad.

Tabla 6. Relaciones entre clases (RC).

Atributo que afecta	Modo en que lo afecta
Acoplamiento	Un aumento del RC implica un aumento del acoplamiento de la

	clase.
Complejidad de mantenimiento	Un aumento del RC implica un aumento de la complejidad del mantenimiento de la clase.
Reutilización	Un aumento del RC implica una disminución en el grado de reutilización de la clase.
Cantidad de pruebas	Un aumento del RC implica un aumento de la Cantidad de pruebas de unidad necesarias para probar una clase.

Tabla 7. Rango de valores de para la evaluación técnica de los atributos de calidad (Acoplamiento, Complejidad de Mantenimiento, Reutilización y Cantidad de Pruebas) relacionados con la métrica RC.

Atributo	Categoría	Criterio
Acoplamiento	Ninguno	0
	Bajo	1
	Medio	2
	Alto	>2
Complejidad de mantenimiento	Baja	\leq Promedio
	Media	Entre Promedio y $2 \times$ Promedio
	Alta	$>2 \times$ Promedio
Reutilización	Baja	$>2 \times$ Promedio
	Media	Entre Promedio y $2 \times$ Promedio
	Alta	\leq Promedio

Cantidad de pruebas	Baja	\leq Promedio
	Media	Entre Promedio y $2 \times$ Promedio
	Alta	$> 2 \times$ Promedio

La tabla 8 ofrece las clases del sistema a las que se le aplicó la métrica y los resultados obtenidos para cada atributo evaluado.

Tabla 8. Resultados de la aplicación de la métrica RC para cada clase del sistema.

No	Clase	Cantidad de Relaciones de Uso	Acoplamiento	Complejidad Mant.	Reutilización	Cantidad de Pruebas
1	GestionarclasificacionController	1	Bajo	Baja	Alta	Baja
2	GestionarcompetenciaController	2	Medio	Media	Media	Media
3	NivelcompetenciaController	4	Alto	Alta	Baja	Alta
4	ProfesiogramaController	4	Alto	Alta	Baja	Alta
5	DatClcompetenciaModel	1	Bajo	Baja	Alta	Baja
6	DatClexpedientecompetenciaModel	1	Bajo	Baja	Alta	Baja
7	DatClprofesiogramaModel	2	Medio	Media	Media	Media
8	NomClclasificacioncompModel	1	Bajo	Baja	Alta	Baja
9	DatClcompetencia	0	Ninguno	Baja	Alta	Baja
10	DatClexpedientecompetencia	0	Ninguno	Baja	Alta	Baja
11	DatClprofesiograma	0	Ninguno	Baja	Alta	Baja
12	NomClclasificacioncomp	0	Ninguno	Baja	Alta	Baja

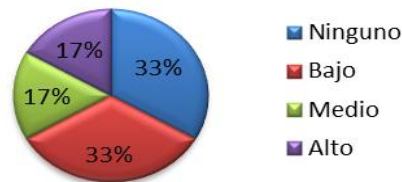


Figura 17. Representación en % de los resultados de la evaluación de la métrica RC en el atributo Acoplamiento.

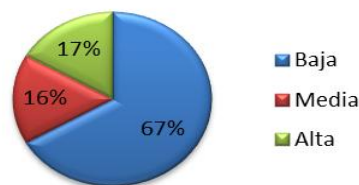


Figura 18. Representación en % de los resultados de la evaluación de la métrica RC en el atributo Complejidad de mantenimiento.

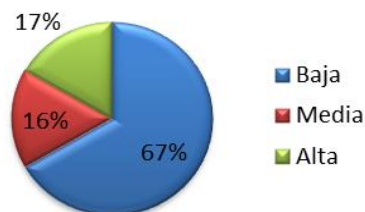


Figura 19. Representación en % de los resultados de la evaluación de la métrica RC en el atributo Reutilización.

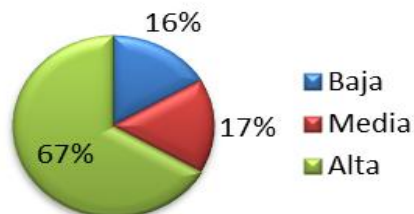


Figura 20. Representación en % de los resultados de la evaluación de la métrica RC en el atributo Cantidad de pruebas.

Luego de aplicarse la métrica de diseño RC y obtenidos los resultados de la evaluación del instrumento de medición de la métrica, se puede concluir que el diseño propuesto tiene una calidad

aceptable teniendo en cuenta que el 67% de las clases empleadas poseen menos de 3 dependencias de otras clases, lo que lleva a evaluaciones positivas de los atributos de calidad involucrados (acoplamiento, complejidad de mantenimiento, cantidad de pruebas y reutilización). Favoreciendo de esta manera la reutilización de las clases así como la modificación e implantación del diseño.

Al aplicar las métricas a las clases del diseño definidas se cumple que las métricas de Tamaño de la clase y Relaciones entre clases cumplen las restricciones para la aceptación de las mismas.

3.3. Pruebas de software

Las pruebas de software son el conjunto de técnicas que permiten determinar la calidad de un producto. Estas se integran dentro de las diferentes fases del ciclo de vida del software dentro de la ingeniería. Así se ejecuta un programa y mediante técnicas experimentales se trata de descubrir qué errores presenta. La calidad de un sistema informático es algo subjetivo que depende del contexto y del objeto que se pretenda conseguir. Para determinar dicho nivel de calidad se deben efectuar medidas o pruebas que permitan comprobar el grado de cumplimiento con respecto a las especificaciones iniciales del sistema. (Juristo, et al., 2005)

3.4. Métodos de prueba

Para probar y corregir los posibles errores en el sistema implementado, fueron utilizadas las pruebas de Caja negra y Caja blanca también llamadas pruebas funcionales o estructurales respectivamente.

3.4.1. Pruebas de Caja blanca o Estructurales

La prueba de Caja blanca del software se basa en un examen cercano al detalle procedimental. Se prueban las rutas lógicas del software y la colaboración entre componentes, al proporcionar casos de prueba que ejerciten conjuntos específicos de condiciones, bucles o ambos. Este tipo de pruebas en ocasiones llamada prueba de caja de cristal es un método de diseño que usa la estructura de control descrita como parte del diseño al nivel de componentes para derivar los casos de prueba. (Pressman, 2005)

A continuación se citan algunas de las técnicas de prueba de Caja blanca:

- ✓ Condición.

- ✓ Flujo de Datos.
- ✓ Bucles.
- ✓ Camino Básico.

Dentro de la prueba de Caja blanca, la técnica que se utilizó fue la de prueba del camino básico. Esta prueba permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución, además también garantiza que durante la prueba en los casos de prueba obtenidos a través del camino básico se ejecute cada sentencia del programa por lo menos una vez. (Pressman, 2005)

Para aplicar la técnica del camino básico primeramente hay que calcular la complejidad ciclomática y para esto hay que elaborar el gráfico de flujo.

La complejidad ciclomática es una métrica del software que proporciona una medición cuantitativa de la complejidad lógica de un programa. Cuando se usa en el contexto del método de prueba del camino básico, el valor calculado como complejidad ciclomática define el número de caminos independientes del conjunto básico de un programa y nos da un límite superior para el número de pruebas que se deben realizar para asegurar que se ejecuta cada sentencia al menos una vez. (Pressman, 2005)

Para realizar la técnica de prueba de caja blanca, específicamente la prueba del camino básico es necesario calcular antes la complejidad ciclomática del algoritmo o fragmento de código a analizar. La complejidad ciclomática se puede calcular de tres formas:

1. El número de regiones del grafo de flujo coincide con la complejidad ciclomática.
2. La complejidad ciclomática, $V(G)$, de un grafo de flujo G se define como:

$$V(G) = A - N + 2$$

Donde A es el número de aristas del grafo de flujo y N es el número de nodos del mismo.

3. La complejidad ciclomática, $V(G)$, de un grafo de flujo G también se define como:

$$V(G) = P + 1$$

Donde P es el número de nodos predicado² contenidos en el grafo de flujo G . (Pressman, 2005)

² Cuando en una condición aparecen uno o más operadores lógicos (AND, OR, XOR,...) se crea un nodo distinto por cada una de las condiciones simples. Cada nodo generado de esta forma se denomina nodo predicado.

A continuación se enumera las sentencias de código del procedimiento realizado sobre el método `cargarpujestotrabajo ()`, a partir del cual se creará el grafo de flujo para calcular la complejidad ciclomática. Este método es el encargado de devolver un arreglo con las competencias de cada estructura para luego insertarle la denominación de la estructura común a través del servicio `DameEstrucutra ()`, que me servirá para mostrar dicha denominación en la interfaz profesiograma.

```
function cargarpujestotrabajoAction() {  
1     $limite=$this->_request->getPost('limit');  
1     $inicio=$this->_request->getPost('start');  
1     $id= $this->_request->getPost('id');  
1     $nivel= $this->_request->getPost('nivel');  
1     $obj=new DatClprofesiogramaModel();  
1     $result=$obj->cargarCompetenciasPorNivel($inicio,$limite,$id,$nivel);  
2     if(count ($result[datos])>0) {  
3         $idestructuracomun = $result['datos'][0]['idestructuracomun'];  
3         $datos = $this->integrator->metadatos->DameEstructura($idestructuracomun);  
3         $nombre = $datos[0]->denominacion;  
4         for($i = 0; $i < count ($result[datos]); $i ++) {  
5             $result[datos][$i]['nombre']=$nombre;  
6         }  
7     }  
8     echo json_encode(array('cantidad'=>$result[cantidad] , 'datos' => $result[datos]));  
}
```

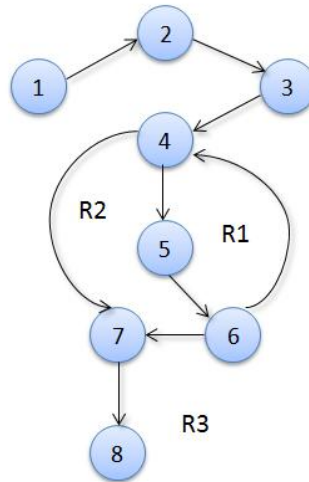


Figura 21. Grafo de flujo asociado al método cargapuestotrabajo ().

Aplicando las tres formas para calcular la complejidad ciclomática se tiene:

$$V(G) = \text{Regiones del grafo} = 3$$

$$V(G) = A - N + 2 = 9 - 8 + 2 = 3$$

A es el número de aristas del grafo y N número de nodos.

$$V(G) = P + 1 = 2 + 1 = 3$$

P es el número de nodos predicados.

Todas las fórmulas arrojaron el mismo resultado. Lo que demuestra que 3 es el número de caminos independientes y también la cantidad superior de casos de pruebas que se deben realizar para asegurar que se ejecuta cada sentencia al menos una vez.

Tabla 9. Caminos independientes del flujo.

	Caminos
1	1-2-8
2	1-2-3-4-7-8
3	1-2-3-4-5-6-4-7-8

Para cada camino obtenido se realiza un caso de prueba.

Tabla 10. Caso de prueba para el camino básico 1

Camino básico #1: 1-2-8	
Descripción	A partir del identificador y el nivel de la estructura, se obtienen todas las competencias asociadas a esta. Después si existen competencias se obtiene el nombre de la estructura común para posteriormente mostrarlo en la interfaz profesiograma.
Condición de ejecución	Se debe tener el identificador de la estructura ('id') Se debe tener el nivel de la estructura ('nivel')
Entrada	\$id= \$this->_request->getPost('900001') \$nivel= \$this->_request->getPost('1')
Resultados esperados	La estructura no tiene competencias asociadas, por lo que no devuelve nada.
Resultados	cantidad 0, datos []. No existen competencias, por lo que la sentencia no se cumple mediante este camino básico.
Salida	0

Tabla 11. Caso de prueba para el camino básico 2

Camino básico #2: 1-2-3-4-7-8	
Descripción	A partir del identificador y el nivel de la estructura, se obtienen todas las competencias asociadas a esta. Después si existen competencias se obtiene el nombre de la estructura común para posteriormente mostrarlo en la interfaz profesiograma.
Condición de ejecución	Se debe tener el identificador de la estructura ('id') Se debe tener el nivel de la estructura ('nivel')
Entrada	\$id= \$this->_request->getPost('900000004') \$nivel= \$this->_request->getPost('0')

Resultados esperados	La estructura tiene competencias asociadas, por lo que devuelve todas las competencias de la misma.
Resultados	cantidad 2, datos array 1 [idcompetencia, nivel, idorganizacion, indicador, certificador]; array 2 [idcompetencia, nivel, idorganizacion, indicador, certificador]. Con este resultado se demuestra que existen competencias asociadas a esta estructura por lo que se ejecuta la sentencia y se obtiene el nombre de la estructura común.
Salida	2

Es necesario aclarar que el camino básico número 2 nunca se llegará a ejecutar, ya que para realizar el ciclo primeramente se debe cumplir la condición de que existan competencias en la estructura, si existen el ciclo siempre se cumplirá, por lo que este camino no cumple objetivo.

Tabla 12. Caso de prueba para el camino básico 3

Camino básico #3: 1-2-3-4-5-6-4-7-8	
Descripción	A partir del identificador y el nivel de la estructura, se obtienen todas las competencias asociadas a esta. Después si existen competencias se obtiene el nombre de la estructura común para posteriormente mostrarlo en la interfaz profesiograma.
Condición de ejecución	Se debe tener el identificador de la estructura ('id') Se debe tener el nivel de la estructura ('nivel')
Entrada	\$id= \$this->_request->getPost('900000004') \$nivel= \$this->_request->getPost('0')
Resultados esperados	La estructura tiene competencias asociadas, por lo que devuelve todas las competencias de la misma.
Resultados	cantidad 2, datos array 1 [idcompetencia, nivel, idorganizacion, indicador, certificador, denominacion]; array 2 [idcompetencia, nivel, idorganizacion, indicador, certificador, denominacion]. Este resultado

	demuestra que existen competencias asociadas a esta estructura por lo que se ejecuta la sentencia y se obtiene el nombre de la estructura común. Posteriormente realiza el ciclo para insertar dentro de los datos de las competencias el nombre obtenido en la interfaz.
Salida	2

Con la aplicación de los casos de pruebas expuestos anteriormente se comprobó que cada sentencia del algoritmo es ejecutado al menos una vez por lo cual se cumple el objetivo de la prueba.

3.4.2 Pruebas de Caja negra

Las pruebas de Caja negra son las que se aplican a la interfaz del software. Una prueba de este tipo examina algún aspecto funcional de un sistema que tiene poca relación con la estructura lógica interna del software. Las pruebas de Caja negra, también denominadas pruebas de comportamiento, se concentran en los requisitos funcionales del software, es decir permiten al ingeniero de software derivar conjuntos de condiciones de entrada que ejercitarán por completo todos los requisitos funcionales de un programa. Las pruebas de Caja negra tratan de encontrar errores de las siguientes categorías: funciones incorrectas o faltantes, errores de interfaz, errores de estructuras de datos o en acceso a datos externos, errores de comportamiento o desempeño y errores de inicialización y término. (Pressman, 2005)

Muchos autores consideran que estas pruebas permiten encontrar: (Pressman, 2005)

- ✓ Funciones incorrectas o ausentes.
- ✓ Errores de interfaz.
- ✓ Errores en estructuras de datos o en accesos a las bases de datos externas.
- ✓ Errores de rendimiento.
- ✓ Errores de inicialización y terminación.

Entre los métodos utilizados para realizar las pruebas de caja negra a la aplicación se encuentran:

- ✓ **Partición equivalente**

La partición equivalente es un método de prueba de caja negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Un caso de prueba ideal descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución

de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar. (Pressman, 2005)

✓ **Análisis de valores límite**

Los errores tienden a darse más en los límites del campo de entrada que en el centro. Por ello, se ha desarrollado el análisis de valores límites (AVL) como técnica de prueba. El análisis de valores límite nos lleva a una elección de casos de prueba que ejerciten los valores límite. El análisis de valores límite es una técnica de diseño de casos de prueba que complementa a la partición equivalente. En lugar de seleccionar cualquier elemento de una clase de equivalencia, el AVL lleva a la elección de casos de prueba en los extremos de la clase. En lugar de centrarse solamente en las condiciones de entrada, el AVL obtiene casos de prueba también para el campo de salida. (Pressman, 2005)

Para cada requisito fueron diseñados casos de pruebas, a continuación se presenta el caso de Prueba para el requisito **Adicionar competencia**.

Descripción general

Se debe seleccionar en el menú principal el subsistema Capital humano/Competencia laboral /Gestionar competencia.

Condiciones de ejecución:

- ✓ Se debe identificar y autenticar ante el sistema y además debe tener los permisos para ejecutar esta acción.
- ✓ Debe haber clasificaciones registradas en el sistema.
- ✓ Se debe presionar el botón Adicionar de la barra de opciones.

Tabla 13. Posibles escenarios al adicionar una competencia

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
1: Adicionar competencia.	EC 1.1: Adicionar competencia introduciendo datos válidos.	Se adiciona la competencia correctamente.	<ul style="list-style-type: none"> ✓ Se introducen los datos de la competencia correctamente. ✓ Se presiona el botón Aceptar.

		<ul style="list-style-type: none"> ✓ Se muestra un mensaje de información. ✓ Se presiona el botón Aceptar.
EC 1.2: Adicionar competencia introduciendo datos válidos presionando el botón Aplicar .	Se adiciona la competencia correctamente luego de presionar el botón Aplicar.	<ul style="list-style-type: none"> ✓ Se introducen los datos de la competencia correctamente. ✓ Se presiona el botón Aplicar. ✓ Se muestra un mensaje de información. ✓ Se presiona el botón Aceptar.
EC 1.3: Adicionar competencia introduciendo datos inválidos.	No se puede adicionar la competencia pues se introducen datos inválidos.	<ul style="list-style-type: none"> ✓ Se introducen los datos inválidos de la competencia. ✓ Se presiona el botón Aceptar. ✓ Se muestra un mensaje informando del error.
EC 1.4: Adicionar competencia introduciendo un código existente.	No se puede adicionar la competencia pues se introduce un código que ya existe.	<ul style="list-style-type: none"> ✓ Se introducen los datos de la competencia con un código existente. ✓ Se presiona el botón Aceptar. ✓ Se muestra un mensaje informando del error.
EC 1.5: Adicionar competencia dejando	No se puede adicionar la competencia pues se deja(n)	<ul style="list-style-type: none"> ✓ Se introducen los datos dejando algún campo en blanco.

campos vacíos.	algún(os) campo(s) vacío(s).	✓	Se presiona el botón Aceptar .
		✓	Se muestra un mensaje informando del error.
EC 1.6: Cancelar.	No se realiza ninguna acción.	✓	Se introducen o no los datos de la competencia.
		✓	Se presiona el botón Cancelar .

Los casos de prueba para el resto de los requisitos se encuentran en los documentos: CIG-ERP-N-CH-i51201, CSG-ERP-N-CH-i51202, CSG-ERP-N-CH-i51203, CSG-ERP-N-CH-i51204, CSG-ERP-N-CH-i51205, CIG-ERP-N-CH-i51206, CIG-ERP-N-CH-i51207, CIG-ERP-N-CH-i51208, CIG-ERP-N-CH-i51209, CIG-ERP-N-CH-i51210, CIG-ERP-N-CH-i51211, CIG-ERP-N-CH-i51212, CIG-ERP-N-CH-i51213, CIG-ERP-N-CH-i51214, CIG-ERP-N-CH-i51215, CIG-ERP-N-CH-i51216, CIG-ERP-N-CH-i51217, CIG-ERP-N-CH-i51218, para un total de 19 casos de prueba.

Las pruebas de caja negra a la solución fueron desarrolladas por el Grupo de Aseguramiento de la Calidad del Centro de Informatización para la Gestión de Entidades (CEIGE). Estas se realizaron en tres iteraciones, donde finalmente se comprobó en la tercera iteración que el componente estaba libre de no conformidades culminando así la fase de pruebas internas. En el anexo 2 se puede observar el acta de liberación emitida. La tabla que se muestra a continuación ofrece detalles de los resultados obtenidos.

Tabla 14. Resultado de las iteraciones de las pruebas de caja negra realizadas por el Grupo de Aseguramiento de la Calidad.

Agrupación de requisitos	No Conformidades		
	Iteración 1	Iteración 2	Iteración 3
Gestionar Clasificación de competencia	3	0	0
Gestionar competencia	4	0	0

Gestionar expediente de competencia de la persona	3	1	0
Gestionar profesiograma	1	1	0
Total	11	2	0

En las mismas se detectaron errores mayormente de faltas de ortografías en los mensajes de información o de errores, que no coincidían con los descritos en los casos de prueba, así como errores en algunas validaciones.

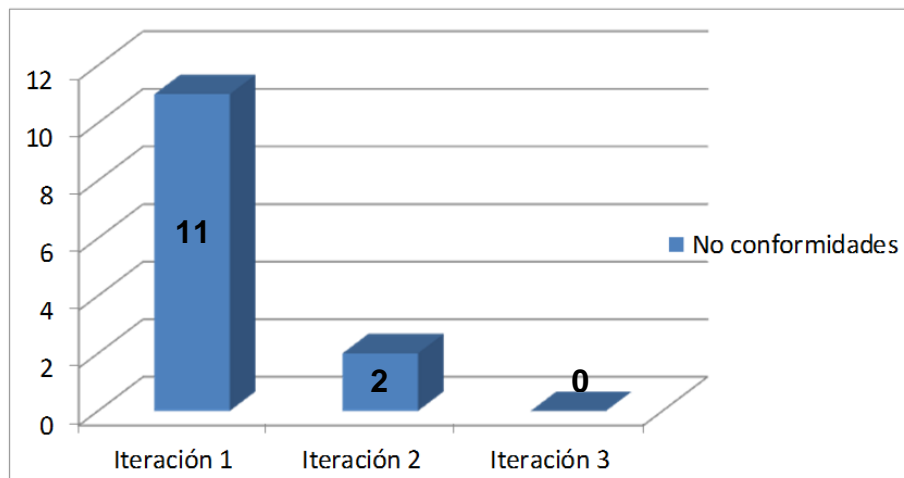


Figura 22. Resultado de las Iteraciones

Luego de ser corregidos los errores encontrados en las pruebas, se pudo comprobar que el flujo de trabajo de las interfaces estaba correcto ya que cumplían con las condiciones necesarias que se habían planteado.

3.5. Conclusiones parciales

Las pruebas realizadas al software arrojaron resultados de gran importancia, ya que ayudaron a identificar algunos errores que habían pasado desapercibidos durante la etapa de codificación, además de validar el código fuente y servir como respaldo funcional al mismo. Se hicieron pruebas de caja blanca que permitieron conocer la complejidad ciclomática de cada funcionalidad, se probaron además cada una de las interfaces mostrando cómo respondían a los requisitos funcionales

a través de las pruebas de caja negra. Estas pruebas se llevaron a cabo en tres iteraciones, corrigiéndose los errores encontrados en cada una de ellas, asegurando una aplicación totalmente funcional y que cumple con los requisitos descritos.

CONCLUSIONES

Una vez culminado el presente trabajo de diploma se logró obtener un componente configurable y totalmente funcional que cumple con los objetivos trazados. Este componente permitirá gestionar las competencias laborales posibilitando una adecuada selección del personal con el Sistema Integral de Gestión Cedrux y aumentar de manera eficiente la capacitación de los recursos humanos en una entidad.

Para obtener la aplicación resultante de este trabajo se realizó un estudio de varios sistemas que incluyen este proceso, concluyendo a la necesidad de crear una solución propia que responda a las características de las entidades cubanas y que cumpla con el paradigma de independencia tecnológica por la que aboga el país. Se detallaron las tecnologías y herramientas a emplear para la implementación del componente y se realizó una descripción de la implementación de los escenarios que dan solución al problema planteado. Por último se realizaron pruebas al sistema que permitieron validar el código fuente y las interfaces pertenecientes al componente competencia laboral como parte de los procesos de gestión de Capital Humano.

En sentido general se obtuvieron resultados satisfactorios que dieron cumplimiento a los objetivos propuestos.

RECOMENDACIONES

- ✓ Continuar realizando pruebas de calidad al componente por parte del Centro de calidad de software CALISOFT.
- ✓ Incluir las opciones de imprimir en los requerimientos.
- ✓ Proseguir con la investigación e identificar nuevos requisitos para actualizar el sistema desarrollado.
- ✓ Realizar el despliegue del componente propuesto como parte del subsistema de Capital humano del sistema Cedrux.

REFERENCIAS BIBLIOGRÁFICAS

Alfaro, Félix Murillo. 1999. Herramientas Case. [En línea] 1999. <http://www.inei.gob.pe/biblioineipub/bancopub/Inf/Lib5103/Libro.pdf>.

Alvaro, Mamani Paco Edwin y Eloy, Sirpa Laura Hector. Análisis y Diseño de sistemas II. *Diagrama de despliegue*.

Baryolo, Oiner Gómez. 2008. Arquitectura tecnológica para el desarrollo de software. 2008.

Ciberaula International Training, S.L. 2010. Ciberaula. *Una Introducción a APACHE*. [En línea] Ciberaula International Training, S.L., 2010. http://linux.ciberaula.com/articulo/linux_apache_intro/.

Cross-Browser, Rich. Ext JS. *Internet Application Framework*. [En línea] <http://www.sencha.com/products/extjs/>.

Definición.de. 2008-2012. Definición.de. [En línea] 2008-2012. <http://definicion.de/modelo-de-datos/>.

Díaz, Dora Rodríguez. 2011. Sistema de Capital Humano en la empresa INPUD. [En línea] 8 de 4 de 2011.

<http://www.gestiopolis.com/organizacion-talento-2/sistema-capital-humano-empresa-inpud-cuba.html>.

Fernando Vargas, Fernando Casanova, Laura Montanaro. 2001. El enfoque de Competencia Laboral. Manual de Formación. Montevideo : Departamento de Publicaciones de Cinterfor/OIT, 2001.

Genix, Alina de la Concepción Isasi, Roblejo, Yenlis González y Guibert, Pedro Valdés. 2011. Estudio sobre la información que se gestiona de los recursos humanos, en los ERP. La Habana : s.n., 2011.

Juristo, Natalia, Moreno, Ana M. y Vegas, Sira. 2005. TÉCNICAS DE EVALUACIÓN DE SOFTWARE. 2005. Vol. 12.

López, Alejandro Cadavid. 2004. Maestros de la web. *Mozilla Firefox, el navegador web del momento*. [En línea] Junio de 2004. <http://www.maestrosdelweb.com/>.

Marin, Marvin David Arias. 2008. Editboard.com. *Definición de lenguaje de programación. Tipos. Ejemplos*. [En línea] 16 de Octubre de 2008. <http://catedraprogramacion.foroactivo.net/t83-definicion-de-lenguaje-de-programacion-tipos-ejemplos>.

Microsoft Corporation. 2011. What's new in Microsoft Dynamics AX 2012. 2011.

Miniet, Yanet Vega, y otros. 2009. Ciclo de vida del proyecto. 2009.

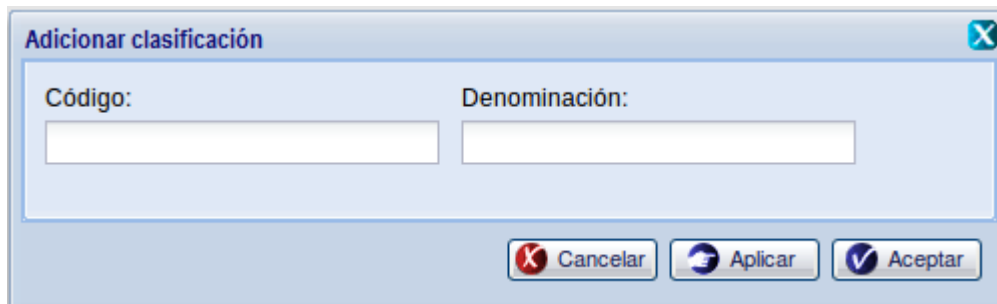
Muñoz, Peggy Karen Cruz y Lopez, Georgina M. Vega. 2001. La Gestión por Competencias: Una nueva herramienta en la planificación estratégica del recurso humano. Antofagasta : s.n., 2001.

- Nicola, Melbys. 2011.** Datazucar ofrece soluciones de negocio. 2011.
- Norma Cubana 3000. 2007.** Sistema de Gestión integrada de capital humano. s.l. : 1 Edición, 2007.
- Oracle Corporation. 2011.** NetBeans. [En línea] 2011. http://netbeans.org/index_es.html.
- Osmosis Latina. 2005.** Osmosis Latina. [En línea] 20 de Octubre de 2005. <http://www.osmosislatina.com/subversion/basico.htm>.
- Pedro Boda, K.N., Martinez, Javier y Gonzales, Benjamín. 2009.** Zend Framework Manual en Español. 2009.
- PostgreSQL Cuba. *Comunidad Técnica Cubana*. [En línea] <http://postgresql.uci.cu/node/63>.
- Pressman, Roger S. 2005.** *Ingeniería del Software: Un enfoque práctico*. 2005.
- QUALEM. 2002.** Administración de la capacitación por competencias. Notas técnicas utilizadas para el diplomado en Recursos Humanos en la Universidad de las Américas. Puebla : Consultoria Empresaria, 2002.
- Quevedo, Yicel Romero, y otros. 2010.** eumed.net. [En línea] 2010. <http://www.eumed.net/ce/2009b/qasr.htm>.
- Santos, Armando Cuesta. 1999.** *Conformación de los perfiles de Competencias Directivas por el método Delphi*. La Habana : s.n., 1999.
- . 2010. *Prof.Dr. Tecnología de Gestion de Recursos Humanos*. s.l. : Felix Varela y Academia, 2010.
- Stig Sæther Bakken, Alexander Aulbach, y otros. 2002.** Manual de PHP. 2002. Vol. II.
- Talentsoft. 2012.** Talentsoft. *Sage Francia gestiona sus competencias con TalentSoft*. [En línea] 2012. <http://www.talentsoftspain.es/?p=154>.
- Taxonomía estructural de Sauxe. .
- 2001.** territoriopc. [En línea] 2001. http://www.territoriopc.com/javascript/tutorial_javascript_introduccion.php.
- Tovar, Rivas. 2002.** *Gestión integral de recursos humanos*. L.A : Taller abierto, 2002.
- Vargas, F. 2004.** La formación basada en competencias en América Latina y el Caribe. Desarrollo reciente. Algunas experiencias. [En línea] 2004. <http://www.cinterfor.org.uy/public/spanish/region/ampro/cinterfor/temas/complab/observ/vargas/iii>.
- Vélez, Alvaro Uribe. 2005.** Instructivo para el ajuste del manual específico de funciones y de competencias laborales. Bogotá : Colombia, 2005.
- Zoder, Nancy. 2010.** PeopleSoft Enterprise Human Capital Management. 2010.

Zúñiga, F. V. 2004. 40 preguntas sobre competencia laboral. 2004.

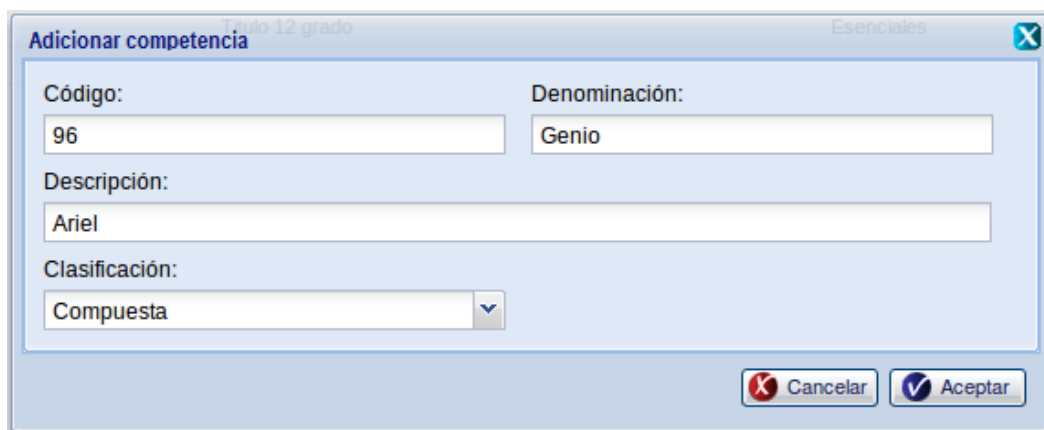
ANEXOS

Anexo 1: Interfaces relacionadas con Gestionar clasificación de competencia y Gestionar competencia.



The screenshot shows a dialog box titled "Adicionar clasificación". It contains two input fields: "Código:" and "Denominación:". Below the fields are three buttons: "Cancelar" (with a red X icon), "Aplicar" (with a blue circular arrow icon), and "Aceptar" (with a blue checkmark icon).

Figura 23. Interfaz Adicionar clasificación de competencia.



The screenshot shows a dialog box titled "Adicionar competencia". It contains four input fields: "Código:" (with the value "96"), "Denominación:" (with the value "Genio"), "Descripción:" (with the value "Ariel"), and "Clasificación:" (with a dropdown menu showing "Compuesta"). Below the fields are two buttons: "Cancelar" (with a red X icon) and "Aceptar" (with a blue checkmark icon). The dialog box also shows "Título 12 grado" and "Esenciales" in the top right corner.

Figura 24. Interfaz Adicionar competencia.

Anexo 2. Acta de liberación



CENTRO PARA LA INFORMATIZACIÓN
DE GESTIÓN DE ENTIDADES

Validación y Verificación de Requisitos (V&V)

ACTA DE LIBERACIÓN

Acta de Liberación, v 1.0, 6/12/2012

Datos del producto

Emitida a favor de: Componente Competencia laboral (Subsistema Capital Humano del Sistema Integral de Gestión Cedrux)

Responsable: Ing. Arnolis Salgueiro Arzuaga

Cargo: Jefe de Proyecto.

Clasificado como:

- Componente.

Detalle de los elementos probados y su estado final:

Artefacto	Estado Final
Aplicación	0 No Conformidad
...	...

Cantidad de iteraciones:

Para la revisión se emplearon un total de 3 iteraciones para lograr el resultado de 0 (cero) No Conformidad.

Elementos revisados o probados y herramientas utilizadas

Elemento	Herramienta
Aplicación	Estándar de interfaz Lista de Chequeo de Diseño de Interfaz

Cantidad total de horas empleadas y rango de fechas:

Se emplearon un total de 4 horas efectivas de trabajo con la siguiente distribución: 2h (06/05/2012), 1h (06/06/2012), 1h (06/07/2012).

Estructura del equipo de prueba empleado y turnos de trabajo:

Las pruebas se realizaron en un total de 2 turnos de trabajo, con un 1 probador en cada turno y toda la actividad estuvo dirigida por un Jefe de Pruebas.

Evaluado por:

Especialista principal Asignado: Ing. Iliannis Pupo Leyva

Otro personal especializado participante:

Nombre	Cargo
Martha Rocío Fonseca Vega	Probador
Ing. Arnolis Salgueiro Arzuaga	Jefe de Proyecto

Ing. Iliannis Pupo Leyva
Especialista del Laboratorio de Calidad

Ing. Arnolis Salgueiro Arzuaga
Responsable por el Equipo de Desarrollo

GLOSARIO

Capital Humano: Cuantificación y valoración de los recursos humanos. Valor de las habilidades, capacidades, experiencias y conocimientos de las personas que integran una organización.

Gestión: Es el conjunto de diligencias que se realizan para desarrollar un proceso o para lograr un producto determinado. Es también la dirección o administración de una empresa o de un negocio.

Entidad: Empresa, unidad presupuestada u otro tipo de organización similar con una gestión económica, financiera, organizativa, técnica, productiva, comercial, laboral y contractual, con autonomía controlada, en cumplimiento de lo establecido por el Gobierno.

Competencias laborales: son los conocimientos, características y habilidades que posee una persona para ejercer determinada ocupación, que deben ser medibles y que de estas depende el logro que pueda llegar a alcanzar una organización.

Proceso: Según lo establecido en el apartado 3.4.1 de la NC ISO 9000:2005, el proceso es un conjunto de actividades mutuamente relacionadas o que interactúan, las cuales transforman elementos de entrada en resultados.

Componente: Elemento de un sistema software que ofrece un conjunto de servicios, o funcionalidades, a través de interfaces definidas.

Subsistema: Cada uno de los componentes principales de un sistema que este dividido en componentes. Cada subsistema abarca aspectos del sistema que comparten alguna propiedad común.

Trabajador: Es aquella persona que ocupa un puesto de trabajo en la entidad y desempeña una labor determinada por la cual recibe haberes o salarios.