



Universidad de Cienfuegos "Carlos Rafael Rodríguez"

Facultad de Ingeniería

Carrera de Ingeniería Informática

Sistema Informático para Apoyar el Proceso de Selección de Recursos Humanos

Trabajo de diploma para optar por el título de Ingeniería en Informática

Autor:

Armando Enrique Pentón Luis.

Tutores:

Msc. Hugandy Álvarez Acosta.

Ing. Jorge Luis de Armas Díaz.

Cienfuegos, Cuba

Curso 2013 – 2014



Dedicatoria

Dedico este trabajo especialmente a mis padres: Rafael Armando Pentón Larrañaga y Adys Luis Ramírez, a toda mi familia en general y a todos aquellos que de una forma u otra hicieron posible la realización de esta investigación.



Resumen

La presente investigación está encaminada a la elaboración de un sistema informático capaz de apoyar el proceso de selección de recursos humanos en la Universidad de Cienfuegos “Carlos Rafael Rodríguez”, basándose en la disminución del tiempo promedio que se utiliza para la asignación de candidatos a cargos, la determinación del cargo adecuado para un candidato o el cubrimiento óptimo de los cargos con los candidatos en correspondencia con las competencias asociadas a cada perfil. Actualmente este proceso se basa en la experiencia de los especialistas, aplicando criterios subjetivos para la evaluación cualitativa y selección de los candidatos, por lo que requiere un prolongado período de tiempo para el análisis de la correspondencia entre las competencias de cada candidato con los requerimientos del perfil. El software propuesto permite automatizar el proceso de selección, una vez que han sido evaluadas las competencias. En el documento de la investigación se describen los elementos del análisis, diseño e implementación del sistema propuesto, en correspondencia con lo establecido por la metodología desarrollo, utilizando el Lenguaje Unificado de Modelado. Para la implementación fue empleado el PostgreSQL como Sistema Gestor de Bases de Datos, Apache como servidor Web, PHP como lenguaje de programación y el framework CodeIgniter. La solución informática resultante de la investigación cumple las necesidades de los usuarios finales, garantizando una reducción significativa del tiempo en el proceso de asignación de candidatos.



Abstract

This research is aimed at developing a computer system capable of supporting the process of selection of human resources at the University of Cienfuegos "Carlos Rafael Rodríguez", based on the decrease in the average time used for the allocation of candidates; the establishment of the appropriate fee for a candidate or the optimal coverage of the charges in correspondence with candidates with the skills associated with each profile. Currently this process is based on the experience of specialists, using subjective criteria for qualitative evaluation and selection of candidates, so it requires a long time to analyze the correspondence between the qualification of each candidate requirements profile. The proposed software allows to await the selection process, once the skills have been evaluated. In the research paper elements of analysis, design and implementation of the proposed system are described, corresponding to the provisions of the development methodology, using the Unified Modeling Language. For the implementation was built on the PostgreSQL System Manager Database, Apache as web server, PHP as the programming language and framework CodeIgniter. The software solution resulting from the research meets the needs of end users, ensuring a significant reduction in the time allocation process candidates.



Índice

Introducción	1
Problema Científico	2
Idea a Defender	3
Objeto de Estudio	3
Campo de Acción	3
Objetivo General.....	3
Objetivos Específicos.....	3
Tareas	3
Marco Metodológico.....	4
Aporte Práctico	5
Capítulo 1: Fundamentación Teórica.....	7
1.1 Introducción	7
1.2 Principales conceptos	7
1.3 Lógica difusa.....	8
1.4 Algoritmos genéticos	9
1.5 Fundamentación matemática	11
1.6 Sistemas automatizados existentes vinculadas al campo de acción	15
1.7 Metodologías de desarrollo	16
1.8 FDD	18
1.9 Lenguaje de modelado, UML	19
1.10 Herramienta de modelado, Visual Paradigm	19
1.11 Aplicaciones web	19
1.12 Elementos de arquitectura.....	20



1.13	Servidor web, Apache	21
1.14	Herramienta para el almacenamiento de la información, PostgreSQL.....	21
1.15	Lenguaje utilizado del lado del servidor, PHP	23
1.16	Framework del lado del servidor, CodeIgniter.....	24
1.17	Lenguaje utilizado del lado del cliente, JavaScript.....	24
1.18	Entorno de desarrollo integrado, NetBeans.....	26
1.19	Herramienta de diseño gráfico, GIMP	27
1.20	Conclusiones.....	27
Capítulo 2: Descripción del sistema		28
2.1	Introducción	28
2.2	Desarrollo de un modelo global.....	28
2.2.1	Flujo actual de procesos involucrados en el campo de acción.....	28
2.2.2	Objeto de automatización	30
2.2.3	Modelo de dominio. Diagrama de objetos.....	31
2.2.4	Reglas a considerar	31
2.3	Construcción de una lista de funcionalidades.....	32
2.3.1	Requerimientos funcionales.....	32
2.3.2	Requerimientos no funcionales.....	37
2.4	Estudio de factibilidad	39
2.4.1	Beneficios tangibles e intangibles	39
2.4.2	Análisis de costo y beneficios	48
2.5	Planeación por funcionalidad	48
2.6	Diseño por funcionalidad.....	48
2.6.1	Principios de diseño.....	49
2.6.2	Modelo conceptual de base de datos.....	49
2.6.3	Modelo lógico de la base de datos.....	51



2.6.4	Modelo físico de la base de datos.....	52
2.7	Conclusiones.....	53
Capítulo 3: Construcción del sistema.....		54
3.1	Introducción	54
3.2	Descripción de la arquitectura propuesta	54
3.3	Diagrama de clases.....	55
3.4	Estándar de implementación	56
3.5	Seguridad del sistema.....	57
3.6	Tratamiento de errores.....	57
3.7	Diagrama de despliegue	58
3.8	Validación de la solución propuesta	59
3.8.1	Resultados estadísticos de las mediciones realizadas	59
3.9	Conclusiones.....	63
Conclusiones.....		64
Recomendaciones.....		65
Referencia Bibliográfica		66
Bibliografía Consultada		68
Anexos		1



Índice de Tablas

Tabla 1.1 Comparación entre metodologías.....	18
Tabla 2.1 Primer bloque de funcionalidades.	32
Tabla 2.2 Segundo bloque de funcionalidades.....	33
Tabla 2.3 Tercer bloque de funcionalidades.....	33
Tabla 2.4 Cuarto bloque de funcionalidades.	34
Tabla 2.5 Quinto bloque de funcionalidades.	34
Tabla 2.6 Sexto bloque de funcionalidades.....	35
Tabla 2.7 Séptimo bloque de funcionalidades.....	35
Tabla 2.8 Octavo bloque de funcionalidades.....	35
Tabla 2.9 Noveno bloque de funcionalidades.....	36
Tabla 2.10 Décimo bloque de funcionalidades.....	36
Tabla 2.11 Onceno bloque de funcionalidades.	37
Tabla 2.12 Entradas externas.	42
Tabla 2.13 Salidas externas.....	42
Tabla 2.14 Consultas externas.....	43
Tabla 2.15 Archivos lógicos internos.....	44
Tabla 2.16 Puntos de función.....	44
Tabla 2.17 Variables escalares.	45
Tabla 2.18 Multiplicadores de esfuerzo.....	46
Tabla 2.19 Resumen de resultados.....	47



Tabla 3.1 Tiempos antes y después de la realización del proceso.	60
--	----



Índice de Figuras

Figura 1.1 Ejemplo de evaluación de aptitud de las soluciones.	12
Figura 1.2 Procedimiento de cruce.	13
Figura 1.3 Estructura del cromosoma.	14
Figura 2.1 Modelo de objetos	31
Figura 2.2 Modelo conceptual de la base de datos.	50
Figura 2.3 Modelo lógico de la base de datos.	51
Figura 2.4 Modelo físico de la base de datos.	52
Figura 3.1 Diagrama de clases del modelo.	55
Figura 3.2 Diagrama de clases del controlador.	56
Figura 3.3 Diagrama de despliegue del sistema.	58
Figura 3.4 Comparación del tiempo promedio antes y después de instalado el sistema.	61
Figura 3.5 Prueba de Kolmogorov-Smirnov.	62
Figura 3.6 Prueba de T.	63



Introducción

A lo largo de la historia de la humanidad, el hombre ha necesitado emplear eficientemente los recursos para producir los bienes materiales que garantizan su subsistencia y desarrollo, así como el incremento de la riqueza espiritual que le ha permitido mejorar su calidad de vida.

Con el desarrollo de las fuerzas productivas, se fueron perfeccionando las organizaciones mediante las cuales se aúnan los esfuerzos, así como los sistemas de dirección que permiten la asignación de diversas funciones a los grupos e individuos con posibilidades de llevarlas a cabo, en aras de lograr los objetivos comunes.

Es así que durante el siglo XX, surgieron tendencias administrativas que pusieron al hombre en el centro de su atención, considerándolo como el principal de los recursos con que cuentan las organizaciones, a partir de su potencialidad de influir decisivamente en el curso de los procesos productivos y de servicios, tanto de forma positiva como negativa, en dependencia de la presencia o no de determinadas competencias para el cumplimiento de las tareas.

La búsqueda de una herramienta que permita ayudar de forma precisa esta operación, constituye la vía para garantizar la asignación de los trabajadores idóneos y minimizar los riesgos que representan el cumplimiento de actividades por personas incompetentes.

El departamento de Recursos Humanos de la Universidad de Cienfuegos controla la gestión del personal del centro.

Este departamento tiene como tarea realizar las siguientes funciones [1]:

- ✓ Cambio de categoría
- ✓ Capacitación
- ✓ Estimulación
- ✓ Evaluación de desempeño



- ✓ Organización del trabajo
- ✓ Planeación
- ✓ Ratificación
- ✓ Contratación
- ✓ Asignación

El proceso de asignación del personal actualmente se realiza a partir de la experiencia de los especialistas, basándose en la evaluación cualitativa del mejor candidato para ubicar en un perfil, mediante criterios subjetivos, teniendo en cuenta la posible utilidad que se obtendrá al realizar dicha asignación; para esto se requiere un prolongado período de tiempo, debido a que para determinar la correspondencia entre las competencias de cada uno de los candidatos con los requerimientos de cada perfil, se reúnen los miembros del tribunal designado, analizando comparativamente los resultados de cada una de las evaluaciones, hasta llegar a la conclusión de cuál es el que más se ajusta, debiendo posteriormente redactar el acta en que se informa el candidato elegido. Esto puede hacer muy engorroso el proceso, principalmente cuando existen varias plazas en convocatoria y resulta necesario repetir para cada una todo lo anteriormente descrito.

Estas dificultades constituyen la **situación problemática** que da lugar a la realización de la presente investigación.

Problema Científico

¿Cómo disminuir el tiempo en la asignación de candidatos a cargos, en el Departamento de Recursos Humanos?



Idea a Defender

La aplicación informática para asignar candidatos a cargos teniendo en cuenta el grado de cumplimiento de las competencias asociadas a cada perfil, sobre un entorno web, contribuye a la disminución del tiempo necesario para el desarrollo del proceso.

Objeto de Estudio

El proceso de asignación de recursos humanos.

Campo de Acción

Asignación de candidatos a cargos a partir del nivel de cumplimiento de las competencias, en el Departamento de Recursos Humanos de la Universidad de Cienfuegos.

Objetivo General

Implementar una aplicación informática para disminuir el tiempo que se utiliza en la asignación de candidatos a cargos, en el proceso de selección de recursos humanos.

Objetivos Específicos

1. Analizar teorías, investigaciones y antecedentes relacionados con el proceso de selección de recursos humanos y técnicas de inteligencia artificial.
2. Diseñar el software propuesto.
3. Implementar una aplicación que se ajuste a las necesidades del Departamento de Recursos Humanos.
4. Validar la aplicación propuesta.

Tareas

1. Análisis de teorías e investigaciones relacionadas con la inteligencia artificial, especificando lo relacionado con la gestión de recursos humanos.
2. Estudio de los software existentes para la asignación de recursos.



3. Estudio de modelos matemáticos para la selección de personal utilizando la inteligencia artificial, determinando el más adecuado para la investigación.
4. Análisis de factibilidad de la construcción de un software para la asignación de candidatos basada en competencias.
5. Diseño de la base de datos.
6. Diseño de la interfaz de la aplicación.
7. Instalación del software para su etapa de prueba.
8. Documentación de la información generada durante el análisis, diseño, implementación y validación del sistema.

Marco Metodológico

Los métodos científicos empleados para el desarrollo de la investigación son los siguientes:

1. Métodos generales.
 - a) Hipotético-deductivo: para elaborar la idea a defender y proyectar las Tareas de Investigación.
 - b) Sistémico: para el desarrollo del software logrando que los elementos que lo conforman constituyan un todo que funcione armónicamente.
 - c) Histórico-lógico: para el estudio crítico de trabajos anteriores relacionados con el tema, utilizándolos como punto de referencia para la comparación de los resultados.
2. Métodos lógicos.
 - a) Analítico-sintético: para descomponer el problema de investigación en elementos separados con vistas a su estudio, para luego sintetizarlos en la solución de la propuesta.

b) Inducción-deducción: como vía de la constatación teórica durante el desarrollo de la investigación.

c) Modelación: para el desarrollo de los algoritmos.

3. Métodos empíricos.

a) Coloquial: para la presentación y discusión de los resultados.

b) Estadístico descriptivo: para la validación de la aplicación informática a través de la medición de los tiempos empleados en el proceso de selección de personal antes y después de su implementación.

Aporte Práctico

Se dota al Departamento de Recursos Humanos de la Universidad de Cienfuegos, con una herramienta informática capaz de facilitar los procesos de asignación de candidatos, incrementando su eficiencia sobre la base de las competencias, lo cual se logra a partir de un modelo matemático de tres dimensiones, que se ejecuta sobre un entorno web y utiliza las ventajas que han traído consigo la madurez de estándares como HTML5, CSS3 y la interoperabilidad.

El Trabajo de Diploma se estructura en tres capítulos mediante los cuales se logra la siguiente organización del contenido:

Capítulo I: “Fundamentación Teórica”. Agrupa la conceptualización del empleo de la lógica difusa y los algoritmos genéticos en la construcción de software, la descripción de las prácticas más extendidas para la asignación de recursos humanos, así como el análisis de investigaciones existentes, vinculados con el objeto de estudio.

Capítulo II: “Descripción del sistema”: Se describen los procesos del negocio y los principios de diseño para crear la interfaz. Son identificados los requerimientos funcionales y no funcionales y se realiza el modelo lógico y físico de la base de datos. Por último se realiza el estudio de la factibilidad del sistema.



Capítulo III: “Construcción del sistema”: En este capítulo se realiza una descripción de la construcción del sistema, usando los diagramas de clases por cada capa y la descripción de la arquitectura. Se detalla lo relacionado con la seguridad y el tratamiento de errores, así como la validación.

Como resultado del procesamiento de toda la información, se emiten las conclusiones generales y las recomendaciones.



Capítulo 1: Fundamentación Teórica

1.1 Introducción

Este capítulo constituye el marco referencial de la investigación, que incluye el estudio de un grupo de investigaciones anteriores vinculadas al campo de acción, la comparación entre los aspectos que caracterizan a las metodologías para el desarrollo de software, así como los principales conceptos relacionados con el dominio del problema. Además, se realiza el análisis del objeto de estudio y de la situación problémica que da lugar a la presente investigación. Por último, son expuestas las diferentes metodologías, tecnologías y herramientas que pueden ser utilizadas en la elaboración de la aplicación informática propuesta.

1.2 Principales conceptos

Problema de Asignación: Es la ubicación de recursos que se destinan a cierto grupo de tareas. Su objetivo es seleccionar al candidato idóneo para desempeñar una función específica, lo que precisa ser más coherentes y no simplistas en el tratamiento de la información.

Candidato: “Persona que pretende alguna dignidad, honor o cargo”[2]. Para esta investigación: Persona específica propuesta o con posibilidades para cumplir una determinada función.

Perfil: “Conjunto de rasgos particulares que caracterizan a alguien o algo”[2]. Para esta investigación: Conjunto de condiciones particulares que debe cumplir el candidato para desempeñar una función específica.

Competencia: Para esta investigación: Cada una de las características específicas que conforman un perfil.



1.3 Lógica difusa

La **lógica difusa** o **lógica heurística** se basa en lo relativo de lo observado como posición diferencial. Este tipo de lógica toma dos valores aleatorios, pero contextualizados y referidos entre sí. Así, por ejemplo, una persona que mida 2 metros es claramente una persona alta, si previamente se ha tomado el valor de persona baja y se ha establecido en 1 metro. Ambos valores están contextualizados a personas y referidos a una medida métrica lineal. [3]

Aplicaciones generales:

La lógica difusa se utiliza cuando la complejidad del proceso en cuestión es muy alta y no existen modelos matemáticos precisos, para procesos altamente no lineales y cuando se envuelven definiciones y conocimiento no estrictamente definido (impreciso o subjetivo).

En cambio, no es una buena idea usarla cuando algún modelo matemático ya soluciona eficientemente el problema, cuando los problemas son lineales o cuando no tienen solución.[3]

Esta técnica se ha empleado con bastante éxito en la industria, principalmente en Japón, y cada vez se está usando en gran multitud de campos. La primera vez que se usó de forma importante fue en el metro japonés, con excelentes resultados. A continuación se citan algunos ejemplos de su aplicación [3]:

- ✓ Sistemas de control de acondicionadores de aire.
- ✓ Sistemas de foco automático en cámaras fotográficas.
- ✓ Electrodomésticos familiares (frigoríficos, lavadoras, etc).
- ✓ Optimización de sistemas de control industriales.
- ✓ Sistemas de escritura.
- ✓ Mejora en la eficiencia del uso de combustible en motores.



- ✓ Sistemas expertos del conocimiento (simular el comportamiento de un experto humano).
- ✓ Tecnología informática.
- ✓ Bases de datos difusas: Almacenar y consultar información imprecisa. Para este punto, por ejemplo, existe el lenguaje FSQL.

1.4 Algoritmos genéticos

Un algoritmo es una serie de pasos organizados que describe el proceso que se debe seguir, para dar solución a un problema específico. En los años 1970, de la mano de John Henry Holland, surgió una de las líneas más prometedoras de la inteligencia artificial, la de los algoritmos genéticos. Son llamados así porque se inspiran en la evolución biológica y su base genético-molecular. Estos algoritmos hacen evolucionar una población de individuos sometiéndola a acciones aleatorias semejantes a las que actúan en la evolución biológica (mutaciones y recombinaciones genéticas), así como también a una Selección de acuerdo con algún criterio, en función del cual se decide cuáles son los individuos más adaptados, que sobreviven, y cuáles los menos aptos, que son descartados. Es incluido dentro de los algoritmos evolutivos, que incluyen también las estrategias evolutivas, la programación evolutiva y la programación genética. [4]

Funcionamiento:

Los algoritmos entran en el conjunto de soluciones de un problema, llamado fenotipo, y el conjunto de individuos de una población natural, codificando la información de cada solución en una cadena, generalmente binaria, llamada cromosoma. Los símbolos que forman la cadena son llamados genes. Cuando la representación de los cromosomas se hace con cadenas de dígitos binarios se le conoce como genotipo. Los cromosomas evolucionan a través de iteraciones, llamadas generaciones. En cada generación, los cromosomas son evaluados usando alguna medida de aptitud.

Aplicaciones [4]:

- ✓ Diseño automatizado, incluyendo investigación en diseño de materiales y diseño multiobjetivo de componentes automovilísticos: mejor comportamiento ante choques, ahorros de peso, mejora de aerodinámica, etc.
- ✓ Diseño automatizado de equipamiento industrial.
- ✓ Diseño automatizado de sistemas de comercio en el sector financiero.
- ✓ Construcción de árboles filogenéticos.
- ✓ Optimización de carga de contenedores.
- ✓ Diseño de sistemas de distribución de aguas.
- ✓ Diseño de topologías de circuitos impresos.
- ✓ En Teoría de juegos, resolución de equilibrios.
- ✓ Análisis de expresión de genes.
- ✓ Aprendizaje de reglas de lógica difusa.
- ✓ Análisis lingüístico, incluyendo inducción gramática, y otros aspectos de Procesamiento de lenguajes naturales, tales como eliminación de ambigüedad de sentido.
- ✓ Infraestructura de redes de comunicaciones móviles.
- ✓ Optimización de estructuras moleculares.
- ✓ Planificación de producción multicriterio.
- ✓ Predicción.
- ✓ Aplicación de Algoritmos Genéticos al Dilema del Prisionero Iterado
- ✓ Optimización de sistemas de compresión de datos, por ejemplo, usando wavelets.



- ✓ Optimización de Layout.
- ✓ Predicción de estructura de ARN.
- ✓ En bioinformática, Alineamiento múltiple de secuencias.
- ✓ Aplicaciones en planificación de procesos industriales, incluyendo planificación job-shop.
- ✓ Selección óptima de modelos matemáticos para la descripción de sistemas biológicos.
- ✓ Manejo de residuos sólidos.
- ✓ Ingeniería de software.
- ✓ Construcción de horarios en grandes universidades, evitando conflictos de clases.
- ✓ Hallazgo de errores en programas.
- ✓ Calibración y detección de daños en estructuras civiles.

1.5 Fundamentación matemática

Como parte del proceso de análisis bibliográfico para el desarrollo del presente trabajo, fueron consultadas investigaciones que se relacionan con el desarrollo de algoritmos para la asignación de personal y solución de problemas organizativos, llevadas a cabo en el mundo y en particular en Cuba, las cuales tienen como factor común el empleo de algoritmos genéticos y en general de la inteligencia artificial. Entre dichos estudios se encuentran:

- ✓ La selección del personal con un algoritmo genético borroso
- ✓ Asignación de recursos humanos mediante el uso de algoritmos genéticos
- ✓ Algoritmos de trayectoria multiobjetivo aplicados al problema de asignación de recursos humanos a equipos de proyecto de software
- ✓ Algoritmos genéticos aplicados a problemas de planificación

haya sido asignado a la tarea 1, por lo tanto esta operación se debe realizar para las demás 9 asignaciones del cromosoma.

Para llevar a cabo el proceso de selección se emplea el método elitismo, el cual consiste en seleccionar los mejores individuos de una generación para ser padres de un porcentaje de hijos de la siguiente, evitando que se destruyan buenas soluciones al emplear operadores genéticos. Por otra parte, la mejor solución de cada generación es copiada a la siguiente generación después del último individuo generado por el porcentaje seleccionado de mejores padres para que no se pierda.

El cruce se hace por un punto, pues es el método más efectivo cuando se repiten números. Por ejemplo:

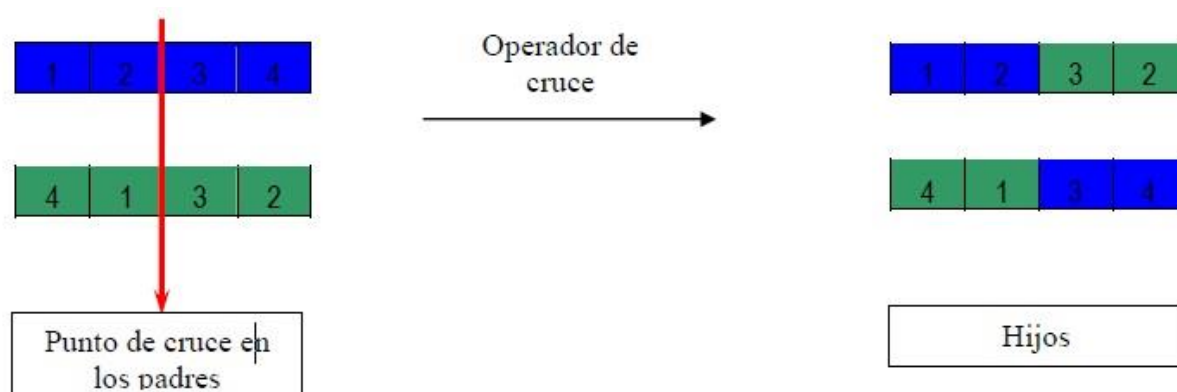


Figura 1.2 Procedimiento de cruce.

Para añadir diversidad a las soluciones se genera un número aleatorio entre 0 y un porcentaje determinado de mutación, cada vez que se va a hacer una copia de un bit en el proceso de cruce. Si el número aleatorio está entre 0 y el porcentaje de cruzamiento, se genera un número aleatorio entre 1 y el número de candidatos que reemplaza al bit a ser copiado; de lo contrario se copia el que corresponde.

Cada solución será una cadena con un número de bits igual al número de tareas a asignar. Por ejemplo, para el caso de 10 tareas una posible solución sería:

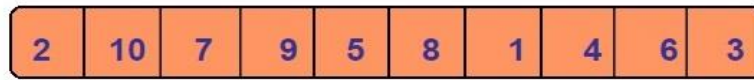


Figura 1.3 Estructura del cromosoma.

Donde el candidato 2 ejecutaría la tarea 1, por ocupar la primera posición dentro de la cadena, el candidato 10 la tarea 2 y así sucesivamente.

La Función Objetivo de dicho algoritmo es la siguiente:

Teniendo:

$$\sum_{i=1}^a C = \{X_1; X_2; \dots; X_n\}$$

Siendo:

C = Cromosoma.

n = Cantidad de genes del cromosoma.

Entonces:

$$FO = \sum_{i=1}^n \left(\sum_{j=1}^m (Ct_{ji} * CC_{jci}) \right) + \left(\sum_{b=1}^k (Net_b * NE_{bci}) \right) + \left(\sum_{p=1}^t (TEt_{pi} * TE_{pci}) \right)$$

Donde:

m = Cantidad de filas de la matriz de importancia.

k = Cantidad de filas de la matriz de experiencia.

t = Tamaño del vector de tiempo.

Ct: Matriz de importancia de las competencias de los perfiles.



CC: Matriz de calificaciones de los candidatos en las competencias.

NEt: Matriz de la experiencia requerida por los perfiles.

NE: Matriz de experiencia de los candidatos en los perfiles.

TEt: Matriz de tiempo requerido por los perfiles.

TE: Matriz de tiempo disponible por los candidatos para los perfiles.

Luego:

$$f (F_0) = \text{Max} (F_0)$$

1.6 Sistemas automatizados existentes vinculadas al campo de acción

ArmSTRONG: Incluye la automatización de evaluaciones cognoscitivas y la aplicación de baterías psicométricas, así como el proceso de entrevistas digitalizadas (general, de presión, profunda y de salida), administración de competencias, generación de perfiles de puesto y evaluaciones de desempeño, además de contar con retroalimentación mediante un buzón de sugerencias electrónico y la creación del expediente electrónico del empleado. Está integrado por un grupo de módulos que funcionan individualmente y se complementan entre sí, en un ambiente de red o de forma local, bajo la plataforma Windows.

HUM&SOFTWARE: Creado para satisfacer todas las necesidades del proceso de Reclutamiento, Selección y Evaluación de Personal en las Empresas así como para los Outsourcers dedicados a esta actividad. Este Software incluye la Bolsa de Trabajo electrónica, la automatización de Pruebas Psicométricas y permite llevar un control total de las Requisiciones de Personal dando un seguimiento detallado al proceso de cada uno de los candidatos.

Huma Nex: Diseñado para ayudar y contribuir a la mejora de la gestión del proceso completo de selección y la gestión de ofertas. Ofrece funcionalidades de búsqueda para la identificación de los candidatos más apropiados, permite la gestión del proceso de selección. Ofrece un informe de seguimiento de la acción de selección on line.



Hum&Select: Elaborado para satisfacer las necesidades del proceso de reclutamiento, selección y evaluación psicométrica de personal en las empresas. Ofrece funcionalidades como el seguimiento detallado al proceso de reclutamiento de cada uno de los candidatos, control de las requisiciones de personal, compara las habilidades de sus candidatos contra el perfil de cada puesto en su empresa, etcétera.

HUNTER HRMS: es un Software de selección de personal creado por la empresa NILOOSOFT que desarrolla Software de reclutamiento y selección de personal. Es distribuido en México y América Latina por empresa que ofrece el Software reclutamiento y selección de personal para áreas de Recursos Humanos.

Los sistemas anteriormente mencionados son software privados, colocados en línea para su empleo de forma directa por los clientes potenciales mediante el pago previo de los derechos; es decir, la organización que requiera su empleo debe contratar el servicio a la empresa correspondiente y utilizarlo directamente en la red. Esto trae consigo los siguientes inconvenientes principales:

- ✓ Su uso significa una erogación financiera a una empresa extranjera. Además de la afectación económica, esto tiene como dificultad que la mayoría de dichas empresas no contratarían el servicio a instituciones cubanas como efecto del bloqueo económico.
- ✓ En general, aunque poseen funcionalidades que se corresponden con las necesidades del Departamento de Recursos Humanos de la Universidad de Cienfuegos, poseen otras que no le son necesarias y no cuentan con la generación del ranking de candidatos y la impresión de modelos e informes.

1.7 Metodologías de desarrollo

Las metodologías de desarrollo de software son un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información.

A lo largo del tiempo, una gran cantidad de métodos han sido desarrollados diferenciándose por su fortaleza y debilidad.

Existen dos tipos de metodologías de desarrollo de software las ágiles (FDD, ASD, AUP, XP, Scrum, etcétera) y las pesadas. La tabla que se muestra a continuación hace una breve comparación entre metodologías de desarrollo ágiles y metodologías de desarrollo pesadas.[6]

Metodologías Ágiles	Metodologías Pesadas
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Especialmente preparados para cambios durante el proyecto
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos

Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Tabla 1.1 Comparación entre metodologías.

En la presente investigación utilizaremos la metodología ágil FDD, por sus ventajas de poco tiempo de elaboración, grupos pequeños de trabajo y aceptación por parte de los desarrolladores.

1.8 FDD

FDD con sus siglas en inglés Feature Driven Development es un enfoque ágil para el desarrollo de sistemas. Fue desarrollado por Jeff De Luca y el viejo gurú de la Orientación a Objetos. Como las otras metodologías adaptables, se enfoca en iteraciones cortas que entregan funcionalidad tangible. Dicho enfoque no hace énfasis en la obtención de los requerimientos sino en cómo se realizan las fases de diseño y construcción. Sin embargo, fue diseñado para trabajar con otras actividades de desarrollo de software y no requiere la utilización de ningún modelo de proceso específico. Además, hace énfasis en aspectos de calidad durante todo el proceso e incluye un monitoreo permanente del avance del proyecto. Al contrario de otras metodologías, FDD afirma ser conveniente para el desarrollo de sistemas críticos y para proyectos pequeños. [7]

Fases de FDD [7]:

- ✓ Desarrollo de un modelo global.
- ✓ Construcción de una lista de funcionalidades.
- ✓ Plan de release en base a las funcionalidades a implementar.
- ✓ Diseñar en base a las funcionalidades.



- ✓ Implementar en base a las funcionalidades.

1.9 Lenguaje de modelado, UML

UML o Lenguaje Unificado de Modelado, está destinado para la especificación, visualización, construcción y documentación de los artefactos de un proceso de sistema intensivo. [8]

UML está consolidado como lenguaje estándar en el análisis y diseño de sistemas de cómputo. Mediante este lenguaje es posible establecer la serie de requerimientos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código. [8]

Aunque UML es un lenguaje, posee más características visuales que programáticas, las cuales facilitan a los integrantes de un equipo multidisciplinario participar e intercomunicarse fácilmente, siendo estos los analistas, diseñadores, especialistas de área y desde luego los programadores. [8]

1.10 Herramienta de modelado, Visual Paradigm

Visual Paradigm para UML es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores a menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. [9]

1.11 Aplicaciones web

En la ingeniería de software se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. Es decir, una aplicación que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución del navegador.



Características:

- ✓ Portabilidad de los datos.
- ✓ Solución a problemas reales.
- ✓ Multiplataforma.
- ✓ Deberían tener una visión.
- ✓ Deberían estar en constante mejora.
- ✓ Deberían tener una excelente documentación.
- ✓ Deberían ser amistosas.
- ✓ Visualmente atractivas.

1.12 Elementos de arquitectura

El concepto de arquitectura de software está muy extendido, sin embargo no existe una definición única y estándar para este concepto. Un autor, cuya contribución en el mundo de las arquitecturas de software y las componentes ha sido muy aceptada, es David Garlan. Su definición de arquitectura es una de las más utilizadas en la literatura. “La arquitectura de software está compuesta por la estructura de los componentes de un programa o sistema, sus interrelaciones, y los principios y reglas que gobiernan su diseño y evolución a lo largo del tiempo.” Por las características del proyecto y por las tendencias de las tecnologías en la actualidad se pretende implementar una arquitectura Modelo - Vista - Controlador y se empleará la programación orientada a objetos pues la aplicación se desarrollará con PHP4, este es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones, está basado en varias técnicas, incluyendo herencia, modularidad, polimorfismo, y encapsulamiento. Se hará uso además de frameworks que organicen toda la estructura de la aplicación y minimicen el tiempo de desarrollo.



Modelo Vista Controlador (MVC)

Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos. El patrón se observa frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la capa de acceso a datos, y el controlador es el responsable de recibir los eventos de entrada desde la vista y de la lógica del negocio. [10]

1.13 Servidor web, Apache

El servidor Web es una herramienta que implementa el protocolo HTTP (Hypertext Transfer Protocol). Está elaborado para traducir lenguajes y sentencias a una interfaz entendible por el usuario, entre ellos: textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonido.

El servidor Apache es un servidor HTTP de código abierto para varias plataformas. Presenta mensajes de error altamente configurables, Base de Datos de Autenticación y negociado de contenidos. Es el servidor HTTP más usado en la actualidad. El mismo es capaz de transformar lenguaje PHP a lenguaje HTML que es el que soporta el navegador con el cual se le presenta la interfaz al usuario.

1.14 Herramienta para el almacenamiento de la información, PostgreSQL

PostgreSQL es un servidor de base de datos relacional orientada a objetos de software libre, liberado bajo la licencia BSD. El desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group). [11]

Principales características [11]:

- ✓ Alta concurrencia: Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso



escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo commit. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

- ✓ Amplia variedad de tipos nativos: PostgreSQL provee nativamente soporte para:
 - Números de precisión arbitraria.
 - Texto de largo ilimitado.
 - Figuras geométricas (con una variedad de funciones asociadas)
 - Direcciones IP (IPv4 e IPv6).
 - Bloques de direcciones estilo CIDR.
 - Direcciones MAC.
 - Arrays

Funciones [11]:

Bloques de código que se ejecutan en el servidor. Pueden ser escritos en varios lenguajes, con la potencia que cada uno de ellos da, desde las operaciones básicas de programación, tales como bifurcaciones y bucles, hasta las complejidades de la programación orientada a objetos o la programación funcional.

Los disparadores (triggers en inglés) son funciones enlazadas a operaciones sobre los datos.

Algunos de los lenguajes que se pueden usar son los siguientes:

- ✓ Un lenguaje propio llamado PL/PgSQL (similar al PL/SQL de Oracle).
- ✓ C.
- ✓ C++.



- ✓ Java PL/Java web.
- ✓ PL/Perl.
- ✓ PL/Python.
- ✓ PL/Ruby.
- ✓ PL/sh.
- ✓ PL/Tcl.
- ✓ PL/Scheme.
- ✓ Lenguaje para aplicaciones estadísticas R por medio de PL/R.

PostgreSQL soporta funciones que retornan "filas", donde la salida puede tratarse como un conjunto de valores que pueden ser tratados igual a una fila retornada por una consulta (query en inglés).

Las funciones pueden ser definidas para ejecutarse con los derechos del usuario ejecutor o con los derechos de un usuario previamente definido. El concepto de funciones, en otros DBMS, son muchas veces referidas como "procedimientos almacenados".

1.15 Lenguaje utilizado del lado del servidor, PHP

PHP es un lenguaje de programación muy potente que, junto con HTML, permite crear sitios Web dinámicos. PHP se instala en el servidor y funciona con versiones de Apache, Microsoft IIS, Netscape Enterprise Server y otros. [12]

La forma de usar PHP es insertando código del lenguaje dentro del código HTML de un sitio Web. Cuando un cliente visita la página Web que contiene éste código, el servidor lo ejecuta y el cliente sólo recibe el resultado. Su ejecución, es por tanto en el servidor, a diferencia de otros lenguajes de programación que se ejecutan en el navegador. [12]

PHP permite la conexión a numerosas bases de datos, incluyendo MySQL, Postgres, Oracle, ODBC, IBMDB2, Microsoft SQL Server y SQLite. PHP tiene la capacidad de ser ejecutado en



la mayoría de los sistemas operativos tales como UNIX, Linux, Windows y MacOSX, y puede interactuar con los servidores de Web más populares. [12]

1.16 Framework del lado del servidor, CodeIgniter

CodeIgniter es un conjunto de herramientas para construir aplicaciones web usando PHP. Su objetivo es permitir desarrollar proyectos mucho más rápido de lo que podría si lo escribiese desde cero, provee un rico juego de librerías para tareas comúnmente necesarias, así como una interface simple y estructura lógica para acceder a esas librerías. CodeIgniter permite creativamente enfocarse en el proyecto minimizando la cantidad de código necesaria para una tarea dada. [13]

CodeIgniter se encuentra bajo una licencia open source Apache/BSD-style, así que puede ser usado en cualquier lugar. [13]

CodeIgniter está escrito para ser compatible con PHP 4. Aunque hubiese sido mucho mejor tomar ventaja del manejo de objetos en PHP 5 ya que hubiese simplificado algunas cosas, se ha tenido que buscar soluciones creativas. Los principales proveedores de Sistemas Operativos como RedHat se están moviendo lentamente a soportar PHP 5, y es poco probable que lo hagan en el corto plazo, así que no serviría a los mejores intereses de la comunidad de PHP escribir CodeIgniter en PHP 5. [13]

1.17 Lenguaje utilizado del lado del cliente

HTML

HTML (Hyper Text Markup Language) es el lenguaje con el que se escriben las páginas web. Es un lenguaje de hipertexto, es decir, un lenguaje que permite escribir texto de forma estructurada, y que está compuesto por etiquetas, que marcan el inicio y el fin de cada elemento del documento.

Un documento hipertexto no sólo se compone de texto, puede contener imagen, sonido, video, etc., por lo que el resultado puede considerarse como un documento multimedia. Los documentos HTML deben tener la extensión html o htm, para que puedan



ser visualizados en los navegadores (programas que permiten visualizar las páginas web).[14]

CSS

Hojas de Estilo en Cascada (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos. [15]

CSS se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los Estilos definen la forma de mostrar los elementos HTML y XML. CSS permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento. [15]

JavaScript

JavaScript es un lenguaje interpretado, al igual que Visual Basic Script, Perl, TCL, (Lenguajes de script) sin embargo, posee una característica que lo hace especialmente idóneo para trabajar en Web, porque es el navegador que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. Con JavaScript podemos crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones JavaScript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador.

JavaScript es el siguiente paso, después del HTML, que puede dar un programador de la Web que decida mejorar sus páginas y la potencia de sus proyectos. Es un lenguaje de programación bastante sencillo y pensado para hacer las cosas con rapidez, a veces con



ligereza. Incluso las personas que no tengan una experiencia previa en la programación podrán aprender este lenguaje con facilidad y utilizarlo en toda su potencia con solo un poco de práctica.

“Javascript es un lenguaje orientado a objetos. El modelo de objetos JavaScript está reducido y simplificado, pero incluye los elementos necesarios para que los Scripts puedan acceder a la información de una página y puedan interactuar sobre la interfaz del navegador”.

Entre las acciones típicas que se pueden realizar en JavaScript tenemos dos vertientes. Por un lado los efectos especiales sobre páginas Web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. Por el otro, JavaScript nos permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que podemos crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo. [16]

1.18 Entorno de desarrollo integrado, NetBeans

NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto libre y gratuito sin restricciones de uso. [17]

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Micro Systems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos. [17]

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.



1.19 Herramienta de diseño gráfico, GIMP

El tratamiento de imágenes se hace necesario cada vez más en un entorno de trabajo donde la elaboración de documentos electrónicos es básica para cualquier tarea diaria. GIMP es una herramienta, de gran potencia y reconocido prestigio en el entorno del software libre. Se trata de un programa que permite retocar fotografías, crear imágenes para la Web y otras muchas utilidades. Se presenta bajo licencia libre, lo que permite al usuario distribuirlo y adaptarlo a sus necesidades. Otra característica interesante es su disponibilidad tanto para plataformas Linux como Windows con lo que se garantiza la posibilidad de una implantación progresiva y adaptada a las necesidades de la administración. La interfaz de GIMP está disponible en varios idiomas, entre ellos: español, inglés, catalán, gallego, euskera, francés, italiano, ruso, sueco, noruego, coreano y neerlandés. [18]

GIMP es también conocido por ser quizás la primera gran aplicación libre para usuarios no profesionales o expertos. Productos originados anteriormente, como GCC, el núcleo Linux, etc., eran principalmente herramientas de programadores para programadores. GIMP es considerado por algunos como una demostración fehaciente de que el proceso de desarrollo de software libre puede crear aplicaciones que los usuarios comunes, no avanzados, pueden usar de manera productiva. De esta forma, Gimp ha abierto el camino a otros proyectos como KDE, GNOME, Mozilla Firefox, OpenOffice.org y otras aplicaciones posteriores.

1.20 Conclusiones

Como resultado del estudio de investigaciones vinculadas con el campo de acción en el mundo y en Cuba, así como de la situación actual del objeto de estudio, se puede concluir que dada la importancia que reviste en la actualidad la asignación de los recursos humanos sobre la base de las competencias para la competitividad e incluso la supervivencia de las organizaciones, se ratifica la necesidad del desarrollo de una aplicación informática que contribuya a perfeccionar este proceso en la Universidad de Cienfuegos.

Además, quedaron definidas las metodologías, lenguajes, tecnologías y herramientas más adecuadas para su empleo en el desarrollo de la aplicación.



Capítulo 2: Descripción del sistema

2.1 Introducción

En el presente capítulo se realiza la descripción de cada una de las fases de la metodología FDD, empleada en la investigación por las ventajas que presenta con respecto al resto, así como del flujo actual de los procesos que intervienen en el campo de acción. Además son explicados el objeto de automatización, los conceptos fundamentales asociados al dominio del sistema, la elaboración del modelo de objeto, la especificación de los requerimientos del sistema y el estudio de factibilidad de la aplicación.

2.2 Desarrollo de un modelo global

Cuando esta fase comienza, los expertos del dominio ya tienen una idea del contexto y los requerimientos del sistema. Es probable que el documento de especificación de requerimientos ya exista. Sin embargo, FDD no hace énfasis en la recolección y administración de los requerimientos. [7]

El dominio global es dividido en diferentes áreas y se realiza un informe detallado para cada una de dichas áreas por parte de los expertos del dominio. Luego de cada informe, un grupo de desarrolladores realizan un modelo de objetos para el área del dominio. Además, el equipo de desarrollo discute y decide cual es el modelo de objetos más apropiado para cada área del dominio. Simultáneamente, el modelo global del sistema es construido. [7]

2.2.1 Flujo actual de procesos involucrados en el campo de acción

Entre las actividades que se desarrollan en el Departamento de Recursos Humanos de la Universidad de Cienfuegos “Carlos Rafael Rodríguez”, se encuentra la asignación de personal para el desempeño de los puestos de trabajo basada en sus competencias, a partir de los candidatos existentes, proceso que será objeto de automatización mediante el presente trabajo.



El proceso de selección e integración de candidatos fluye de la siguiente forma (Anexo 1):

La facultad o departamento que presente alguna plaza vacante informa a Recursos Humanos la necesidad de realizar el proceso para cubrirla y le entrega el perfil de la misma, a partir de lo cual son establecidos los requisitos generales para desempeñar las funciones correspondientes y es puesta en convocatoria durante siete días hábiles, utilizando todos los medios disponibles: murales informativos de la universidad, medios de difusión masiva, sitio web, etcétera.

A partir de este paso, el proceso posee dos vertientes en dependencia de si las plazas en convocatoria se clasifican como docentes o no docentes, además de una tercera destinada diferenciadamente a la ubicación de adiestrados.

Docentes:

Se realiza mediante reclutamiento mixto; es decir, que pueden presentarse candidatos tanto pertenecientes a la universidad como de fuera de ella, en igualdad de condiciones.

Cada uno de los candidatos entrega a la facultad o área que oferta la plaza su solicitud por escrito, unida al resto de la documentación requerida, seguido de lo cual se presenta ante un tribunal designado para evaluar el ejercicio de oposición, consistente en la impartición de una clase comprobatoria, de conjunto con la información personal aportada en los documentos presentados.

En caso de que el candidato no satisfaga los requisitos, se le informa directamente. El tribunal informa al Departamento de Recursos Humanos mediante un acta los datos del candidato que es considerado como más idóneo para ocupar la plaza, a partir de lo cual se le realiza la contratación por tiempo indeterminado.

No docentes:

Se realiza en primera instancia mediante reclutamiento interno entre los trabajadores de la propia universidad. En caso de no presentarse ningún candidato interno, entonces se realiza el reclutamiento externo.



Los candidatos entregan a la facultad o área la solicitud por escrito, junto con el resto de la documentación requerida, la que se evalúa por un Comité de Expertos para determinar cuál reúne los requisitos necesarios, informándoles directamente a los candidatos no seleccionados y mediante un acta al Departamento de Recursos Humanos los datos del que se determina que puede ocupar la plaza.

Al candidato seleccionado se le realiza un contrato por período a prueba de 1 a 3 meses, luego de lo cual es evaluado por el Comité de Ingreso de la universidad. Si es desaprobado se le informa al interesado y si se aprueba se elabora un acta que se remite al Departamento de Recursos Humanos para efectuar el contrato por tiempo indeterminado.

Adiestrados:

Cada adiestrado entrega al Departamento de Recursos Humanos la boleta de adiestramiento. Seguidamente se le realiza el contrato y se establece el programa de orientación, inducción y seguimiento, el cual se cumple durante el período de servicio social, teniendo la posibilidad al concluir el mismo de permanecer en el puesto de acuerdo a su desempeño e intereses, para lo cual debe cumplir una de las dos vertientes del proceso anteriormente descritas.

2.2.2 Objeto de automatización

Se pretende automatizar los procesos relacionados con la selección de candidatos basada en competencias, enfocándose específicamente en las vertientes docente y no docente. En el sistema se almacenarán los datos relacionados con las convocatorias, perfiles, competencias y candidatos. Así mismo se introducirá la evaluación de cada candidato en las competencias correspondientes a los perfiles, a partir de lo cual determinará el más adecuado para ocupar la plaza vacante. Por último generará un acta con los resultados obtenidos.

2.2.3 Modelo de dominio. Diagrama de objetos

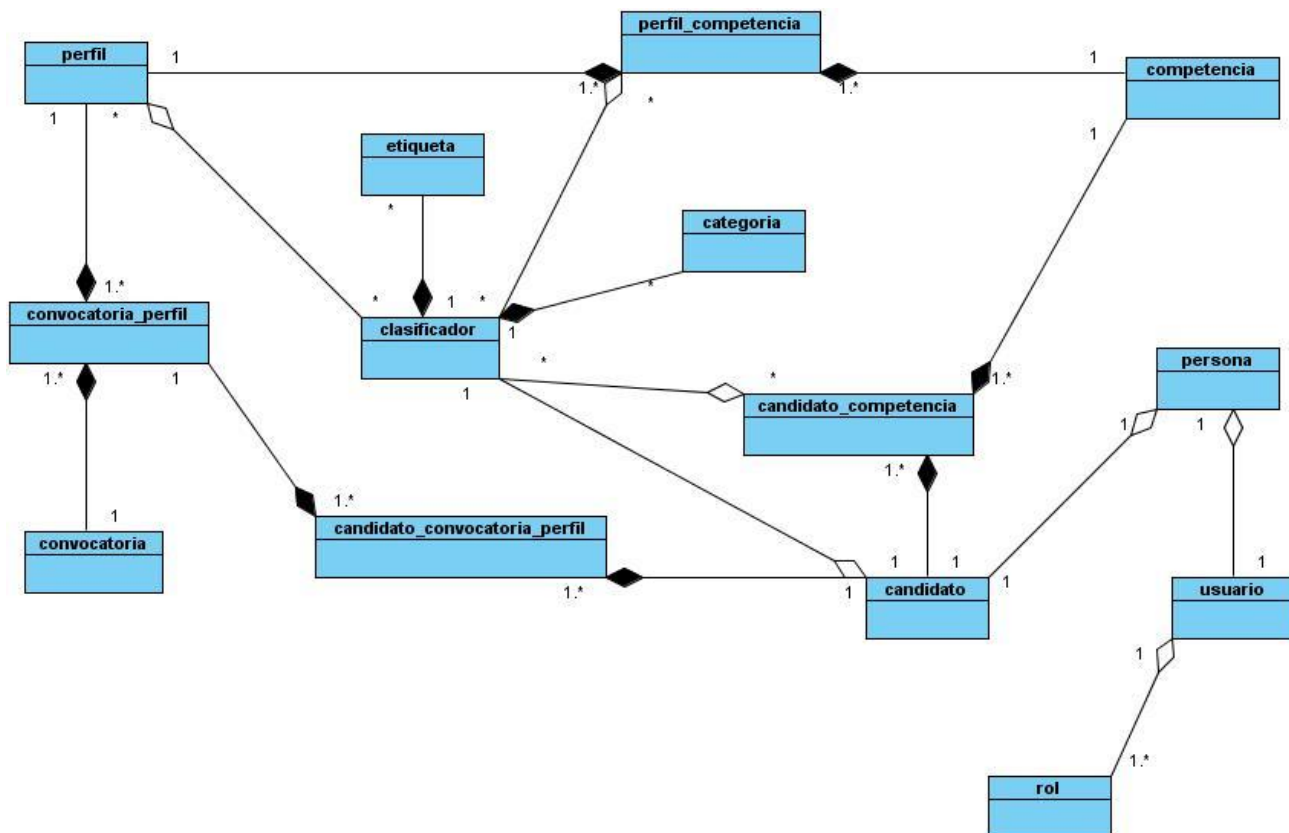


Figura 2.1 Modelo de objetos

2.2.4 Reglas a considerar

- ✓ Cada convocatoria es creada por los usuarios con el rol Gestor de Convocatoria.
- ✓ La gestión de usuario solo podrá ser realizada por el rol Administrador.
- ✓ Para que un candidato opte por un puesto tiene que haber una convocatoria creada.
- ✓ Cada convocatoria es lanzada cuando exista alguna plaza vacante dentro de la Universidad.

2.3 Construcción de una lista de funcionalidades

El modelo de objetos y los requerimientos existentes ofrecen una buena base para construir una lista de funcionalidades que resuma la funcionalidad del sistema a ser desarrollado. En dicha lista, el equipo de desarrolladores presenta cada una de las funcionalidades evaluadas por el cliente. Las funcionalidades son presentadas por cada área del dominio y éstas forman una Lista de Funcionalidades. Dicha lista es dividida en subconjuntos en base a la funcionalidad. Estas representan diferentes actividades con un área específica del dominio. La lista de funcionalidades es revisada por los usuarios y sponsors del sistema para su validación y aprobación. [7]

2.3.1 Requerimientos funcionales

Para determinar lo que debe cumplir la aplicación se realiza una lista de funcionalidades, las cuales son las capacidades o condiciones que el sistema debe cumplir, definiendo el comportamiento del software.

Gestionar persona:

Nro.	Funcionalidad	Prioridad
1	Insertar persona	A
2	Eliminar persona	A
3	Editar persona	A
4	Listar persona	A
5	Filtrar persona por ID	A
6	Filtrar persona por CI	A
7	Filtrar persona por nombre	A

Tabla 2.1 Primer bloque de funcionalidades.

Gestionar candidato:

Nro.	Funcionalidad	Prioridad
8	Insertar candidato	A

9	Eliminar candidato	A
10	Editar candidato	A
11	Listar candidato	A
12	Filtrar candidato por ID	A
13	Filtrar candidato por CI	A
14	Filtrar candidato asignado a perfil de convocatoria	A
15	Listar competencias del candidato	A
16	Asignar competencias al candidato	A
17	Evaluar competencias del candidato	A
18	Eliminar competencias del candidato	A
19	Agregar candidato a perfil de convocatoria	A
20	Eliminar candidato de perfil de convocatoria	A
21	Asignar candidato a puesto en convocatoria	A

Tabla 2.2 Segundo bloque de funcionalidades.

Gestionar convocatoria:

Nro.	Funcionalidad	Prioridad
22	Insertar convocatoria	A
23	Eliminar convocatoria	A
24	Editar convocatoria	A
25	Listar convocatoria	A
26	Filtrar convocatoria por ID	A
27	Filtrar convocatoria por nombre	A
28	Filtrar convocatoria por rango de fecha	A
29	Filtrar convocatoria por solucionada	A

Tabla 2.3 Tercer bloque de funcionalidades.

Gestionar perfil:

Nro.	Funcionalidad	Prioridad
30	Insertar perfil	A
31	Eliminar perfil	A
32	Editar perfil	A
33	Listar perfil	A
34	Filtrar perfil por ID	A
35	Asignar perfil a convocatoria	A
36	Eliminar perfil de convocatoria	A

Tabla 2.4 Cuarto bloque de funcionalidades.

Gestionar competencia:

Nro.	Funcionalidad	Prioridad
37	Insertar competencia	A
38	Eliminar competencia	A
39	Editar competencia	A
40	Listar competencia	A
41	Filtrar competencia por ID	A
42	Filtrar competencia por nombre	A
43	Filtrar competencia por perfil	A
44	Asignar competencia a perfil	A
45	Eliminar competencia de perfil	A

Tabla 2.5 Quinto bloque de funcionalidades.

Gestionar categoría:

Nro.	Funcionalidad	Prioridad
46	Insertar categoría	A
47	Eliminar categoría	A

48	Editar categoría	A
49	Listar categoría	A
50	Filtrar categoría por ID	A
51	Filtrar categoría por nombre	A

Tabla 2.6 Sexto bloque de funcionalidades.

Gestionar etiqueta:

Nro.	Funcionalidad	Prioridad
52	Insertar etiqueta	A
53	Eliminar etiqueta	A
54	Editar etiqueta	A
55	Listar etiqueta	A
56	Filtrar etiqueta por ID	A
57	Filtrar etiqueta por nombre	A

Tabla 2.7 Séptimo bloque de funcionalidades.

Gestionar clasificador:

Nro.	Funcionalidad	Prioridad
58	Insertar clasificador	A
59	Eliminar clasificador	A
60	Editar clasificador	A
61	Listar clasificador	A
62	Filtrar clasificador por etiqueta	A
63	Filtrar clasificador por categoría	A
64	Filtrar clasificador por rango	A

Tabla 2.8 Octavo bloque de funcionalidades.

Gestionar rol:

Nro.	Funcionalidad	Prioridad
65	Insertar rol	A
66	Eliminar rol	A
67	Editar rol	A
68	Listar rol	A
69	Filtrar rol por ID	A
70	Filtrar rol por nombre	A

Tabla 2.9 Noveno bloque de funcionalidades.

Gestionar usuario:

Nro.	Funcionalidad	Prioridad
71	Insertar usuario	A
72	Eliminar usuario	A
73	Editar usuario (contraseña y rol)	A
74	Autoedición de usuario (contraseña y email)	A
75	Listar usuario	A
76	Filtrar usuario por ID	A
77	Filtrar usuario por CI	A
78	Filtrar usuario por rol	A
79	Login de usuario	A
80	Logout de usuario	A

Tabla 2.10 Décimo bloque de funcionalidades.

Soluciones:

Nro.	Funcionalidad	Prioridad
81	Hacer reporte de las N primeras soluciones para una convocatoria	A
82	Imprimir reporte de las N primeras soluciones para una convocatoria	B

83	Filtrar soluciones dentro de un reporte	A
84	Guardar solución de convocatoria	A
85	Crear modelo de evaluación para una convocatoria por candidato	A
86	Imprimir modelo de evaluación del candidato	B

Tabla 2.11 Onceno bloque de funcionalidades.

2.3.2 Requerimientos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos los requisitos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requisitos funcionales, es decir una vez que se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. [8]

Requerimientos de apariencia o interfaz externa:

- ✓ El sistema debe tener una interfaz sencilla, legible y fácil de usar.
- ✓ La interfaz debe ser realizada respetando los parámetros de diseño.

Requerimientos de usabilidad:

- ✓ La interfaz de usuario debe ser lo más similar posible a otras aplicaciones que el cliente haya utilizado.

Requerimientos de rendimiento:

- ✓ La rapidez en el procesamiento de la información y en la determinación del candidato competente para una plaza es una de las características fundamentales del sistema propuesto. El tiempo para realizar acciones complejas que no dependan del acceso de las colecciones de datos no debe ser superior a 5 segundos.



Requerimientos de portabilidad:

- ✓ Para el desarrollo de la aplicación propuesta fue empleada la plataforma Windows, siendo posible su instalación en cualquiera de sus versiones desde 2000 o superior, así como en el Sistema Operativo Linux.

Requerimientos de seguridad:

- ✓ La seguridad de la información debe ser garantizada mediante el establecimiento de niveles de acceso y el estricto control de los mismos. Los usuarios tendrán definidos sus privilegios según rol que desempeñan, asegurándose de esa forma que las consultas de la información sean realizadas en correspondencia con el nivel de acceso.
- ✓ Los requerimientos de seguridad no se convertirán en un retardo para el empleo del sistema, garantizando el acceso rápido y operativo por los usuarios.
- ✓ Por razones de seguridad las contraseñas no deben viajar al servidor en texto plano, por lo que se utiliza un mecanismo de encriptación, empleando para ello la función hash MD5.

La función hash es un tipo de algoritmo que calcula un mensaje implícito o valor hash para cualquier mensaje que se le indique. El valor generado por el algoritmo no es importante, lo importante es que el resultado sea fijo, es decir, que sea el mismo cada vez que se utiliza una entrada dada, que sea pequeño y que el algoritmo sea rápido. La función MD5 calcula y devuelve el hash MD5 que consiste en una cadena de 32 números en formato hexadecimal. [12]

Requerimientos de hardware:

- ✓ Las máquinas que usarán el sistema deberán estar conectadas a la red de la Universidad.



- ✓ Los requisitos mínimos para una estación de trabajo deben ser de 512 MB de memoria RAM, 800 MHz, 2GB de disco duro disponible e impresora EpsonLX300 o superior.
- ✓ El servidor debe poseer como mínimo 1GB de memoria RAM, 1.4GHz y 10GB de disco duro disponible

Requerimientos de software:

- ✓ Los requerimientos mínimos para una estación de trabajo serán:
 - Navegador Mozilla Firefox 7.0, Google Chrome 25.0 o una de sus versiones superiores.
- ✓ Los requerimientos que debe cumplir la máquina servidor ser:
 - Servidor Web Apache 5.3.1 o superior.
 - Lenguaje de programación PHP 4.
 - Servidor de base de datos PostgreSQL 8.3.4.
 - Navegador Mozilla Firefox 7.0, Google Chrome 25.0 o una de sus versiones superiores.

2.4 Estudio de factibilidad

La técnica de Análisis de Puntos de Función es una de las más utilizadas para estimar el tamaño de un sistema sobre la base de sus requerimientos, permitiendo realizar la cuantificación en unidades que no dependen del lenguaje de programación, metodologías, plataformas y/o tecnologías aplicadas.

En cuanto a la estimación del esfuerzo, durante los últimos años se ha mantenido vigente la propuesta presentada por el Instituto de Ingeniería de Software (en inglés SEI, Software Engineering Institute) de un método nombrado COCOMO II, el cual se basa en el empleo de ecuaciones matemáticas que parten de ciertas métricas de tamaño estimado, entre las

que se encuentran el Análisis de Puntos de Función y las líneas de código fuente (en inglés SLOC, Source Line Of Code).

El presente epígrafe describe la propuesta del empleo combinado de ambos métodos (Análisis de Puntos de Función y COCOMO II) con el objetivo de lograr mayor precisión en la estimación.

El Análisis de Puntos de Función se aplica con el fin de determinar las sentencias de código del proyecto.

A continuación, basándose en la información obtenida, se aplica el método COCOMO II, mediante el cual se llega a la estimación exacta de las horas hombre necesarias, así como de la duración total del proyecto.

Una vez clasificadas cada una de las funcionalidades previstas, fueron obtenidos los resultados siguientes:

Nombre de la entrada externa	Cantidad de ficheros referenciados	Cantidad de elementos de datos	Complejidad (Simple, Medio o Complejo)
1	1	2	Simple
2	2	3	Simple
3	2	2	Simple
8	1	2	Simple
9	1	2	Simple
10	1	2	Simple
16	3	5	Medio
17	2	4	Simple
18	2	4	Simple
19	2	5	Medio
20	2	5	Medio
21	2	4	Simple

22	1	4	Simple
23	3	8	Medio
24	1	4	Simple
30	1	5	Simple
31	2	7	Medio
32	1	4	Simple
35	2	3	Simple
36	2	3	Simple
37	1	2	Simple
38	1	2	Simple
39	1	2	Simple
44	2	4	Simple
45	2	4	Simple
46	1	3	Simple
47	2	4	Simple
48	1	3	Simple
52	1	1	Simple
53	1	1	Simple
54	1	1	Simple
58	1	6	Simple
59	3	6	Medio
60	1	6	Simple
65	1	1	Simple
66	1	1	Simple
67	1	1	Simple
71	1	4	Simple
72	2	4	Simple
73	2	5	Medio
74	1	3	Simple

79	2	4	Simple
80	1	2	Simple
82	3	4	Simple
84	2	4	Simple
85	2	5	Medio
86	3	4	Simple

Tabla 2.12 Entradas externas.

Nombre de la salida externa	Cantidad de ficheros referenciados	Cantidad de elementos de datos	Complejidad (Simple, Medio o Complejo)
81	5	9	Complejo

Tabla 2.13 Salidas externas.

Nombre de la consulta externa	Cantidad de ficheros referenciados	Cantidad de elementos de datos	Complejidad (Simple, Medio o Complejo)
4	2	3	Simple
5	1	1	Simple
6	1	1	Simple
7	1	1	Simple
11	1	2	Simple
12	1	1	Simple
13	1	1	Simple
14	2	3	Simple
15	2	4	Simple
25	1	6	Simple
26	1	1	Simple
27	1	1	Simple
28	1	2	Simple

29	1	2	Simple
33	1	4	Simple
34	1	1	Simple
40	3	3	Simple
41	1	1	Simple
42	1	1	Simple
43	2	3	Simple
49	1	2	Simple
50	1	1	Simple
51	1	1	Simple
55	1	1	Simple
56	1	1	Simple
57	1	1	Simple
61	3	3	Simple
62	2	3	Simple
63	2	3	Simple
64	1	3	Simple
68	1	1	Simple
69	1	1	Simple
70	1	1	Simple
75	2	4	Simple
76	1	1	Simple
77	1	1	Simple
78	2	4	Simple
83	3	2	Simple

Tabla 2.14 Consultas externas.

Nombre del fichero lógico interno	Cantidad de ficheros referenciados	Cantidad de elementos de datos	Complejidad (Simple, Medio o Complejo)
Convocatoria	2	5	Simple
Perfil	3	5	Simple
Competencia	2	3	Simple
Clasificador	3	6	Simple
Etiqueta	1	2	Simple
Categoría	1	4	Simple
Persona	2	3	Simple
Candidato	3	3	Simple
Usuario	1	5	Simple
Rol	1	2	Simple

Tabla 2.15 Archivos lógicos internos.

Elementos	Baja	X Peso	Media	X Peso	Alta	X Peso	Subtotal de puntos de función
Entradas externas	39	3	8	4	0	6	149
Salidas externas	0	4	0	5	1	7	7
Consultas externas	38	3	0	3	0	6	114
Ficheros lógicos internos	10	7	0	10	0	15	70
						Total	340

Tabla 2.16 Puntos de función.



$$\text{UFP} = 340$$

$$\text{PM}_{\text{nominal}} = A \times (\text{Size})^B$$

A: se toma el valor por defecto del modelo, ajustado en 2.94.

Size: se calcula como el producto de los puntos de función sin ajustar por un factor de conversión que depende del lenguaje a utilizar en el desarrollo del sistema. En este caso se utiliza PHP (factor de conversión medio para lenguajes orientados a objetos = 30 SLOC/UFP).

$$\text{Size} = \text{UFP} \times \text{FC}$$

$$\text{Size} = 340 \times 30 = 10200 \text{ SLOC} = 10.2 \text{ KLOC}$$

B: se calcula ponderando las variables mediante la siguiente ecuación:

$$B = 0.91 + 0.01 \times \sum (W_i)$$

Donde las W_i se muestran en la siguiente tabla:

Variable	Ponderación	Valor
PREC	Alto	2.40
FLEX	Nominal	3.04
RESL	Alto	2.83
TEAM	Alto	3.29
PMAT	Nominal	4.68

Tabla 2.17 Variables escalares.

Factores de escala:

$$\text{SF} = \sum (W_i) = \text{PREC} + \text{FLEX} + \text{RESL} + \text{TEAM} + \text{PMAT}$$

$$\text{SF} = \sum (W_i) = 2.4 + 3.04 + 2.83 + 3.29 + 4.68 = 16.24$$

$$B = 0.91 + 0.01 \times 16.24$$



$$B = 1.0724 \approx 1.07$$

Luego el esfuerzo nominal es:

$$PM_{\text{nominal}} = A \times (\text{Size})^B$$

$$PM_{\text{nominal}} = 2.94 \times (10.2 \text{ KLOC})^{1.07} \approx 35,28 \text{ Meses} - \text{Hombre}$$

Para completar la estimación se ajusta el esfuerzo nominal según las particulares del proyecto. Para esto se realiza la siguiente ecuación:

$$PM_{\text{ajustado}} = PM_{\text{nominal}} \times \prod (ME_i)$$

Donde los $\prod (ME_i)$ (Multiplicadores de esfuerzo) varían en función del modelo de estimación usado (Diseño Preliminar o Post Arquitectura). Para este proyecto se emplea el modelo de Diseño Preliminar. Luego se cuantifican los multiplicadores de esfuerzo se la siguiente manera:

Multiplicador	Ponderación	Valor
PERS	Alto	0.83
RCPX	Nominal	1
RUSE	Nominal	1
PDIF	Bajo	0.87
PREX	Alto	0.87
SCED	Nominal	1
FCIL	Muy Alto	0.73

Tabla 2.18 Multiplicadores de esfuerzo.

Multiplicadores de esfuerzo:

$$E_M = \prod (ME_i) = PESR \times RCPX \times RUSE \times PDIF \times PREX \times SCED \times FCIL$$

$$E_M = \prod (ME_i) = 0.83 \times 1 \times 1 \times 0.87 \times 0.87 \times 1 \times 0.73$$

$$E_M = 0.45860571 \approx 0.46$$

Teniendo estos valores el ajuste del esfuerzo resulta:

$$PM_{\text{ajustado}} = 35.28 \times 0.46 = 16.2 \text{ Meses} - \text{Hombre} \approx 16 \text{ Meses} - \text{Hombre}$$

El resultado arrojado se interpreta como el tiempo requerido para que una persona desarrolle la aplicación.

Costo:

Se asume como salario promedio mensual \$450.00.

CH: Cantidad de hombres = 3.

Tiempo: Tiempo total de desarrollo del software.

$$\text{Tiempo} = PM_{\text{ajustado}} \div CH = 16 \div 3 = 5,3 \approx 5 \text{ Meses}$$

$$\text{Costo} = \text{SalarioPromedio} \times CH \times \text{Tiempo} = 450 \times 3 \times 5 = \$6750$$

Cálculo	Valor
Esfuerzo nominal	35.28 Meses - Hombre
Esfuerzo ajustado	16 Meses - Hombre
Cantidad de hombres	3
Salario promedio	\$450
Tiempo	5 Meses
Costo	\$6750

Tabla 2.19 Resumen de resultados.

Interpretando los resultados obtenidos se puede concluir que con 3 hombres trabajando en el desarrollo de la aplicación se tarda 5 meses y su costo total se estima en \$6750.

2.4.1 Beneficios tangibles e intangibles

El principal de los beneficios tangibles que reporta la elaboración del software propuesto está relacionado con el ahorro en el empleo de los recursos humanos e insumos de oficina durante el proceso de selección de candidatos, permitiendo la gestión rápida y eficiente de



la información asociada al mismo. Esto se convierte en un beneficio económico de consideración, pues además de la disminución del gasto de medios materiales, los trabajadores emplearán un tiempo significativamente menor para el cumplimiento de sus funciones.

La seguridad y confiabilidad que se alcanza en la gestión de la información y en los resultados del proceso, las cuales representan una importante ayuda a la toma de decisiones por parte de los directivos, constituyen los beneficios intangibles de la aplicación.

2.4.2 Análisis de costo y beneficios

En el epígrafe anterior quedaron plasmados los beneficios que aporta el sistema que se propone como resultado del presente trabajo, los que determinan las ventajas económicas y en el procesamiento de información para la toma de decisiones. A esto se une que su desarrollo no requiere de un considerable gasto de recursos y que las tecnologías utilizadas no requieren el pago de licencias.

De lo anterior se deriva que los beneficios que traerán su despliegue y aplicación son superiores en comparación con su costo, por lo que es factible la realización de este proyecto.

2.5 Planeación por funcionalidad

En esta etapa se incluye un plan de alto nivel, en la cual se deja plasmada la lista de funcionalidades con sus prioridades y dependencias. Además son asignadas las clases identificadas a cada programador. (Anexo 2)

2.6 Diseño por funcionalidad

De la lista de funcionalidades se seleccionan un conjunto de ellas.

El diseño y construcción de la funcionalidad es un proceso iterativo durante el cual las funcionalidades seleccionadas son producidas. Una iteración puede llevar desde unos pocos días a un máximo de dos semanas. Este proceso iterativo incluye tareas como



inspección del diseño, codificación, prueba unitaria, integración e inspección del código. Luego que la iteración llega a su fin se realiza una construcción de la funcionalidad en el cual esta es integrada. [7]

2.6.1 Principios de diseño

Para la construcción del software se utilizó un lenguaje técnico tomado de la experiencia de los usuarios del mismo. Además se tuvieron en cuenta los siguientes aspectos:

- ✓ Los menús poseen pocas opciones.
- ✓ Las operaciones comparables se activan de la misma forma.
- ✓ La introducción de datos en los campos de un formulario se realiza de forma sencilla.
- ✓ El sistema no provoca sorpresa a los usuarios.
- ✓ La interfaz suministra retroalimentación significativa.
- ✓ La interfaz posee diferentes tipos de características para la interacción de los disímiles usuarios.
- ✓ Para la introducción de los datos no se requiere teclear mucho.

2.6.2 Modelo conceptual de base de datos

El modelo conceptual por su elevado nivel de abstracción y riqueza semántica proyecta una interfaz útil entre el informático y los usuarios finales en las primeras etapas del proceso de diseño de bases de datos, pues detalla en mayor medida el significado de los datos y las relaciones entre sí.

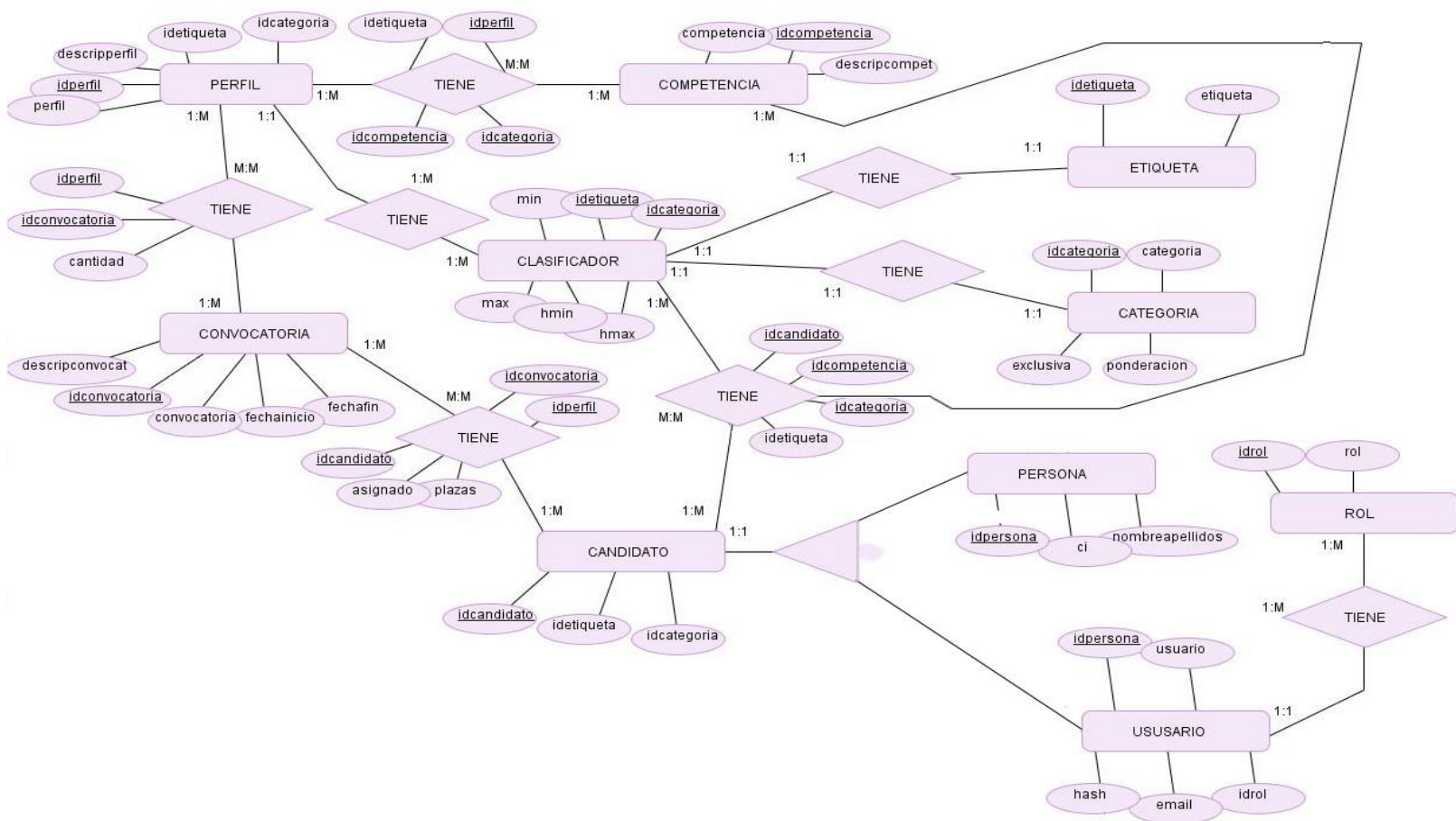


Figura 2.2 Modelo conceptual de la base de datos.

2.6.3 Modelo lógico de la base de datos

El diagrama del modelo lógico de datos o diagrama de clases persistentes, muestra las clases capaces de mantener su valor en el espacio y en el tiempo.

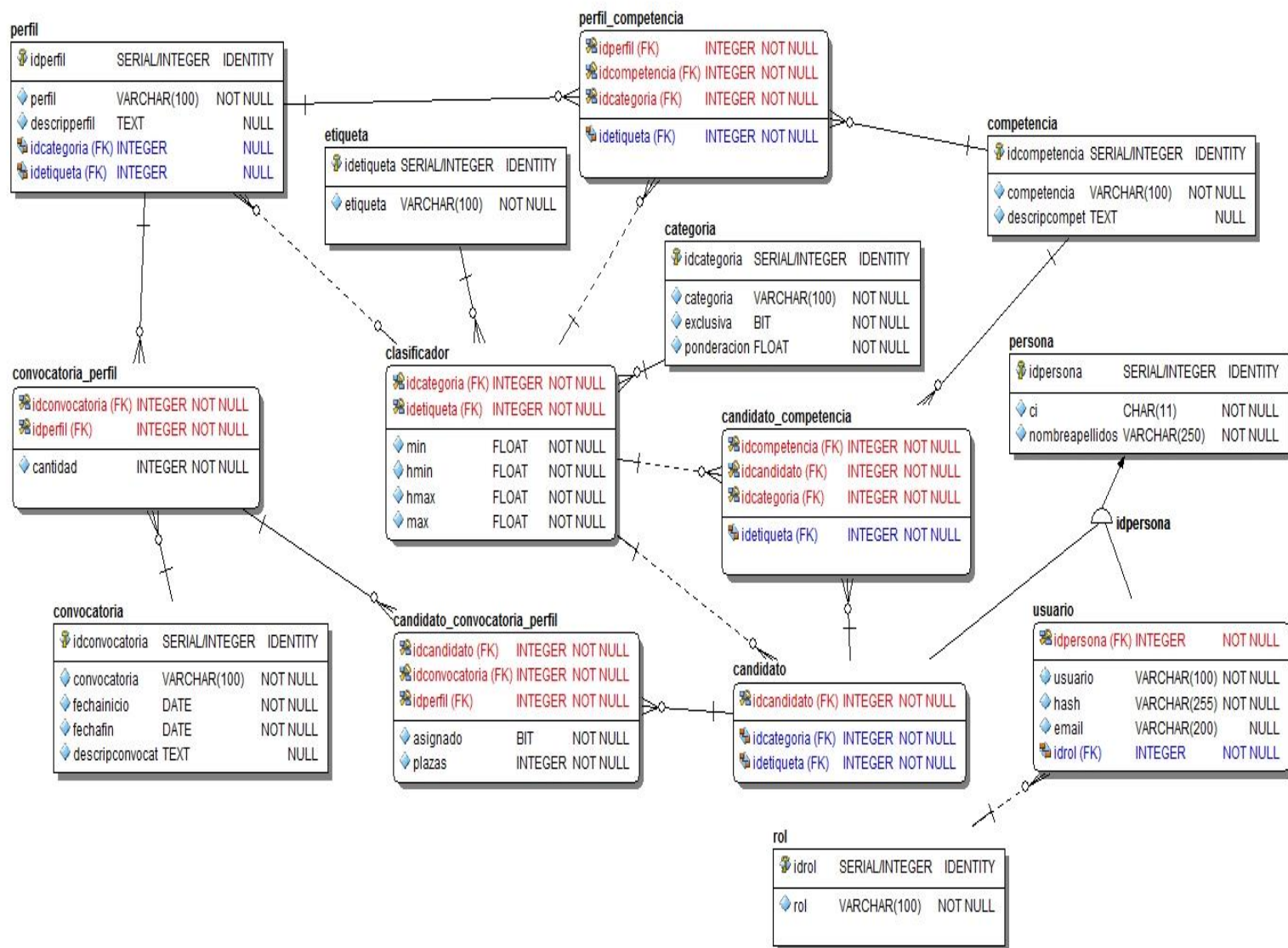


Figura 2.3 Modelo lógico de la base de datos.

2.6.4 Modelo físico de la base de datos

El modelo físico de datos incluye todos los aspectos de diseño de un modelo de base de datos, que se pueden modificar sin cambiar los componentes de la aplicación.

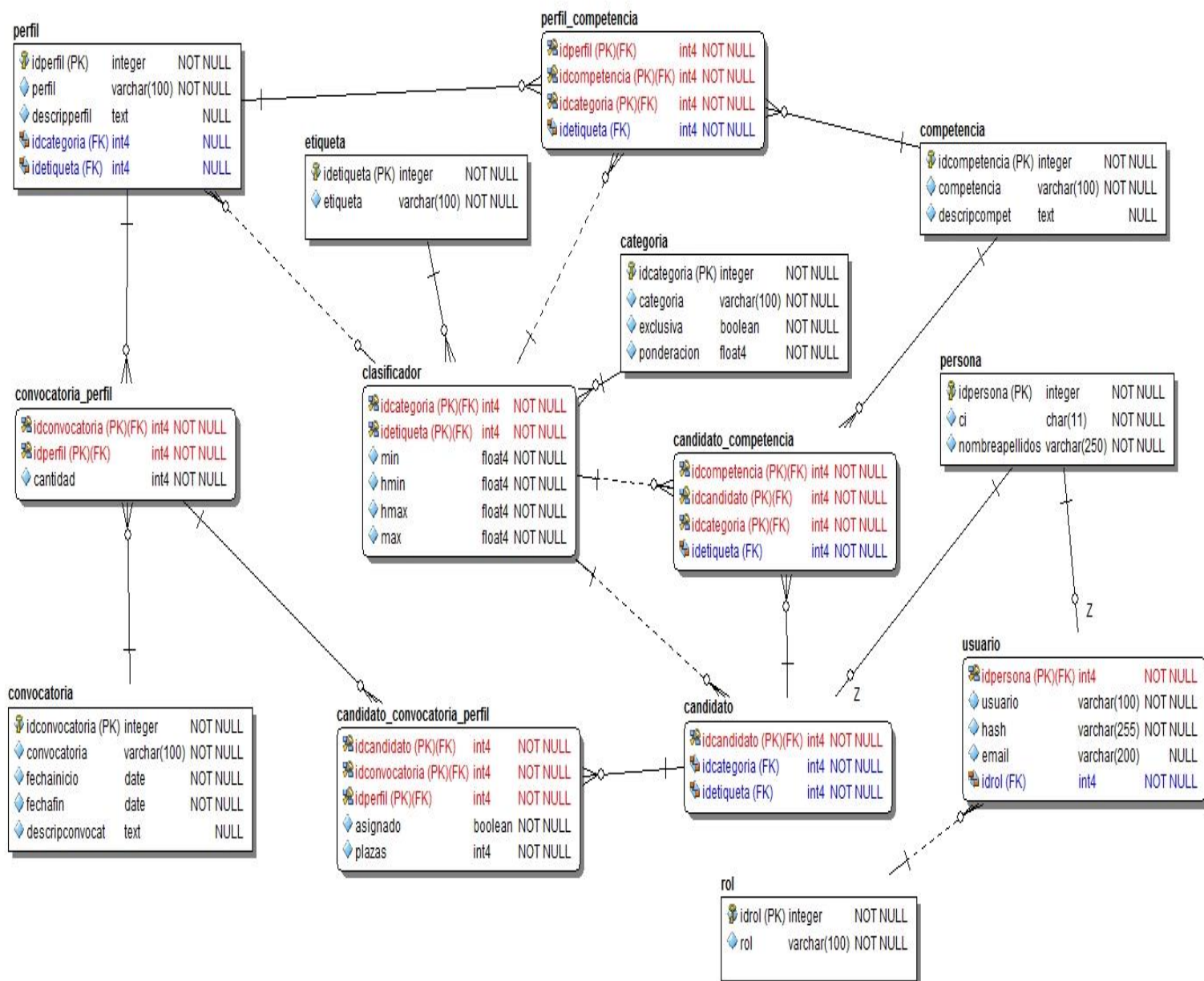


Figura 2.4 Modelo físico de la base de datos.



2.7 Conclusiones

En el capítulo fueron creados los artefactos de las fases que plantea la metodología de desarrollo FDD utilizando para ella la notación UML. Se describió una explicación detallada del flujo de procesos llevados a cabo en el departamento de Recursos Humanos de la Universidad de Cienfuegos. También quedaron plasmados los requerimientos del sistema, así como la planeación y el diseño por funcionalidades. A su vez se realizó el estudio de factibilidad del software el cual proporcionó como resultado un costo estimado de \$6750 con un tiempo de desarrollo de cinco meses para tres personas. Después de analizar todos estos factores y teniendo en cuenta los beneficios del establecimiento de dicho software se concluyó que el desarrollo del proyecto es factible.

Capítulo 3: Construcción del sistema

3.1 Introducción

El objetivo de este capítulo es describir el diseño del sistema a través de los diagramas de clases. Se recoge las especificaciones relacionadas con la seguridad del sistema y el tratamiento de errores. Finalmente se realiza la validación de la solución propuesta, mediante métodos estadísticos.

3.2 Descripción de la arquitectura propuesta

CodeIgniter está basado en el patrón de desarrollo Modelo-Vista-Controlador. Este patrón es una aproximación al software que separa la lógica de la aplicación de la presentación.

El Modelo representa la estructura de datos. Típicamente sus clases de modelo contendrán funciones que lo ayudarán a recuperar, insertar y actualizar información en su base de datos. Se implementó una clase del modelo para cada tabla de la base de datos. [13]

La Vista es la información que es presentada al usuario. La Vista normalmente será una página web, pero en CodeIgniter, una vista también puede ser un fragmento de una página como un encabezado o un pie de página. Las vistas de la aplicación poseen elementos comunes que fueron definidos en la plantilla además de las vistas diseñadas para cada una de las funcionalidades. [13]

El Controlador sirve como un intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para procesar la petición HTTP y generar una página web. Se definieron doce clases controladoras, de manera tal que cada una gestiona la información relacionada con una clase entidad, además una para la administración y la principal que es la que se configura para ser cargada por la vista principal del sistema.[13]

Teniendo en cuenta que la aplicación se implementó utilizando como base CodeIgniter desde el punto de vista técnico y arquitectónico posee las siguientes características:

- ✓ **Bajo Acoplamiento:** Acoplamiento es el grado que los componentes de un sistema dependen entre ellos. Mientras menos componentes dependan de otro, más reusable y flexible el sistema se vuelva. [13]
- ✓ **Singularidad del Componente:** Singularidad es el grado que más componentes tienen un propósito en el que enfocarse más estrecho. Cada clase y sus funciones son altamente autónomas para permitir máxima utilidad. [13]

De manera general el sistema es poco acoplado con gran singularidad de componente. Posee simplicidad, flexibilidad y buen rendimiento.

3.3 Diagrama de clases

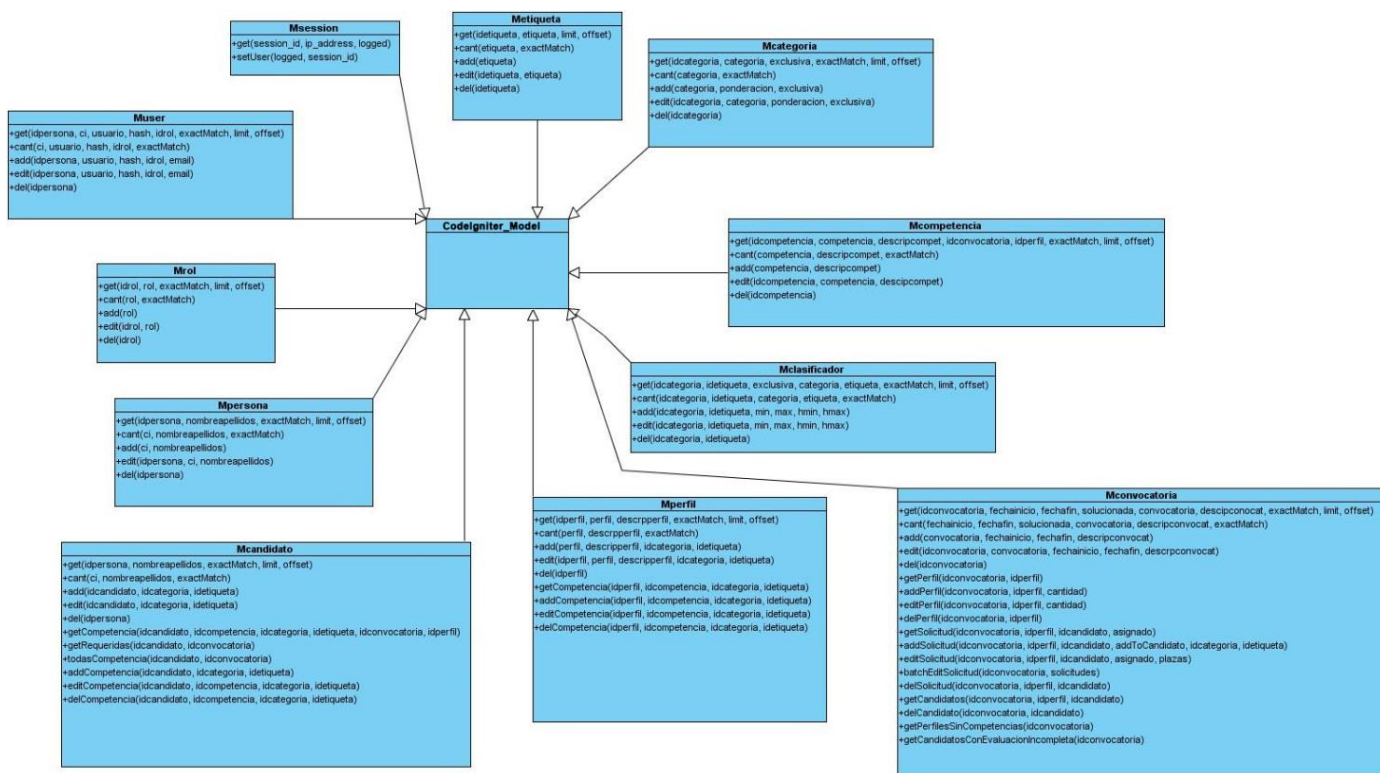


Figura 3.1 Diagrama de clases del modelo.

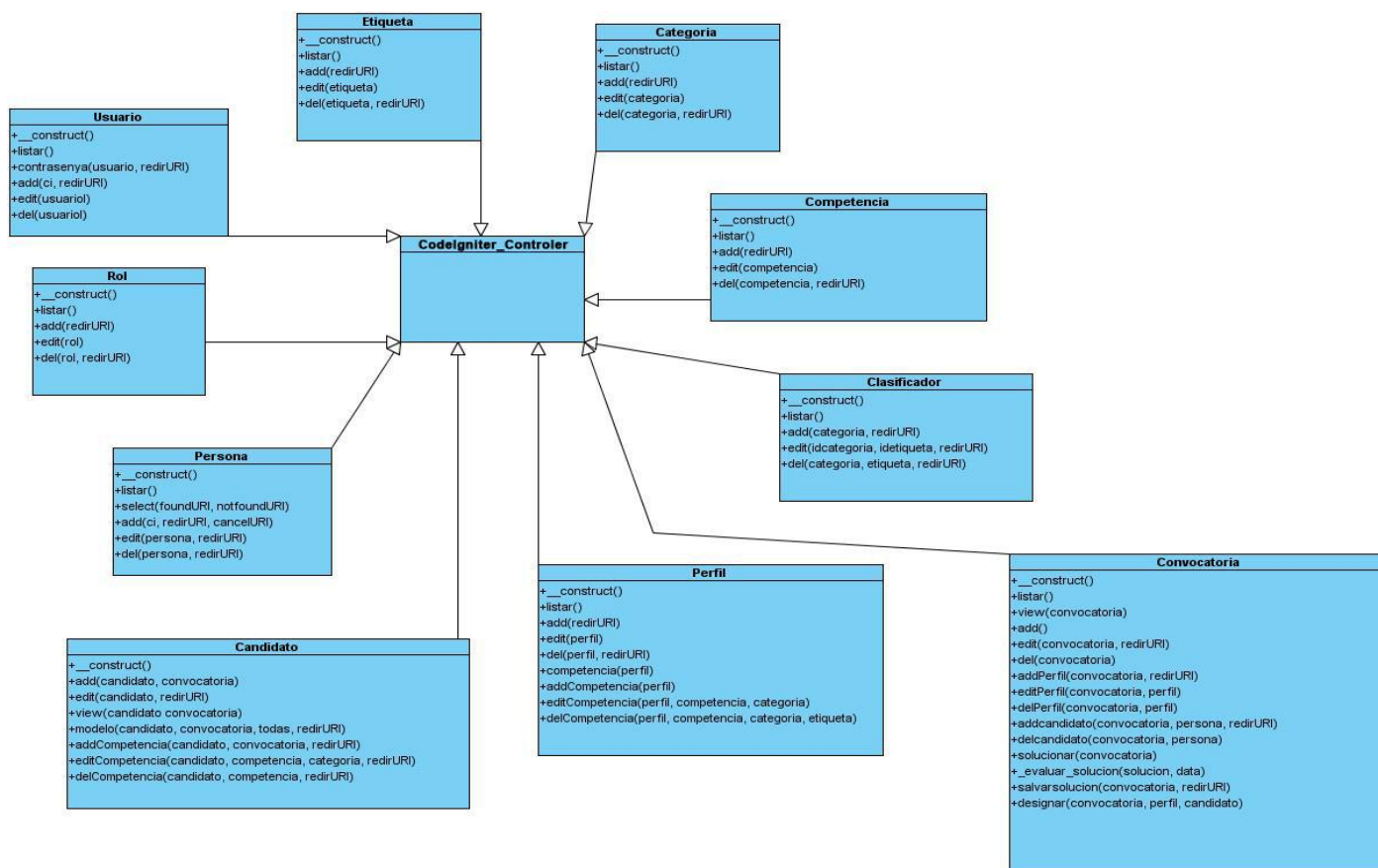


Figura 3.2 Diagrama de clases del controlador.

3.4 Estándar de implementación

Para el apropiado mantenimiento del código del sistema, así como para un mejor entendimiento, es necesario establecer un estándar de codificación a emplear. En este caso podemos decir que los inicios ({} y cierres (}) de ámbito están alineados al mismo nivel de la declaración a la que pertenece. Las variables, nombres de funciones, nombres de clases y los objetos que fueron declarados en las páginas son cortos, claros y describen claramente su propósito. Los objetos se nombran según el valor de su contenido. Los signos lógicos y de operación se separan por un espacio antes y después. Se utilizan comentarios para detallar determinadas líneas de código. Se emplean nombres similares entre clases, para así establecer notoriamente la relación entre la clase del



modelo, la controladora que maneja sus datos y la tabla de la base de datos de donde obtiene la información.

3.5 Seguridad del sistema

- ✓ El mecanismo de seguridad y protección de la aplicación se basa en el empleo del nombre de usuario y la contraseña para acceder a las funcionalidades a las que tienen acceso.
- ✓ Cada usuario tiene definido a que módulo e información puede acceder, teniendo en cuenta la política de seguridad del departamento y de la Universidad.
- ✓ Los usuarios una vez registrados en su módulo no podrán acceder haciendo uso de la URL a otra página a la cual no tiene acceso.
- ✓ Los usuarios una vez que se desconectan de su módulo no podrán regresar y realizar acciones dentro del mismo sin antes autenticarse.
- ✓ Se utilizaron mecanismos de encriptación de los datos usando para ello la función hash MD5.

3.6 Tratamiento de errores

Es muy común que los usuarios de sistemas informáticos cometan errores que deriven en una afectación en el adecuado funcionamiento de los mismos. Para contrarrestar dicha situación es necesario siempre concebir un eficaz tratamiento para situaciones excepcionales que puedan suceder. Ejemplo de esto podría ser la inserción de datos incorrectos o en un formato no admitido.

El sistema se diseñó e implementó de forma que las posibilidades de insertar información errónea sean casi nulas. Como en algunos casos es necesario teclear datos y seleccionar elementos, se mantiene un nivel elevado de validación. Además para ayudar al usuario se muestran comentarios sobre el formato de la información a introducir en las áreas de textos de las pantallas. Adicionalmente los mensajes de error que se muestra poseen un lenguaje fácil de comprender.

3.7 Diagrama de despliegue

El diagrama de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Es una colección de nodos y arcos; donde cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar. [8]

El siguiente diagrama muestra la configuración hardware del sistema y los nodos físicos que lo componen. El sistema estará estructurado según la arquitectura cliente - servidor. En el lado del servidor estarán en funcionamiento, el servidor de bases de datos PostgreSQL y el servidor Web Apache. Esta se comunicará con el cliente de la Intranet a través del protocolo HTTP. El cliente podrá visualizar la aplicación con el Mozilla Firefox 7.0 o superior. Además se cuenta con una impresora conectada a la máquina cliente que le brindará la posibilidad al usuario de imprimir la información que desee.

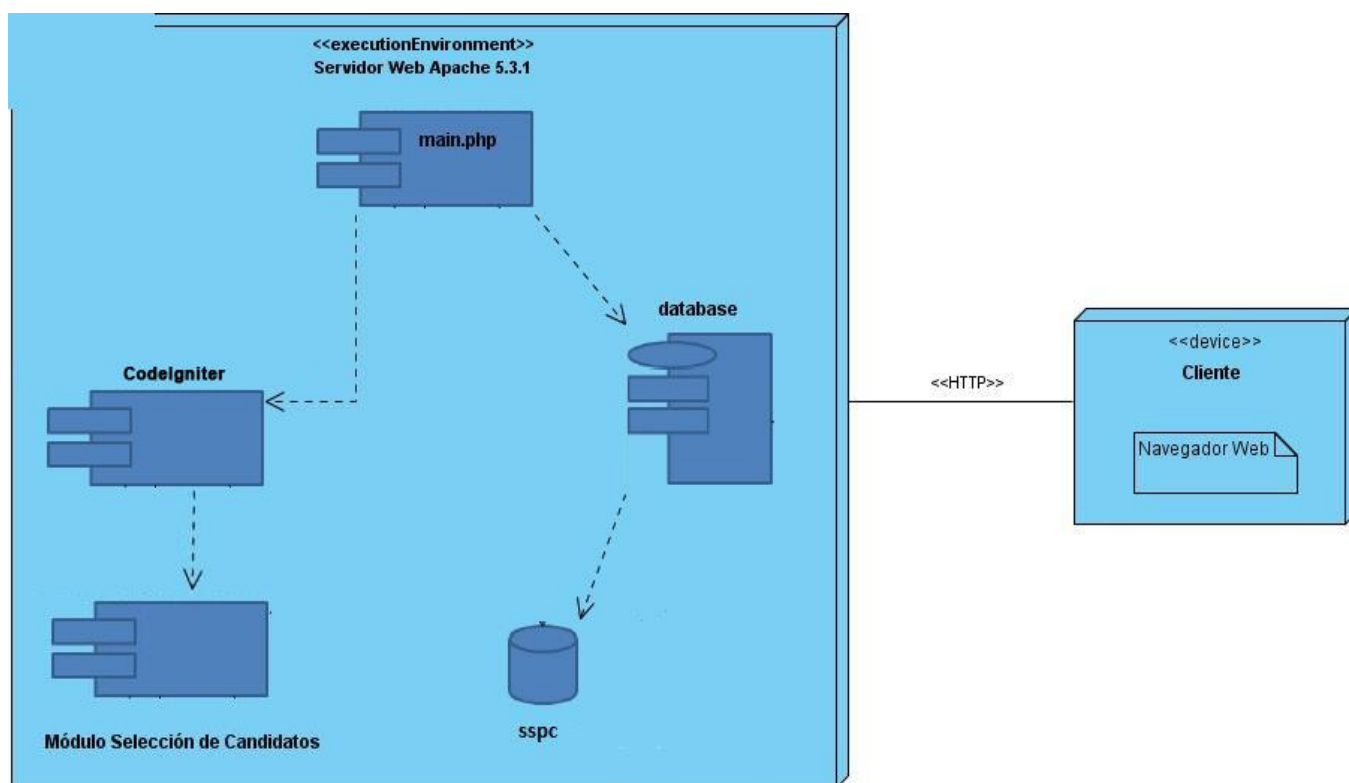


Figura 3.3 Diagrama de despliegue del sistema.

3.8 Validación de la solución propuesta

En esta fase se realizaron mediciones de la variable de tiempo promedio de duración del proceso de selección de candidatos a una plaza una vez evaluados en las disímiles competencias de la misma, antes y después de la configuración del sistema en la red de la universidad

3.8.1 Resultados estadísticos de las mediciones realizadas

Se efectuaron mediciones del tiempo que demoraba el proceso antes y después de la configuración del sistema. Se tomó una muestra diaria, en el período comprendido por un mes. Los datos recopilados se procesaron en el software de procesamiento estadístico SPSS.

Tiempo (minutos) de realización del proceso de la variable en análisis, antes y después de la utilización del sistema:

	Antes	Después
Muestra	60	0.02
	75	0.03
	70	0.03
	90	0.05
	85	0.04
	78	0.03
	70	0.03
	65	0.02
	88	0.05
	55	0.01
	75	0.03
	80	0.03
	67	0.02
	63	0.02

	65	0.02
	70	0.03
	63	0.02
	58	0.01
	65	0.02
	75	0.03
	55	0.02
	80	0.04
	67	0.02
	87	0.04
	70	0.03
	65	0.02
	60	0.02
	75	0.03
	73	0.03
	68	0.02
Media	70.6	0.027

Tabla 3.1 Tiempos antes y después de la realización del proceso.

Proceso de selección de candidatos a una plaza una vez evaluados en las disímiles competencias de la misma

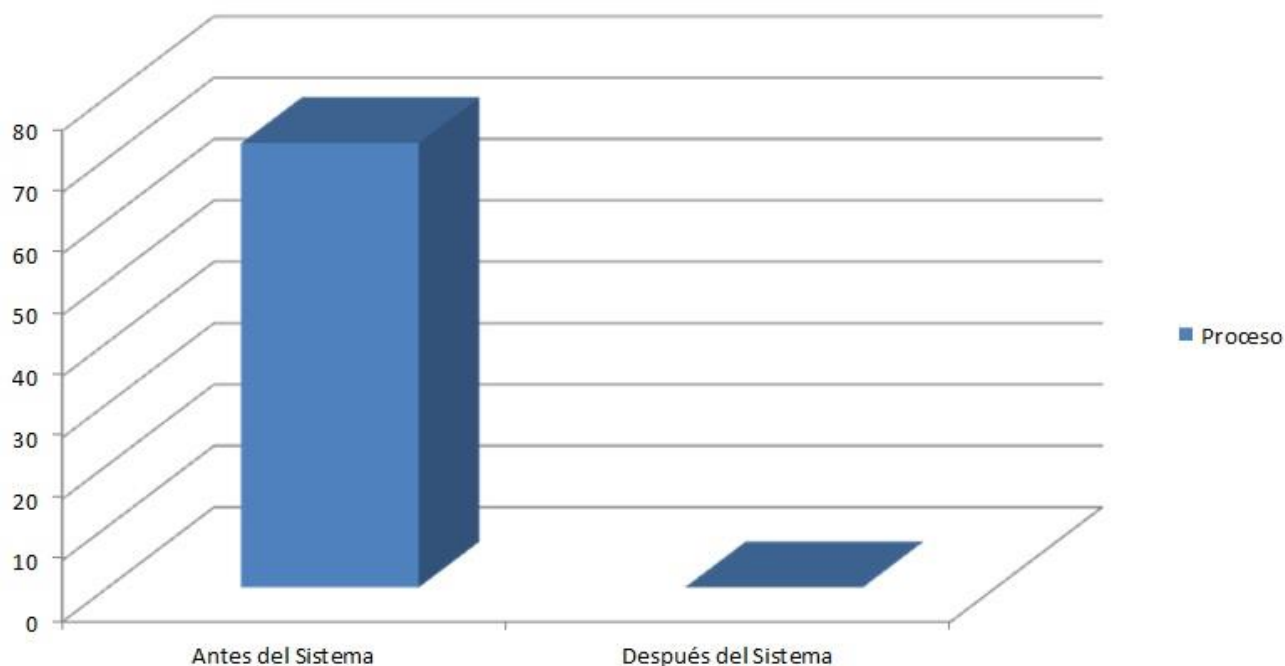


Figura 3.4 Comparación del tiempo promedio antes y después de instalado el sistema.

Con los datos anteriormente calculados se puede observar la significativa diferencia entre las medias de los valores de antes y después de la variable en cuestión, pero de igual forma es necesario comprobarlo estadísticamente para tener una fundamentación aún más precisa.

A continuación se muestra el estudio realizado utilizando la prueba de Kolmogorov-Smirnov para comprobar que los valores siguen una distribución de contraste normal.

Esta prueba realiza un contraste entre la hipótesis nula y la hipótesis alternativa planteando que la variable sigue una distribución normal y que la variable no sigue una distribución normal correspondientemente. Se tomó un nivel de significación de 5%, si este es menor que la significación asintótica, entonces se rechaza la hipótesis nula, de lo contrario se acepta.

La prueba de Kolmogorov-Smirnov arrojó el siguiente reporte:

		Antes	Después
N		30	30
Parámetros normales ^{a,b}	Media	70,5667	,0267
	Desviación típica	9,49658	,01028
Diferencias más extremas	Absoluta	,124	,208
	Positiva	,124	,208
	Negativa	-,069	-,160
Z de Kolmogorov-Smirnov		,678	1,141
Sig. asintót. (bilateral)		,747	,148

a. La distribución de contraste es la Normal.
b. Se han calculado a partir de los datos.

Figura 3.5 Prueba de Kolmogorov-Smirnov.

Después de analizar los resultados se concluye que se acepta la hipótesis nula, demostrando que la variable sigue una distribución normal.

A partir del este resultado se pasa a realizar la Prueba T, con el objetivo de demostrar que existe una diferencia significativa de un antes con respecto a un después.

Dicha prueba para muestras relacionadas plantea como hipótesis nula que la media de la variable X es igual a la media de la variable Y, considerando que no hay diferencias significativas entre ellas. La Prueba T para muestras relacionadas plantea como hipótesis alternativa que la media de X es diferente a la media Y, existiendo diferencias significativas entre ambas variables. Para este caso se utilizó un nivel de significación de 5%, y se obtuvo el siguiente reporte de datos:

Puede concluirse que se rechaza la hipótesis nula, y se acepta la hipótesis alternativa, demostrando que hay diferencias significativas entre ambas medias.

	Diferencias relacionadas					t	gl	Sig. (bilateral)
	Media	Desviación típ.	Error típ. de la media	95% Intervalo de confianza para la diferencia				
				Inferior	Superior			
Par1 Antes - Después	70,54000	9,48679	1,73204	66,99757	74,08243	40,726	29	,000

Figura 3.6 Prueba de T.

Por lo analizado y demostrado anteriormente se puede afirmar que el sistema realiza el proceso de selección de candidatos a una plaza una vez evaluados en las disímiles competencias de la misma en menos tiempo que antes de ser informatizado.

3.9 Conclusiones

En el presente capítulo se trata en primer lugar la descripción de la arquitectura propuesta. Además se muestran los diagramas de clases del modelo y del controlador, el estándar de implementación, los estándares de seguridad del sistema, el tratamiento de errores, así como la elaboración del diagrama de despliegue.

Por último, se realizó la validación de la solución propuesta mediante el tratamiento estadístico de las mediciones del tiempo de duración del proceso antes y después de la implementación del sistema, aplicando el método Kolmogorov – Smirnov y la Prueba T, con el empleo del SPSS. Con esto quedó demostrada la mejora del proceso de selección de candidatos en el Departamento de Recursos Humanos de la Universidad de Cienfuegos, a partir de la disminución significativa del tiempo promedio.



Conclusiones

El desarrollo de la investigación que se resume en los diferentes capítulos del presente trabajo de diploma, permite arribar a las siguientes conclusiones generales:

- ✓ Existen investigaciones que proporcionan el basamento matemático para el desarrollo de software que permitan agilizar el proceso de asignación de personal, determinándose que el trabajo nombrado “Asignación de recursos humanos mediante el uso de algoritmos genéticos” se ajusta perfectamente a las necesidades de la presente investigación.
- ✓ El software diseñado como resultado de este trabajo de diploma cumple los requisitos propuestos por el Departamento de Recursos Humanos al automatizar la selección de candidatos a una plaza una vez que han sido evaluadas sus competencias.
- ✓ La solución informática propuesta permite reducir significativamente el tiempo promedio del proceso.



Recomendaciones

Una vez cumplidos los objetivos trazados para el desarrollo de este trabajo, se recomienda:

- ✓ Continuar estudiando el proceso de selección de personal, con el objetivo de ampliar las funcionalidades de la aplicación.
- ✓ Desplegar la aplicación por todos los departamentos de Recursos Humanos de las universidades del país.



Referencia Bibliográfica

- [1] Departamento de Recursos Humanos, «Levantamiento de Procesos de Recursos Humanos». .
- [2] «Real Academia Española», *Diccionario de la Real Academia Española*. .
- [3] J. A. Alvaro Rodríguez, J. G. García Rodríguez, y y otros, *Lógica Difusa*. .
- [4] «Algoritmos Genéticos», *EcuRed*. .
- [5] L. M. Martínez Hernández, «Asignación de Recursos Humanos Mediante la Aplicación de Algoritmos Genéticos», Universidad de la Sabana, Colombia, 2004.
- [6] «Metodologías tradicionales y metodologías ágiles». [En línea]. Disponible en: <http://masteringenieriasoft.blogspot.com/2012/04/metodologias-de-desarrollo-de-software.html>.
- [7] «FDD». [En línea]. Disponible en: http://ingenieriadesoftware.mex.tl/61162_FDD.html.
- [8] G. B. I. Jacobson y J. Rumbaugh, *El Proceso Unificado de Desarrollo de Software*. Addison Wesley, 1999.
- [9] «Visual Paradigm for UML». [En línea]. Disponible en: <http://www.freedownloadmanager.org/es/>.
- [10] P. Craig Larman, *Una introducción al análisis y diseño orientado a objetos y al proceso unificado*. 2000.
- [11] «Sistemas de Información Gerencial». [En línea]. Disponible en: http://www.geocities.ws/jjaimes6603/fase3/t4_diferencias.html.
- [12] W y L thomson, *Desarrollo web con PHP y MySQL*. 2012.
- [13] R. Ellis, *Guia Definitiva de CodeIgniter*. .
- [14] «Lenguaje de Marcado de Hipertexto», *Ecured*. .



- [15] «Guía Breve de CCS». [En línea]. Disponible en:
<http://www.w3c.es/Divulgacion/GuiasBreves/hojasEstilo>.
- [16] «Qué es Javascript». [En línea]. Disponible en:
<http://www.desarrolloweb.com/articulos/25.php>.
- [17] «Netbeans 6.8 liberado», 2010. [En línea]. Disponible en:
<http://blogultura.com/java/netbeans-6-8-liberado>.
- [18] «GIMP 2.8». [En línea]. Disponible en: <http://www.gimp.org/>.



Bibliografía Consultada

- [1] Taymi Ceruto Cordovés, «Algoritmo heurísticos aplicados a la toma de decisiones en un proceso de selección de personal».
- [2] A. L. Infante Abreu, «ALGORITMOS DE TRAYECTORIA MULTIOBJETIVO APLICADOS AL PROBLEMA DE ASIGNACIÓN DE PERSONAL», CUJAE, La Habana, 2012.
- [3] «Algoritmos Genéticos», EcuRed. .
- [4] K. M. Cobas Friman, «Algoritmos genéticos aplicados a problemas de planificación», UCI, Manzanillo, 2012.
- [5] L. M. Martínez Hernández, «Asignación de Recursos Humanos Mediante la Aplicación de Algoritmos Genéticos», Universidad de la Sabana, Colombia, 2004.
- [6] W y L thomson, Desarrollo web con PHP y MySQL. 2012.
- [7] G. B. I. Jacobson y J. Rumbaugh, El Proceso Unificado de Desarrollo de Software. Addison Wesley, 1999.
- [8] M. Peralta, Estimación del esfuerzo basada en casos de uso. .
- [9] «Fundamentos de JQuery». [En línea]. Disponible en: <http://librojquery.com/>.
- [10] «GIMP 2.8». [En línea]. Disponible en: <http://www.gimp.org/>.
- [11] «Guía Breve de CCS». [En línea]. Disponible en: <http://www.w3c.es/Divulgacion/GuiasBreves/hojasEstilo>.
- [12] R. Ellis, Guia Definitiva de CodeIgniter. .
- [13] «Hum&Software - Tour del Sistema Hum&Select». [En línea]. Disponible en: www.humsoftware.com.



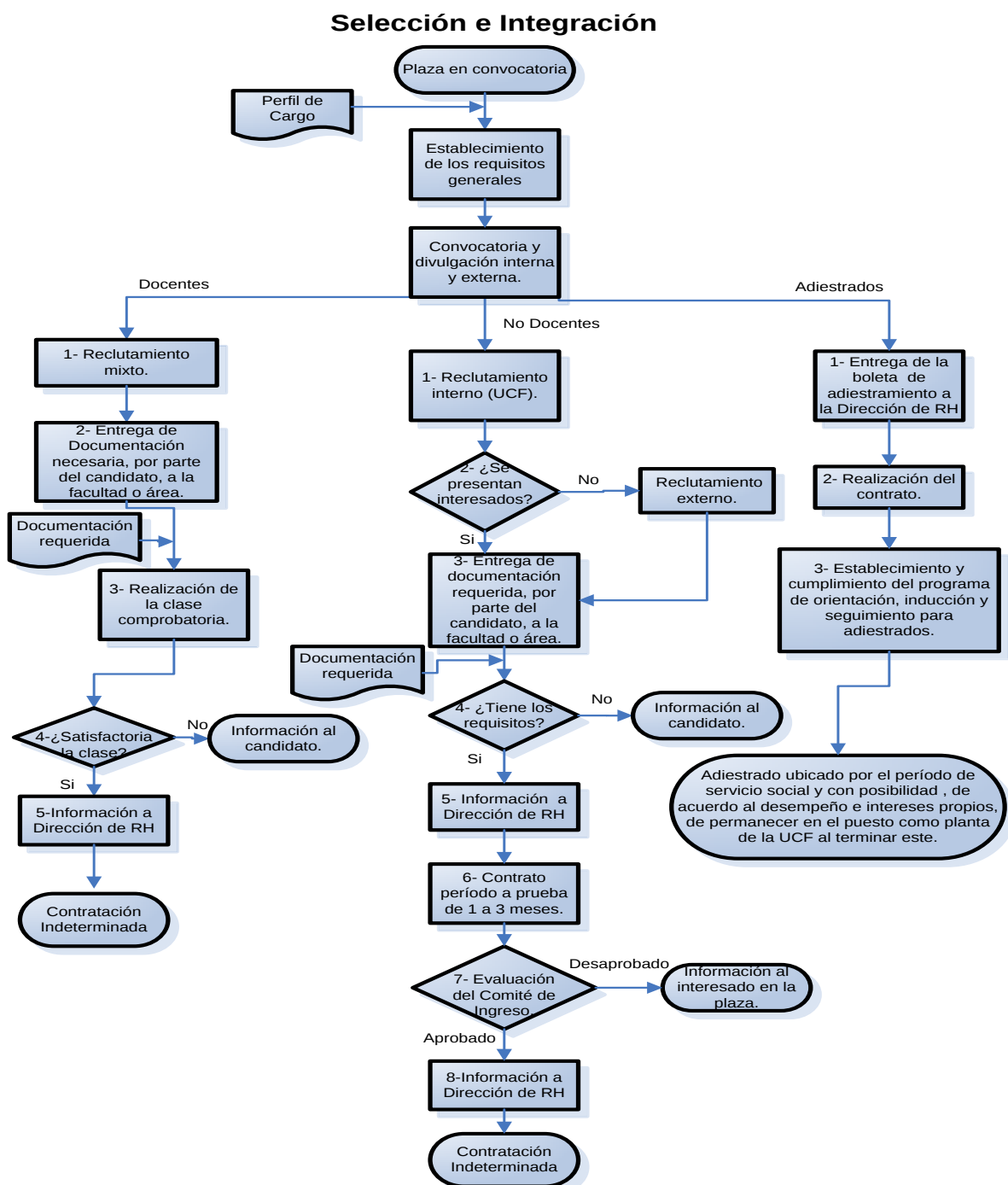
- [14] «Huma Nex - software para la gestión de procesos de selección y ofertas -. Nexo Digital». [En línea]. Disponible en: www.nexo-digital.com/huma-nex.
- [15] «Informacion ArmSTRONG». [En línea]. Disponible en: virtual5.com.mx/pdf/InformacionArmSTRONG.pdf.
- [16] «La selección de personal», EcuRed. .
- [17] López González E., Mendaña Cuervo C., y Rodríguez Fernández M. A., «La selección del personal con un algoritmo genético borroso».
- [18] E. López González, C. mendaña Cuervo, y M. A. Rodríguez Fernández, «LA SELECCIÓN DEL PERSONAL CON UN ALGORITMO GENÉTICO BORROSO1», Universidad de Leon, España.
- [19] «Lenguaje de Marcado de Hipertexto», Ecured. .
- [20] Departamento de Recursos Humanos, «Levantamiento de Procesos de Recursos Humanos». .
- [21] J. A. Alvaro Rodríguez, J. G. García Rodríguez, y y otros, Lógica Difusa. .
- [22] Robert G. Figueroa, «Metodologías tradicionales Vs. Metodologías ágiles».
- [23] «Metodologias tradicionales y metodologias agiles». [En línea]. Disponible en: <http://masteringenieriasoft.blogspot.com/2012/04/metodologias-de-desarrollo-de-software.html>.
- [24] Andres Ramos, Pedro Sánchez, José María Ferrer, Julián Barquin, y Pedro Linares, «Modelos Matemáticos de Optimización».
- [25] «Netbeans 6.8 liberado», 2010. [En línea]. Disponible en: <http://blogcultura.com/java/netbeans-6-8-liberado>.



- [26] Y. Arias Bravo, A. Rosete Suárez, y R. Rodríguez Martínez, «PROPUESTA INFORMATICA PARA SELECCIONAR PERSONAL POR COMPETENCIAS UTILIZANDO TÉCNICAS DE INTELIGENCIA ARTIFICIAL», 2006.
- [27] «Qué es Javascript». [En línea]. Disponible en: <http://www.desarrolloweb.com/articulos/25.php>.
- [28] «Real Academia Española», Diccionario de la Real Academia Española. .
- [29] German Leonardo Acevedo Orduña, «Selección de personal mediante redes neuronales artificiales», 2004.
- [30] J. L. de Armas Díaz, «Sistema Informático para el Control del Transporte en la Universidad de Cienfuegos», Carlos Rafael Rodríguez, 2012.
- [31] A. Pimentel Rivero y A. Hernández Domínguez, «Sistema Informático para la Gestión de Auditoría y Control (SIGAC). Modulo de Planificación», UCI, 2009.
- [32] «Sistemas de Información Gerencial». [En línea]. Disponible en: http://www.geocities.ws/jjaimes6603/fase3/t4_diferencias.html.
- [33] «Software de Reclutamiento y Selección de Personal HUNTERHRMS». [En línea]. Disponible en: www.humselect.com.
- [34] P. Craig Larman, Una introducción al análisis y diseño orientado a objetos y al proceso unificado. 2000.
- [35] «Visual Paradigm for UML». [En línea]. Disponible en: <http://www.freedownloadmanager.org/es/>.

Anexos

Anexo 1: Diagrama de flujo del proceso de Selección e Integración





Anexo 2: Planeación por funcionalidades.

Nro.	Prioridad	Fecha inicial	Fecha final	Días
1	A	15/01/2014	17/01/2014	2
2	A	18/01/2014	19/01/2014	1
3	A	19/01/2014	21/01/2014	2
4	A	21/01/2014	23/01/2014	2
5	A	23/01/2014	24/01/2014	1
6	A	24/01/2014	25/01/2014	1
7	A	25/01/2014	26/01/2014	1
8	A	26/01/2014	27/01/2014	1
9	A	28/01/2014	29/01/2014	1
10	A	29/01/2014	30/01/2014	1
11	A	21/01/2014	01/02/2014	1
12	A	01/02/2014	02/02/2014	1
13	A	02/02/2014	03/02/2014	1
14	A	04/02/2014	05/02/2014	1
15	A	05/02/2014	08/02/2014	3
16	A	09/02/2014	12/02/2014	3
17	A	12/02/2014	15/02/2014	3
18	A	17/02/2014	19/02/2014	2
19	A	20/02/2014	24/02/2014	4
20	A	24/02/2014	25/02/2014	1
21	A	25/02/2014	28/02/2014	3
22	A	01/03/2014	02/03/2014	1
23	A	02/03/2014	03/03/2014	1
24	A	04/03/2014	06/03/2014	2
25	A	06/03/2014	07/03/2014	1
26	A	07/03/2014	08/03/2014	1
27	A	08/03/2014	09/03/2014	1



28	A	10/03/2014	11/03/2014	1
29	A	12/03/2014	14/03/2014	2
30	A	15/03/2014	16/03/2014	1
31	A	18/03/2014	19/03/2014	1
32	A	19/03/2014	20/03/2014	1
33	A	20/03/2014	21/03/2014	1
34	A	21/03/2014	22/03/2014	1
35	A	22/03/2014	25/03/2014	3
36	A	25/03/2014	27/03/2014	2
37	A	27/03/2014	30/03/2014	3
38	A	31/03/2014	01/04/2014	1
39	A	01/04/2014	03/04/2014	2
40	A	03/04/2014	04/04/2014	1
41	A	04/04/2014	06/04/2014	2
42	A	07/04/2014	08/04/2014	1
43	A	08/04/2014	09/04/2014	1
44	A	09/04/2014	12/04/2014	3
45	A	12/04/2014	14/04/2014	2
46	A	14/04/2014	15/04/2014	1
47	A	15/04/2014	17/04/2014	2
48	A	17/04/2014	19/04/2014	2
49	A	19/04/2014	20/04/2014	1
50	A	20/04/2014	22/04/2014	2
51	A	22/04/2014	23/04/2014	1
52	A	23/04/2014	25/04/2014	2
53	A	25/04/2014	26/04/2014	1
54	A	26/04/2014	27/04/2014	1
55	A	27/04/2014	28/04/2014	1
56	A	28/04/2014	29/04/2014	1



57	A	29/04/2014	30/04/2014	1
58	A	01/05/2014	05/05/2014	4
59	A	05/05/2014	06/05/2014	1
60	A	06/05/2014	07/05/2014	1
61	A	07/05/2014	09/05/2014	2
62	A	10/05/2014	11/05/2014	1
63	A	11/05/2014	12/05/2014	1
64	A	12/05/2014	14/05/2014	2
65	A	14/05/2014	15/05/2014	1
66	A	15/05/2014	17/05/2014	2
67	A	18/05/2014	19/05/2014	1
68	A	20/05/2014	22/05/2014	2
69	A	22/05/2014	23/05/2014	1
70	A	23/05/2014	24/05/2014	1
71	A	24/05/2014	26/05/2014	2
72	A	26/05/2014	27/05/2014	1
73	A	28/05/2014	31/05/2014	3
74	A	01/06/2014	02/06/2014	2
75	A	02/06/2014	03/06/2014	1
76	A	04/06/2014	07/06/2014	3
77	A	07/06/2014	09/06/2014	2
78	A	09/06/2014	11/06/2014	2
79	A	11/06/2014	14/06/2014	3
80	A	14/06/2014	15/06/2014	1
81	A	15/06/2014	25/06/2014	10
82	B	26/06/2014	27/06/2014	1
83	A	27/06/2014	01/07/2014	4
84	A	01/07/2014	03/07/2014	2
85	A	03/07/2014	05/07/2014	2



86	B	05/07/2014	06/07/2014	1
----	---	------------	------------	---