



# **Implementación del Subsistema de Gestión de Información Personal de la Facultad 3.**

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

**Autor:**

Leonardo Darell Antúnez Naranjo

**Tutor:**

Ing. Zénel Pérez Reyes

Facultad 3  
Junio de 2012

## DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 3 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Leonardo Darell Antúnez Naranjo

Zénel Reyes Pérez

---

---

Firma de Autor

Firma de Tutor

# Agradecimientos

*A mis padres Teresa Naranjo y Leonardo Roberto Antúnez por traerme al mundo, por su constancia, su confianza, su sacrificio y su ejemplo.*

*A mis abuelos Elsa, Celeste, Agustina, Erasmo y Arjimiro por su amor, comprensión y dedicación en todo momento.*

*A mis tíos Damaris, Sandra, Olga, Julianis, Félix, Erasmo y Francisco por su preocupación, su apoyo incondicional y por tratarme como un hijo más.*

*A mis primos Daymaris, Licet, Yanet, Francisco, Félix, José y Geovanis por su cariño, su amor y más que nada por ser los hermanos que nunca tuve.*

*A toda mi gran familia.*

*A mis hermanos Evelín, Lizandra, Fernando, Leandro, Osmán, Rafael y Yannier por su lealtad y su apoyo en los momentos más difíciles.*

*A mi tutor Zénel por toda su ayuda y sus enseñanzas.*

*A los profesores que a lo largo de mi vida como estudiante han hecho de mi un profesional y sobre todo una mejor persona.*

*A Rayner y Mirta, por ser mis padres en la Universidad.*

*A Yorlen y Kamelia, por demostrarme que hermanos no solo son los que comparten la misma sangre.*

*A todos los amigos y compañeros que he tenido el placer de conocer en estos cinco años, en especial para Eddie, Roknier, Yailen, Mailin, Juan Antonio, Lissys, Rogelio, José Daniel, Arletys y Lien.*

*A todo mi equipo de desarrollo y amigos de la Plataforma Informativa SO3: Alieski, Reynaldo, Endel, Juan Ignacio, Yaicel, Eduardo, José Julio y Andy.*

*A Pedro Lluis y Ramón Pérez Medina por su ejemplo como revolucionarios, su ayuda y su confianza en mí.*

*A Iralda, Palanco, Osbel, Mity, Eloy, Raúl y otros compañeros de trabajo de mis padres por tratarme como un hijo más.*

*A Fidel, a la Revolución y al pueblo de Cuba, por darme la oportunidad de estudiar y formarme como profesional.*

*A todos... muchas gracias.*

## **Dedicatoria**

*Dedico el presente trabajo de diploma a mis padres y a mi familia, en especial a mis abuelos Elsa y Erasmo por todo su amor, sus sacrificios, su cariño y su confianza. Su recuerdo fue fuente de inspiración para la realización de este trabajo.*

*A Fidel y a la Revolución.*

*A todos mis compañeros y amigos.*

*A la Universidad de las Ciencias Informáticas.*

*Leonardo D. Antúnez Naranjo*

## Resumen

La Universidad de las Ciencias Informáticas, institución fundada en 2002, se ha caracterizado por la gran cantidad de recursos humanos que en ella se desenvuelven, llegando a contar en determinados periodos con más de 14 mil personas vinculadas a los procesos de formación, producción, investigación y otros servicios tercerizados. Los datos de estas personas son gestionados en diferentes sistemas especializados en determinados negocios. Dichos sistemas al estar centrados en solucionar determinados problemas dentro de la Universidad, no tienen en cuenta otros datos importantes para las áreas de la institución o estos datos no son actualizados con la frecuencia necesaria.

La situación anteriormente planteada ha provocado que para gestionar este tipo de información, las estructuras de la Universidad deben desarrollar sus propias aplicaciones, gestionarlos con hojas de cálculo o sencillamente utilizar el papel.

El presente trabajo de diploma tiene como objetivo fundamental dar solución al problema planteado anteriormente mediante la implementación de un subsistema que centralice la información personal concerniente a los recursos humanos de la Facultad 3, permitiendo una adecuada toma de decisiones y una eficiente gestión de la información por parte de sus directivos.

Como resultado de este trabajo se obtuvo una aplicación web capaz de centralizar y gestionar la información de los recursos humanos adscritos a las Facultad 3 de la Universidad de las Ciencias Informáticas y brindar estos datos a otras aplicaciones a través de servicios web.

Palabras claves: centralización, Facultad 3, gestión, información, plataforma, recursos humanos.

## Tabla de contenido

Agradecimientos .....	2
Resumen .....	4
Introducción .....	1
Capítulo 1. Fundamentación teórica .....	5
1.1.  Introducción.....	5
1.2.  Gestión de recursos humanos .....	5
1.3.  Soluciones informáticas de gestión de recursos humanos .....	7
1.3.1.  SAP ERP Human Capital Management (HCM).....	8
1.3.2.  OrangeHRM Open Source.....	9
1.3.3.  Sage AbraHRMS .....	10
1.3.4.  ASSETS NS .....	10
1.3.5.  VERSAT -Sarasola .....	11
1.3.6.  CEDRUX.....	12
1.3.7.  Resultados del estudio de las soluciones informáticas de gestión de recursos humanos.....	13
1.4.  Estudio de metodologías, tecnologías y herramientas. ....	14
1.4.1.  Proceso de desarrollo de software .....	15
1.4.2.  Lenguaje de Modelado .....	18
1.4.2.1.  Notación de Modelado de Procesos de Negocio (BPMN).....	18
1.4.2.2.  Lenguaje Unificado de Modelado (UML) .....	19
1.4.3.  Herramientas de Ingeniería de Software Asistida por Computación. ....	19
1.4.3.1.  Visual Paradigm .....	19
1.4.4.  Lenguajes de programación. ....	20
1.4.4.1.  PHP .....	21
1.4.4.2.  Java .....	21
1.4.5.  Librerías y frameworks de desarrollo. ....	22
1.4.5.1.  Symfony.....	22
1.4.5.2.  Zend Framework. ....	23
1.4.5.3.  jQuery.....	23
1.4.5.4.  Sencha ExtJS. ....	24

1.4.6.	Herramientas IDE.....	25
1.4.6.1.	Eclipse.....	25
1.4.6.2.	Netbeans 7.1.....	25
1.4.7.	Gestor de base de datos.....	26
1.4.7.1.	MySQL.....	26
1.4.7.2.	PostgreSQL.....	27
1.5.	Conclusiones.....	28
Capítulo 2. Desarrollo de la solución.....		29
2.1.	Introducción.....	29
2.2.	Descripción de la propuesta de solución.....	29
2.3.	Modelo de dominio.....	33
2.4.	Prototipos de interfaz de usuario.....	33
2.4.1.	Búsqueda simple.....	34
2.4.2.	Búsqueda avanzada.....	34
2.4.3.	Ficha básica.....	35
2.4.4.	Editar perfil.....	36
2.4.5.	Gestionar información personal.....	36
2.5.	Modelo de diseño.....	36
2.5.1.	Patrones arquitectónicos y de diseño.....	37
2.5.1.1.	Patrón Modelo Vista Controlador (MVC).....	37
2.5.1.2.	Patrones GRASP.....	38
2.5.1.3.	Patrones GoF.....	38
2.5.2.	Diagrama de clases del diseño.....	39
2.5.3.	Paquetes de clases.....	40
2.5.4.	Diagrama de clases de la solución.....	43
2.5.5.	Diseño de la Base de Datos.....	44
2.6.	Implementación de la solución.....	45
2.6.1.	Estándar de codificación.....	46
2.6.2.	Tratamiento de errores.....	47
2.6.3.	Diagrama de componentes.....	47
2.6.4.	Diagrama de despliegue.....	48
2.7.	Conclusiones.....	49

Capítulo 3 Validación de la solución propuesta .....	50
3.1. Introducción.....	50
3.2. Pruebas de software .....	50
3.2.1. Niveles de prueba. ....	51
3.2.1.1. Pruebas Unitarias .....	51
3.2.1.2. Pruebas de integración.....	51
3.2.1.3. Pruebas de sistema.....	52
3.2.1.4. Pruebas de aceptación.....	52
3.2.2. Tipos de pruebas. ....	53
3.2.2.1. Pruebas de caja negra.....	53
3.2.2.2. Pruebas de caja blanca. ....	54
3.2.2.3. Pruebas de rendimiento.....	54
3.2.2.4. Pruebas de resistencia (stress).....	55
3.3. Validación de la implementación. ....	56
3.3.1. Evaluación de la ejecución de las pruebas de caja negra.....	56
3.3.2. Conclusiones parciales sobre las pruebas de caja negra. ....	60
3.3.3. Evaluación de la ejecución de las pruebas de resistencia y rendimiento.....	60
3.3.3.1. Selección de la muestra de las funcionalidades del sistema.....	62
3.3.3.2. Ejecución de las pruebas. ....	63
3.4. Conclusiones.....	67
Conclusiones .....	68
Recomendaciones .....	69
Referencias bibliográficas.....	70



# Introducción

El desarrollo avanzado de las nuevas Tecnologías de la Informática y las Comunicaciones (TICs), ha provocado profundos cambios en el contexto social y laboral en el que se desenvuelve el hombre moderno. Como resultado directo de este proceso ha emergido el concepto de Sociedad de la Información, el cual ha irrumpido como un nuevo fenómeno a escala mundial donde la información y el conocimiento ocupan un lugar privilegiado en la sociedad, razón por la cual el proceso de creación, distribución y administración de la información constituyen un punto neurálgico a tener en cuenta en nuestras naciones y forma parte primordial de las actividades económicas de las mismas.

La alta disponibilidad y accesibilidad de la información en la sociedad actual, ha abierto el espectro a un cúmulo de aplicaciones orientados al manejo de la ciclópea cantidad de datos que día tras día es generada en todos los rincones del mundo. Estos sistemas responden a las necesidades específicas del dominio o negocio para el cual fueron creados, motivo por el cual el grado de descentralización y dispersión de los datos que en estos es procesado, es un tema a tener en consideración, sobre todo en una sociedad que consume enormes volúmenes de información.

Cuba no ha estado exenta de este fenómeno y muchos han sido los esfuerzos del gobierno encaminados a dotar al país de la infraestructura necesaria para la implementación de los procesos relacionados con la gestión de la información. Un importante papel en este proceso lo ha ocupado la Universidad de las Ciencias Informáticas (UCI), institución innovadora, de excelencia científica, académica y productiva que forma de manera continua profesionales integrales comprometidos con la patria, soporte de la informatización del país y la competitividad internacional de la industria cubana del software (UCI, 2008).

Desde la fundación de la UCI, esta se ha caracterizado por la gran cantidad de recursos humanos que en ella se desenvuelven, llegando a tener en determinados periodos más de 14 mil personas vinculadas a los procesos de formación, producción, investigación y otros servicios tercerizados. Se puede concluir, por tanto, que el manejo de la información laboral, académica, productiva e investigativa en una Universidad tan dinámica y numerosa, reviste vital importancia en el desarrollo integral de la institución. Si a lo anteriormente planteado se añade el papel que asume la UCI en la informatización del país, la gestión de los datos del personal

debe estar orientada al uso de sistemas informáticos que garanticen la disponibilidad, confidencialidad e integridad de la información.

Actualmente existen en la institución varios sistemas informáticos dedicados a la gestión de la información de los recursos humanos. Estas soluciones informáticas están orientadas a satisfacer las necesidades de un negocio o dominio específico en el entorno UCI, por lo que al especializarse en determinadas áreas, estas aplicaciones llegan a convertirse en plataformas y brindan servicios a otros sistemas, complementando de esta forma las necesidades de información en la Universidad. Entre estas aplicaciones se puede citar Akademos (Sistema de Gestión Académica, el cual recientemente migró hacia la plataforma de Gestión Universitaria), ASSETS (Sistema de Capital Humano), Registro de Residentes entre otros.

Gran parte de los recursos humanos con los que cuenta la Universidad están adscritos a las diferentes facultades creadas en la institución, las cuales a su vez poseen estructuras que le permiten controlar y orientar el trabajo de los individuos en el seno de las mismas.

La Facultad 3, surgida tras la fusión de las facultades 3 y 4 en febrero de 2009, aglutina en la actualidad más de 1400 personas entre estudiantes, trabajadores docentes y no docentes, cuyos datos se encuentran disponibles en los sistemas informáticos citados anteriormente. Como es de suponer, la información se encuentra dispersa en estas aplicaciones por lo que si se desea obtener un perfil íntegro de cierta persona, es preciso consultar o consumir los servicios que brindan las soluciones informáticas asociadas a cada área de la UCI.

Esta distribución descentralizada de los datos personales posee indudablemente sus ventajas ante un modelo centralizado, ya que permite a la aplicación reaccionar con rapidez ante las condiciones locales, lo cual responde al principio de que las organizaciones de hoy en día se han vuelto mucho más complejas y deben ser capaces de evolucionar ante los cambios dinámicos en sus entornos.

En el contexto de la Universidad, la descentralización de la información en diferentes sistemas informáticos es ciertamente relativa, debido a que muchos de estos sistemas a pesar de estar distribuidos, dependen de una actualización centralizada, afectando de este modo los procesos que se llevan a cabo en estructuras inferiores que dependen de estos datos. Cada facultad posee una estructura interna que puede ser flexible, dicha flexibilidad es sumamente difícil de representar en aplicaciones diseñadas para funcionar de forma genérica en diferentes

escenarios, por lo que se dificulta considerablemente la correlación y veracidad de los datos reales con respecto a los que se almacenan en los sistemas centrales.

La información es, sin lugar a dudas, una fuente de aprendizaje. Sin embargo, a menos que se organice, procese, y esté disponible para las personas adecuadas en un formato adecuado para la toma de decisiones, es una carga, no un beneficio. (Pollard, y otros, 2000). Precisamente la situación existente en la Universidad atenta contra la correcta toma de decisiones por parte de los directivos, ya que los datos relacionados con los recursos humanos se encuentran altamente dispersos y descentralizados. Por otro lado las aplicaciones creadas al efecto proveen esquemas rígidos que no se adaptan lo suficiente a las necesidades del negocio en cuestión y no permiten realizar búsquedas históricas.

Teniendo en cuenta la situación planteada anteriormente, se define como **problema a resolver**: cómo asegurar la certificación, accesibilidad, disponibilidad y actualización de los datos personales concernientes a los recursos humanos adscritos a la Facultad 3. Este problema conduce a que se plantee como **hipótesis**: si se implementa un subsistema que organice la información concerniente a los recursos humanos de la Facultad 3, entonces se podrá asegurar la certificación, accesibilidad, disponibilidad y actualización de la información personal concerniente a los recursos humanos adscritos a dicha facultad.

Para ello se tendrá como **objeto de estudio** la gestión de la información personal de los recursos humanos en la Facultad 3.

Se identifica además como **campo de acción** las soluciones informáticas para la gestión de la información personal de los recursos humanos en la Facultad 3 y como **objetivo general** se traza implementar un subsistema que organice la información personal concerniente a los recursos humanos de la Facultad 3, garantizando la certificación, accesibilidad, disponibilidad y actualización de la información.

Para dar cumplimiento al **objetivo general**, se consideran los siguientes **objetivos específicos**:

- Realizar el marco teórico de la investigación.
- Implementar la solución propuesta.
- Validar y evaluar técnicamente la solución.

Para dar cumplimiento al **objetivo general** propuesto y solucionar la situación problemática planteada se plantean un grupo de **tareas investigativas**:

- Elaboración del diseño teórico-metodológico de la investigación.
- Definición de situación problemática, problema, objetivos, novedad científica, aportes teóricos y prácticos, impacto social del tema a desarrollar.
- Revisión del estado del arte, efectuando una profunda revisión bibliográfica del tema que permita identificar soluciones informáticas similares y tomar posición al respecto.
- Definición de la propuesta de los elementos que formarán parte de la arquitectura base de la solución.
- Implementación de la solución informática.
- Evaluación técnica de la propuesta.

El presente trabajo de diploma está estructurado en 3 capítulos:

### **Capítulo 1: Fundamentación teórica**

En este capítulo se expone la fundamentación teórica del trabajo, incluyéndose en el mismo el estudio de otras soluciones informáticas existentes en la actualidad y de las metodologías, modelos de desarrollo de software y herramientas a utilizar.

### **Capítulo 2: Desarrollo de la solución**

En este capítulo se realiza la propuesta de solución al problema planteado con anterioridad, exponiendo los prototipos de interfaces, análisis y diseño de clases y los artefactos generados en la implementación.

### **Capítulo 3: Validación de la solución propuesta**

En este capítulo se plasman los resultados de los casos de prueba, obteniendo una valoración integral de la solución propuesta que permita determinar los puntos débiles del sistema.

# Capítulo 1. Fundamentación teórica

## 1.1. Introducción

El desarrollo vertiginoso de las TICs ha provisto al hombre de numerosas herramientas que mejoran considerablemente las condiciones de vida de las sociedad. Muchos han sido los avances en disímiles áreas como la telefonía celular, la creación de hardware y software cada vez mejores y más baratos, la televisión digital (DTV)<sup>1</sup> e Internet. Esta evolución hacia una sociedad cada vez más necesitada de información, ha sido el combustible ideal de esta revolución del conocimiento que se ha vuelto, en los albores del siglo XXI, parte intrínseca de todos los sectores de la sociedad. Por estas razones, el uso de tecnologías adecuadas diseñadas con especial atención a los aspectos éticos, culturales, sociales y económicos de la comunidad a la que se dirigen (FAO, 2009), cobra vital importancia en un mundo que se mueve entre bits y señales satelitales.

## 1.2. Gestión de recursos humanos.

La gestión de recursos humanos (HRM<sup>2</sup>) es la función dentro de una organización que se centra en la contratación, gestión y orientación de las personas que trabajan en la organización. Dicha función también se ocupa de cuestiones relacionadas con las personas, como la compensación, contratación, gestión del rendimiento, desarrollo de la organización, seguridad, bienestar, beneficios, motivación de los empleados, comunicación, administración y capacitación (Rudman, 2010).

La gestión de los recursos humanos constituye pieza clave en la organización de las estructuras de las naciones del mundo y cobra vital importancia para el desarrollo económico y social. Precisamente esta actividad ha estado a la vanguardia de las transformaciones tecnológicas acometidas internacionalmente, sobre todo en los procesos de automatización de dicha gestión.

Un breve recorrido por la historia de la gestión de los recursos humanos, permite conocer que este no es un término nuevo, pues estuvo siendo utilizado, principalmente en los Estados Unidos, como sinónimo de gestión de personal en la década de 1950. Gestión de personal

---

<sup>1</sup> Tecnología de transmisión avanzada que ha transformado la experiencia de ver televisión. La DTV ha permitido a las transmisoras ofrecer televisión con mejor calidad de imagen y sonido. También ofrece múltiples opciones de programación, lo que se denomina multifusión, y funciones interactivas (Dávalos, 2009).

<sup>2</sup> Siglas en inglés para Human Resources Management

tiene una historia mucho más larga, que se remonta a la Revolución Industrial e incluso mucho antes.

Las personas han sido administradas siempre que han trabajado para otros, sin embargo, los orígenes de la gestión moderna de personal se remonta fundamentalmente a la Revolución Industrial. Dos temas principales comenzaron a surgir en estos días (Rudman, 2010):

- La preocupación por el bienestar de los trabajadores.
- Los empleadores necesitaban guiar y controlar los trabajadores y sus esfuerzos.

La Revolución Industrial introdujo la especialización, la división del trabajo y la concentración del empleo en las fábricas. Así nace la gestión científica, movimiento que se inició a finales del siglo XIX. Su representante más prominente es el Ingeniero estadounidense Frederick Winslow Taylor, quien consideraba que los administradores podrían hacer uso de técnicas científicas para mejorar la productividad y la eficiencia (Rudman, 2010). Taylor es considerado el padre de la gerencia científica y en 1911 publica su obra más relevante: Principios de la Administración Científica<sup>3</sup> en la que sentó las bases de la gestión de personal a través de cuatro principios fundamentales (Bloomsbury, 2002).

El trabajo de Taylor tuvo un impacto significativo, ya que desarrolló el concepto del diseño y medición del trabajo así como del control de producción y otras funciones, que cambiaron por completo la naturaleza de la industria. Antes de la gestión científica, departamentos como los de recursos humanos, mantenimiento y control de la calidad no existían.

El término HRM refiriéndose a la gestión de recursos humanos fue acuñado en Estados Unidos a inicios de la década del 60 por Peter Drucker y Douglas McGregor (Rudman, 2010). Es a partir de este momento en que el movimiento de gestión de recursos humanos cobró mayor impulso en la nación nortea debido a la aprobación de varias leyes como la Ley de Igualdad Salarial de 1963, la Ley de Derechos Civiles de 1964, el Retiro de los Empleados de Seguridad de Ingreso de 1974 (ERISA), y la Ley de Salud y Seguridad de 1970. Para mediados de los 80 casi todas las industrias a gran escala y medianas empresas contaban con un departamento para gestionar su contratación, relaciones laborales, el mantenimiento de registros, los sueldos y salarios, etc. (Khilawala, 2010).

---

<sup>3</sup> The Principles of Scientific Management (1911), Nueva York. EE.UU

### 1.3. Soluciones informáticas de gestión de recursos humanos

La gestión de los recursos humanos a todos los niveles, va cobrando cada día una mayor importancia. La necesidad de aprovechar al máximo su potencial, se ha convertido en una necesidad en estos tiempos, motivo por el cual resulta imperioso el uso de soluciones informáticas para la gestión de estos recursos (HRMS)<sup>4</sup> las cuales permiten a las empresas automatizar muchos aspectos de la gestión de recursos humanos, logrando reducir de este modo la carga de trabajo del departamento de recursos humanos, y al mismo tiempo aumentar la eficiencia del departamento a través de la estandarización de los procesos de RRHH<sup>5</sup>. Numerosas han sido las soluciones informáticas surgidas desde la década del 80 y enfocadas al procesamiento de este tipo de información.

La gestión de los datos personales y la información personal de los recursos humanos de determinada estructura es manejada generalmente por herramientas HRMS, las cuales proporcionan las funcionalidades y técnicas fundamentales para la administración de este tipo de información. Es por esta razón que el estudio de las soluciones informáticas enfocadas a este tipo de actividad se basa fundamentalmente en soluciones de tipo HRMS.

Las HRMS por lo general son implementadas y utilizadas de forma modular. Cada módulo o subsistema está diseñado para realizar ciertas funciones y para que puedan integrarse con otros subsistemas, lo que permite al cliente adaptar el software y utilizar solamente las funcionalidades que realmente necesita. Es usual que estas aplicaciones cuenten con las cuatro funciones básicas del departamento de RRHH (Business Software, 2011):

- Planificación y modelación.
- Reclutamiento y contratación de personal.
- Gestión de información del personal.
- Gestión del rendimiento.
- Administración de beneficios.
- Tiempo y asistencia.
- Capacitación y educación.

Si bien estos módulos, cada uno proporciona beneficios para el departamento de recursos humanos, las aplicaciones también ayudan a la empresa como un todo. Las HRMS convierten la información de recursos humanos en formato digital, lo que permite que la información se

---

<sup>4</sup> Siglas en inglés para Human Resources Management Systems.

<sup>5</sup> Siglas empleadas para referirse a los Recursos Humanos.

añada a los sistemas de gestión del conocimiento de la empresa. El resultado de esto es que los datos de recursos humanos pueden ser integrados en la solución de planificación de recursos empresariales (ERP)<sup>6</sup> de la empresa u organismo (Heathfield, 2012).

El análisis de los datos obtenidos de estos sistemas provee a la organización de información veraz y muy valiosa, lo que permite mejorar la capacidad de toma de decisiones de los directivos. A continuación se describen algunas de las HRMS de interés para la investigación, ya sea por su notoriedad en el mercado o por sus características que las relacionan con el contexto del problema enunciado.

### 1.3.1. SAP ERP Human Capital Management (HCM)

SAP ERP Human Capital Management (SAP ERP HCM) es una solución integrada de gestión de recursos humanos que ayuda a las organizaciones a aumentar su rendimiento a través de una fuerza de trabajo eficiente, comprometida y motivada. SAP ERP HCM ofrece a organizaciones de todos los tamaños, sectores y regiones las herramientas necesarias para gestionar los recursos humanos. El sistema proporciona el marco ideal para:

- Alinear los objetivos de la estrategia corporativa a través del análisis integral, involucrar a los empleados e identificar, retener y desarrollar el talento clave a través de una oferta de gestión del talento robusto construido sobre una base central HCM.
- Aumentar la eficiencia y reducir los costes mediante la automatización de procesos transaccionales, así como servicios. (SAP ERP, 2011).

La suite<sup>7</sup> de soluciones SAP cuenta con aplicaciones y servicios bajo licencias y en demanda, e implementa una Arquitectura Orientadas a Servicios (SOA)<sup>8</sup> el cual provee un modelo para una arquitectura adaptable, flexible y abierta para el desarrollo de servicios basados en soluciones de escala empresarial de negocios.

La suite está compuesta por varios subcomponentes, entre los más importantes se pueden citar:

- **Gestión de personal:** Centraliza las actividades relacionadas con los recursos humanos y contribuye a las tareas de planificación.

<sup>6</sup> Siglas en inglés para Enterprise Resource Planning

<sup>7</sup> Colección de programas de computadora con funcionalidades similares.

<sup>8</sup> Siglas en inglés para Service-Oriented Architecture.



- **Gestión de tiempos:** Contempla los procesos que implican la planificación, registro y valoración del trabajo interno y externo realizado por los empleados, así como los tiempos de ausentismo.
- **Cálculo de nómina:** Se utiliza para calcular la remuneración de cada empleado.

### 1.3.2. OrangeHRM Open Source

OrangeHRM Open Source es un sistema de información que facilita la gestión de recursos humanos, además de proporcionar una amplia gama de características para la gestión de la organización. OrangeHRM proporciona una plataforma perfecta para alinear los procesos de recursos humanos con los objetivos de la empresa (Borrego, 2009). A continuación se citan los principales componentes o funcionalidades del sistema:

- **Reclutamiento:** permite llevar un listado de ofertas de trabajo y selección de personal.
- **Administración:** consisten en la definición de estructura de la empresa, los grados de remuneración y demás información que sirve como la columna vertebral para el resto del sistema.
- **Administrador de Información Personal (PIM)**<sup>9</sup>: este módulo mantiene toda la información relacionada con los empleados.
- **Auto Servicio de Empleado (ESS)**<sup>10</sup>: brinda a los empleados de la empresa la capacidad de visualizar información pertinente, vía web, sin tener que recurrir directamente al personal de recursos humanos.
- **Reportes:** esta característica permite al personal del departamento de recursos humanos elaborar informes personalizados de acuerdo a sus necesidades.
- **Licencias y vacaciones:** posee amplias posibilidades de definir los tipos de licencia y más.
- **Tiempo y Asistencia:** automatiza los procesos relacionados con el tiempo de seguimiento. El módulo ayuda a organizar el trabajo de manera eficiente, mejora los datos de gestión de la mano de obra y reduce al mínimo los errores en la ejecución de las políticas de la empresa de asistencia (OrangeHRM, 2011).

El sistema OrangeHRM se encuentra bajo licencia GNU General Public License. Por otro lado, se pueden contratar servicios de soporte, variando su precio entre \$50 – \$5,000 anuales dependiendo del plan contratado. La instalación puede realizarse mediante un servidor web,

<sup>9</sup> Siglas en inglés para Personal Information Management.

<sup>10</sup> Siglas en inglés para Employee Self-Service.

permitiendo así el acceso mediante Internet o en un servidor local, que cuente con PHP, MySQL y Apache (Borrego, 2009).

### **1.3.3. Sage AbraHRMS**

SageAbra HRMS ofrece servicios de automatización de los procesos de recursos humanos y nómina, permitiendo automatizar las actividades y el cumplimiento de estas por el personal de la organización, además de permitir realizar un seguimiento de la asistencia, los empleados y de la nómina.

Entre sus funcionalidades principales se encuentran:

- Simplificar las tareas rutinarias de gestión de los empleados mediante la automatización de su trabajo administrativo.
- Administrar y realizar un seguimiento de su cumplimiento de los requisitos del gobierno.
- Simplificar la contratación para que se pueda obtener el empleado correcto para el trabajo.
- Llevar un registro de asistencia de los empleados.
- Automatizar la formación de los empleados.
- Proporcionar los datos de actualizaciones empleadas y la inscripción de beneficios en línea (Sage HRMS, 2011).

Sage HRIS está disponible solamente en plataformas Windows y sistemas que utilizan SQL Server (HR Software Comparison, 2011).

### **1.3.4. ASSETS NS**

ASSETS NS es un Sistema de Gestión Integral estándar y parametrizado que permite el control de los procesos de Compras, Ventas, Producción, Taller, Inventario, Finanzas, Contabilidad, Presupuesto, Activos Fijos, Útiles y Herramientas y Recursos Humanos (ASSETS, 2004). Como Sistema Integral, todos sus módulos trabajan en estrecha relación, tributando automáticamente al Módulo de Contabilidad lo que permite que se pueda trabajar bajo el principio de Contabilidad al Día.

Es un sistema flexible, amigable, con ayuda en línea que puede ser instalado en una microcomputadora o sobre varias. ASSETS NS facilita el uso de la parametrización para adaptarse a las exigencias de cada entidad en particular, garantizando que sus reportes tengan la forma y el contenido que el usuario les defina.

ASSETS NS está diseñado para Multi Compañía, con una estructura organizativa a varios niveles, en la que podrán existir: Grupo Corporativo, Corporativo, Grupo de Agrupaciones, Agrupación, Almacenes y Centros de Costos. ASSETS NS, por las características anteriormente mencionadas, está considerado como una de las mejores soluciones informáticas de este tipo para el país.

### **1.3.5. VERSAT -Sarasola**

El VERSAT - Sarasola es un sistema integrado de gestión económica, diseñado para ser utilizado por el sector empresarial cubano, que se adecua a las características de cada entidad, ya que es configurable por cada una de ellas en el momento de su instalación y tiene como objetivo fundamental ofrecerle a los usuarios la posibilidad de contar con un instrumento seguro, rápido, eficaz y de fácil manejo para la Planificación, Control y el Análisis de la Gestión Económica (Cabrera González, y otros, 2011).

Permite llevar el control y el registro contable individual de todos los hechos económicos que se originan en las estructuras internas de las entidades y obtener los Estados Financieros y Análisis Económicos y Financieros en estos niveles.

Es un producto cubano, donde están incorporados los principios de Control Interno y demás Regulaciones vigentes en el País, con la garantía de continuar desarrollándose nuevas opciones y brindar una respuesta rápida a las nuevas legislaciones y mecanismos que se establezcan por los Organismos Rectores de las actividades Económicas y Financieras correspondientes.

Entre las características fundamentales del VERSAT Sarasola se pueden enumerar:

- Herramienta para la planificación económica, el control y el análisis de gestión.
- Diseñado para su empleo en cualquier tipo de entidad empresarial y presupuestada.
- Permite llevar el control y registro contable individual de todos los hechos económicos que se originan en las estructuras internas de las entidades, así como exponer el estado financiero y toda la información económica y contable en este universo.
- Se estructura en un grupo de subsistemas en los cuales se procesan y contabilizan los documentos primarios, donde se anotan los movimientos, los recursos materiales, laborales y financieros que se utilizan en una entidad.
- Se logra establecer un proceso de interacción usuario-sistema.

- Rapidez y fiabilidad, a partir de la configuración del proceso de contabilización de los documentos primarios y de las propias posibilidades de trabajo contenidas en cada subsistema.

### **1.3.6. CEDRUX**

CEDRUX es una solución genérica para la gestión de entidades basada en los principios de independencia tecnológica. El sistema automatiza funcionalidades generales de los procesos que tienen lugar en los organismos, teniendo en cuenta las particularidades de la economía cubana. Entre sus principales beneficios está la sustitución de importaciones por conceptos de Sistemas Informáticos de Gestión Empresarial, además de que servirá como herramienta fundamental para el control de los recursos del país y contribuirá a evitar pérdidas por desvíos y robos.

Pone al servicio de la entidad las potencialidades de la tecnología informática y provee facilidades para la integración de las diferentes áreas productivas y departamentos administrativos.

Características generales de la solución:

- Independencia Tecnológica.
- Plataforma Web.
- Interoperabilidad con otros sistemas.
- Multimoneda.
- Multientidades.
- Control de fechas.
- Transaccional.
- Integralidad funcional.
- Tratamiento Estadístico del procesamiento de la información.

El sistema posee una completa y moderna plataforma de aplicaciones Web, la cual será fácilmente accesible desde cualquier puesto de trabajo, evitando de este modo las limitantes de las aplicaciones de escritorio y además se disminuirán los ciclos de reposición de equipamiento informático por este concepto. Por otro lado, la plataforma está preparada para el trabajo orientado a servicios, creando las condiciones necesarias para la inclusión de nuevos módulos y la conexión a otras aplicaciones existentes en la entidad, brindando de esta forma una altísima adaptabilidad a las nuevas condiciones y cambios que ocurran en las entidades.

### 1.3.7. Resultados del estudio de las soluciones informáticas de gestión de recursos humanos.

Las soluciones informáticas orientadas a la gestión de los recursos humanos que han sido descritas en esta investigación, responden a determinadas necesidades del entorno empresarial para el cual fueron creadas, razón por la cual no son lo suficientemente flexibles como para adaptarse a las necesidades planteadas en la situación problemática de esta investigación. Por otro lado también deben tenerse en cuenta varias desventajas y aspectos negativos sobre estos productos que imposibilitan su uso en la Facultad 3.

**SAP ERP HCM**, respaldado por el renombre de su creador, el gigante SAP AG, es sin lugar a dudas una de las mejores soluciones empresariales para la gestión de recursos humanos, a lo que debemos sumar el soporte que brinda a sus usuarios. Sin embargo esta solución es privativa y su adquisición y puesta en marcha requiere el desembolso de altas sumas de dinero. Esta situación obviamente no está acorde con las políticas de independencia tecnológica adoptadas en nuestro país y en la UCI (UCI, 2008). En el mismo caso se encuentra la solución **Sage HRMS. OrangeHRM Open Source**, a pesar de poseer una licencia GNU/GPL, no es una opción recomendable puesto que entre las tecnologías que emplea se encuentra el Sistema Gestor de Base de Datos (SGBD) MySQL, propiedad de Oracle. Dicho gestor no está incluido en la arquitectura la plataforma SO3<sup>11</sup>, debido a la sensibilidad de la información manejada y a la posibilidad de que Oracle retire el soporte al SGBD o peor aún, cambie la licencia del software.

En cuanto a las soluciones nacionales, **ASSETS NS** provee gran cantidad de facilidades para una gestión personalizada de los recursos humanos, pero utiliza tecnologías privativas como el SGBD Microsoft SQL Server, además se debe apuntar que es un software de escritorio, lo cual trae consigo varias limitaciones como el riesgo de que la rotura de una estación de trabajo afecte el desarrollo de las actividades de la Facultad. **VERSAT Sarasola**, se encuentra limitado por situaciones similares, al ser desarrollado para plataformas Windows y emplear SGBD privativos.

De las soluciones analizadas, **CEDRUX** se aproxima considerablemente a las necesidades de la Facultad, por su flexibilidad y adaptabilidad a procesos existentes en la economía cubana. Es un software desarrollado en la UCI, lo cual garantiza el soporte al producto y el uso de tecnologías libres. Sin embargo el sistema de Capital Humano de dicho producto está diseñado

---

<sup>11</sup> La plataforma informativa SO3 surge con el propósito de crear un conjunto de aplicaciones destinadas a informatizar los procesos que tienen lugar en la Facultad 3, garantizando la interacción dichas aplicaciones y otras existentes en la UCI.

para ser la fuente primara de datos en la entidad donde se despliegue, lo cual entra en contradicción con las necesidades de la solución que se desea implementar, puesto que esta requiere que gran parte de sus datos sean cargados de los sistemas centrales de la Universidad.

Por otro lado no resulta viable la actualización del subsistema antes mencionado debido a que se atentaría contra la propia concepción del sistema de Capital Humano de CEDRUX. Además las diferencias respecto a las tecnologías empleadas en el desarrollo de la plataforma SO3 atentarían contra el posterior desarrollo de la solución informática y su interacción con las demás aplicaciones de la plataforma SO3.

Por lo anterior expuesto se concluye que:

- Ninguna de las soluciones informáticas existentes en el mercado suple las necesidades específicas del problema planteado.
- La mayoría de estas soluciones son privativas o emplean tecnologías privativas.
- Debido a la ausencia de una solución informática que gestione la información personal concerniente a los recursos humanos de la Facultad 3, se hace necesario la implementación de un sistema que sea capaz de mantener actualizado el perfil del personal adscrito a la Facultad, así como brindar estos datos a otros sistemas de manera segura a través de servicios.
- La solución debe ser una aplicación web, por las bondades que brinda el uso de la arquitectura cliente-servidor y debe utilizarse herramientas de software libre para su desarrollo.
- La solución, además de ser modular y extensible, debe brindar servicios web a otras aplicaciones.

#### **1.4. Estudio de metodologías, tecnologías y herramientas.**

En la actualidad, el uso intensivo de todo tipo de soluciones informáticas en múltiples sectores de la sociedad y sobre todo en los procesos productivos, ha provocado un notable crecimiento en la confiabilidad y calidad que estos deben poseer. Para desarrollar un producto que satisfaga estos indicadores se debe tener en cuenta una serie de tendencias, tecnologías, lenguajes, metodologías y herramientas indispensables para garantizar la entrega de un software competitivo y ajustado a las necesidades de los clientes.

A continuación se realiza un estudio de metodologías, tecnologías y herramientas con el propósito de determinar las más convenientes para la solución propuesta, garantizando de esta forma que la solución informática utilice las tecnologías propuestas en la arquitectura de la plataforma SO3.

#### **1.4.1. Proceso de desarrollo de software.**

El proceso de desarrollo de software "es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Concretamente "es una estructura aplicada al desarrollo de un producto software que define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo" (Jacobson, 1998).

El proceso de desarrollo de software requiere por un lado un conjunto de conceptos, una metodología y un lenguaje propio. A este proceso también se le llama el ciclo de vida del software que comprende cuatro grandes fases: concepción, elaboración, construcción y transición (Zavala Ruiz, 2008).

Para lograr la consecución exitosa de un proyecto de software, existen un conjunto de modelos que permiten perfilar el proceso de desarrollo, las cuales deben ser analizadas cuidadosamente. Usualmente la selección de uno de estos modelos o de la unión de varios responde a las necesidades y condiciones específicas del proyecto.

Entre estos modelos o metodologías se pueden clasificar en cinco grandes grupos (Software Development Methodologies, 2012):

- **Modelo de cascada:** provee un enfoque muy lineal para el desarrollo de software, ya que cada paso en esencia lleva al siguiente hasta que todo el software se ha completado.
- **Modelo iterativo incremental:** en muchos aspectos, el modelo incremental emula el modelo de cascada, con la diferencia que recomienda la construcción de secciones reducidas de software que irán ganando en tamaño para facilitar así la detección de problemas de importancia antes de que sea demasiado tarde.
- **Modelo espiral:** en cierto modo, el modelo de espiral es como una versión más grande e intensa del modelo incremental. Este modelo también es conocido por permitir el desarrollo de software de alta calidad, este resultado es obtenido gracias a la

característica fundamental del modelo que es la gestión de riesgos de forma periódica en el ciclo de desarrollo.

- **Modelos ágiles:** en la gestión de proyectos ágiles, existe un esqueleto básico conformado con las mejores prácticas basadas en el sentido común, pero la gestión global se basa en otros métodos. Este modelo utiliza un desarrollo iterativo como base para abogar por un punto de vista más ligero y más centrado en las personas. Los procesos ágiles utilizan retroalimentación en lugar de planificación como principal mecanismo de control.
- **Modelo de prototipos:** En el método de creación de prototipos, el software es desarrollado en forma similar a una cebolla. Cuando el proyecto se inicia, el equipo se centra en la construcción de un prototipo con todos o la mayoría de las características y construye todo un programa para que el cliente lo utilice. Después de que el cliente acepta el prototipo, el equipo de programación crea la siguiente capa mediante la corrección de los errores y basándose en los requerimientos fundamentales.

Teniendo en cuenta los modelos anteriormente planteadas, las características de la solución que se desea desarrollar y el equipo de trabajo, los modelos ágiles, junto a los modelos de prototipos, proveen un marco idóneo para el desarrollo del producto, debido a que este tipo de modelos ofrece las siguientes ventajas (Sousa, 2009):

- Ahorro de tiempo y recursos.
- Movimiento rápido, los avances más recientes pueden ser rápidamente codificados y probados usando este método.
- Los errores pueden ser fácilmente corregidos.
- Es un método controlado dinámicamente, que insiste en la actualización frecuente de los progresos en el trabajo a través de reuniones regulares.
- Se requiere constante retroalimentación del cliente.
- Debido a la retroalimentación constante, se hace más fácil hacer frente a los cambios.
- Los chequeos son realizados con frecuencia, lo que hace posible medir la productividad individual, esto conduce a la mejora del rendimiento de cada uno de los miembros del equipo.
- Los problemas se identifican con suficiente antelación a través de las reuniones frecuentes del equipo y por lo tanto se pueden resolver con rapidez.

Entre las principales metodologías ágiles se puede encontrar (Appelo, 2008):



- **Scrum** es una metodología de gestión de proyectos ágil, creado por Ken Schwaber y Sutherland Jeff. Se trata de un esqueleto que incluye un pequeño conjunto de prácticas y roles predefinidos. Scrum se está convirtiendo en un estándar de facto para la gestión de proyectos ágiles de desarrollo de software. Una de las razones de la popularidad de Scrum es que sólo consta de ciertas prácticas de sentido común que pueden aplicarse en muchas situaciones. Esto también significa que Scrum por sí mismo nunca es suficiente, y que los equipos de desarrollo tienen que utilizar simultáneamente otros métodos (por lo general XP) para prácticas adicionales.
- **Extreme Programming (XP)** es una metodología de ingeniería de software ágil, creado por Kent Beck. Se trata de un conjunto de buenas prácticas de los cuales algunos son llevados a niveles “extremos”. Al igual que con otros métodos ágiles, XP se refiere a los cambios en curso como un aspecto natural y deseable en el desarrollo de software. XP es a menudo visto como un complemento a Scrum, llenando la mayor parte de los agujeros que deja abiertos Scrum.
- **Agile Unified Process:** en español Proceso Unificado Ágil de Scott Ambler o (AUP) en inglés, es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. Abarca siete flujos de trabajos, cuatro ingenieriles y tres de apoyo: Modelado, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyectos y Ambiente.

Para el desarrollo de la solución se ha seleccionado **Scrum**, para ello se ha observado que una de las principales ventajas de este modelo es su flexibilidad y capacidad de adaptación ante los cambios en los requerimientos (Scrum, 2009). Además proporciona mecanismos de control para la planificación las liberaciones del producto. Esto significa que el proyecto puede ser alterado y modificado en función de las necesidades cambiantes, y al final se las arregla para establecer y ofrecer la versión más adecuada, que surge de la posibilidad de ajustar el trabajo a las cambiantes expectativas una vez que el proyecto está en marcha.

Debido a que los procesos manejados con Scrum también ofrecen mucho espacio para el trabajo individual y la contribución, los desarrolladores son libres de elaborar ideas y soluciones. Por lo general, estas ideas son pioneras e innovadoras ya que el equipo se basa en la mejor fórmula posible para la realización de su trabajo, a fin de terminar el proyecto en forma

adecuada y lo más eficientemente posible. El modelo Scrum emplea a personas expertas para trabajar esencialmente sin supervisión dentro de sus propias divisiones.

A modo de conclusión, el uso de Scrum mejora considerablemente la eficiencia del trabajo al estar fuertemente basado en la experiencia y la fiabilidad de las personas involucradas en el proyecto, proporcionando no sólo mayor unidad y aumento de la auto eficacia en los miembros del equipo, sino también un espacio que permite mejorarlos valores éticos y morales de los desarrolladores y la innovación en el proyecto. Por otro lado el entorno del grupo de desarrollo (equipo pequeño, dinamismo en el desarrollo de las aplicaciones) es propicio para el uso de Scrum.

### **1.4.2. Lenguaje de Modelado**

Usualmente para desarrollar una solución informática con calidad y competitiva, el uso de lenguajes de modelado es prácticamente indispensable. Esta necesidad responde a que el uso de estos lenguajes permite que el desarrollo de software se formalice a través de estándares unificados. Se entiende por lenguaje de modelado cualquier lenguaje artificial que puede ser utilizado para expresar la información, el conocimiento o sistemas en una estructura que está definida por un conjunto coherente de reglas. Las reglas se utilizan para la interpretación del significado de los componentes en la estructura. Los lenguajes de modelado utilizan modelos visuales y diagramas que realizan esa representación de manera precisa, entendible y económica, lo que facilita su uso en las herramientas CASE convencionales (Zapata, 2009).

Estos lenguajes se clasifican en dos tipos:

- Lenguajes gráficos: utiliza una técnica de diagramas con símbolos que representan los conceptos mencionados y las líneas que conectan los símbolos representan las relaciones.
- Lenguajes textuales: suelen utilizar palabras clave estandarizadas acompañadas de parámetros para que las expresiones sean interpretables por un ordenador.

#### **1.4.2.1. Notación de Modelado de Procesos de Negocio (BPMN<sup>12</sup>)**

Un modelo de proceso de negocio consiste en un conjunto de modelos de actividad y las limitaciones de ejecución entre ellos. Una instancia de proceso de negocios representa un caso concreto en el negocio operativo de una empresa, que consiste en las instancias de cada actividad. Cada modelo de procesos de negocio actúa como un modelo para un conjunto de

---

<sup>12</sup> Siglas en inglés para Business Process Modeling Notation

instancias de procesos de negocio, y a la vez cada actividad actúa como modelo para un conjunto de instancias de actividad (Weske, 2007).

BPMN (es el estándar más reciente para modelado de procesos del negocio y servicios. BPMN es una notación necesaria para expresar los procesos de negocio en un único diagrama de proceso de negocio (Business Process Diagram) BPMN permite hacer un mejor uso de la gestión de procesos del negocio (BPM), ya que normaliza el método de notación que sirve como ayuda en la automatización de los procesos (Fajardo Ugarte, 2008).

El principal objetivo de BPMN es proporcionar una notación estándar fácilmente comprensible por todos los miembros de la empresa. Por lo tanto, BPMN sirve como un lenguaje común, reduciendo la brecha de comunicación que se produce con frecuencia entre el diseño de procesos de negocio y la implementación.

#### **1.4.2.2. Lenguaje Unificado de Modelado (UML<sup>13</sup>)**

El Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado estandarizado de propósito general en el campo de la ingeniería de software orientada a objetos. La norma fue creada por el Object Management Group (OMG) en 1997, y desde entonces se ha convertido en el estándar del sector para el modelado de sistemas de software. UML incluye un conjunto de técnicas de notación gráfica para crear modelos visuales de programación orientada a objetos (Pressman, 2004).

Es importante resaltar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir.

Independientemente de las características de BPMN, los estándares que propone el lenguaje UML ofrecen innumerables ventajas para el proyecto. Teniendo en cuenta que la solución que se desea desarrollar se desenvuelve en un dominio, resulta más complicada la representación a través de procesos de negocio, por lo que el uso de UML se hace indispensable.

### **1.4.3. Herramientas de Ingeniería de Software Asistida por Computación.**

#### **1.4.3.1. Visual Paradigm**

Visual Paradigm es una herramienta de Ingeniería de Software Asistida por Computación (CASE<sup>14</sup>). La misma propicia un conjunto de ayudas para el desarrollo de programas

---

<sup>13</sup> Siglas en inglés para Unified Modeling Language

informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación (Visual Paradigm, 2012).

Visual Paradigm ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Constituye una herramienta de software libre de probada utilidad para el analista que fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque Orientado a Objetos.

Entre sus principales características se pueden mencionar (Visual Paradigm, 2012):

- Entorno de creación de diagramas para UML.
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa (versión profesional) e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo. Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad de integrarse en los principales IDEs.
- Generación de bases de datos.
- Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- Importación y exportación de ficheros XML.

Por las características antes mencionadas, así como las numerosas posibilidades que se brindan al usuario y a su condición de software libre, se selecciona Visual Paradigm como herramienta CASE.

#### **1.4.4. Lenguajes de programación.**

Un lenguaje de programación es un idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina o para expresar algoritmos con precisión. A continuación se describen algunos de los lenguajes de programación más populares y acordes a las necesidades de la solución informática que se desea desarrollar.

---

<sup>14</sup> Siglas en inglés para Computer Assisted Software Engineering

#### **1.4.4.1. PHP.**

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. PHP es ampliamente utilizado para fines generales aunque es especialmente adecuado para el desarrollo web y puede ser embebido en páginas HTML (The PHP Group, 2012).

PHP brinda la posibilidad de usar programación procedimental o programación orientada a objetos (POO), o una mezcla de ambos. Una de las características más fuertes y más importantes de PHP es su soporte para una amplia gama de bases de datos. Escribir una página web habilitada para la base de datos es muy simple utilizando una de las extensiones de bases de datos específicas (por ejemplo, para MySQL), o el uso de una capa de abstracción como PDO, o conectarse a cualquier base de datos que soporte el estándar de base de datos de conexión abierta a través de la extensión de ODBC.

Como se puede apreciar, son innumerables las características y beneficios que PHP ofrece, principalmente a las aplicaciones web, como es el caso de la solución informática que se desea desarrollar. Si a ello se agrega la existencia de una amplia comunidad formada por expertos programadores y la disponibilidad de frameworks de desarrollo que facilitan el trabajo con el lenguaje, PHP se convierte en una excelente opción para llevar a feliz término la aplicación que se desea desarrollar. La versión 5.3 del lenguaje provee características que mejoran el rendimiento y seguridad del software, permitiendo de esta manera una eficiente gestión de la información.

#### **1.4.4.2. Java**

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Java puede ser considerado un lenguaje de propósito general, pues es utilizado en el desarrollo de aplicaciones de escritorio, pasando por aplicaciones web, hasta las aplicaciones para teléfono móviles. Su sintaxis es fácil de usar y está soportado por una amplia comunidad de programadores expertos.

Java utiliza una máquina virtual (bytecode) para interpretar y traducir las instrucciones de alto nivel a lenguaje binario entendible para las computadoras. El uso de este intermediario supone

un uso menos eficiente de la memoria, pero suprime gran cantidad de errores originados por la incorrecta manipulación de punteros.

Entre noviembre de 2006 y mayo de 2007, Sun Microsystems liberó la mayor parte de sus tecnologías Java bajo la licencia GNU GPL, de acuerdo con las especificaciones del Java Community Process, de tal forma que prácticamente todo el Java de Sun es ahora software libre.

#### **1.4.5. Librerías y frameworks de desarrollo.**

Un framework simplifica considerablemente el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Proporciona una estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener, además de facilitar la programación de aplicaciones pues encapsula operaciones complejas en instrucciones sencillas. Por otro lado las librerías persiguen un objetivo similar, con la diferencia de que están orientadas a un uso más pasivo y usualmente no generan código fuente en la aplicación sino que el desarrollador hace uso de las funcionalidades que esta provee. A continuación se describen algunos de los principales frameworks de desarrollo para PHP y librerías Javascript.

##### **1.4.5.1.        **Symfony.****

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web (Potencier, 2008).

Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas Unix, como en plataformas Windows. A continuación se muestran algunas de sus características (Potencier, 2008).

- Fácil de instalar y configurar en la mayoría de plataformas.
- Independiente del sistema gestor de bases de datos.

- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

Symfony constituye una perfecta elección para el desarrollo de la solución informática que se desea desarrollar, hecho que responde a las características y buenas prácticas de diseño que hacen de este framework líder indiscutible en la implementación de aplicaciones web con PHP.

Symfony provee características y filosofías fundamentales para el desarrollo de una plataforma, de la cual la solución que se desea desarrollar constituye un núcleo base de vital importancia. Si a ellos le sumamos la existencia de gran cantidad de plugins y de una numerosa y activa comunidad, Symfony se convierte en un excelente opción para desarrollar este tipo de aplicaciones.

#### **1.4.5.2. Zend Framework.**

Zend Framework es un marco de trabajo que se basa en la simplicidad, está completamente orientado a objetos gracias al uso de buenas prácticas y posee una base de código rigurosamente ágil. Zend Framework se centra en la construcción de soluciones webs seguras, confiables y modernos. También es utilizado extensivamente en la creación de servicios web y APIs (Zend Technologies Ltd., 2012).

La estructura de los componentes de Zend Framework es algo único; cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado.

Zend Framework ofrece un gran rendimiento y una robusta implementación del Patrón Modelo Vista Controlador, una abstracción de base de datos fácil de usar, y un componente de formularios que implementa la prestación de formularios HTML, validación y filtrado para que los desarrolladores puedan consolidar todas las operaciones usando de una manera sencilla la interfaz orientada a objetos.

#### **1.4.5.3. jQuery.**

jQuery es una librería de JavaScript rápida y concisa que simplifica el uso del documento HTML, el manejo de eventos, la animación, y las interacciones Ajax para el desarrollo web rápido (The jQuery Project, 2010).

Anunciado en 2006 por su creador, John Resig, jQuery ganó rápidamente popularidad y apoyo al convertirse en una nueva forma de utilizar JavaScript para interactuar con HTML y CSS. Los selectores de jQuery son simples, imitando selectores CSS, por lo que es una biblioteca familiar y fácil de aprender para los diseñadores y desarrolladores por igual. La librería jQuery borró la preocupación que los desarrolladores web había sufrido al intentar crear sitios interactivos a través de una amplia gama de navegadores por el manejo de la mayoría de los problemas de compatibilidad de navegadores (Castledine, y otros, 2010).

Aparte de ser muy fácil de usar, una de las mayores ventajas de jQuery es que maneja una gran cantidad de “estándares” para varios navegadores web. En la actualidad esta librería es ampliamente utilizada en disímiles aplicaciones web. Su reputación ha crecido exponencialmente al igual que la comunidad de desarrolladores que comparten sus experiencias así como plugins que complementan el trabajo con la librería.

La última versión estable de jQuery es la 1.7.2, la cual es seleccionada para ser utilizada en el desarrollo de la solución informática propuesta.

#### **1.4.5.4. Sencha ExtJS.**

Sencha ExtJS (anterior ExtJS) es una biblioteca de JavaScript para el desarrollo de aplicaciones web interactivas que además de flexibilizar el manejo de componentes de la página como el DOM, Peticiones AJAX, DHTML, tiene la gran funcionalidad de crear interfaces de usuario bastante funcionales. Fue creada en el año 2007 por Jack Slocum, Brian Moeskau, Aaron Conran y Rich Waters (Frederick, y otros, 2008).

Sencha ExtJS ofrece una atractiva y fácil de usar interfaz de usuario, similar a una aplicación de escritorio. Esto permite a los desarrolladores web concentrarse en la funcionalidad de las aplicaciones web en lugar de las advertencias técnicas y las incompatibilidades respecto a los navegadores (Frederick, y otros, 2008).

Su selección para ser utilizado en la solución informática propuesta, responde a las numerosas ventajas que aporta la librería, sobre todo al trabajo con peticiones asincrónicas al servidor, las cuales mejoran considerablemente la experiencia del usuario final de la aplicación, sin dejar de mencionar las herramientas que brinda para la creación de ventanas, formularios y validadores.



### **1.4.6. Herramientas IDE**

#### **1.4.6.1. Eclipse.**

Eclipse es un entorno de desarrollo integrado (IDE), de código abierto y multiplataforma. Es una potente y completa plataforma de programación, desarrollo y compilación de elementos tan variados como sitios web, programas en C++ o aplicaciones Java.

Eclipse, al mismo tiempo, es una comunidad de código abierto, cuyos proyectos se centran en la construcción de una plataforma de desarrollo abierta compuesta por marcos extensibles, herramientas de tiempos de ejecución para la construcción, implementación y administración de software a través del ciclo de vida de los mismos. La Fundación Eclipse es una organización sin fines de lucro que aloja los proyectos de Eclipse y ayuda a cultivar tanto una comunidad de código abierto y un ecosistema de productos y servicios complementarios (Eclipse Foundation, 2012).

#### **1.4.6.2. Netbeans 7.1.**

Netbeans es un entorno de desarrollo integrado de código abierto para desarrolladores de software. Cuenta con todas las herramientas necesarias para crear aplicaciones profesionales de escritorio, empresariales, web y aplicaciones móviles con la plataforma Java, así como con C / C + +, PHP, JavaScript y Groovy (Oracle Corporation, 2012).

NetBeans no sólo es la mejor opción para el desarrollo de Java, sino que también recibe el soporte integrado para lenguajes dinámicos, todo en una herramienta de gran alcance. El editor de PHP de NetBeans ofrece plantillas de código y generación (getters y setters), la refactorización, información sobre herramientas de parámetros, consejos y soluciones rápidas, y la finalización de código inteligente. Provee de un resaltado de código sintáctico y semántico, documentación emergente, formateo de código y plegado, y el marcado de los sucesos y puntos de salida.

El IDE también ofrece un completo editor de HTML, JavaScript y CSS, con la ventaja de proveer un resaltado de sintaxis completo, completamiento de código inteligente, documentación pop-up, y la comprobación de errores para HTML, CSS y JavaScript, incluyendo HTML 5, JavaScript 1.7. El editor reconoce código HTML en los archivos de JavaScript y viceversa. El editor reconoce también HTML y JavaScript en XHTML, PHP y archivos JSP.

Teniendo en cuenta las características anteriormente planteadas, se selecciona el IDE Netbeans para desarrollar la solución propuesta, por todas las facilidades que brinda para el trabajo con proyectos PHP y al soporte a tecnologías de última generación como HTML5 y CSS3.

#### **1.4.7. Gestor de base de datos**

Un sistema de gestor de bases de datos (DBMS<sup>15</sup>), algunas veces llamada simplemente un administrador de base de datos, es un programa que permite a uno o más usuarios de computadoras crear y acceder a los datos en una base de datos. El DBMS gestiona las peticiones del usuario (y las solicitudes de otros programas) para que estos sean libres de entender que los datos se encuentran físicamente en los medios de almacenamiento. En el manejo de solicitudes de los usuarios, el DBMS asegura la integridad de los datos (es decir, asegurándose de que sigue siendo accesible y es constantemente organizada según lo previsto) y seguridad (asegurándose de que sólo aquellos con privilegios de acceso pueden acceder a los datos). El más típico DBMS es un sistema de base de datos relacional (Christiansen, 2012).

La selección adecuada del DBMS proveerá a la solución informática que se desea desarrollar numerosas herramientas, además de seguridad a los datos que se almacenan y mejorará considerablemente el desempeño de la aplicación final. A continuación se describen algunos de los DBMS más utilizados en software con características similares.

##### **1.4.7.1. MySQL**

MySQL, es el más popular sistema de gestión de base de datos multiplataforma de código abierto. Es desarrollado, distribuido y soportado por Oracle Corporation. Fue creado en 1995 por David Axmark, Allan Larsson y Michael Widenius ( Oracle Corporation, 2012).

MySQL se caracteriza fundamentalmente por su velocidad, rendimiento y fiabilidad, motivo por el cual es ampliamente utilizado en sitios web y aplicaciones de escritorio de baja y mediana complejidad. Muchas de las organizaciones más grandes y de más rápido crecimiento del mundo, incluyendo Facebook, Google, Adobe, Alcatel Lucent y Zappos basan sus sistemas en MySQL.

Tanto el software del servidor de MySQL en sí y las bibliotecas de cliente utilizan una doble licencia de distribución. Se ofrecen bajo licencia GPL, a partir del 28 de junio de 2000 (que

---

<sup>15</sup> Siglas en inglés para DataBase Management System

Oracle ha ampliado con una excepción de licencia de software libre) (Oracle Corporation, 2012) o una licencia propietaria. Precisamente por esta razón el uso de MySQL no es recomendado para la solución informática propuesta, puesto que Oracle puede retirar en cualquier momento la licencia GPL del DBMS y se estarían incumpliendo las políticas de soberanía tecnológica de la Universidad y del país.

#### **1.4.7.2. PostgreSQL**

PostgreSQL es un avanzado servidor de base de datos SQL, disponible en una amplia gama de plataformas. Uno de los beneficios más claros de PostgreSQL es que es de código abierto, además de que sus versiones son mantenidas por largos periodos de tiempo y requiere poco o ningún mantenimiento en muchos casos (Riggs, y otros, 2010).

Entre las características fundamentales de PostgreSQL se pueden encontrar (Riggs, y otros, 2010):

- Excelente cumplimiento de los estándares SQL 2008.
- Arquitectura cliente-servidor.
- Diseño altamente concurrentes donde los lectores y escritores no se bloquean entre sí.
- Altamente configurable y extensible para muchos tipos de aplicaciones.
- Integridad referencial (ForeignKeys).
- Triggers Vistas (Views).
- Integridad Transaccional (ACID).
- Control de versionado concurrente (MVCC).
- Tiene licencia de tipo BSD.

PostgreSQL se caracteriza además por ser un sistema de base de datos de propósito general que permite crear sus propias funciones de servidor en una docena de idiomas diferentes. El DBMS es altamente extensible, de modo tal que es posible agregar sus propios tipos de datos, operadores y los tipos de índices.

PostgreSQL en su versión 9.1 es seleccionado como sistema gestor de base de datos debido a su condición de software de código abierto y a la potencia que brinda en el procesamiento de la información, además de ser un software estable, con gran escalabilidad y que cuenta con funcionalidades que lo destacan como uno de los DBMS más potentes en la actualidad. El sistema que se desea desarrollar requiere la gestión de altos volúmenes de información, por lo

que se necesita un sistema que maneje eficientemente las transacciones y la persistencia de los datos de manera segura.

### **1.5. Conclusiones**

En el capítulo se ha realizado un estudio minucioso de los principales conceptos asociados a la gestión de información personal de los recursos humanos, logrando esclarecer aún más los objetivos de la investigación.

Se realizó un estudio del estado del arte de las principales tendencias observables a lo largo de la historia de los sistemas de gestión de recursos humanos. A partir del estudio se determinó que la solución no es un HRMS, debido a que estos sistemas por lo general no solo gestionan la información de las personas, sino que también incluyen otras funcionalidades como reclutamiento, gestión de contratos, nóminas, entre otras. A pesar de que la solución que se desea desarrollar no constituye un HRMS, si se pueden extraer experiencias y buenas prácticas utilizadas en este tipo de sistemas de gestión.

Como parte de la definición de la propuesta de los elementos que formarán parte de la arquitectura base de la solución se plantea el uso de herramientas libres que estén acordes con las políticas tecnológicas de la Universidad y del país. Una vez realizadas las investigaciones pertinentes, se concluye que el proceso de desarrollo de la solución debe ser guiado por la metodología ágil Scrum, se debe utilizar el lenguaje UML para estandarizar la descripción del proyecto y Visual Paradigm como herramienta CASE.

Como lenguajes de programación se seleccionan PHP 5.3 del lado del servidor y Javascript como lenguaje script del lado del cliente. Además se utilizará Symfony 1.4 como framework de desarrollo con Propel 1.6 como ORM, así como las librerías jQuery 1.7 y Sencha Ext JS 3.3.1. Netbeans 7.1 será el IDE empleado en el desarrollo de la solución y como gestor de base de datos se utilizará PostgreSQL 9.1.

## **Capítulo 2. Desarrollo de la solución.**

### **2.1. Introducción.**

El siguiente capítulo tiene como objetivo fundamental la presentación de la solución propuesta, así como la modelación del dominio que permita una mayor comprensión de las funcionalidades que se desean implementar. En el acápite se muestran además las principales clases identificadas, el modelo de datos de la aplicación, se describen algunos flujos significativos para el desarrollo del software, los patrones de diseño utilizados y se presentan los diagramas de componentes y de despliegue.

### **2.2. Descripción de la propuesta de solución.**

Teniendo en cuenta los objetivos que persigue el presente trabajo de diploma y en aras de utilizar la información recopilada en el capítulo anterior, se propone como solución desarrollar una aplicación web que posibilite la gestión de la información personal de los recursos humanos adscritos a la Facultad 3 de las Universidad de las Ciencias Informáticas.

La solución informática debe ser capaz de sincronizar en bases de datos locales, los datos que se brindan a través de servicios webs por los diferentes sistemas existentes en la Universidad. El hecho de duplicar la información se debe a la necesidad de conservar un registro histórico con los datos de las personas, funcionalidad que en la actualidad no brinda la infraestructura de servicios webs de la UCI, lo que dificulta la gestión de la información en la Facultad 3.

Esta sincronización puede realizarse tanto manual como automáticamente en dependencia de las necesidades de información de la facultad. Por otro lado, la solución también debe permitir realizar búsquedas de personas atendiendo a diferentes criterios de selección, los cuales están en dependencia del rol y nivel de acceso que tenga el usuario, garantizando de esta forma la privacidad y confidencialidad de la información sensible de las personas registradas en la aplicación.

La solución propuesta debe permitir además la gestión personal de los datos de cada usuario. Esta administración será realizada a modo de edición de perfil personal, donde cada persona tendrá la opción de gestionar su propia información.

Existen datos de interés para la facultad que no pueden ser tomados de servicios webs ni las personas cuentan con los permisos suficientes para editarlos, por lo que se requiere además

de un sistema de gestión donde varias personas identificadas en sus respectivas áreas, puedan editar dicha información.

A continuación se muestra una tabla con los datos a gestionar por la aplicación con el origen de los mismos, y los responsables de editarlos.

Dato	Disponibilidad			Fuente de información		
	Estudiantes	Trabajadores docentes	Trabajadores no docentes	Web services	Perfil de usuario	Gestión por áreas
Usuario	Disponible	Disponible	Disponible	X		
Solapín	Disponible	Disponible	Disponible	X		
Área	Disponible	Disponible	Disponible	X		
Ocupación	Disponible	Disponible	Disponible	X		
Graduado de	No Disponible	Disponible	Disponible		X	
Graduado en	No Disponible	Disponible	Disponible		X	
Año de graduado	No Disponible	Disponible	Disponible		X	
Línea de investigación	Disponible	Disponible	Disponible		X	
Categoría docente	No Disponible	Disponible	No Disponible			X
Centro	Disponible	Disponible	Disponible	X		
Departamento	No Disponible	Disponible	Disponible	X		
Proyecto	Disponible	Disponible	Disponible	X		
Rol	Disponible	Disponible	Disponible	X		
Profesor guía	Disponible	Disponible	No Disponible			X
Cargo	No Disponible	Disponible	Disponible	X		X
Dirección UCI	Disponible	Disponible	Disponible	X		
Teléfono	Disponible	Disponible	Disponible	X		
Organizaciones	Disponible	Disponible	Disponible			X
Provincia	Disponible	Disponible	Disponible	X		
Municipio	Disponible	Disponible	Disponible	X		
Ubicación en la defensa	Disponible	Disponible	Disponible			X
Asignatura	Disponible	Disponible	No Disponible			X
Grupo	Disponible	No Disponible	No Disponible	X		

Deportista	Disponible	Disponible	Disponible			X
Artista	Disponible	Disponible	Disponible			X
Tutor	Disponible	No Disponible	No Disponible			X
Alumno ayudante	Disponible	No Disponible	No Disponible			X

Tabla 1 Relación de datos a gestionar por la solución.

Por último, la solución debe ser capaz de integrarse con otras aplicaciones de la plataforma SO3, brindando información esencial para el funcionamiento de estas. Esta integración debe realizarse a través de servicios web siempre garantizando la disponibilidad, confidencialidad e integridad de la información.

En el siguiente diagrama se muestra a grandes rasgos el entorno en el cual debe desenvolverse la solución informática propuesta. El Subsistema de Gestión de Información Personal (SGIP) forma parte fundamental del núcleo de la plataforma SO3, junto al Subsistema de Gestión Integral de Seguridad. Ambos deben proveer a la plataforma los datos básicos del personal y seguridad respectivamente.

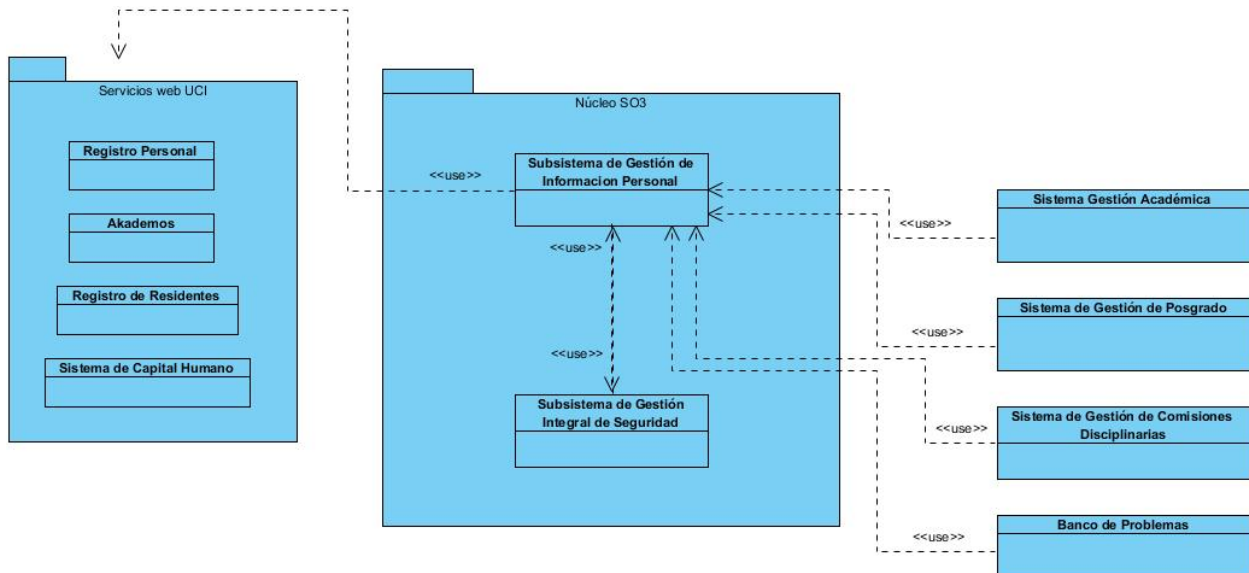


Diagrama 1 Representación del entorno del Subsistema de Gestión de Información Personal

Para llevar a cabo el desarrollo de la propuesta, se partió del análisis realizado, identificándose los siguientes requisitos funcionales<sup>16</sup>:

<sup>16</sup> La descripción de los requisitos anteriormente mencionados puede encontrarse en el documento **001\_Descripción de historia de usuario** anexo como un entregable del presente trabajo de diploma.

F01- Realizar búsqueda simple de persona.  
(Funcionalidad crítica)<sup>17</sup>

F02- Realizar búsqueda avanzada de persona. (Funcionalidad crítica)

F03- Mostrar la ficha básica de una persona. (Funcionalidad crítica)

F04- Obtener datos de servicios web.  
(Funcionalidad crítica)

F05- Gestionar datos de las graduaciones.

F06- Gestionar responsabilidades.

F07- Gestionar datos del tutor.

F08- Gestionar categoría docente.

F09- Gestionar cargos.

F10- Gestionar profesor guía.

F11- Gestionar organizaciones.

F12- Gestionar asignación de materias.

F13- Gestionar alumno ayudante por materia.

F14- Gestionar deportes por persona.

F15- Gestionar manifestación artística por persona.

F16- Gestionar ubicación para la defensa.

F17- Gestionar asignación grupos por profesores.

F18- Generar PDF.

F19- Generar fichero CSV.

F20- Generar lista de correo.  
(Funcionalidad crítica)

F21- Enviar correo. (Funcionalidad crítica)

---

<sup>17</sup> Se identifica como funcionalidad crítica todas aquellas que recibirán gran cantidad de peticiones por parte de los usuarios o que sean importantes para la aplicación.



### 2.3. Modelo de dominio.

El entorno en el que circunscribe la solución que se desea desarrollar cuenta con flujos de información difusos con tienen múltiples orígenes, lo que conlleva al uso de un modelo de dominio para modelar el negocio a informatizar. El objetivo principal de este modelo es comprender y describir las clases más importantes dentro del contexto del sistema a través de diagramas UML.

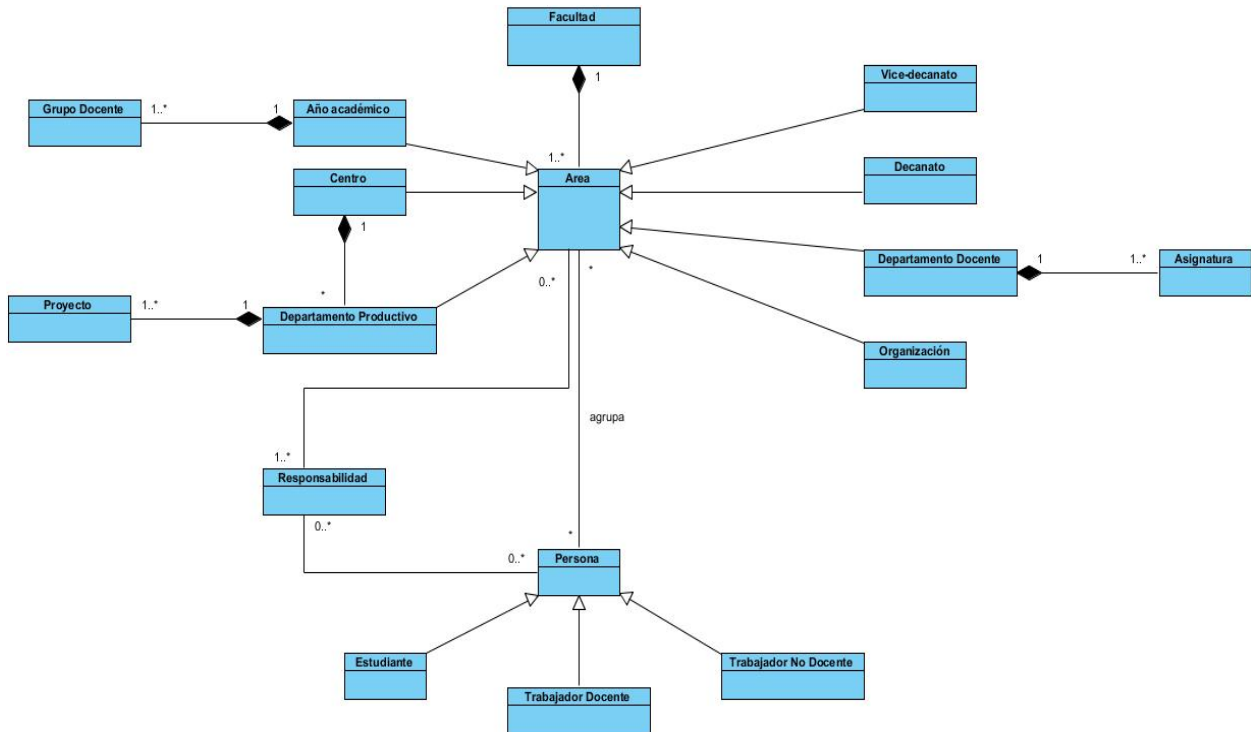


Diagrama 2 Modelo de dominio de la solución

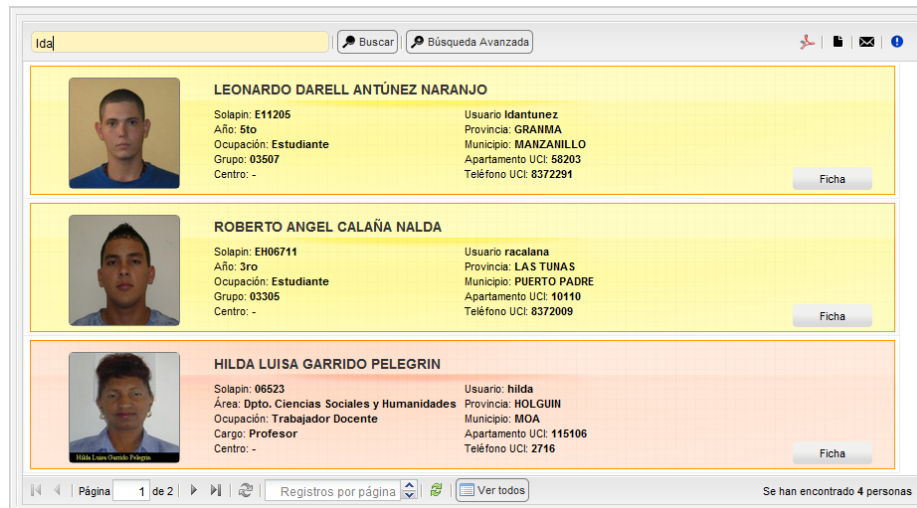
### 2.4. Prototipos de interfaz de usuario.

La elaboración de los prototipos de interfaz de usuario está encaminada a la creación de un guía que permitirá posteriormente a los diseñadores y desarrolladores comprender el flujo de los procesos del negocio así como establecer un estándar en cuanto a al diseño y la manera en que se muestra la información.

A continuación se muestran los prototipos de interfaz de usuario de la solución propuesta con una breve descripción del flujo a realizar por los usuarios.

### 2.4.1. Búsqueda simple.

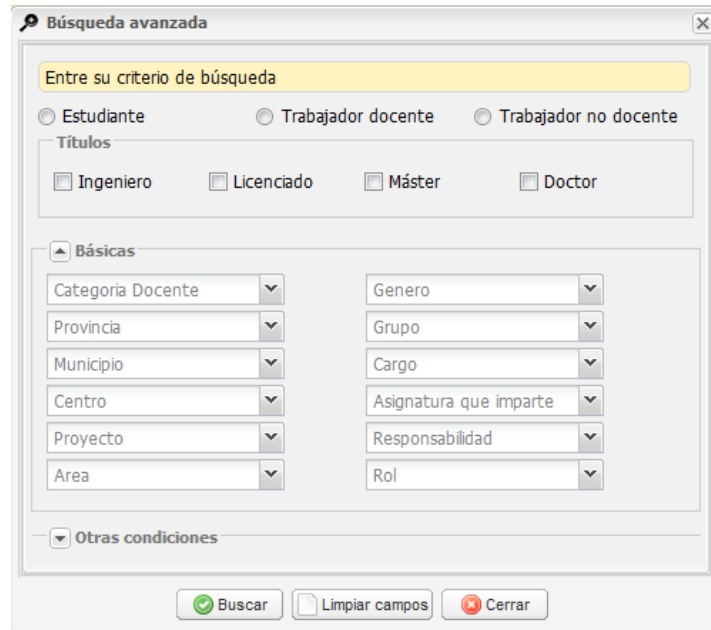
La interfaz de Búsqueda está orientada a la búsqueda de personas registradas en la base de datos a partir de datos comunes como el nombre, apellidos o usuario del dominio. Al no permitir la inclusión de criterios avanzados, esta interfaz no presenta restricciones para ningún usuario que la solicite. Cuenta además con una paginación de resultados para que el usuario personalice la cantidad que desea visualizar.



Interfaz 1 Búsqueda simple

### 2.4.2. Búsqueda avanzada.

En caso de que se desee realizar una búsqueda con otros criterios no incluidos en el buscador simple, el usuario tiene la posibilidad de efectuar una búsqueda avanzada. Una vez seleccionados los criterios, los resultados de la búsqueda se mostrarán en la interfaz búsqueda simple.



Interfaz 2 Búsqueda avanzada

### 2.4.3. Ficha básica.

La ficha básica muestra un conjunto de información elemental de las personas registradas en la base de datos de la plataforma. Esta ficha es mostrada al hacer clic en el botón **Ver ficha** de la interfaz Búsqueda simple.



Interfaz 3 Ficha con datos extendidos de la persona

#### 2.4.4. Editar perfil.

Esta es la interfaz creada para que un usuario determinado actualice parte de los datos que se muestran en su ficha, el panel izquierdo es cargado automáticamente y es un componente llamado en lo adelante debido a que esta información es tomada de los servicios web de la Universidad y no pueden ser modificados.

Interfaz 4 Editar perfil

#### 2.4.5. Gestionar información personal.

Este paquete de interfaces fue creada con el fin de que personas autorizadas por las diferentes áreas donde se gestiona la información puedan actualizar cualquier registro en la base de datos de la aplicación, dejando la evidencia correspondiente. Esta es una GUI con permisos especiales a la cual un usuario común no tiene acceso.

Para ver una descripción más detallada del flujo de cada interfaz puede consultarse el **Anexo 1** del presente trabajo de diploma.

#### 2.5. Modelo de diseño.

El objetivo fundamental de este modelo es lograr una solución lógica basada en el paradigma orientado a objetos. Su esencia es la elaboración de diagramas que muestran gráficamente cómo los objetos se comunicarán entre ellos a fin de cumplir con los requerimientos. Para

preparar un buen diseño de clases se hace sumamente necesaria la aplicación de los principios de la asignación de responsabilidades y utilizar los patrones de diseño (Larman, 1999).

### 2.5.1. Patrones arquitectónicos y de diseño.

Si bien la elaboración de un buen diseño del software contribuye directamente a la calidad del producto final, gran parte de esta yace en la utilización adecuada de los patrones de diseño y arquitectónicos existentes en la actualidad. Los patrones son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son soluciones basadas en la experiencia y que se ha demostrado que funcionan.

Con empleo del framework Symfony se garantiza la utilización de varios de estos patrones, permitiendo a los desarrolladores utilizar buenas prácticas de programación y ahorrar tiempo y recursos. A continuación se describen los principales patrones empleados en el diseño e implementación de la solución.

#### 2.5.1.1. Patrón Modelo Vista Controlador (MVC).

El patrón MVC obliga a dividir y organizar el código de acuerdo a las convenciones establecidas por el framework, la interfaz de usuario se guarda en la vista, la manipulación de datos se guarda en el modelo y el procesamiento de las peticiones constituye el controlador. El empleo del patrón MVC en un sistema resulta bastante útil además de restrictivo. Symfony está basado en el patrón de arquitectura conocido MVC, formado por tres niveles: el Modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio, la Vista es la encargada de originar las páginas que son mostradas como resultado de las acciones y el Controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista. Symfony toma lo mejor de la arquitectura MVC y la implementa de forma que el desarrollo de aplicaciones sea rápido y sencillo.

La implementación que realiza Symfony de la arquitectura MVC incluye varias clases como son:

- **sfController:** es la clase del controlador y se encarga de decodificar la petición y transferirla a la acción correspondiente.
- **sfRequest:** guarda todos los elementos que integran la petición (parámetros, cookies, cabeceras, entre otros).
- **sfResponse:** posee las cabeceras de la respuesta y los contenidos. El contenido de este objeto se convierte en la respuesta HTML que se remite al usuario.

### 2.5.1.2. Patrones GRASP

- **Controlador:** Symfony implementa un controlador frontal para atender todas las peticiones web, es el único punto de entrada de toda la aplicación en un determinado entorno. Cuando recibe una petición, utiliza el sistema de enrutamiento para enviar la acción al controlador responsable de la solución y se encarga de manejar la seguridad y ejecución de los filtros. Este patrón se evidencia en las clases `sfFrontController`, `sfWebFrontController`, `sfContext`, los "actions" y el `index.php` del ambiente.
- **Alta Cohesión:** Symfony permite la organización del trabajo en cuanto a la estructura del proyecto y la asignación de responsabilidades con una alta cohesión. Un ejemplo de ello son las clases actions, las cuales son responsables de definir las acciones para las plantillas y colaborar con otras para realizar diferentes operaciones, instanciar objetos y acceder a las propiedades.
- **Bajo Acoplamiento:** las clases relacionadas con el controlador frontal y que se encuentran en el paquete actions heredan únicamente de `sfActions` para alcanzar un bajo acoplamiento entre clases. Las clases que implementan la lógica del negocio y de acceso a datos se encuentran en el modelo, las cuáles no se asocian directamente con las clases de la vista o el controlador, lo que proporciona que la dependencia en este caso sea baja.
- **Experto:** Symfony utiliza Propel para realizar su capa de abstracción en el modelo, encapsular toda la lógica de los datos y generar las clases con todas las funcionalidades comunes de las entidades, las clases de abstracción de datos poseen un grupo de funcionalidades que están relacionadas directamente con la entidad que representan y contienen la información necesaria de la tabla que representan.

### 2.5.1.3. Patrones GoF

- **Singleton:** Symfony emplea el patrón de instancia única en varias de sus clases principales. Por ejemplo, las clases `sfRouting` y `sfContext` presentan el método `getInstance()`, que devuelve una instancia única de estas clases.
- **Command:** se encuentra presente en la clase `sfWebFrontController`, en el método `dispatch()`, es la encargada de establecer el módulo y la acción que se va a usar según la petición del usuario. Además se aplica en la clase `sfRouting`, que está desactivada por defecto y procede según las necesidades del administrador del sistema donde se aplique el framework, la cual se puede activar o desactivar.

- **Decorator:** la clase abstracta `sfView`, padre de todas las vistas, contienen un decorador para permitir agregar funcionalidades dinámicamente. El archivo `layout.php`, conocido como plantilla global, contiene el código HTML que es tradicional en todas las páginas del sistema, para no tener que repetirlo. El contenido de la plantilla se integra en el `layout`, o si se mira desde el otro punto de vista, el `layout`<sup>18</sup> decora la plantilla.

### 2.5.2. Diagrama de clases del diseño.

Los diagramas de clases del diseño son diagramas de estructura estática que muestran las clases del sistema y sus interrelaciones. El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces (las de Java, por ejemplo) en una aplicación (Larman, 1999). Estos son considerados como el pilar básico del modelado con UML porque son utilizados, tanto para mostrar lo que el sistema puede hacer (análisis) como para mostrar cómo puede ser construido (diseño).

La utilización del framework `Symfony` así como las librerías `ExtJS` permiten desarrollar un diseño con una estructura similar para varios escenarios. La página servidora que en el caso de la solución que se desea implementar es **`directorio.php`** se encargará de atender las peticiones que realiza el cliente y en dependencia del módulo al que se haga la petición, esta página servidora incluirá las clases **`busquedaActions`**, **`perfilActions`**, **`loaderActions`** o **`administracionActions`**; las cuales construyen el `layout` utilizando las clases plantilla definidas con anterioridad.

El `layout` global carga en el cuerpo la plantilla o página cliente (**`indexSuccess.php`**) todas las interfaces de usuario del módulo, estas interfaces están definidas en el fichero de configuración `view.yml` y se encuentran representadas dentro del paquete `Interfaces JS`. Es muy frecuente la existencia de una relación de agregación entre la página cliente y los formularios, que contienen los componentes para la entrada de datos que serán enviados al servidor y procesados por las funciones de la clase **`action`** que se corresponda con el módulo al cual se realiza la operación.

A continuación se muestra el diagrama de clases del diseño para el módulo de búsquedas.

---

<sup>18</sup> Archivo que utiliza el framework `Symfony` para la creación de una plantilla común para los módulos de determinada aplicación.

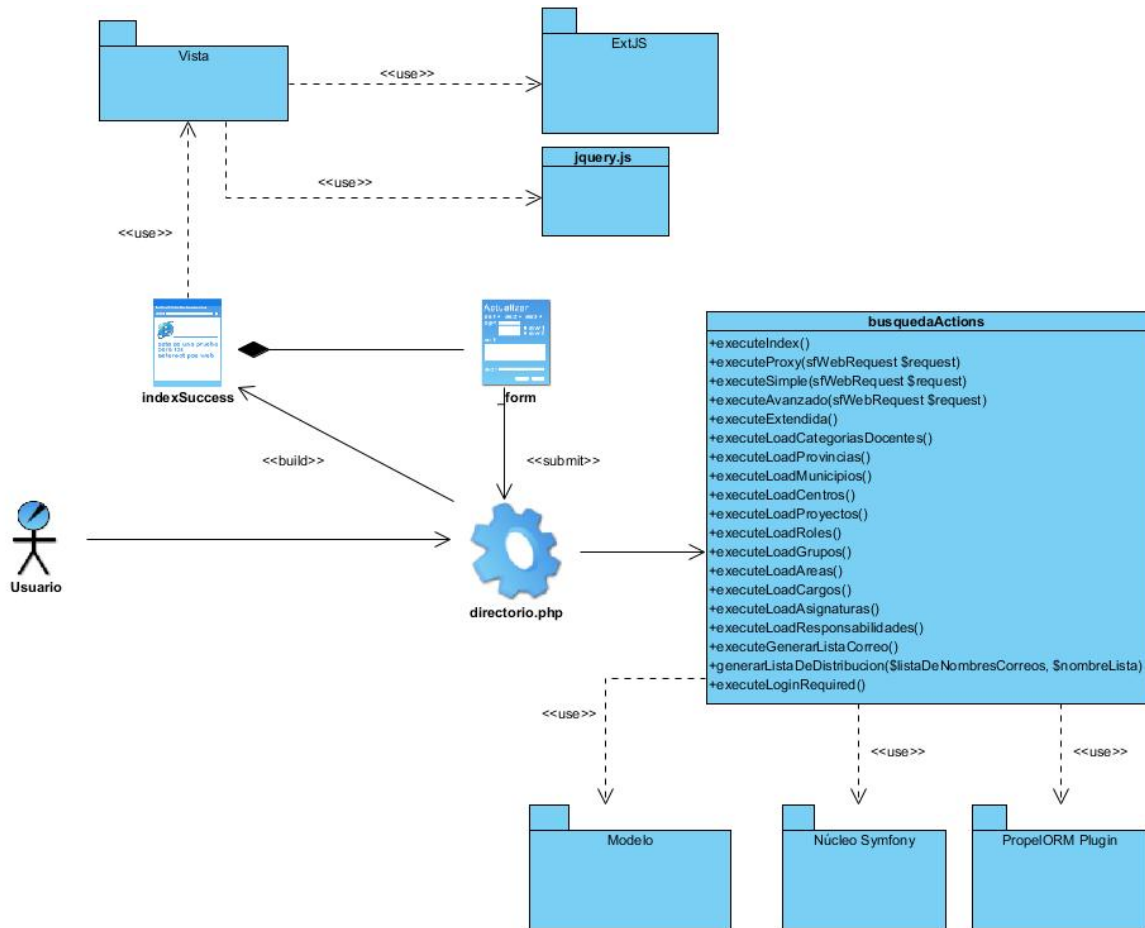


Diagrama 3 Diagrama de clases del diseño con estereotipos web del módulo Búsqueda.

### 2.5.3. Paquetes de clases

El diagrama de paquetes es utilizado para mostrar las agrupaciones lógicas en las que se encuentra dividido un sistema y las dependencias entre ellas. Suministran una descomposición de la jerarquía lógica de un sistema. Los paquetes contienen elementos del modelo al más alto nivel, tales como clases y sus relaciones, máquinas de estado, diagramas de casos de uso, interacciones, y colaboraciones: cualquier elemento que no esté contenido en otro.

A continuación se describe la distribución por paquetes diseñada para la posterior implementación de la solución.



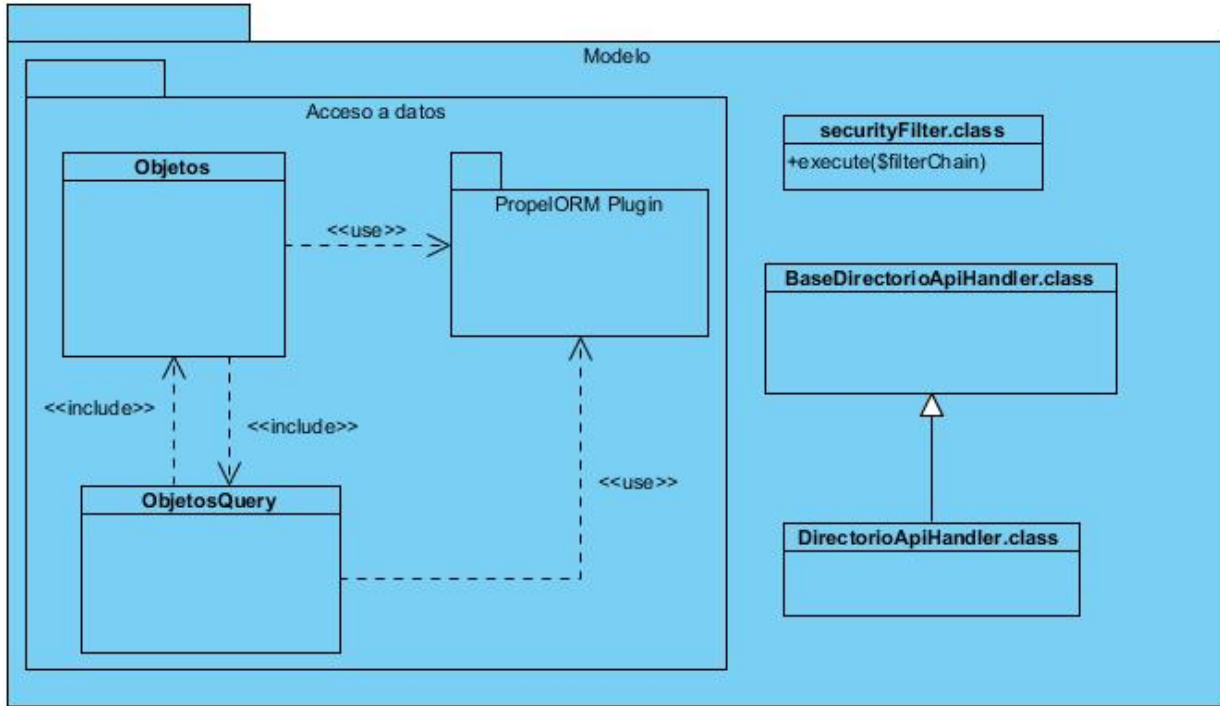


Diagrama 4 Paquete de clases pertenecientes al Modelo de la solución.

El paquete del Modelo está compuesto por un paquete de Acceso a datos cuyas funcionalidades fundamentales están asociadas al ORM Propel. El paquete PropelORMPlugin representa la capa de abstracción de objetos/relacional utilizada por la aplicación.

El paquete Objetos tiene las clases de la lógica de la aplicación, que contienen los atributos y métodos necesarios. Por cada entidad o tabla del modelo físico de datos, existen tres clases que pertenecen al paquete Objetos, por ejemplo: la tabla **dir\_persona** genera la clase DirPersona que hereda de BaseDirPersona. Puede ser necesario añadir métodos y propiedades personalizadas en los objetos, estos se efectúan sobre las clases hijas (DirPersona), pues las clases padre (BaseDaRemisionEntrada) poseen las funciones para acceder a la base de datos mediante Propel, e incluye del paquete ObjetosQuery su clase correspondiente (ejemplo DirPersonaQuery).

Las clases del paquete ObjetosQuery proporcionan los medios necesarios el trabajo con los registros en la base de datos. Sus métodos devuelven normalmente un objeto o una colección de objetos de la clase objeto relacionada.

En el paquete de Modelo se encuentra además la clase securityFilter.class.php. Dicha clase, a través del método execute() es la encargada de realizar el chequeo de seguridad previo a la

llamada de cada una de las acciones de la aplicación. Esta clase hace uso de los filtros especificados en el archivo filters.yml, así como de las funcionalidades que provee el Framework para el trabajo con las sesiones y los datos de usuario.

La clase DirectorioApiHandler.class.php, que hereda de BaseDirectorioApiHandler.class.php, es la encargada de manejar las llamadas a servicios realizadas a la aplicación. Estas clases hacen uso de CkWebServicePlugin como librería de terceros que permita la comunicación y el intercambio de datos entre aplicaciones en diferentes lenguajes de programación a través de servicios web.

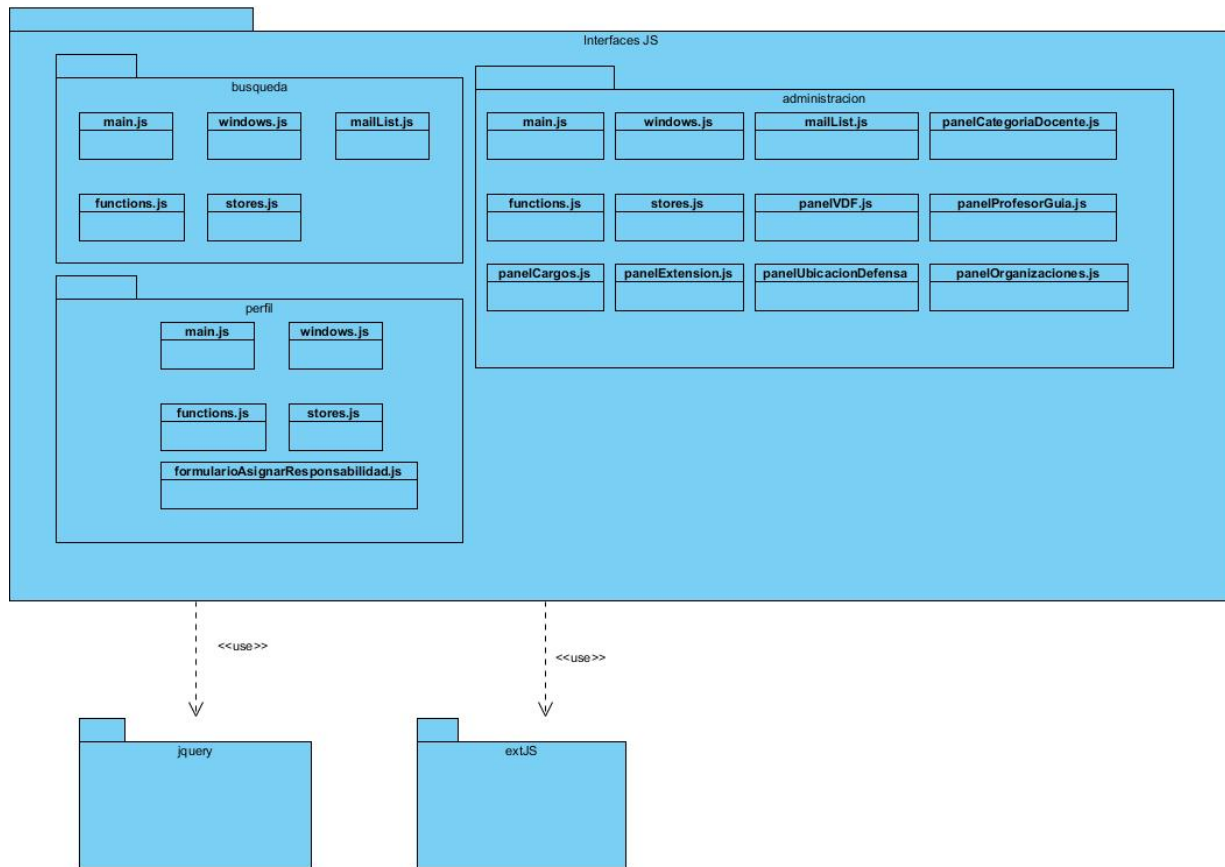


Diagrama 5 Paquete de Interfaces JS

El paquete Interfaces JS está compuesto por los archivos Javascript (extensión .js) necesarios para mostrar la interfaz de usuario de la aplicación. Estos archivos están organizados en paquetes relacionados con los módulos a los cuales pertenecen. Todos los paquetes contienen el archivo main.js que es la clase principal y punto de entrada a la interfaz de cada módulo. El paquete Interfaces JS hace uso de los paquetes ExtJS y jQuery que se corresponden con las

librerías Javascript empleadas en el desarrollo de la aplicación. Cada uno de los paquetes contiene los siguientes archivos:

- **windows.js**: utilizado para el lanzamiento de ventanas modales
- **functions.js**: empleado para la implementación de funciones o métodos que no proveen las librerías Javascript usadas y que complementan las demás interfaces del módulo.
- **stores.js**: destinado a la declaración de objetos comunes de la clase Store de la librería ExtJS.

#### **2.5.4. Diagrama de clases de la solución.**

Un diagrama de clases es una descripción de las clases en un sistema y sus relaciones. No describe el comportamiento dinámico del sistema sino la estructura del mismo mostrando sus clases, atributos y relaciones entre ellos. (Pressman, 2004)

A continuación se muestran algunas de las clases identificadas durante el diseño de la solución. Dichas representación sirve como una primera aproximación al diseño definitivo del modelo de datos.

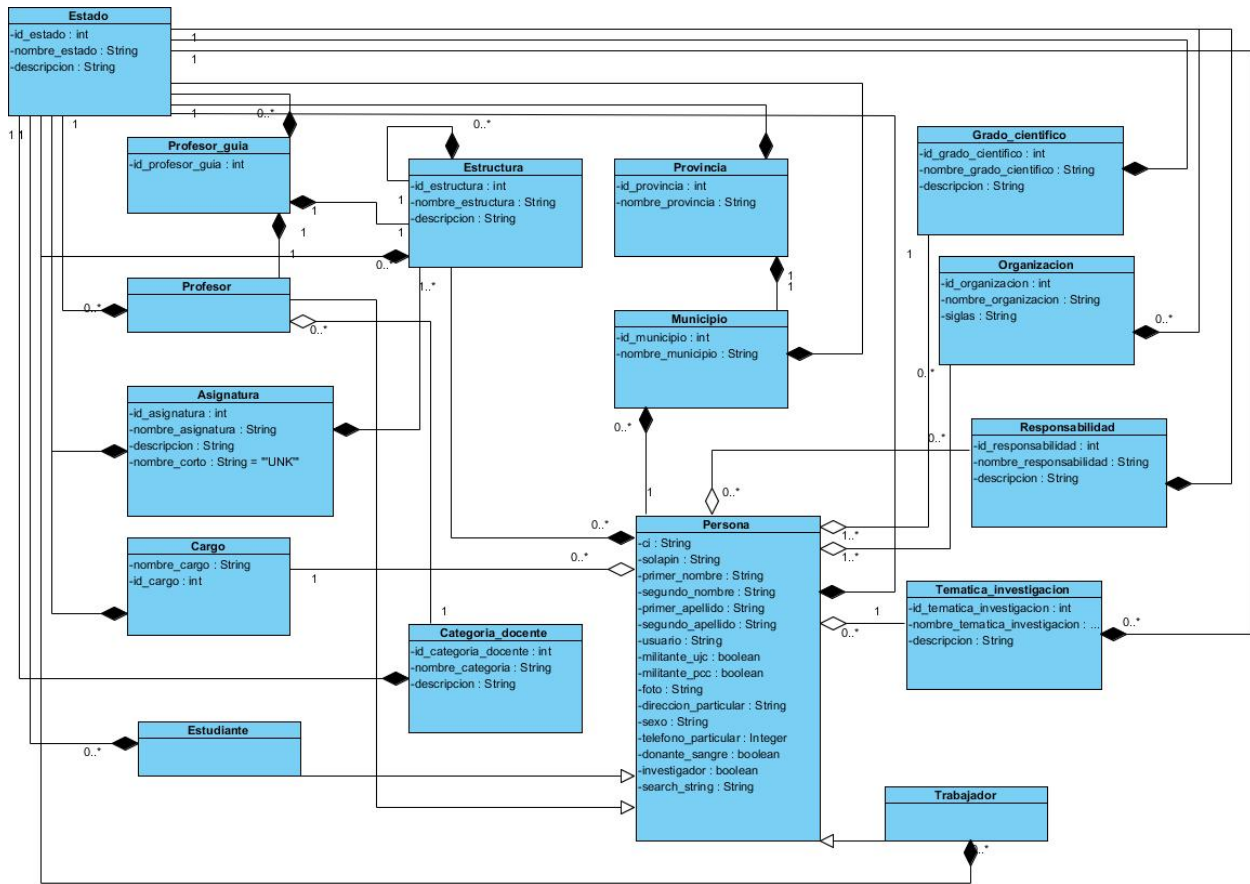


Diagrama 6 Muestra reducida del diagrama de clases de la solución.

### 2.5.5. Diseño de la Base de Datos.

A continuación se muestra una selección de las entidades fundamentales contenidas en el diagrama Entidad – Relación correspondiente a la aplicación. Con el color amarillo se muestran las entidades más importantes de la solución, mientras que con el color gris se representan las entidades de control necesarias para la realización de búsquedas de datos históricos. Con el color verde se resaltan las entidades consideradas nomencladores de la aplicación y por el último con el color naranja se simbolizan otras entidades existentes en el negocio.

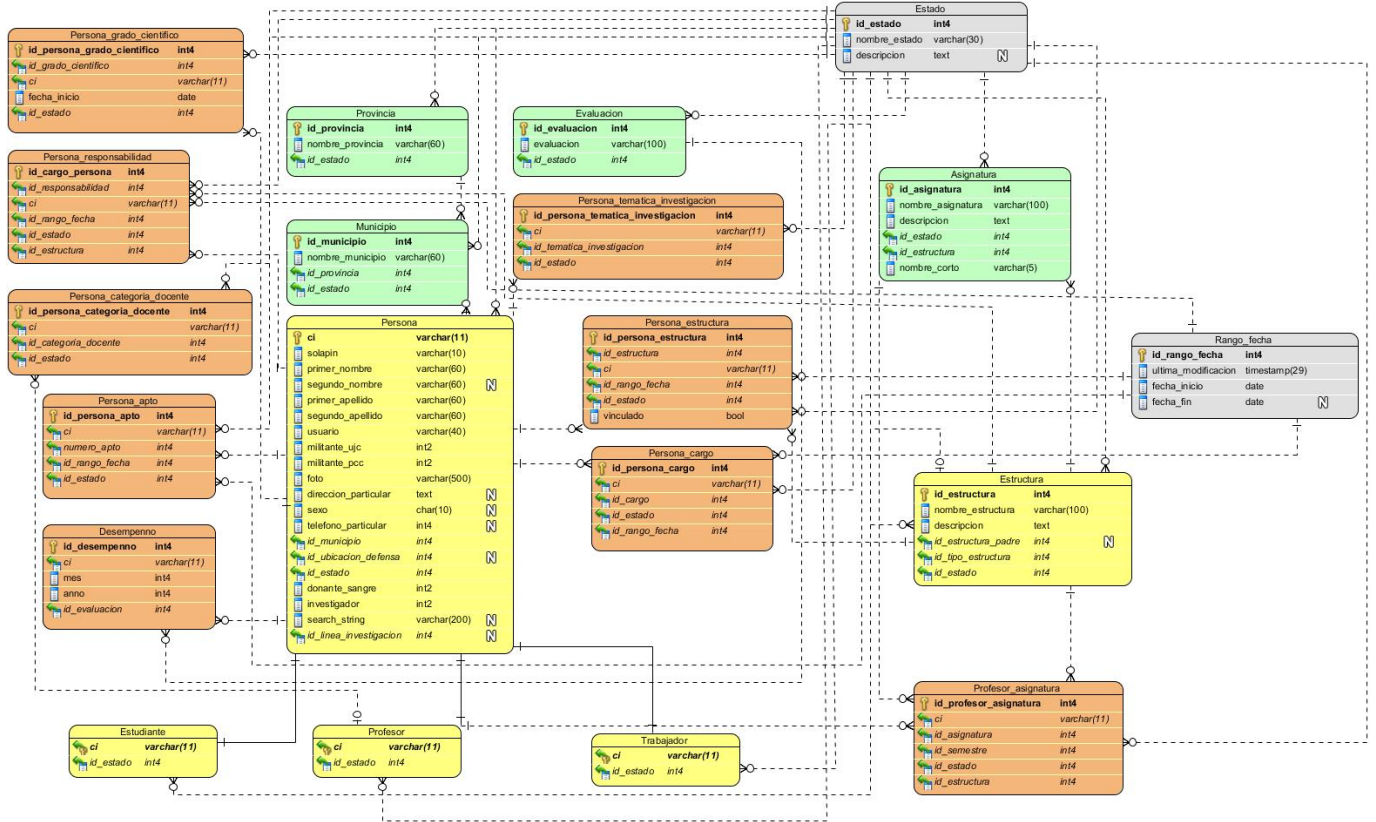


Diagrama 7 Muestra del Diagrama Entidad-Relación de la aplicación.

## 2.6. Implementación de la solución.

Partiendo del diseño realizado con anterioridad, comienza la implementación de la solución, que describe cómo los elementos del modelo de diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo con los nodos específicos en el modelo de despliegue.

Durante este flujo además se define la organización de la aplicación, en términos de los subsistemas o componentes por los que estará compuesto, así como se implementan los archivos del código fuente, binarios, programas ejecutables y otros a partir de los artefactos generados por el diseño. Por último en este flujo, se integrarán los resultados producidos por los desarrolladores individuales (o equipos) en un sistema ejecutable.

### 2.6.1. Estándar de codificación.

En la actualidad existen diferentes estándares de codificación cuya utilización favorece la comunicación fluida y directa entre los desarrolladores, permitiendo el aumento de la reutilización y el mantenimiento de los sistemas.

Para implementar la solución propuesta se hacen uso de los estándares establecidos por la plataforma SO3, además de lo que ya utiliza el Framework utilizado. Dichos estándares se describen a continuación:

Acciones (métodos o funcionalidades de la clase nombreDelModuloAction.class.php):

- Dentro de las especificaciones del Framework está que cada una de estas acciones debe comenzar con la palabra execute.
- Todos los nombres de acciones deben estar en la nomenclatura CamelCase comenzando por la palabra execute.
- En caso de ser acciones referentes a la modificación o remoción de una tupla de determinada tabla deben ser nombres específicos como executeModificarNombreDeTabla o executeEliminarNombreDeTabla.
- Los nombres de las acciones deben especificar con la menor cantidad de palabras cual es el objetivo de la acción, de ser posible estar en infinitivo.

Nombres de las clases:

- Los nombres de las clases deben estar expresados en notación UpperCamelCase.
- No se deben utilizar guiones bajos en su nombre.
- Deben expresar con claridad cuál es el alcance y la responsabilidad de la clase.
- Los nombres de las clases no deben estar atados a las clases de las que se deriva, cada clase debe tener un significado por ella misma, no en dependencia de la clase de la que deriva.

Nombres de las funciones:

- Todas las funciones definidas por los desarrolladores deben seguir la nomenclatura CamelCase.

Variables:

- Los nombres de las variables deben expresar claramente el contenido de la misma.

- Deben seguir la nomenclatura UpperCamelCase.
- Pueden estar referidas en singular o plural.
- Se definen al principio de las estructuras donde son utilizadas.
- En caso de que no se le asigne un valor inicial se deben inicializar con un valor que indique el tipo de dato más general al que debe pertenecer.
- De esta nomenclatura se exceptúan las variables generadas por el Framework.

### **2.6.2. Tratamiento de errores.**

Un aspecto importante a tener en cuenta a la hora de desarrollar un software es el tratamiento de errores, tanto los generados por los usuarios como los producidos por el sistema.

Las validaciones del negocio se encargan de controlar el flujo de los datos recibidos en el controlador para evitar la inconsistencia de la información almacenada en la base de datos.

Las validaciones realizadas en las vistas juegan un papel primordial en la entrada correcta de los datos a la aplicación. Tienen como función principal evitar que se introduzcan datos incorrectos en los diferentes campos de los formularios para impedir que datos inconsistentes o incorrectos lleguen al servidor. Con el objetivo de detectar y mostrar las alertas y mensajes de notificación de errores en la interfaz del sistema se utilizan las validaciones que proveen Ext JS así como las ventanas modales.

Por otro lado en cada acción implementada en Symfony se realiza un riguroso escrutinio de los valores de las variables que llegan a la funcionalidad en el servidor y en caso de detectarse errores son enviados en forma de cadena JSON a la interfaz para que sean notificados al usuario.

### **2.6.3. Diagrama de componentes.**

Este tipo de diagrama muestra el conjunto de componentes y sus relaciones entre sí, representando cómo un sistema de software es dividido en componentes así como las dependencias entre estos (Pressman, 2004). Permite además, describir la vista de implementación estática de un sistema.

Un componente se corresponde con una o más clases, interfaces o colaboraciones. A continuación se muestra el diagrama de componentes de la aplicación.

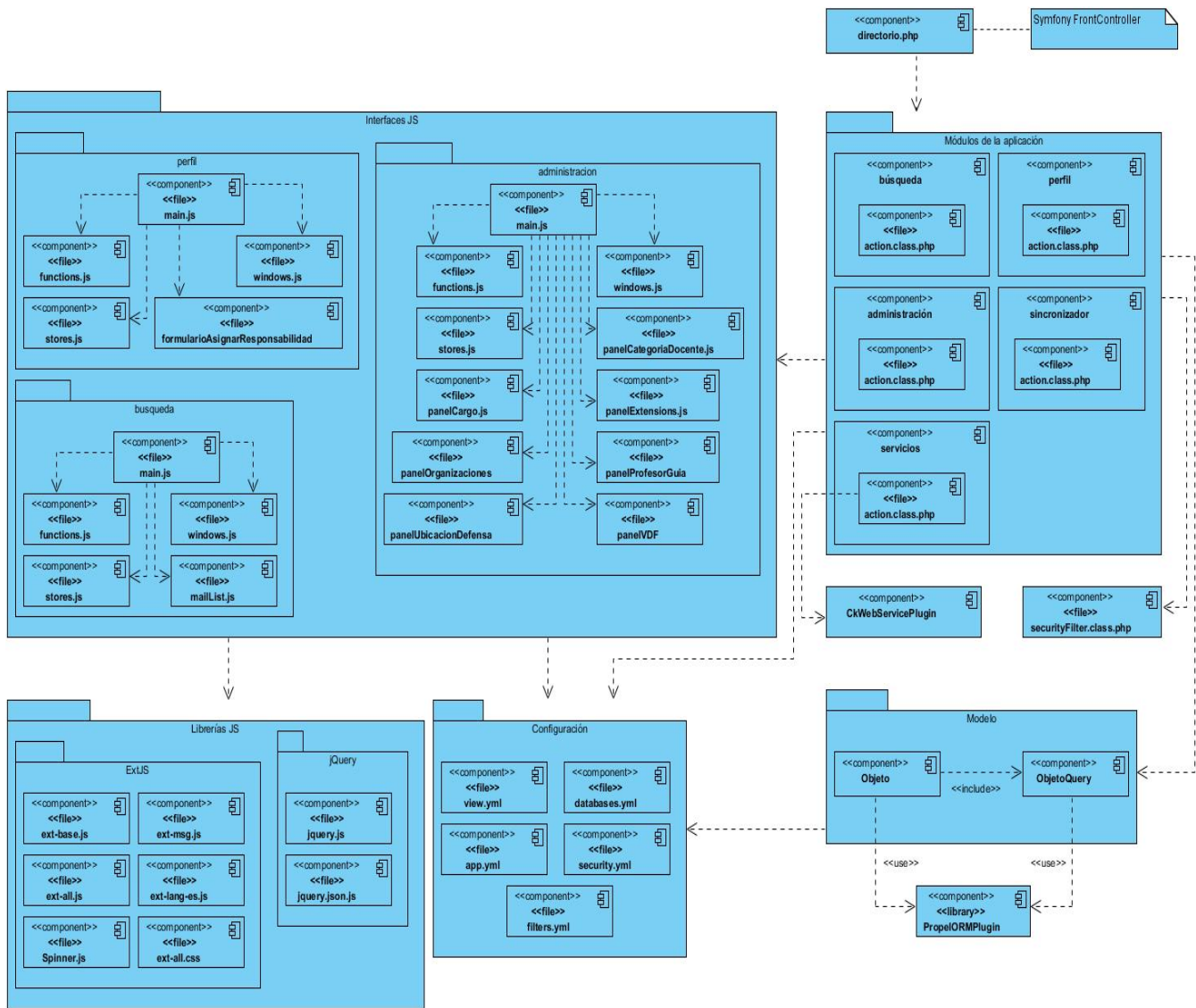


Diagrama 8 Diagrama de componentes de la aplicación.

### 2.6.4. 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.

Este tipo de diagramas muestran la configuración de los componentes de hardware, los procesos así como los elementos de procesamientos en tiempo de ejecución. En estos esquemas intervienen nodos, asociaciones de comunicación, componentes dentro de los nodos y objetos que se encuentran a su vez dentro de los componentes.



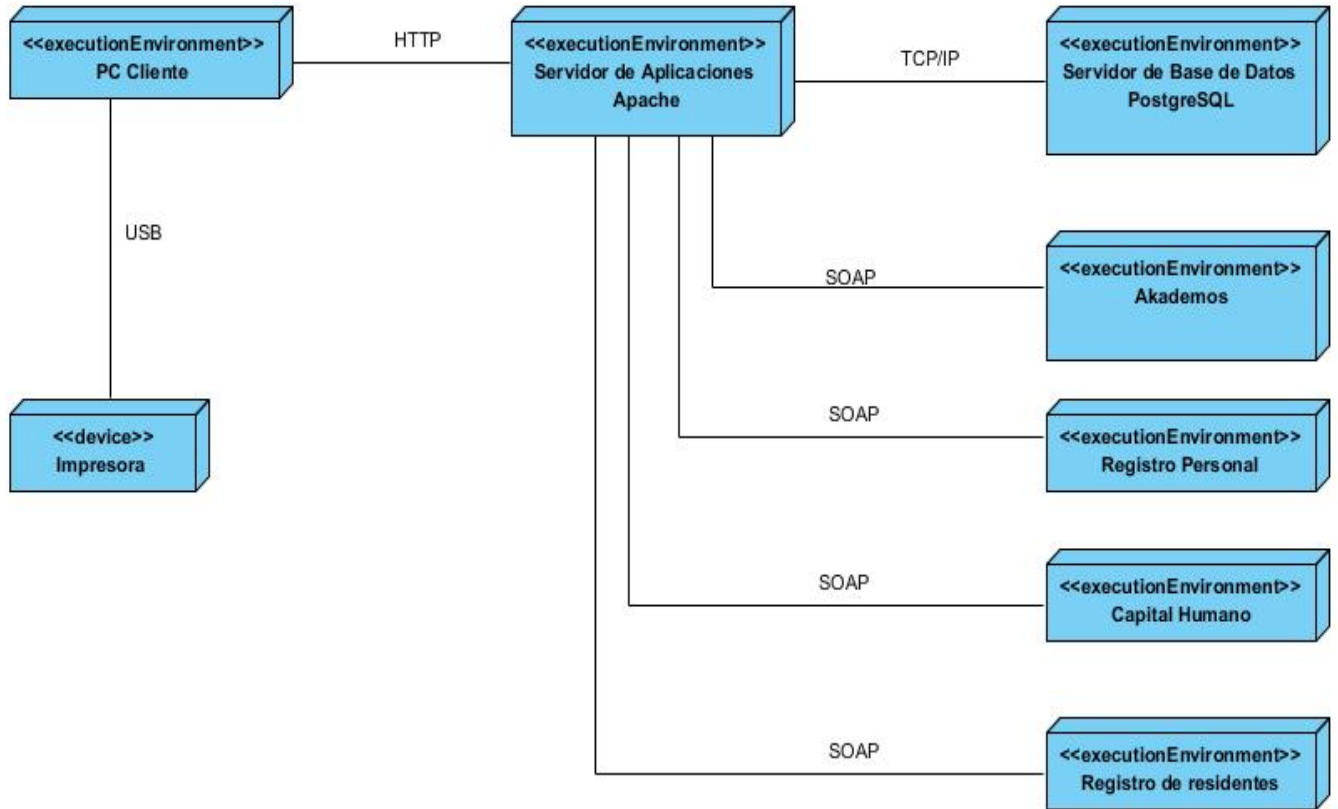


Diagrama 9 Diagrama de despliegue de la aplicación.

## 2.7. Conclusiones.

En este capítulo se describe brevemente el diseño del sistema y se presentan los artefactos más significativos generados durante la etapa de implementación de la aplicación. Se hace además especial énfasis en los prototipos de interfaz de usuarios, los patrones arquitectónicos y de diseño empleados, así como los estándares de codificación y técnicas de validación de errores seleccionados para la implementación de la solución. Por último se presentan algunos de los diagramas más importantes obtenidos del diseño realizado con anterioridad y de la implementación, tales como el modelo del dominio, diagrama de clases del diseño, síntesis del modelo de datos, diagrama de componentes y diagrama de despliegue.

# Capítulo 3 Validación de la solución propuesta

## 3.1. Introducción.

El desarrollo de un producto software es un proceso altamente complejo y no está exento a la existencia de errores, por lo que el desarrollo del mismo debe estar acompañado de actividades o subprocesos que garantice la calidad. Entre estas actividades podemos mencionar la gestión de riesgos y las pruebas.

Las pruebas de software son el proceso de identificar las fallas o defectos en el sistema y asegúrese de que el mismo cumple con los requisitos del cliente antes de lanzarlo al mercado. Con el fin de mejorar la calidad del software, ciertos protocolos y reglas necesitan ser establecido para complementar la eficiencia del software.

En este capítulo se realiza la validación a la solución propuesta en el capítulo anterior, a través del uso de diferentes técnicas de prueba, de manera que se pueda comprobar la eficiencia de las clases u operaciones utilizadas para dar respuesta a las necesidades de los clientes.

## 3.2. Pruebas de software

Las pruebas de software son una investigación orientada a proporcionar a los interesados información sobre la calidad del producto o servicio bajo prueba. Las pruebas de software también pueden proporcionar una visión objetiva e independiente del software para permitir a los desarrolladores apreciar y entender los riesgos del software puesto en práctica (Kaner, 2006).

Por lo general, las pruebas involucran operaciones del sistema evaluando los resultados bajo condiciones controladas, lo que hace que la realización de pruebas al software sea un factor de vital importancia. En pos de comprobar el estado de la calidad del software, se siguen las siguientes etapas:

- Verificación de la interacción de componentes.
- Verificación de la integración adecuada de los componentes.
- Verificación de que todos los requisitos se han implementado correctamente.
- Identificación de los defectos encontrados y aseguramiento de su corrección antes de entregar el software.
- Diseño de pruebas que sistemáticamente identifiquen diferentes clases de errores, con la menor cantidad de tiempo y esfuerzo.

### 3.2.1. Niveles de prueba.

En la actualidad existen varios niveles de prueba, cada uno de ellos se relaciona con determinado momento del ciclo de vida del software. A continuación se describen cada uno de estos niveles (Pressman, 2004).

#### 3.2.1.1. Pruebas Unitarias

En este nivel de pruebas se ejecutan fundamentalmente casos de prueba de caja blanca diseñados para

- Probar las estructuras de datos locales para asegurar que los datos conserven su integridad durante los pasos de ejecución del algoritmo.
- Probar las condiciones límites que aseguren el correcto funcionamiento del componente en los límites establecidos como restricciones de procesamiento.
- Ejercitar todos los caminos independientes de la estructura de control con el fin de asegurar que todas las sentencias del módulo o componente se ejecutan por lo menos una vez.

Los casos de prueba diseñados para ejecutar a este nivel debe descubrir errores como:

- Comparaciones ente tipos de datos distintos.
- Operadores lógicos o de procedencia incorrectos
- Igualdad esperada cuando los errores de precisión la hacen poco probable.
- Variables o comparaciones incorrectas.
- Terminación de bucles inexistente.
- Fallo de salida en iteraciones divergentes.
- Variables de bucles modificadas inapropiadamente.

En este contexto el uso de las pruebas de límites son de extrema utilidad, puesto que el software tiende a fallar frecuentemente al enfrentarse a situaciones extremas. La selección correcta de los juegos de datos que con los que se realizará la prueba constituye un paso primordial en esta etapa (Pressman, 2004).

#### 3.2.1.2. Pruebas de integración.

Las pruebas de integración es una técnica sistemática para construir la estructura del programa mientras que, al mismo tiempo, se llevan a cabo pruebas para detectar errores asociados con la interacción. El objetivo es seleccionar los módulos a los que se les han realizado las pruebas

de unidad y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño del software. En el proceso de integración puede derivarse la realización de los casos de uso del sistema, estos describen las formas de interacción entre las clases, los objetos y los componentes. (Pressman, 2004)

Estos tipos de casos de prueba por lo general son derivados de diagramas de secuencia en los que se describe una o varias secuencias del negocio. Al ejecutarse el caso de prueba, se crean trazas de ejecución que son comparadas con los diagramas de secuencia y en caso de no coincidir estos se está ante un defecto que puede convertirse en un error del software.

### **3.2.1.3. Pruebas de sistema.**

Las pruebas de sistemas caen fuera del ámbito del proceso de software y no las realiza únicamente el desarrollador del software. Sin embargo los pasos durante el diseño del software y durante la prueba pueden mejorar la posibilidad de éxito de las pruebas de sistema. Este tipo de pruebas se centran fundamentalmente en verificar la interacción de los actores con el sistema, por lo que a menudo los casos de prueba se obtienen a partir de las descripciones de los casos de uso, aunque también se le aplican prueba al sistema como un todo. (Pressman, 2004)

### **3.2.1.4. Pruebas de aceptación.**

La mayoría de los desarrolladores de software llevan a cabo un proceso denominado Alfa o Beta para descubrir errores solo el usuario final puede descubrir (Pressman, 2004). Las Pruebas Alfa consisten en llevar al cliente al entorno de desarrollo para probar el sistema. De esta forma el cliente trabaja en un entorno controlado y siempre tiene un experto a mano para ayudarlo a usar el sistema y para analizar los resultados.

Las Pruebas Beta por su parte, se realizan luego de culminadas satisfactoriamente las Pruebas Alfa. Se desarrollan en el entorno del cliente, es decir, en el lugar de trabajo para el que fue diseñado el sistema. De esta manera el cliente puede probar el producto a su manera, tratando siempre de encontrarle fallos. Las Pruebas de Aceptación del Cliente incluyen dos tipos de pruebas a dos niveles diferentes. Las Pruebas de Aceptación propiamente dichas y las Pruebas Piloto. Ambas se corresponden con los niveles de Pruebas Alfa y Pruebas Beta respectivamente (Cabrera y otros, 2008).

### 3.2.2. Tipos de pruebas.

#### 3.2.2.1. Pruebas de caja negra.

Las pruebas de caja negra son un método de ensayo en el que los datos de prueba se derivan de los requisitos funcionales especificados sin tener en cuenta la estructura final del programa. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos (Pressman, 2004).

Mientras más amplio sea el espectro de elementos de entrada para realizar la prueba, las probabilidades de encontrar problemas en el software aumentan y por lo tanto, será más confiable la calidad del software.

Este tipo de pruebas permite encontrar:

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

Para preparar los casos de prueba es necesario un número de datos que ayuden a la ejecución de estos casos y que permitan que el sistema se ejecute en todas sus variantes. Estos datos pueden ser válidos o no para el software y son seleccionados atendiendo a las especificidades de la funcionalidad a probar.

Para realiza estas pruebas existen varias técnicas:

- Partición de equivalencia: divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software. Esta técnica es muy efectiva puesto que permite examinar los valores válidos e inválidos de las entradas existentes en el producto, descubre de forma inmediata una clase de errores que, de otro modo, requerirían la realización de muchos casos antes de detectar el error genérico. La partición de equivalente se dirige a la definición de casos de pruebas que descubran clases de errores, reduciendo de este modo el número de pruebas a realizar.

- Análisis de los valores límites: prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.

### 3.2.2.2. Pruebas de caja blanca.

Contrariamente a las pruebas de caja negra, en las de caja blanca el software se ve como un cuadro blanco, o de vidrio, que permite ver la estructura y el flujo del software bajo prueba. Los planes de pruebas se hacen de acuerdo a los detalles de la implementación del software, tales como lenguaje de programación, la lógica y estilos. Los casos de prueba se derivan de la estructura del programa.

Estas pruebas se basan en el minucioso examen de los detalles procedimentales, donde se comprueban los caminos lógicos del software proponiendo casos de prueba que examinen que todas las condiciones y/o bucles estén correctas para determinar si el estado real coincide con el esperado o afirmado (Pressman, 2004).

Los objetivos fundamentales de estas pruebas son:

- Garantizar que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
- Ejercitar todas las decisiones lógicas en las vertientes verdadera y falsa.
- Ejecutar todos los bucles en sus límites operacionales.
- Ejercitar las estructuras internas de datos para asegurar su validez.

Debido a la disminución en un alto por ciento del número de errores existentes en los sistemas gracias a la aplicación de las pruebas de caja blanca, estas son consideradas como uno de los tipos de pruebas más importantes, por mejorar la calidad y confiabilidad del software analizado.

### 3.2.2.3. Pruebas de rendimiento.

Las pruebas de performance o rendimiento permiten conocer y mitigar los riesgos relacionados con el mal desempeño de las aplicaciones en los entornos de producción y realizar las correcciones necesarias antes de salir al mercado (CES, 2012).

Entre las acciones a realizar están:

- Cuantificación de la capacidad de la infraestructura.
- Validación de los requerimientos de rendimiento, la escalabilidad de las plataformas y del sistema a probar.

De esta manera, los desarrolladores pueden conocer qué cantidad de clientes simultáneos soporta su producto, con tiempos y datos razonables sobre la infraestructura y las plataformas propuestas. Asimismo, puede saber si es suficiente el hardware para soportar el nivel propuesto de transacciones y qué expectativa de crecimiento soporta. Es relevante la selección de la herramienta adecuada, la definición de los escenarios de prueba y la configuración del entorno.

Una vez ejecutadas las pruebas, es importante la evaluación de resultados para identificar los posibles problemas y las posibilidades de mejora sobre el desempeño del sistema.

La prueba de rendimiento se da durante todos los pasos del proceso de la prueba. Incluso al nivel de unidad, se debe asegurar el rendimiento de los módulos individuales a medida que se llevan a cabo las pruebas de caja blanca. Sin embargo, hasta que no estén completamente integrados todos los elementos del sistema no se puede asegurar realmente el rendimiento del mismo (Pressman, 2004).

#### **3.2.2.4. Pruebas de resistencia (stress)**

Durante los pasos de prueba anteriores, las técnicas de caja blanca y de caja negra daban como resultado la evaluación del funcionamiento y del rendimiento normal del programa. Las pruebas de resistencia están diseñadas para enfrentar a los programas con situaciones anormales (Pressman, 2004).

Una variante de la prueba de resistencia es una técnica denominada prueba de sensibilidad. En algunas situaciones (la más común se da con algoritmos matemáticos), un rango de datos muy pequeño dentro de los límites de una entrada válida para un programa puede producir un proceso exagerado e incluso erróneo o una profunda degradación del rendimiento. Esta situación es análoga a una singularidad en una función matemática. La prueba de sensibilidad intenta descubrir combinaciones de datos dentro de una clase de entrada válida que pueda producir inestabilidad o un proceso incorrecto.

### 3.3. Validación de la implementación.

#### 3.3.1. Evaluación de la ejecución de las pruebas de caja negra.

A continuación se describen los casos de prueba para el módulo de búsquedas de la aplicación. El resto de los diseños de los casos de prueba de la solución puede encontrarse en los documentos entregables anexados al presente trabajo de diploma.

#### Módulo Buscador

Las pruebas realizadas a este módulo son:

- EC 1.1 Buscar persona (simple)
- EC 1.2 Buscar persona (avanzada).
- EC 1.3. Mostrar ficha de la persona.
- EC 1.4. Exportar lista de correo.

**Caso de prueba del EC 1.1:** Buscar persona (simple).

#### Descripción:

El caso de uso se inicia cuando el usuario necesita realizar una búsqueda simple de personas en el directorio. Selecciona la opción en el menú principal de la plataforma y hace uso de la barra de búsqueda de la aplicación.

#### Flujo central:

- El usuario selecciona la opción de buscar en el menú principal de la plataforma.
- El sistema muestra la interfaz del buscador simple.
- El usuario llena los datos en la barra de búsqueda.
- El usuario presiona el botón Buscar o la tecla Intro.
- El sistema muestra los resultados de la búsqueda.

Clases válidas	Clases no válidas	Resultados de la prueba	Resultados de la prueba	Observaciones	Cumplimiento
El usuario selecciona la opción buscar del menú principal de la plataforma.	No se muestra la opción de buscar. No se muestra la	Muestra la interfaz del buscador	Satisfactorias	La operación se realizó correctamente	100%



	interfaz del buscador.				
El usuario llena los datos en la barra de búsqueda y ejecuta la búsqueda.	El sistema no muestra los resultados de la búsqueda	El sistema muestra los resultados de la búsqueda	Satisfactorias	La operación se realizó correctamente	100 %

### **Caso de prueba del EC 1.2:** Buscar persona (avanzado).

#### **Descripción:**

El caso de uso se inicia cuando el usuario necesita realizar una búsqueda avanzada de personas en el directorio. Selecciona la opción Búsqueda avanzada en la barra de herramientas del buscador y hace uso de los criterios de búsqueda de la aplicación.

#### **Flujo central:**

- El usuario selecciona la opción de búsqueda avanzada en la barra de herramientas de la aplicación.
- El sistema muestra la interfaz del buscador avanzado.
- El usuario selecciona los criterios de la búsqueda.
- El usuario presiona el botón Buscar o la tecla Intro.
- El sistema muestra los resultados de la búsqueda.

#### **Condiciones de ejecución:**

- Ocupación (en caso de que existan personas en el sistema y sea seleccionado este criterio).
- Título (en caso de que existan trabajadores en el sistema y sea seleccionado este criterio).
- Categoría docente (en caso de que existan trabajadores docentes en el sistema y sea seleccionado este criterio).
- Provincia (en caso de que existan personas en el sistema y sea seleccionado este criterio).
- Municipio (en caso de que existan personas en el sistema y sea seleccionado este criterio).

- Centro (en caso de que existan personas en el sistema y sea seleccionado este criterio).
- Proyecto (en caso de que existan personas en el sistema y sea seleccionado este criterio).
- Área (en caso de que existan personas en el sistema y sea seleccionado este criterio).
- Género (en caso de que existan personas en el sistema y sea seleccionado este criterio).
- Grupo (en caso de que existan estudiantes en el sistema y sea seleccionado este criterio).
- Cargo (en caso de que existan trabajadores en el sistema y sea seleccionado este criterio).
- Asignatura que imparte (en caso de que existan trabajadores docentes en el sistema y sea seleccionado este criterio).
- Responsabilidad (en caso de que existan personas en el sistema y sea seleccionado este criterio).
- Rol (en caso de que existan personas en el sistema y sea seleccionado este criterio).

Clases válidas	Clases no válidas	Resultados de la prueba	Resultados de la prueba	Observaciones	Cumplimiento
El usuario selecciona la opción buscar del menú principal de la plataforma.	No se muestra la opción de buscar. No se muestra la interfaz del buscador.	Muestra la interfaz del buscador	Satisfactorias	La operación se realizó correctamente	100%
El usuario llena los datos en la barra de búsqueda y ejecuta la búsqueda.	El sistema no muestra los resultados de la búsqueda	El sistema muestra los resultados de la búsqueda	Satisfactorias	La operación se realizó correctamente	100 %

**Caso de prueba del EC 1.3:** Mostrar ficha de la persona.

**Descripción:**

El caso de uso se inicia cuando el usuario una vez realizada la búsqueda (simple o avanzada) desea obtener información más detallada de las personas que devolvió la búsqueda.

**Flujo central:**

- El usuario selecciona la opción Ver ficha contenida en cada registro.
- El sistema muestra la interfaz con los datos extendidos de la persona.

**Condiciones de ejecución:**

- El usuario debe haber realizado una búsqueda y esta debe mostrar al menos un resultado.

Clases válidas	Clases no válidas	Resultados de la prueba	Resultados de la prueba	Observaciones	Cumplimiento
El usuario selecciona la opción Ver ficha.	No se muestra la interfaz con los datos extendidos de la persona	Muestra la interfaz con los datos extendidos de la persona	Satisfactorias	La operación se realizó correctamente	100%

**Caso de prueba del EC 1.4:** Exportar lista de correo.

**Descripción:**

El caso de uso se inicia cuando el usuario, una vez realizada la búsqueda (simple o avanzada) desea obtener una lista de correo que incluya las personas que devolvió la búsqueda.

**Flujo central:**

- El usuario selecciona la opción Exportar lista de correo de la barra de herramientas de la aplicación.
- El sistema muestra la interfaz que permite exportar la lista de correo.
- El usuario inserta el nombre que desea para la lista y presiona el botón Aceptar.
- El sistema permite descargar la lista de correo generada.

**Condiciones de ejecución:**

- El usuario debe haber realizado una búsqueda y esta debe mostrar al menos un resultado.

Clases válidas	Clases no válidas	Resultados de la prueba	Resultados de la prueba	Observaciones	Cumplimiento
El usuario selecciona la opción Exportar lista de correo	No se muestra la interfaz Exportar lista de correo	Muestra la interfaz Exportar lista de correo	Satisfactorias	La operación se realizó correctamente	100%
El usuario inserta el nombre de la lista y presiona Aceptar	El sistema no permite descargar la lista de correo generada	El sistema permite descargar la lista de correo generada	Satisfactorias	La operación se realizó correctamente	100%

### 3.3.2. Conclusiones parciales sobre las pruebas de caja negra.

A partir de la realización de las pruebas de caja negra, como parte de las pruebas funcionales, se demostró el correcto funcionamiento de la solución a través del cumplimiento de los requerimientos funcionales planteados por los clientes. La realización de estas pruebas permitió también encontrar clases no válidas que no se correspondían con las esperadas. Dichas clases no válidas fueron analizadas y revisadas para garantizar el cumplimiento con los requerimientos del cliente.

### 3.3.3. Evaluación de la ejecución de las pruebas de resistencia y rendimiento.

Las pruebas de resistencia y rendimiento<sup>19</sup> por lo general son realizadas por programas encargados de simular condiciones extremas para el software a probar, aunque también pueden ser realizadas manualmente por miembros del equipo u otras personas seleccionadas mediante diferentes métodos.

La solución implementada, por las características que posee y el negocio al que está enfocado, recibirá una notable cantidad de peticiones por parte de los usuarios o sistemas que usen los servicios de la misma. Por esta razón se hace necesario realizar un estudio de la resistencia y rendimiento de las diferentes funcionalidades del sistema.

Inicialmente se efectuaron pruebas con los miembros del equipo de desarrollo, pero a pesar de obtener resultados satisfactorios, se hizo necesaria la realización de un estudio más profundo en aras de reconocer las debilidades de la solución ante un flujo mayor de peticiones.

En vistas a encontrar una herramienta que fuera capaz de dar respuesta a esta situación se realizó una investigación encontrando tres herramientas. Estas herramientas cumplen con las

<sup>19</sup> En otras bibliografías pueden encontrarse como pruebas de carga y estrés.

características necesarias para ejecutar las pruebas de carga y estrés o apoyar el proceso de pruebas. Una breve descripción de ellas se muestra a continuación (Almenares, 2009).

- **QALoad.**

Desarrollado por la empresa Compuware es una herramienta de automatización de pruebas de carga para web, Java, .NET, aplicaciones ERP y CRM. Simula miles de usuarios desempeñando transacciones de negocio claves de una aplicación para asegurar su desempeño y escalabilidad previa a su puesta en producción. Con esta herramienta, los equipos de prueba pueden rápidamente detectar problemas, optimizar el desempeño de los sistemas y ayudar a asegurar un despliegue de aplicaciones exitoso.

- **LoadRunner.**

Es una herramienta para realizar pruebas de carga de Mercury Interactive que permite prever el comportamiento y el rendimiento del sistema. Además, permite poner a prueba toda la infraestructura corporativa para identificar y aislar los posibles problemas mediante la simulación de la actividad de miles de usuarios.

Realiza pruebas en toda la infraestructura corporativa, que comprende las soluciones e-business, ERP, CRM y las aplicaciones personalizadas, simulando la actividad de miles de usuarios, con lo que los equipos de desarrollo de aplicaciones y sitios Web pueden mejorar el rendimiento de las aplicaciones.

Es la herramienta de pruebas de carga más escalable que permite simular la actividad de miles de usuarios con los mínimos recursos de hardware. Se integra a la perfección con las herramientas de gestión del rendimiento de Mercury Interactive. Los mismos scripts creados durante las pruebas pueden volverse a utilizar para monitorizar la aplicación una vez terminada su implantación. Presenta una interfaz API abierta, con la que los usuarios y otros fabricantes pueden integrar LoadRunner en sus propios entornos.

- **JMeter**

Es una herramienta Java desarrollada dentro del proyecto Jakarta, que permite realizar Pruebas de Rendimiento y Pruebas Funcionales sobre Aplicaciones Web. Permite realizar pruebas Web clásicas, pero también permite realizar test de FTP, JDBC, JNDI, LDAP, SOAP/XML-RPC, y servicios web. Permite la ejecución de pruebas distribuidas entre distintos ordenadores, para realizar pruebas de rendimiento.

Además es posible activar o desactivar una parte del test, lo que es muy útil cuando se está desarrollando un test largo, y se desea deshabilitar ciertas partes iniciales que sean muy pesadas o largas. Tiene la forma de generar un caso de prueba a través de una navegación de usuario.

Se decidió utilizar para las pruebas de carga y estrés la herramienta **JMeter** por las funcionalidades que presenta, ser un software libre y gratis. Además se pueden utilizar muchas de sus funcionalidades para probar aspectos de las pruebas de Disponibilidad y Red.

**JMeter** se clasifica como una herramienta dinámica de medida por las funcionalidades que incorpora, como la manipulación de datos de entrada para la ejecución de las pruebas y la comparación de los resultados para la evaluación realizados aplicando el método de caja negra. Además se clasifica por su especialización en una herramienta de grabación y reproducción.

La utilización del **JMeter** supone un 95 % de tiempo menos para la realización de estas pruebas. La mayor inversión de tiempo que se necesita para la realización de las pruebas es la fase de estudio de los casos de uso críticos en la aplicación Web y la elaboración del plan de pruebas en la herramienta. Permite almacenar los resultados de la prueba y generar gráficos que representan los aspectos que se han probado.

#### **3.3.3.1. Selección de la muestra de las funcionalidades del sistema.**

Como parte de la elaboración del *plan de pruebas de resistencia y rendimiento*<sup>20</sup>, se realiza la selección de las funcionalidades del sistema a ser analizadas. A continuación se relacionan las funcionalidades seleccionadas para ser parte de la muestra a estudiar con la herramienta JMeter, junto con el criterio de selección:

- Buscador de personas (funcionalidad crítica).
- Obtener alumnos ayudantes (seleccionada aleatoriamente).
- Asignar asignaturas (funcionalidad relevante).
- Modificar perfil (funcionalidad crítica).
- Obtener Artistas Aficionados (seleccionada aleatoriamente).
- Cargar áreas (funcionalidad relevante).
- Obtener ficha (funcionalidad crítica).

---

<sup>20</sup> El plan de pruebas de resistencia y rendimiento es un entregable del presente trabajo de diploma y se anexa al mismo con el nombre **005\_Plan de pruebas de resistencia y rendimiento**.

Las características de la estación de trabajo en las que se realizarán las pruebas de rendimiento y resistencia son las siguientes:

- Memoria de 1 GB RAM DDR2.
- Microprocesador Intel™ Dual Core™ 2.2 GHz.
- Disco duro de 160 GB.
- Sistema operativo Linux Mint 12 con entorno de escritorio Cinnamon.
- Servidor de aplicaciones Apache 2.2.
- Lenguaje de programación de lado del servidor PHP 5.3.6.
- Gestor de base de datos PostgreSQL 9.1.
- Versión de JRE<sup>21</sup> para JMeter: 7

### 3.3.3.2. Ejecución de las pruebas.

Las primeras pruebas se iniciaron con una muestra de 4 peticiones las cuales arrojaron como resultados un 100% de éxito en las peticiones así como un promedio de 3 713 milisegundos (ms) de tiempo de respuesta por funcionalidad. A continuación se muestran tablas y gráficas que complementan los datos obtenidos en las pruebas.

Funcionalidad	Peticiones <sup>22</sup>	Promedio <sup>23</sup> (ms)	Media <sup>24</sup> (ms)	Mínima <sup>25</sup> (ms)	Máximo <sup>26</sup> (ms)
Buscador de personas	4	1195	1344	984	1384
Obtener Alumnos Ayudantes	4	20559	20990	19293	21729
Asignar asignatura	4	654	719	451	803
Modificar perfil	4	897	844	616	1507
Obtener Artistas Aficionados	4	922	977	753	1064
Cargar Áreas	4	509	481	420	675
Obtener ficha	4	1258	1504	829	1606
<b>TOTAL</b>	<b>28</b>	<b>3713</b>	<b>977</b>	<b>420</b>	<b>21729</b>

Tabla 2 Resultados de las pruebas de carga y estrés con 4 peticiones

La realización de las pruebas de resistencia y rendimiento arrojó como resultado la existencia de una funcionalidad (Obtener Alumnos Ayudantes) cuyo tiempo de respuesta supera con

<sup>21</sup> Siglas en inglés para Entorno de Ejecución de Java (Java RuntimeEnvironment).

<sup>22</sup> Cantidad de peticiones concurrentes realizadas por el software.

<sup>23</sup> Promedio de tiempo de respuesta de las peticiones realizadas expresado en milisegundos.

<sup>24</sup> Media del tiempo de respuesta de las peticiones realizadas expresada en milisegundos.

<sup>25</sup> Menor tiempo de respuesta expresado en milisegundos registrado por funcionalidad.

<sup>26</sup> Mayor tiempo de respuesta expresado en milisegundos registrado por funcionalidad.

creces la media registrada por las demás funcionalidades de la aplicación. Gracias al estudio y análisis de estas pruebas, fue posible corregir y rediseñar el flujo de peticiones similares afectadas por el tiempo en que demoraban en brindar una respuesta.

Sin embargo, teniendo en cuenta que la funcionalidad afectada forma parte del módulo de gestión de información por áreas, esta será solicitada por solo un usuario y en determinados momentos del curso y además el flujo que sigue actualmente es un requerimiento del cliente por lo que se determina mantener el procedimiento actual de la funcionalidad pero optimizar las consultas a las base de datos y el procesamiento en el lado del servidor.

Por otro lado existen un conjunto de funcionalidades identificadas como críticas para la aplicación de cuyo correcto funcionamiento depende la estabilidad de la solución de forma integral. Por tratarse de funcionalidades que recibirán gran cantidad de peticiones se decidió diseñar un nuevo caso de prueba que simulara una mayor cantidad de usuarios conectados, en este caso 20.

Al efectuarse la prueba se obtuvo como resultado un 100% de respuestas satisfactorias con tiempo de respuesta promedio de 380 ms, el cual se puede calificar de satisfactorio. A continuación se muestran tablas y gráficas que complementan los datos obtenidos en las pruebas.

Funcionalidad	Peticiones	Promedio (ms)	Media (ms)	Mínima (ms)	Máxima (ms)
Buscador de personas	20	532	422	415	2472
Modificar perfil	20	314	252	244	1169
Cargar Áreas	20	221	217	214	280
Obtener ficha	20	454	447	445	597
<b>TOTAL</b>	<b>80</b>	<b>380</b>	<b>418</b>	<b>214</b>	<b>2472</b>

Tabla 3 Resultados de las pruebas de carga y estrés a funcionalidades críticas con 20 peticiones.



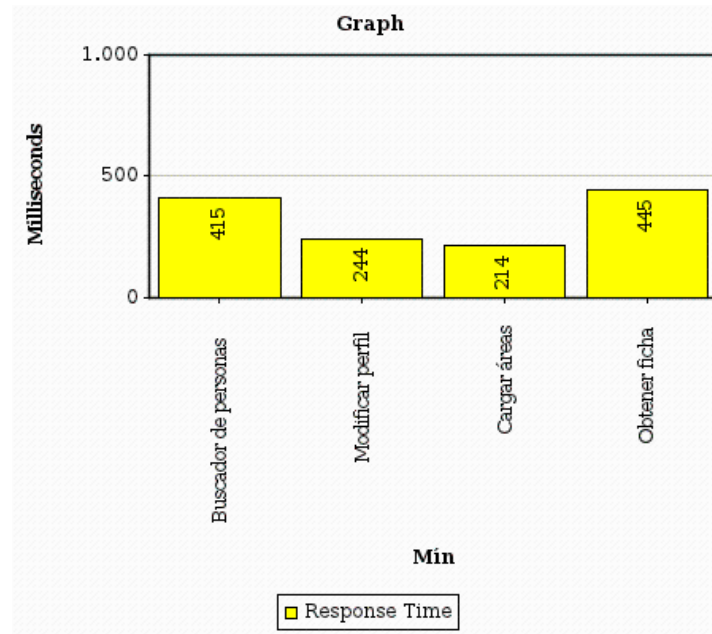


Figura 1 Tiempo de respuesta mínimo para funcionalidades críticas con 20 peticiones

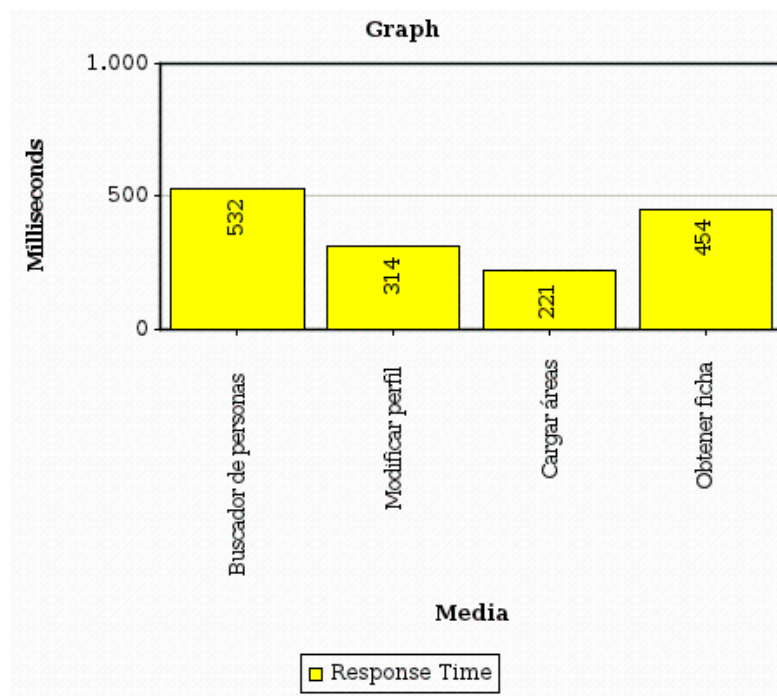


Figura 2 Media del tiempo de respuesta para funcionalidades críticas con 20 peticiones

Los resultados obtenidos en estas pruebas son satisfactorios. El 100% de las funcionalidades dio respuesta a la petición en menos de 1.5 segundos lo cual evidencia la optimización realizada en algunas funcionalidades y el correcto funcionamiento en general de la aplicación.

No obstante estos resultados se decidió aumentar la carga de usuarios y así verificar el desenvolvimiento del sistema en condiciones de estrés. Para ello se rediseñó el plan de

pruebas para trabajar con 80 peticiones arrojando como resultado un 100% de éxito en las llamadas así como un tiempo de respuesta promedio de 723 ms. A continuación se muestran tablas y gráficas que complementan los datos obtenidos en las pruebas.

Funcionalidad	Peticiones	Promedio (ms)	Media (ms)	Mínima (ms)	Máxima (ms)
Buscador de personas	80	878	890	480	1074
Modificar perfil	80	517	519	368	757
Cargar Áreas	80	442	444	307	577
Obtener ficha	80	1057	1073	640	1316
<b>TOTAL</b>	<b>320</b>	<b>723</b>	<b>648</b>	<b>307</b>	<b>1316</b>

Tabla 4 Resultados de las pruebas de carga y estrés a funcionalidades críticas con 20 peticiones.

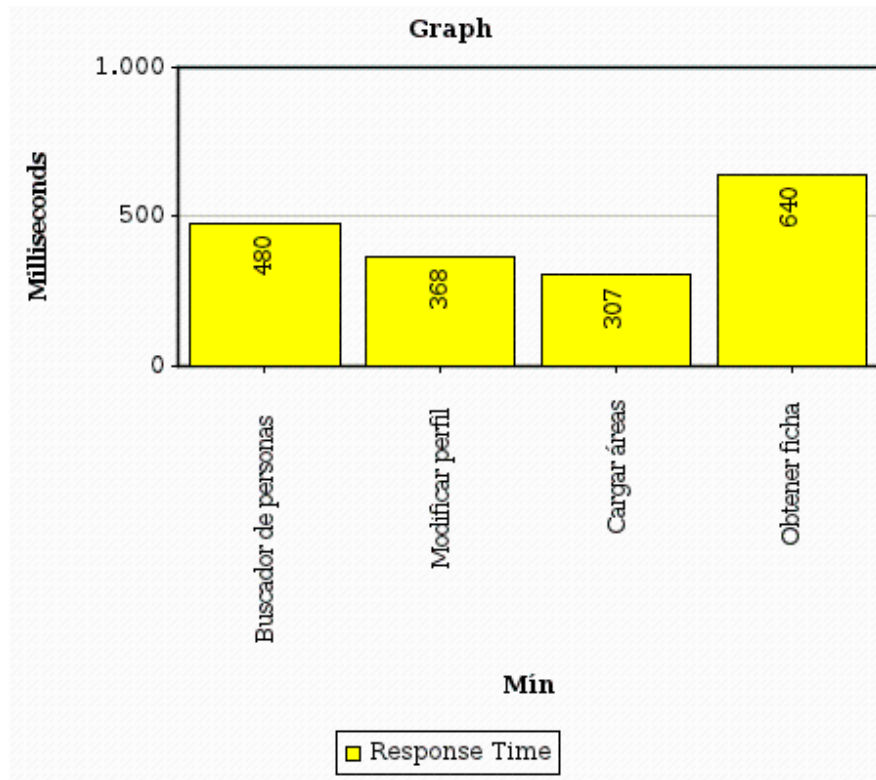


Figura 3 Tiempo de respuesta mínimo para funcionalidades críticas con 80 peticiones

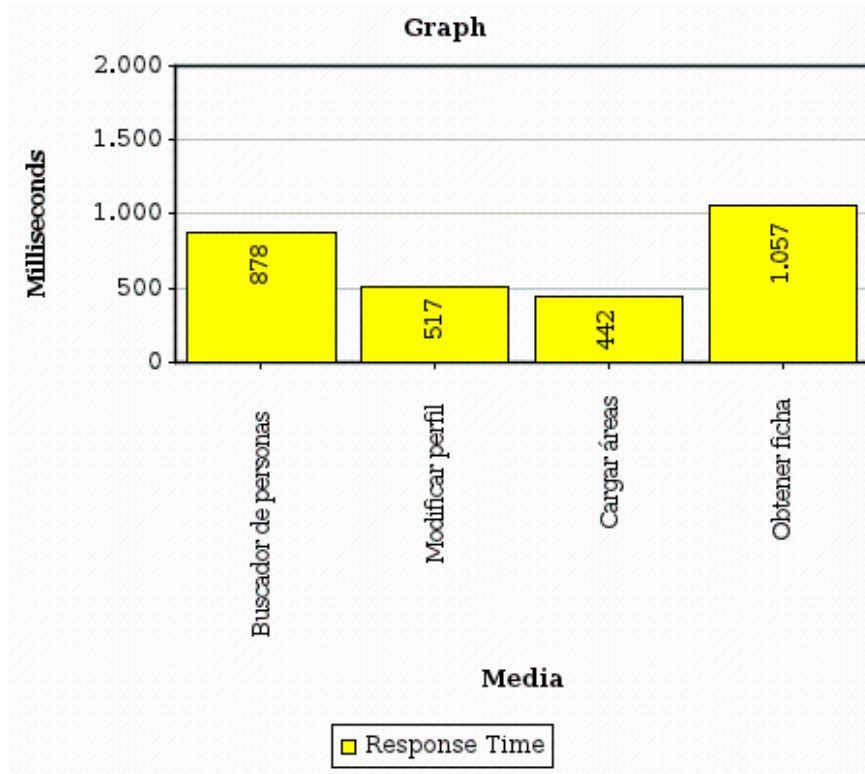


Figura 4 Tiempo medio de respuesta para funcionalidades críticas con 80 peticiones

### 3.4. Conclusiones

En este capítulo se realizó un estudio de los principales tipos y métodos de pruebas que se llevan a cabo durante el ciclo de vida del software. Se describieron las pruebas de caja negra efectuadas al escenario Buscador. Dichas pruebas corroboraron el cumplimiento de los requerimientos planteados por los clientes así como la obtención de los datos que se esperaban como respuesta de la aplicación. Por otro lado, para realizar las pruebas de resistencia y rendimiento fue necesario estudiar las principales herramientas existentes en el mercado, seleccionándose JMeter como software para medir los niveles de carga y estrés que soporta la aplicación.

Se identificaron las funcionalidades críticas de la solución con el objetivo de chequear su funcionamiento bajo condiciones de estrés, llegándose a obtener resultados satisfactorios. Además otras funcionalidades también fueron analizadas, siempre teniendo en cuenta las características del servidor donde fueron probadas las funcionalidades, lo que permitió establecer estimados para un servidor de producción y llegar a la conclusión de que el sistema funciona correctamente y está listo para su implantación.

## Conclusiones

Al finalizar el presente trabajo de diploma se logró cumplir de manera satisfactoria los objetivos planteados al inicio de la misma, obteniéndose como resultado directo el Subsistema de Gestión de Información Personal perteneciente al núcleo de la Plataforma Informativa SO3. Para el desarrollo de dicha solución fue necesaria la realización de un profundo estudio del estado del arte de las principales soluciones informáticas existentes para la gestión de los recursos humanos, la cual permitió establecer diferencias, fortalezas, debilidades así como puntos en común entre las mismas.

El estudio brindó como resultado que no es factible utilizar ninguno de los sistemas analizados en la Plataforma Informativa SO3, teniendo en cuenta que la mayoría no cumple con las especificaciones definidas en dicha plataforma y que son desarrolladas o son herramientas privativas. Una vez concluido el estudio del estado del arte se procedió a la selección de las herramientas, tecnologías y metodologías a utilizar para el desarrollo de la solución informática.

Posteriormente se identificaron las principales funcionalidades de la solución, determinándose los diferentes módulos por los que estaría compuesta la misma. Teniendo como base el análisis hecho con anterioridad, se analizaron y seleccionaron los patrones de diseño a utilizar en la solución planteada y se diseñaron los diagramas establecidos en el flujo de implementación que dieron paso a la programación de los componentes que forman parte del subsistema.

El correcto funcionamiento de la solución desarrollada fue validado mediante la realización de pruebas funcionales, de rendimiento y de resistencia; chequeándose no solo el cumplimiento de los requerimientos planteados por los clientes sino también el comportamiento de la aplicación bajo altos niveles de carga y estrés. La realización de estas pruebas permitió además encontrar y corregir funcionalidades defectuosas en la aplicación.

El desarrollo del Subsistema de Gestión de Información Personal de la Facultad 3 aportará una mayor eficiencia y confiabilidad en la gestión y control de los datos personales asociados a los recursos humanos de dicha facultad, además de brindar servicios a otras aplicaciones a través de una interfaz con grandes volúmenes de información disponible. La solución desarrollada contribuirá notablemente no sólo a la toma de decisiones por parte de los directivos de la facultad sino también a una eficiente gestión de la información a todos los niveles de la institución.

## Recomendaciones

Los objetivos trazados en el presente trabajo de diploma fueron alcanzados satisfactoriamente; sin embargo se recomienda:

- Desplegar la aplicación en la Plataforma Informativa SO3.
- Desarrollar nuevos servicios para el sistema en la medida que surjan necesidades para otros sistemas de la Plataforma Informativa SO3.
- Definir un plan de capacitación para los usuarios que van a administrar la información por las diferentes áreas de la facultad.
- Continuar el estudio de los procesos asociados a la gestión de información personal de los recursos humanos en la Universidad de las Ciencias Informáticas.

## Referencias bibliográficas

**Oracle Corporation. 2012.** MySQL The world's most popular open source database. *Why MySQL?* [Online] 2012. [Cited: Enero 11, 2012.] <http://www.mysql.com/why-mysql/>.

**Alarcon Aguin, José Manuel. 2010.** *Tecnologías ASP.net 4.0.* s.l. : Krasis Press, 2010.

**Appelo, Jurgen. 2008.** Noop. *The Definitive List of Software Development Methodologies.* [Online] 2008. [Cited: enero 9, 2012.] <http://www.noop.nl/2008/07/the-definitive-list-of-software-development-methodologies.html>.

**ASSETS. 2004.** ASSETS. *Sistema de Gestión Integral.* [Online] 2004. [Cited: 12 9, 2011.] <http://assets.co.cu/assets.asp>.

**Bloomsbury. 2002.** *The Ultimate Resource.* Londres : Bloomsbury Publishing, 2002.

**Borrego, Daniel. 2009.** Herramientas Pymes. *Gestión del Capital Humano: OrangeHRM.* [Online] 2009. [Cited: 12 9, 2012.] <http://www.herramientasparapymes.com/gestion-del-capital-humano-orangehrm/>.

**Business Software. 2011.** *Top 10 Human Capital Management Software Vendors.* 2011.

**Cabrera González, Miguel P., et al. 2011.** *Sistema económico integrado Versat - Sarasola.* Santa Clara : s.n., 2011.

**Castledine, Earle and Sharkie, Craig . 2010.** *jQuery: Novice to Ninja.* Cambridge : SitePoint, 2010. ISBN 978-0-9805768-5-6.

**CES. 2012.** CES. *Testing de performance, carga, stress y volumen.* [Online] 2012. [Cited: Abril 10, 2012.] <http://www.ces.com.uy/index.php/servicios/testing-independiente/testing-de-performance>.

**Christiansen, Simon. 2012.** *Database Management System (DBMS).* [Online] 2012. [Cited: Enero 11, 2012.] <http://searchsqlserver.techtarget.com/definition/database-management-system>.

**CodeIgniter. 2012.** CodeIgniter. [Online] 2012. [Cited: Enero 10, 2012.] <http://codeigniter.com/>.

**Dávalos, Gabriel. 2009.** En vivo. [Online] 2009. <http://envivo.cu>.

**Earle Castledine y Craig Sharkie. 2010.** *jQuery: Novice to Ninja.* Cambridge : SitePoint, 2010. ISBN 978-0-9805768-5-6.

**Eclipse Foundation. 2012.** Eclipse: The Eclipse Foundation. *About the Eclipse Foundation.* [Online] 2012. [Cited: Enero 11, 2012.] <http://www.eclipse.org/org/>.

**Fajardo Ugarte, Jorge. 2008.** *BPMN: estandar para modelamiento de procesos.* 2008.

**FAO. 2009.** *Publicaciones de Medio Ambiente, Cambio Climático y Bioenergía.* 2009. Vol. 8.

**Heathfield, Susan M. 2012.** *Five Steps to a Human Resources Software Technology System.* Boston, MA : s.n., 2012, American Society for Training and Development.

**Frederick, Shea and Ramsay, Colin. 2008.** *Learning Ext JS.* Birmingham : Packt Publishing Ltd., 2008. ISBN 978-1-847195-14-2.

- HR Software Comparison. 2011.** HR Software Comparison. [Online] 2011. [Cited: 12 9, 2011.] <http://www.comparehrsoftware.com/compare-vendors/>.
- Jacobson, I. 1998.** *Applying UML in The Unified Process*. 1998.
- Kaner, Cem. 2006.** *Exploratory Testing*. Quality Assurance Institute Worldwide Annual Software Testing Conference. Orlando, FL : s.n., 2006.
- Khilawala, Rashida. 2010.** Buzzle. *History of Human Resource Management*. [Online] 2010. <http://www.buzzle.com/articles/history-of-human-resource-management.html>.
- Larman, Craig. 1999.** *Applying UML and Pattern. An Introduction to Object-Oriented Analysis and Design*. Upper Saddle River : Prentice Hall, Inc., 1999. ISBN: 970-17-0261-1.
- Object Management Group. 2012.** OMG. *Introduction To OMG's Unified Modeling Language*. [Online] 2012. [Cited: 01 10, 2012.] [http://www.omg.org/gettingstarted/what\\_is\\_uml.htm](http://www.omg.org/gettingstarted/what_is_uml.htm).
- Oracle Corporation. 2012.** FOSS License Exception. [Online] 2012. [Cited: Enero 11, 2012.] <http://www.mysql.com/about/legal/licensing/foss-exception/>.
- . 2012. Netbeans IDE. *Netbeans IDE. Features*. [Online] 2012. [Cited: Enero 11, 2012.] <http://netbeans.org/features/index.html>.
- OrangeHRM. 2011.** OrangeHRM. [Online] 2011. [Cited: 12 9, 2011.] <http://www.orangehrm.com/open-source-product-features.shtml>.
- Pollard, William and Cantu, Carlos H. 2000.** *The Soul of the Firm*. s.l. : Zondervan, 2000.
- Potencier, Fabien. 2008.** *La Guía Definitiva de Symfony*. 2008.
- Pressman, Roger S. 2004.** *Ingeniería del Software. Un enfoque práctico*. Quinta Edición. s.l. : McGraw Hill, 2004.
- Almenares, Liudmila Sanchez. 2009.** *Prueba Automática de Carga y Estrés*. No. 2, La Habana : Grupo Editorial Ediciones Futuro, 2009, Serie Científica de la Universidad de las Ciencias Informáticas, Vol. 2.
- Napal Torres, Irina and Díaz Pérez, Heney. 2008.** *Pruebas de Aceptación y Piloto*. No. 7, La Habana : Grupo Editorial Ediciones Futuro, 2008, Serie Científica de la Universidad de las Ciencias Informáticas, Vol. 1.
- Real Academia Española. 2010.** Diccionario de la Real Academia Española. [Online] 2010. <http://buscon.rae.es/drae/>.
- Riggs, Simon and Krosign, Hannu. 2010.** *PostgreSQL 9 Administration Cookbook*. Birmingham : Packt Publishing Ltd., 2010. ISBN 978-1-849510-28-8.
- Rudman, Richard. 2010.** History and development of human resources management. 2010.
- Sage HRMS. 2011.** Sage HRMS. *Sage Human Resources Manager and Personal Software*. [Online] 2011. [Cited: 12 9, 2011.] [http://www.sagehrms.com/products\\_and\\_services/products\\_by\\_name/sage-abra-hrms](http://www.sagehrms.com/products_and_services/products_by_name/sage-abra-hrms).

- SAP ERP. 2011.** SAP ERP HUMAN CAPITAL MANAGEMENT. [Online] 2011. [Cited: 12 8, 2011.] <http://www.sap.com/solutions/business-suite/erp/hcm/index.epx>.
- Scrum. 2009.** Scrum methodology. *Advantages of Scrum methodology*. [Online] 2009. [Cited: enero 10, 2012.] <http://www.scrummethodology.org/scrum-advantages.html>.
- Software Development Methodologies. 2012.** Software Development Methodologies. [Online] 2012. [Cited: enero 8, 2012.] <http://softwaredevelopmentmethodologies.org/>.
- Sousa, Susan de. 2009.** My PM Expert. *The Advantages and Disadvantages of Agile SCRUM Software Development*. [Online] 2009. [Cited: enero 9, 2012.] <http://www.my-project-management-expert.com/the-advantages-and-disadvantages-of-agile-scrum-software-development.html>.
- Taylor, Frederick W. 1911.** *The Principles of Scientific Management*. New York : Harper & Brothers, 1911.
- The jQuery Project. 2010.** jQuery: The Write Less, Do More, JavaScript Library. [Online] 2010. [Cited: Enero 10, 2012.] <http://jquery.com/>.
- The PHP Group. 2012.** PHP: Hypertext Preprocessor. [Online] 2012. [Cited: Enero 10, 2012.] <http://www.php.net/>.
- UCI. 2008.** *Proyecto Estratégico*. La Habana : s.n., 2008.
- UML. 2011.** UML Forum. *UMLFAQs*. [Online] 2011. [Cited: 01 10, 2012.]
- Visual Paradigm. 2012.** Visual Paradigm. *UML, BPMN and Database Tool for Software Development*. [Online] 2012. [Cited: Enero 10, 2012.] <http://www.visual-paradigm.com>.
- W3Schools. 2012.** W3Schools. *JavaScript Introduction*. [Online] 2012. [Cited: Enero 10, 2012.] [http://www.w3schools.com/js/js\\_intro.asp](http://www.w3schools.com/js/js_intro.asp).
- Weske, Mathias. 2007.** *Business Process Management*. Postdam : Springer, 2007. ISBN 978-3-540-73521-2.
- Zapata, Carlos Mario. 2009.** Scielo. [Online] 2009. [Cited: enero 10, 2012.] <http://www.scielo.org.co/pdf/ring/n29/n29a3.pdf>.
- Zavala Ruiz, J. 2008.** *Análisis organizacional de empresas de software. Una primera aproximación a una visión alterna de la Ingeniería de Software*. Universidad Autónoma Metropolitana. México : s.n., 2008.
- Zend Technologies Ltd. 2012.** Zend Framework. [Online] 2012. [Cited: Enero 10, 2012.] <http://framework.zend.com/about/overview>.