



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 1

Versión 2.0 del componente Personal del Sistema de Gestión
Universitaria, Akademos

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS**

Autor: Alejandro García Gómez

Tutores: Ing. Alexander Rodríguez Mompíe

Ing. Yasmany Tellez Collazo

Co-tutores: Ing. Mairelys Boeras Velázquez

Ing. Enmanuel Azaharez Reyes

La Habana, Cuba. Junio, 2016

Declaración de autoría

Declaro ser autor de la presente tesis que tiene por título: Versión 2.0 del componente Personal del Sistema de Gestión Universitaria, Akademos y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmamos la presente tesis a los ___ días del mes de junio del año 2015.

Alejandro García Gómez

Firma del autor

Ing. Alexander Rodríguez Mompíe

Ing. Yasmany Tellez Collazo

Firma del tutor

Firma del tutor

Resumen

El Departamento de Desarrollo de la Dirección de Informatización de la Universidad de las Ciencias Informáticas es la encargada de desarrollar el Sistema de Gestión Universitaria. Este sistema está integrado por varios componentes, donde el componente Personal se encarga de la gestión y control de las personas en la Universidad. Las funcionalidades presentes en el componente Personal no son suficientes para las necesidades de informatización identificadas en el área de Capital Humano y en la Oficina de Identificación. Con el presente trabajo se desarrolla una nueva versión del componente Personal, donde se le incluyen funcionalidades nuevas y otras que complementan algunas ya existentes de la versión anterior. Se incorporan acciones para el control de la identificación impersonal emitida para visitantes y el seguimiento y control de la información del proceso de selección e integración del personal en la Universidad. Este además permite la gestión de la información de los periodos de licencias de maternidad y licencias no retribuidas asignadas a los trabajadores de la institución; así como también posibilita una mejor confección de los reportes, funcionalidad que se encontraba inhabilitada en la versión anterior.

Palabras clave: componente, gestión, información, personal.

Índice

Introducción	1
1. Capítulo 1: Fundamentación teórica	5
1.1 Conceptos asociados al dominio de la aplicación	5
1.2 Análisis de sistemas homólogos	6
1.2.1 Sistemas de gestión personal en el mundo	6
1.2.2 Sistemas de gestión personal en Cuba	7
1.2.3 Resultado del análisis de las soluciones	8
1.3 Metodología y herramientas utilizadas en la implementación de la solución	10
1.3.1 Lenguajes	11
1.3.2 Herramientas	14
1.3.3 Marco de trabajo	17
1.4 Conclusiones parciales	18
2. Capítulo 2: Descripción de la propuesta de solución	19
2.1 Procesos del negocio	19
2.1.1 Diagrama de clases del dominio (Reportes)	23
2.1.2 Bases legales	24
2.1.3 Reglas del negocio	25
2.2 Requisitos del sistema	26
2.2.1 Definición de las técnicas de obtención de requisitos	26
2.2.2 Definición de requisitos funcionales	26
2.2.3 Definición de requisitos no funcionales	30
2.2.4 Descripción de requisitos	32
2.3 Descripción de la propuesta de solución	36
2.4 Descripción de la arquitectura y el diseño	39
2.4.1 Arquitectura Cliente-Servidor	39
2.4.2 Patrón de arquitectura	40
2.4.3 Modelo de despliegue	42
2.4.4 Patrones de diseño	43
2.4.5 Diagramación de las vistas	47
2.4.6 Sistema de mensajes de la propuesta	48
2.4.7 Mapa de navegación	50
2.4.8 Patrones de base de datos	50

2.4.9. Modelo de datos.....	51
2.5. Conclusiones parciales.....	52
3. Capítulo 3: Implementación y validación de la solución	53
3.1 Estándares de codificación	53
3.1.1 Identación, llaves de apertura y cierre y tamaño de las líneas	53
3.1.2 Conversión de nomenclatura	53
3.1.3 Estructura de control.....	55
3.1.4 Documentación	55
3.1.5 Buenas prácticas	56
3.2 Validación de requisitos	56
3.2.1 Técnicas de validación de requisitos	57
3.3 Validación de la aplicación	58
3.3.1 Prueba de unidad	59
3.3.2 Pruebas de integración.....	61
3.3.4. Pruebas de rendimiento	62
3.3.5 Pruebas de sistema.....	64
3.4 Conclusiones parciales.....	68
Conclusiones generales	69
Recomendaciones	70
Referencias bibliográficas.....	71

Índice de figuras

Figura 1. Modelo del proceso DAC [12].....	11
Figura 2. Diagrama de proceso de negocio de licencia de maternidad	23
Figura 3. Diagrama de clases del dominio	24
Figura 4. Arquitectura Cliente-Servidor [40].	40
Figura 5. Patrón Modelo-Vista-Controlador en GUUD [42]	40
Figura 6. Representación del modelo de despliegue	42
Figura 7. Patrón Creador	44
Figura 8. Patrón Controlador.....	44
Figura 9. Patrón Mediador para la controladora.....	46
Figura 10. Patrón Mediador para la librería	46
Figura 11. Patrón Instancia única	47
Figura 12. Áreas de la vista de presentación [48]	47

Figura 13. Áreas de la vista de escritorio [48]	48
Figura 14. Áreas de la vista de gestión de procesos [48]	48
Figura 15. Mensaje incrustado en formularios [48]	49
Figura 16. Mensaje de información en ventana emergente [48].....	49
Figura 17. Mensaje de error en ventana emergente [48]	49
Figura 18. Mensaje de confirmación en ventana emergente [48]	49
Figura 19. Mapa de navegación principal del Componente Personal	50
Figura 20. Patrón de diseño de base de datos esquema de estrella.....	51
Figura 21. Modelo físico de datos de las licencias.....	52
Figura 22. Identación, llaves de apertura y cierre y tamaño de las líneas	53
Figura 23. Conversión de nomenclatura de variables	54
Figura 24. Conversión de nomenclatura de clases.....	54
Figura 25. Conversión de nomenclatura de funciones.....	54
Figura 26. Conversión de nomenclatura de estructuras de control.....	55
Figura 27. Documentación de una clase.....	55
Figura 28. Documentación de una función	56
Figura 29. Informe agregado resultado de la prueba de rendimiento.....	63
Figura 30. Árbol de resultado (ícono en verde, resultado exitoso).....	64

Índice de tablas

Tabla 1. Listado de requisitos funcionales.....	27
Tabla 2. Listado de requisitos no funcionales	30
Tabla 3. Descripción de requisitos (Listar licencia de maternidad).....	32
Tabla 4. Estrategia de pruebas de software	58
Tabla 5. Caso de prueba unitaria de la funcionalidad Registrar licencia de maternidad.....	59
Tabla 6. Iteración de las pruebas de caja blanca.....	61
Tabla 7. Caso de prueba de integración con el componente Estructura y Composición.....	62
Tabla 8. Caso de prueba Listar Licencia de maternidad (Parte I)	65
Tabla 9. Caso de prueba Registrar especialista (Parte II)	66
Tabla 10. Iteraciones de las pruebas de pruebas de validación.....	68

Introducción

En el campo científico-técnico se evidencian notables cambios que se adecuan a las emergencias tecnológicas que exige la sociedad, donde han ido evolucionando radicalmente incorporándose al quehacer cotidiano de las personas. Esto ha provocado un gran impacto en todas las esferas sociales, transformado a las Tecnologías de la Información y las Comunicaciones (TIC) en un elemento estratégico para el crecimiento, la madurez y la transformación de las organizaciones, proporcionando y demostrando que los productos de software necesitan estar en un continuo perfeccionamiento en todas las esferas de la sociedad.

Cuba, en su progreso continuo, trata de convertir las TIC en un sector de desarrollo estratégico para la sociedad. Los recursos humanos juegan un papel importante en el cumplimiento de este objetivo, por ser los responsables de llevar a cabo los avances, los logros y los errores de las organizaciones. Por eso, no es exagerado afirmar que constituyen junto a una correcta gestión, el recurso máspreciado y valioso para lograr el éxito.

La gestión de los Recursos Humanos (RR.HH.) son un eslabón fundamental para concentrar, definir y desarrollar procesos encaminados a obtener y coordinar a las personas de forma coherente en las organizaciones; por eso la necesidad de controlar y mejorar continuamente dichos procesos. Siendo la informatización de los mismos, una forma de contribuir al perfeccionamiento de las instituciones.

Una de las instituciones capaces de elevar el nivel de informatización en el país es la Universidad de las Ciencias Informáticas (UCI). En ella se han desarrollado varias soluciones informáticas donde el denominado Sistema de Gestión Universitaria (SGU) constituye un solución única e integradora de los procesos sustantivos de la comunidad universitaria. Este sistema está integrado por varios componentes, donde el componente Personal relaciona, organiza y gestiona los datos propios de la persona, tales como: información nominal, datos personales, dirección particular, rasgos fenotípicos y datos de superación. El componente Personal ha tenido gran nivel de aceptación y utilización por parte de los usuarios, no obstante, luego de realizar varias entrevistas a especialistas de diferentes áreas, se detectaron un conjunto de problemas y necesidades existentes en la institución relacionados con la gestión de las personas.

Entre las necesidades identificadas, gestión y control de la selección e integración del personal (contratación), es un proceso ejecutado por varias personas que intervienen en sus diferentes niveles de aprobación y verificación. Estas personas se encuentran en diferentes áreas dispersas físicamente en la Universidad lo que dificulta y demora la agilidad del proceso, generando gran cúmulo de documentación

física debido a que se elabora un expediente con información duplicada por cada persona que solicite una plaza.

Por otra parte, licencia de maternidad y la licencia no retribuida son etapas por las que puede transitar un trabajador durante su vida laboral. Una vez que se solicita una de estas alternativas de seguridad social, el trabajador debe prescindir de los servicios que brinda la Universidad. Actualmente no se cuenta con un mecanismo confiable que controle el flujo de información que se genera cuando un trabajador transita por una de estas etapas.

Además, la Universidad en diferentes etapas del año desarrolla y apoya eventos y actividades de diferentes magnitudes, lo que implica que arriben a ella visitantes con el fin de participar en estos. En ocasiones estos visitantes pueden llegar de forma masiva y su acreditación se puede tornar engorrosa debido a que se cuenta con una sola oficina de identificación y un limitado número de trabajadores. Dicha oficina tiene implementado el proceso de acreditación de forma personal, lo que arriba a un problema cuando se necesita acreditar de forma masiva y en un limitado tiempo.

Dentro del componente Personal anteriormente descrito, la agrupación funcional Reporte se encarga de brindar la información referente a los procesos y el movimiento en Personal. En ocasiones se hace necesario obtener de forma eficiente de acuerdo con determinados criterios de búsqueda información básica para el usuario, información que es gestionada de igual manera en el componente Pregrado, pero actualmente el componente Personal no se realiza el proceso de manera homogénea para la gestión de los reportes. Esto se debe principalmente a que no se cuenta con un mecanismo que permita a los usuarios obtener de forma rápida y centralizada, información referente al personal de la Universidad.

El análisis de lo planteado anteriormente permite definir como **problema de investigación**: ¿Cómo mejorar el componente Personal del Sistema de Gestión Universitaria?

Ante el problema definido se identifica como **objeto de estudio**: los procesos de gestión administrativa orientados a la gestión del personal y específicamente su **campo de acción** la gestión de los procesos de contratación, licencias y solicitudes de visitantes en la Universidad de las Ciencias Informáticas.

Se define como **objetivo general**: desarrollar la versión 2.0 del componente Personal que permita la obtención de información a través de los reportes, facilitando la toma de decisiones y la gestión de los procesos solicitudes visitantes, contratación y licencias en el Sistema de Gestión Universitaria de la Universidad de las Ciencias Informáticas.

Se identificaron los siguientes **objetivos específicos** siguientes:

- Elaborar el marco teórico de la investigación.
- Diseñar la propuesta de solución a partir del proceso de desarrollo de software utilizado.
- Implementar las funcionalidades definidas en el análisis del componente.
- Validar el correcto funcionamiento de la solución desarrollada a partir de la realización de pruebas de software.

La investigación se sustenta en la siguiente **idea a defender**: con el desarrollo de la versión 2.0 del componente Personal del Sistema de Gestión Universitaria, contribuirá a la mejora de los procesos de contratación, licencias y las solicitudes de visitantes, así como la toma de decisiones en la Universidad de las Ciencias Informáticas, mediante la obtención de reportes asociados a la información del personal universitario.

Para cumplir con los objetivos específicos planteados se proponen las siguientes **tareas de la investigación**:

- Elaboración del diseño teórico de la investigación.
- Estudio de las tendencias actuales que intervienen en los procesos de gestión de la información de Recursos Humanos.
- Análisis de las tecnologías a utilizar para el desarrollo de la aplicación.
- Identificación de los requisitos según el proceso de desarrollo aplicado para elaborar la propuesta de solución.
- Validación de los requisitos del sistema.
- Modelado de la base de datos que permita establecer un modelo lógico y físico de la estructura de datos asociada a la solución.
- Implementación de las funcionalidades definidas para la solución propuesta.
- Validación del sistema para garantizar su funcionamiento adecuado.

Con el desarrollo de la investigación se pretende obtener como **resultado** el desarrollo de la versión 2.0 del componente Personal, que contribuirá a la gestión y administración de la información personal del Sistema de Gestión Universitaria en la Universidad de las Ciencias Informáticas.

Los **métodos científicos** utilizados para la realización de la investigación son los siguientes:

➤ **Métodos teóricos:**

- **Analítico-sintético:** en el estudio y resumen de la información tratada durante la consulta del estado del arte, las herramientas, los lenguajes propuestos para dar solución al problema y para identificar las principales necesidades que el sistema debe satisfacer.
- **Histórico-lógico:** se evidencia en el estudio de los sistemas de gestión de información de Recursos Humanos, así como en el análisis de sus características y objetivos, permitiendo entender su funcionamiento y definir sus objetivos.
- **Modelación:** se utilizó para representar a través de diagramas, la lógica de negocio y las clases que conforman la solución informática.

➤ **Métodos empíricos:**

- **Entrevista:** se utiliza para lograr un mejor entendimiento sobre las características y distintas etapas en el funcionamiento de los procesos de Gestión de licencias, contratación y solicitudes de visitantes en la Universidad; sus diseños están accesibles en el **Anexo 1**.

La presente investigación se estructura en tres capítulos describiéndose de la siguiente manera:

Capítulo 1. Fundamentación teórica. Se expone el estado del arte del objeto de estudio de la presente investigación y se definen conceptos fundamentales para su mejor entendimiento. Se realiza la descripción y análisis de los sistemas existentes y una caracterización de las herramientas, técnicas, metodología y el proceso de desarrollo a utilizar en la construcción de la propuesta de solución.

Capítulo 2. Concepción de la propuesta de solución. Se describe la propuesta de solución que permite describir las reglas del negocio, construir el modelo conceptual y definir mediante las técnicas de obtención de requisitos, los requisitos funcionales y no funcionales. Además, se especifica la arquitectura, los patrones de diseño y de base de datos que se utilizan en la investigación, para la elaboración del diagrama de despliegue y el modelo de base de datos.

En el Capítulo 3. Implementación y validación de la solución. Se definen los estándares de codificación a utilizar y la integración con otras aplicaciones. Además, se muestra la estrategia de pruebas utilizada para validar la propuesta de solución.

1. Capítulo 1: Fundamentación teórica

El presente capítulo tiene como objetivo sentar las bases en la especificación de conceptos y aspectos más significativos en relación con los sistemas de software basados en los Recursos Humanos, enfatizado en diferentes fuentes bibliográficas para profundizar y ampliar el estudio del arte. Luego se caracterizan las tecnologías y la metodología de desarrollo de software a utilizar, definidas por el Departamento de Desarrollo de la Dirección de Informatización (DIN).

1.1 Conceptos asociados al dominio de la aplicación

Persona: organización de personas y de bienes a la que el derecho reconoce capacidad unitaria para ser sujeto de derechos y obligaciones, como las corporaciones, asociaciones, sociedades y fundaciones [1].

Capital humano: potencialidad productiva de las personas de una empresa en función de su educación, formación y capacidades [2].

Recursos humanos: personas con las que una organización (con o sin fines de lucro, y de cualquier tipo de asociación) cuenta para desarrollar y ejecutar de manera correcta las acciones, actividades, labores y tareas que deben realizarse y que han sido solicitadas a dichas personas [3].

Credencial: documento que acredita a una persona para desempeñar una determinada función [4].

Licencia: autorización que se concede para explotar con fines industriales o comerciales una patente, marca o derecho [5].

Reporte: informe. Dicho de una cosa: producir algún beneficio o ventaja, o, por el contrario, dificultades o disgustos. Otra variante cuando se habla de un reporte es el cruce de datos realizado por un software determinado sobre una base de datos relacional. Una base de datos aloja información sobre tablas y permite su consulta mediante el Lenguaje de Consultas Estructurado (SQL). Un reporte, en este caso, haría un cruce de datos significativo para dar cuenta de una determinada situación [6]. Un reporte es aquel documento que se utilizará cuando se quiera informar o dar noticia acerca de una determinada cuestión. Aunque básicamente su objetivo será el de informar, los reportes también podrán incluir algunos elementos persuasivos, como pueden ser recomendaciones o sugerencias y también algunas conclusiones a través de las cuales se le indique al lector alguna acción o conducta a adoptar en el futuro.

1.2 Análisis de sistemas homólogos

Con el objetivo de mejorar el nivel funcional del componente Personal, se realizó un análisis de los sistemas nacionales e internacionales, que permitan apoyar el desarrollo de la investigación para lograr adaptar el SGU a las necesidades de los clientes. A continuación, se describen algunos que serán útiles en el desarrollo de la investigación.

1.2.1 Sistemas de gestión personal en el mundo

DSR Gestión del Personal

Es un software desarrollado por *Dataproces Soluciones Registradas (DSR)* que proporciona medios fáciles y flexibles para organizar y controlar los datos del empleado. En su base de datos la información almacenada en cada una de las fichas de empleado contiene, tanto los datos personales del empleado como varios códigos de estados administrativos. Entre estos códigos existe uno para contratos de empleo, que puede utilizar para asignar niveles o contratos de empleo estandarizados [7].

El sistema realiza el proceso de selección e integración de personal definiendo el perfil a seleccionar como primer paso, luego se recogen todos los currículums del personal que solicita el puesto de trabajo y se realiza una valoración automática de las calificaciones de los aspirantes al puesto. Este proceso, no sólo puede realizar selecciones de personal externo de la empresa, sino que también puede buscarse entre los empleados de la misma, bien porque haya solicitado participar o bien porque el proceso de selección los haya considerado adecuados. Al finalizar el proceso de selección con la elección del candidato a entrar en la empresa, se pasan automáticamente sus datos de la ficha de aspirante a la de empleado. [7].

La información confidencial de un empleado se controla a través de una tabla asociada a la ficha del empleado. Esto le permite dar un acceso relativamente amplio a la información básica del archivo del empleado y poner unas restricciones de seguridad más rigurosas en los datos confidenciales. [7].

Altamira Employees

Altamira HRM¹ es una plataforma basada en la red, que maneja los datos de los empleados y los principales procesos empresariales que los involucran. Almacena datos personales, funciones y competencias, proporcionando los instrumentos necesarios para el manejo de las evaluaciones de desempeño y para la administración de la formación a nivel empresarial. Generalmente, la plataforma se vincula con el software

¹ **HRM:** (*Human Resource Management*, o sus siglas en español Gestión de Recursos Humanos).

de nómina para así completar los datos que este maneja. Es un sistema de información de gran alcance que permite analizar los datos y suministrarle a la alta gerencia un auténtico entorno de inteligencia de negocio. Altamira Employees no solamente proporciona un servicio a los departamentos de Recursos Humanos, es también una herramienta de gestión y comunicación para toda la organización [8].

Altamira Employees es un software de gestión de RR.HH. altamente personalizable desarrollado para la administración de recursos humanos, dando la flexibilidad de diseñar la plataforma HRM, según los requerimientos de cada organización. Toda la información importante sobre un empleado del personal, tales como: datos personales, posiciones, beneficios, retribuciones, etc. puede ser rastreada históricamente a través de una base de datos en línea que permite rastrear el historial de la información del empleado y así se pueden conocer las diferentes versiones realizadas a dicho empleado [8].

1.2.2 Sistemas de gestión personal en Cuba

ASSET (Sistema de gestión integral)

ASSETS es un Sistema de Gestión Integral estándar y el parámetro que permite el control de los procesos de compras, ventas, producción, taller, inventario, finanzas, contabilidad, presupuesto, activos fijos, útiles y herramientas y Recursos Humanos [9].

El módulo de Recursos Humanos (Versión 3.1 Access 97) está concebido para calcular las nóminas y controlar los recursos laborales de una entidad; tiene una base de datos única, por lo que se reduce considerablemente la posibilidad de ocurrencia de errores. Por Recursos Humanos se pueden controlar íntegramente los recursos laborales: empleados, estructura organizativa de la entidad y plantilla. Siempre que se introducen altas, bajas y otros movimientos, se actualiza automáticamente el registro de empleados: catálogo de empleados y se generan los reportes correspondientes con el formato oficial. Es posible modificar plantillas, introducir cambios en la estructura organizativa, crear nuevos cargos y realizar conversiones de plazas [9].

Componente Personal v1.0 del Sistema de Gestión Universitaria

El componente Personal del SGU se encarga de la informatización de procesos de gestión de la información de los datos de cada persona en Universidad. La herramienta fue desarrollada utilizando el marco de trabajo GUUD² en su versión 2.0. Este sistema cuenta con varias agrupaciones funcionales complementarias como

² Constituye un híbrido entre el marco de trabajo de PHP CodeIgniter y la librería de JavaScript, JQuery. Implementa el patrón Modelo-Vista-Controlador (MVC) así como la programación orientada a aspectos.

Nómina, Saneamiento de personas y Configuración, las cuales enriquecen el correcto funcionamiento del componente. Las agrupaciones funcionales fundamentales que brinda el componente son:

- Identificación, mecanismo de acreditación de una persona en la Universidad.
- Personal, su alcance es basado en la introducción de un individuo al sistema y la búsqueda de datos de una persona de interés.
- Proceso de baja, gestiona la información asociada al proceso de baja de una persona en la Universidad.
- Planificación de acceso, recopila toda la información referente al acceso de las personas a la Universidad.
- Evaluación y Proyección, permite realizar pronósticos de la movilidad del personal en la Universidad.

1.2.3 Resultado del análisis de las soluciones

La elección de la herramienta más conveniente depende de las necesidades de los clientes, así como de las funcionalidades que ofrezcan de base para soportar una implementación que responda a dichas necesidades. Durante el estudio de las diversas soluciones existentes que manejan información asociada a la gestión personal, se pudo constatar que poseen una gran variedad de funcionalidades en dependencia del entorno empresarial que se encuentre.

Las soluciones anteriormente planteadas se complementan en un denominador común y es que todas de alguna manera son sistemas de planificación de recursos empresariales ('ERP', por sus siglas en inglés, *Enterprise Resource Planning*) los cuales permiten la gestión de la información que integran y automatizan muchas de las prácticas de negocio asociadas con los aspectos operativos o productivos de la empresa. Estos sistemas integran bajo el mismo paraguas todo el software que necesita una empresa para el correcto funcionamiento de su entorno de negocio. Se pretende que la información esté disponible para todo el mundo todo el tiempo. Para ello, los sistemas ERP mantienen todas las operaciones y procesos de la empresa bajo una misma base de datos compartida. Estos permiten a las empresas evaluar, controlar y gestionar más fácilmente su negocio en todos los ámbitos. A su vez, permiten agilizar los diferentes tipos de trabajo de cada usuario, reduciendo en tiempo real las tareas repetitivas y permitiendo el aumento de la comunicación entre todas las áreas que integran la empresa. También son conocidos como sistemas integrales de empresa o sistemas integrados de gestión. Ellos tienen entre sus objetivos principales el

satisfacer las diferentes necesidades de información de la empresa para conseguir que los gestores empresariales dispongan de un soporte para tomar decisiones y controlar el cumplimiento de objetivos [10].

DSR Gestión del Personal es una solución donde la culminación de la implementación depende de la capacidad y destreza de la mano de obra, implica también la educación y capacitación, para hacer que el software se aplique correctamente. Este sistema ERP, sus beneficios no se presentan de inmediato con la implementación del software, estos serán evidentes tiempo después de que la aplicación esté en ejecución.

Altamira Employees es una potente solución que genera el problema de la gestión de información del personal, pero este presenta algunas irregularidades en comparación con las reglas del negocio dentro del SGU. El sistema permite a los empleados el acceso a la plataforma a través de sus cuentas particulares, visualizar y actualizar sus datos. Para el desarrollo de la investigación estos servicios son peligrosos por concepto de información confiable, en consideración de lo anteriormente planteado se evidencia que el personal no debe ser capaz de actualizar sus datos personales; la información versionada debe ser única en la base de datos y solamente actualizada por el personal correspondiente. Este tipo de software puede generar un incremento de la productividad, pero existen muchas empresas que le resulta casi imposible pagar el costo de las licencias, implementación y sobre todo el mantenimiento del mismo, por ser este un sistema dinámico y de nada sirve tener una misma solución en una empresa que crece y cambia día a día.

Los sistemas como DSR Gestión del Personal y Altamira Employees abarcan una amplia gama de funcionalidades que van encaminadas a la gestión de los RR.HH., estos sistemas constituyen una importante fuente de estudio por presentar características que se acoplan a las necesidades del versionado. Pero también existen características que descartan que se puedan integrar a la arquitectura definida para el SGU, basándose en las limitaciones anteriormente planteadas; que con vista a mejorar el proceso de gestión de la información personal en la Universidad ve la necesidad de una solución propia que responda a las necesidades específicas de la UCI. Ignorando soluciones generales para el sector empresarial que no pueden ser adaptadas a las características del SGU.

ASSET es utilizado en área de RR.HH. en la UCI y aunque es una herramienta modelo con años de servicio presenta deficiencias como: software privativo, obsoleto, se incurre en gastos por su utilización en la Universidad, no se tiene acceso al código fuente, la información de las personas solo se encuentra en áreas autorizadas, es decir que, aunque una persona esté autorizado al acceso a la información también tiene que estar en un área determinada. Por tanto, no resuelve los problemas de acceso a la información desde

cualquier área de la Universidad e imposibilita la incorporación de nuevos procesos que gestionen la información cada vez más vinculada a la Universidad en temas de RRHH.

El componente Personal presenta un conjunto de aristas que evidencian lo calificado que se encuentra para ser la línea a seguir en el desarrollo de la propuesta de solución. Esta herramienta en conjunto con la herramienta ASSETS, son las que gestionan la información referente a los RR.HH. en la Universidad. El componente Personal posibilita incorporar nuevos procesos necesarios en la comunidad universitaria, por cumplir este sistema con todas las reglas o pautas que se describen en el desarrollo de un software en la Universidad, tales como ser multiplataforma, software libre, su base de datos es altamente reutilizable, no incurre en ningún costo su reutilización e implementación y por su alta explotación surge la presente investigación para desarrollar una solución informática integrada al SGU que mejore la gestión de información de las personas en la UCI en función de elevar la calidad del proceso docente-educativo en la institución.

1.3 Metodología y herramientas utilizadas en la implementación de la solución

Una metodología de desarrollo de software es una estructura de trabajo usada para planificar y controlar el proceso de desarrollo en sistemas de información. Cada metodología de desarrollo de software tiene más o menos su propio enfoque para el desarrollo de software [11].

La metodología para el desarrollo ágil con calidad (DAC) es un proceso de desarrollo de software que combina las metas y prácticas de las áreas de procesos del nivel 2 de Integración de Modelos de Madurez de Capacidades (CMMI por sus siglas en inglés) con las buenas prácticas de la dirección y desarrollo ágil de proyectos de software. Es un proceso colaborativo, recursivo-iterativo, incremental y guiado por procesos y requisitos. Su modelo del proceso es una adaptación del modelo en Cascada a los modelos Programación Extrema y Desarrollo concurrente. La metodología está enfocada a proyectos pequeños o proyectos grandes divididos en sub-proyectos que desarrollan software de gestión basado en componentes. El proceso tiene 8 actividades del marco de trabajo del proceso común llamadas Fases o Procesos del ciclo de vida las cuales son: Inicio, Análisis y diseño arquitectónico, Requisitos, Construcción, Cierre de iteración (opcional), Liberación, Transición y Cierre, ocurriendo las iteraciones concurrentes entre las fases de Requisitos, Construcción y Cierre de iteración. Entre las fases de Requisitos y Construcción puede ocurrir un ciclo pues a medida que los requisitos son especificados pueden ir entrando a la fase de Construcción. El proceso tiene dos Áreas de procesos de protección: Gestión de proyectos y Soporte, así como dos Procesos horizontales cuyas tareas están presentes en varias de las fases del proceso común en forma de

subprocesos: Arquitectura y Planificación [12]. La Figura 1 muestra el modelo de proceso de la metodología DAC.

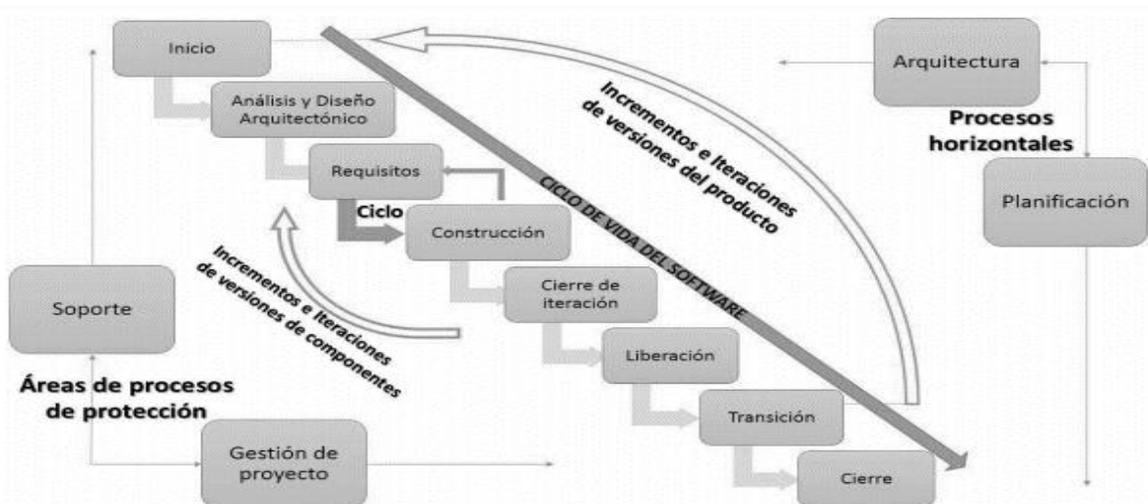


Figura 1. Modelo del proceso DAC [12]

1.3.1 Lenguajes

Los lenguajes de programación son una sintaxis codificada usada por los programadores para comunicarse con la computadora. Es el único lenguaje que entienden tanto los programas de software como los de hardware. El lenguaje le permite al usuario dictar los comandos que la computadora debe entender para procesar los datos, estos vienen en varios tipos que emplean conjuntos de sintaxis diferentes [13].

Nombre: HTML 4. **Descripción de su uso:** lenguaje de Marcas de Hipertexto (HTML por sus siglas en inglés), hace referencia al lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes, se escribe en forma de *etiquetas*, rodeadas por corchetes angulares (<,>), también puede describir hasta un cierto punto, la apariencia de un documento y es el lenguaje de publicación del *World Wide Web* (WWW³) [14].

Nombre: PHP 5.6.7. **Descripción de su uso:** preprocesador de Hipertexto (PHP por sus siglas en inglés) es un lenguaje de *código abierto* interpretado, de alto nivel, embebido en páginas HTML y ejecutado en el servidor. Con PHP no se encuentra limitado a resultados en HTML. También puede presentar otros

³**WWW:** En informática, la *World Wide Web* (WWW) o Red informática mundial comúnmente conocida como la Web, es un sistema de distribución de documentos de hipertexto o hipermedias interconectados y accesibles vía Internet. Con un navegador web, un usuario visualiza sitios web que pueden contener texto, imágenes, vídeos u otros contenidos multimedia, y navega a través de esas páginas usando hiperenlaces.

resultados, como XHTML⁴ y archivos XML⁵. Así mismo, puede autogenerar estos archivos y almacenarlos en el sistema de archivos en vez de presentarlos en la pantalla. Quizás la característica más potente y destacable es su soporte para una gran cantidad de base de datos. Escribir una interfaz vía web para una base de datos es una tarea simple con PHP [15].

Nombre: JavaScript1.8. **Descripción de su uso:** JavaScript es un lenguaje de programación interpretado, se define como orientado a objetos y basado en prototipos. Es dinámico, responde a eventos en tiempo real como presionar un botón, pasar el puntero del ratón sobre un determinado texto o el simple hecho de cargar la página o caducar un tiempo. Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, aunque existe una forma de JavaScript del lado del servidor. JavaScript puede incluirse en cualquier documento y es compatible con HTML en el navegador del cliente, ya sea PHP, ASP⁶, JSP⁷ y SVG⁸ [16].

Nombre: XML 1.0. **Descripción de su uso:** el Lenguaje de Marcas Extensibles (XML por sus siglas en inglés) es un lenguaje de marcas desarrollado por el *World Wide Web Consortium (W3C)* utilizado para almacenar datos en forma legible. Deriva del lenguaje SGML⁹ y permite definir la gramática de lenguajes específicos para estructurar documentos grandes. A diferencia de otros lenguajes, XML da soporte a bases de datos, siendo útil cuando varias aplicaciones se deben comunicar entre sí o integrar información. Después de diseñado y puesto en producción, es posible extender XML con la adición de nuevas etiquetas, de modo que se pueda continuar utilizando sin complicación alguna. Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarla. XML mejora la compatibilidad entre aplicaciones. Se puede comunicar aplicaciones de distintas plataformas, sin que importe el origen de los datos [17].

Nombre: CSS 3. **Descripción de su uso:** las Hojas de Estilo en Cascada (CSS por sus siglas en inglés) hacen referencia a un lenguaje usado para definir la presentación de un documento estructurado escrito en

⁴**XHTML:** Siglas del inglés *eXtensible Hyper Text Markup Language*. XHTML es básicamente HTML expresado como XML válido. Es más estricto a nivel técnico, pero esto permite que posteriormente sea más fácil al hacer cambios o buscar errores entre otros.

⁵**XML:** Por siglas en inglés *eXtensible Markup Language* ('lenguaje de marcas extensible'), es un lenguaje de marcas desarrollado por el *World Wide Web Consortium (W3C)* utilizado para almacenar datos en forma legible.

⁶*Active Server Pages (ASP)* es una tecnología de Microsoft del tipo "lado del servidor" para páginas web generadas dinámicamente.

⁷*Java Server Pages (JSP)* es una tecnología que ayuda a los desarrolladores de software a crear páginas web dinámicas.

⁸Gráficos Vectoriales Redimensionables (del inglés *Scalable Vector Graphics*) o SVG son una especificación para describir gráficos vectoriales bidimensionales, tanto estáticos como animados.

⁹**SGML** son las siglas de *Standard Generalized Markup Language* o "Estándar de Lenguaje de Marcado Generalizado".

HTML o XML (y por extensión en XHTML). La información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML. En este último caso podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo `<style>`. CSS permite el control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo. Los navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web, con lo que aumenta considerablemente la accesibilidad. Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario [18].

Nombre: Lenguaje de Modelado Unificado (UML por sus siglas en inglés) 2.0. **Descripción de su uso:** lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el *Object Management Group* (OMG). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un *plano* del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. Es importante resaltar que UML es un lenguaje de modelado para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo [19].

Nombre: Notación para el Modelado de Procesos de Negocio (BPMN por sus siglas en inglés) 2.0. **Descripción de su uso:** proporciona a empresas y organismos, la capacidad de entender a través de la modelación sus procedimientos internos de negocios en una notación gráfica y darán a las organizaciones la capacidad de comunicar estos procedimientos de manera estándar. Además, la notación gráfica facilitará la comprensión de las colaboraciones de rendimiento y las transacciones comerciales entre las organizaciones. Esto asegurará que las empresas se entiendan a sí mismas y a los participantes en sus negocios y permitirá a las organizaciones a adaptarse a las nuevas circunstancias de negocio internos [20].

Nombre: SQL Lenguaje de Consultas Estructurado (SQL por sus siglas en inglés) del 2012. **Descripción de su uso:** lenguaje declarativo de alto nivel que por sus siglas significan lenguaje de consultas estructurado, un tipo de lenguaje vinculado con la gestión de bases de datos de carácter relacional, que permite la especificación de distintas clases de operaciones entre estas. En esencia, el SQL es un lenguaje declarativo de alto nivel ya que, al manejar conjuntos de registros y no registros individuales, ofrece una elevada productividad en la codificación y en la orientación a objetos. Una sentencia de SQL puede resultar

equivalente a más de un programa que emplee un lenguaje de bajo nivel. Se habla por tanto de un lenguaje normalizado que permite trabajar con (PHP) en combinación con cualquier tipo de base de datos (MS Access, SQL Server, MySQL) [21].

1.3.2 Herramientas

Las herramientas informáticas son programas, aplicaciones o simplemente instrucciones usadas para efectuar otras tareas de modo más sencillo. Cada herramienta se crea y diseña para una o varias funciones determinadas, por lo que existen diversos tipos de herramientas informáticas según el campo al que se dediquen.

Nombre: entorno de desarrollo integrado NetBeans 8.0. **Licencia del producto:** doble licencia; licencia de desarrollo y distribución común (CDDL) y licencia pública general de GNU versión 2 con excepción de rutas de clases GNU *General Public License* versión 2 (GPL2). **Descripción de su uso:** NetBeans es un Entorno de Desarrollo Integrado (IDE por sus siglas en inglés) escrito en Java, de código abierto, gratuito para desarrolladores de software. Ofrece todas las herramientas necesarias para crear aplicaciones profesionales, empresariales, web y móviles con el lenguaje Java, JavaFX, C/C ++ y lenguajes dinámicos como PHP, JavaScript, Groovy y Ruby, es fácil de instalar y se puede ejecutar tanto en Windows, como en Linux, Mac OS X y Solaris. NetBeans ofrece una versión del IDE hecho a medida para el desarrollo de sitios web en PHP que comprenden una variedad de secuencias de comandos y lenguajes de marcado; se integra dinámicamente con HTML, JavaScript y CSS; cuando se configura el proyecto, se ejecuta en el servidor web incorporado a PHP; es totalmente compatible con el desarrollo iterativo, por lo que las pruebas proyectos PHP sigue los patrones clásicos familiares para los desarrolladores web [22].

Nombre: Evolus Pencil. **Licencia del producto:** se encuentra bajo licencia de código abierto GPL (GNU *General Public License*). **Descripción de su uso:** herramienta de prototipo libre y de código abierto con interfaz gráfica de usuario, que se instala con facilidad y crea maquetas de plataformas de escritorio populares. Evolus Pencil brinda disponibilidad para todas las plataformas. Los proyectos se pueden exportar en los formatos HTML, PNG¹⁰, documento Open office, documento de Word y PDF. Permite instalar plantillas definidas por el usuario, operaciones de dibujo estándar: alinea, escala y rota, y la adición de los objetos externos [23].

¹⁰ **PNG** (siglas en inglés de Gráficos de Red Portátiles, pronunciadas "ping") es un formato gráfico basado en un algoritmo de compresión sin pérdida para bitmaps no sujeto a patentes.

Nombre: Suite de Visual Paradigm 8.0. **Descripción de su uso:** Visual Paradigm es una herramienta UML. La herramienta está diseñada para una amplia gama de usuarios, incluyendo ingenieros de software, analistas de sistemas, analistas de negocios y arquitectos de sistemas, o para cualquier persona que esté interesada en la construcción de forma fiable de sistemas de software a gran escala con un enfoque orientado a objetos. Además, Visual Paradigm soporta los últimos estándares de la notación UML. Visual Paradigm International Limited es un proveedor líder de soluciones de software que permiten a las organizaciones para desarrollar aplicaciones de calidad más rápido, mejor y más barato. Visual Paradigm se dedica a desarrollar continuamente y la entrega de software, servicios y alianzas para ayudar a los clientes a transformar con precisión sus requisitos del sistema en soluciones de software de alta calidad, todo ello con el mínimo riesgo y el máximo retorno de la inversión. Todos los productos de Visual Paradigm están diseñados y desarrollados para eliminar la complejidad, mejorar la productividad y comprimir el desarrollo de software según plazos emitidos por los clientes [24].

Nombre: PostgreSQL 9.4.1. **Licencia del producto:** distribuida bajo la licencia de PostgreSQL, un liberal de licencia de código abierto, similar a las licencias BSD¹¹ o MIT¹². **Descripción de su uso:** PostgreSQL es un sistema de gestión de bases de datos objeto-relacional. PostgreSQL es un descendiente de código abierto de este código original de la Universidad de California en Berkeley. Es compatible con una gran parte del estándar SQL y ofrece muchas características modernas como: consultas complejas, claves externas, disparadores, vistas actualizables e integridad transaccional. También, PostgreSQL se puede extender por el usuario en muchas maneras, por ejemplo, mediante la adición de nuevos tipos de datos, funciones, operadores, funciones de agregado, métodos de índice y lenguas de procedimiento. Debido a la licencia liberal, PostgreSQL puede ser utilizado, modificado y distribuido por cualquier persona de forma gratuita para cualquier propósito, ya sea privada, comercial o académico [25].

Nombre: PgAdmin III 1.14.0. **Licencia del producto:** licencia BSD. **Descripción de su uso:** es una aplicación de diseño y manejo de bases de datos para su uso con PostgreSQL, de código abierto y se considera la más completa entre las de su tipo. La aplicación se puede utilizar para manejar PostgreSQL 7.3 y superiores, funciona sobre casi todas las plataformas. PgAdmin III por su diseño, responde a las necesidades de todos los usuarios, desde escribir consultas SQL¹³ simples hasta desarrollar bases de datos

¹¹La licencia BSD es la licencia de software otorgada principalmente para los sistemas BSD (*Berkeley Software Distribution*).

¹²La licencia MIT es una de tantas licencias de software que ha empleado el Instituto Tecnológico de Massachusetts (MIT).

¹³**SQL** (*Structured query language*) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas.

complejas. La interfaz gráfica soporta todas las características de PostgreSQL y facilita enormemente la administración. Incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados, por citar algunos. La conexión al servidor se hace mediante conexión TCP/IP¹⁴ y puede cifrada mediante SSL¹⁵ para mayor seguridad [26].

Nombre: Apache 2.4.7. **Descripción de su uso:** Apache es software libre y un proyecto de la Fundación de Software Apache, con el objetivo de suministrar un servidor seguro, eficiente y extensible que proporcione servicios HTTP¹⁶ en sincronía con los estándares HTTP actuales. Apache es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos HTTP. Es multiplataforma y modular, puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona y con la API¹⁷ de programación de módulos, para el desarrollo de módulos específicos. Además, es extensible ya que gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor. El Servidor HTTP Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras [27].

Nombre de la herramienta: Apache JMeter 2.3.1. **Descripción de su uso:** Jmeter es un proyecto de Apache que puede ser utilizado como una herramienta de prueba de carga para analizar y medir el desempeño de una variedad de servicios, con énfasis en aplicaciones web. Jmeter puede ser usado como una herramienta de pruebas unitarias para conexiones de bases de datos con FTP, LDAP, Servicios web, HTTP y conexiones TCP genéricas. Apache Jmeter es una aplicación de escritorio de código abierto diseñada para cargar el comportamiento funcional de prueba y medir el rendimiento. Originalmente fue creada para probar aplicaciones web, pero se ha expandido a otras funciones de prueba. Se puede utilizar para simular una carga pesada en el servidor, de red o un objeto para probar su resistencia o para analizar el rendimiento general bajo diferentes tipos de carga [28].

¹⁴**TCP/IP** es un conjunto de protocolos de red en los que se basa Internet y que permiten la transmisión de datos entre computadoras.

¹⁵Secure Sockets Layer o SSL es un protocolo criptográfico que proporciona comunicaciones seguras por una red, comúnmente Internet.

¹⁶**HTTP** (*Hypertext Transfer Protocol*) es el protocolo usado en cada transacción de la *World Wide Web*.

¹⁷**API** (*Application Programming Interface*) es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

1.3.3 Marco de trabajo

En el desarrollo de software, un marco de trabajo es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software [29].

Nombre: GUUD 2.0. **Descripción de su uso:** marco de trabajo desarrollado por el Departamento de Desarrollo de la Dirección de Informatización para la creación de aplicaciones web escritas en PHP. Constituye un híbrido entre el marco de trabajo de PHP CodeIgniter y la librería de JavaScript, JQuery. Implementa el patrón Modelo-Vista-Controlador (MVC) así como la programación orientada a aspectos. Realiza un manejo de excepciones y mensajes. Este hace modular el marco de trabajo CodeIgniter e implementa como estrategia de comunicación entre módulos la Inversión de Control (IoC). Hace uso de plantilla para el renderizado de las vistas. Contiene una serie de componentes visuales que permiten la fácil interacción del usuario con la aplicación tales como el calendario y el *grid*, entre otros [30].

Nombre: CodeIgniter 1.7.3. **Descripción de su uso:** CodeIgniter es un marco de trabajo para el desarrollo de aplicaciones, una herramienta para personas que crean webs usando PHP. Es de código abierto. Su meta es permitir desarrollar proyectos mucho más rápido que si se hiciera escribiendo el código desde cero, proporcionando una gran variedad de librerías para las tareas más corrientes, así como una interfaz simple y una estructura lógica para acceder a estas librerías. CodeIgniter contiene una serie de librerías que sirven para el desarrollo de aplicaciones web y además propone una manera de desarrollarlas que debemos seguir para obtener provecho de la aplicación. CodeIgniter implementa el proceso de desarrollo llamado Modelo-Vista-Controlador (MVC), que es un estándar de programación de aplicaciones, utilizado tanto para hacer sitios web como programas tradicionales [31].

Nombre: JQuery 1.9.2. **Licencia del producto:** posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2. **Descripción de su uso:** JQuery es una biblioteca de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM¹⁸, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX¹⁹ a páginas web. Es software libre y de código abierto. JQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio [32].

¹⁸**DOM** (*Document Object Model*) es esencialmente una interfaz de programación de aplicaciones que proporciona un conjunto estándar de objetos para representar documentos HTML y XML.

¹⁹**AJAX** (*Asynchronous JavaScript And XML*) es una técnica de desarrollo web para crear aplicaciones interactivas.

1.4 Conclusiones parciales

El estudio de los sistemas y los conceptos teóricos permitió sentar las bases teóricas para comprender y analizar la gestión de información que integran y automatizan muchas de las prácticas de negocio asociadas con los aspectos operativos o productivos de las empresas vinculadas a la gestión personal, a través de diseños más adaptables y seguros que incorporan condiciones requeridas. El análisis del ambiente de desarrollo permitió establecer la metodología, las herramientas, lenguajes y tecnologías a utilizar en el desarrollo del sistema de acuerdo a las características deseadas por el Departamento de Desarrollo de la DIN. Esto permite crear una solución multiplataforma para el desarrollo de una solución informática que facilite la gestión de la información del componente Personal en la UCI.

2. Capítulo 2: Descripción de la propuesta de solución

En el capítulo se exponen las características de la propuesta de solución, iniciándose con el análisis del modelo de negocio para una mejor comprensión del entorno de la investigación. Se presenta una descripción detallada de los requisitos funcionales y no funcionales, así como las técnicas para su obtención. Se describe la arquitectura, el diseño del sistema y los patrones utilizados. Se muestra el diagrama lógico de datos, las principales vistas del sistema y el diagrama de despliegue.

2.1 Procesos del negocio

Los procesos de negocio son un conjunto estructurado y medible de actividades diseñadas para producir un producto especificado para un cliente o mercado específico. Implica un fuerte énfasis en cómo se ejecuta el trabajo dentro de la organización [33].

Selección e integración del personal (contratación)

Cuando un área de la Universidad necesite ocupar una plaza vacante, debe presentar a la Dirección de Capital Humano (DCH) los fundamentos necesarios que validen la necesidad de ocupar esta plaza, esto debe realizarse durante la última semana del mes, donde se recogerá una plantilla de cargos u ocupaciones en la Dirección de Capital Humano. El Sistema de Gestión Universitaria debe facilitar a la DCH avalar la plaza vacante para ser ocupada. Posteriormente, se presentará a la rectora por el SGU la autorización a ocupar la plaza vacante, proceso que se desarrollará en la primera semana del mes. Si la necesidad de ocupar una nueva plaza en la plantilla de cargos u ocupaciones de una dirección, es autorizada por la rectora, se procede como se explica a continuación.

Cuando resulte necesario cubrir una plaza definitiva o temporalmente vacante o de nueva creación, la dirección del área valora la posibilidad de ocuparla con personal declarado disponible de la misma entidad, en caso de que exista y que reúna los requisitos establecidos. Capital Humano (CH) lanza la convocatoria de las plazas aprobadas en puntos de mucha fluctuación de trabajadores. La duración de la convocatoria según el convenio colectivo de trabajo será de 7 días de publicación. La convocatoria debe estar firmada por CH y el sindicato de la Universidad [34].

Las solicitudes de las personas que van a optar por la plaza vacante debe ser entregadas en la DCH de la Universidad, quien se encargará de realizar las revisiones pertinentes para determinar los candidatos que reúnen los requisitos para ocupar la plaza y confecciona un listado de aspirantes por cada plaza que será entregado al director de cada área con las plazas en convocatorias para ser analizados en el Comité de

Descripción de la propuesta de solución

Expertos. El trabajador que va a efectuar la solicitud de alguna plaza, debe informarlo previamente a su jefe inmediato superior. El jefe se reserva el derecho de retener al trabajador por el término del tiempo que establecen las resoluciones vigentes. En caso de ser trabajador de la Universidad se realiza la verificación de los requisitos en el expediente laboral. El Departamento de Recursos Laborales de la DCH debe llenar la ficha para la verificación de los requisitos del trabajador. Para llenar el punto en la ficha, si la persona que solicita la plaza es trabajador de la Universidad, la DCH debe verificar con la Dirección Jurídica y la dirección del área a la que pertenece el trabajador, si es objeto de medida disciplinaria aplicada o en proceso. Una vez determinadas las personas que reúnen los requisitos para ocupar la plaza vacante, estas son informadas al área para que la Comisión de Expertos continúe el proceso de selección. La Comisión de Expertos del área se encarga de sugerir al jefe del área las personas idóneas para ocupar la plaza. La verificación por el Grupo de Selección y Reclutamiento de la DCH realiza, en caso de ser externo a la Universidad. Si es escogido se procede a realizar la verificación en las organizaciones de masas a la cual pertenece la persona con ayuda de los Órganos del Ministerio del Interior (MININT), donde se determina si la persona esta apta para ocupar la plaza [34].

Solicitudes de visitantes

La Oficina de Identificación en la Universidad, es la encargada de gestionar la identificación para las personas que estudien, trabajen, vivan o la visiten. El control de la asignación de credenciales para las personas que visitan el Centro se desarrolla teniendo en cuenta los tipos de identificación para visitantes:

- Identificación personal: es emitida cuando los visitantes se presentan en la Oficina de Identificación y se imprime con los datos biométricos y biográficos. Luego de esta impresión en el Sistema de Identificación y en el SGU quedan registrados sus datos con el número de serie y de identificación. Este registro posibilita que se habiliten para el visitante servicios de la Universidad, tales como: acceso a comedores, servicios telemáticos, etc.
- Identificación impersonal: es emitida cuando los visitantes no se presentan en la Oficina de Identificación, en este caso, la identificación es emitida sin datos biométricos y biográficos. Esta situación ocurre cuando la cantidad de visitantes que acceden a la UCI al mismo tiempo es elevada; en esos momentos la capacidad operacional de la Oficina de Identificación es insuficiente para prestar un servicio con agilidad.

Actualmente, las áreas solicitan a la Oficina de Identificación la emisión de documentos de identificación para visitantes que deben permanecer en la UCI por un periodo. La solicitud de identificaciones por parte

Descripción de la propuesta de solución

de las áreas es realizada usando el correo electrónico. Luego, la Oficina de Identificación emite y entrega al área la cantidad de identificaciones solicitadas, en este caso las identificaciones son de tipo impersonal debido a que la Oficina no conoce los datos biográficos y biométricos de las personas y la entrega se realiza conformando un documento digital con los seriales y los números de cada identificación entregada. Después, las identificaciones son entregadas por el área a cada visitante, y el registro de cuál persona tiene qué número de credencial se registra en un procesador de texto. Este documento es guardado por la dirección del área hasta que las identificaciones son recogidas al terminar el periodo de estancia de los visitantes en la Universidad. Una vez que se entregan las identificaciones a los visitantes, los servicios de acceso a comedores, acceso a la Universidad, servicios telemáticos, etc.; son activados para estas personas y retirados una vez que terminan su estancia.

Licencia no retribuida

En el sector estatal a solicitud del trabajador, el jefe de la entidad puede conceder licencias no retribuidas a trabajadores, estas deben determinarse por escrito con la fecha de inicio y conclusión de la licencia concedida periodo el cual no debe excederse de 3 meses y en casos muy particulares desde 6 meses hasta un año. En el periodo que el trabajador decida y el jefe de área lo autorice se le descuentan los días de vacaciones si este tiene alguna acumulada a las licencias no retribuidas. Por ejemplo, un trabajador que pida un mes de licencia y tenga 15 días de vacaciones no se le pagarán los días restante solamente. El directivo del área puede otorgar la licencia de acuerdo con la causa planteada por el trabajador o puede reservarse el derecho de no concederla, esto último no se aplica a licencias no retribuida a mujeres que han terminado el proceso de prestación social por concepto de licencia de maternidad recogido en el Reglamento del Decreto-Ley No. 234 de 13 de agosto del 2003, “De la Maternidad de la Trabajadora” [35].

Las causas por las que un trabajador puede pedir licencia no retribuida según la Universidad de las Ciencias Informáticas, se recogen en tres claves fundamentales. La primera clave vinculada con las licencias no retribuidas, tiene que ver con las licencias de maternidad post prestación social anteriormente planteado, la segunda por hijos menores de 16 en el cual la madre o el padre tienen que hacerse cargo del cuidado del hijo al que se le ha suspendido la escuela, el círculo infantil o esté enfermo y no exista otro familiar que pueda sustituirle, la tercera y última clave se debe a problemas personales en esta se recogen las siguientes normativas: salida del país están impedidos de concurrir al trabajo, por haber perdido la vivienda, como consecuencia de desastres; se encuentran evacuados, trasladados a otra vivienda o requieran permanecer en la suya para su protección, como consecuencia de inundación, incomunicación y otras causas asociadas a desastres; viajes al exterior por asuntos particulares, a trabajadores que laboran en la reconstrucción de

sus viviendas por esfuerzo propio, afectada como consecuencia de desastres, se suspende temporalmente la relación de trabajo, independientemente del contrato de trabajo que tengan suscrito con la entidad, por el periodo que se determine entre el jefe de la entidad y el trabajador. Durante el periodo en que se suspende la relación de trabajo, se le concede al trabajador una licencia no retribuida y se suscribe un suplemento al contrato de trabajo [35].

Los procesos antes descritos se especifican en los diagramas de procesos de negocios recogidos en el **Anexo 2**.

Licencia de maternidad

Instrucción No. 8/2004. Mediante la Resolución No. 22 de 23 de octubre del 2003 se estableció el Reglamento del Decreto-Ley No. 234 de 13 de agosto del 2003, "De la Maternidad de la Trabajadora". En el cumplimiento del apartado Tercero de la mencionada Resolución es necesario dictar la presente Instrucción con el objetivo de disponer lo siguiente: **PRIMERO**: poner en vigor las Aclaraciones sobre la nueva legislación de maternidad, para su mejor aplicación. **SEGUNDO**: para la implementación del Decreto-Ley citado en las Unidades Básicas de Producción Cooperativa, se mantendrá lo establecido la Resolución No. 10 de 28 de septiembre de 1998 en su Apartado Cuatro, en cuanto a la base de cálculo, en el sentido de que sólo se podrán tomar las cantidades netas devengadas por la cooperativista por concepto de anticipo, para determinar la cuantía del ingreso a tener en cuenta para efectuar el cálculo de la prestación económica por maternidad [36]. La trabajadora que se encuentra embarazada entrega el certificado de Licencia Prenatal (de 34 semanas) al técnico del Grupo Sistema de la Dirección de Desarrollo de Capital Humano, quien procesa el certificado para el pago de la maternidad. Al técnico deben entregarle el certificado de postnatal, cuando la trabajadora dé a luz. Posteriormente se debe entregar el certificado de segunda postnatal una vez vencido la primera postnatal (a los 42 días). Al vencimiento del periodo de la licencia de postnatal, la madre y el padre decidirán y comunicarán por escrito a sus respectivas administraciones, dentro del término de los 60 días naturales posteriores, cuál de ellos cuidará a su hijo o hija hasta su primer año de vida. La trabajadora se acoge a la Ley de Prestación Social hasta que su hijo cumpla el año. Al técnico del Grupo Sistema de la Dirección de Desarrollo de capital Humano deben entregarle una carta donde se explique la necesidad de la trabajadora de acogerse a la Ley de Prestación Social hasta que su hijo cumpla el año y le otorguen al círculo infantil. Una vez que la trabajadora se acoge a la Ley de Prestación Social el técnico le realiza los cálculos pertinentes para el pago de la licencia de maternidad hasta que el niño cumpla el año [36]. Para una mayor comprensión del proceso se muestra el diagrama del proceso de negocio en la Figura 2, donde el SGU automatizará los procesos de color gris.

Descripción de la propuesta de solución

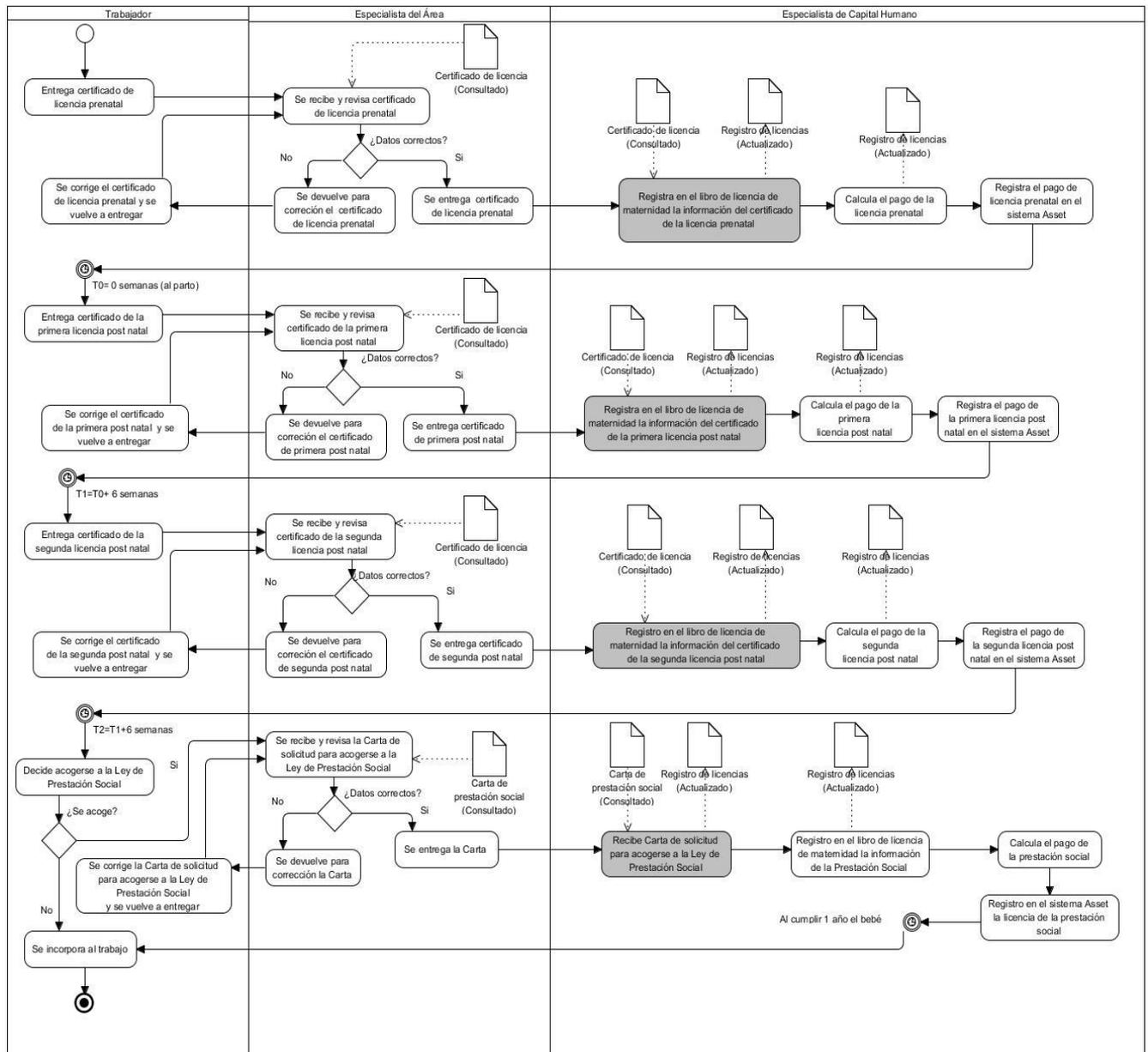


Figura 2. Diagrama de proceso de negocio de licencia de maternidad

2.1.1 Diagrama de clases del dominio (Reportes)

La presente propuesta, tiene como función la de obtener y filtrar información gestionada por otros componentes del SGU, actualmente no existe un negocio bien definido para la gestión de reportes en el componente Personal. Lo anterior, unido a que este sistema está igualmente en desarrollo, ha propiciado que se haga un modelado de dominio en lugar del negocio. El objetivo del modelado del dominio es capturar,

Descripción de la propuesta de solución

comprender y describir las clases más importantes dentro del contexto del sistema. El modelo del dominio y diccionario de clases del dominio, ayuda a los usuarios, clientes, desarrolladores y otros interesados a utilizar un vocabulario común. En la Figura 3 se muestra el diagrama de clases del dominio.

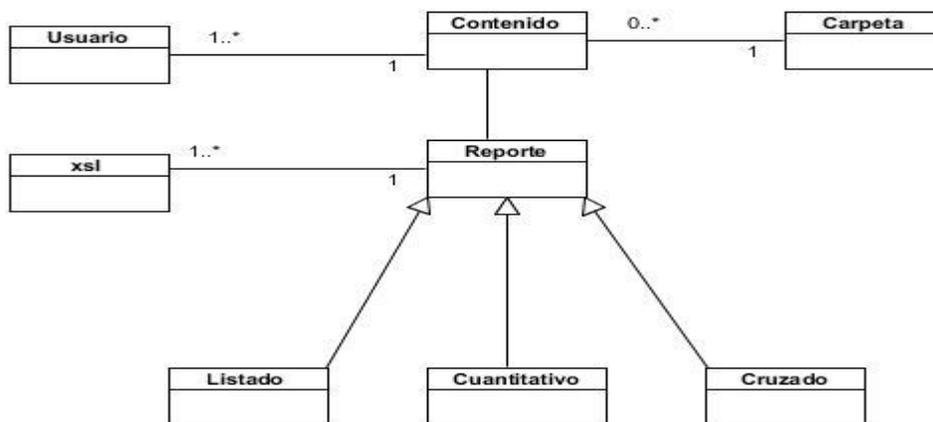


Figura 3. Diagrama de clases del dominio

El diccionario de clases del dominio describe textualmente las clases identificadas durante el modelado del dominio del problema. Este diccionario sirve como glosario de términos y se muestra a continuación:

Usuario: representa a todas las personas autenticadas en el sistema y que acceden al componente con el fin de crear determinado tipo de contenido. *Contenido*: elemento que puede ser gestionado por los usuarios en el sistema. *Reporte*: tipo de contenido que representa un informe que organiza y exhibe la información necesitada por el usuario. Su función es aplicar un formato determinado a los datos para mostrarlos por medio de un diseño atractivo y que sea fácil de interpretar. *Carpeta*: tipo de contenido que representa un espacio donde el usuario puede almacenar contenidos siguiendo una estructura jerárquica. *Listado*: tipo de reporte que permite obtener un listado con información nominal de personas que cumplen con determinadas especificidades. *Cuantitativo*: tipo de reporte para realizar conteos de personas por tantos conceptos²⁰ como el usuario desee. *Cruzado*: tipo de reporte que establece la relación existente entre dos conceptos. *XSL*: hojas de estilo que realizan la transformación del reporte utilizando una o varias reglas de plantilla, dichas reglas son interpretadas por un procesador de XSL²⁰ el cual realiza las transformaciones necesarias colocando el resultado en un archivo de salida.

2.1.2 Bases legales

- ✓ Decreto-Ley No. 116 “Código de Trabajo”. 20 de diciembre de 2013.

²⁰En este contexto se refiere campos en la base de datos.

- ✓ Decreto-Ley No. 326/14 “Reglamento del Código de Trabajo”. 12 de junio de 2014.
- ✓ Decreto-Ley No. 234 “De la Maternidad de la Trabajadora”. 13 de agosto de 2003.
- ✓ Decreto-Ley No. 105 del Ministerio de Trabajo y la Seguridad Social.

2.1.3 Reglas del negocio

Las reglas de negocio son condiciones, estándares o reglas que deben ser cumplidas y controladas por la organización durante el flujo del proceso [37]. El proceso de gestión de información de la versión 2.0 del componente Personal presenta reglas que permiten su ejecución con la mayor calidad.

- ✓ Licencia de maternidad
 - Las fechas de inicio no puede solapar las fechas de fin de los periodos prenatal, primera posnatal, segunda posnatal y prestación social y viceversa.
 - Solo se permite la inserción de datos en los dos primeros campos fechas del periodo prenatal de la licencia de maternidad, luego del registro de esos datos es que se puede acceder a la introducción de las fechas en el periodo siguiente y así sucesivamente.
 - Los hombres pueden tener licencias de maternidad, donde se sobreentiende que tienen derecho solo a la prestación social.
- ✓ Licencia no retribuida
 - El directivo de cada área es el que decide si se le otorga la licencia al trabajador.
 - La fecha está comprendida hasta tres meses y de seis meses a un año en caso excepcionales como es el caso de una mujer que ha cumplido su prestación social, la ley la ampara.
- ✓ Solicitudes de visitantes
 - La fecha de entrada y salida del visitante no pueden estar solapadas.
 - Los visitantes a la universidad pueden necesitar o no alojamiento en la misma.
 - Todo visitante debe tener una credencial asociada.

2.2 Requisitos del sistema

Los requisitos para un sistema de software determinan lo que debe hacer el sistema y definen las restricciones en su funcionamiento e implementación, es decir, lo que el software debe hacer y bajo qué circunstancias debe hacerlo [38]. La captura de requisitos es un paso fundamental para saber exactamente lo que debe hacer y las cualidades o propiedades que deben tener los componentes definidos. Para el desarrollo de la propuesta de solución se identificaron requisitos funcionales y no funcionales, siendo los primeros guiados mediante técnicas de obtención que permiten establecer una comunicación más específica entre el cliente y el equipo de desarrollo.

2.2.1 Definición de las técnicas de obtención de requisitos

La obtención de requerimientos es una de las cuatro actividades que define la Ingeniería de Requisitos, para ayudar a reconocer la importancia que tiene una especificación y administración adecuada de los requerimientos de los usuarios. La obtención de requisitos es el proceso donde los interesados en un sistema de software descubren, revelan y entienden sus requerimientos, evidenciando la necesidad de trabajar con el cliente debido a que sus requisitos abarcan las tareas que se necesitan para llevar a cabo el desarrollo del sistema [11]. En la identificación de los requisitos de la propuesta de solución se utilizaron las siguientes técnicas:

- **Entrevista:** se utilizó con el fin de obtener diversas opiniones y recomendaciones en un intercambio mediante preguntas con los especialistas del SGU, para esclarecer con precisión el funcionamiento del desarrollo del componente Personal. Ver en el **Anexo1**.
- **Prototipos:** con el empleo de esta técnica se realizaron las vistas del sistema, mediante la construcción de prototipos. Esto derivó en un conjunto de ideas entre los clientes y el equipo de desarrollo para un mejor entendimiento de la solución. Además, permitió formalizar la aceptación previa de los requisitos.

2.2.2 Definición de requisitos funcionales

Los requerimientos funcionales de un sistema describen lo que el sistema debe hacer. Dependen del tipo de software que se desarrolle, de los posibles usuarios y del enfoque general de la organización tomado al redactar los requerimientos [39].

Para guiar el establecimiento de la complejidad y la prioridad de los requisitos funcionales del componente Personal, se utilizó el documento del expediente de proyecto del Núcleo, denominado Evaluación de

Descripción de la propuesta de solución

requisitos del Sistema de Gestión Universitaria. En la siguiente tabla se muestran los requerimientos funcionales definidos para la propuesta de solución con la complejidad y la prioridad para el cliente.

Tabla 1. Listado de requisitos funcionales

No.	Requisito	Complejidad	Prioridad para el cliente
Licencia configuración			
RF1	Listar especialista	Baja	Alta
RF2	Crear especialista	Baja	Alta
RF3	Modificar especialista	Baja	Alta
RF4	Ver detalles especialista	Baja	Alta
Licencia de maternidad			
RF5	Listar licencia de maternidad	Media	Alta
RF6	Crear licencia de maternidad	Baja	Media
RF7	Adicionar periodos de licencia de maternidad	Media	Alta
RF8	Ver detalles de licencia de maternidad	Media	Alta
RF9	Interrumpir licencia de maternidad	Media	Alta
Licencia no retribuida			
RF10	Listar licencia no retribuida	Media	Alta
RF11	Crear licencia no retribuida	Media	Alta
RF12	Modificar licencia no retribuida	Media	Alta
RF13	Ver detalles licencia no retribuida	Baja	Alta
RF14	Interrumpir licencia no retribuida	Media	Media
Solicitudes visitantes			
RF15	Listar solicitudes visitantes	Media	Media

Descripción de la propuesta de solución

RF16	Crear solicitud visitante	Media	Media
RF17	Configurar solicitudes	Baja	Media
RF18	Dar entrada al visitante	Media	Media
Salida visitante			
RF19	Listar salidas visitantes	Media	Media
RF20	Dar salida al visitante	Baja	Media
RF21	Modificar credencial al visitante	Baja	Media
RF22	Ver detalles visitante	Baja	Baja
Reportes			
RF23	Crear directorio de reportes	Baja	Media
RF24	Renombrar directorio de reporte	Baja	Media
RF25	Abrir directorio de reportes	Baja	Baja
RF26	Eliminar directorio de reporte	Baja	Media
RF27	Ver detalles de directorio de reporte	Baja	Media
RF28	Copiar directorio de reporte	Baja	Media
RF29	Pegar directorio de reporte	Baja	Media
RF30	Cortar directorio de reporte	Baja	Media
RF31	Mostrar directorio de reporte	Baja	Media
RF32	Cerrar directorio de reporte	Baja	Media
RF33	Crear reporte estadístico	Alta	Alta
RF34	Modificar reporte estadístico	Media	Alta

Descripción de la propuesta de solución

RF35	Ver detalles de reporte estadístico	Alta	Alta
RF36	Mostrar reporte estadístico	Alta	Alta
RF37	Crear reporte cuantitativo	Alta	Alta
RF38	Modificar reporte cuantitativo	Media	Alta
RF39	Mostrar reporte cuantitativo	Alta	Alta
RF40	Ver detalles de reporte cuantitativo	Media	Media
RF41	Crear reporte de listado	Alta	Alta
RF42	Modificar reporte listado	Media	Alta
RF43	Mostrar reporte listado	Alta	Alta
RF44	Ver detalles de reporte de listado	Media	Media
RF45	Copiar reporte	Baja	Media
RF46	Pegar reporte	Baja	Media
RF47	Cortar reporte	Baja	Media
RF48	Eliminar reporte	Baja	Media
Selección e integración del personal (contratación)			
RF49	Listar solicitudes de plazas	Media	Alta
RF50	Crear solicitud de plaza	Alta	Alta
RF51	Modificar solicitud de plaza	Media	Media
RF52	Cancelar solicitud de plaza	Baja	Baja
RF53	Crear persona no existente en el sistema	Media	Alta

Descripción de la propuesta de solución

RF54	Seleccionar solicitud de plaza para verificación	Media	Alta
RF55	Verificar la solicitud de plaza	Baja	Media
RF56	Aprobar solicitud de plaza	Baja	Media
RF57	Ver detalles de solicitud de plaza	Baja	Baja
RF58	Denegar solicitud de plaza	Baja	Baja
RF59	Listar plazas para convocatoria	Baja	Media
RF60	Crear plaza para convocatoria	Media	Media
RF61	Modificar plaza para convocatoria	Baja	Baja
RF62	Ver detalles de plaza para convocatoria	Baja	Baja
RF63	Reporte por estado de solicitud de plaza	Alta	Alta

2.2.3 Definición de requisitos no funcionales

Los requisitos no funcionales imponen restricciones en el diseño o la implementación. Son propiedades o cualidades que el producto debe tener [39]. En la Tabla 2 se muestran los requisitos no funcionales definidos para la propuesta de solución.

Tabla 2. Listado de requisitos no funcionales

Usabilidad	
RNF1	Desarrollar una solución web para los procesos de gestión que se desplegará en una red cerrada.
RNF2	Solo se mostrarán a los usuarios aquellas acciones o informaciones a las que por su responsabilidad o rol dentro del negocio necesitan acceder mostrando en la vista mediante iconos y menús el acceso.
RNF3	La información que se registra estará protegida de modificaciones no deseadas de acuerdo a la estrategia de seguridad definida.
RNF4	Las vistas del sistema deben indicar en cada momento la acción que se está realizando así como lo iconos deben estar representados por una imagen acorde a la acción que se realiza.
RNF5	El sistema debe presentar un menú lateral y una barra de iconos flotantes que permitan el acceso rápido a la información por parte de los usuarios, aprovechando así las potencialidades de estas estructuras.
Software	
RNF6	Para el despliegue del componente se debe contar en el servidor de bases de datos con PostgreSQL 9.4, bajo el sistema operativo CentOS 6.6 o superior.

Descripción de la propuesta de solución

RNF7	Para el despliegue del componente se debe contar en el servidor de aplicaciones web con: PHP v5.6 con las librerías php5-ldap, php5-gd, php5-mcrypt, php5-pgsql, php5-xsl, php5-openssl y Apache 2.2 con el módulo <i>rewrite</i> activado.
RNF8	Para el uso del componente se requiere una PC cliente con cualquier sistema operativo, que se pueda instalar el navegador web Mozilla Firefox 26.0 o superior para el uso de la aplicación web.
Hardware	
RNF9	Para la ejecución del componente se requiere que la PC cliente tenga los siguientes componentes de hardware: Pentium 4 o superior, 512 MB RAM y 200 MB disco duro disponible como mínimo.
RNF10	La comunicación entre el cliente y el servidor de aplicaciones se realiza a través del protocolo HTTPS.
RNF11	El servidor de aplicación y el de base de datos deberán tener los siguientes componentes de hardware para su funcionamiento en el ambiente en que se encuentra desplegado en el componente Personal: microprocesador de 8 núcleos, 4GB de memoria RAM, 250 GB disco duro y una fuente de 800 W como mínimo.
Confiabilidad	
RNF12	El sistema puede permanecer inactivo durante 10 minutos. Al cumplirse el término se cerrará la sesión.
RNF13	El tratamiento de las excepciones permitirá un seguimiento hasta guardar información, acerca del lugar en que se produjo el error y de los parámetros utilizados en el sistema que lo provocaron. Cuando ocurre una excepción el sistema mostrará un mensaje explicativo del error ocurrido.
RNF14	El sistema de autenticación deberá recoger datos necesarios de los usuarios y tendrá control de identidades.
RNF15	El sistema de autenticación debe permitir auditoría de trazas de las acciones de los usuarios para lo que se proveerá algún mecanismo para la configuración dinámica y centralizada de la gestión de historiales.
RNF16	El servidor de aplicaciones y de base de datos deberá mantener una seguridad mediante firewall para proteger el código y la información.
Eficiencia	
RNF17	El sistema soportará la conexión simultánea de un promedio de 50 y un máximo de 500.
RNF18	El tiempo promedio de respuesta del sistema no excederá los 5 segundos de manera general.
Soporte	
RNF19	El componente cumplirá con las normas de codificación, conversiones para nomenclatura, bibliotecas de clase definidas para el Sistema de Gestión Universitaria en el documento SGU-NUC-010217_EC.
RNF20	Los productos de trabajo generados en el desarrollo del componente se registrarán por las pautas de configuración debidamente documentadas en el expediente de proyecto asociada a su proceso de desarrollo para las actividades de soporte.
Restricciones de diseño	
RNF21	El software deberá ser desarrollado en su totalidad con tecnologías y componentes de código abierto.
RNF22	El componente estará desarrollado con las herramientas definidas para el Sistema de Gestión Universitaria en el documento SGU-INV-010216_8_ATI_v1.0.doc

RNF23	El componente cumplirá con la arquitectura de información definida para el Sistema de Gestión Universitaria en el documento Vista de Arquitectura de Presentación.
Interfaz	
RNF24	La interfaz de usuario se guiará por la vista de arquitectura de presentación definidas en el documento Pautas de diseño XAUCE-SGU.
Requisitos de licencia	
RNF25	El sistema se caracterizará por un licenciamiento cerrado. El producto está basado en código abierto pero no está disponible bajo licenciamiento de código abierto.
Seguridad	
RNF26	El sistema será utilizado solo por los usuarios con permisos, usuario administrador, usuario registrado con permisos.
RNF27	Los errores deben mostrar la menor cantidad de detalles posible, para evitar brindar información que comprometa la seguridad e integridad del sistema.

2.2.4 Descripción de requisitos

Con el objetivo de presentar los requisitos funcionales de una manera más consistente y por ende más entendible, se realizó la descripción de cada requisito funcional. Se definieron secciones para describir detalladamente cada requisito, las acciones que se deben llevar a cabo por el usuario para cumplir con esa funcionalidad en el sistema. A continuación, se muestra en la Tabla 3 una de las descripciones de requisitos diseñadas, las restantes están accesibles en el **Anexo 3**.

Tabla 3. Descripción de requisitos (Listar licencia de maternidad)

Código	Nombre	Descripción	Complejidad	Prioridad para cliente
RFPers186	Listar personas con licencia de maternidad	<ul style="list-style-type: none"> El administrador selecciona del componente horizontal Personal del Sistema de Gestión Universitaria, luego en el menú lateral la agrupación funcional Seguridad social, la funcionalidad Licencia de maternidad. Se muestran los datos de las licencias de maternidad que estén en proceso, en estado “Iniciada” o “En proceso”. 	<i>Baja</i>	Alta
Actor		Precondición	Poscondición	

Descripción de la propuesta de solución

Especialista de CH Especialista de área	<ul style="list-style-type: none"> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. 	-
--	--	---

Prototipo



Botones e hipervínculos

Nombre	Descripción de flujo básico	Descripción de flujo alterno
Botón Buscar	<ul style="list-style-type: none"> Permite buscar en la lista de licencias de maternidad las personas que coincidan con los parámetros de búsqueda que se introduzcan. Parámetros: Nombre y apellidos, Carné de identidad, Credencial. 	-
Ícono Crear	<ul style="list-style-type: none"> Se muestra para los usuarios con rol Especialista de área. Muestra la interfaz que permite registrar licencia de maternidad. 	-
Ícono Adicionar	<ul style="list-style-type: none"> Se muestra para los usuarios con rol Especialista de CH. Se muestra para todos los elementos de la lista. 	-

Descripción de la propuesta de solución

	<ul style="list-style-type: none"> • Muestra la interfaz que permite adicionar periodos de licencia de maternidad 	
Ícono Interrumpir	<ul style="list-style-type: none"> • Se muestra para los usuarios con rol Especialista de CH. • Se muestra para todos los elementos de la lista. • Muestra la interfaz que permite Interrumpir licencia de maternidad. 	-
Ícono Ver detalles	<ul style="list-style-type: none"> • Se muestra para los usuarios con rol Especialista de CH y Especialista de área. • Se muestra para todos los elementos de la lista. • Muestra la interfaz que permite ver detalles de licencia de maternidad. 	-
Ícono Terminar	<ul style="list-style-type: none"> • Se muestra para los usuarios con rol Especialista de CH. • Se muestra para las licencias en estado “En proceso” con todos los periodos de licencia registrados. • Se muestra el mensaje: ¿Está seguro que desea realizar la acción? Si selecciona "Aceptar" pasa la solicitud de licencia de maternidad a estado “Finalizada” y se actualiza la interfaz para Lista de licencias de maternidad. 	<ul style="list-style-type: none"> • Si selecciona "Cancelar" se cierra el mensaje y se mantiene en la misma interfaz.
Comportamiento adicional		
Elemento	Nombre	Descripción del comportamiento

Descripción de la propuesta de solución

Campo de selección simple	Filtrar búsqueda	<ul style="list-style-type: none"> • Permite seleccionar el elemento por los que se puede filtrar la lista. El elemento es: Estructura. • Al ser seleccionado se muestran los campos de selección: Categoría y Estructura.
Campo de selección simple	Filtro “Categoría”	<ul style="list-style-type: none"> • Muestra todas las Categorías Activas. • Al seleccionar una categoría se muestran en el Filtro “Estructura” las estructuras que tienen como categoría la seleccionada. • Se debe organizar alfabéticamente de A-Z.
Campo de selección simple	Filtro “Estructura”	<ul style="list-style-type: none"> • Muestra la lista de las Estructuras que estén Activas. • Al seleccionar la estructura se muestran en la lista solo aquellas en la que la persona pertenezca a la estructura seleccionada. • Al cerrar el Filtro “Estructura” se cierran los dos filtros: Categoría y Estructura. • Si no hay seleccionado valor en el filtro “Categoría” solo muestra el valor <Selecione>. • Se debe organizar alfabéticamente de A-Z.
Tabla	Listado de licencias de maternidad	<ul style="list-style-type: none"> • Muestra la lista de solicitudes de licencia de maternidad donde el

Descripción de la propuesta de solución

		<p>usuario autenticado y las solicitudes de licencia tengan la misma Estructura.</p> <ul style="list-style-type: none"> • Muestra las solicitudes de licencia de maternidad que estén en estado: "Iniciada" y "En proceso". • Muestra las columnas: Foto, Nombres y apellidos, Área, Estado y Opciones (Adicionar periodo, Ver detalles, Interrumpir). • La lista se debe organizar alfabéticamente de A-Z por el nombre de las personas.
Entidades		
<ul style="list-style-type: none"> • E_Estructura • E_Persona • E_Licencia • E_Periodo • E_Estado • E_Especialista 		
Observaciones		
-		

2.3 Descripción de la propuesta de solución

A continuación, se expondrán los procesos de negocios, licencia de maternidad, licencia no retribuida, solicitudes visitantes selección e integración del personal (contratación) y el asistente de gestión de reporte.

Licencias: se propone dentro de la implementación del componente Personal la agrupación funcional *Seguridad Social* del SGU. La agrupación funcional contará con tres funcionalidades: *Licencias configuración*, *Licencias de maternidad* y *Licencias no retribuidas*, donde la funcionalidad *Licencias configuración*, permitirá la búsqueda de personas forma rápida y exacta, siguiendo criterios de búsqueda

Descripción de la propuesta de solución

por el nombre, apellidos, documento nacional de identidad o por la credencial emitida en la UCI. La funcionalidad permitirá crear permisos a usuarios asociándole a cada uno el área o las áreas a las que este tendrá privilegios de realizarle por sistema una licencia de maternidad o una licencia no retribuida a trabajadores que así lo requieran. En esta funcionalidad se podrán ver los detalles de las personas que tienen los permisos anteriormente mencionados. Se encargará además realizar las modificaciones sobre la información previamente almacenada, así como la eliminación en caso de que sea necesario.

La funcionalidad *Licencias de maternidad* permitirá llevar el control por sistema de las personas que se encuentran en este tipo de trámite, las cuales no solo pueden ser mujeres, el sistema da la posibilidad de incluir también a los hombres en caso de que la madre falleciese en el parto y su hijo quedara al cuidado del padre o por una cuestión económica si es la madre la de mayores ingresos, esta funcionalidad permitirá adicionar los periodos por los que pasa una mujer que se encuentre de licencia de maternidad, llevando el control por cada certificado presentado al área de Recursos Humanos. En caso de ser hombre solo persistirá el periodo de prestación social, dado que como es lógico, este no puede pasar por los periodos prenatal, primera y segunda posnatal por la que pasa una mujer embarazada. La funcionalidad permitirá ver detalles los datos esenciales de cada persona de licencia de maternidad, permitiendo eliminar una licencia del sistema y por último interrumpirla en caso de que una trabajadora perdiera el embarazo o falleciera.

En la funcionalidad *Licencias no retribuidas* se podrá consultar el listado de trabajadores que estén en este trámite. Estas licencias pueden ser o no aprobadas por no cumplir requisitos legales necesarias para ser otorgadas. La funcionalidad permitirá modificar los datos esenciales de la licencia en caso de ser necesario. También permitirá interrumpir una licencia en caso de que el trabajador no la necesite o por cualquier otra causa justificable. Además, en la funcionalidad se podrá finalizar la licencia archivándose en la base de datos del SGU para posteriores reportes donde se necesite saber las personas que estuvieron de licencia sin sueldo y si puede esta solicitar otra licencia no retribuida teniendo en cuenta la fecha. Este aspecto es una cuestión legal, donde el trabajador se le permite otra solicitud de este tipo de licencia si previamente ya ha pasado cierta cantidad de tiempo. Por último, se permitirá eliminar del sistema la licencia en caso de ser necesario.

Solicitudes visitantes: el control de personal relacionado con los visitantes y la acreditación regulada de estos para permanecer y recibir los servicios de la Universidad se propone en la solución como dos funcionalidades: *Solicitudes visitantes* y *Salidas Visitantes* dentro de la agrupación funcional *Identificación*. *Solicitudes visitantes* presentará un listado de los visitantes que están en la Universidad resaltando si tienen alojamiento y si este alojamiento implica apartamento independiente; también se mostrará en este listado

Descripción de la propuesta de solución

las fechas de inicio y fin de cada una de las visitas, así como el área vinculada a la visita. La funcionalidad permitirá crear una solicitud con la particularidad de buscar si esa persona previamente estuvo de visita en la Universidad, de no ser así se insertará esta persona en la base de datos del sistema, especificando fecha de inicio y fin de la visita, además del motivo de esta. En la funcionalidad presentará además un ver detalles del visitante seleccionado y la modificación y configuración de los datos en caso de que ocurriesen errores en la captura inicial de los datos.

Por su parte, la funcionalidad *Salidas visitantes* se encargará de mostrar un listado de visitantes que concluyeron su estadía en la Universidad, permitiendo dar baja al visitante que concluyó su actividad en la institución, eliminándolo del listado de la aplicación; pero no de la base de datos del sistema, por ser estos datos valiosos en caso de que esta persona regrese como visitante. Esta funcionalidad permitirá además, modificar la credencial asignada al visitante, así como mostrar los detalles del mismo.

Selección e Integración del Personal (contratación): el proceso de selección e integración del personal se realizará en la agrupación funcional *Contratación*, perteneciente al componente Personal. Esta contará con las funcionalidades *Plazas* y *Solicitud plaza*. En la funcionalidad *Plazas* se realizará el proceso de creación de una plaza, insertándose los datos necesarios para que el proceso de selección e integración del personal sea de forma eficiente, en ella se mostrará el listado de plazas solicitadas por los directivos de las áreas, especificando la responsabilidad a ocupar, el área de la universidad a la que pertenece, el número de referencia de la plaza, cuántas plazas de este tipo hay en la Universidad y de ella cuantas se encontrarán en proceso de selección o integración del personal. Esta funcionalidad es complemento de la funcionalidad *Solicitud plaza* siendo esta, la funcionalidad clave para el proceso de selección e integración del personal. Su objetivo es llevar el control y gestión de la información de la persona que está siendo verificada para el otorgamiento de una plaza solicitada, la cual atraviesa por un conjunto de etapas o estados, finalizando cuando la persona que solicitó una de las plazas vacantes podrá ser elegida para ocupar la plaza que eligió, o descartada en las etapas de selección.

Reportes: debido a que el componente Personal es un sistema integrador del SGU, se hace necesario el mejoramiento de la agrupación funcional Reportes, permitiendo hacer reportes de manera fácil e intuitiva y que garantice la seguridad y confiabilidad de la información manejada por estos. Los reportes a gestionar se clasifican en varios tipos dependiendo de las necesidades del usuario. Además, dicho módulo permitirá elaborar gráficas a partir de los datos obtenidos de los reportes generados por el usuario, así como exportar dichos datos a diferentes formatos como pdf y xls. Esta solución permite el uso de un conjunto de facilidades como la reutilización de varios componentes implementados por otros subsistemas y la eliminación de

redundancias en el código ya que la arquitectura del sistema tiene concebidos mecanismos para que cualquier componente de cualquier subsistema pueda acceder a funcionalidades y componentes dentro del SGU. Precisamente la solución que se propone utiliza elementos necesarios de otros componentes del SGU, en consecuencia, no será necesario implementar un gran número de funcionalidades. En sentido general, el módulo que se propone debe permitir a los usuarios abstraerse de conocimientos relacionados con los gestores de bases de datos y proporcionar un 100 % de confiabilidad de la información contenida en los reportes. De igual forma, debe garantizar que toda acción realizada sea ejecutada en tiempos inferiores a 5 segundos.

2.4 Descripción de la arquitectura y el diseño

La arquitectura de un software es la estructura u organización de un sistema que incluye los componentes de este, las propiedades visibles externas de esos componentes y las relaciones que existen entre ellos [40]. Es una representación que permite que un ingeniero de software analice la efectividad del diseño para cumplir con los requisitos establecidos, considere opciones arquitectónicas en una etapa en que aún resulta relativamente fácil hacer cambios al diseño y reduzca los riesgos asociados con la construcción del software.

2.4.1 Arquitectura Cliente-Servidor

La arquitectura Cliente-Servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente hace una petición de un servicio y recibe la respuesta a dicha petición; un servidor recibe y procesa la petición y devuelve la respuesta solicitada. Los clientes y servidores se comunican a través de una red de comunicaciones para realizar una o varias tareas de forma conjunta [40]. Esta arquitectura facilita la integración entre sistemas diferentes y comparte información, permitiendo por ejemplo que las máquinas ya existentes puedan ser utilizadas, pero utilizando interfaces más amenas para el usuario [40].

La Figura 4 es una representación gráfica del estilo arquitectónico. El sistema se hospedará en el servidor web de aplicaciones Apache 2.4.7 instalado en uno de los servidores de nodo central de la Universidad y se podrá acceder mediante la dirección URL <https://akademos.uci.cu>. El uso de esta arquitectura permite que los usuarios puedan acceder al sistema en cualquier dispositivo conectado a la red UCI logrando centralizar la información y facilitar el acceso a esta, sin necesidad de que todos tengan que utilizar el mismo sistema operativo.

Descripción de la propuesta de solución



Figura 4. Arquitectura Cliente-Servidor [40].

2.4.2. Patrón de arquitectura

El patrón Modelo-Vista-Controlador (MVC) es el patrón de desarrollo en el que se basa el marco de trabajo CodeIgniter, el cual es utilizado para el desarrollo del sistema. El MVC es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el componente encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado, define componentes para la representación de la información y por el otro la interacción el usuario [41].

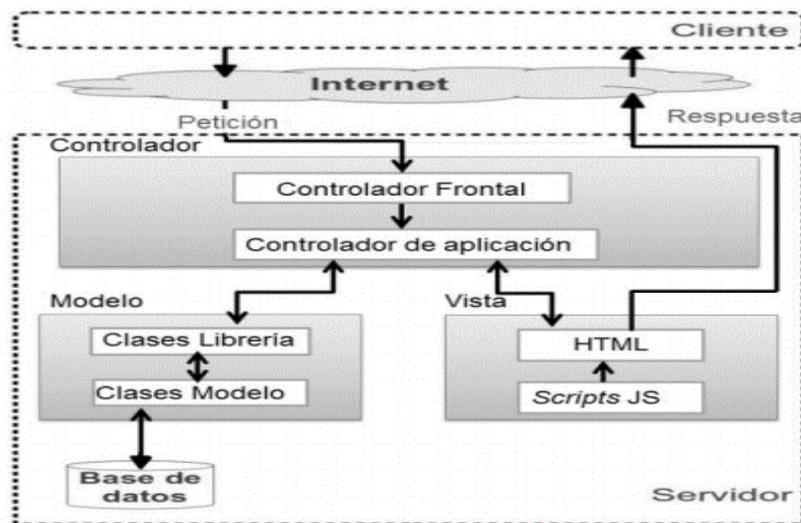


Figura 5. Patrón Modelo-Vista-Controlador en GUUD [42]

Implementación del MVC que realiza GUUD

La Figura 5 ilustra la particularidad del MVC implementado por el GUUD y que a continuación se describe. El marco de trabajo GUUD cuenta con un controlador frontal que inicializa los recursos básicos necesarios para usar el CodeIgniter (parte estructural del GUUD). Una vez realizada una petición HTTP por un cliente,

el controlador frontal se encarga de analizar la URI²¹ y a partir de esta determina cuál controlador de la aplicación (controlador de un determinado componente) debe ser cargado para atender la petición realizada. Cada controlador de la aplicación tiene asociada una o varias librerías responsables de procesar los datos e implementar la lógica del negocio inherente a las acciones relacionadas con dicho controlador. De manera similar cada librería tiene asociado uno o varios modelos encargados del acceso a los datos. Cuando un controlador de la aplicación es cargado, este examina la petición para determinar si solo debe cargar una vista determinada o si es necesario interactuar con la base de datos. En este último caso el controlador de la aplicación envía los datos recibidos a la o las librerías. Estas a su vez cargan los modelos necesarios para obtener, registrar o actualizar en la base de datos la información solicitada o enviada. Para realizar esta tarea los modelos hacen uso de la clase Active Record. Esta clase permite consultar la base de datos con mínima codificación y abstrayéndose del gestor que se use. La sintaxis de la consulta es generada por cada adaptador de base de datos. Cuando los datos son obtenidos, se retornan al controlador de la aplicación en un proceso inverso al descrito anteriormente. Posteriormente, el controlador carga estos datos a archivos escritos en HTML los cuales pueden incluir llamadas a archivos escritos en JavaScript para manejar dinámicamente su contenido o hacer uso de asistentes (helpers) para la creación de forma simplificada de código HTML. Finalmente, el resultado obtenido de todo este proceso es enviado al navegador web como respuesta a la petición inicial [42].

Modelo: es la representación de la información con la que el sistema opera, por lo que gestiona los accesos, las consultas y actualizaciones de los datos del sistema. Interactúa con los controladores de la aplicación, respondiendo sus peticiones de acceso al sistema de persistencia de información.

Vista: es la presentación de la información en el sistema. Se comunica con el controlador del que recibe los datos para su visualización y captura los eventos que desencadenan acciones de acceso a datos.

Controlador: reacciona a los eventos desencadenados por el cliente, ejecutando la acción adecuada e invoca peticiones a los modelos que necesite cuando se hace alguna solicitud sobre la información. El controlador hace de intermediario entre la vista (forma en que se presenta la información) y el modelo (proveedor de la información).

²¹Un Uniform Resource Identifier o URI (en español, Identificador Uniforme de Recurso) es una cadena de caracteres corta que identifica inequívocamente un recurso (servicio, página, documento, dirección de correo electrónico, enciclopedia, etcétera). Su principal diferencia con el Uniform Resource Identifier o URL (en español, Localizador Uniforme de Recurso) es que permite especificar que parte del recurso se solicita (segmentos).

2.4.3. Modelo de despliegue

El diagrama de despliegue se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. Describe la topología del sistema: la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos, una representación de dicha estructura se muestra en la Figura 6 [43].

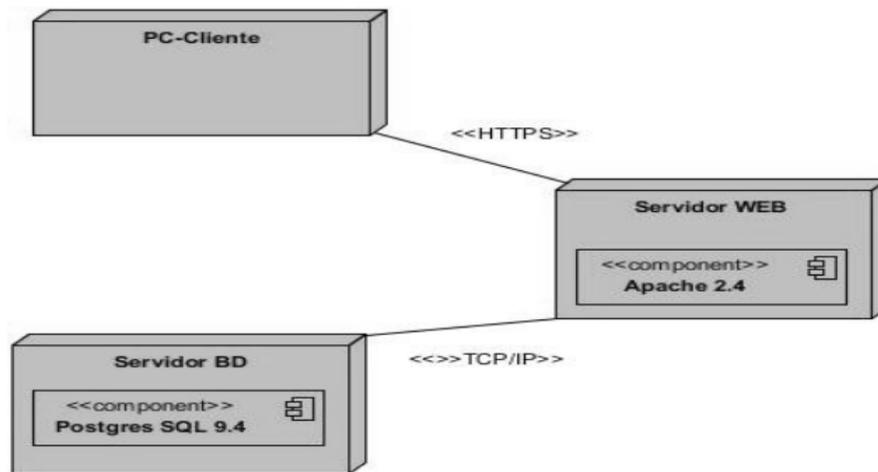


Figura 6. Representación del modelo de despliegue

Descripción de elementos e interfaces de comunicación

PC cliente: su función es acceder e interactuar con el sistema según sus necesidades. Al estar la aplicación desarrollada sobre la web, la máquina cliente necesita de muy pocas prestaciones puesto que solo necesita un navegador web para poder acceder al sistema y realizar las operaciones necesarias.

Servidor aplicaciones: aquí reside la capa de presentación del sistema, la que es accedida por las máquinas clientes a través de un navegador web. Contiene las funcionalidades del sistema.

Servidor de base de datos: es el encargado de almacenar la información que se gestiona en el del sistema.

<<HTTPS>>: el protocolo seguro de transferencia de hipertexto es un estándar de red que sigue el esquema petición-respuesta entre un cliente y un servidor. Está destinado a la transferencia segura de datos de hipertexto.

<<TCP/IP>>: la familia de protocolos de Internet es un conjunto de estándares de red en la que se basa la red global y que permite la transmisión de datos entre redes de computadoras sin importar el tamaño.

2.4.4. Patrones de diseño

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular. Cada patrón describe un problema que ocurre una y otra vez en el entorno y describe también el núcleo de la solución al problema. Esta solución provee una forma confiable, segura y probada para resolver problemas recurrentes en el diseño de software. Los patrones de diseño tienen como finalidad precisar en detalle los subsistemas y componentes de la aplicación [11].

Patrones GRASP

Los Patrones Generales de Software para Asignación de Responsabilidades (GRASP por sus siglas en inglés) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en formas de patrones. Ayudan a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. Las responsabilidades están relacionadas con las obligaciones de un objeto en cuanto a su comportamiento [44]. A continuación, se describen los patrones GRASP utilizados en el desarrollo de la propuesta de solución:

Experto: propone asignar una responsabilidad a la clase que cuenta con la información necesaria para cumplirla. Indica que la responsabilidad de la creación de un objeto debe recaer sobre la clase que conoce toda la información necesaria para crearlo. El uso del patrón se evidencia en las clases librerías, que son las que cuentan con la información necesaria para cumplir las responsabilidades sobre los elementos de negocio. Un ejemplo del uso del patrón en la propuesta de solución es la librería encargada de la información de las licencias de maternidad, la que posee las responsabilidades relacionadas con la información de las licencias de maternidad.

Creador: expresa la asignación a una clase la responsabilidad de crear una instancia de otra. El uso del patrón se evidencia en la clase *loader* que se encarga de cargar los elementos del marco de trabajo dígame, librerías, modelos. En la propuesta de solución cada clase controladora crea una instancia de las librerías que necesita para su funcionamiento a partir de la clase *loader*. En la Figura 7 se pone de manifiesto en la clase *loader*, pues esta es la responsable de la creación (o instanciación) de nuevos objetos o clases.

```

public function __construct() {
    parent::__construct();
    $this->load->library('licencia_maternidad_lib');
    $this->load->library('personas_lib');
}
public function indexListar() {
    echo $this->template->render('licencia_maternidad/listar_licencia_maternidad_view');
}
    
```

Figura 7. Patrón Creador

Controlador: define que se le debe asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. El uso del patrón se evidencia en las clases controladoras que se encargan de obtener los datos y enviarlos a las librerías y las vistas, así como de manejar los posibles errores o mensajes que se muestran. En la Figura 8, el menú de la izquierda define cual es la clase la controladora de la propuesta de solución licencia de maternidad, dicha controladora es la que recibirá las peticiones asociadas a esta funcionalidad.

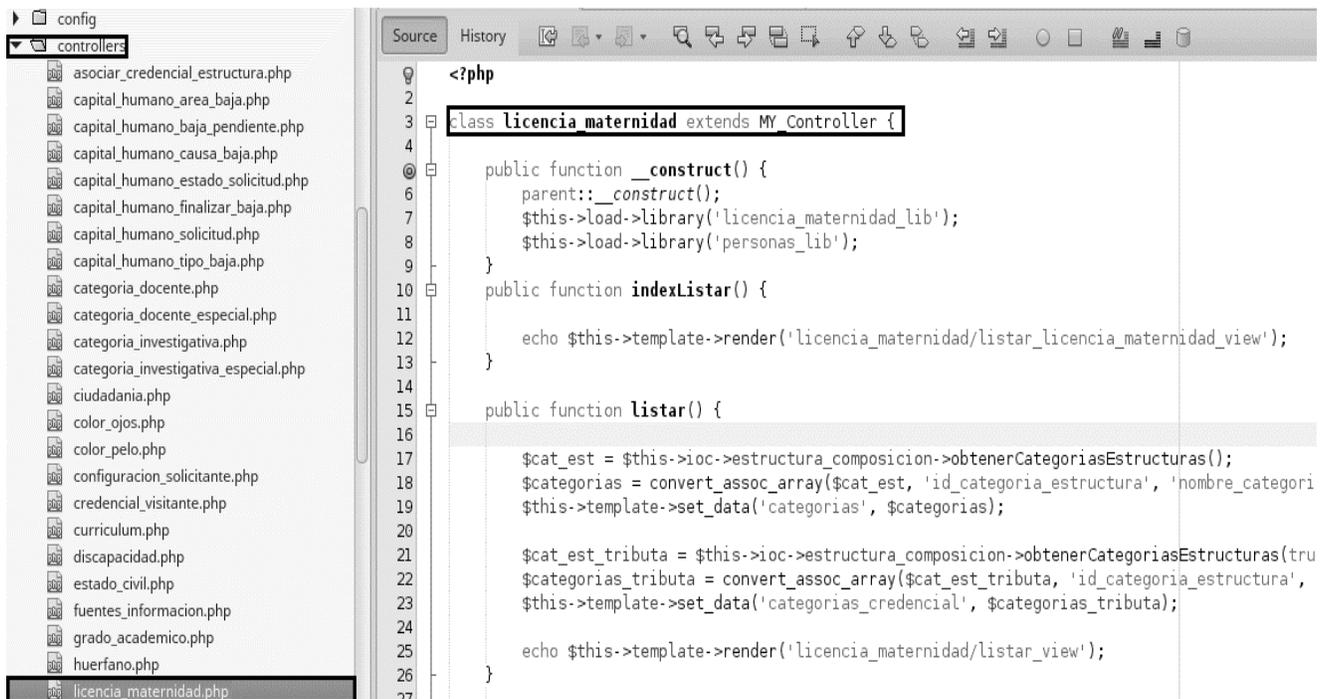


Figura 8. Patrón Controlador

Alta cohesión: la cohesión es una medida de cuan relacionadas y enfocadas están las responsabilidades de una clase. Una clase con baja cohesión hace muchas tareas no afines o un trabajo excesivo. La librería `licencia_maternidad_lib.php` es ella y solo ella la única encargada de la lógica de esa parte del negocio.

Bajo acoplamiento: el acoplamiento es una medida de la fuerza con que una clase está conectada con otras clases. Una clase con bajo acoplamiento no depende de muchas clases. El patrón propone que cada clase debe tener un bajo grado de dependencia con otras clases en la medida de lo posible. Un ejemplo es la librería `licencia_maternidad_lib.php` en su implementación solo contiene la lógica de su negocio y el acceso a datos, esta clase no tiene instancia de más ninguna clase que refleje dependencias.

Patrones GoF

Los patrones *Banda de los Cuatro (Gang-of-Four)* describen las formas comunes en que diferentes tipos de objetos, pueden ser organizados para trabajar unos con otros. Tratan la relación entre clases la formación de estructuras de mayor complejidad. Además, permiten crear grupos de objetos para ayudarnos a realizar tareas complejas. Existen tres tipos de patrones: de creación, estructurales y de comportamiento. Los patrones de creación abstraen la forma en la que se crean los objetos, permitiendo tratar las clases a crear de forma genérica dejando para más tarde la decisión de qué clases crear o cómo crearlas [44]. Los patrones GoF que se utilizaron en el desarrollo del componente son:

Patrón de comportamiento: comprenden la asignación de responsabilidades entre objetos y algoritmos. Estos no solo conciernen a los objetos y las clases sino también la comunicación entre estas, además caracterizan flujos de control complejos que son difíciles de seguir en tiempo de ejecución [45].

Mediador: define un objeto que coordine la comunicación entre objetos de distintas clases, pero que funcionan como un conjunto. Los tipos de clases librerías implementadas en GUUD funcionan como mediadoras entre las clases controladoras y las modelos o acceso a datos. A la izquierda de la Figura 9 se muestra el método `registrarLicencia` de la clase controladora `licencia_maternidad.php`, que recibe las peticiones y envía las respuestas pertinentes; luego en de la Figura 10 se muestra la librería `licencia_maternidad_lib.php`, que será una capa intermedia que implementa la lógica de registrar una licencia apoyándose de la clase modelo `tb_dlicencia_md1` para el acceso a datos.

```

public function registrarLicencia() {
    $llPost = $this->input->all_post(true);
    if ($this->licencia_maternidad_lib->registrarLicenciaM($llPost) != false) {
        $this->message('SYS001');
    } else {
        throw new Exception_Error('SYS006');
    }
}

```

Figura 9. Patrón Mediator para la controladora

```

public function registrarLicenciaM($datos) {
    $this->_ci->db->trans_start();
    $id_estado_licencia_terminada = (int) $this->_ci->xml->configuracion->sistema->personal->seguridad_social->estado_licencia_maternidad->terminada;
    $data = $this->_ci->tb_dlicencia_mdL->obtenerDadoAtributos(array('id_persona' => $datos['idPersona'],
                                                                'id_estado_licencia' => $id_estado_licencia_terminada));
    if (!empty($data)){
        $id_estado_licencia_iniciada = (int) $this->_ci->xml->configuracion->sistema->personal->seguridad_social->estado_licencia_maternidad->iniciada;
        $array_datos = array();
        $array_datos['id_persona'] = $datos['idPersona'];
        $array_datos['id_estado_licencia'] = $id_estado_licencia_iniciada;
        $this->_ci->tb_dlicencia_mdL->registrar($array_datos);
    }

    $this->_ci->db->trans_complete();
    return $this->_ci->db->trans_status();
}

```

Figura 10. Patrón Mediator para la librería

Patrones de creación: se encargan de la creación de instancias de los objetos. Abstraen la forma en que se crean los objetos, permitiendo tratar las clases a crear de forma genérica, dejando para después la decisión de que clase crear o cómo crearla [46].

Instancia única (Singleton): garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Restringe la instanciación de una clase o valor de un tipo a un solo objeto [47]. En la Figura 11 se muestra la clase librería licencia_maternidad_lib.php la cual tiene en su constructor la instancia de la modelo tb_rlicencia_periodo_licencia_mdL.php, de esta manera no hay necesidad de instanciar esta modelo en cualquier otra parte de la clase. Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella. Para asegurar que la clase no puede ser instanciada nuevamente se regula el alcance del constructor.

```
class licencia_maternidad_lib {
    public function construct() {
        $this->ci = & get instance();
        $this->_ci->load->model('tb_dlicencia_mdl');
        $this->_ci->load->model('tb_dpersona_mdl');
        $this->_ci->load->model('tb_rllicencia_periodo_licencia_mdl');
        $this->_ci->load->model('tb_ncausa_interrupcion_licencia_mdl');
    }
}
```

Figura 11. Patrón Instancia única

2.4.5. Diagramación de las vistas

El componente Personal, sistema al que se integrará la propuesta de solución, cuenta con 3 tipos de vistas principales en las que se distribuyen las funcionalidades de acuerdo con su objetivo. A continuación, se describen las áreas de contenido de las vistas del sistema.

Vista de presentación: es la primera pantalla que se visualiza al acceder al sistema. Brinda la facilidad de autenticarse. Está compuesta por 2 áreas de contenido, el área 1 que incluye el logotipo de la UCI a la izquierda y el nombre del sistema a la derecha, y el área 2 donde se encuentra el formulario de autenticación del sistema como se muestra en la Figura 12.

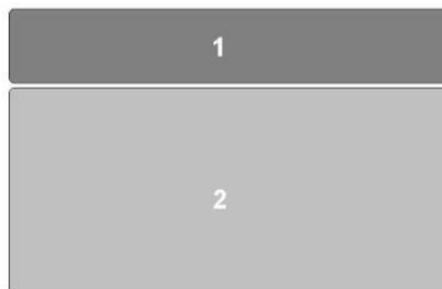


Figura 12. Áreas de la vista de presentación [48]

Vista de escritorio: es la pantalla mediante la que se accede a subsistemas o componentes. Está formada por 4 áreas. La primera constituye el encabezado de la página, incluye la iconografía para el acceso a los componentes del Núcleo del SGU, así como su ayuda general, el nombre del sistema, los datos del usuario autenticado y la opción de cierre de sesión. La segunda constituye el área de contenido que incluye los vínculos de acceso a los diferentes sistemas que están embebidos en el SGU. El área 3 se conforma por el menú lateral correspondiente para la vista de escritorio. La última de las áreas tiene el pie de página incluyendo el nombre de la Universidad y el tipo de diseño XAUCE-SGU. La Figura 13 detalla la vista de escritorio.

Descripción de la propuesta de solución

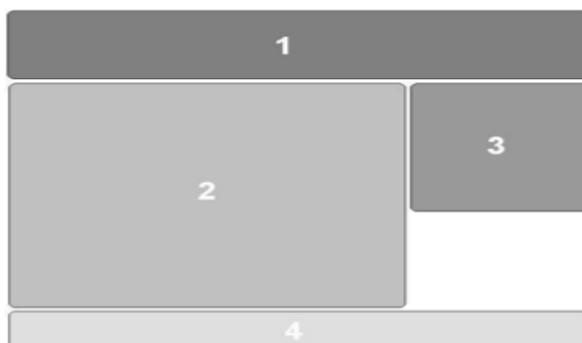


Figura 13. Áreas de la vista de escritorio [48]

Vista de gestión de procesos: es la pantalla que permite la gestión de la información en los diferentes componentes del sistema. La vista tiene 6 áreas de contenido basadas en la vista de escritorio donde el área 3 contiene la barra de iconos flotantes para una determinada funcionalidad y el área 4 es donde se realiza la gestión de dicha funcionalidad. Las demás áreas de contenido coinciden con las definidas en la vista de escritorio como se muestran en la Figura 13 a continuación la Figura 14 detalla de la vista de gestión de procesos.

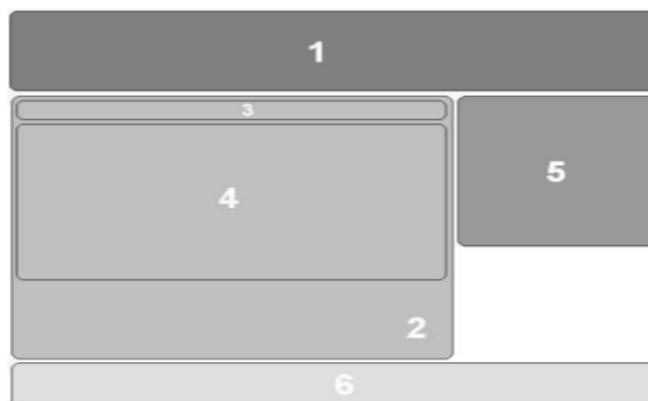


Figura 14. Áreas de la vista de gestión de procesos [48]

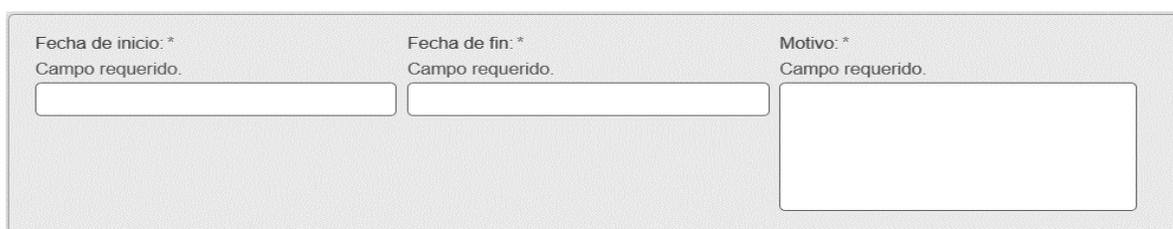
2.4.6. Sistema de mensajes de la propuesta

El componente Personal cuenta con 2 tipos de mensajes que posibilitan la interacción con el usuario en correspondencia con las acciones realizadas en el sistema. A continuación, se describen los tipos de mensajes de la propuesta de solución.

Mensajes incrustados en formularios: representan mensajes de error y generalmente se muestran sobre un componente de la vista cuando falla su validación determinada. La Figura 15 muestra un ejemplo de validación de formulario que informa al usuario el error de campos obligatorios vacíos.

Descripción de la propuesta de solución

Mensajes en ventanas emergentes: representan mensajes en ventanas emergentes para informarle al usuario un determinado suceso en el sistema. Pueden ser de información, de confirmación o de error. Los mensajes de información se muestran como resultado de una acción realizada satisfactoriamente en el sistema. Los mensajes de error ocurren como resultado de la captura de una excepción sobre una acción realizada. Los mensajes de confirmación verifican si el usuario está seguro de realizar la acción que pretende ejecutar. Las Figuras 16, 17 y 18 muestran los distintos tipos de mensajes en ventanas emergentes.

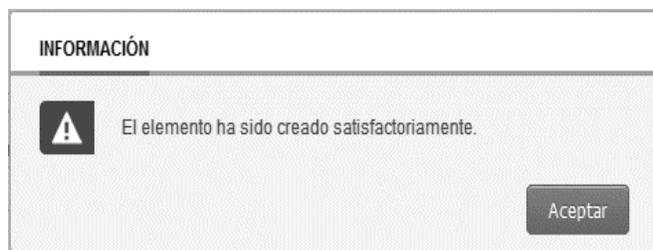


Fecha de inicio: *
Campo requerido.

Fecha de fin: *
Campo requerido.

Motivo: *
Campo requerido.

Figura 15. Mensaje incrustado en formularios [48]

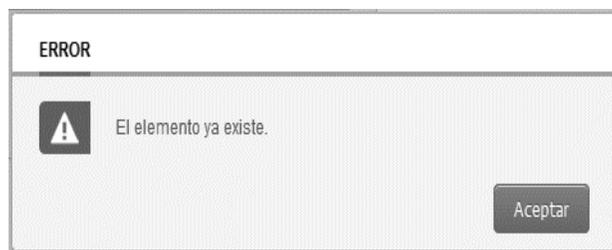


INFORMACIÓN

El elemento ha sido creado satisfactoriamente.

Aceptar

Figura 16. Mensaje de información en ventana emergente [48]

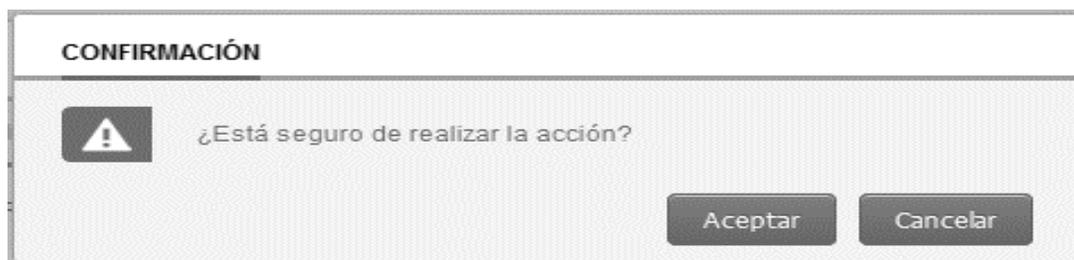


ERROR

El elemento ya existe.

Aceptar

Figura 17. Mensaje de error en ventana emergente [48]



CONFIRMACIÓN

¿Está seguro de realizar la acción?

Aceptar Cancelar

Figura 18. Mensaje de confirmación en ventana emergente [48]

2.4.7. Mapa de navegación

Un mapa de navegación es una representación resumida del sistema, orienta al usuario durante el recorrido en la aplicación y facilita un acceso directo a las funcionalidades del sistema. En la Figura 19 se muestra la distribución de la funcionalidad licencia de maternidad, licencia no retribuida para la agrupación funcional Seguridad Social, Solicitudes visitantes en la agrupación funcional Identificación y Contratación y Reportes en las agrupaciones funcionales del mismo nombre. Una representación más detallada se especifica en el **Anexo 5**.



Figura 19. Mapa de navegación principal del Componente Personal

2.4.8. Patrones de base de datos

Los patrones de diseño de base de datos son una plantilla que ya ha sido evaluada como la responsable de resolver un problema, es la guía para apoyarse en realizar el trabajo. Los patrones de diseño permiten crear una de base de datos más fortalecida por el usuario [49]. A continuación, se describen los patrones de diseño de base de datos que se utilizan en la funcionalidad licencia e maternidad.

Esquema en estrella: representa los datos como hechos que están vinculados a las dimensiones que especifican características de los datos. El esquema en estrella es el enfoque habitual para almacenes de datos, pero no se limita a este tipo de sistemas, pudiéndose utilizar para aplicaciones con muchas operaciones de lectura y poca escritura. En el modelo físico de la propuesta de solución licencia de maternidad se evidencia el patrón en la relación que existe entre las entidades licencia de maternidad, licencia interrumpida, estado licencia y la tabla relación licencia periodo, donde la relación se puede

Descripción de la propuesta de solución

considerar como una tabla de hechos y las tablas relacionadas como dimensiones. La Figura 20 muestra una representación del uso del patrón.

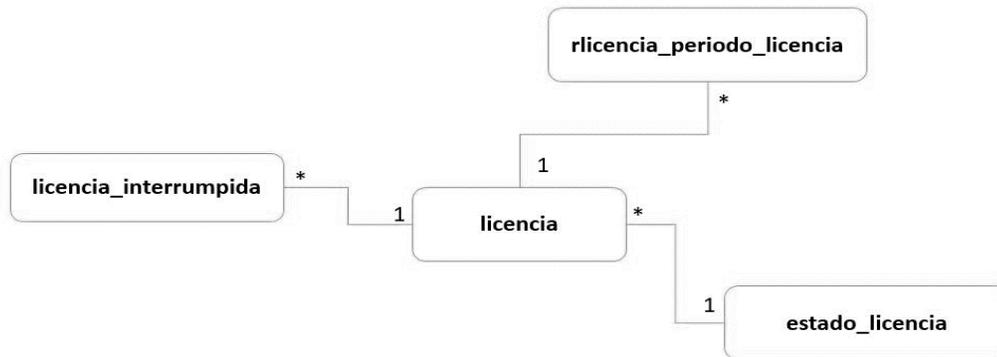


Figura 20. Patrón de diseño de base de datos esquema de estrella

Llaves subrogadas: plantea que se genere una llave primaria única para cada entidad, en vez de usar un atributo identificador en el contexto dado. El uso del patrón permite que las tablas sean más fáciles de consultar a partir del identificador, pues los tipos de datos son iguales en cada una de las tablas. El patrón se manifiesta en la definición de la llave primaria de todas las tablas del modelo físico porque se generan llaves primarias independientes a los atributos de las entidades.

2.4.9. Modelo de datos

Un modelo de datos es un conjunto de conceptos que sirven para describir la estructura de una base de datos: la información, las relaciones y las restricciones que deben cumplirse. Los modelos de datos contienen también un conjunto de operaciones básicas para la realización de consultas y actualizaciones de datos [50].

El modelo de la base de datos consta de tres fases: diseño conceptual, lógico y físico de la base de datos. La primera fase consiste en la producción de un esquema conceptual que es independiente de las consideraciones físicas. Luego el modelo se refina en un esquema lógico eliminando las construcciones que no se puede representar en el modelo de base de datos escogido. En la tercera fase, el esquema lógico se transforma en un esquema físico para el sistema gestor de base de datos elegido. La fase de diseño físico considera las estructuras de almacenamiento y los métodos de acceso necesarios para proporcionar un acceso eficiente a la base de datos. A continuación, se muestra la Figura 21 que representa el modelo físico de la propuesta de solución para las licencias, los modelos físicos restantes de los procesos anteriores se encuentran accesible en el **Anexo 4**.

Descripción de la propuesta de solución

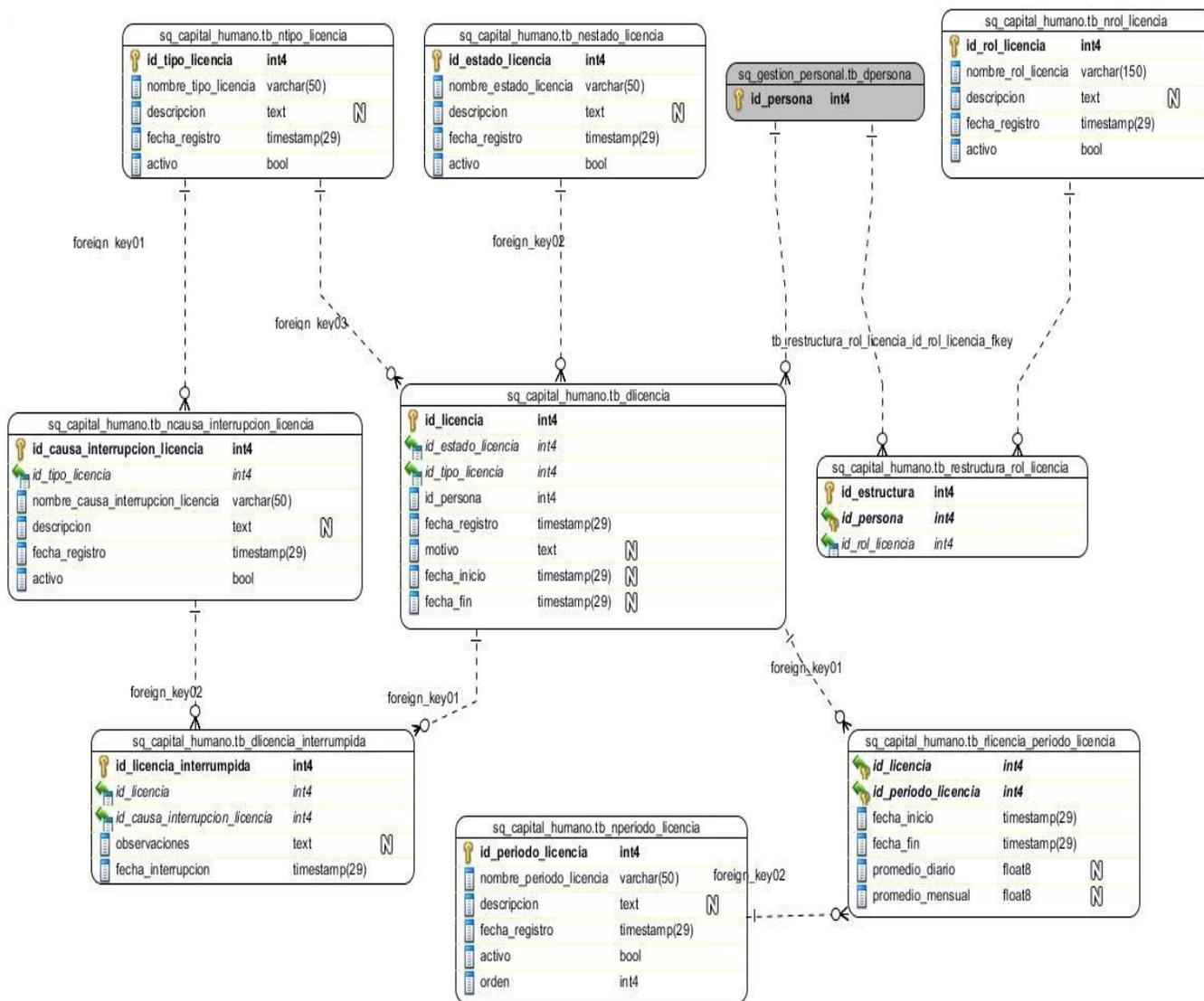


Figura 21. Modelo físico de datos de las licencias

2.5. Conclusiones parciales

Luego de describir las características que debe cumplir el componente basado en tecnologías libres y de realizar el análisis y diseño correspondiente, se describieron las técnicas para la obtención de requisitos posibilitando comprender, identificar y describir las exigencias funcionales y no funcionales que deberá cumplir la solución. La utilización de patrones de diseño y arquitectura permitió diseñar una solución robusta, flexible y escalable. A partir del análisis, el diseño realizado y de los artefactos generados quedan sentadas las bases para la implementación y validación de la solución propuesta.

3. Capítulo 3: Implementación y validación de la solución

Un producto listo para ser entregado, requiere el completo desarrollo y validación de las funcionalidades previamente definidas. En el actual capítulo se describe la construcción de la solución propuesta teniendo en cuenta las técnicas de programación, estrategias de codificación y la integración con otras aplicaciones. Se define y aplica la estrategia de pruebas del producto, especificando las evaluaciones realizadas a la propuesta y los resultados.

3.1 Estándares de codificación

Los estándares de codificación son un conjunto de reglas a seguir por los desarrolladores con el objetivo de establecer un orden y un formato común en el código fuente del software en desarrollo. Son de vital importancia durante la etapa de construcción del software ya que permite que el personal del proyecto pueda entender de forma fácil el código, garantizándose la organización y estructura del código fuente [51].

Para el desarrollo de la solución se utilizan los estándares de codificación establecidos por el grupo de arquitectura del Departamento de Desarrollo de la DIN, con el propósito de estandarizar las nomenclaturas en la implementación del sistema y obtener un producto estable eficiente.

3.1.1 Identación, llaves de apertura y cierre y tamaño de las líneas

Se usa una indentación sin tabulaciones, con un equivalente a 4 espacios, para mantener integridad en las revisiones. El uso de las llaves “{}” será en una nueva línea y la longitud de las líneas de código es aproximadamente de 70-80 caracteres para mantener la legibilidad del código. La Figura 22 se muestra un ejemplo de cómo debe quedar el formato del código de acuerdo con el estándar utilizado.

```
public function obtenerCantidadPersonaBuscarIdentif($datosBusqueda)
{
    return count($this->obtenerPersonaBuscarIdentif(null, null, null, null, $datosBusqueda));
}
```

Figura 22. Identación, llaves de apertura y cierre y tamaño de las líneas

3.1.2 Conversión de nomenclatura

Variables: se rigen por la nomenclatura *camelCase*²². Siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula. La Figura 23 muestra el uso de la notación.

²² *CamelCase* es un estilo de escritura que se aplica a frases o palabras compuestas. En este caso se usa el *lowerCamelCase*, cuando la primera letra de cada una de las palabras es mayúscula con la excepción de que la primera letra es minúscula.

```
$token = $this->session->userdata('token');
$elementoDadoToken = $this->ioc->seguridad->obtenerElementos($token);
```

Figura 23. Conversión de nomenclatura de variables

Clases: siempre comienzan con mayúscula, en caso de nombre compuesto las palabras se separan con guion bajo “_” y el resto en minúscula como se muestra a continuación en la Figura 24.

```
class Licencia_maternidad extends MY_controller {

    public function __construct() {
        parent::__construct();
        $this->load->library('Licencia_maternidad_lib');
        $this->load->library('Personas_lib');
    }
}
```

Figura 24. Conversión de nomenclatura de clases

Funciones: se rigen por la nomenclatura camelCase. Siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula. Los parámetros son separados por una coma seguida de un espacio entre 2 consecutivos. La Figura 25 ejemplifica el uso de la nomenclatura.

```
public function obtenerPersonaLicenConf($inicio = null, $limite = null, $dirOrdenar = 'asc', $datosBusqueda) {
    $criterio = null;
    if (!empty($datosBusqueda->text_buscar))
        $criterio = $datosBusqueda->text_buscar;
    else
        $criterio = "";
    if ((empty($criterio) && empty($datosBusqueda->id_estructura)))
        return array();
}
```

Figura 25. Conversión de nomenclatura de funciones

Ficheros: los nombres se escriben siempre en minúscula y en caso de nombres compuestos se usa el guion bajo, seguido del sufijo definido para cada tipo de fichero.

Vista: intuitivo y relacionado con el formulario y/o vista que representa con el sufijo “_view”.

Modelo: nombre de la tabla de la base de datos para la que se crea con el sufijo “_mdl”.

Librería: nombre de la clase que representa con el sufijo “_lib”.

Controladora: nombre de la clase que representa.

3.1.3 Estructura de control

Entre las estructuras de control y los paréntesis debe de existir un espacio. Se utiliza siempre llaves de apertura y cierre, incluso en situaciones en las que técnicamente son opcionales aumentando la legibilidad del código y disminuyendo la probabilidad de errores lógicos. La Figura 26 ejemplifica el uso de las estructuras de control en la solución.

```
foreach ($resultado as $element)
{
    $element->mostrar = '0';
    if ($element->id_estado_licencia == $idEstadoLicenciaIniciada)
    {
        $element->mostrar = '1';
    }
}
```

Figura 26. Conversión de nomenclatura de estructuras de control

3.1.4 Documentación

Todos los archivos deben de tener la documentación asociada al mismo. Cada clase y función se deben describir con el autor, una breve descripción, la categoría en caso de las clases y los parámetros en caso de las funciones. Las Figura 27 muestra la documentación de una clase y una función respectivamente en la solución.

```
/**
 * Class Licencia maternidad lib
 * Esta clase funciona como intermediaria entre la controladora y
 * las modelos
 * @package Personal
 * @subpackage Libraries
 * @category Libraries
 * @author Alejandro Garcia Gomez <alggomez@estudiantes.uci.cu>
 * @property licencia_maternidad $_ci
 */
class Licencia_maternidad_lib {
```

Figura 27. Documentación de una clase

```

/**
 * Esta función recibe como parámetro el identificador de una persona
 * y devuelve las áreas o el área a la que pertenece la persona
 * @param varchar $idPersona
 * @param array
 * @author Alejandro García Gómez <algomez@estudiantes.uci.cu>
 */
public function obtenerEstructurasDadasDadoIdPersonaLib($idPersona)
{
    return $this->_ci->tb_reestructura_rol_licencia_md1->obtenerDadoIdPersona($idPersona);
}

```

Figura 28. Documentación de una función

3.1.5 Buenas prácticas

Los valores booleanos y nulos siempre se escriben con mayúscula, para facilitar la legibilidad del código. Se debe usar un *salto de línea* antes de las estructuras de control y definición de las funciones.

3.2 Validación de requisitos

La validación de requisitos examina las descripciones para asegurar que todos los requisitos del sistema han sido establecidos sin ambigüedad, sin inconsistencias, sin omisiones, que los errores detectados hayan sido corregidos y que el resultado del trabajo se ajusta a los estándares establecidos para el proceso, el proyecto y el producto [52].

Criterios para validar los requisitos

Para validar los requisitos del sistema según Pressman²³ se pueden chequear mediante un cuestionario guiado por un conjunto de interrogantes con el objetivo de descubrir la mayor cantidad de errores posibles. A continuación, se muestra las preguntas utilizadas, algunas fueron agregadas a las planteadas por Pressman por ser consideradas necesarias para la validación.

Interrogantes para la validación de requisitos:

- a) ¿Está el requisito claramente definido? ¿Puede interpretarse mal?
- b) ¿Está identificado el origen del requisito (por ejemplo: persona, norma, documento)?
- c) ¿El planteamiento final del requisito ha sido contrastado con la fuente original?

²³Roger Pressman, es una autoridad internacionalmente reconocida en la mejora de procesos de software y en tecnologías de ingeniería de software. Por más de tres décadas, ha trabajado como ingeniero, gerente, profesor, autor y consultor de software en temas de ingeniería de software. Actualmente es presidente de R.S. Pressman and ssociates, Inc., una firma consultora especialista en métodos y entrenamiento en ingeniería de software.

- d) ¿El requisito está delimitado en términos cuantitativos?
- e) ¿Qué otros requisitos hacen referencia al requisito estudiado?
- f) ¿El requisito incumple alguna restricción definida?
- g) ¿El requisito es verificable? Si es así, ¿se podrá efectuar pruebas para verificar el requisito?
- h) ¿Se puede seguir el requisito en el modelo del sistema que se ha desarrollado?
- i) ¿Está el requisito asociado con los rendimientos del sistema o con su comportamiento?
- j) ¿El requisito está implícitamente definido?
- k) ¿El requisito es modificable?
- l) ¿El requisito está completo?
- m) ¿El requisito puede ser implementado?
- n) ¿El requisito puede ser probado?
- o) ¿El resultado de la evaluación de impacto es positivo?

Resultado de aplicar los criterios de validación

Luego de aplicar el conjunto de interrogantes para validar los requisitos definidos para el desarrollo del componente, se obtuvo el 100 % de aprobación por parte del cliente.

3.2.1 Técnicas de validación de requisitos

Con el objetivo de obtener una mayor calidad y demostrar que los requisitos definidos realmente describen al sistema que el cliente necesita; se utilizaron las técnicas para la validación de requisitos siguientes:

- **Revisión técnica formal de requisitos:** se realizan reuniones donde se examinan las descripciones del sistema buscando errores en el contenido o en la interpretación, ideas donde se necesitan aclaraciones, información incompleta, inconsistencias, requisitos contradictorios o requisitos imposibles o inalcanzables.
- **Construcción de prototipos:** la confección de los prototipos permite que el usuario pueda conocer cómo es la propuesta que el equipo de desarrollo implementa y puede aprobar la idea o corregir los elementos ajenos a los requisitos funcionales descritos.
- **Generación de casos de prueba:** la elaboración de casos de prueba permite comprobar el cumplimiento de los requisitos funcionales y que estos presentan la calidad requerida.

Resultado de aplicar las técnicas de validación

Al concluir el proceso de revisión de requisitos se detectaron algunas inconsistencias que fueron erradicadas de inmediato. Entre las más comunes se pueden citar:

- Se interpretaron de forma incorrecta algunas de las funcionalidades y características solicitadas por el cliente.
- Errores ortográficos o tipográficos.

3.3 Validación de la aplicación

La prueba de software es una actividad donde un sistema o uno de sus componentes se ejecutan en circunstancias previamente descritas, registrándose los resultados obtenidos. El principal objetivo del diseño de casos de prueba es obtener un conjunto de pruebas que tengan la mayor probabilidad de descubrir los defectos del software. Las pruebas se ejecutan en la aplicación y mediante técnicas experimentales se trata de descubrir que errores presenta, para determinar el nivel de calidad y comprobar el grado de cumplimiento respecto de las especificaciones iniciales del sistema [52].

En la investigación se utilizan algunas de las pruebas definidas por Pressman en su libro “Ingeniería de software: Un enfoque práctico”, donde se pretende que el producto se pruebe desde la parte más pequeña del software hasta la más grande en forma de espiral usando los niveles, tipos, técnicas y herramientas de prueba. Para comprobar que la solución implementada es válida y que cumple con los requisitos acordados con el cliente, se propone la siguiente estrategia de pruebas:

Tabla 4. Estrategia de pruebas de software

Nivel	Tipo de prueba	Método	Técnica
Unidad	Funcional	Caja blanca	Prueba de camino básico
Integración	Funcional	Caja negra	Incremental
Sistema	Rendimiento	Caja negra	Automática
	Funcional	Caja negra	Particiones equivalentes

Pruebas de caja blanca: se centran en la estructura de control del programa. Mediante los métodos de prueba de caja blanca, el ingeniero del software puede obtener casos de prueba que garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada componente, ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa, ejecuten todos los bucles en sus límites y con sus límites operacionales y ejerciten las estructuras internas de datos para asegurar su validez. [52]

Pruebas de caja negra: son diseñadas para validar los requisitos funcionales sin fijarse en el funcionamiento interno de un programa. La prueba de caja negra permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. La prueba de caja negra no es una alternativa a las técnicas de prueba de caja blanca; se trata de un enfoque complementario que intenta descubrir diferentes tipos de errores que los métodos de caja blanca como son errores de funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a bases de datos externas, errores de rendimiento y errores de inicialización y de terminación. Las técnicas de prueba de caja negra se centran en el ámbito de información de un programa, de forma que se proporcione una cobertura completa de la prueba. [52]

3.3.1 Prueba de unidad

Las pruebas de unidad se centran en la verificación de los elementos más pequeños del software que se puedan probar examinando las estructuras de datos locales. Su objetivo es asegurar que las funcionalidades mantienen su integridad durante los pasos de ejecución de los algoritmos. Para la prueba se utiliza el método de caja blanca o estructural que se basa en un minucioso examen de los detalles procedimentales del código a evaluar.

Existen varias técnicas para el diseño de las pruebas de unidad que permiten decidir qué sentencias o caminos se deben examinar con los casos de prueba. La técnica del camino básico se basa en obtener una medida de la complejidad del diseño procedimental de un programa. La medida de complejidad del procedimiento es la complejidad ciclomática y representa un límite inferior para el número de casos de pruebas que se deben realizar para asegurar que se ejecuta cada camino del programa [11]. La realización de las pruebas se emplea un mecanismo que posee GUUD para su automatización. El marco de trabajo posee una librería o clase especializada en la ejecución de pruebas estructurales. Aunque es sencilla cuenta con una función de evaluación y dos funciones de resultados que permite determinar con certeza si un código específico produce el tipo de dato y resultado esperado. Para ejecutar la prueba se utiliza la línea de código: `$this->unit->run (código, resultado esperado, 'nombre de prueba');` donde el *código* es el segmento de código que se desea probar, *resultado esperado* es lo que debe devolver la evaluación del código que puede ser un tipo de dato o un valor literal y *nombre de prueba* es el nombre que se le puede dar a la prueba. La Tabla 5 muestra el diseño de una prueba unitaria y su resultado suministrado por GUUD. Los diseños restantes están accesibles en el **Anexo 7**.

Tabla 5. Caso de prueba unitaria de la funcionalidad Registrar licencia de maternidad.

Caso de prueba unitaria de la funcionalidad Registrar licencia de maternidad
--

Probador: Alejandro García Gómez	
Tipo de dato esperado:	<u>Boolean.</u>
Código al que se aplica:	
<pre> public function registrarLicenciaMaternidad(\$datos) { \$this->_ci->db->trans_start(); \$idTipoLicencia = (int) \$this->_ci->xml->configuracion->sistema->personal->seguridad_social->tipo_licencia->maternidad; \$idEstadoLicenciaTerminada = (int) \$this->_ci->xml->configuracion->sistema->personal->seguridad_social->estado_licencia_maternidad->terminada; \$data = \$this->_ci->tb_dlicencia_md1->obtenerDadosAtributos(array('id_persona' => \$datos['idPersona'], 'id_estado_licencia' => \$idEstadoLicenciaTerminada)); \$dataPersona = \$this->_ci->tb_dlicencia_md1->obtenerDadosAtributos(array('id_persona' => \$datos['idPersona'])); if (!empty(\$dataPersona)) { if (!empty(\$data)) { \$idEstadoLicenciaIniciada = (int) \$this->_ci->xml->configuracion->sistema->personal->seguridad_social->estado_licencia_maternidad->iniciada; \$arrayDatos = array(); \$arrayDatos['id_persona'] = \$datos['idPersona']; \$arrayDatos['id_estado_licencia'] = \$idEstadoLicenciaIniciada; \$arrayDatos['id_tipo_licencia'] = \$idTipoLicencia; \$this->_ci->tb_dlicencia_md1->registrar(\$arrayDatos); } } else { \$idEstadoLicenciaIniciada = (int) \$this->_ci->xml->configuracion->sistema->personal->seguridad_social->estado_licencia_maternidad->iniciada; \$arrayDatos = array(); \$arrayDatos['id_persona'] = \$datos['idPersona']; \$arrayDatos['id_estado_licencia'] = \$idEstadoLicenciaIniciada; \$arrayDatos['id_tipo_licencia'] = \$idTipoLicencia; \$this->_ci->tb_dlicencia_md1->registrar(\$arrayDatos); } \$this->_ci->db->trans_complete(); return \$this->_ci->db->trans_status(); } </pre>	
Complejidad ciclomática:	Representación del grafo:
$V(G) = (A - N) + 2$ $V(G) = (7 - 6) + 2 = 3$ <p>Caminos independientes:</p> <ol style="list-style-type: none"> 1. 1, 2, 3, 6 2. 1, 2, 3, 4, 6 3. 1, 2, 4, 6 	<pre> graph TD 1((1)) --> 2((2)) 2((2)) --> 3((3)) 2((2)) --> 4((4)) 3((3)) --> 4((4)) 3((3)) --> 5((5)) 4((4)) --> 6((6)) 5((5)) --> 6((6)) </pre>
Caso de prueba para los caminos básicos	
Tipo de dato esperado:	Boolean

<pre>public function pruebaDeUnidad1() { \$idPersona['idPersona']='0dcd56ac01313c17c33581a189d239ca'; return \$this->_ci->unit->run(\$this->registrarLicenciaM(\$idPersona),TRUE, 'Registrar Licencia de Maternidad'); }</pre>	
Test Name	Registrar Licencia de Maternidad
Test Datatype	Boolean
Expected Datatype	Boolean
Result	Passed
File Name	/var/www/Nucleo/base/application/sistema/personal/libraries/licencia_maternidad_lit
Line Number	85
Evaluación del caso de prueba:	Satisfactoria

Se llevaron a cabo tres iteraciones para detectar no conformidades, las cuales arrojaron los siguientes resultados como se muestra en la tabla 5. Las no conformidades encontradas fueron resueltas en su totalidad.

Tabla 6. Iteración de las pruebas de caja blanca

Iteración	Cantidad de no conformidades	Asociados a
1	30	Errores de validación, entradas y salidas incorrectas.
2	19	Errores de validación y salidas incorrectas.
3	10	Errores de validación.
4	0	-

3.3.2 Pruebas de integración

Las pruebas de integración son una técnica sistemática para construir la arquitectura de un software a la vez que se aplican las pruebas para encontrar errores asociados a las interfaces. [11] Además, comprueban que los componentes integrados realmente funcionan juntos, que sean llamados correctamente y que transfieran los datos correctos en el tiempo preciso a través de sus interfaces. Existen dos tipos de integración: incremental y no incremental. La integración no incremental consiste en combinar los componentes como un todo y probar el sistema, lo que arroja un gran número de errores. La integración incremental se fundamenta en dividir la integración en pequeños incrementos (módulos) y realizar la integración por separado. Así es más fácil aislar y corregir los errores.

Se empleó el método incremental ascendente que comienza la construcción y las pruebas en los niveles más bajos de la estructura y combina el siguiente que se debe probar con el conjunto que ya ha sido

probado. En la validación de la solución se utiliza el método de caja negra para probar su integración con los componentes del Núcleo. La Tabla 7 muestra un caso de prueba de integración, los demás están accesibles en el **Anexo 8**. Las pruebas integración realizadas con los distintos componentes con los que se relaciona Personal, en todos los casos, arrojaron resultados satisfactorios.

Tabla 7. Caso de prueba de integración con el componente Estructura y Composición

Caso de prueba de integración con el componente Estructura y Composición	
Componente al que se integra	Estructura y Composición.
Condiciones de ejecución:	El componente de Estructura y Composición haya introducido los datos en la base de datos central y exista conexión con la misma.
Descripción de la prueba	Comprobar que el SGU en el componente Personal es capaz de gestionar las estructuras y responsabilidades involucradas con las nuevas funcionalidades.
Entradas/Pasos de ejecución	El componente de Estructura y Composición introduce en la base de datos central los datos y el SGU en el componente Personal consulta estos y gestiona las estructuras y responsabilidades involucradas con las nuevas funcionalidades.
Resultado esperado	Se gestionan las estructuras y responsabilidades involucradas con las nuevas funcionalidades.
Evaluación	Prueba satisfactoria.

3.3.4. Pruebas de rendimiento

Las pruebas de rendimiento son realizadas luego de que el sistema está completamente integrado, con el objetivo de observar el comportamiento del sistema bajo una cantidad de peticiones esperadas y comprobar que cumpla determinadas tareas en condiciones particulares de trabajo. Se encargan de demostrar que el sistema satisface sus requerimientos y monitorizar los comportamientos en cuanto a tiempo de respuesta de la petición, y otros componentes que se vean afectados por la prueba.

Las pruebas de rendimiento realizadas a la solución se aplicaron con la herramienta Apache JMeter para realizar mediciones exactas y evaluar el rendimiento. Las pruebas realizadas consistieron en ensayos de 50 a 500 accesos de usuarios respectivamente.

Las cantidades de usuarios seleccionadas para la prueba tienen como objetivo dar cumplimiento al requisito no funcional de eficiencia RNF17, que especifica que la solución soportará la conexión simultánea promedio de 50 y un máximo de 500 usuarios conectados concurrentemente. Se definió una lista de enlaces a los que se simuló el acceso aleatorio y se recolectaron los datos necesarios para su interpretación.

Las Figuras 29 y 30 muestran el tipo de respuesta que brindó la herramienta JMeter a realizar un ensayo con la funcionalidad Crear licencia de maternidad. En el **Anexo 9** se encuentran accesibles un resultado detallado de las pruebas de rendimiento.

Informe Agregado										
Nombre: Informe Agregado										
Comentarios										
Escribir todos los datos a Archivo										
Nombre de archivo: Licencia de Maternidad <input type="button" value="Navegar..."/> Log/Display Only: <input type="checkbox"/> Escribir en Log Sólo Errores <input type="checkbox"/> Successes <input type="button" value="Configurar"/>										
Label	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec	
/sgu/base/perso...	50	2844	2696	4208	806	12579	0,00%	3,7/sec	53,3	
/sgu/base/publi...	50	32	17	67	3	355	0,00%	4,0/sec	172,5	
/sgu/base/publi...	50	9	8	14	2	56	0,00%	4,0/sec	38,7	
/sgu/base/publi...	50	10	10	18	1	47	0,00%	4,0/sec	4,6	
/sgu/base/publi...	50	13	8	26	2	159	0,00%	4,0/sec	4,0	
/sgu/base/publi...	50	8	7	15	1	42	0,00%	4,0/sec	3,1	
/sgu/base/publi...	50	12	8	18	1	202	0,00%	4,0/sec	2,4	
/sgu/base/publi...	50	13	9	33	1	101	0,00%	4,0/sec	3,0	
/sgu/base/perso...	50	5229	4682	7379	2067	11179	0,00%	3,4/sec	59,1	
/sgu/base/perso...	50	9245	10085	12478	5290	13319	0,00%	3,1/sec	5,3	
/sgu/base/perso...	50	1735	1488	2927	608	6465	0,00%	4,6/sec	534,6	
TOTAL	550	1741	14	6275	1	13319	0,00%	25,0/sec	470,8	

Figura 29. Informe agregado resultado de la prueba de rendimiento

- **# Muestras:** indica la cantidad de usuarios haciendo peticiones de manera concurrente.
- **Media:** representa el tiempo de ejecución promedio de una petición.
- **Mediana:** significa que el 50% de las peticiones realizadas tardaron menos del valor reflejado.
- **Mínimo:** indica el mínimo de tiempo de ejecución invertido para una petición.
- **Máximo:** indica el máximo de tiempo de ejecución invertido para una petición.
- **% Error:** indica la relación entre el total de peticiones y el número de peticiones que originaron errores.
- **Rendimiento (Peticiones/seg.):** hace referencia al número de peticiones que el servidor puede procesar en un segundo.
- **Rendimiento (Kb/seg.):** hace referencia a la cantidad de datos que el servidor puede procesar en un segundo.

Resultados de las pruebas de rendimiento aplicadas al sistema

De manera general, en la figura 29 se puede observar que las peticiones son ejecutadas en tiempos inferiores a 5 segundos en la mayoría de los casos cumpliendo así con el requisito no funcional de eficiencia RNF18, que especifica que los tiempos de ejecución no excederán los 5 segundos de manera general. Para todas las muestras de usuarios la ocurrencia de errores se mantiene en 0.0%, lo que quiere decir que las peticiones realizadas se ejecutan satisfactoriamente.

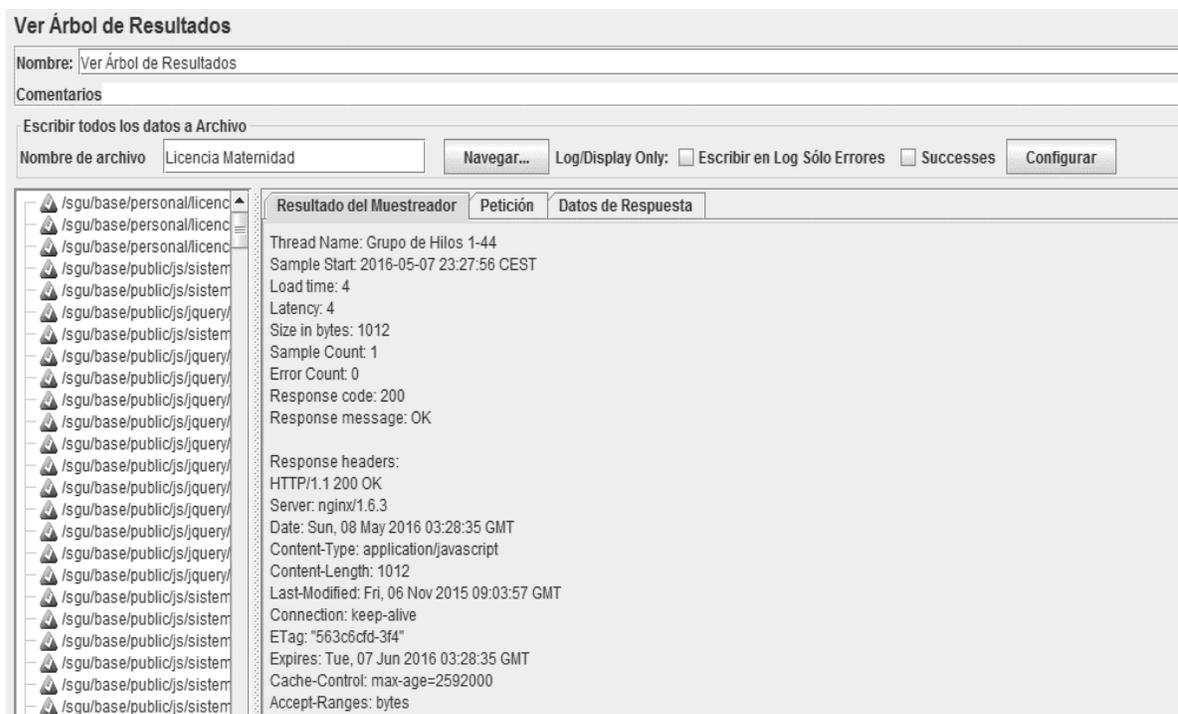


Figura 30. Árbol de resultado (ícono en verde, resultado exitoso)

3.3.5 Pruebas de sistema

Las pruebas funcionales tienen por objetivo probar que los sistemas desarrollados cumplen con las funciones específicas para los que han sido creados. Se utiliza el método de caja negra para evaluar funcionalmente la solución donde los probadores se enfocan en el funcionamiento de la interfaz del sistema a partir del estudio de sus entradas y salidas. [11]

Para confeccionar los casos de prueba de caja negra existen distintas técnicas entre las que se encuentra la técnica de particiones equivalentes. Se basa en identificar las particiones para un sistema o componente. Las particiones de equivalencia son un conjunto de datos segregados por su validez en el sistema, donde cada uno de los miembros de los conjuntos debería ser procesado equivalentemente. Por cada requerimiento funcional del sistema se generó un caso de prueba donde se recogen los datos necesarios para probarlo. La Tabla 8 describen el diseño del caso de prueba Registrar especialista, los restantes están accesibles en el **Anexo 10**.

En el diseño de caso de prueba hay que tener en cuenta que **V** indica Válido, **I** indica Inválido y **NA** que No Aplica. Las variables Tipo especialista, Persona seleccionada y Estructura, significan los valores de entrada de datos para los casos de pruebas en cada escenario; la respuesta del sistema indica su comportamiento y el flujo central, los pasos que se deben ejecutar para completar cada escenario. La condición para ejecutar la prueba es que el usuario debe estar autenticado en el sistema con el rol de administrador.

Tabla 8. Caso de prueba Listar Licencia de maternidad (Parte I)

Escenario	Descripción	Cantidad por página	Número de página	Buscar	Filtros
EC 1.1 Mostrar datos correctamente.	Mediante este escenario se muestra un listado de personas con licencia de maternidad con las opciones: Ver detalles de la licencia, Modificar licencia, Eliminar licencia e Interrumpir la licencia.	NA	NA	NA	NA
EC 1.2 Buscar	Mediante este escenario el usuario puede realizar una búsqueda de los elementos del listado. El parámetro de búsqueda es el Nombre, Apellidos, CI y Credencial de la persona que puede que se encuentre de licencia.	NA	NA	V Nombre:"Ana"	NA
		NA	NA	V "Vacio"	NA
		NA	NA	V Nombre:"Ab"	NA
EC 1.3 Filtrar	Mediante este escenario el usuario puede filtrar los elementos a mostrar en el listado. Los criterios de filtrado pueden ser: Categoría de estructura, Estructura.	NA	NA	NA	V Categoría de estructura: "Dirección"
EC 1.4 Seleccionar Cantidad por página.	Mediante este escenario el usuario escoge la cantidad de elementos a mostrar por página (los valores a escoger son 5, 10, 15 y 20).	V El usuario escoge los valores 5, 10, 15 o 20.	NA	NA	NA
EC 1.5 Escribir número de página	Mediante este escenario se escribe un número de página debajo del listado.	NA	V "2"	NA	NA
EC 1.6 Escribir un número mayor	Mediante este escenario el usuario escribe el número de la página debajo del listado.	NA	I "5452"	NA	NA

que la cantidad de páginas existentes.					
EC 1.7 Escribir en el campo número de página un carácter diferente a un valor numérico.	Mediante este escenario el usuario escribe en el campo de texto debajo del listado un carácter inválido	NA	I “seg”	NA	NA
EC 1.8 No existen elementos creados.	Mediante este escenario se muestra un listado vacío porque no se han creado elementos.	NA	NA	NA	NA

Tabla 9. Caso de prueba Registrar especialista (Parte II)

Respuesta del sistema	Flujo central
EC 1.1 El sistema muestra un listado de licencias existentes.	Al autenticarse el usuario, el sistema muestra por defecto todos los subsistemas que están contenidos dentro. Una vez autenticado el usuario en el sistema, selecciona. “Personal “en la barra de íconos de los procesos horizontales. Selecciona en la agrupación funcional, Seguridad Social de la derecha, la funcionalidad, “Licencia de maternidad”. El sistema muestra la lista de licencias de maternidad existentes hasta la fecha.
EC 1.2 El sistema muestra el listado de los elementos que coincidan con el criterio de búsqueda especificado.	EC 1.2 Al autenticarse el usuario, el sistema muestra por defecto todos los subsistemas que están contenidos dentro. Una vez autenticado el usuario en el sistema, selecciona. “Personal “en la barra de íconos de los procesos horizontales. Selecciona en la agrupación funcional, Seguridad Social de la derecha, la funcionalidad, “Licencia de maternidad”. El sistema muestra la lista de licencias de maternidad existentes hasta la fecha. El usuario introduce el criterio de búsquedas presiona el botón Buscar.
EC 1.2 El sistema muestra un listado con todos los militantes registrados en el sistema.	
EC 1.2 El sistema muestra el listado vacío porque no encontró coincidencias.	

<p>EC 1.3 El sistema muestra las personas que se encuentran de licencia de maternidad que coincida con el filtro o los filtros especificados.</p>	<p>Al autenticarse el usuario, el sistema muestra por defecto todos los subsistemas que están contenidos dentro. Una vez autenticado el usuario en el sistema, selecciona. “Personal “en la barra de íconos de los procesos horizontales. Selecciona en la agrupación funcional, Seguridad Social de la derecha, la funcionalidad, “Licencia de maternidad”. El sistema muestra la lista de licencias de maternidad existentes hasta la fecha. El usuario introduce el filtro deseado y selecciona o introduce un valor.</p>
<p>EC 1.4 El sistema muestra la cantidad de elementos por página escogida por el usuario.</p>	<p>Al autenticarse el usuario, el sistema muestra por defecto todos los subsistemas que están contenidos dentro. Una vez autenticado el usuario en el sistema, selecciona. “Personal “en la barra de íconos de los procesos horizontales. Selecciona en la agrupación funcional, Seguridad Social de la derecha, la funcionalidad, “Licencia de maternidad”. El sistema muestra la lista de licencias de maternidad existentes hasta la fecha. Se selecciona la opción de ver 5, 10, 15 o 20 elementos.</p>
<p>EC 1.5 El sistema muestra la página solicitada por el usuario.</p>	<p>Al autenticarse el usuario, el sistema muestra por defecto todos los subsistemas que están contenidos dentro. Una vez autenticado el usuario en el sistema, selecciona. “Personal “en la barra de íconos de los procesos horizontales. Selecciona en la agrupación funcional, Seguridad Social de la derecha, la funcionalidad, “Licencia de maternidad”. El sistema muestra la lista de licencias de maternidad existentes hasta la fecha. El usuario escribe el número de página deseado.</p>
<p>EC 1.6 El sistema muestra la última página del listado.</p>	<p>Al autenticarse el usuario, el sistema muestra por defecto todos los subsistemas que están contenidos dentro. Una vez autenticado el usuario en el sistema, selecciona. “Personal “en la barra de íconos de los procesos horizontales. Selecciona en la agrupación funcional, Seguridad Social de la derecha, la funcionalidad, “Licencia de maternidad”. El sistema muestra la lista de licencias de maternidad existentes hasta la fecha. El usuario escribe un número de página mayor que la cantidad de páginas existentes y el sistema muestra la última página del listado.</p>

<p>EC1.7 El sistema no deja escribir caracteres diferentes a un valor numérico.</p>	<p>Al autenticarse el usuario, el sistema muestra por defecto todos los subsistemas que están contenidos dentro. Una vez autenticado el usuario en el sistema, selecciona. “Personal “en la barra de íconos de los procesos horizontales. Selecciona en la agrupación funcional, Seguridad Social de la derecha, la funcionalidad, “Licencia de maternidad”. El sistema muestra la lista de licencias de maternidad existentes hasta la fecha.</p>
<p>EC 1.8 El sistema muestra el listado vacío.</p>	<p>Al autenticarse el usuario, el sistema muestra por defecto todos los subsistemas que están contenidos dentro. Una vez autenticado el usuario en el sistema, selecciona. “Personal “en la barra de íconos de los procesos horizontales. Selecciona en la agrupación funcional, Seguridad Social de la derecha, la funcionalidad, “Licencia de maternidad”. El sistema un listado vacío.</p>

Para obtener la calidad del funcionamiento de los requisitos funcionales se llevaron a cabo tres iteraciones, las cuales arrojaron los resultados que se muestra en la tabla 11. Las no conformidades encontradas fueron resueltas en su totalidad.

Tabla 10. Iteraciones de las pruebas de pruebas de validación

Iteración	Cantidad de no conformidades	Asociados a
1	35	Errores de interfaz, de validación y ortografía.
2	18	Errores de interfaz, de validación y ortografía.
3	0	-

3.4 Conclusiones parciales

La aplicación de los estándares de codificación, permitió construir un código limpio, legible y de fácil mantenimiento. El manejo de los paradigmas de programación establecidos en el marco de trabajo GUUD, permitió establecer un método por el cual guiarse para programar y organizar el desarrollo de la solución. Tras efectuar todas las pruebas previstas y haber alcanzado el 100% de las pruebas con resultado satisfactorio.

Conclusiones generales

- El análisis realizado a diferentes sistemas de planificación de recursos empresariales ERP, principalmente los vinculados a la gestión de los recursos humanos, demostró que no se ajustan a las necesidades y especificaciones de la UCI, aunque poseen características que sirvieron como base para elaborar la solución.
- La aplicación de la metodología de desarrollo de software DAC, facilitó la obtención y especificación de los requisitos, así como elaboración de los artefactos propuestos en el diseño de la solución.
- Las implementaciones de los requisitos permitieron construir una aplicación que lleve el control y la gestión del flujo de información que generan los procesos contratación, licencias y solicitudes de visitantes.
- Una vez realizadas las pruebas de software se demostró que la aplicación cumple con los requisitos y satisface las necesidades del cliente.

Recomendaciones

Una vez concluida la investigación y a medida que se ha desarrollado el componente Personal se recomienda:

- Incluir en el manual de usuario del componente Personal las funcionalidades asociadas a la propuesta de solución.

Referencias bibliográficas

1. RAE. *Diccionario de la lengua española*. [En línea]. Persona, 2014. [Consultado: 1 de 12 de 2015]. Disponible en: <http://dle.rae.es/?id=SjUIL8Z>.
2. RAE. *Diccionario de la lengua española*. [En línea]. Capital Humano, 2014. [Consultado: 1 de 12 de 2015]. Disponible en: <http://dle.rae.es/?id=7K8odkF>.
3. ALLES, M. A. *Diccionario de términos de Recursos Humanos*. Ediciones Granica SA, 2011. 87p.
4. RAE. *Diccionario de la lengua española*. [En línea]. Credencial, 2014. [Consultado: 1 de 12 de 2015]. Disponible en: <http://dle.rae.es/?id=BDVMojB>.
5. RAE. *Diccionario de la lengua española*. [En línea]. Licencia, 2014. [Consultado: 1 de 12 de 2015]. Disponible en: <http://dle.rae.es/?id=NG6NB42>.
6. KROENKE, D. M. *Procesamiento de bases de datos: fundamentos, diseño e implementación*. Pearson Educación, 2003. 165p.
7. DSR. *Software Gestión de Nóminas y RRHH*. [En línea]. 2014. [Consultado: 2 de 12 de 2015]. Disponible en: <http://www.gestiondelpersonal.com/software-gestion-nominas.html>.
8. ALTAMIRA. *Recursos Humanos Simplificados*. [En línea]. 2015. [Consultado: 2 de 12 de 2015]. Disponible en: <http://www.altamirahrm.com/es/software/software-rrhh>.
9. ASSETS. *Sistema de Gestión Integral*. [En línea]. 2010. [Consultado: 2 de 12 de 2015]. Disponible en: <http://www.assets.co.cu/assets.asp>.
10. ERP. *Your ERP software*. [En línea]. 2015. [Consultado: 3 de 12 de 2015]. Disponible en: <http://www.yourerpsoftware.com/>.
11. PRESSMAN, R. S. *Ingeniería de Software, un enfoque práctico, Séptima Edición*. New York, McGraw Hill Companies, 2007. 264p.
12. MÉNDEZ, A. S. *Proceso de desarrollo de software DAC*. Universidad de las Ciencias Informáticas, La Lisa, 2013.
13. SANDERS, D. *Informática Presente y Futuro, Tercera Edición*. Mexico, McGrawHill, 2010.
14. W3C. *World Wide Web Consortium*. [En línea]. 2015. [Consultado: 3 de 12 de 2015]. Disponible en: <http://www.w3.org/html>.
15. PHP. *Hypertext Preprocessor*. [En línea]. 2015. [Consultado: 3 de 12 de 2015]. Disponible en: <http://us3.php.net/manual/es/preface.php>.
16. LIBROSWEB. *Introducción a JavaScript*. [En línea]. 2015. [Consultado: 3 de 12 de 2015]. Disponible en: http://librosweb.es/libro/javascript/capitulo_1.html.

17. W3C. *World Wide Web Consortium*. [En línea]. 2015. [Consultado: 4 de 12 de 2015]. Disponible en: <http://www.w3.org/TR/REC-xml/>.
18. W3C. *World Wide Web Consortium*. [En línea]. 2015. [Consultado: 4 de 12 de 2015]. Disponible en: <http://www.w3.org/TR/CSS/>.
19. UNAD. *Universidad Nacional Abierta y a Distancia*. [En línea]. 2013. [Consultado: 4 de 12 de 2015]. Disponible en: <http://datateca.unad.edu.co/contenidos>.
20. OMG. *Object Management Group Business Process Model and Notation*. [En línea]. 2015. [Consultado: 5 de 12 de 2015]. Disponible en: <http://www.bpmn.org/>.
21. MASTERMAGAZINE. *Mastermagazine Official Blog*. [En línea]. 2012. [Consultado: 5 de 12 de 2015]. Disponible en: <http://www.mastermagazine.info/termino/5234.php#ixzz2qa0JtGOq>.
22. NETBEANS. *NetBeans IDE Features*. [En línea]. 2010. [Consultado: 5 de 12 de 2015]. Disponible en: <https://netbeans.org/features/php/>.
23. PENCIL. *Pencil Project*. [En línea]. 2010. [Consultado: 5 de 12 de 2015]. Disponible en: <http://pencil.evolus.vn/wiki/devguide/Introduction.html>.
24. VISUAL P. *Visual Paradigm*. [En línea]. 2010. [Consultado: 6 de 12 de 2015]. Disponible en: <http://www.visual-paradigm.com/aboutus/newsreleases/vp122.jsp>.
25. POSTGRESSQL. *PostgreSQL*. [En línea]. 2010. [Consultado: 6 de 12 de 2015]. Disponible en: http://www.postgresql.org.es/sobre_postgresql#intro.
26. PGADMIN. *pgAdmin: PostgreSQL administration and management tools*. [En línea]. 2014. [Consultado: 6 de 12 de 2015]. Disponible en: <http://www.pgadmin.org/>.
27. OPENSUSE. *Apache - openSUSE*. [En línea]. 2014. [Consultado: 6 de 12 de 2015]. Disponible en: <https://es.opensuse.org/Apache>.
28. APACHE. *The Apache Software Fundación*. [En línea]. 2015. [Consultado: 6 de 12 de 2015]. Disponible en: <http://jmeter.apache.org/>.
29. ALEGSA. *Diccionario de Informática*. [En línea]. 2011. [Consultado: 7 de 12 de 2015]. Disponible en: <http://www.alegsa.com.ar/Dic/framework.php>.
30. COLLAZO, Y. T. *Arquitectura GUUD*. La Lisa. Universidad de las Ciencias Informáticas: Ministerio de Educación Superior, 2013.
31. CODEIGNITER. *Desarrolloweb*. [En línea]. 2015. [Consultado: 8 de 12 de 2015]. Disponible en: <http://www.desarrolloweb.com/articulos/codeigniter.html>.
32. JQUERY. *jQuery*. [En línea]. 2015. [Consultado: 8 de 12 de 2015]. Disponible en: <http://jquery.com/>.
33. DAVENPORT, T. H. *Process innovation: reengineering work through information*. Boston, Harvard Business School Press, 2010.

34. ASAMBLEA, N. *Gaceta Oficial de la República de Cuba Ministerio de Justicia, Selección e Integración del Personal*. La Habana, 2014.
35. ASAMBLEA, N. *Gaceta Oficial de la República de Cuba Ministerio de Justicia, Licencia no Retribuida*. La Habana, 2014.
36. ASAMBLEA, N. *Gaceta Oficial de la República de Cuba Ministerio de Justicia, Seguridad Social*. La Habana, 2014.
37. SOMMERVILLE, I. *Ingeniería de Software, Octava Edición*. New York, Pearson Education, 2002.
38. BIZAGI. *Guía de usuario aplica para la Versión 10.3.0 de Bizagi BMP Suite*. [En línea]. 2013. [Consultado: 5 de 1 de 2016]. Disponible en: <http://help.bizagi.com/bpmsuite>
39. PRESSMAN, R. S. *Ingeniería del Software*. Mexico, Mcgraw-Hill, 2005.
40. KIOSKEA. *Entorno cliente/servidor*. [En línea]. 2015. [Consultado: 15 de 1 de 2016]. Disponible en: <http://es.kioskea.net/contents/148-entorno-cliente-servidor>.
41. CODEIGNITER. *Codeigniter*. [En línea]. 2015. [Consultado: 6 de 2 de 2016]. Disponible en: http://escodeigniter.com/guia_usuario/overview/mvc.html.
42. NAYILET, M. S. y YANDER S. *Diseñador de plantillas para el módulo Reportes del Sistema de Gestión Académica de Pregrado*. Universidad de las Ciencias Informáticas, La Habana, 2013.
43. BOOCH, G. *El proceso unificado de desarrollo de software*. Madrid, Pearson Educación, 2000.
44. LARMAN, C. *UML Y PATRONES, Una Introducción al análisis y diseño orientado a objetos y al proceso unificado*. Madrid, Pearson Educación, 2003.
45. GAMMA, E. *Patrones de diseño Elementos de software orientado a objetos reutilizable*. Madrid, Addison-Wesley, 2002.
46. VENETE, A. *Introducción a los Patrones de Diseño*. Universidad Jaume, Castellón de la Plana, 2011.
47. MICROSOFT. *Developer Network*. [En línea]. 2016. [Consultado: 20 de 2 de 2016]. Disponible en: <https://msdn.microsoft.com/eses/library/bb972272.aspx#EEAA>.
48. COLLAZO, Y. T. *Arquitectura de Software Vista de Presentación DAC: Sisitema de Gestión Universitaria (Núcleo)*. La Lisa. Universidad de las Ciencias Informáticas2014.
49. BLAHA, M. *Patterns of data modeling*. Segunda Edición. New York, CRC Press Taylor & Francis Group, 2010.
50. EHOW. *Tres fases del diseño tradicional de bases de datos*. [En línea]. 2015. [Consultado: 25 de 3 de 2016]. Disponible en: http://www.ehowenespanol.com/tres-fases-del-diseno-tradicional-bases-datosinfo_197446/.
51. MICROSOFT. *Revisiones de código y estándares de codificación*. [En línea]. 2014. [Consultado: 1 de 4 de 2016]. Disponible en: <http://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>.

52. PRESSMAN, R. S. *Ingeniería de software: Un enfoque práctico, Quinta edición*. Madrid, McGraw-Hill, 2002.