

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



FACULTAD 6

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

Título: Sistema de control de asistencia y gestión de pre-
nóminas de la Facultad 3.

Autor: Boris Ramón Rivero Fernández.

Tutores: Ing. Liliam Celia Beyris Soulary.

Lic. Karelys Borges Beato.

La Habana

Marzo del 2013

Declaración de Autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Boris Ramón Rivero Fernández

Autor

Lic. Karelys Borges Beato

Tutor

Ing. Liliam Celia Beyris Soulary

Tutor

Autor:

- Boris Ramón Rivero Fernández.
- Universidad de las Ciencias Informáticas.
- e-mail: borisrrf@uci.cu

Tutor:

- Karelys Borges Beato.
- Universidad de las Ciencias Informáticas.
- e-mail: sylerak@uci.cu

Licenciatura en informática 2006.

Cargo: Especialista General. Pertenece al departamento tecnológico de la facultad 3, con experiencia en el desarrollo de los procesos tecnológicos del área.

Estudios realizados: Curso de postgrado del CENSAI, Técnicas de dirección, Infotecnología, Curso preparatorio para la explotación del GARENLAB, cursado hasta el segundo bloque de la maestría en las TIC. Ha participado en varios eventos como informática y Forum a nivel de dirección.

Experiencia como tribunal de tesis, tutor y oponente en pregrado.

Tutor:

- Liliam Celia Beyris Soulyary.
- Universidad de las Ciencias Informáticas.
- e-mail: lcbeyris@uci.cu

Ingeniero en Ciencias Informáticas 2008.

Cargo: Profesor, analista de sistemas informáticos. Pertenece al centro de Ingeniería Industrial (CEDIN), con experiencia en el desarrollo de software de procesos industriales y software distribuidos (SCADA).

Estudios realizados: Curso de postgrado de Administración de Requisitos, Técnicas de Negociación y Contratación, Gestión por procesos, Gestión de los procesos de la Organización, Programación en Qt, Ingeniería de Software y Calidad, Introducción a la Ingeniería de Control. Ha participado en varios eventos como informática, Fordes y Uciencia,

Experiencia como tribunal de tesis, tutor y oponente en pregrado.



"Universidad de excelencia es una idea de hacer lo mejor que pueda hacerse, es desarrollar la mejor universidad y podemos hacerlo buscando lo óptimo, lo más perfecto posible dentro de las cosas humanas, lo más nuevo, lo más creativo, algo que no sólo sirva a los intereses de nuestro país".

A handwritten signature in black ink, which appears to be 'Che Guevara'.

Resumen

Como parte del proceso de Informatización de la sociedad cubana, se crea la Universidad de las Ciencias Informáticas (UCI), la cual cuenta con una extensa población universitaria. Debido a las deficiencias que trae consigo que el control de la asistencia y la confección de la pre-nómina salarial de los trabajadores se lleve a cabo de manera manual o por simple observación; se desarrolló un estudio donde se determinó que era de vital importancia implementar una herramienta que sirviera de apoyo a los responsables de llevar a cabo dicha tarea.

La presente investigación consiste en la implementación de un sistema capaz de controlar los elementos de la asistencia, incidencias laborales y realizar la confección de la pre-nómina salarial. El sistema permitirá un aumento de la disponibilidad de la información, la confidencialidad y la seguridad de toda la documentación generada en la ejecución de este proceso. Se realizó un estudio de las herramientas, metodologías y lenguajes existentes que permitieran la construcción de un sistema informático afín a las necesidades y condiciones tecnológicas existentes en cada una de las áreas donde se pondrá en explotación el software.

Palabras Clave

Sistema, control, asistencia, pre-nómina, evaluación.

Tabla de Contenido

INTRODUCCIÓN.....	1
CAPÍTULO I FUNDAMENTACIÓN TEÓRICA.....	5
1.1 Introducción.....	5
1.2 Conceptos Fundamentales.....	5
1.2.1 Sistemas existentes para el control de la asistencia y gestión de nóminas	6
1.3 Metodologías de Desarrollo de Software.....	10
1.3.1 Metodología ágil.....	11
1.3.2 Metodología Tradicional.....	12
1.3.3 Análisis de las metodologías.....	13
1.4 Lenguaje Unificado de Modelado.....	15
1.5 Herramientas CASE.....	16
1.5.1 Rational Rose.....	16
1.5.2 Visual Paradigm 8.0.....	16
1.6 Lenguajes de Programación, herramientas y tecnologías:.....	16
1.6.1 Servidor Web Apache 2.0.....	17
1.6.2 JavaScript.....	17
1.6.3 Pre-procesador de Hipertexto (PHP5).....	17
1.6.4 CCS.....	19
1.6.5 Symfony 2.0.....	19
1.6.6 NetBeans7.2.....	20
1.6.7 jQuery.....	20
1.6.8 Boostrap.....	20
1.6.9 Sistema Gestor de Base de Datos.....	21
1.7 Conclusiones.....	22
CAPÍTULO II CARACTERÍSTICAS DEL SISTEMA.....	23
2.1 Introducción.....	23

2.2	Descripción del sistema propuesto.....	23
2.3	Modelo de dominio	23
2.3.1	Descripción de las entidades del Modelo de Dominio	24
2.4	Ingeniería de Requisitos.....	25
2.4.1	Técnicas de obtención de requisitos	26
2.4.2	Especificación de requisitos.....	27
2.4.2.1	Requisitos funcionales	28
2.4.2.2	Requisitos No Funcionales	29
2.4.3	Patrones de Casos de Uso	30
2.4.4	Casos de Uso del sistema	31
2.4.5	Técnicas para validar requisitos.....	36
2.5	Conclusiones.....	37
CAPÍTULO III ANÁLISIS Y DISEÑO DEL SISTEMA.....		38
3.1	Introducción.....	38
3.2	Análisis	38
3.2.1	Diagramas de clases del análisis.....	38
3.3	Patrones arquitectónicos y de diseño	40
3.3.1	Arquitectura Cliente/Servidor.....	40
3.4	Patrón Arquitectónico Modelo-Vista-Controlador	40
3.5	Patrones GRASP.....	41
3.6	Diagramas de clases del diseño	42
CAPÍTULO IV CONSTRUCCIÓN Y PRUEBA DE LA SOLUCIÓN		45
4.1	Introducción.....	45
4.2	Diagrama de Componentes.....	45
4.2.1	Descripción general de los componentes.....	45
4.3	Diagrama de Despliegue.....	47
4.4	Estilos y Estándares de Codificación.	48

4.5	Pantallas de la Aplicación	51
4.6	Pruebas de Software.....	52
	Conclusiones.....	56
	Conclusiones Generales	56
	Recomendaciones.....	57
	Bibliografía.....	58
•	CU 1. Autenticar Usuario.....	61
•	CU 2. Gestionar Facultad.....	62
•	CU 3. Gestionar Área.....	65
•	CU 4. Gestionar Turno de trabajo.....	68
•	CU 5. Gestionar Usuario	¡Error! Marcador no definido.
•	CU 6. Gestionar Sanción.....	¡Error! Marcador no definido.
•	CU 7. Especificar Causa de Ausencia	¡Error! Marcador no definido.
•	CU 8. Confeccionar Pre-nómina	¡Error! Marcador no definido.

INTRODUCCIÓN

En la actualidad, con el desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) se hace imprescindible su presencia en las actividades de la vida cotidiana. Cuba además de estar inmersa en una verdadera revolución educacional se ha dado a la tarea de automatizar al máximo la sociedad cubana actual. El control es uno de los objetivos que se tiene en cuenta para la actualización del modelo económico cubano y es premisa para asegurar que todos los procesos se cumplan de la manera continua, sin atrasos ni desvíos. Esto se pone de manifiesto en todas las esferas, la gestión de los recursos humanos no está exenta a esto.

En la esfera de los recursos humanos quedan aún diversos desafíos con el fin de lograr el cumplimiento de los objetivos trazados por las organizaciones. En términos concretos, el principal desafío para el especialista de recursos humanos es lograr el mejoramiento permanente de las organizaciones de las que forman parte de manera que sean aceptables para la sociedad.

En la Universidad de las Ciencias Informáticas (UCI), en correspondencia con el actual desarrollo y puesta en marcha de proyectos en el ámbito de la informática y las telecomunicaciones, se hace necesaria la tarea de gestionar las pre-nóminas de cada trabajador para luego entregarlas al departamento de recursos humanos; así como mejorar el control sistemático de la asistencia y puntualidad de cada trabajador, y optimizar la gestión de evaluaciones de los trabajadores de la Facultad 3.

Actualmente en la Facultad 3 de la Universidad de las Ciencias Informáticas (UCI)

- No existe un sistema informático capaz de realizar las tareas anteriormente expuestas, lo que trae consigo que ocurran las siguientes agravantes:
- El control de la asistencia no es del todo fiable pues se lleva a cabo mediante el empleo de libros convencionales de firmas, lo cual permite que ocurran irregularidades por parte de los trabajadores que atentan contra la integridad del centro.
- El tiempo empleado por el responsable de realizar la pre-nómina al final del mes se torna excesivo, producto de que está obligado a revisar las hojas de firmas de cada trabajador para realizar los cálculos pertinentes.
- El proceso de remitir reportes a niveles superiores se hace un tanto tedioso. Al no estar automatizados los registros de los trabajadores, el encargado debe consultar las planillas archivadas y confeccionar manualmente el informe solicitado; dígame vacaciones, certificados médicos y licencias.

- Inexistencia de un sistema informatizado capaz de darle solución al problema, esto trae consigo que con el tiempo exista un gran cúmulo de papeles, lo que puede causar deterioro de la misma, pérdidas de información, así como retrasos y equivocaciones a la hora de evaluar a los trabajadores a final de cada mes.

La Facultad 3 es una de las direcciones que más personal posee en la universidad. Se encuentra dividida por departamentos que cuentan con un determinado número de trabajadores los cuales a su vez están clasificados por categorías: trabajadores docentes, especialistas y trabajadores no docentes. Todos poseen un sistema evaluativo y horarios laborales diferentes. Algunos de estos trabajadores laboran en las noches y por consiguiente reciben un estipendio adicional de nocturnidad. Otro elemento que se debe analizar es el tema de las evaluaciones; no se realiza de forma homogénea en todos los departamentos puesto que en algunos casos la evaluación tiene validez por un período de tres meses, y en otros departamentos se lleva a cabo de forma mensual. Por todo lo antes expuesto se requiere de un sistema capaz de adaptarse a las especificaciones propias de cada uno de los departamentos de la Facultad para así lograr que el producto tenga el alcance deseado.

Todo lo antes expuesto genera como **problema científico**: ¿Cómo contribuir a la calidad del control de la asistencia y gestión de pre-nóminas de la Facultad 3?

Surge como **objeto de estudio**: Sistema para el control de asistencia y gestión de pre-nóminas, y como **campo de acción**: Control de asistencia y gestión de pre-nóminas de los trabajadores pertenecientes a la Facultad 3.

Para darle solución al problema en cuestión se trazó como **objetivo general de la investigación**: Desarrollar un sistema informático que permita el control de la asistencia y gestión de pre-nóminas de la Facultad 3.

Variables:

Variable dependiente: Calidad del control de asistencia y gestión de pre-nóminas.

Variable independiente: Sistema informático.

Tareas de la investigación:

1. Elaboración del marco teórico a partir del estado del arte existente sobre el tema a investigar.
2. Selección de las tecnologías, metodología, lenguajes y herramientas de desarrollo a utilizar para la implementación de la solución.

3. Identificación de los requerimientos necesarios para la implementación de sistema informático que resuelva la situación problemática existente.
4. Realización del análisis y diseño del sistema informático en cuestión.
5. Implementación del sistema informático según los requisitos seleccionados.
6. Validación del sistema mediante datos de pruebas.

Durante la realización de este trabajo se utilizarán diferentes métodos científicos para estudiar las características de objeto de investigación:

Métodos Teóricos:

- **Análítico-sintético:** Para la obtención de las características, rasgos y elementos más importantes de los sistemas de gestión.
- **Análisis histórico lógico:** Para inspeccionar sobre el desarrollo de la gestión de los recursos humanos en el ámbito de los sistemas informáticos tanto en Cuba como en el resto del mundo.
- **Modelación:** Es utilizado para confeccionar los diagramas para el sistema en cuestión.

Métodos Empírico:

- **Entrevista:** Es utilizado para obtener información acerca de los procesos registrar asistencia y confeccionar la pre-nómina de pago salarial, además de obtener opiniones de los trabajadores de la DGT respecto a estos procesos.
- **Observación:** Se utiliza mediante el contacto visual para analizar las posibles soluciones a la situación polémica y elaborar la pre-nómina de pago salarial.
- **Revisión de la documentación:** Para seleccionar la información necesaria en la investigación a partir del estudio de documentos y diferentes bibliografías.
- **Método de inducción-deducción:** de importancia para la selección de una metodología apropiada para definir el desarrollo de la aplicación y asegurar su correcto funcionamiento.

Este trabajo está dividido en cuatro capítulos:

Capítulo I. Fundamentación teórica: Es donde se realiza un estudio del arte sobre el tema en cuestión y se describen las herramientas a utilizar para la implementación de la aplicación.

Capítulo II. Características del sistema: En este capítulo se realiza un estudio preliminar del sistema y el proceso de negocio en el cual se especifican los objetivos y los requerimientos.

Capítulo III. Análisis y Diseño del sistema: Se exponen detalles relacionados con el análisis, arquitectura y diseño del sistema propuesto y diagramas de clases.

Capítulo IV. Construcción y prueba de la solución: contempla la implementación del Sistema a partir de los diagramas de despliegue y de componentes. Se determinan y realizan las distintas pruebas a las funcionalidades.

CAPÍTULO I FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En el presente capítulo se analizan los sistemas más importantes para llevar a cabo el control de asistencia y el estado de las nóminas salariales en el mundo, en Cuba y en la UCI. Se realiza un análisis del estado del arte de los distintos sistemas para detectar su utilidad en las diferentes esferas en las que se emplean, además de detallar sus principales ventajas y desventajas. Se puntualizan las tendencias y tecnologías actuales, metodologías y herramientas para el desarrollo de la aplicación web que dará solución al problema científico.

1.2 Conceptos Fundamentales

Sistema de información: Es un conjunto de componentes relacionados entre sí que permite recoger, procesar, almacenar e intercambiar información. Indispensable a la hora de llevar a cabo la toma de decisiones, para la coordinación y control de una organización. Los sistemas de información también pueden ayudar a los administradores y trabajadores a analizar problemas, visualizar sujetos completos y crear productos nuevos.

Control de asistencia: permite registrar de forma sencilla y efectiva los tiempos de llegada y salida de los empleados de la empresa, al igual que vela porque los tiempos de permanencia de los empleados en la misma se ajusten a las normas establecidas. (1)

Proceso nómina salarial: Es un recibo de salario en el que la empresa acredita el pago de las diferentes cantidades que forman el sueldo. Es un proceso que agrupa a todas las operaciones que sean posibles realizar sobre una nómina. Consta de los siguientes subprocesos:

- **Incidencias y ausencias:** Permite informar a la nómina de los cambios salariales temporales, Entre ellos se tiene: turnos de trabajo, horas extras, entradas tardes, vacaciones, días festivos, remuneraciones y otras funciones más como las dietas, el kilometraje, los variables, etcétera. También informa de la ocurrencia de los días de ausencia del empleado: Permisos retribuidos, Permisos no retribuidos, etcétera. (1)
- **Pre-nóminas:** Es donde se realizan todos los cálculo y la verificación de las nóminas para luego proceder al pago de los trabajadores. A diferencia de la nómina, esta puede ser calculada tantas veces como sea necesario y consta de varios estados los cuales se almacenan en la ficha del empleado: (1)
 - **Generada:** Es un proceso de simulación por la que pasa la pre-nómina antes de convertirse en nómina.

- **Pagada:** No es más que la nómina una vez realizado el pago al trabajador. En este momento aún es posible realizar modificaciones en la misma, pero ya en esta etapa para poder llevar a cabo algún cambio, el sistema debe generar una pre-nómina con la diferencia.
- **Impresa:** Es el recibo del salario que se le entrega al trabajador una vez que ha sido adjudicado el pago.
- **Cotizada:** Es el momento en el que la pre-nómina cambia al estado de nómina salarial puesto que ha sido cotizada a la Seguridad Social. Realizar cualquier rectificación posteriormente implica la realización de una cotización complementaria, con sus correspondientes costes añadidos. (2)

1.2.1 Sistemas existentes para el control de la asistencia y gestión de nóminas

En la actualidad el desarrollo tecnológico ha posibilitado llevar la automatización a todas las esferas de la sociedad. En contradicción se puede decir que una de las áreas que más ha tardado en automatizarse a nivel mundial ha sido la del Control de Asistencia. Esto se debe a la aún amplia utilización de sistemas tradicionales de control basados en los antiguos relojes mecánicos que utilizan tarjetas de cartón, o simplemente llevaban el registro de la hora en la cual ingresan o salen de la empresa de forma manual a través de un libro de firma. En la actualidad existen mecanismos que posibilitan que dicho control se lleve a cabo de forma más precisa y controlada. Usando códigos de Barras, Tarjetas de Proximidad por radio frecuencia (RFID), Tarjetas de Banda Magnética. Sistemas Biométricos de Huellas Digitales y con la ayuda de un software capaz de controlar dicha entrada de datos se puede conseguir un mecanismo de control de asistencia mucho más eficaz y seguro.

1.2.1.1 Sistemas similares empleados en el mundo:

- **GrupoTress:** Concentra toda la información relacionada con los empleados de la empresa para agilizar la administración, consulta, modificación y organización de la misma. GrupoTress entre otras funcionalidades brinda al cliente la posibilidad de, mediante una tarjeta magnética de uso personal llevar el control de la asistencia, retardos, horas extras de cada trabajador. Además es capaz de generación de forma automática la pre-nómina. Brinda informes de horas trabajadas durante el periodo de nómina. Manejo de horas de comida y descanso, entre otros. Además nos proporciona una gran cantidad de reportes con el objetivo de optimizar el control empresarial.(3)
- **Infinite Consulting.SA:** Es una empresa especializada en la instalación de dispositivos biométricos. Cuenta con el software **Mantra** para la gestión de asistencia que le permitirá crear

los reportes característicos de un control de asistencia. Sus principales beneficios son: Permite crear reportes de asistencia dentro de los parámetros de tiempo que el cliente elija. Evitará posibles fraudes de usurpación de identidad muy comunes por el uso de tarjetas. Permite saber en tiempo real quien está o no, podrá hacer seguimientos globales o personales para saber el desempeño asistencial de su personal.(4)

- **‘Doce’ versión 2:** es un programa de control de asistencia que incorpora un registro de datos del personal de una empresa o institución. El control de asistencia se efectúa usando el teclado de una computadora en el cual el personal registra un código individual de la misma manera en que se podría hacer con un reloj tarjetero. Permite utilizar una cámara web para que tome fotografías de la persona que marca su entrada y salida de manera que, posteriormente, a través de una revisión manual sea posible identificar algún fraude cometido por personas que marcan a cuenta de otras. Los reportes que el programa puede generar son: Distintos modelos de nóminas. Historial. Reporte de atrasos y faltas. Control de marcas en el reloj. Asistencia del personal por día. Planilla mensual. Tiempo no trabajado por atrasos o salidas anticipadas. Selección manual. Cuadro mensual exportado a MS Excel.(5)

Análisis de los sistemas empleados en el mundo:

- Los software **GrupoTress** y **Mantra** brindan muchas funcionalidades con relación al control de la asistencia que pudieran ser de utilidad, pero requieren leer los datos provenientes de terminales colectoras de datos, usando tarjetas de banda magnética el primero y a través de dispositivos biométricos en el caso de **Mantra**. Deja de ser viable su utilización de ambos productos debido al elevado costo que representa en la actualidad instalar este tipo de dispositivos, la Universidad de las Ciencias Informáticas no tendría la posibilidad de adquirir esta tecnología, esto unido al hecho de que todos presentan licencia privativa hace imposible la utilización de algunos de estos sistemas.
- Por otra parte Doce versión 2 posee características similares a las de los sistemas anteriores a pesar de tener una menor cantidad de prestaciones. Tiene la peculiaridad de que a diferencia de los anteriormente analizados, este no necesita obligatoriamente contar con lectores de datos tan exigentes para la captura de los mismos. Cuenta con un mecanismo para llevar a cabo el control de asistencia mucho más simple, se puede realizar usando el teclado de una computadora en la cual el personal registra un código individual de la misma manera en que se podría hacer con un reloj tarjetero. Además este producto posee ciertas funcionalidades que se adaptan a las necesidades de la presente investigación; pero se imposibilita la adecuación a

los requerimientos de la situación problemática porque al igual que los dos anteriores todos poseen licencia privativa.

1.2.1.2 Sistemas para el control de asistencia y gestión de nóminas en Cuba:

- **Fastos (Sistema Integral para la gestión de Recursos Humanos):** El sistema de Recursos Humanos, está formado por los módulos Configuración, Personal, Capacitación, Cuadros, y Evaluación de Desempeño. Permite controlar las informaciones fundamentales de los empleados de una entidad, también realizar varios procesos y operaciones que son inherentes al área de recursos humanos, tales como:
 - Registro de los empleados: se guardan los datos de los empleados, así como informaciones referentes a los reporte de vacaciones, certificados médicos, licencias, resolución.
 - Control de la plantilla: Permite establecer la estructura organizativa de las plazas de la entidad.
 - Control de asistencia: lo cual incorpora el control de claves de asistencias, turnos de trabajos, horarios, tarjeta de asistencia e incidencias de cada empleado.
 - Permite acoplar relojes (RTA 600) para actualizar la información de la tarjeta de asistencia de forma automática.
 - Informes y modelos: Permite obtener un total de 56 informes, por ejemplo cierre del periodo, análisis de fondo de tiempo, estadísticos, entre otros.(6)
- **Rodas XXI:** Sistema Integral Económico Administrativo desarrollado por la empresa cubana CITMATEL. El sistema es capaz de guardar por trabajador los pagos y retenciones fijas que se realizan a cada uno de ellos por lo que para la confección de las nóminas cada mes sólo es necesario actualizar las incidencias que correspondan y todo el trabajo posterior de cálculo es realizado de forma automática. Este módulo permite el control, planificación y gestión de la actividad de recursos humanos aplicable en todas las entidades. Incluye Administración de personal y Cuadros. Cuenta con ocho módulos:
 - **Contabilidad:** Incluye la importación de comprobantes generados por las operaciones en el resto de los módulos permitiendo su revisión antes de ser traspasados al Mayor General, es posible incluso realizar ajustes a los comprobantes importados antes de traspasarlos. Este módulo le brinda, mediante una opción, la posibilidad de revertir comprobantes de forma automática facilitando la realización de ajustes.

- **Finanzas:** Este subsistema permite tener el registro de cheques tanto emitidos, como recibidos, así como llevar el registro de otros instrumentos de pago y efectuar las operaciones de cobros y de pagos. Además le permite tener un control de dietas, reembolso, vales para pagos menores y el registro de ingresos. Usted podrá llevar los submayores bancarios, realizar la conciliación bancaria y tener todo el control de caja con la posibilidad de realizar el arqueo correspondiente.
- **Activos fijos:** le permite tener un control detallado de los activos fijos de su entidad, realizando en el mismo momento que se registra un movimiento, su contabilización. Se pueden realizar todo tipo de operaciones de activos fijos con facilidad en el momento que se desee, generando el documento asociado al movimiento de que se trate de forma automática.
- **Nóminas:** brinda grandes facilidades a las entidades para el cálculo y emisión de sus nóminas. Se pueden calcular y emitir todos los tipos de nóminas que se utilizan en nuestro país.
- **Inventario:** le permite tener un control detallado de los inventarios de su entidad, realizando en el mismo momento que se registra un movimiento, su contabilización.
- **Facturación:** Permite Pre facturar, Facturar y Re facturar.
- **Recursos humanos:** Este módulo permite el control, planificación y gestión de la actividad de recursos humanos aplicable en todas las entidades. Incluye Administración de personal y Cuadros.
- **Tele Cobranzas:** permite organizar estratégicamente la función de cobros y efectuar un seguimiento más estrecho de las gestiones con el cliente.(7)

1.2.1.3 Análisis de los sistemas empleados en Cuba:

Ambos sistemas poseen funcionalidades que podría servirnos ya que cuentan con características similares a las que se desea obtener. Sin embargo, Fastos requiere la utilización de relojes RTA 600 lo que ocasionaría un gasto innecesario para la Facultad por lo que no es factible su utilización.

El Sistema Integral Económico Administrativo Rodas, como su nombre lo indica, posee varios módulos capaces de solucionar la mayoría de los problemas de una empresa. Por su complejidad y la cantidad de módulos que opera se cree que su utilización se ajusta mejor a otros departamentos de la empresa.

Por lo tanto, se concluye que ambos sistemas presentan especificaciones que no se adecuan con las características particulares de la dirección ni con la existencia de horarios de trabajos atípicos con que se cuenta actualmente. Además de que ambos poseen licencia privativa.

1.2.1.4 Sistemas similares realizados en la UCI

En la UCI existe un proyecto llamado Planificación de Recursos Empresariales (ERP) el cual ha desarrollado un Sistema de Gestión de Capital Humano (SGCH) que además ofrece un módulo para la confección de la nómina salarial de los trabajadores de cualquier entidad a nivel empresarial. Posee un conjunto de subsistemas tales como: Capital Humano, Contabilidad, Activos fijos tangibles e intangibles, Inventario, Facturación, Caja, Banco, Estructura y composición, Cobros y pagos, Costos y procesos, Configuración, Multimoneda, Planificación y Auditoría, los cuales están en correspondencia con las soluciones de ERP que se comercializan en el mercado a nivel mundial pero que han sido personalizadas de acuerdo a las condiciones actuales de nuestro país. Este proyecto aún continúa su desarrollo a fin de obtener un sistema estándar para las empresas cubanas. Teniendo en cuenta la magnitud de este proyecto y el tiempo que demorará su culminación se decidió implementar un sistema solo con las funcionalidades inherentes a las necesidades del área donde se va a poner en práctica.

1.3 Metodologías de Desarrollo de Software

El éxito del producto depende en gran parte de la metodología escogida, ya sea tradicional o ágil, donde los equipos maximicen su potencial, aumenten la calidad del producto con los recursos y tiempos establecidos.

Una metodología es un conjunto de estrategias, procedimientos, métodos o actividades intencionadas, organizadas, secuenciadas e integradas, que permitan el logro de aprendizajes significativos y de calidad.(8)Una metodología está compuesta por:

- **Qué tareas se llevan a cabo en cada etapa:** Actividades elementales en que se dividen los procesos.
- **Procedimiento:** Definición de la forma de ejecutar la tarea.
- **Técnica:** Herramienta utilizada para aplicar un procedimiento.
- **Herramienta:** Para realizar una técnica, podemos apoyarnos en las herramientas software que automatizan su aplicación.
- **Producto:** Resultado de cada etapa.

1.3.1 Metodología ágil

El enfoque de este tipo de tecnología nació en los años 90 y se basa en dos aspectos puntuales: el retrasar las decisiones y la planificación adaptativa; permitiendo potencia aún más el desarrollo de software a gran escala. Entre sus principales características se encuentran:

- Los individuos y las interacciones entre ellos son más importantes que las herramientas y los procesos empleados.
- Es más importante crear un producto software que funcione que escribir documentación exhaustiva.
- La colaboración con el cliente debe prevalecer sobre la negociación de contratos.
- La capacidad de respuesta ante un cambio es más importante que el seguimiento estricto de un plan.

Las metodologías ágiles se centran en desarrollar métodos de trabajo flexibles que permitan adaptarse al cambio. A “ir trabajando sobre la marcha”, Se trabaja en ciclos cortos (como mini-proyectos) que implementan una parte de las funcionalidades definiendo el camino según va avanzando el proyecto, pero siempre sin perder el rumbo. A continuación se presentan tres características de estas metodologías:(9)

1.3.1.1 SCRUM

Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Scrum es una metodología para la gestión de proyectos. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos:

- El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente.
- Reuniones a lo largo del proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.
- Scrum, presenta solo tres roles principales, en los cuales se dividen todas las responsabilidades de manejo de un proyecto:
- ProductOwner (Hombre de Negocios o Dueño del Producto): Es quien representa a los interesados (stakeholders) y es parte de la compañía que solicita el producto.
- TheTeam (El Equipo): Es un equipo auto-gestionado, auto-organizado y multi-funcional, que incluye a los desarrolladores. Tiene la responsabilidad de entregar el producto con calidad.

- Scrum Master (Maestro Scrum o Facilitador): Se encarga de mantener los procesos y tareas de manera similar a un Jefe de Proyecto. (9)

1.3.1.2 Extreme Programming (XP)

Es la más destacada de las metodologías ágiles de desarrollo de software formulada por Kent Beck.

El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

El ciclo de vida ideal de XP consta de seis fases: Exploración, Planificación de la Entrega (Release), Iteraciones, Producción, Mantenimiento y Fin del Proyecto.(9)

1.3.2 Metodología Tradicional

Al inicio el desarrollo de software era artesanal en su totalidad, la fuerte necesidad de mejorar el proceso y llevar los proyectos a la meta deseada, tuvieron que importarse la concepción y fundamentos de metodologías existentes en otras áreas y adaptarlas al desarrollo de software. Esta nueva etapa de adaptación contenía el desarrollo dividido en etapas de manera secuencial que de algo mejoraba la necesidad latente en el campo del software.

Las metodologías tradicionales (formales) se focalizan en documentación, planificación y procesos. (Plantillas, técnicas de administración, revisiones, etc.), a continuación se detallan los métodos tradicionales. (10)

1.3.2.1 Microsoft Solution Framework (MSF)

MSF es un compendio de las mejores prácticas en cuanto a administración de proyectos se refiere. Más que una metodología rígida de administración de proyectos, MSF es una serie de modelos que puede adaptarse a cualquier proyecto de tecnología de información.

Consta de cinco fases principales:

- Visión y Alcance.
- Planificación.
- Desarrollo.
- Estabilización.
- Implantación.(10)

1.3.2.2 Rational Unified Process (RUP)

RUP (del Inglés RationalUnifiedProcess), Proceso Unificado de Desarrollo, que es un proceso de desarrollo de software que junto con el Lenguaje Unificado de Modelado UML, forman la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. El Proceso Unificado es más que un simple proceso, provee un marco genérico de trabajo que puede especializarse para una gran variedad de sistemas de software.

RUP en su modelación define como sus principales elementos:

Trabajadores: Son los que realizan las actividades y son propietarios de elementos.

Actividades: Tareas que tienen un propósito claro.

Artefactos: Productos tangibles de un proyecto como: modelos, código fuente y ejecutables.

Un proyecto realizado siguiendo RUP se divide en cuatro fases:

- Inicio: Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.
- Elaboración: Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen.
- Construcción: Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario.
- Transición: El release ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores.

Las características fundamentales que se definen en el Proceso Unificado son: dirigido por casos de uso, iterativo e incremental y centrado en la arquitectura. Además de estas Está basado en componentes. Estos componentes a su vez están conectados entre sí a través de interfaces y utiliza el UML como notación básica. (8)

1.3.3 Análisis de las metodologías

Una vez detalladas las principales metodologías ágiles y tradicionales más usadas a nivel mundial se puede deducir que sus principales exponentes respectivamente son XP y RUP, por tener las

características más afines al trabajo que se pretende realizar. A continuación se realiza una comparación entre sus principales características:

Metodología	XP	RUP
Característica		
Orientación a UML	Casi nulo.	Usa UML como lenguaje básico.
Organización	Poco organizado.	Muy organizado
Roles	Orientado a desarrolladores.	Presenta los roles necesarios para cada flujo de trabajo.
Documentación	Genera poca documentación.	Genera abundante documentación.
Flujos de trabajo	Se orienta específicamente a la implementación.	Se le presta especial atención a cada uno de los flujos de trabajo, generando artefactos en cada uno de estos
Interacción con el cliente	El cliente recibe después de cada iteración una parte funcional del software	Una vez terminado un flujo de trabajo este pasa por diferentes pruebas de software y se le presenta al cliente una versión y cuando este lo acepta se puede pasar a otra fase.

Tabla 1: Comparación entre las metodologías RUP y XP.

Como metodología a utilizar, se definió optar por RUP como proceso rector para guiar el desarrollo de este trabajo porque además de las características antes expuestas en la *Tabla 1*, esta metodología hace énfasis en una buena captura de los requisitos, lo que garantiza la satisfacción de los clientes. Es apropiada en proyectos donde la documentación es de suma importancia por si necesita cambios en los procesos, particularidades que no poseen las metodologías ágiles y permite dejar bien documentada la aplicación. Otras características que dieron lugar el empleo de esta metodología es que incorpora propiedades como: el establecimiento de la arquitectura que no se vea fuertemente

impactada ante los posibles cambios posteriores que pueden ocurrir durante la construcción y el mantenimiento del producto, con su enfoque basado en modelos se permite un buen entendimiento entre clientes y los desarrolladores facilitando la obtención de un producto con altos niveles de calidad; es iterativo: la metodología parte de que se trabajará en iteraciones de corto tiempo y con metas muy claras.

1.4 Lenguaje Unificado de Modelado

La metodología RUP utiliza el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; aún cuando todavía no es un estándar oficial, está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos. La finalidad de estos diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. Un modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema.

Un modelo UML está compuesto por tres clases de bloques de construcción:

- Elementos: Los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, etc).
- Relaciones: relaciona los elementos entre sí.
- Diagramas: Son colecciones de elementos con sus relaciones.

Permite la modelación del ciclo completo de desarrollo de software y contiene:

- Diagrama de casos de uso.
- Diagrama de clases.
- Diagrama de objetos.
- Diagrama de secuencia.
- Diagrama de colaboración.
- Diagrama de estados.
- Diagrama de actividades.
- Diagrama de componentes.
- Diagrama de despliegue.(11)

1.5 Herramientas CASE

Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) se definen como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software.(12)

1.5.1 Rational Rose

Es una de las mejores herramientas CASE de modelado visual en el mundo y se encuentra a la cabeza en cuanto al desarrollo de UML, que se ha convertido en la notación estandarizada empleada en Rational Rose para especificar, visualizar y construir desarrollos de software y sistemas. Rational es actualmente conocida como una familia de software de IBM para el levantamiento de requerimientos, diseño, construcción, pruebas y administración de proyectos en el proceso desarrollo de software y sus productos están centrados en la metodología del Proceso Racional Unificado o RUP. Rational Rose es la herramienta CASE que comercializan los desarrolladores de UML y que soporta de forma completa la especificación del UML 1. Esta herramienta propone la utilización de cuatro tipos de modelo para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software. (12)

1.5.2 Visual Paradigm 8.0

Visual Paradigm para UML es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. (13)

1.6 Lenguajes de Programación, herramientas y tecnologías:

La palabra programación se define como el proceso de creación de un programa de computadora, mediante la aplicación de procedimientos lógicos, a través de los siguientes pasos:

- El desarrollo lógico del programa para resolver un problema en particular.
- Escritura de la lógica del programa empleando un lenguaje de programación específico (codificación del programa).
- El desarrollo lógico del programa para resolver un problema en particular.

- Ensamblaje o compilación del programa hasta convertirlo en lenguaje de máquina
- Prueba y depuración del programa.
- Desarrollo de la documentación.(14)

1.6.1 Servidor Web Apache 2.0

Apache es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos (HTTP 1.1). Su flexibilidad, potencia, seguridad y bajo costo lo han convertido en el número uno de manera indiscutida. Entre sus características destacan:

- Multiplataforma.
- Es un servidor de web conforme al protocolo HTTP/1.1.
- Modular: puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con las API de programación de módulos, para el desarrollo de módulos específicos.
- Basado en hebras.
- Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- Se desarrolla de forma abierta.
- Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor. (15)

1.6.2 JavaScript

JavaScript es un lenguaje de programación que permite a los desarrolladores crear acciones en sus páginas web. No requiere de compilación ya que el lenguaje funciona del lado del cliente. Tiene como principal característica ser un lenguaje independiente de la plataforma. Se puede crear todo tipo de programa que puede ser ejecutado en cualquier ordenador del mercado: Linux, Windows, Mac, etc. Debido a sus características también es muy utilizado para internet. Es un lenguaje basado en acciones que posee menos restricciones. Gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros. (16)

1.6.3 Pre-procesador de Hipertexto (PHP5)

PHP es el acrónimo de Hypertext Preprocessor (Preprocesador de Hipertexto). Se trata de un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. Está muy orientado al desarrollo de aplicaciones web y permite insertar contenidos dinámicos en las páginas.

Con PHP3 y PHP4 se había conseguido una plataforma potente y estable para la programación de páginas del lado del servidor. Haciendo posible que PHP sea el lenguaje más utilizado en la web para la realización de páginas avanzadas. Con la versión 5 se solucionaron algunos aspectos que se echaron en falta en versiones anteriores. El principal objetivo de PHP5 ha sido mejorar los mecanismos de programación orientada a objetos. Un paso necesario para lograr que PHP sea un lenguaje apto para todo tipo de aplicaciones y entornos.

- **Multiplataforma:** PHP funciona tanto en sistemas Unix o Linux con servidor web Apache como en sistemas Windows con Microsoft Internet Information Server, de forma que el código generado por cualquiera de estas plataformas no debe ser modificado al pasar a la otra. Sin embargo, las mejores prestaciones del lenguaje se obtienen trabajando en un entorno Unix o Linux con servidor web Apache.
- **Ejecución en Servidor:** Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la página PHP. Dado que la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.
- **Licencia de software libre:** PHP es un lenguaje basado en herramientas con licencia de software libre, es decir, no hay que pagar licencias, ni se limita su distribución y, es posible ampliarlo con nuevas funcionalidades si así como se desee.
- **Sintaxis cómoda:** PHP cuenta con una sintaxis similar a la de C, C++ o Perl. Lo más destacado ocurre a nivel semántico: el tipado es muy poco estricto. Es decir, cuando crea una variable no se tiene que indicar de qué tipo es, pudiendo guardar en ella datos de cualquier tipo. Esto es muy flexible y cómodo para el desarrollador, aunque los errores que se cometen pueden ser muchos más graves y difíciles de corregir al reducirse mucho las posibilidades del intérprete para detectar incompatibilidades entre variables.
- **Soporta objetos y herencia:** PHP tiene soporte para la programación orientada a objetos, es decir, es posible crear clases para la construcción de objetos, con sus constructores, etc. Además soporta herencia, aunque no múltiple.
- **Extensa librería de funciones:** PHP cuenta con una extensa librería de funciones que facilitan enormemente el trabajo de los desarrolladores.
- **Compatibilidad con bases de datos:** Quizá la característica más fuerte de PHP sea su amplio soporte para una gran cantidad de bases de datos. Tiene acceso un gran número de gestores

de bases de datos: Adabas D, dBase, Empress, Ingress, InterBase, FrontBase, DB2, Informix, mSQL, MySQL, ODBC, Oracle, PostgreSQL, Sybase, etc.

- **Expansión:** PHP está alcanzando unos niveles de uso tan elevados que hacen que su conocimiento sea algo indispensable para los profesionales del desarrollo en Internet. Se estima que PHP es usado por cientos de miles de programadores y muchos millones de sitios informan que lo tienen instalado, sumando más del 20% de los dominios en Internet.(15)

1.6.4 CCS

El modo de funcionamiento de las CSS consiste en definir, mediante una sintaxis especial, la forma de presentación que se le aplicará a:

- Un web entero, de modo que se puede definir la forma de todo el web de una sola vez.
- Un documento HTML o página, se puede definir la forma, en un pequeño trozo de código en la cabecera, a toda la página.
- Una porción del documento, aplicando estilos visibles en un trozo de la página.
- Una etiqueta en concreto, llegando incluso a poder definir varios estilos diferentes para una sola etiqueta. Esto es muy importante ya que ofrece potencia en nuestra programación. Se puede definir, por ejemplo, varios tipos de párrafos: en rojo, en azul, con márgenes, sin ellos.

Esta tecnología es bastante nueva, por lo que no todos los navegadores la soportan. En concreto, sólo los navegadores de Netscape versiones de la 4 en adelante y de Microsoft a partir de la versión 3 son capaces de comprender los estilos en sintaxis CSS. Además cabe destacar que no todos los navegadores implementan las mismas funciones de hojas de estilos, por ejemplo, Microsoft Internet Explorer 3 no soporta todo lo relativo a capas. (17)

1.6.5 Symfony 2.0

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

Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows. (18)

1.6.6 NetBeans7.2

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

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software. (19)

1.6.7 JQuery

En términos generales **jQuery** es una librería de JavaScript rápida y concisa que simplifica la ejecución de un documento HTML, manejo de eventos, animación, y las interacciones Ajax para el desarrollo web ágil. jQuery está diseñado para cambiar la forma en que escribes JavaScript. es una librería de JavaScript, creada inicialmente por Jhon Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Posee un doble licenciamiento bajo la Licencia MIT y la Licencia Publica General de MIC, permitiendo su uso en proyectos libres y privativos. **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.(20)

1.6.8 Bootstrap

Es un conjunto de herramientas de interfaz que nos permiten desarrollar rápidamente aplicaciones en una web. A través del uso de técnicas compatibles con los nuevos navegadores, pone a nuestra disposición, por ejemplo, formularios, botones, tablas, cuadrículas, menús y muchas otras herramientas para acelerar nuestro trabajo. El kit de herramientas fue creado inicialmente como un proyecto interno de Twitter, para superar los problemas obvios de **escalabilidad y facilidad de mantenimiento** del uso de diferentes bibliotecas externas. Sólo más tarde se decidió hacer público el proyecto. (21)

1.6.9 Sistema Gestor de Base de Datos

Un sistema gestor de base de datos se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos. Ayuda a realizar las siguientes acciones:

- Definición de los datos.
- Mantenimiento de la integridad de los datos dentro de la base de datos.
- Control de la seguridad y privacidad de los datos.
- Manipulación de los datos.(22)

1.6.9.1 PostgreSQL 9.0

Es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. Sus características técnicas lo hacen uno de los gestores de bases de datos más potentes y robustos del mercado. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. (23)

1.6.9.2 EMS SQL Manager Para PostgreSQL

EMS SQL Manager for PostgreSQL es una poderosa herramienta gráfica para la administración y desarrollo de PostgreSQL Database Server (servidor de bases de datos PostgreSQL). PostgreSQL Manager funciona con cualquier versión de PostgreSQL, hasta la 8.1, y soporta todas las nuevas características de PostgreSQL incluyendo espacios de tablas (tablespaces), argumentos nombrados (namedarguments) en funciones y otras más. Ofrece una gran variedad de herramientas poderosas para usuarios avanzados, tales como Visual Database Designer (diseñador visual de base de datos), Visual Query Builder (constructor visual de consultas), y un poderoso editor de objetos binarios (BLOB) para satisfacer todas sus necesidades. PostgreSQL Manager cuenta con una nueva y avanzada interfaz gráfica de usuario con un sistema asistente bastante descriptivo, tan claro en su uso que ni un principiante se podrá confundir. Sus características principales son:

- Soporte completo de PostgreSQL hasta la versión 8.1.
- Nueva interfaz gráfica de usuario último modelo.
- Ágil navegación y administración de base de datos.
- Administración sencilla de todos los objetos PostgreSQL.
- Herramientas de manipulación de datos avanzada.

- Administración efectiva de seguridad.
- Excelentes Herramientas visuales y de texto para elaboración de consultas.
- Acceso al servidor PostgreSQL a través del protocolo HTTP.
- Conexión vía reenvío de puerto local a través del túnel SSH.
- Impresionantes opciones de exportación e importación de datos.
- Poderoso diseñador visual de base de datos.
- Asistentes fáciles de usar que realizan mantenimiento de tareas de PostgreSQL.(23)

1.7 Conclusiones

En este capítulo se detallaron los conceptos fundamentales que se deben conocer para lograr un mejor entendimiento del trabajo en cuestión, lo que constituye la base del presente Trabajo de Diploma. Se realizó un estudio detallado de las principales características de los sistemas similares más usados a nivel mundial, en Cuba y en la UCI. Además se efectuó un profundo análisis para escoger las metodologías, herramientas y lenguajes idóneos para la realización del trabajo.

CAPÍTULO II CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción

En este capítulo se analizan y se describen las características principales del sistema que se desea implementar. Se determinan los requisitos funcionales y no funcionales, las entidades y actores que intervienen, se realiza el diagrama de casos de uso del sistema y el modelo de dominio. Esto se hace con el objetivo de brindar una panorámica al lector sobre los principales conceptos que se manejan en el dominio del sistema en desarrollo.

2.2 Descripción del sistema propuesto

El sistema informático que se presenta, permitirá establecer un control constante de la asistencia de los trabajadores de la Facultad, independientemente del horario de trabajo que tengan establecido. Facilitará la toma de decisiones de los directivos a la hora de llevar a cabo la evaluación de desempeño y la gestión de las sanciones administrativas. Además el sistema será capaz de ir confeccionando la pre-nóminas según se vaya registrando la asistencia del trabajador.

La propuesta será una valiosa herramienta de control que permitirán facilitar la confección de la pre-nómina salarial en la Facultad 3 para su posterior exportación hacia el departamento de recursos humanos.

2.3 Modelo de dominio

Un modelo de dominio representa un modelo conceptual de las entidades que participan en el negocio y sus relaciones. Permite definir el alcance de un sistema, quiénes y qué elementos están involucrados. El modelo de dominio constituye un artefacto del modelo de desarrollo moderno de software que representa una primera versión del sistema.

Luego de realizar un profundo análisis de los procesos que se desarrollarán, se concluye que el negocio estudiado cuenta con un bajo nivel de estructuración, la información es imprecisa e intervienen varias personas en el momento de realizar una acción. Es por ello que se considera el empleo de esta variante del modelo de negocio como la que mejor se adapta al ámbito del proyecto.

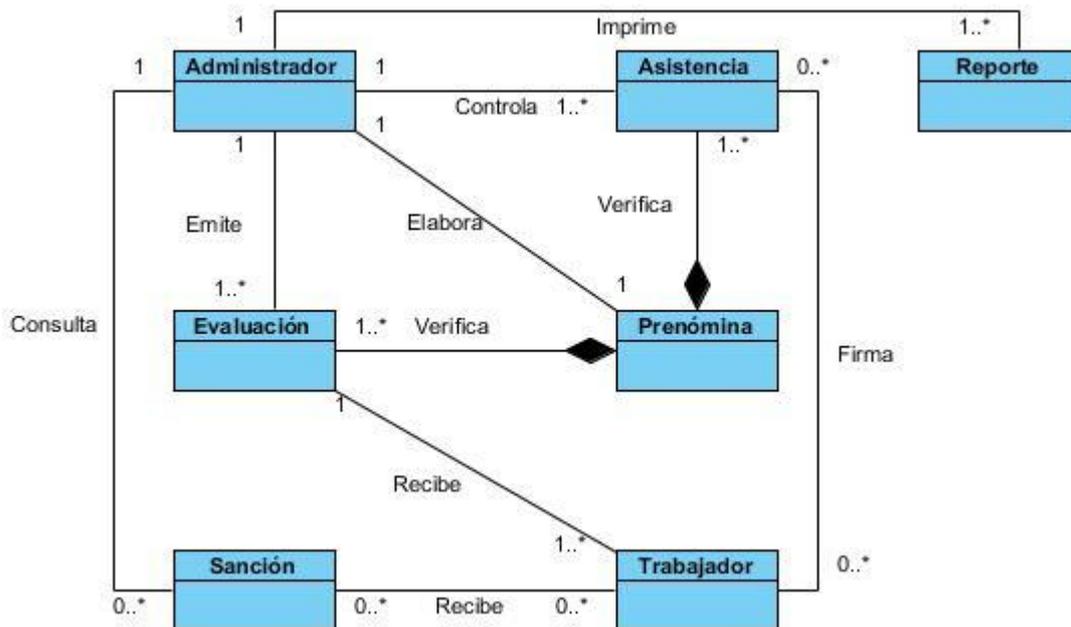


Figura 1: Modelo de dominio.

2.3.1 Descripción de las entidades del Modelo de Dominio

En la siguiente tabla se describen las entidades que conforman el modelo del dominio:

Entidad	Descripción
Administrador	Persona encargada de controlar la asistencia, emitir la evaluación y finalmente confeccionar la pre-nómina de los trabajadores.
Trabajador	Persona que firma la asistencia, recibe una evaluación y puede tener o no sanciones.
Asistencia	Entidad responsable de recolectar las firmas de entrada y salida del horario de cada trabajador.
Evaluación	Valoración del desempeño laboral del trabajador en un periodo de tiempo definido.
Pre-nómina	Entidad que contiene los datos solicitados por Capital Humano, entre estos la asistencia y evaluación.
Sanción	Amonestación que recibe el trabajador cuando incumple con sus responsabilidades.
Reporte	Información impresa, contiene datos de la pre-nómina que necesita usar

en determinado momento el responsable.
--

Tabla 2: Descripción de las entidades del Modelo de dominio.

2.4 Ingeniería de Requisitos

En el proceso de desarrollo de un sistema, sea o no para la web, el equipo de desarrollo se enfrenta al problema de la identificación de requisitos. La definición de las necesidades del sistema es un proceso complejo, pues en él hay que identificar los requisitos que el sistema debe cumplir para satisfacer las necesidades de los usuarios finales y de los clientes.

Para realizar este proceso, no existe una única técnica estandarizada y estructurada que ofrezca un marco de desarrollo que garantice la calidad del resultado. Existe en cambio un conjunto de técnicas, cuyo uso proponen las diferentes metodologías para el desarrollo de aplicaciones web.

El proceso de la ingeniería de requisitos se puede dividir en tres grandes actividades:

- 1- Obtención de requisitos.
- 2- Especificación de requisitos.
- 3- Validación de requisitos.

El proceso comienza con la realización de la captura de requisitos, el grupo de técnicos toma la información suministrada por los usuarios y clientes. Esta información puede provenir de fuentes muy diversas: documentos, aplicaciones existentes, a través de entrevistas, etc. En base a esta información, el equipo de desarrollo elabora el catálogo de requisitos. Finalmente con la validación de requisitos se realiza la valoración de los mismos, comprobando si existen inconsistencias, errores o si faltan requisitos por definir. (24)Se puede observar con mayor claridad en la siguiente diagrama.

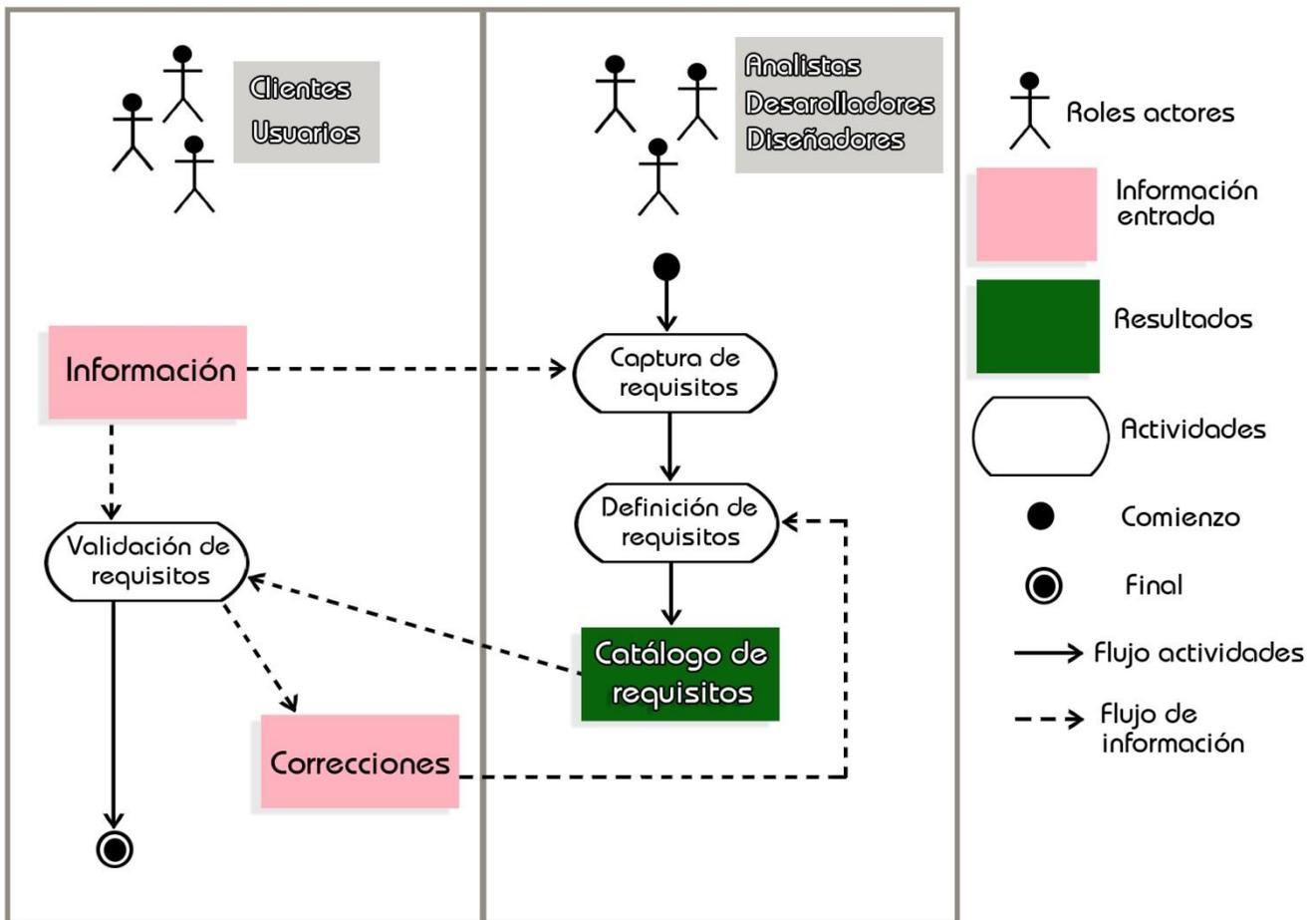


Figura 2: Proceso de Ingeniería de Requisitos.

2.4.1 Técnicas de obtención de requisitos

La captura de requisitos es la actividad mediante la que el equipo de desarrollo de un sistema de software extrae, de cualquier fuente de información disponible, las necesidades que debe cubrir dicho sistema. El proceso de captura de requisitos puede resultar complejo, principalmente si el entorno de trabajo es desconocido para el equipo de analistas, y depende mucho de las personas que participen en él. Por la complejidad que todo esto puede implicar, la ingeniería de requisitos ha trabajado desde hace años en desarrollar técnicas que permitan hacer este proceso de una forma más eficiente y precisa.

Durante la primera fase del proyecto el analista se vale de diferentes técnicas para obtener requisitos. Para la realización del presente sistema se utilizaron las siguientes técnicas:

- **Entrevistas:** Las entrevistas le permiten al analista tomar conocimiento del problema y comprender los objetivos de la solución. A través de esta técnica el equipo de trabajo se acerca al problema de una forma natural, mediante la conversación amena y tranquila sin abrumar al

proveedor de los requisitos. Existen muchos tipos de entrevistas y son muchos los autores que han trabajado en definir su estructura y dar guías para su correcta realización. Básicamente, la estructura de la entrevista abarca tres pasos:

- Identificación de los entrevistados.
 - Preparación de la entrevista.
 - Realización de la entrevista.
 - Documentación de los resultados.
- **Tormenta de ideas:** Esta técnica es bastante efectiva, es sencilla de usar y de aplicar. Es una técnica de reuniones en grupo cuyo objetivo es que los participantes muestren sus ideas de forma libre. Participan el equipo de desarrollo en su mayoría y él o los clientes aunque el total no debe exceder la cifra de 10 participantes. Todos exponen sus criterios y existe un moderador que será el que más dominio tenga del lenguaje y del negocio del problema dentro del equipo de desarrollo (este rol lo desempeña principalmente el analista) pero sin carácter controlador. Suele ofrecer una visión general de las necesidades del sistema, pero normalmente no sirve para obtener detalles concretos del mismo, por lo que suele aplicarse en los primeros encuentros.
 - **Mapas conceptuales:** Son gráficos donde se representan los conceptos involucrados y las relaciones entre ellos. Estos gráficos de relaciones se desarrollan con el usuario y sirven para aclarar los conceptos relacionados con el sistema a desarrollar. Se usan mucho en la ingeniería de requisitos, pues son fáciles de entender por el usuario, más aún si el equipo de desarrollo hace el esfuerzo de elaborarlo en el lenguaje de éste. (25)

2.4.2 Especificación de requisitos

La especificación de requisitos se divide en dos actividades: análisis de requisitos funcionales y análisis de requisitos no funcionales. La especificación de requisitos comienza haciendo un estudio de los diferentes roles de usuario que van a interactuar con el sistema. Cada actor potencialmente distinto tendrá su modelo de requisitos de navegación y de requisitos funcionales. El modelo de requisitos funcionales es representado como un modelo de casos de uso tal y como se propone en UML. En él se representa la funcionalidad principal asociada a cada rol y las interacciones que se producen entre el sistema y cada rol. El segundo modelo consiste en otro diagrama de casos de uso pero que no representa funcionalidad sino posibilidades de navegación de cada actor. La representación gráfica es realizada con una extensión de UML.

2.4.2.1 Requisitos funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. Constituyen una base para poder identificar lo que realmente debe hacer el sistema. Los mismos deben ser comprendidos tanto por los desarrolladores como por los clientes. A continuación se muestran los que la aplicación a desarrollar debe cumplir:

R1- Autenticar Usuario.

R2-Gestionar Facultad.

R2.1-Insertar Facultad.

R2.2-Modificar Facultad.

R2.3-Eliminar Facultad.

R2.4-Buscar Facultad.

R3-Gestionar Área.

R3.1-Insertar Área.

R3.2-Modificar Área.

R3.3-Elimiar Área.

R3.4-Buscar Área.

R4-Gestionar Usuario.

R4.1-Insertar Usuario.

R4.2-Modificar Usuario.

R4.3-Eliminar Usuario.

R4.4-Buscar Usuario.

R5- Asignar Permisos a Usuarios.

R6-Generar Horarios de Trabajos.

R7-Gestionar Turnos de Trabajo.

R7.1-Insertar Turnos de Trabajo.

R7.2-Modificar Turnos de Trabajo.

R7.3-Eliminar Turnos de Trabajo.

R7.4-Buscar Turnos de Trabajo.

R8-RegistrarAsistencia.

R8.1-FirmarEntrada.

R8.2-FirmarSalida.

R9-Gestionar Evaluación.

R9.1-Insertar Evaluación.

R9.2-Eliminar Evaluación.

R9.3-Modificar Evaluación.

R9.4-Buscar Evaluación.

R10-Especificar Causa de Ausencia.

R10.1-Asignar Código de Ausencia.

R11-Gestionar Sanción.

R11.1-Insertar Sanción.

R11.2-Eliminar Sanción.

R11.3-Modificar Sanción.

R11.4-Buscar Sanción.

R12-Calcular Nocturnidad.

R13-Confeccionar Pre-nómina.

R14-Visualizar Reporte Personal.

R15-Vizualizar Reporte General.

R15.1-Imprimir Reporte General.

2.4.2.2 Requisitos No Funcionales

Los requisitos no funcionales son las propiedades o cualidades que un sistema debe tener. Para obtener un óptimo funcionamiento del sistema se requiere la existencia de los siguientes requisitos en el servidor y las máquinas clientes que harán uso de la aplicación.

- **Requisitos de Software**

Cliente

- Windows2000 o superior/Cualquier distribución del sistema operativo Linux.
- Red activa.
- Cualquier navegador web.

Servidor (Requerimientos mínimos)

- Sistema Operativo Linux.
- Servidor de Base de Datos PostgreSQL 9.1.
- Servidor Web Apache 2.26.
- Lenguaje PHP 5.3.

- **Requisitos de Hardware**

Cliente

- Tarjeta de red de 10 Megabytes (Mb) o superior.
- 128 Mb de RAM.

Servidor

- 1 Gb de memoria RAM.
- 80 Gb de Disco Duro.
- Procesador Intel Dual Core a 1,86 GHz.

- **Requisitos de Seguridad**

Se deberán establecer los permisos para el acceso a las funcionalidades de administración del sistema y garantizar que las mismas se visualicen de acuerdo al nivel de usuario que está activo.

El producto deberá tener protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.

- **Requisitos de Portabilidad**

El sistema deberá ser multiplataforma y permitir que se acceda a él desde cualquier navegador.

- **Requisitos de Interfaz**

El producto deberá ser legible y acatar las normas y estándares de la universidad. La interfaz del sistema se realizará mediante una página web y su diseño deberá estar orientado a una navegación sencilla y fácil de usar.

- **Usabilidad**

La interfaz no puede tener más de 15 pestañas. Tamaño de letra mayor de 10 puntos.

- **Requisitos Legales**

Se rige por las normas y leyes establecidas por el país y la Universidad deberá estar basada en la licencia GNU/GPL.

2.4.3 Patrones de Casos de Uso

Al desarrollar o construir algo, como por ejemplo una casa o una máquina, es muy útil apoyarse en la experiencia anterior, ya sea de uno mismo o de otros. De esta manera se sabrá que la solución va a funcionar, y se tendrán identificados los problemas potenciales, así como soluciones para éstos. Estas soluciones a problemas comunes se conocen como patrones, y se utilizan en diversos aspectos del desarrollo de software.

Específicamente los patrones de casos de uso son procedimientos que deben existir en el sistema, ayudan a describir qué es lo que el sistema debe hacer, es decir, describen el uso del sistema y cómo este interactúa con los usuarios. La aplicación de patrones de casos de uso aporta los siguientes beneficios:

- Aumentar la productividad.
- Reutilizar elementos existentes (en este caso fragmentos de modelos).
- Evitar el trabajo doble por errores.
- No invertir tiempo en resolver problemas ya resueltos.
- Aplicar la teoría al trabajo práctico.
- Habilitar las herramientas de soporte para modelar el desarrollo.

Es por ello que en la investigación se plantea el uso de los mismos.

2.4.3.1 Múltiples actores

Se basa en una Generalización /Especialización, varios actores realizan una misma función. En el caso de la actual investigación se evidencia con el actor "Usuario" en los casos de uso: Registrar Firma, Visualizar reporte personal y Autenticar Usuario (Ver [Figura 3](#)).

2.4.3.2 CRUD

Su nombre es un acrónimo de las palabras en inglés: Create, Read, Update, Delete. En español: Crear, Leer, Actualizar, Eliminar. Consiste en un caso de uso para administrar la información (CRUD Información), permite modelar las diferentes operaciones para administrar una entidad de información, tales como crear, leer, cambiar y eliminar o dar de baja. Se empleará ser usado cuando todas las operaciones contribuyen al mismo valor de negocio y todas son cortas y simples.

2.4.4 Casos de Uso del sistema

Los casos de uso son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. Los casos de uso permiten mostrar el contorno (actores) y el alcance (requisitos funcionales expresados como casos de uso) de un sistema. Un caso de uso describe la secuencia de interacciones que se producen entre el sistema y los actores del mismo para realizar una determinada función. Los actores son elementos externos (personas, otros sistemas, etc.) que interactúan con el sistema como si de una caja negra se tratase. Un actor puede participar en varios casos de uso y un caso de uso puede interactuar con varios actores.

2.4.4.1 Actores del sistema

Nombre del Actor	Descripción
Usuario	Representa al actor encargado de interactuar con el sistema para beneficiarse de las funcionalidades.
Administrador	Realiza la mayoría de las acciones dentro del sistema. Es el encargado de Confeccionar las pre-nóminas, gestionar las evaluaciones, turnos de trabajo, sanciones, vacaciones y realizar reportes generales. Es el actor que más interactúa con la aplicación.

Tabla 3: Descripción de los actores del sistema.

2.4.4.2 Diagrama de Casos de Uso del sistema

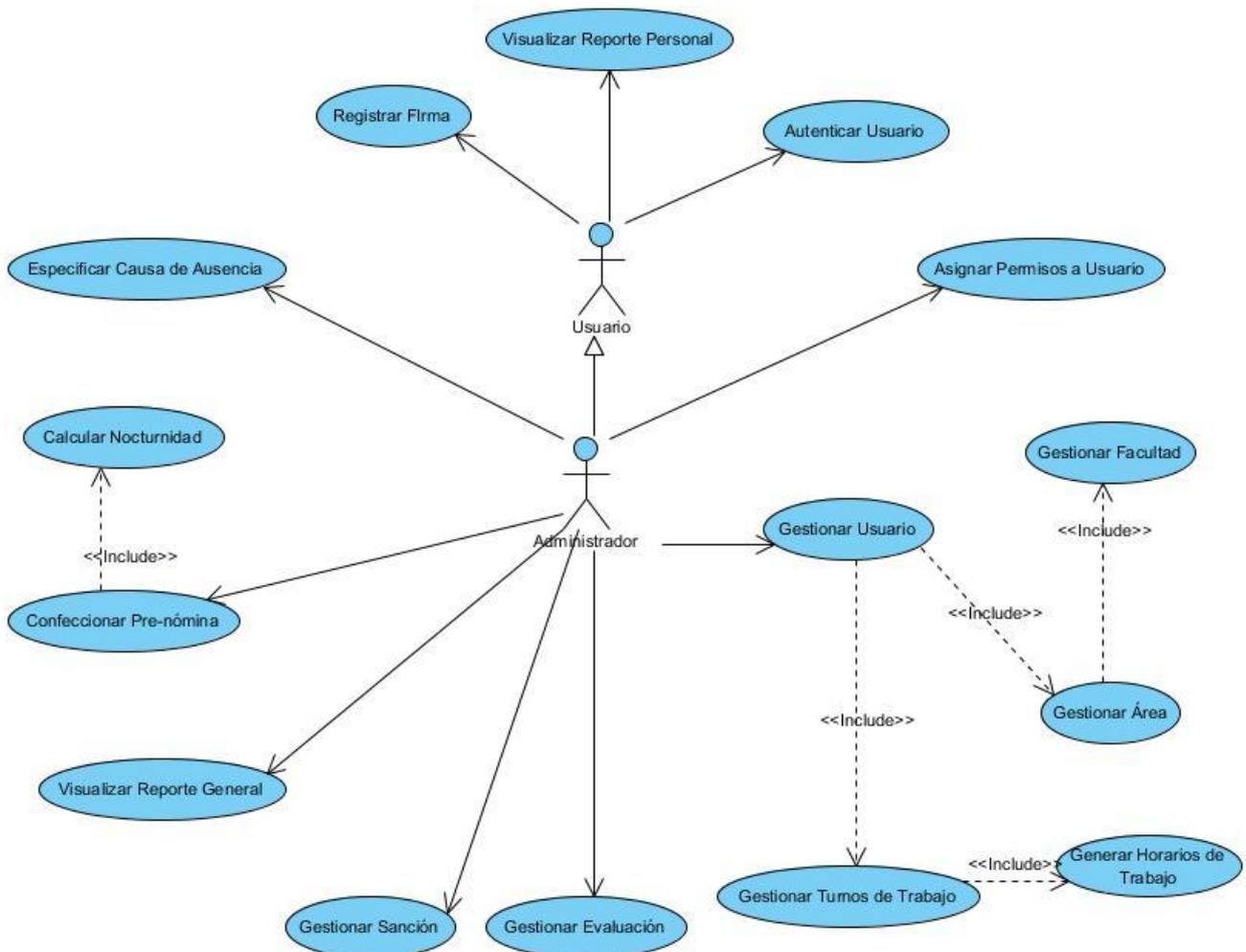


Figura 3: Diagrama de Casos de Uso del Sistema.

2.4.4.3 Descripción de Casos de Uso del sistema

Caso de Uso	Descripción	Criticidad
Autenticar Usuario	Es un mecanismo de seguridad esencial dentro del sistema ya que garantiza que solo los usuarios registrados puedan acceder a la aplicación y desarrollar la función que les toque, en dependencia del rol que se le haya asignado previamente.	1
Gestionar Facultad	Se introducen al sistema las facultades que lo utilizarán.	2
Gestionar Área	El administrador será el responsable de introducir el área o turnos de trabajos. Este caso de uso no se puede realizar sin antes haber introducido por lo menos una facultad.	2
Gestionar Usuario	Solo el administrador será el encargado de llevar a cabo esta tarea, la cual permitirá crear, eliminar y modificar los usuarios que accederán al sistema.	1
Registrar Firma	Mecanismo de seguridad digital que permite validar la autenticidad del usuario que ingresa la firma y así, registrar su asistencia diaria y puntualidad a la hora de la entrada y de la salida para comprobar el cumplimiento de su jornada laboral.	1
Generar Horarios de	Es sistema permite escoger entre 3	1

Trabajos	horarios de trabajos predefinidos, o crear un nuevo horario.	
Visualizar Reporte Personal	Funcionalidad del sistema que permite visualizar determinados reportes que pueden ser de carácter informativo para los usuarios del sistema y de vital importancia para el responsable de la confección de la pre-nómina ya que facilita en gran medida su trabajo.	2
Controlar Asistencia	El administrador puede tener control sobre las firmas que ingresa cada trabajador diariamente para modificarlas en caso de detectar alguna irregularidad o simple mente verificar los horarios en que firmo cada trabajador.	1
Especificar Causa de Ausencia	Es administrador al ver que algún trabajador no firmo debe comprobar la motivo de la ausencia para luego asignarle el códigos establecidos por Capital Humano que define el motivo de la ausencia.	2
Gestionar Turnos de Trabajo	Funcionalidad que será usada solo por el responsable de la pre-nómina. Le informa al sistema en que horarios se va a trabajar.	1
Gestionar Evaluación	El administrador que a su vez es el responsable de la confección de la pre-nómina basándose en los estándares establecidos en la universidad, asigna la evaluación a cada trabajador. Esta funcionalidad varía en dependencia de	1

	la plaza que ocupe el trabajador.	
Gestionar Sanción	Esta funcionalidad es de utilidad para el administrador de la pre-nómina. Establece un orden de prioridad a cada sanción y almacena durante un periodo de tiempo determinado por el cual es la sanción que tiene cada trabajador.	2
Asignar Vacaciones	Permite establecer el día en que saldrá de vacaciones un trabajador y la cantidad de días que disfrutara de sus vacaciones.	2
Calcular Nocturnidad	Funcionalidad que permite de forma automática abonar una retribución extra a los trabajadores que laboren en horario nocturno.	1
Confeccionar Pre-nómina	Constituye el objetivo principal del sistema. Se confecciona la pre-nómina salarial apoyándose en los datos recogidos en los casos de uso anteriores. Se realizan los cálculos pertinentes a partir del modelo establecido por el departamento de capital humano.	1
Visualizar Reporte General	Funcionalidad que permite visualizar el reporte de la pre-nómina actual o la de meses anteriores que se fueron guardando.	2

Tabla 4: Descripción de Casos de Uso del Sistema

2.4.5 Técnicas para validar requisitos

Al obtener y especificar los requerimientos, siempre se hace énfasis en que deben ser detallados por escrito y posibles de probar. La validación de requerimientos, es la etapa final de este proceso y se retroalimenta de las etapas anteriores de la ingeniería de requisitos, examinando las especificaciones para asegurar que todos los requisitos del sistema han sido establecidos con la mayor precisión. Tiene como misión demostrar que la definición de los requisitos define realmente el sistema que el usuario necesita o el cliente desea. Es necesario asegurar que el análisis realizado y los resultados obtenidos de la etapa de definición de requisitos son correctos. Su preocupación es mostrar que los requisitos realmente reflejan las necesidades de los usuarios. Buscan identificar problemas de: omisión, inconsistencia y/o ambigüedad. Se verifica que los errores detectados hayan sido corregidos y que el resultado del trabajo se ajusta a los estándares y criterios de calidad establecidos para el proceso, el proyecto y el producto.

Existen varias técnicas que posibilitan la verificación de los requerimientos entre las que se destacan:

- **Revisiones:** Como su nombre lo indica, está basada en revisar las especificaciones de los requerimientos. Es una técnica manual, cuyo principio es analizar nuevamente los requerimientos para emitir una versión final de los mismos. Su objetivo es detectar posibles conflictos (defectos y/o contradicciones) en el producto que se examina, de forma que puedan plantearse alternativas y los participantes aumenten su conocimiento sobre el producto en cuestión.
- **Construcción de prototipos:** Un prototipo es una representación limitada del diseño de un producto que permite a las partes responsables de su creación experimentar, probarlo en situaciones reales y explorar su uso. Un prototipo es un modelo fácilmente ampliable y modificable de un sistema planificado, probablemente incluyendo su interfaz y su funcionalidad de entradas y salidas.

Un prototipo puede ser cualquier cosa, desde un trozo de papel con dibujos sencillos hasta un complejo software. Es un modelo a escala o facsímil de lo real, pero no tan funcional para que equivalga a un producto final, ya que no lleva a cabo la totalidad de las funciones necesarias del sistema final proporcionando una retroalimentación temprana por parte de los usuarios acerca del Sistema.

Esta técnica se utiliza para que el cliente valore si el prototipo tiene las capacidades de desempeño que la aplicación requiere. Explora las características que puedan poner en peligro su utilización futura, evalúa su funcionalidad y estabilidad. En función de las opiniones

obtenidas de los clientes se pueden cambiar e incluso desechar algunas ideas para desarrollos futuros pero el objetivo principal de esta técnica es lograr una aceptación de los requisitos funcionales. Esta técnica tiene el problema de que el usuario debe entender que lo que está viendo es un prototipo y no el sistema final.

- **Matrices de trazabilidad:** Esta técnica consiste en marcar los objetivos del sistema y chequearlos contra los requisitos del mismo. Es necesario ir viendo que objetivos cubre cada requisito, de esta forma se podrán detectar inconsistencias u objetos no cubiertos. La trazabilidad ha venido a constituir un gran apoyo, que debe formar parte en todas las etapas del desarrollo de un sistema, desde la captura de requerimientos hasta la implementación. Esto contribuirá al seguimiento de un requerimiento hacia adelante o hacia atrás, reflejando los efectos que puede tener, la inclusión o exclusión de un nuevo requisito. La trazabilidad también ayuda en la integración de nuevos módulos reflejando.(25)

(15) Use Case	(43) Requirement	Adicionar Evaluación	Adicionar Facultad	Adicionar Sanción	Adicionar Turno de Trabajo	Adicionar Usuario	Adicionar Área	Asignar Permisos a Usuarios	Autenticar Usuario	Buscar Facultad	Buscar Evaluación	Buscar Sanción	Buscar Turno de Trabajo	Buscar Usuario	Buscar Área	Calcular Nocturnidad	Confeccionar Pre-nómina	Eliminar Evaluación	Eliminar Facultad	Eliminar Sanción	Eliminar Turno de Trabajo	Eliminar Usuario	Eliminar Área	Especificar Causa de Ausencia	Generar Horarios de Trabajo	Gestionar Evaluación	Gestionar Facultad	Gestionar Sanción	Gestionar Turnos de Trabajo	Gestionar Usuario	Gestionar Área	Imprimir Reporte General	Modificar Evaluación	Modificar Facultad	Modificar Sanción	Modificar Turno de Trabajo	Modificar Usuario	Modificar Área	Registrar Asistencia	Registrar Entrada	Registrar Salida	Verificar Código de Ausencia	Visualizar Reporte General	Visualizar Reporte Personal							
Asignar Permisos a Usuario								✓																																											
Autenticar Usuario									✓																																										
Calcular Nocturnidad																✓																																			
Confeccionar Pre-nómina																	✓																																		
Especificar Causa de Ausencia																								✓																											
Generar Horarios de Trabajo																									✓																										
Gestionar Evaluación		✓									✓								✓																																
Gestionar Facultad			✓							✓										✓							✓																								
Gestionar Sanción				✓								✓									✓							✓																							
Gestionar Turnos de Trabajo					✓								✓									✓							✓																						
Gestionar Usuario						✓								✓									✓																												
Gestionar Área							✓								✓								✓																												
Registrar Firma																																																			
Visualizar Reporte General																																																			
Visualizar Reporte Personal																																																			

Figura 4: Matriz de Trazabilidad Requisitos-Casos de Uso.

2.5 Conclusiones

En este capítulo se pudo apreciar las características fundamentales del negocio, se definieron y describieron los principales procesos del negocio. Se modelaron los artefactos correspondientes a los flujos de trabajo del modelamiento del negocio y obtención de requisitos, lo cual proporciona una visión general de lo que el sistema debe hacer. Con esto quedan sembradas las pautas para dar comienzo al proceso de análisis y diseño del “Sistema de Control de Asistencia y Gestión de la Pre-nómina de la Facultad 3”.

CAPÍTULO III ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 Introducción

En el presente capítulo se analizan los casos de uso que fueron descritos en el capítulo anterior, refinándolos y estructurándolos para lograr una mejor comprensión de los mismos. Además se propondrá el patrón arquitectónico y los de diseño para obtener los cimientos bajo los cuales se va a desarrollar la aplicación, además se generarán los artefactos obtenidos en este flujo de trabajo, tales como, diagramas de clases de análisis, diagramas de interacción y diagramas de clases del diseño.

3.2 Análisis

Durante el análisis, se analizan los requisitos que fueron descritos en la captura de requisitos, refinándolos y estructurándolos. Esta estructura (basada en clases de análisis) es independiente de la estructura que se dio a los requisitos (basada en casos de uso). Sin embargo existe una trazabilidad directa entre esas distintas estructuras, la cual se define entre casos de uso del modelo de casos de uso y realizaciones del caso de uso en el modelo de análisis. En la construcción del modelo de análisis se tienen que identificar las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellas.

Es importante señalar que aunque en el modelo del análisis hay un refinamiento de los requisitos, no se tiene en cuenta el lenguaje de programación a usar en la construcción, la plataforma en la que se ejecutará la aplicación, los componentes prefabricados o reusables de otras aplicaciones, entre otras características que afectan al sistema, ya que el objetivo del análisis es comprender perfectamente los requisitos del software y no precisar cómo se implementará la solución.

3.2.1 Diagramas de clases del análisis

A continuación se ejemplifican tres de los diagramas clases del análisis pertenecientes a tres de los casos de uso críticos del sistema.

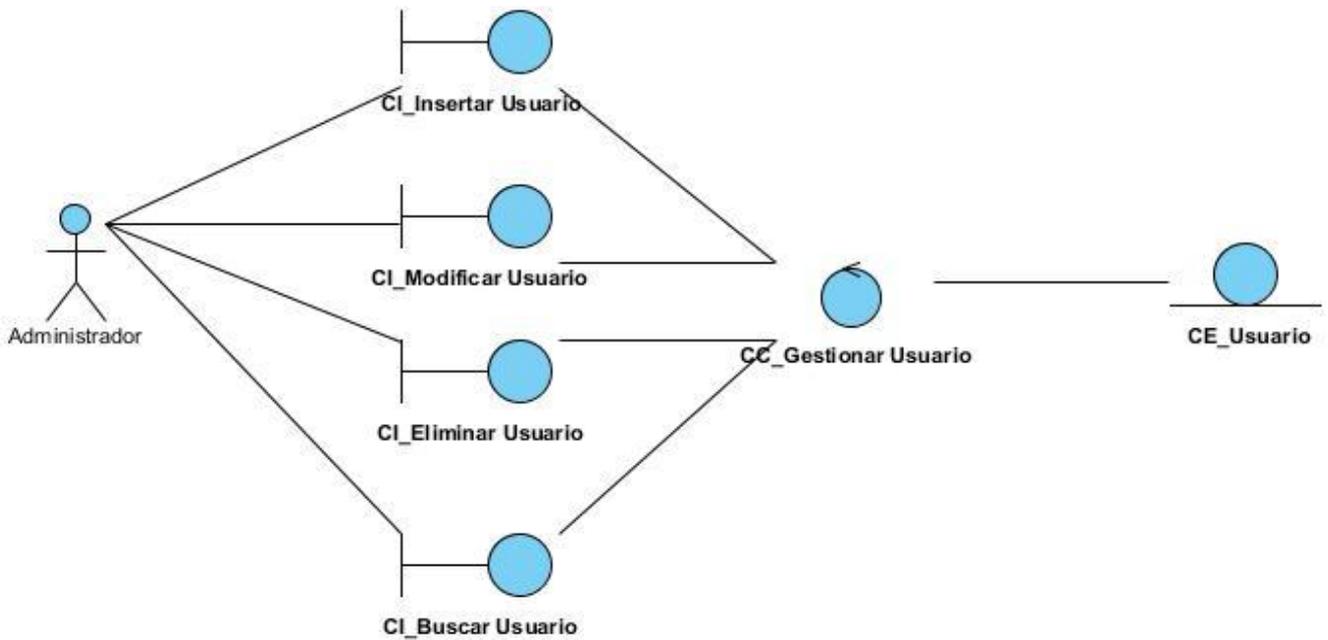


Figura 5: Diagrama de Clases del Análisis (CU-Gestionar Usuario).

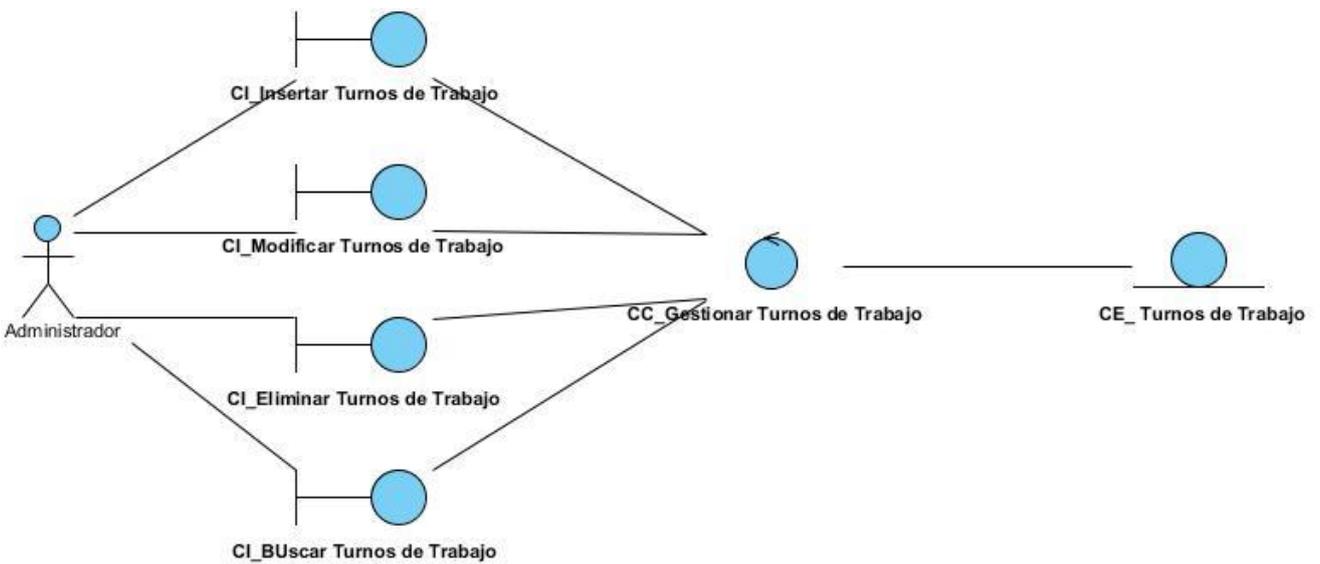


Figura 6: Diagrama de Clases del Análisis (CU-Gestionar Turnos de Trabajo).

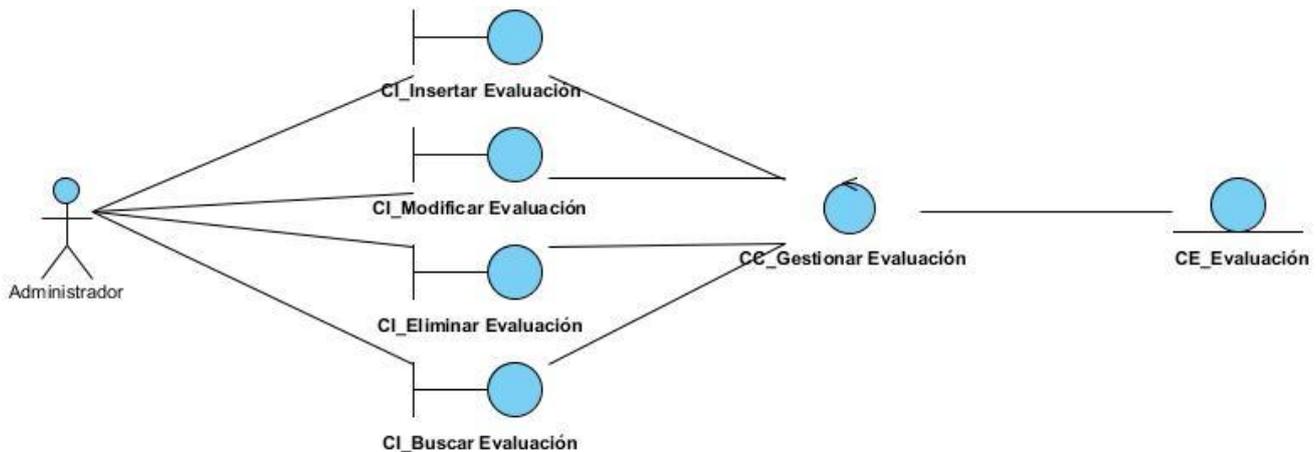


Figura 7: Diagrama de Clases del Análisis (CU-Gestionar Evaluación).

3.3 Patrones arquitectónicos y de diseño

Un estilo arquitectónico es un patrón de diseño o práctica recurrente, expresa esquemas de organización estructural fundamentales para los sistemas de software. Proporcionan un conjunto de subsistemas predefinidos, especifican sus responsabilidades e incluyen guías y lineamientos para organizar las relaciones entre ellos. A continuación se describen los que se han utilizado para la realización del presente trabajo de diploma.(26)

3.3.1 Arquitectura Cliente/Servidor

El modelo Cliente-Servidor se utiliza básicamente para referirse a las PC conectadas en red. El protocolo de Trasmisión y protocolo de internet TCP/IP se rige por lo que se llama básicamente modelo Cliente-servidor. Se divide en dos partes claramente diferenciadas, clientes y servidor. Los clientes por lo general son estaciones de trabajo que acceden desde diferentes servicios al servidor. El servidor por lo general son máquinas potentes que funcionan como un Sistema Gestor de Base de Datos y proveen al cliente el servicio solicitado.

Se designó el modelo cliente-servidor como el más acertado para la realización del sistema que se pretende desarrollar, puesto que las características del sistema encajan a la perfección con las de dicho estilo arquitectónico.

3.4 Patrón Arquitectónico Modelo-Vista-Controlador

El patrón Modelo-Vista-Controlador es una guía para el diseño de arquitecturas de aplicaciones que ofrezcan una fuerte interactividad con usuarios. Este patrón organiza la aplicación en tres modelos separados, el primero es un modelo que representa los datos de la aplicación y sus reglas de negocio, el segundo es un conjunto de vistas que representa los formularios de entrada y salida de información,

el tercero es un conjunto de controladores que procesa las peticiones de los usuarios y controla el flujo de ejecución del sistema. (26)En la Figura 6, se muestra el funcionamiento del mismo.

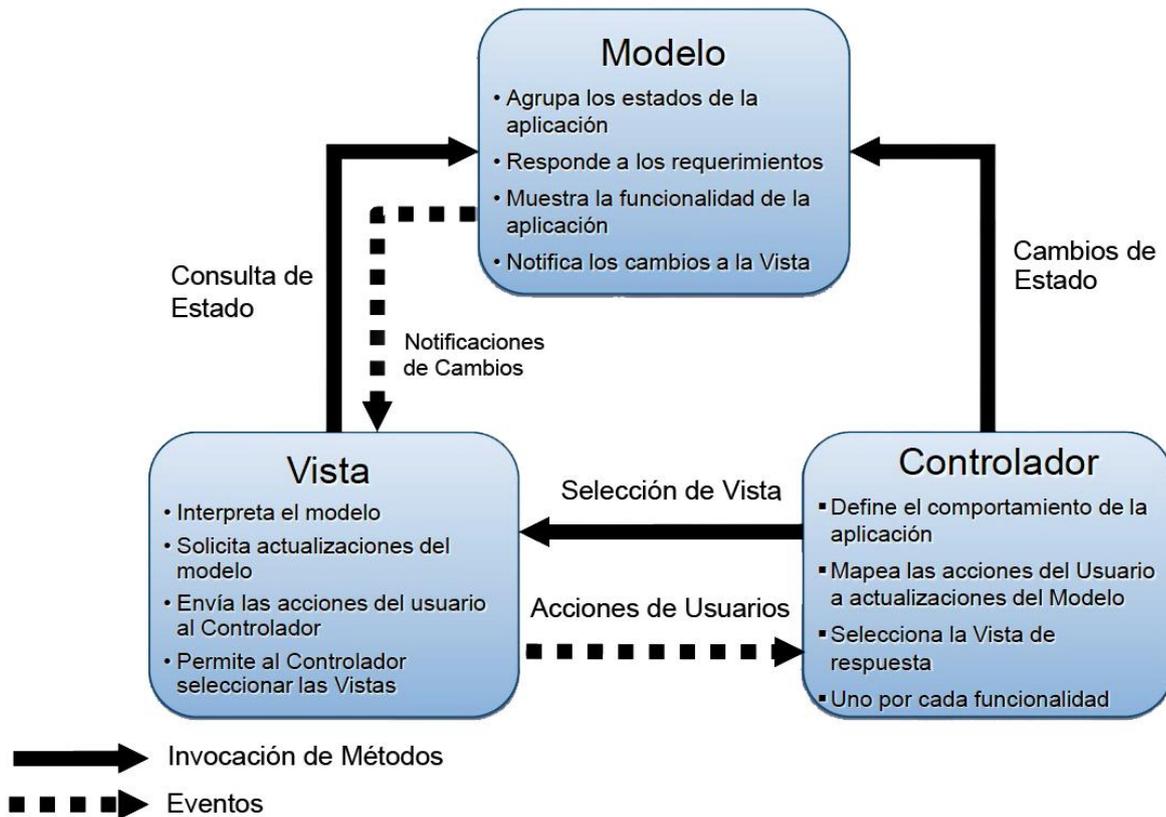


Figura 8: Patrón Modelo-Vista-Controlador.

3.5 Patrones GRASP

Los patrones de diseño GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en formas de patrones. GRASP es un acrónimo que significa General Responsibility Assignment Software Patterns. El nombre se eligió para indicar la importancia de captar estos principios, si se quiere diseñar eficazmente el software orientado a objetos. Se pasa a enumerarlos junto con su definición formal:

- **Experto:** Se encarga de asignar una responsabilidad al experto en información, o sea, aquella clase que cuenta con la información necesaria para cumplir la responsabilidad.
- **Creador:** Permite crear una nueva instancia por la clase que:
 - Tiene la información necesaria para realizar la creación del objeto.
 - Usa directamente las instancias creadas del objeto.

- Almacena o maneja varias instancias de la clase.
- Contiene o maneja la clase.
- **Alta Cohesión:** Asigna una responsabilidad de forma tal que información que maneja una clase sea coherente y esté relacionada con la clase.
- **Bajo Acoplamiento:** Diseñar con el objetivo de tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases.
- **Controlador:** Este patrón tiene la responsabilidad de gestionar un mensaje de un evento del sistema a una clase que represente una de estas dos opciones:
 - Representa el sistema global, dispositivo o subsistema (controlador de fachada).
 - Representa un escenario de caso de uso en el que tiene lugar el evento del sistema (controlador de caso de uso o de sesión).
- **Polimorfismo:** Cuando las alternativas o comportamientos relacionados varían según el tipo, se debe asignar la responsabilidad del comportamiento utilizando operaciones polimórficas a los tipos para los que varía el comportamiento.(26)

3.6 Diagramas de clases del diseño

Los diagramas de clases del diseño se utilizan durante el proceso de análisis y diseño del sistema, en el cual se crea un diseño conceptual de la información que se maneja. Describen gráficamente las especificaciones del sistema. Contiene clases, métodos, atributos, constantes, interfaces con sus operaciones, asociaciones, navegabilidad y dependencias. Se realizó un diagrama de clases del diseño por cada uno de los casos de uso.

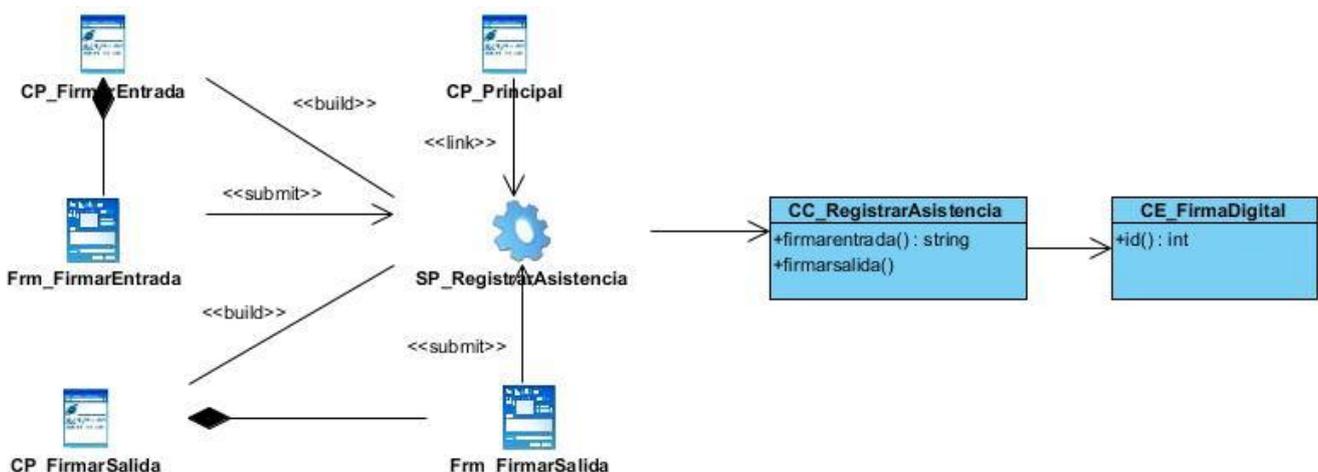


Figura 9: Diagrama de Clases del Diseño (CU-Registrar Asistencia).

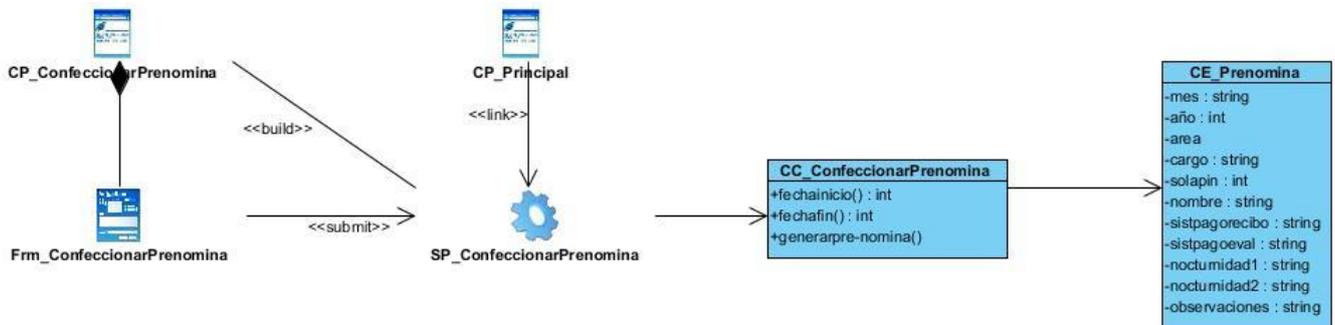


Figura 10: Diagrama de Clases del Diseño (CU-Confeccionar Pre-nómina).

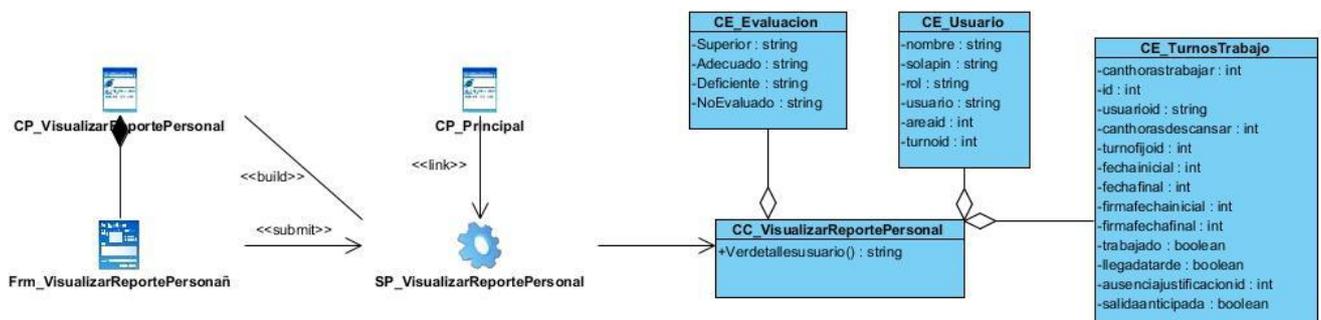


Figura 11: Diagrama de Clases del Diseño (CU-Gestionar Visualizar Reporte Personal).

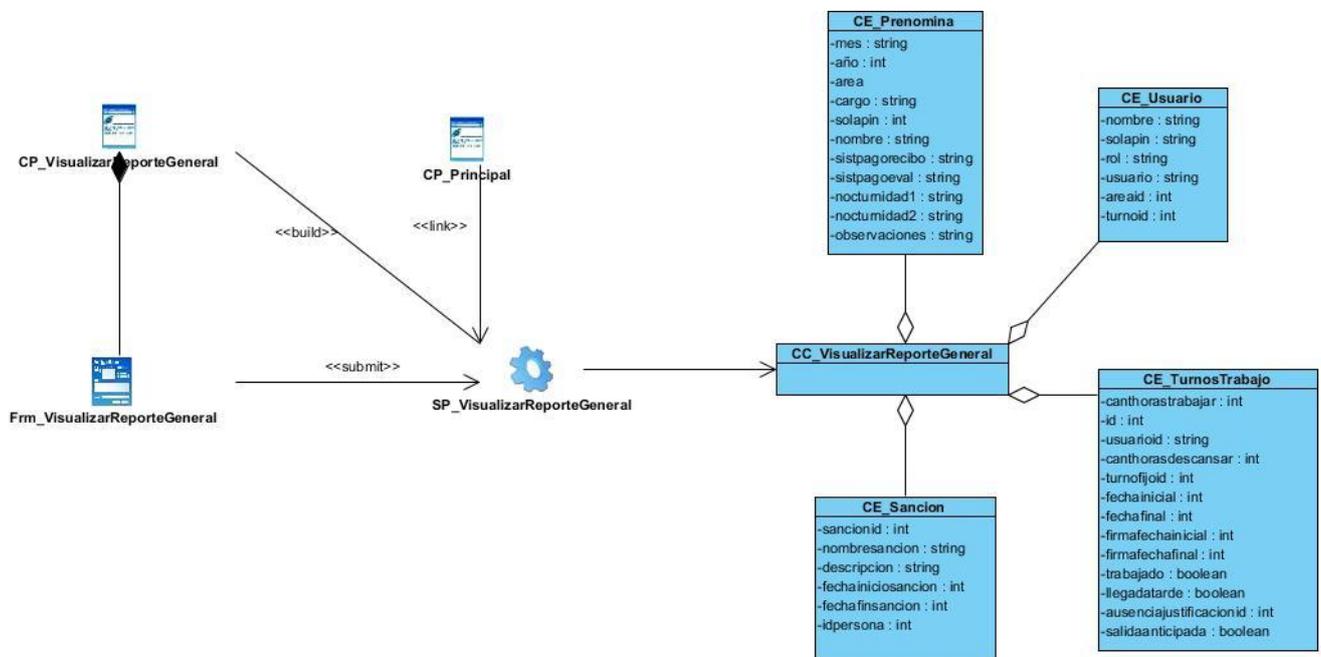


Figura 12: Diagrama de Clases del Diseño (CU-Gestionar Visualizar Reporte General).

3.7 Diseño de la Base de Datos

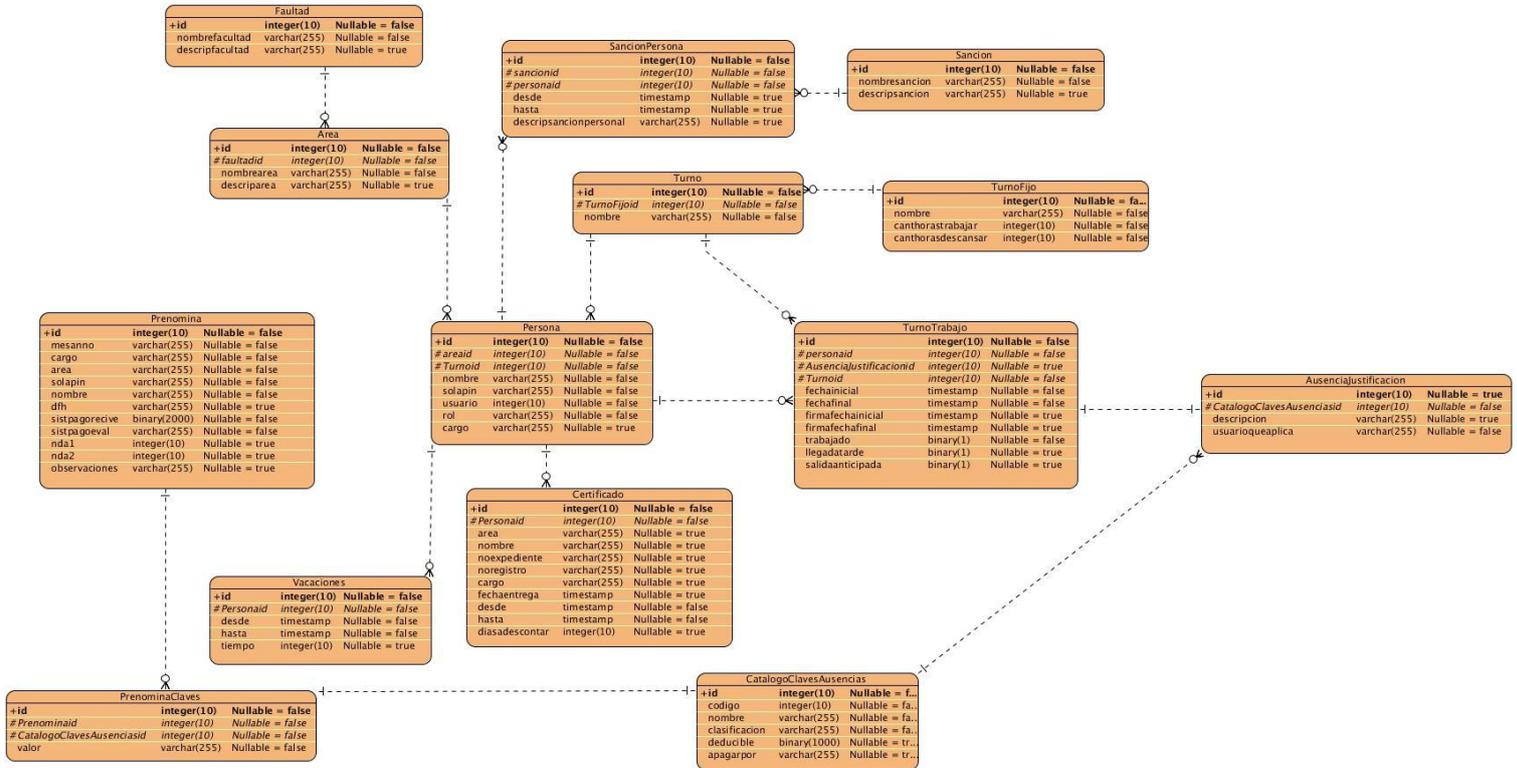


Figura13: Modelo Físico de Datos.

3.8 Conclusiones

Durante este capítulo se realizó el flujo de trabajo Análisis y Diseño propuesto por la metodología RUP, donde se construyeron los diagramas de clases de Análisis y del Diseño aplicando los estereotipos Web. Se realizó el diseño del modelo físico de la base de datos. Una vez culminada la etapa de análisis y diseño se puede avanzar a la siguiente etapa: Construcción y prueba de la solución.

CAPÍTULO IV CONSTRUCCIÓN Y PRUEBA DE LA SOLUCIÓN

4.1 Introducción

En el siguiente capítulo se contempla la implementación del Sistema para el Control de la asistencia y la gestión de la pre-nómina a partir de los diagramas de despliegue y de componentes. Además se realiza un análisis de los resultados obtenidos en la aplicación mediante un conjunto de pruebas realizadas a las funcionalidades.

4.2 Diagrama de Componentes

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean éstos componentes de código fuente, binarios o ejecutables. Permite visualizar con mayor facilidad la estructura general del sistema y el comportamiento del servicio que estos proporcionan y utilizan a través de las interfaces.

4.2.1 Descripción general de los componentes

- ActionClass:



Figura 14: Componente “Action Class”.

Son las clases controladoras, las cuales se encargan de interactuar con el modelo y ejecutar las vistas. Contienen toda la lógica de la aplicación.

- Success:



Figura 15: Componente “Success”.

Son las encargadas de producir las páginas que se muestran como resultado de las acciones.

- ModelClass:

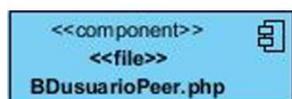


Figura 16: Componente “Model Class”.

Contienen la lógica de negocio y el acceso a los datos (Peer).

A continuación se muestran algunos diagramas de componentes pertenecientes a la presente investigación:

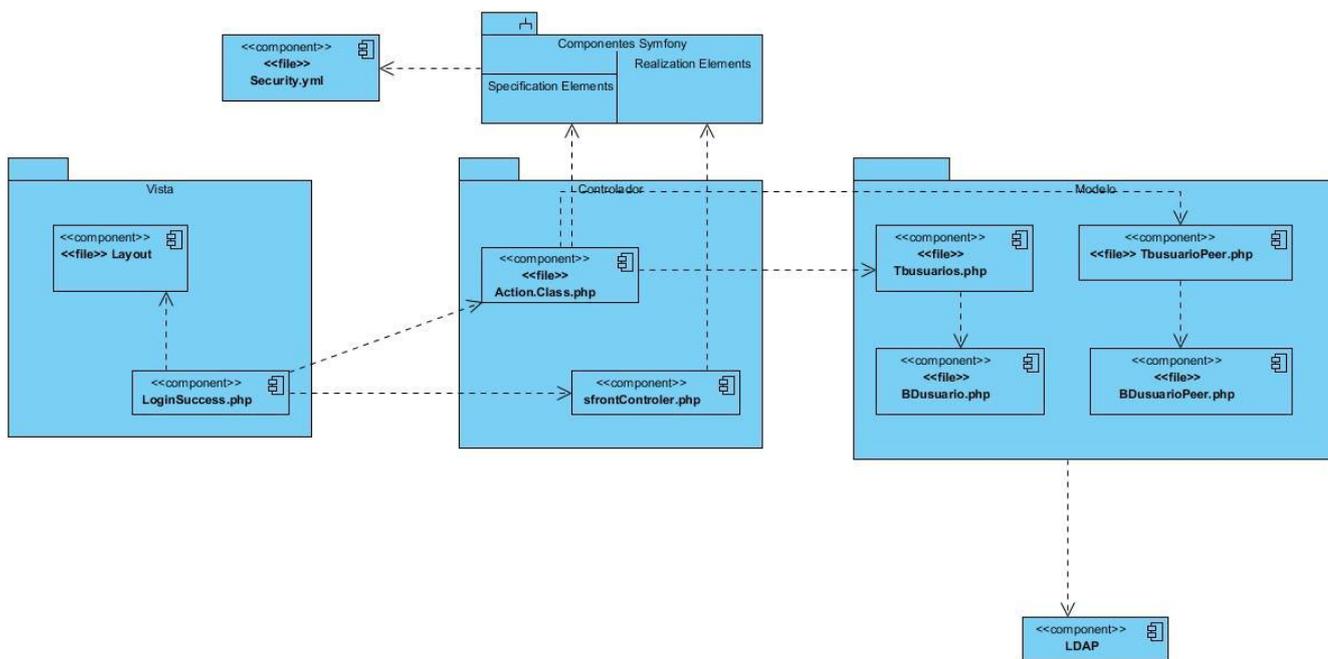


Figura 17: Diagrama de Componentes "Autenticar Usuario".

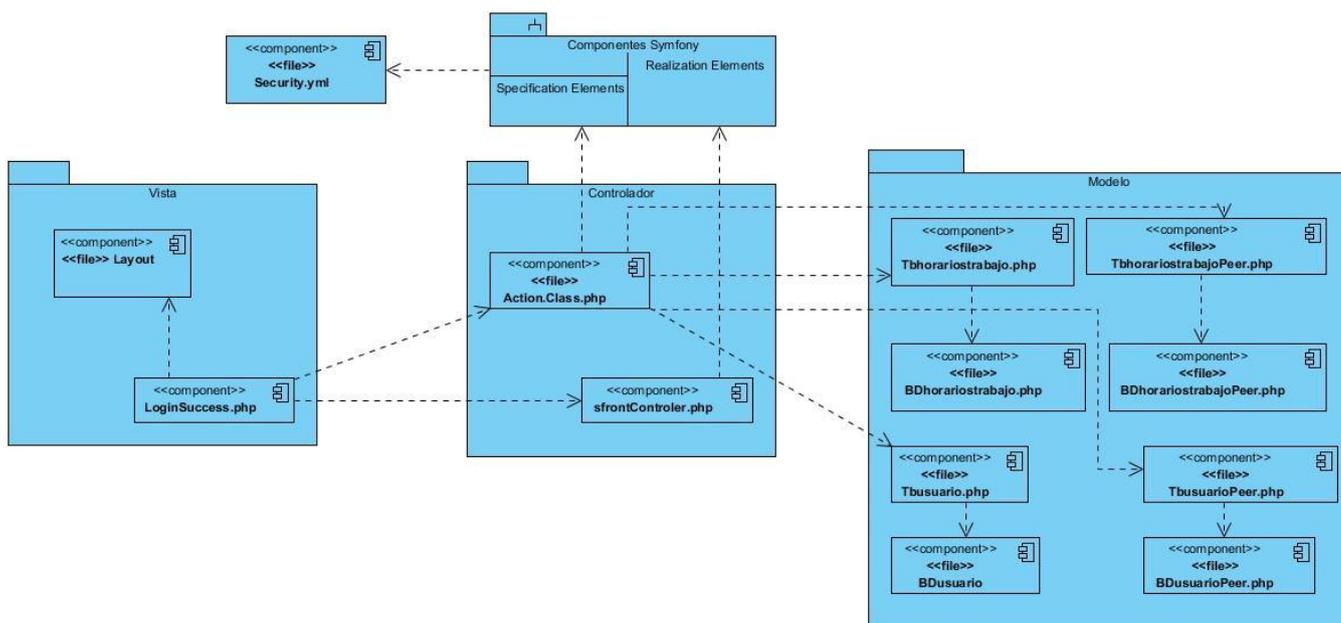


Figura 18: Diagrama de Componentes “Gestionar Turnos de Trabajo”.

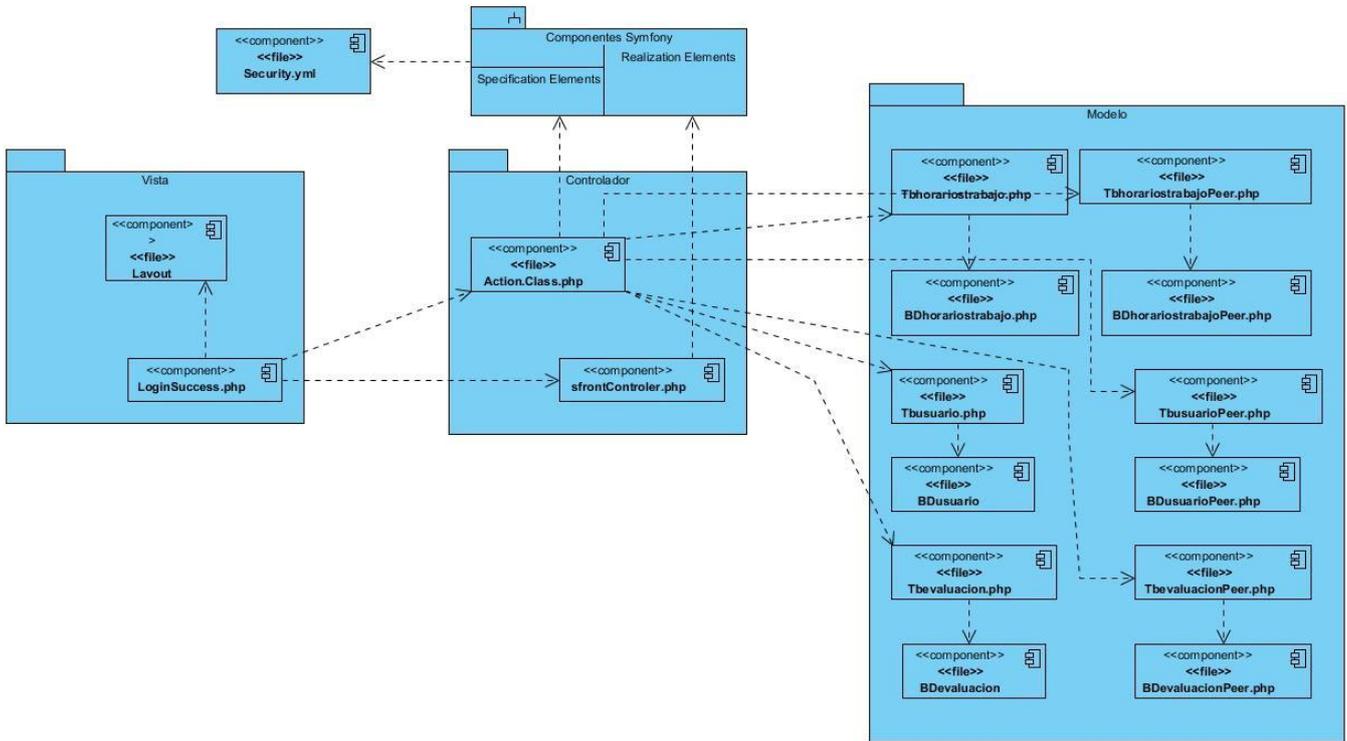


Figura 19: Diagrama de Componentes “Gestionar Pre-nómina”.

4.3 Diagrama de Despliegue

Los diagramas de despliegue son los complementos de los diagramas de componentes que, unidos, proveen la vista de implementación del sistema. Describen la topología del sistema la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos. Los diagramas de despliegue representan a los nodos y sus relaciones.

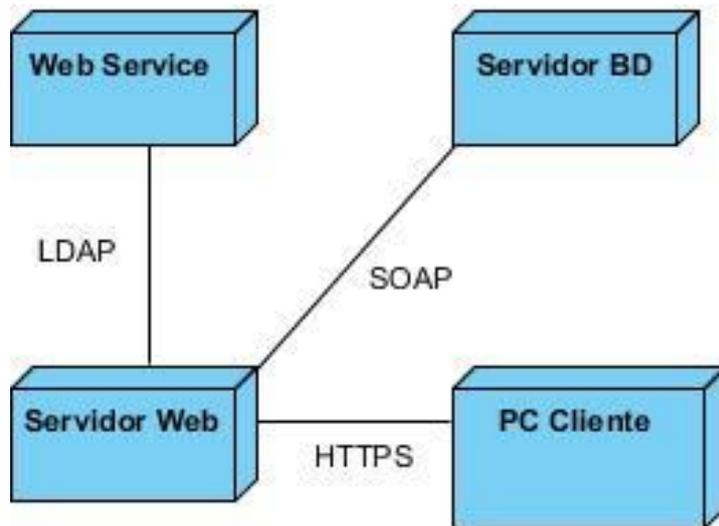


Figura 20: Diagrama de Despliegue.

4.4 Estilos y Estándares de Codificación.

Se refiere a la forma en que se da formato al código fuente. Esto involucra la forma en que se ubican las llaves, se indenta el código y se utilizan los paréntesis. Lo más importante es que el código sea consistente dentro de un programa o una biblioteca. El código con un formato desordenado no es aceptable debido a que es difícil de leer.

Symfony tiene definido en su página principal (<http://www.symfony-project.org/>) sus propios estándares de codificación:

- Nunca use las tabulaciones en el código. La sangría se hace por pasos de 2 espacios:

```

<?php
class sfFoo
{
    public function bar()
    {
        sfCoffee::make();
    }
}
  
```

- No ponga espacios después de un paréntesis de apertura y antes de un cierre:

```

<?php
if ($myVar == getRequestValue($name)) // correcto
if ( $myVar == getRequestValue($name) ) // incorrecto
  
```

- Llaves van siempre en su propia línea.

- Utilice llaves para indicar la estructura de control del cuerpo, independientemente del número de declaraciones que contiene.
- Symfony está escrito en PHP5, por lo que cada método de definición de clase o miembro debe declarar explícitamente su visibilidad utilizando las palabras clave privada, protegida o pública.
- En un cuerpo de la función, las declaraciones de retorno debe tener una línea en blanco antes de él para aumentar la legibilidad.
- Todo lo que uno comente en una línea debe estar en este formato:

```
<?php
// space first, with no full stop needed
```

- Evite la evaluación de las variables dentro de cadenas, en lugar de optar por la concatenación.
- Utilice minúsculas constantes de PHP: falso, verdadero y nulo. Lo mismo ocurre con array (). Al contrario, siempre use cadenas en mayúsculas con las constantes definidas por el usuario, como la define ('MY_CONSTANT', 'foo / bar "). Mejor, trate de utilizar siempre las constantes de clase:

```
<?php
class sfCoffee
{
    const HAZ_SUGAR = true;
}
var_dump(sfCoffee::HAZ_SUGAR);
```

- Para comprobar si una variable es nula o no, no utilice la función de PHP is_null ():

```
<?php
if (null !== $coffee)
{
    echo 'I can haz coffee';
}
```

- Formato de archivo:

Para aquellos archivos que contienen sólo código PHP los tags de demarcación (“<? “) no estarán permitidos, además no es requerido por PHP y omitirlos nos previene de algún accidente ocasionado por un espacio en blanco.

- Parámetros:

Los parámetros van siempre en minúsculas.

- Variables:

El nombre de las variables debe estar compuesto de caracteres alfanuméricos, el carácter Underscore está permitido. Siempre tiene que comenzar con letra minúscula. Además siempre debe inicializarse y sobre todo deben tener nombres significativos.

➤ String literales:

Cuando se le asigna un texto literal (sin contenido de variables) se utilizarán comillas dobles.

```
<?php
$a = "Texto de ejemplo";
```

➤ Concatenación:

Para concatenar Strings se utilizará el operador "." (punto), con un espacio entre medio para mejorar la lectura:

```
<?php
$company = 'Zend' . " " . 'Technologies';
```

➤ Control de flujo:

En las declaraciones if/then/else deberá tener un espacio antes y después del paréntesis condicional, lo mismo se aplica al elseif, a continuación un ejemplo que lo ilustra:

```
<?php
if ($a != 2)
{
    $a = 2;
}
elseif ($a == 3)
{
    $a = 4;
}
else
{
    $a = 6;
}
```

➤ Documentación:

Las complicadas funciones y métodos deberán tener un bloque de documentación. El mismo será entre `/**/` cuando sean de 2 líneas en adelante y `//` cuando sea una sola línea. (33)

El propio Symfony fuerza al desarrollador a crear código legible, fácil de entender y mantener. Por tal motivo se hará uso de estos estándares de codificación definidos por la comunidad de desarrolladores del framework los cuales fueron mencionados anteriormente.

4.5 Pantallas de la Aplicación

Las pantallas de la aplicación no son más que instantáneas tomadas de la aplicación en donde se observan algunas de las principales funcionalidades del sistema durante su funcionamiento. De esta forma se puede apreciar parte de los resultados obtenidos con el presente trabajo.



Figura 21: Pantalla Iniciar Sesión.

La Figura 21 permite apreciarla pantalla de acceso a la aplicación la cual posibilita acceder a las funcionalidades que brinda el sistema, siempre y cuando el usuario y contraseña sea válido.

Hoja de firmas										borisrrf Configuración Administración Salir												
15:32:06																						
UCI							Registro de asistencia							Nombres y Apellidos							Solapin	
Organismo: MES														Boris Rivero							7814	
Dirección:														Mes: Febrero							Año: 2013	
Área: Direccion Lab Facultad 3														Segunda quincena							Nocturnidad	
Primera quincena							Nocturnidad							Segunda quincena							Nocturnidad	
Día	Ent.	Salid.	Firma	Ent.	Salid.	Firma	0,08	0,16	Día	Ent.	Salid.	Firma	Ent.	Salid.	Firma	0,08	0,16					
1									17													
2									18					08:00								
3									19													
4									20													
5									21													
6									22					08:00								
7									23													

Figura 22: Pantalla de Hoja de Firma.

La anterior Figura 22 muestra una tabla con el modelo de la hoja de firma del trabajador. Se puede apreciar los turnos que no ha firmado. Y en caso de coincidir el día actual con el de su turno de trabajo, se habilita la opción para introducir la firma digital y de esta forma validar su asistencia.

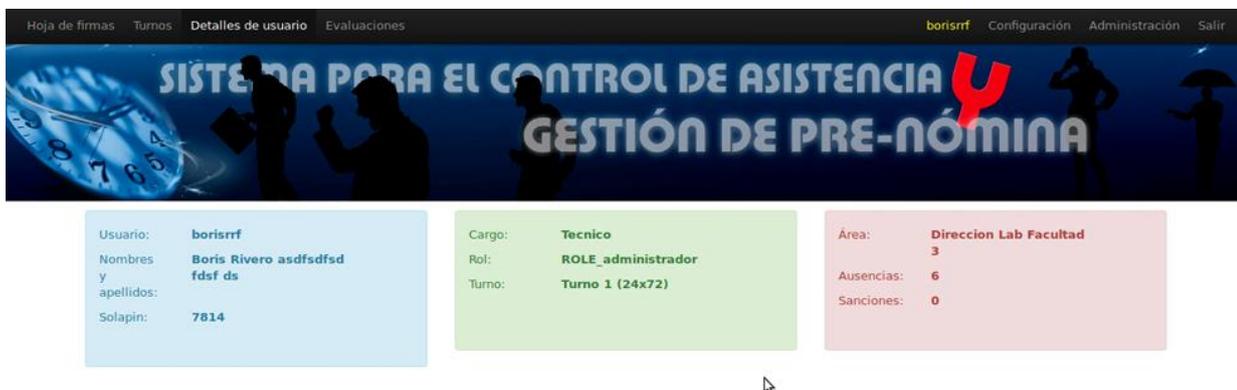


Figura 23: Pantalla de Detalles Usuario.

La Figura 23 muestra un reporte actualizado de los datos personales del usuario que se haya registrado en la aplicación.



Figura23: Pantalla de Mostrar Pre-nómina.

La Figura 23 muestra un reporte de la pre-nómina generada. Se diseñó de forma tal que el modelo coincida con el de capital humano, para facilitar su posterior impresión y aceptación por el departamento de Capital Humano.

4.6 Pruebas de Software

Es el estudio del producto completo o del sistema, ya sea para decidir acerca de su aceptación, para analizar defectos globales o para estudiar aspectos específicos de su comportamiento. Una prueba

tiene éxito si descubre algún defecto y fracasa si existen defectos y estos no son descubiertos durante el periodo de pruebas.

Con el fin de obtener una correcta validación de la especificación de los requisitos del sistema se decidió emplear el método de Pruebas de Caja Negra.

4.6.1 Pruebas de caja negra

También conocidas como Pruebas Funcionales, se centran en lo que se espera de un módulo. El probador se limita a suministrarle datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo el módulo por dentro. Al sistema se le debe pasar un amplio conjunto de datos con el objetivo de analizar la respuesta del sistema en caso de entradas erróneas. Estos casos de prueba corresponden a los casos de uso que necesitan la entrada de datos y cada uno está dividido en tantas secciones como escenarios tenga el caso de uso, incluyendo los flujos de datos alternos. A continuación se exponen los casos de prueba:

Entrada	Respuesta del Sistema	Resultados
El usuario introduce usuario y contraseña y presiona el botón Aceptar.	El sistema verifica que todos los campos estén llenos. Verifica que los datos de entrada correspondan con los que se encuentran en el servidor. Se autentica el usuario.	Satisfactorio
El usuario introduce datos incorrectos.	El sistema verifica que los datos introducidos en el sistema correspondan con los del usuario en cuestión. Al no coincidir, lanza un mensaje de error.	Satisfactorio

Tabla 5: Caso de Prueba "Autenticar Usuario".

Entrada	Respuesta del Sistema	Resultados
El administrador introduce los datos del nuevo usuario y presiona el botón Aceptar.	El sistema verifica que todos los campos estén llenos. Verifica que los datos de entrada correspondan con los que se encuentran se solicitan y que el usuario no haya sido	Satisfactorio

	previamente añadido. Se Adiciona el usuario.	
El administrador deja campos vacíos.	Muestra un mensaje de error "No se pueden dejar los campos vacíos". Brinda nuevamente la posibilidad de incorporar los datos que faltan.	Satisfactorio
El administrador introduce el nombre de un usuario que ya existe y presiona el botón Aceptar	Muestra un mensaje de error "Ya existe el usuario".	Satisfactorio
El administrador introduce los datos del nuevo usuario y presiona el botón Cancelar.	No introduce el usuario	Satisfactorio

Tabla 6: Caso de Prueba Gestionar Usuario, escenario "Adicionar Usuario".

Entrada	Respuesta del Sistema	Resultados
El administrador decide cambiar los datos de un usuario, lo escoge y presiona el botón Aceptar.	El sistema verifica que todos los campos estén llenos. Verifica que los datos de entrada correspondan con los que se encuentran se solicitan. Se actualizan los valores para el usuario.	Satisfactorio
El administrador deja campos vacíos.	Muestra un mensaje de error "No se pueden dejar los campos vacíos". Brinda nuevamente la posibilidad de incorporar los datos que faltan.	Satisfactorio
El administrador introduce los datos del nuevo usuario y presiona el botón Cancelar.	No se realizan cambios en los datos del usuario	Satisfactorio

Tabla 7: Caso de Prueba Gestionar Usuario, escenario "Modificar Usuario".

Entrada	Respuesta del Sistema	Resultados
----------------	------------------------------	-------------------

El administrador decide eliminar un usuario, lo escoge y presiona el botón Sí.	El sistema realiza una pregunta de comprobación: “¿Estás seguro?” y elimina el usuario del sistema.	Satisfactorio
El administrador decide eliminar un usuario, lo escoge y presiona el botón No.	El sistema no elimina al usuario.	Satisfactorio

Tabla 8: Caso de Prueba Gestionar Usuario, escenario “Eliminar Usuario”.

Entrada	Respuesta del Sistema	Resultados
El administrador escoge el criterio de búsqueda (usuario, rol, área, etc.) y selecciona la botón Buscar.	Muestra el (los) usuario correspondiente al criterio buscado.	Satisfactorio
El administrador omite el criterio de búsqueda y presiona el botón Buscar.	El sistema no busca al usuario.	Satisfactorio
El administrador introduce criterios de búsqueda erróneos.	El sistema no encuentra usuarios.	Satisfactorio

Tabla 9: Caso de Prueba Gestionar Usuario, escenario “Buscar Usuario”.

Entrada	Respuesta del Sistema	Resultados
El usuario escoge la opción Detalles de Usuario.	Muestra los detalles del usuario.	Satisfactorio
El usuario marca otra opción.	El sistema no busca los detalles del usuario.	Satisfactorio

Tabla 10: Caso de Prueba Visualizar Reporte Personal.

Entrada	Respuesta del Sistema	Resultados
El usuario presiona el botón Firmar, carga el p12 y escribe la contraseña y selecciona la opción Aceptar.	Valida que la firma y el p12 se correspondan con las del usuario y muestra la hoja de firmas con el espacio “Firmado”.	Satisfactorio
El usuario introduce datos incorrectos.	El sistema no realiza la acción.	Satisfactorio

Tabla 11: Caso de Prueba Registrar Asistencia.

Conclusiones

Al concluir el presente capítulo se logró completar la fase de implementación, se concretaron los diagramas de componentes y se mostraron algunas imágenes o pantallas de la aplicación durante su funcionamiento. Se establecieron escenarios de prueba a la interfaz de usuario con el objetivo de comprobar el correcto funcionamiento de la aplicación.

Conclusiones Generales

Al terminar la presente investigación se arribó a los siguientes resultados:

- Se cumplió la elaboración del marco teórico a partir del estado del arte existente sobre el tema en cuestión.
- Se escogieron las herramientas, metodologías, lenguajes y las tecnologías de desarrollo a utilizar para la implementación del sistema.
- Se identificaron los requerimientos necesarios para la implementación de sistema informático para el control de la asistencia y la gestión de pre-nóminas.
- Se realizó el análisis y diseño del sistema informático.
- Se logró implementar el sistema informático según los requisitos seleccionados.
- Se demostró el correcto funcionamiento del sistema mediante datos de pruebas obteniendo resultados satisfactorios.

Recomendaciones

- Realizar el despliegue del sistema a todas las áreas de la Facultad 3 y posteriormente a otras áreas de la UCI.
- Incorporar nuevos elementos y funcionalidades al sistema con el objetivo de aumentar su alcance. Por ejemplo, se le pudiera incluir un módulo para la gestión de las guardias obreras.
- Realizar un manual de usuario y de administración.

Bibliografía

1. **SlideShare.** SlideShare. [En línea] [Citado el: 20 de 10 de 2012.] <http://www.slideshare.net/normyser/qu-son-los-sistemas-de-informacin-cap-1>.
2. **Enomina, S.L.** Enomina Personas y Tecnología. [En línea] 2012. [Citado el: 20 de 10 de 2012.] <http://www.enomina.net/ayuda/contenido/procesos.html>.
3. **GrupoTress.** Grupo Tress Internacional. [En línea] [Citado el: 23 de 10 de 2012.] <http://www.tress.com.mx/esp/LinkClick.aspx?fileticket=Vby5GtGCO8%3D&tabid=61>.
4. **Infinite Consulting.SA.** Infinite Consulting. C.A. [En línea] MANTIA ®. [Citado el: 21 de 10 de 2012.] <http://www.infiniteconsulting.com.ve/>.
5. **INTERSHARE, S.L.** 'Doce' versión 2. [En línea] 2012. [Citado el: 23 de 10 de 2012.] <http://doce2.softonic.com/>.
6. **Empresa Nacional de Software, Desoft.** Portal de Matanzas. [En línea] Fastos y Pagos RRHH y Nóminas, 2011. [Citado el: 23 de 10 de 2012.] <http://www.expomatanzas.cu/empresa.php?prd=936&emp=151&catprd=5>.
7. **Rodas XXI.** Rodas XXI. [En línea] [Citado el: 10 de diciembre de 2012.] <http://www.rodasxxi.cu/>.
8. **Roberth G. Figueroa, Camilo J. Solís,.** METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES. [En línea] 2012. [Citado el: 10 de 12 de 2012.] http://www.google.com/cu/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&ved=0CCwQFjAA&url=http%3A%2F%2Fadonisnet.files.wordpress.com%2F2008%2F06%2Farticulo-metodologia-de-sw-formato.doc&ei=Unv7UOm8B4Gc8gTUsoGACQ&usg=AFQjCNH6ClwoJOFA1MLY_6yuYhpBZUKGFQ&bv.
9. **Patricio Letelier, M^a Carmen Penadés.** Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). [En línea] abril de 2006. [Citado el: 12 de 12 de 2012.] <http://www.cyta.com.ar/ta0502/v5n2a1.htm>. ISSN 1666-1680.

10. **Alfaro, Econ. Félix Murillo.** Instituto Nacional de Estadística e Informática. [En línea] 1999. [Citado el: 14 de 12 de 2012.] <http://www.inei.gob.pe/biblioineipub/bancopub/Inf/Lib5103/Libro.pdf>.
11. **Brigitte Prestegui, Cherry.** Scrib. [En línea] Scrib Inc., 2012. [Citado el: 12 de 12 de 2012.] <http://es.scribd.com/doc/41067750/Lenguaje-Unificado-de-Modelado>.
12. **María Elena Tirira.** bibdigital. [En línea] 2008. [Citado el: 14 de 12 de 2012.] <http://bibdigital.epn.edu.ec/bitstream/15000/2140/1/CD-1839%282009-01-21-12-32-39%29.pdf>.
13. **Paradigma visual para UML .** [En línea] 2007. [Citado el: 14 de 12 de 2012.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/.
14. **Ecured.** Ecured. [En línea] 2011. [Citado el: 14 de 12 de 2012.] http://www.ecured.cu/index.php/Lenguaje_de_Programaci%C3%B3n.
15. **MATEO, ARIS.** Software LIBRE. [En línea] 2009. [Citado el: 14 de 12 de 2012.] http://www.oocities.org/es/mateoa_postgrado/ads/TFADS.htm.
16. **Yuri Bernal.** Slideshare-Desarrollo de Aplicaciones Comerciales en Java. [En línea] 2009. [Citado el: 15 de 12 de 2012.] <http://www.slideshare.net/guest044583b/javascript-1562899>.
17. **Miguel Angel Alvarez.** Desarrollo Web.com. [En línea] 2004. [Citado el: 25 de enero de 2013.] <http://www.desarrolloweb.com/articulos/182.php>.
18. **Egluiluz, Javiel.** LibrosWeb. [En línea] 2009. [Citado el: 14 de 12 de 2012.] http://librosweb.es/symfony_1_4/capitulo_1/symfony_en_pocas_palabras.html.
19. **Zisman, Michael.** netbeans.org. A Brief History of NetBeans. [En línea] 2012. <http://netbeans.org/about/history.html> .
20. **N D, Daver.** WordPress.com. Scrolling – jQuery | WebMatrix. [En línea] 2010. [Citado el: 23 de enero de 2013.] <http://daverndn.wordpress.com/2012/10/09/scroll-horizontal-jquery-webmatrix/>.
21. Bootstrap de twitters para web adaptables. [En línea] 2013. [Citado el: 28 de enero de 2013.] <http://bcn-cluster.com/bootstrap-web-adaptable-responsive/>.
22. **M A, Rafael.** Postgresql.org. Sobre PostgreSQL. [En línea] 2010. [Citado el: 24 de 12 de 2012.] http://www.postgresql.org/es/sobre_postgresql.
23. **Lynch, Kevyn.** sqlmanager. Administrador EMS de PostgreSQL. [En línea] 2009. http://www.freedownloadmanager.org/es/downloads/postgresql_pgadmin_libre/.
24. **Ralp R. Young.** Entorno Virtual de Aprendizaje. [En línea] 2004. [Citado el: 14 de 12 de 2012.] http://eva.uci.cu/file.php/161/Documentos/Materiales_basicos/Materiales_basicos_de_la_Unidad_2/The_Requirements_Engineering_HandBook.pdf.

25. **María José Escalona, Nora Koch.** Ingeniería de Requisitos en Aplicaciones para la Web – Un estudio comparativo. [En línea] 2002. [Citado el: 15 de enero de 2013.] http://www.sistemas.edu.bo/Isalgado/sis3390/Requisitos/Ingenier_a%20de%20Requisitos%20en%20Aplicaciones%20Web%20-%20Un%20estudio%20compatativo.pdf.
26. **moodle.** Comunidad Virtual Moodle de la Universidad del Cauca. [En línea] 2012. [Citado el: 15 de 12 de 2012.] HTTP://pis.unicauca.edu.co/moodle/file.php/312/.../Estilos_y_patrones.doc.
27. **Ian Sommerville.** Entorno Virtual de Aprendizaje. [En línea] 2004. [Citado el: 14 de 12 de 2012.] http://eva.uci.cu/file.php/161/Documentos/Materiales_basicos/Materiales_basicos_de_la_Unidad_2/Otros_materiales/0_Enfoques_de_la_IR/Requirements_Engineering_Processes.pdf.
28. **Quintín, Saul Cuzcano.** Scrib. [En línea] 2012. <http://es.scribd.com/doc/7930106/Modelo-de-Analisis-Saul-Cuzcano-Quintin>.
29. **Gutiérrez., Javier J.** Departamento de Lenguajes y Sistemas Informáticos. [En línea] 2012. [Citado el: 15 de 12 de 2012.] http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf.
30. **Andrés Grosso.** Prácticas de Software. [En línea] 21 de marzo de 2011. [Citado el: 15 de enero de 2013.] <http://www.practicadesoftware.com.ar/2011/03/patrones-grasp/>.
31. **Ecured.** Ecured. [En línea] 2011. [Citado el: 15 de enero de 2012.] http://www.ecured.cu/index.php/Diagrama_de_despliegue.
32. **Turner, Kevin.** es.kioskea.net. Introducción - Bases de datos. [En línea] [Citado el: 20 de enero de 2013.] <http://es.kioskea.net/contents/bdd/bddintro.php3>.
33. **Fontela, Carlos.** Técnicas de aseguramiento de la calidad del producto. CyS Ingeniería de Software [En línea] 19 de enero de 2009. [Citado el 10 de 01 de 2013] <http://cysingsoft.wordpress.com/2008/06/24/tecnicas-de-aseguramiento-de-la-calidad-del-producto/>.

Anexos

Especificación de Casos de Uso del sistema:

- **CU 1. Autenticar Usuario**

Objetivo	El usuario introduce usuario y contraseña para entrar en el sistema.	
Actores	Usuario: (Inicia) Se autentica en el sistema.	
Resumen	El usuario introduce usuario y contraseña y se introduce en el sistema.	
Complejidad	Media	
Prioridad	Crítico	
Precondiciones	El usuario debe estar previamente registrado en el sistema.	
Postcondiciones	El usuario está autenticado y puede acceder al sistema.	
Flujo de eventos		
Flujo básico <Autenticar Usuario>		
	Actor	Sistema
1.	Introduce usuario y contraseña y presiona clic en el botón Aceptar.	Valida que los datos son correctos y muestra al usuario la interfaz principal del sistema. Termina el caso de uso.
Flujos alternos		
Nº Evento <Los datos son incorrectos>		
	Actor	Sistema
1.	Introduce usuario y contraseña incorrectos y presiona clic en el botón Aceptar.	Muestra un mensaje de error .
2.	Rectifica los datos y presiona clic en el botón Aceptar.	Valida que los datos son correctos y muestra al usuario la interfaz principal del sistema. Termina el caso de uso.
Relaciones	CU Incluidos	

	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

- CU 2. Gestionar Facultad**

Objetivo	El administrador adiciona, modifica, elimina o busca una facultad.	
Actores	Administrador: (Inicia)	
Resumen	Adiciona, modifica o elimina los datos de una facultad en el sistema.	
Complejidad	Media	
Prioridad	Secundario	
Precondiciones	-	
Postcondiciones	La facultad ha sido creada, modificada o eliminada.	
Flujo de eventos		
Flujo básico <Gestionar Facultad>		
	Actor	Sistema
2.	Desea crear, modificar o eliminar una facultad.	Permite realizar varias opciones: <ul style="list-style-type: none"> - Crear una nueva facultad. Sección 1: "Crear facultad". - Modificar una facultad. Sección 2: "Modificar facultad". - Eliminar una facultad. Sección 3: "Eliminar facultad".
Sección 1: "Adicionar facultad"		
Flujo básico < Adicionar facultad>		
	Actor	Sistema

1.	Escoge la opción Crear facultad.	<p>Muestra una ventana con los siguientes datos:</p> <ul style="list-style-type: none"> - Nombre. - Descripción. 
2.	Llena los datos y presiona el botón Adicionar.	<p>Adiciona una facultad.</p> <p>Termina el caso de uso.</p>

Flujos alternos

2 <Dejar campos vacíos>

	Actor	Sistema
2.	Deja campos vacíos y presiona el botón Adicionar.	Muestra un mensaje de error: “Este valor no debería estar vacío”.

Sección 2: “Modificar facultad”

Flujo básico < Modificar facultad>

	Actor	Sistema
1.	Escoge la opción Modificar facultad.	<p>Muestra una ventana con los siguientes datos:</p> <ul style="list-style-type: none"> - Nombre. - Descripción.

		<p>Nombre: <input type="text" value="Facultad 3"/></p> <p>Descripción: <input type="text"/></p> <p><input type="button" value="Editar"/></p> <p><input type="button" value="Borrar"/></p> <p>Regresar</p>
2.	Cambia los datos y presiona el botón Editar.	<p>Modifica una facultad.</p> <p>Termina el caso de uso.</p>
Flujos alternos		
2 <Dejar campos vacíos>		
	Actor	Sistema
2.	Deja campos vacíos y presiona el botón Adicionar.	Muestra un mensaje de error: "Este valor no debería estar vacío".
Sección 3: "Eliminar facultad"		
Flujo básico < Eliminar facultad>		
	Actor	Sistema
1.	Escoge la facultad a Eliminar.	Muestra la facultad con el botón eliminar.
2.	Presiona el botón Eliminar.	Muestra el mensaje "¿Estás seguro?"
3.	Escoge la opción Si	<p>Elimina la facultad.</p> <p>Termina el caso de uso.</p>
Flujos alternos		
3 <Escoger la opción No>		
	Actor	Sistema
2.	Escoger la opción No.	<p>Muestra la interfaz con la lista de facultades.</p> <p>Termina el caso de uso.</p>
Relaciones	CU Incluidos	

	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

• **CU 3. Gestionar Área**

Objetivo	El administrador adiciona, modifica, elimina o busca un área.	
Actores	Administrador (Inicia).	
Resumen	Adiciona, modifica o elimina los datos de una facultad en el sistema.	
Complejidad	Media	
Prioridad	Secundario	
Precondiciones	Debe existir al menos una facultad.	
Post condiciones	El área ha sido creada, modificada, eliminada o encontrada.	
Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
3.	Desea crear, modificar, eliminar o buscar un área.	Permite realizar varias opciones: <ul style="list-style-type: none"> - Crear una nueva área. Sección 1: "Crear área". - Modificar una facultad. Sección 2: "Modificar área". - Eliminar un área. Sección 3: "Eliminar área". - Buscar área. Sección 4: "Buscar".

Sección 1: "Crear área"

Flujo básico <Crear área>

	Actor	Sistema
1.	Escoge la opción Crear área.	Muestra una ventana con los siguientes datos: - Nombre. - Descripción. - Facultad. 
2	Introduce los datos y presiona el botón Adicionar.	Adiciona un área al sistema. Termina el caso de uso.

Flujos alternos

2 <Deja el campo nombre vacío>

	Actor	Sistema
1.	Deja el campo nombre vacío y presiona el botón Adicionar.	Muestra el mensaje de error "Este valor no debería estar vacío".
2.	Introduce los datos y presiona el botón Adicionar.	Adiciona el área al sistema. Termina el caso de uso.

Sección 2: "Modificar área"

Flujo básico <Crear área>

	Actor	Sistema
1.	Escoge la opción Modificar área.	Muestra una ventana con los siguientes datos:

		<p>- Nombre.</p> <p>- Descripción.</p> <p>- Facultad.</p> <p>Nombre: <input type="text" value="Direccion Laboratorio"/></p> <p>Descripción: <input type="text"/></p> <p>Facultad: <input type="text" value="Facultad 3"/></p> <p><input type="button" value="Editar"/></p> <p><input type="button" value="Borrar"/></p> <p>Regresar</p>
2	Introduce los datos y presiona el botón Editar.	<p>Modifica un área del sistema.</p> <p>Termina el caso de uso.</p>
Flujos alternos		
2 <Deja el campo nombre vacío>		
	Actor	Sistema
1.	Deja el campo nombre vacío y presiona el botón Adicionar.	Muestra el mensaje de error” Este valor no debería estar vacío”.
2.	Introduce los datos y presiona el botón Editar.	<p>Modifica el área en el sistema.</p> <p>Termina el caso de uso.</p>
Sección 3: “Eliminar área”		
Flujo básico <Crear área>		
	Actor	Sistema
1.	Escoge el área a eliminar.	Muestra una ventana con el área a eliminar
2.	Presiona el botón Borrar.	Muestra el mensaje: “¿Estás seguro?”
3.	Presiona el botón Si.	<p>Borra un área del sistema.</p> <p>Termina el caso de uso.</p>
Flujos alternos		

2 <Escoge la opción Regresar>		
	Actor	Sistema
1.	Presiona el botón Regresar.	Muestra el listado de áreas. Termina el caso de uso.
Sección 3: “Buscar área”		
Flujo básico <Buscar área>		
	Actor	Sistema
1.	Escoge la facultad donde se buscará el área deseada y presiona el botón Buscar.	Muestra las áreas correspondientes a la facultad escogida. Termina el caso de uso.
Flujos alternos		
1 <No escoge la facultad donde se encuentra el área>		
	Actor	Sistema
1.	No escoge la facultad y presiona el botón Buscare.	No realiza acciones. Termina el caso de uso.
Relaciones	CU Incluidos	Ver CU Gestionar Facultad
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

- **CU 4. Gestionar Turno de trabajo**

Objetivo	El administrador adiciona, modifica, elimina o busca un turno.
Actores	Administrador (Inicia).

Resumen	Adiciona, modifica o elimina los datos de un turno de trabajo en el sistema.	
Complejidad	Media	
Prioridad	Crítico.	
Precondiciones	Debe existir al menos un área de trabajo.	
Postcondiciones	El turno ha sido creado, modificado, eliminado o encontrado.	
Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
4.	Desea crear, modificar, eliminar o buscar un turno de trabajo.	Permite realizar varias opciones: <ul style="list-style-type: none"> - Crear un nuevo turno de trabajo. Sección 1: "Crear turno". - Modificar un turno de trabajo. Sección 2: "Modificar". - Eliminar un turno de trabajo. Sección 3: "Eliminar turno". - Buscar turno de trabajo. Sección 4: "Buscar".
Sección 1: "Crear turno"		
Flujo básico <Crear turno>		
	Actor	Sistema
2.	Escoge la opción Crear turno.	Muestra una ventana con los siguientes datos: <ul style="list-style-type: none"> - Nombre. - Horario.

--	--	--

2	Introduce los datos y presiona el botón Adicionar.	<p>Adiciona un turno al sistema.</p> <p>Termina el caso de uso.</p>
---	--	---

Flujos alternos

2 <Deja el campo nombre vacío>

	Actor	Sistema
1.	Deja el campo nombre vacío y presiona el botón Adicionar.	Muestra el mensaje de error” Este valor no debería estar vacío”.
2.	Introduce los datos y presiona el botón Adicionar.	<p>Adiciona el turno al sistema.</p> <p>Termina el caso de uso.</p>

Sección 2: “Modificar turno”

Flujo básico <Modificar turno>

	Actor	Sistema
2.	Escoge el turno en el listado y presiona el botón Editar.	<p>Muestra una ventana con los siguientes datos:</p> <ul style="list-style-type: none"> - Nombre. - Horario.

		<p>Nombre:</p> <input type="text" value="Turno 1"/> <p>Turno fijo:</p> <input type="text" value="24x72"/> <p>Editar</p> <p>Borrar</p> <p>Regresar</p>
2	Modifica los datos y presiona el botón Editar.	<p>Modifica un turno en el sistema.</p> <p>Termina el caso de uso.</p>

Flujos alternos

2 <Deja el campo nombre vacío>

	Actor	Sistema
1.	Deja el campo nombre vacío y presiona el botón Modificar.	Muestra el mensaje de error" Este valor no debería estar vacío".
2.	Introduce los datos y presiona el botón Editar.	<p>Modifica el turno en el sistema.</p> <p>Termina el caso de uso.</p>

Sección 3: "Eliminar turno"

Flujo básico <Eliminar turno>

	Actor	Sistema
1.	Escoge el turno a eliminar.	Muestra una ventana con el turno a eliminar.
2.	Presiona el botón Borrar.	Muestra el mensaje de error:"¿Estás seguro?"
3.	Presiona el botón Sí.	<p>Borra un turno del sistema.</p> <p>Termina el caso de uso.</p>

Flujos alternos

2 <Escoge la opción Regresar>

	Actor	Sistema
1.	Presiona el botón Regresar.	Muestra el listado de turnos de trabajo.

		Termina el caso de uso.
Sección 3: “Buscar turno”		
Flujo básico <Buscar turno>		
	Actor	Sistema
2.	Escoge el o los criterios de búsqueda y presiona el botón Buscar.	 <p>Muestra los turnos correspondientes deseados.</p> <p>Termina el caso de uso.</p>
Flujos alternos		
1 <No escoge criterios de búsqueda>		
	Actor	Sistema
1.	No escoge criterio de búsqueda y presiona el botón Buscar.	No realiza acciones. Termina el caso de uso.
Relaciones	CU Incluidos	<u>Ver CU Generar Turnos de trabajo</u>
	CU Extendidos	
Requisitos no funcionales		