



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 3

Centro de Gobierno Electrónico

**Título: Sistema informático para la gestión administrativa de los Recursos Humanos en el
Centro de Gobierno Electrónico.**

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

Autor: Gilmar Lázaro Díaz Pérez

**Tutores: Ing. Yenier Figueroa Machado
Ing. Darián González Ochoa**

Cotutor: Ing. Alejandro David Álvarez Sosa

**La Habana, junio de 2016
“Año 58 de la Revolución”**

DECLARACIÓN DE AUTORÍA

Declaro ser el autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste, firmo la presente a los 23 días del mes de junio del año 2016.

Gilmar Lázaro Díaz Pérez
Autor

Ing. Yenier Figueroa Machado
Tutor

Ing. Darián González Ochoa
Tutor

DATOS DE CONTACTO

Síntesis del Tutor Yenier Figueroa Machado

El ingeniero Yenier Figueroa Machado, es graduado en Ciencias Informáticas en el año 2007, durante su trayectoria laboral ha ocupado diferentes responsabilidades en las dirección de proyecto y departamentos docente. Tiene una buena referencia a publicaciones en eventos nacionales e internacionales que avalan la profesionalidad de su desempeño en el área.

Email: yfigueroa@uci.cu

Síntesis del Tutor Darián González Ochoa

El Ingeniero Darián González Ochoa, es graduado de la Universidad de las Ciencias Informáticas en el año 2009, durante su trayectoria laboral ha ocupado varias responsabilidades dentro del desarrollo de software específicamente del Centro de Gobierno Electrónico donde actualmente ocupa el cargo de Director de dicho centro. Tiene en su haber varias publicaciones científicas enmarcadas en el desarrollo de sistemas para la administración pública.

Email: dochoa@uci.cu

Síntesis del Cotutor Alejandro David Álvarez Sosa

El Ingeniero Alejandro David Álvarez Sosa, es graduado en Ciencias Informáticas en el año 2015, se desempeña como programador del proyecto para la Empresa de Seguros Nacionales (ESEN) del Centro de Informatización de Entidades.

Email: adsosa@uci.cu

DEDICATORIA

A mi abuela, mi mamá y mi tía, que lo han dado todo por mí, por su amor, esfuerzo y sacrificio, porque sin su confianza y apoyo no hubiese llegado hasta aquí. A mis amigos de adentro y fuera de la universidad que me han aportado miles de experiencias y han aguantado mis excentricidades. A mi pareja por estar siempre pendiente de mí en las buenas y malas y querer conmigo un futuro. Muchas gracias a todos.

AGRADECIMIENTOS

Este trabajo de diploma representa en mi vida el acontecimiento que marca un antes y un después, la experiencia universitaria, camino recorrido lleno de ganancias y pérdidas; la entrega a un futuro que me invita a soñar y a luchar por mis sueños. En estos años mi vida ha sido bendecida con personas especialmente maravillosas, a las cuales agradezco infinitamente por haber dejado su huella en mí, porque sin su presencia no hubiera sido posible este momento.

A Dios por el regalo de estos 5 años, por cada una de las experiencias vividas y por cada persona con la que he compartido. De manera especial por su fidelidad, por ser el sentido de mi vida y guiar mis pasos, por haberme dado la fuerza para llegar hasta este momento.

A toda mi familia tan grande y unida, por estar siempre atenta a mi formación personal y profesional, por su confianza que para mí es lo más importante. A mi abuela, mi mamá, mi tía, por tanto amor y dedicación, por sus esfuerzos para que yo llegara a este momento. A mi primo por su cariño y por regalarme momentos de alegría.

A mis amigos de la UCI, gracias por permitirme compartir junto a ustedes, aprender, reír, llorar, por aguantar mis sermones y también mis malcriadeces.

A Suset, Yesenia, Joenis y Pedro que han sido más que amigos, como hermanos, por aguantarme todos estos años, por aceptarme como soy, por todos los momentos que hemos compartido, por apoyarme siempre, por su amistad incondicional.

A Carlos que en tan poco tiempo se han ganado mi amor, por su apoyo incondicional muchas gracias.

A todos aquellos profesores que durante estos 5 años han pasado por mi vida dejando sus enseñanzas y de manera especial a aquellos que nunca han dejado de estar, que han sido profesores y amigos.

A mis tutores y mi cotutor por su apoyo, paciencia y confianza en mí, por su exigencia y crítica en la elaboración de este trabajo.

A todas aquellas personas que de una forma u otra han contribuido a la realización de este proyecto.

RESUMEN

La elaboración de la nómina es uno de los procesos que tiene en una institución el departamento de los Recursos Humanos y relaciona por cada trabajador el tiempo correspondiente a ausencias, impuntualidades, licencias, subsidios, vacaciones y otros factores que incidan en el tiempo a devengar. En el Centro de Gobierno Electrónico el funcionamiento de este proceso se ve afectado, por lo que se hace necesario y es objetivo del presente trabajo de diploma, desarrollar una aplicación web que permita su gestión administrativa. Para la implementación se tuvo en cuenta el proceso de desarrollo de software guiado por la metodología XP, el marco de trabajo Symfony2, el servidor web Apache, el IDE PhpStorm y el gestor de base de datos PostgreSQL. Este producto fue validado a través de pruebas unitarias y de aceptación, además se utilizó la técnica de ladov para medir el grado de satisfacción de los clientes, obteniéndose excelentes resultados para el universo definido.

Palabras claves: Aplicación web, gestión administrativa, Recursos Humanos.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	5
1.1. Introducción	5
1.3. Sistemas que informatizan el proceso de prenomina	6
1.4. Ingeniería de Software	11
1.5. Lenguaje de Programación	15
1.6. Herramientas CASE	16
1.7. Marco de Trabajo	17
1.8. Servidor Web	18
1.9. Sistema Gestor de Bases de Datos	19
1.10. Entorno de Desarrollo Integrado.....	20
1.11. Variables de la investigación	20
1.12. Conclusiones parciales	22
CAPÍTULO 2. DESCRIPCIÓN DE LA PROPUESTA	24
2.1. Introducción	24
2.2. Relación entre los principales conceptos del negocio.....	24
2.3. Descripción del proceso de negocio prenomina	25
2.4. Descripción de la propuesta de solución	26
2.5. Levantamiento de Requisitos.....	28
2.6. Historias de Usuario	31
2.7. Diseño	40
2.8. Conclusiones parciales	45
CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA	46
3.1. Introducción	46
3.2. Validación de las variables de la investigación	46
3.3. Validación de la Fase de Desarrollo	47
3.3. Conclusiones parciales	53
CONCLUSIONES GENERALES	55
RECOMENDACIONES	56

ÍNDICE

REFERENCIAS BIBLIOGRÁFICAS	57
ANEXOS	60

INTRODUCCIÓN

El área de Recursos Humanos de toda empresa estatal necesita tener establecido un registro primario por métodos automatizados, mecánicos o manuales para anotar diariamente los datos relacionados con la asistencia de los trabajadores, así como las incidencias relativas al cumplimiento de la disciplina laboral. El área de Recursos Humanos tiene que entregar al área de Contabilidad en el tiempo acordado, el modelo SC-4-05 “prenómina”, por el cual el Ministerio de Finanzas y Precios mediante su resolución número 10 del 2007, relaciona por cada trabajador el tiempo correspondiente a ausencias, impuntualidades, vacaciones, licencias, subsidios, interrupciones, movilizaciones, citaciones y prestaciones sociales que inciden en el tiempo a devengar. Debe venir firmado en todas las hojas por quién lo confecciona, revisa y con la aprobación y firma del administrador de la entidad. La prenómina se acompañará cuando corresponda por los modelos SC-4-04 “notificación de vacaciones, deducciones, licencias y subsidios”, donde se informa al área de contabilidad y al trabajador la solicitud y aprobación de las vacaciones, el importe de los descuentos, así como los pagos a efectuar por concepto de licencias o subsidios, sirviendo de base para la confección de la nómina.(Contabilidad 2010)

La República de Cuba como parte de su proceso de actualización del modelo económico ha empezado una nueva etapa de informatización de la sociedad, por lo que los subsistemas organizacionales de contabilidad de las empresas se deciden por sistemas informáticos desarrollados para contemplar su trabajo diario. La Universidad de las Ciencias Informáticas (UCI) y en particular el Centro de Gobierno Electrónico (CEGEL) no están exentos de este proceso.

La gestión administrativa de los Recursos Humanos mencionada anteriormente, se ha visto afectada en el CEGEL por varias limitaciones, comprobándose en la entrevista inicial efectuada a las personas encargadas de realizar el proceso de prenómina en el CEGEL. Se puede mencionar que todas las tareas correspondientes a los Recursos Humanos del centro están parcialmente informatizadas, debido a que solo se utilizan las herramientas de la suite de productos de Microsoft Office. Por otra parte, el personal encargado de la realización del proceso de prenómina comete errores a causa del desconocimiento de aspectos técnicos del negocio como puede ser dar licencia sin sueldo a un trabajador teniendo días de vacaciones acumulados, afectándose la calidad con que se elabora la misma. En adición con esto, las actividades de confección, reporte de incidencias por las diferentes claves, entrega y archivo de las evidencias que respalden cada una de estas actividades y el registro y control por separado de cada incidencia se realiza de forma manual al culminar prácticamente el mes, provocando que producto a errores humanos se queden elementos no tenidos en cuenta por el

corto lapso de tiempo para la conciliación de la nómina, trayendo como consecuencia que en casi el 45 % de las ocasiones se tuviera que realizar actualizaciones de la nómina de pago, incluso cuando varias de estas ya habían sido firmadas por la decana o con el visto bueno de la vicedecana administrativa, lo cual pudiera dejar en entredicho un correcto trabajo del CEGEL en este proceso, se le suma a esto que las actualizaciones antes mencionadas consumen tiempo y que en casi el 60% de las ocasiones se entrega a los especialistas de capital humano o en el tope de fecha o pasadas estas.

Cuando la gestión administrativa se lleva a cabo de la manera en que hasta la actualidad se hace, no se observa el proceso de manera integral sino que está fragmentada la información necesaria que respalda y justifica cada elemento en la nómina de pago y no se obtienen ni mantienen actualizados los registros correspondientes a la misma, tales como: registro de licencias sin sueldo en sus diferentes clasificaciones, registro de licencias de maternidad, registro de prestación social, registro de certificados médicos con sus respectivas clasificaciones (hoy no se conoce que enfermedades de los trabajadores son producto a la actividad laboral, lo que impide hacer un buen plan de salud del trabajo en el área), registro de medidas disciplinarias, registro de consultas médicas de embarazadas, registro y control de las hojas de firmas de los especialistas y registro y control del tiempo laborado fuera de la entidad, lo cual incide directamente como contraparte del consumo de dietas. Todo lo anteriormente planteado ha sido constatado en las visitas integrales del consejo universitario ampliado al área, que es importante señalar que aunque los resultados han sido satisfactorios el camino transitado como proceso no ha sido el deseado.

La situación problemática antes descrita ha generado el siguiente **problema de investigación**: La elaboración, el control de evidencias y el registro ineficientes de la información asociada a la nómina de pago de los trabajadores afectan la gestión administrativa de los Recursos Humanos del CEGEL.

Se define por lo tanto como **objeto de estudio**: Procesos de gestión administrativa para los Recursos Humanos.

Para ello se identifica como **campo de acción**: Informatización de los procesos asociados a la gestión administrativa de los Recursos Humanos del CEGEL.

Para dar solución al problema, se propone como **objetivo general**: Desarrollar una aplicación web para la gestión administrativa de los Recursos Humanos del CEGEL que contribuya a mejorar como proceso continuo la elaboración, el control de evidencias y el registro de información asociados a la nómina de los trabajadores.

Al logro del objetivo general tributan los siguientes **objetivos específicos**:

1. Definir el marco teórico de la investigación mediante el estudio y el análisis de los principales referentes teóricos para el desarrollo de la solución.
2. Realizar el diseño e implementación de la solución para obtener los componentes de software de los procesos asociados a la gestión administrativa de los Recursos Humanos en el CEGEL.
3. Valorar el nivel de aceptación y efectividad de la solución propuesta mediante la técnica de ladov y la realización de pruebas de caja negra y caja blanca.

Planteándose como **posible resultado**: Obtención de una aplicación web para la gestión administrativa de los Recursos Humanos del CEGEL.

Siendo la siguiente **idea a defender**: El desarrollo de una aplicación web para la gestión administrativa de los Recursos Humanos del CEGEL, contribuirá a mejorar como proceso continuo la elaboración, el control de evidencias y el registro de información asociados a la pre Nómina de pago de los trabajadores.

Se define la siguiente **estructura del documento**, quedando conformado en tres capítulos:

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Se describe el estudio de la teoría sobre la investigación y el marco de los conocimientos acerca de las herramientas a trabajar. Se realiza una síntesis en relación al lenguaje de programación, el gestor de base de datos, la metodología a utilizar así como la arquitectura de la aplicación. Se analizan las investigaciones relacionadas con la elaboración, el registro y control de la información asociada al proceso de prenomina y se realiza un estudio del estado el arte. Se conceptualizan elementos importantes para la captura y la validación de requisitos; así como las distintas técnicas para validar la solución propuesta, las herramientas y tecnologías a utilizar para el diseño e implementación de la aplicación.

CAPÍTULO 2: DESCRIPCIÓN DE LA PROPUESTA

En este capítulo se describe la propuesta de solución de la aplicación a desarrollar y el proceso que será informatizado. Se presentarán los requisitos funcionales y no funcionales con los que debe cumplir la aplicación propuesta. Se hará referencia a las técnicas empleadas tanto en la captura como en la validación de los requisitos. Se mostrará el diagrama de actividades del negocio, el modelo conceptual y el diagrama de clases persistentes. Se explicará la utilización de un conjunto de patrones dentro del diseño de la aplicación. En consecuencia, se obtendrá un conjunto de artefactos establecidos por la metodología de desarrollo propuesta en el capítulo 1.

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

En este capítulo se presenta la validación de los requisitos para fundamentar el funcionamiento correcto de la aplicación. Se validará el diseño de la solución a través de pruebas unitarias y de aceptación con los métodos de caja blanca utilizando la técnica del camino básico, caja negra por parte del departamento de calidad del CEGEL y con los casos de pruebas mostrados a los clientes, además se utilizará la técnica de ladov para medir el grado de satisfacción de los clientes con respecto a la aplicación. Después del análisis se expondrán de manera gráfica y cuantitativa los resultados arrojados.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

El presente capítulo se enfoca en la realización de un análisis de las fuentes bibliográficas para obtener información actualizada acerca de la teoría necesaria para la investigación. Se describe el marco teórico de las principales aplicaciones existentes a escala organizacional, nacional e internacional, así como la teoría referente a la Ingeniería de Software (ISW) y sus disciplinas. Por último, se realizará un análisis de las tecnologías, lenguajes de programación y herramientas a utilizar.

1.2. Principales conceptos abordados en la investigación

En este epígrafe se relacionan los conceptos que se utilizarán a lo largo de la investigación para la total comprensión de los términos propios del proceso de negocio de pre Nómina. Se estudiaron, la ley 116 del 2013 del código del trabajo en Cuba, el decreto ley 326 reglamentos del código del trabajo, las bases generales para el pago adicional de los trabajadores de la UCI resolución 116 2009 del Ministerio de Trabajo y Seguridad Social y los conceptos que brinda la Organización Internacional del Trabajo.

Prenómina: Modelo que relaciona por cada trabajador el tiempo correspondiente a ausencias, impuntualidades, vacaciones, licencias, subsidios, interrupciones, movilizaciones, citaciones y prestaciones sociales que inciden en el tiempo a devengar por los trabajadores.

Capital Humano: Se refiere a toda la riqueza individual de la persona, a sus conocimientos y capacidades, a sus actitudes, valores, motivaciones e intereses, a su rol en actividad social de la organización. No debe simplificarse, trabajando y midiendo el impacto financiero de los “conocimientos, habilidades y destrezas” que se consideran competencias claves o críticas que la Organización necesita para alcanzar los resultados propuestos.

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

Conciliación con Capital Humano: Es la retroalimentación que se establece cada mes entre los departamentos de Capital Humano y Recursos Humanos para la discusión de la pre Nómina.

Incidencia: Es una variación que se le realizará posteriormente a la nómina de un trabajador. Se compone de dos factores: Clave y Descripción.

Clave: Es el número con el cual se identifica inequívocamente una incidencia.

Descripción: Consiste propiamente en el nombre que se le asigna a una incidencia.

Evidencia: Documentación asociada a una incidencia en particular.

Sistema de pago: Indicadores que afectan la nómina de los trabajadores. Consta de dos factores: Requisito y Evaluación.

Requisitos: Elementos que se describen en las bases generales del pago adicional especial aprobado para la UCI por el Ministerio de Trabajo y Seguridad Social para acceder al sistema de pago.

Evaluación: Muestra el desempeño laboral de un trabajador. Se puede clasificar en Superior, Adecuado, Deficiente y No Evaluado.

Trabajo extraordinario: Se interpreta como aumento salarial por horas de servicio fuera de la jornada laboral convenida. Se expresa en horas para los trabajadores que tienen cargo Chofer con un máximo en el mes de 48 horas.

Doble turno: Tiempo que el trabajador cumple doble como consecuencia de una inasistencia del trabajador que debía relevarlo.

Nocturnidad: Se interpreta como aumento salarial por tiempo de trabajo nocturno. Se puede clasificar según el tiempo en que se trabaja: hasta las 12:00am (nocturnidad 1 0.08) y (nocturnidad 2 0.16). Se expresa en horas para los trabajadores que tienen cargo Técnico en Ciencias Informáticas con un máximo en el mes de 32 horas para la nocturnidad 1 y de 64 horas para la nocturnidad 2.

Día feriado: Están programados legalmente desde diciembre del año anterior al actual. Se expresa en horas y su pago es al doble del valor.

1.3. Sistemas que informatizan el proceso de pre Nómina

A continuación se describen los principales sistemas informáticos que informatizan el proceso de pre Nómina, o parte de este, a escala organizacional, nacional e internacional y que son relevantes para el desarrollo de esta investigación. Se hace un compendio de las características de los sistemas

basadas en las particularidades del negocio como se muestra en la tabla resumen 1. Por último, los aportes vistos de los sistemas estudiados.

A continuación se describen los sistemas internacionales estudiados.

AdaptaPro, software producido en Venezuela, es una solución completa en el área administrativa ya que comprende todos los procesos de administración de personal y nómina en un mismo sistema, cuenta con una base de datos única. Posee herramientas para el usuario, el ejecutivo y el empleado.(Soluciones Corporativas AdaptaPro 2010)

Características:

- Registra toda la información referente al departamento de Recursos Humanos en una única aplicación.
- Genera reporte de nómina pero no lo exporta.
- No se adapta con los indicadores de la universidad.
- No controla las evidencias de las licencias ni de los certificados médicos.
- Registra las bajas, las licencias de maternidad y los certificados médicos pero no las licencias sin sueldo.

SIoux-RH, software producido en Colombia, es un sistema informático para la gestión de los Recursos Humanos y de nómina, dentro de sus módulos cuenta con control de personal, liquidación de nómina, liquidaciones definitivas, archivos para bancos, exportación contable con provisiones, Informes para EPS y ERP, Parafiscales, control documental y de dotación de empleados, impresión de contratos y carnets.(S.A.S. 2015)

Características:

- Registra toda la información referente al departamento de Recursos Humanos en una única aplicación.
- Genera reporte de nómina pero no lo exporta.
- No se adapta con los indicadores de la universidad.
- No controla las evidencias de las licencias ni de los certificados médicos.
- Registra las bajas, las licencias de maternidad y los certificados médicos pero no las licencias sin sueldo.

ATENNEA AIR, software producido en España, es una solución informática para la gestión de compañías aéreas que cubre las principales áreas operativas y administrativas. Está adaptada a las necesidades de compañías aéreas chárter, regulares, cargo y jet o aviación ejecutiva. ATENNEA AIR contempla la planificación, cotización de vuelos y contratación comercial, la gestión de las operaciones, asignación de tripulaciones e introducción de partes de vuelo, así como el análisis de la información post-vuelo, control de los DOCs (Direct Operational Costs) o información para pre Nómina. Así mismo cuenta con un módulo de presupuestos que se alimenta de la información de los otros módulos. (Sistemas de Gestión de Baleares 2011)

Características:

- Registra toda la información referente al departamento de Recursos Humanos en una única aplicación.
- Genera reporte de pre Nómina pero no lo exporta.
- No se adapta con los indicadores de la universidad.
- No controla las evidencias de las licencias ni de los certificados médicos.
- Registra las bajas, las licencias de maternidad y los certificados médicos pero no las licencias sin sueldo.

A continuación se describen los sistemas nacionales estudiados.

RH Expert es un sistema de gestión integral de los Recursos Humanos, desarrollado por Datazucar, que agrupa las tareas más importantes y comunes de esta actividad tales como: sistema de presencia, evaluación del desempeño laboral, control integral del personal, análisis y desarrollo del potencial directivo y el control de las necesidades de capacitación de la organización. Permite incrementar la eficacia, la productividad, las competencias, desarrollar la comunicación empresarial, la cohesión y la unidad del colectivo; contribuir a la mejora continua; profundizar el control y la evaluación del impacto en el rendimiento y elevar el desempeño laboral. Posibilita con la utilización y puesta en práctica del mismo, un cambio de mentalidad para lograr la optimización del capital humano. (Datazucar 2012)

Características:

- Registra toda la información referente al departamento de Recursos Humanos en una única aplicación.
- Genera reporte de pre Nómina en formato pdf.
- No se adapta con los indicadores de la universidad.

- No controla las evidencias de las licencias ni de los certificados médicos.
- Registra las bajas, las licencias y los certificados médicos.

Fastos es el nombre que se le dio a una aplicación desarrollada por la Empresa Nacional del Software (DESOFT) que se encarga de gestionar los procesos del departamento de Recursos Humanos, entre ellos el proceso de nómina. Es de tipo de software privativo con arquitectura modular, se enlaza automáticamente con el sistema “Pagus”, destinado al procesamiento de las nóminas, desarrollado también por DESOFT.(Desoft 2015)

Características:

- Registra toda la información referente al departamento de Recursos Humanos en una única aplicación.
- Genera reporte de nómina pero no lo exporta.
- No se adapta con los indicadores de la universidad.
- No controla las evidencias de las licencias ni de los certificados médicos.
- Registra las bajas, las licencias y los certificados médicos.

Componente Nómina es el nombre que se le dio al componente que integra parte del proceso de nómina con el proceso de nómina. Se encuentra en fase de implementación dentro del módulo Capital Humano por parte del sistema **CEDRUX** que pertenece al Centro de Informatización de Entidades (CEIGE) en la UCI. Es notable mencionar que CEDRUX no contempla un sistema de nómina, sino que contempla un sistema de capital humano que incluye entre sus funciones la organización del trabajo (gestión de puestos de trabajo), la selección e integración (registro de personas y formalización de la relación laboral), administración de capital humano (todo lo referente a nómina), estimulación moral y material (nóminas) y cierre. El presente trabajo de diploma se centra únicamente en los procesos de gestión de nómina, por lo que sería muy engorroso implantar este componente y además configurar todos los módulos para su correcto funcionamiento, pues se utilizaría sólo una pequeña parte de él. En adición con lo anterior planteado se puede decir que CEDRUX como sistema genérico no contempla incluir importaciones de información ni personalizaciones.

A continuación se muestra la tabla 1 donde se puede apreciar un resumen de las características distintivas para el proceso de nómina en relación con los sistemas antes estudiados.

Tabla 1: Presencia de las funcionalidades asociadas al negocio en los diferentes sistemas estudiados.

Sistemas	AdaptaPro	SIOUX-RH	ATENNEA AIR	RH Expert	Fastos	CEDRUX
Salida de la información en formato .pdf	no	no	no	sí	no	sí
Generación de reportes de prenomina	sí	sí	sí	sí	sí	sí
Adaptación con sistema de pago de la UCI	no	no	no	no	no	no
Control de evidencias de licencias sin sueldo	no	no	no	no	no	no
Control de evidencias de certificados médicos	no	no	no	no	no	no
Control de licencias de maternidad	no	no	no	no	no	no
Registros de bajas	sí	sí	sí	sí	sí	sí
Registros de certificados médicos	sí	sí	sí	sí	sí	sí
Registros de licencias sin	no	no	no	si	si	sí

sueldo						
Registros de licencia de maternidad	sí	sí	sí	sí	sí	sí

Luego de revisar los principales sistemas que informatizan el proceso de pre Nómina, o parte de este, se ha determinado implementar una nueva aplicación pues una de las mayores desventajas es que no son adaptables a los indicadores de la universidad, además la mayoría no exportan la información en formato .pdf y no controlan las evidencias de las incidencias. Para implementar el nuevo sistema se han decidido utilizar características similares a los ya vistos como son la generación de reportes en formato pdf, el control de la veracidad de las incidencias mediante evidencias, el registro de la información asociada a la pre Nómina y la implementación de una aplicación web donde se encuentre centralizada toda la información.

1.4. Ingeniería de Software

La Ingeniería de Software es definida por la IEEE como “la aplicación de un enfoque sistemático, disciplinado y cuantificable para el desarrollo, operación y mantenimiento del software; es decir, la aplicación de ingeniería al software”. Esta disciplina es un área compleja que ofrece un amplio conjunto de herramientas, artefactos, métodos y técnicas con el fin de desarrollar y mantener productos de software de calidad. Sin embargo, seleccionar cada uno de estos elementos requiere un análisis de acuerdo a las características del producto final y principalmente al entorno de desarrollo del software. El presente epígrafe realiza un análisis de las principales herramientas y metodologías existentes seleccionando las más adecuadas para soportar la solución de la presente investigación.

Las metodologías para el desarrollo de software consisten en un conjunto de procedimientos, técnicas, herramientas y soporte documental que ayuda a los desarrolladores a realizar un nuevo software. Una metodología durante su ciclo de vida indica qué es lo que hay que obtener durante todo el proceso de desarrollo, es decir, cómo se obtienen los productos parciales y finales pero no cómo hacerlos. (García 2009-2010)

La selección de la metodología de desarrollo es una de las tareas más importantes para cualquier equipo de desarrollo de software. La misma justifica el uso de artefactos, técnicas, herramientas, procedimientos y un soporte documental correspondientes al ciclo de vida de un producto de software. Las metodologías de desarrollo de software se han clasificado en dos grandes grupos: las ágiles y las tradicionales o pesadas.

Las metodologías pesadas son aquellas que están guiadas por una fuerte planificación durante todo el proceso de desarrollo con numerosas políticas y normas. Comprenden la realización de una intensa etapa de análisis y diseño antes de la construcción del sistema. Generalmente son aplicadas en grandes equipos de desarrollo donde existe además un elevado número de roles. Este tipo de metodologías se basan principalmente en normas provenientes de estándares seguidos por el entorno de desarrollo y poseen una cierta resistencia a los cambios.(Amaro Calderón 2007)

Como una alternativa para las metodologías pesadas surgen en la década de los 90's las metodologías ágiles(Rivadeneira Molina 2013). Al decir de (Letelier 2004) estas metodologías se aplican a un amplio rango de proyectos; principalmente en aquellos donde el equipo de desarrollo es pequeño, existen pocos roles, se generan pocos artefactos, los plazos de entrega son reducidos, existen requisitos volátiles y se basan en las nuevas tecnologías. Este análisis permite decidir utilizar metodologías ágiles para guiar la presente investigación.

Entre las principales metodologías ágiles usadas con mayor frecuencia en el proceso de desarrollo de software se encuentran: eXtreme Programming (XP), Scrum, Microsoft Solutions Framework (MSF), Lean Development entre otras como refiere (Amaro Calderón 2007). En el caso particular de la investigación se decide utilizar XP por ser uno de los máximos exponentes de las metodologías ágiles, además por sus características, flexibilidades y por el amplio material bibliográfico existente en la web relacionado con la forma de utilizarla.

XP es creada por Kent Beck cerca de 1999 siguiendo cuatro valores primordiales: comunicación, simplicidad, coraje y retroalimentación (Beck 1999). Se fundamenta por ser una metodología centrada en fortalecer las relaciones interpersonales como clave para el éxito en el desarrollo de software. Para lograrlo promueve el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. XP se basa además en retroalimentación continua entre el cliente y el equipo de desarrollo, la comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y el coraje para enfrentar los cambios. XP se recomienda especialmente adecuada para proyectos con requisitos imprecisos o muy cambiantes y donde existe un alto riesgo técnico(Canós 2004). Algunos de los artefactos fundamentales que genera XP son las historias de usuario, las tarjetas CRC, el código, las pruebas unitarias, de integración y de aceptación.

Las **historias de usuarios** es una de las técnicas para especificar los requisitos de software, son tarjetas donde se describen de forma muy puntual las características que el sistema debe poseer. Las historias de usuario pueden ser descompuestas en tareas para ser asignadas a varios programadores a la vez. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los

programadores puedan implementarla en unas pocas semanas (Jeffries 2001). Representan una breve descripción del comportamiento del sistema. Su tratamiento resulta muy dinámico y flexible, en cualquier momento pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas (Letelier 2006). Homológamente a las historias de usuario, las **tarjetas CRC** (de clase, responsabilidad y colaboradores) son técnicas para realizar la especificación de cada clase del sistema.

Los **requisitos funcionales** se definen según (Sommerville 2005) como las declaraciones de los servicios que debe ofrecer el sistema, su reacción ante entradas particulares y el comportamiento en situaciones específicas. En algunos casos los requisitos funcionales pueden declarar explícitamente lo que el sistema debe hacer. Por su parte los requisitos no funcionales son aquellos que más allá de las funciones específicas del sistema se refieren a las propiedades que emergen de este como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. En otras palabras, son aquellos requerimientos que definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y las posibles representaciones de los datos en las interfaces del sistema (Sommerville 2005). Además estos requisitos se clasifican según sus características técnicas en: requisitos de seguridad, de integridad, de disponibilidad, de usabilidad, de rendimiento, de portabilidad, de confiabilidad, requisitos de software y de hardware.

XP pone la comprobación como el fundamento del desarrollo, con cada programador escribiendo pruebas cuando crean su código de producción (Fowler 2003). Las pruebas unitarias se enfocan solamente en comprobar pequeñas cantidades de código, solamente prueban el código del requerimiento específico. Este tipo de pruebas se aísla de otro código y de otros programadores evitando interferir con el trabajo realizado por otros (Rojas 2012). Las pruebas de integración, buscan asegurar el correcto funcionamiento del sistema o subsistema así como encontrar problemas que pueden surgir debido a la unificación o integración de los componentes. Finalmente según (Sommerville 2005) las pruebas de aceptación se refieren a la revisión junto al cliente de las funcionalidades implementada buscando la “aceptación” de la solución presentada.

La forma de comunicación entre los programadores en XP es fundamentalmente a través del código, por ello se propone regirse por ciertos estándares de programación así como buenas prácticas, normas y estilos (Canós 2004).

Las pruebas presentan una interesante anomalía para el ingeniero del software. Durante las fases anteriores de definición y de desarrollo, el ingeniero intenta construir el software partiendo de un concepto abstracto y llegando a una implementación tangible. A continuación, llegan las pruebas. El

ingeniero crea una serie de casos de prueba que intentan demoler el software construido. La prueba demuestra hasta qué punto las funciones del software parecen funcionar de acuerdo con las especificaciones y parecen alcanzarse los requisitos de rendimiento. Además, los datos que se van recogiendo a medida que se llevan a cabo las pruebas proporcionan una buena indicación de la fiabilidad del software y, de alguna manera, indican la calidad del software como un todo (Pressman 2005).

Las pruebas de caja blanca (también conocidas como pruebas de caja de cristal o pruebas estructurales) se centran en los detalles procedimentales del software, por lo que su diseño está fuertemente ligado al código fuente. Aunque las pruebas de caja blanca son aplicables a varios niveles —unidad, integración y sistema—, habitualmente se aplican a las unidades de software. Su objetivo es comprobar los flujos de ejecución dentro de cada unidad (función, clase, módulo, etc.) pero también pueden probar los flujos entre unidades durante la integración, e incluso entre subsistemas, durante las pruebas de sistema.

Las pruebas de caja blanca se llevan a cabo en primer lugar, sobre un módulo concreto, para luego realizar las de caja negra sobre varios subsistemas (integración). Mediante los métodos de prueba de caja blanca, se pueden obtener casos de prueba que:

- Garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada módulo. Ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa.
- Ejecuten todos los ciclos en sus límites y con sus límites operacionales.
- Ejerciten las estructuras internas de datos para asegurar su validez.

Las principales técnicas de diseño de pruebas de caja blanca son:

- Pruebas de la estructura de control.
- Pruebas de caminos básicos.(Pressman 2005)

Las pruebas de caja negra, también denominadas pruebas de comportamiento, se centran en los requisitos funcionales del software. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Las pruebas de caja negra no son una alternativa a las técnicas de pruebas de caja blanca, sino que se trata de un enfoque complementario que intenta descubrir diferentes tipos de errores que los métodos de caja blanca. Las pruebas de caja negra intentan encontrar errores de las siguientes categorías: funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a bases de datos externas, errores de rendimiento y errores de inicialización y de terminación.

Para desarrollar la prueba de caja negra existen varias técnicas, entre ellas están:

- Técnica de la Partición de Equivalencia: Esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
- Técnica del Análisis de Valores Límites: Esta Técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
- Técnica de Grafos de Causa-Efecto: Es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.(Pressman 2005)

1.5. Lenguaje de Programación

Un lenguaje de programación es un idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana (Mark 2010). Son numerosos los lenguajes de programación existentes en la actualidad que se dirigen principalmente a dos tipos de software: aplicaciones web o aplicaciones de escritorio.

PHP (acrónimo de "PHP: Hypertext Pre-processor") es un lenguaje de programación web de "código abierto" interpretado, de alto nivel, embebido en páginas HTML y ejecutado en el servidor (Achour 2006). Este lenguaje de programación nace en la década de los noventa del pasado siglo y actualmente es uno de los lenguajes de programación para el desarrollo web más documentados. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno.

A grandes rasgos la lógica de PHP es la siguiente: cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Éste, procesa el script solicitado que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente. PHP permite además la conexión a diferentes servidores de bases de datos tales como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite (Rincón Carrera 2011).

Se decide utilizar PHP como lenguaje de programación web para implementar la aplicación pues es un lenguaje de código abierto interpretado, de alto nivel, embebido en páginas HTML y ejecutado en el servidor; lo que supone su uso de forma libre sin necesidad de pagar una licencia, además de poder implementar las clases de manera más entendible para el desarrollador, sin tener que consumir

muchos recursos de la computadora y que además las instrucciones son ejecutadas directamente sin una previa compilación del programa contribuyendo a que los errores fueran detectados y corregidos de manera más fácil. También se le suma que es uno de los lenguajes de programación más documentados para el desarrollo web y es suficiente para la implementación de la aplicación; así como que puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno (Hughes, Saadi et al. 2009).

JavaScript es un lenguaje de programación que surgió por la necesidad de ampliar las posibilidades del HTML y permite la creación y manipulación de objetos sencillos, y la definición de métodos y propiedades para dichos objetos. JavaScript es un lenguaje interpretado que, al contrario de las aplicaciones normales, que son ejecutadas por el sistema operativo, es ejecutado por el navegador que se utiliza para ver las páginas. Eso hace que se puedan desarrollar aplicaciones de diversos tipos, desde generadores de HTML, comprobadores de formularios, hasta programas que gestionen las capas de una página. Pueden desarrollarse incluso aplicaciones que permitan poder tener capas en una página como si fueran ventanas, y dar la sensación de estar trabajando con una aplicación con interfaz de ventanas.

Entre las acciones típicas que se pueden realizar en JavaScript existen dos vertientes. Por un lado los efectos especiales sobre páginas web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. Por el otro, JavaScript permite ejecutar instrucciones como respuesta a las acciones del usuario (eventos), con lo que se pueden crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo (Pilgrim 2010).

Se decide utilizar JavaScript como lenguaje de programación para las validaciones necesarias en la aplicación pues brinda la posibilidad de validar los datos del lado del cliente sin necesidad de enviar la petición para que sea validada en el servidor, además su sintaxis es simple y admite que una variable pueda tomar valor de un número y más tarde de una cadena, esto se comprueba con `typeof`. También se le suma que es uno de los lenguajes de programación más documentados, es conocido por el desarrollador y que se solventa para las validaciones tanto del lado del cliente o del servidor.

1.6. Herramientas CASE

Las herramientas CASE (Ingeniería de Software Asistida por Computadora) son aplicaciones informáticas dedicadas al aumento de la productividad en el desarrollo de software. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costos, implementación de

parte del código automáticamente con el diseño dado, compilación automática y documentación o detección de errores. (Loucopulos 1995)

Visual Paradigm (VP) 8.0 es una herramienta de modelado profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño, construcción, pruebas y despliegue. El software de modelado UML (Lenguaje de Modelado Unificado) ayuda a una construcción de aplicaciones de calidad, mejor, más rápida y a un menor coste. Permite modelar todos los tipos de diagramas de clases así como generar código inverso, código desde diagramas y documentación. Es fácil de instalar y actualizar además de ser compatible entre ediciones.

Dentro de las principales características de la herramienta se encuentran: disponibilidad de múltiples versiones, de acuerdo a su necesidad, así como la capacidad de integrarse a los principales IDEs, ofrece un diseño centrado en casos de uso y enfocado al negocio de manera que genera un software de mayor calidad. Además la herramienta es colaborativa, es decir, soporta múltiples usuarios trabajando sobre el mismo proyecto. Es capaz de generar la documentación del proyecto en diferentes formatos como web o pdf y permite el control de versiones (paradigm 2012). Soporta el ciclo de vida completo en el desarrollo de software: análisis y desarrollo orientado a objetos, construcción, prueba y despliegue. Permite diseñar todo tipo de diagrama de clases, realizar la ingeniería inversa, generar códigos a partir de diagramas y crear documentación. VP es una herramienta libre y multiplataforma (Rosales Morales 2013).

Se decide utilizar como herramienta CASE el Visual Paradigm para la realización del diagrama de actividades y del diagrama de clases persistentes, debido a sus ventajas y características, además de una alta capacidad de integración con lenguajes de programación. Es una herramienta disponible en varios idiomas y es fácil de instalar y actualizar. Visual Paradigm facilita la organización de los diagramas generando la documentación del proyecto en varios formatos. Además de ser capaz de generar código, modelo de datos, entre otras funcionalidades útiles para el desarrollo de la aplicación.

1.7. Marco de Trabajo

El término “Marco de Trabajo” se refiere a una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. Se puede considerar como una aplicación genérica incompleta y configurable a la que puede añadirse las últimas piezas para construir una aplicación concreta. (Carrasco Puebla 2009)

Symfony 2 es un marco de trabajo rápido, flexible y fácil de aprender que le permite a los desarrolladores construir aplicaciones webs más mantenibles. Symfony 2 ha sido desarrollado

teniendo en cuenta el rendimiento como mayor prioridad. Hasta 3 veces más rápido que Symfony 1.4 o Zend Framework 1.10 y consume la mitad de la memoria virtual. Se construye a base de bundles (conocidos como plugins en Symfony 1). Un bundle es un conjunto estructurado de archivos que implementan una característica única y que puede ser fácilmente compartido con otros desarrolladores. Los bundles permiten que los desarrolladores configuren y personalicen el sistema de una forma limpia. (Eguiluz 2011)

Symfony2 es un marco de trabajo de código abierto, está construido utilizando un contenedor de inyección de dependencias, inspirado en el marco de trabajo Spring de Java. En un proyecto, el desarrollador no interactúa directamente con el contenedor. Todos los detalles de implementación están ocultos detrás de un buen sistema de configuración que permite personalizar todo a través de archivos .yaml o .xml, o incluso a través de código PHP.(Eguiluz 2011)

Para la realización de la aplicación se utiliza Symfony 2. Posee la peculiaridad de tener su arquitectura interna totalmente desacoplada permite que se reemplacen o se eliminen fácilmente aquellas partes que no encajen con el proyecto a desarrollar, lo cual le permite adaptarse con facilidad a los nuevos cambios que puedan surgir durante el proceso de desarrollo del software. También existen módulos que se han desarrollado para ser integrados a Symfony 2, para realizar operaciones tales como: generación de reportes en formato .pdf, manipulación de seguridad y usuarios y gestión de envíos de correos electrónicos entre otros. Symfony 2 posee características que lo hacen único con respecto a otros marcos de trabajo, como el soporte HTTP, el contenedor de inyección de dependencias, el sistema de plantillas Twig y el sistema de bundles. También brinda la facilidad de utilizar el ORM Doctrine 2.0 con el que se logra la independencia del gestor de base de datos. Además Symfony 2 hace uso de los lenguajes HTML 5 y de CCS 3 que se incluyen dentro de las twigs las cuales son las definidas por este marco de trabajo para ofrecerle al usuario un entorno vistoso y una mejor interacción con la aplicación desarrollada.

1.8. Servidor Web

Un servidor de aplicaciones es una tecnología que proporciona la infraestructura y servicios claves a las aplicaciones alojadas en un sistema. Entre los servicios habituales de un servidor de aplicaciones se incluyen los siguientes: agrupación de recursos (por ejemplo, agrupación de conexiones de base de datos y agrupación de objetos), administración de transacciones distribuida, servicios de detección de errores y estado de las aplicaciones, así como seguridad integrada.

Para el desarrollo de la aplicación web se utilizará HTTP Apache porque es un servidor web HTTP el cual se encuentra bajo la licencia de código abierto y además es multiplataforma. Apache es un

servidor potente, flexible y altamente configurable esto permitió que el acceso a la aplicación fuera estable y que hacer cambios en los parámetros de la configuración se realizaran de una manera sencilla. Es un software que está en constante desarrollo y alienta los comentarios de los usuarios a través de nuevas ideas, informes de errores y parches posibilitando que la búsqueda de información asociada a esta tecnología para resolver un problema que se presente, fuera abundante y fácil de encontrar una solución. (Foundation 2011)

1.9. Sistema Gestor de Bases de Datos

Un sistema gestor de base de datos (SGBD) es básicamente un sistema computarizado cuya finalidad general es almacenar información y permitir a los usuarios recuperar y actualizar esa información a través de peticiones o consultas. La información puede ser cualquier dato que resulte importante para el individuo u organización; en otras palabras, todo lo necesario para auxiliar el proceso general de su administración (Date 2001). En estos sistemas se proporciona un conjunto coordinado de programas, procedimientos y lenguajes que permiten a los distintos usuarios realizar sus tareas habituales con los datos, garantizando además la seguridad de los mismos (Sánchez Asenjo 2013). Algunos de los SGBD libres son SQLite, DB2 Express-C, Apache Derby, MariaDB, MySQL y PostgreSQL. Para el manejo de los datos del sistema propuesto en la investigación se utilizará como SGBD PostgreSQL.

Se utilizará PostgreSQL como gestor de bases de datos porque es uno de los gestores de código abierto más avanzado hoy en día, ofreciendo control de concurrencia multi-versión, soportando casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, y tipos y funciones definidas por el usuario) válido en la aplicación para la gestión de la información, contando también con un amplio conjunto de enlaces con lenguajes de programación. La necesidad de almacenar volúmenes de información referente al proceso requiere el uso de gestores de bases de datos que permitan gestionar de manera ágil la información necesaria, útil cuando se trata de la gestión de volúmenes de datos. Además es usado en las soluciones informáticas que se desarrollan en la UCI para las instituciones del estado, lo que facilita la transferencia y la mantenibilidad por parte de los clientes y existe una documentación abundante a la cual se puede acceder en caso de querer solucionar un problema que se presente.

Cuenta, dentro de sus principales características con un rico conjunto de tipos de datos, permitiendo además su extensión mediante tipos y operadores definidos y programados por el usuario. Su administración se basa en usuarios y privilegios y es altamente confiable en cuanto a estabilidad se refiere. Sus ventajas se resumen en que como gestor de bases de datos es una herramienta código

abierto, multiplataforma y sin restricciones de uso, requisitos solicitados por el cliente para el desarrollo de la aplicación. Además la universidad cuenta con la información necesaria gracias a la existencia de una comunidad de PostgreSQL, que realiza actividades conjuntamente con la comunidad nacional e internacional del gestor de base de datos. (Group 1996-2013)

1.10. Entorno de Desarrollo Integrado

Un Entorno de Desarrollo Integrado (en inglés Integrated Development Environment o IDE) es un programa compuesto por una serie de herramientas que utilizan los programadores para desarrollar código. Esta herramienta puede estar pensada para su utilización con un único lenguaje de programación o bien puede dar cabida a varios de estos (Corporation 2013). Un IDE brinda todas las herramientas de programación dentro de un único espacio. Antiguamente los programadores tenían que editar los ficheros, salvarlos, correr el compilador, construir la aplicación y ejecutarla a través de un depurador. Actualmente un IDE integra en una única aplicación: editor, compilador, el depurador y herramientas para la administración de los proyectos. Todas estas posibilidades favorecen el aumento de la productividad del programador (Bolton 2014). Por sus ventajas y facilidades, se decide utilizar el IDE PhpStorm para soportar la implementación de la solución.

PhpStorm es un IDE ligero e inteligente enfocado en la productividad del desarrollador, provee completamiento inteligente de código, navegación rápida y chequeo de errores al momento. Siempre está listo para dar forma al código, ejecutar unit o proveer debugging visual (IDEA 2012).

Entre los principales beneficios de su uso se puede mencionar que es un editor php inteligente con completamiento de código php, detector de código duplicado y mezcla lenguajes (JavaScript, SQL, XML), es un editor de JavaScript avanzado, basado en DOM, navegación de código y búsqueda de usos y Debugger de JavaScript, es un editor HTML/CSS que soporta HTML5, validación y arreglo-rápido y muestra estilos aplicados, es un IDE ligero de instalación fácil, es multiplataforma y de código abierto, tiene un ambiente inteligente debido a unit php visual, los históricos de cambios locales y php UML, además del debugging visual mediante puntos de ruptura en php, js, html, la inspección de variables y el análisis de código Batch.

1.11. Variables de la investigación

En este epígrafe se mostrarán las variables de la investigación que se clasifican en dependientes o independientes. El término variable se refiere a que está sujeto a cambios frecuentes o probables de un determinado valor. Una variable dependiente es aquella cuyos valores dependen de los que tomen otra variable y una variable independiente es aquella cuyo valor no depende del de otra variable. Se

define como variable dependiente: gestión administrativa de los Recursos Humanos y como variables independientes elaboración, registro de información y control de evidencias.

Para cuantificar la variable dependiente y las independientes se utilizará la magnitud tiempo para medir, como se hace en la actualidad (antes) y con la aplicación (después). Con el objetivo de realizar una correcta medición se estableció que el tiempo de respuesta brindado por el sistema será menor de 5 segundos. Para ello se medirá el tiempo de salida de cada una de las variables en un ambiente de trabajo real.

Técnica de ladov

La técnica ladov se realiza para comprobar los niveles de satisfacción de los clientes con el producto final, está conformada por cinco preguntas, tres abiertas y dos cerradas. Las preguntas cerradas se relacionan a través de lo que se denomina el “Cuadro Lógico de ladov”, por otra parte las preguntas abiertas permiten profundizar en la naturaleza de las causas que originan los diferentes niveles de satisfacción. El número resultante de la interrelación de las tres preguntas indica la posición de cada sujeto en la escala de satisfacción, o sea su satisfacción individual. La escala de satisfacción utilizada es la siguiente:

1. Clara satisfacción
2. Más satisfecho que insatisfecho
3. No definida
4. Más insatisfecho que satisfecho
5. Clara insatisfacción
6. Contradictoria

Esta técnica también permite obtener el índice de satisfacción grupal (ISG), para lo cual se trabaja con los diferentes niveles de satisfacción que se expresan en la escala numérica que oscila entre +1 y - 1 como se muestra en la figura 1.

Escala	Resultado
+1	Máximo de satisfacción
0,5	Más satisfecho que insatisfecho
0	No definido y contradictorio
- 0,5	Más insatisfecho que satisfecho
-1	Máxima insatisfacción

Figura 1: Tabla que muestra la escala numérica para trabajar con el ISG.

La satisfacción grupal se calcula por la siguiente fórmula que muestra la figura 2:

$$ISG = \frac{A (+1) + B (+0,5) + C (0) + D (-0,5) + E (-1)}{N}$$

Figura 2: Fórmula para calcular el ISG

En esta fórmula A, B, C, D, E, representan el número de sujetos con índice individual 1; 2; 3 o 6; 4; 5 y donde N representa el número total de sujetos del grupo. El índice grupal arroja valores entre + 1 y - 1. Los valores que se encuentran comprendidos entre - 1 y - 0,5 indican insatisfacción; los comprendidos entre - 0,49 y + 0,49 evidencian contradicción y los que caen entre 0,5 y 1 indican que existe satisfacción, como se muestra en la figura 3.

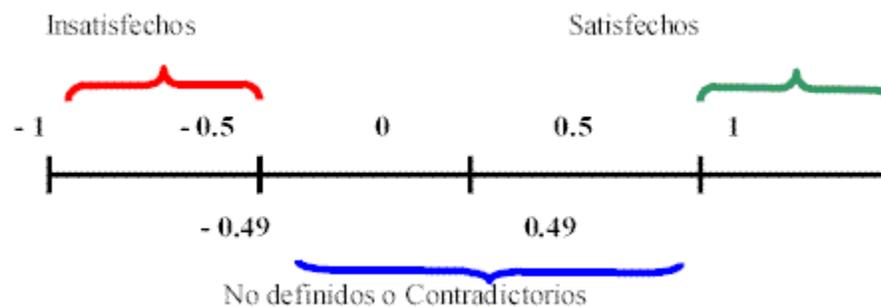


Figura 3: Rayo numérico para ubicar los valores de ISG.

1.12. Conclusiones parciales

Se realizó un análisis de los sistemas informáticos que informatizan el proceso de prenomina a nivel organizacional, nacional e internacional; obteniendo como resultado la propuesta de creación de una aplicación web. Se analizaron metodologías, herramientas, tecnologías y lenguajes, utilizando para

este trabajo XP como metodología, como herramienta CASE Visual Paradigm 8.0 y como lenguaje de modelado UML. Como servidor de bases de datos PostgreSQL en su versión 9.3, como marco de trabajo Symfony 2 y como IDE de desarrollo PhpStorm. Por último, se realiza la validación de las variables de la investigación, definiendo la magnitud por la cual serán medidas y el análisis de la técnica Iadov para medir el grado de satisfacción de los clientes.

CAPÍTULO 2. DESCRIPCIÓN DE LA PROPUESTA

2.1. Introducción

En este capítulo se presentan las características principales de la aplicación web para la gestión administrativa de los Recursos Humanos del CEGEL, definidas a partir del estudio de las tendencias actuales elaborado en el Capítulo 1: Fundamentación Teórica. Se muestran también los requisitos funcionales y no funcionales con los que deberá cumplir la aplicación web, así como algunas particularidades del diseño e implementación de dicha aplicación web.

2.2. Relación entre los principales conceptos del negocio

La prenomina se comienza a elaborar después de una previa conciliación con CH. Para dicha elaboración es necesario tener en cuenta aspectos como días feriados, sistema de pago y las claves de las incidencias. Algunas de las incidencias (ej. Enfermedad de más de tres días) necesitan las evidencias que ratifiquen la veracidad de estas incidencias. Las evidencias tiene modelos definidos legalmente y el personal de Recursos Humanos es el encargado de guardar toda la información. Al finalizar el proceso es enviado a CH el reporte de prenomina para su revisión y detección de elementos erróneos o no tenidos en cuenta. A continuación se muestra la figura 4 donde se muestra para una mayor comprensión un modelo conceptual con los conceptos antes descritos.

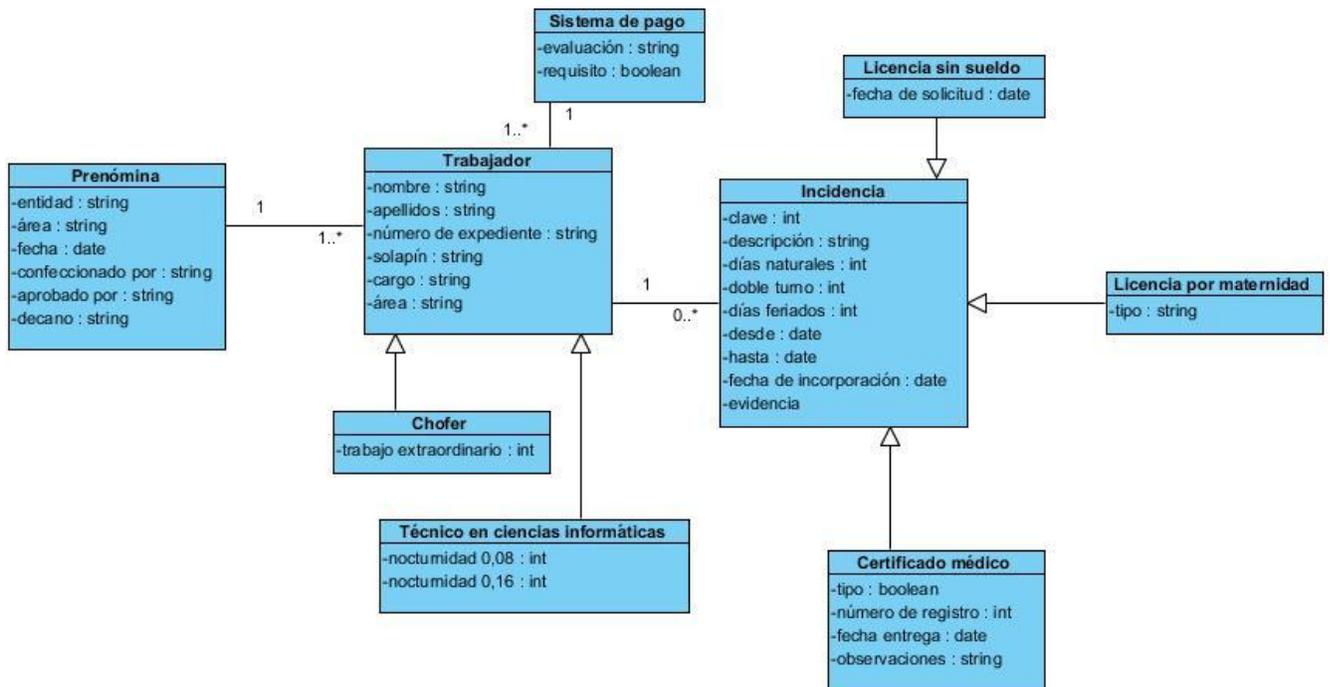


Figura 4: Modelo conceptual del proceso de negocio prenomina.

2.3. Descripción del proceso de negocio pre Nómina

Como parte de la gestión administrativa de los Recursos Humanos en las empresas estatales, centros de dirección del gobierno y ministerios el proceso de pre Nómina es vital para la posterior elaboración de la nómina de los trabajadores. El departamento encargado de realizar la pre Nómina es el de Recursos Humanos, el cual trabaja en estrecha relación con el departamento de contabilidad que entre sus funciones realiza la documentación para efectuar el pago a todos los trabajadores.

En el CEGEL este proceso se realiza de la siguiente manera; capital humano le envía por correo el archivo p4.csv¹ que es el listado de trabajadores con los datos personales a la persona encargada de realizar la pre Nómina y esta va comparando este listado para ver si hay bajas o altas a referenciar en la pre Nómina del mes. Posteriormente se empiezan a indicar las diferentes incidencias correspondientes al análisis de ausentismo por cada trabajador mediante claves numéricas únicas para cada incidencia, así como el pago adicional por conceptos de evaluación, nocturnidad, día feriado, trabajo extraordinario, requisito y doble turno. De las incidencias que necesiten evidencias como certificados médicos o licencias se guardarán en un archivo para la comprobación de la validez y legalidad del proceso de pre Nómina. Una vez terminada esta parte se procede a enviar a capital humano el modelo de pre Nómina listo para que el departamento de contabilidad elabore la nómina. En caso de haber algún error distintivo o duda respecto a algunas de las incidencias se comunican ambos departamentos para aclarar el asunto en cuestión vía correo electrónico o personándose la persona encargada de la pre Nómina en el departamento de Capital Humano. En la figura 5 se muestra el diagrama de actividades correspondiente al proceso de negocio.

¹ Los archivos CSV (del inglés comma-separated values) son un tipo de documento en formato abierto sencillo para representar datos en forma de tabla, en las que las columnas se separan por comas o punto y coma en donde la coma es el separador decimal y las filas por saltos de línea. Los campos que contengan una coma, un salto de línea o una comilla doble deben ser encerrados entre comillas dobles.

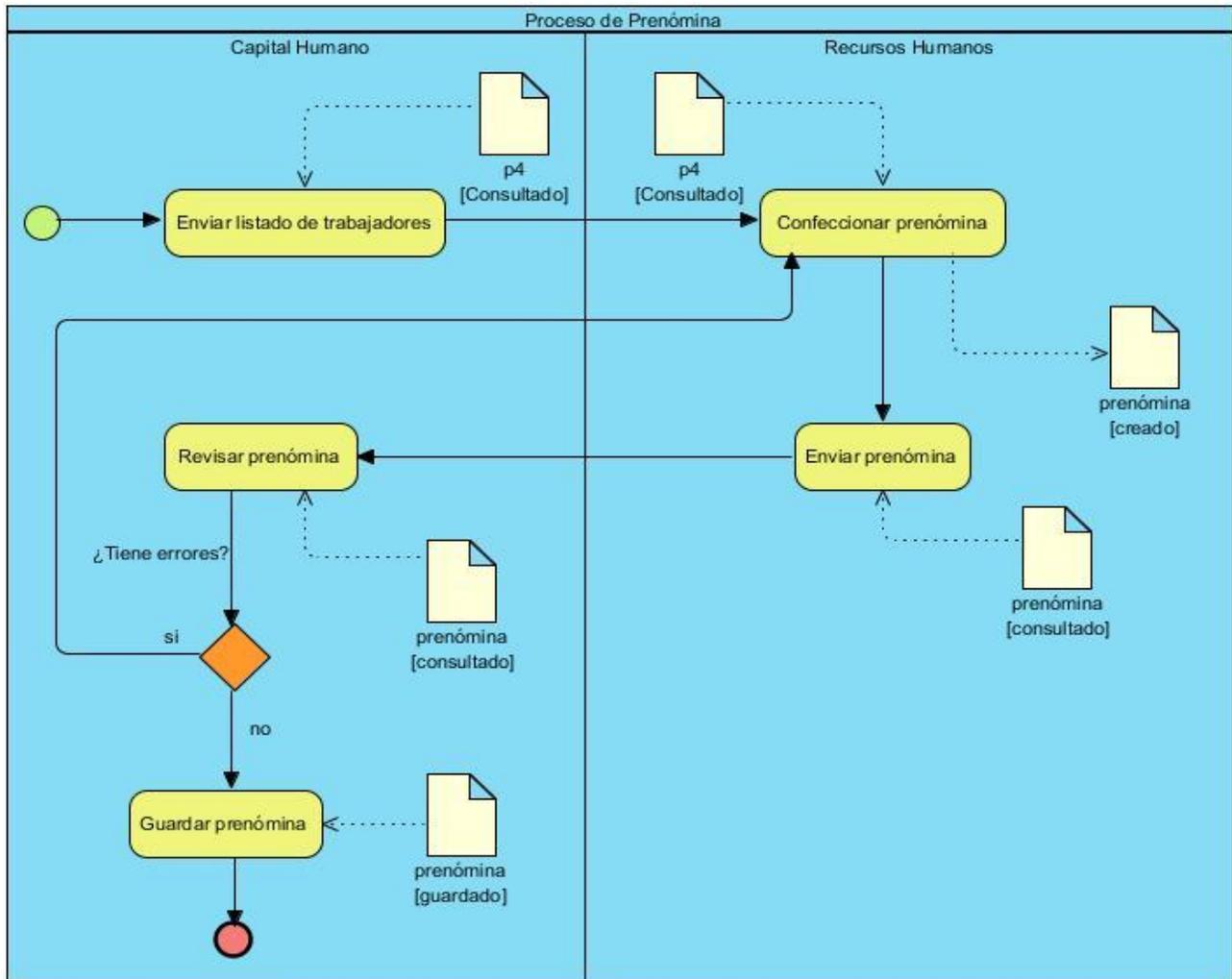


Figura 5: Diagrama de actividades correspondiente al proceso de pre-nómina.

2.4. Descripción de la propuesta de solución

La aplicación web a desarrollar permite la gestión administrativa de los Recursos Humanos del CEGEL y está diseñada de la siguiente manera: en la parte superior izquierda se muestra en nombre de la aplicación “Sistema de pre-nómina” y en la parte superior derecha se muestra el nombre y el usuario de la persona autenticada, debajo de esta estructura se encuentran en la parte izquierda los vínculos a las diferentes interfaces que se mostrarán a la derecha de los vínculos. En la aplicación sólo pueden entrar el usuario “administrador” y la persona encargada de confeccionar la pre-nómina de pago, la cual tiene un usuario en el sistema para poder autenticarse.

Los vínculos son: inicio, usuarios del sistema (solamente se muestra para el administrador), configurar fechas, trabajadores, incidencias y registrar incidencia en una primera parte y más abajo

los registros de prenomina, registro de bajas, registro de certificados médicos, registro de licencia sin sueldo y registro de licencia de maternidad. Una vez que un usuario se autentica se muestra la interfaz de inicio en la cual se carga el fichero p4.csv que capital humano le envía por correo donde se relacionan todos los trabajadores del CEGEL con sus correspondientes datos personales: número de expediente, nombre(s) y apellidos, solapín, cargo y área a la cual pertenecen. Cuando se carga el archivo se muestra el listado de trabajadores con sus datos en la interfaz Trabajadores.

En la interfaz usuarios del sistema se muestra el listado de los usuarios que pueden autenticarse y tres botones para adicionar, modificar y eliminar. Para adicionar un usuario es preciso introducir un nombre, un usuario que no puede estar repetido, una contraseña y la confirmación de la contraseña. En la interfaz configurar fechas se muestra el listado de las fechas deshabilitadas por el usuario, un botón para deshabilitar una nueva fecha y un botón para habilitarlas en caso de ser necesario; la aplicación por defecto deshabilita todos los domingos del año. En la interfaz trabajadores se muestra el listado de trabajadores con botones para adicionar uno nuevo, modificar los datos de uno existente, dar de baja a un trabajador y eliminarlo del listado.

En la interfaz incidencias se muestra el listado de las incidencias con las respectivas claves y descripciones con botones para adicionar una nueva, modificar los datos de una existente y eliminarla del listado. En la interfaz registrar incidencias se muestra el registro de incidencias de los diferentes trabajadores con los botones para adicionar y eliminar. Al adicionar una incidencia además de los campos de trabajador, incidencia, días naturales, doble turno, días feriados, requisito y evaluación se despliegan los campos: trabajo extraordinario si el trabajador es chofer, nocturnidad 0,08 y nocturnidad 0,16 si el trabajador es técnico en ciencias informáticas, desde, hasta, fecha de incorporación, fecha de entrega, número de registro, días a pagar, tipo de certificado, observaciones y evidencia si la incidencia es certificado médico, fecha de solicitud, fecha de salida, vacaciones acumuladas, motivo y evidencia si la incidencia es una licencia sin sueldo y prenatal con sus fechas, primera posnatal con sus fechas y segunda posnatal con sus fechas si la incidencia es licencia por maternidad.

En la interfaz prenomina se muestra el modelo correspondiente con los trabajadores agrupados por los cuatro departamentos: CEGEL, Práctica profesional, Desarrollo de aplicaciones y Desarrollo de componentes, donde al final del modelo hay un botón para exportar en formato .pdf la prenomina. Al hacer clic en los vínculos registro de bajas, registro de certificados médicos, registro de licencias sin sueldo y registro de licencia por maternidad se muestran las respectivas interfaces en una tabla con los datos captados en la interfaz registrar incidencias. A continuación se detallan los requisitos para la implementación de la aplicación.

2.5. Levantamiento de Requisitos

Teniendo en cuenta los aspectos teóricos que identifican requisitos y que fueron abordados en el capítulo 1 epígrafe 1.4, se definieron los siguientes requisitos funcionales (RF) y requisitos no funcionales (RnF), haciéndose necesaria la utilización de algunas técnicas que sirvieron para verificar la veracidad de los objetivos propuestos para el desarrollo de esta aplicación, permitiendo que dieran satisfacción a las necesidades del cliente. Dentro de las técnicas utilizadas se encuentran las entrevistas, aplicadas a las personas encargadas de realizar la pre Nómina en el CEGEL, que fueron consideradas proveedores válidos para las cuales va a ser desarrollada la aplicación. Estas entrevistas posibilitaron los debates, donde los clientes explicaron de forma clara la necesidad de la informatización del proceso de pre Nómina. Para la aplicación se determinaron los siguientes RF y RnF.

Listado de los Requisitos Funcionales:

RF1 – Generar documento de pre Nómina: A partir de que se registre una incidencia se visualice en el registro de pre Nómina.

RF2 – Exportar documento de pre Nómina: Opción para guardar en formato pdf el registro de pre Nómina.

RF3 – Visualizar registro de certificados médicos: Se registran en una misma tabla todos los certificados emitidos con los datos correspondientes del documento.

RF4 – Cargar evidencia del certificado médico: Guarda en el sistema el certificado escaneado.

RF5 – Visualizar registro de licencia sin sueldo: Se registran en una misma tabla todas las licencias sin sueldo solicitadas con los datos correspondientes.

RF6 – Visualizar registro de licencia de maternidad: Se registran en una misma tabla todas las licencias de maternidad solicitadas con los datos correspondientes.

RF7 – Adicionar usuario del sistema: Adiciona un usuario que tendrá privilegios para autenticarse e interactuar con las funciones de la aplicación.

RF8 – Eliminar usuario del sistema: Elimina un usuario del sistema.

RF9 – Adicionar trabajador: Adiciona un trabajador al listado de trabajadores.

RF10 – Eliminar trabajador: Elimina un trabajador del listado de trabajadores.

RF11 – Adicionar incidencia: Adiciona una incidencia con la clave correspondiente.

RF12 – Modificar incidencia: Actualiza una incidencia con la clave correspondiente.

RF13 – Eliminar incidencia: Elimina una incidencia.

RF14 – Registrar incidencia: Registra incidencias por cada trabajador.

RF15 – Cargar evidencia de la licencia: Guarda en el sistema la solicitud de licencia sin sueldo escaneada.

RF16 – Cargar archivo p4.csv: Carga los datos del archivo y los reemplaza en el listado de trabajadores.

RF17 – Dar baja a un trabajador: Permite dar de baja a un trabajador.

RF18 – Habilitar día hábil: Convierte un día natural en un día hábil.

RF19 – Deshabilitar día hábil: Convierte un día hábil en un día natural.

RF20 – Visualizar registro de bajas: Se registran en una misma tabla todas las bajas solicitadas con los datos correspondientes.

RF21 – Exportar registro de bajas: Opción para guardar en formato pdf el registro de bajas.

RF22 – Exportar registro de certificados médicos: Opción para guardar en formato pdf el registro de certificados médicos.

RF23 – Exportar registro de licencia sin sueldo: Opción para guardar en formato pdf el registro de licencia sin sueldo.

RF24 – Exportar registro de licencia de maternidad: Opción para guardar en formato pdf el registro de licencia de maternidad.

Listado de los Requisitos no Funcionales (RnF):

Usabilidad

RnF – 1 El sistema podrá ser utilizado por los usuarios que se creen además del usuario por defecto “administrador”.

RnF – 2 El sistema deberá presentar una interfaz de usuario fácil de entender y usar.

RnF – 3 Agrupar vínculos y botones por grupos funcionales. La consistencia de la interacción entre usuario y sistema estará determinada por el diseño de la interfaz de usuario que mantendrá los elementos como menús, banners y zona de trabajo, en posiciones fijas, además de la mayor uniformidad posible entre cuadros de texto y botones.

RnF – 4 El tiempo de respuesta brindado por el sistema será menor de 5 segundos.

Eficiencia

RnF – 5 El sistema debe ser capaz de dar respuestas a las peticiones con un nivel aceptable de desempeño.

Restricciones de diseño

RnF – 6 Para el montaje del sistema se requerirá del sistema gestor de bases de datos PostgreSQL 9.1 y del servidor de aplicaciones Apache.

Interfaz

RnF – 7 El sistema presentará una interfaz legible, simple de usar e interactiva.

Seguridad

RnF – 8 La información manejada por el sistema podrá ser accedida por las personas que tienen los permisos para ello, por lo que estará protegida de acceso no autorizado y divulgación. Los usuarios accederán solo a la información correspondiente a su rol siempre y cuando se proporcionen los datos necesarios.

Disponibilidad

RnF – 9 El tiempo máximo de inactividad es de 6 minutos mostrando a los 5 minutos un mensaje de alerta, una vez que el sistema quede inactivo el usuario deberá autenticarse nuevamente.

RnF – 10 El acceso a la información y los servicios web deberá estar disponible para los usuarios autorizados en cualquier momento que sea solicitada. Los servicios web deberán estar disponibles las 24 horas del día, los 7 días de la semana y todos los días del año.

Portabilidad

RnF – 11 Los servicios podrán ser consumidos desde cualquier sistema operativo.

Integridad

RnF – 12 Los datos contenidos en la Base de Datos no deberán ser redundantes o falsos. La información generada por el sistema será objeto de un alto nivel de protección contra la corrupción.

2.6. Historias de Usuario

Las historias de usuarios, como se menciona en el capítulo anterior, son uno de los artefactos generados por la metodología XP y se crean a partir de los requisitos funcionales definidos. A continuación se muestran las historias de usuario con prioridad alta concebidas a partir de los requisitos funcionales del sistema.

HISTORIA DE USUARIO		
No: 01	Iteración Asignada: 3	Nombre: Generar documento de pre Nómina.
Prioridad del Negocio: Alta.		Nivel de Complejidad: Alta.
Punto de Estimación: 2		Riesgo en el desarrollo: Alto.
Descripción: El documento de pre Nómina listo para agregar incidencias se genera cuando se carga el archivo p4.csv a la aplicación.		
Observaciones: Debe estar cargado en el sistema el archivo p4.csv correspondiente al listado de trabajadores de CEGEL con sus respectivos datos.		
HISTORIA DE USUARIO		
No: 02	Iteración Asignada: 3	Nombre: Exportar documento de pre Nómina.

Prioridad del Negocio: Alta.	Nivel de Complejidad: Media.
Punto de Estimación: 1	Riesgo en el desarrollo: Medio.
Descripción: Una vez agregadas todas las incidencias e introducidos los datos de los campos fecha, confeccionado por, aprobado por y decano(a), en la interfaz prenomina, el usuario hace clic en el botón exportar a pdf, una vez hecho esto se muestra una ventana para guardar en la computadora el archivo prenomina.pdf.	
Observaciones:	

HISTORIA DE USUARIO		
No: 03	Iteración Asignada: 4	Nombre: Visualizar registro de certificados médicos.
Prioridad del Negocio: Alta.	Nivel de Complejidad: Media.	
Punto de Estimación: 1	Riesgo en el desarrollo: Medio.	
Descripción: Cuando se hace clic en la opción registro certificados médicos del panel izquierdo de la aplicación se muestra en la parte derecha el registro de certificados médicos con los datos captados: número de expediente, solapín, nombre(s) y apellidos, cargo, número de registro, desde, hasta, días naturales, días a descontar, fecha incorporación, fecha entrega, tipo de certificado, observaciones y visualizar.		
Observaciones:		

HISTORIA DE USUARIO		
No: 04	Iteración Asignada: 3	Nombre: Cargar evidencia del certificado médico.
Prioridad del Negocio: Alta.	Nivel de Complejidad: Media.	
Punto de Estimación: 1	Riesgo en el desarrollo: Medio.	
Descripción: Cuando se adiciona una incidencia con clave 27 y descripción enfermedad más		

de 3 días se despliegan los datos que van a ser captados del certificado médico, uno de ellos permite que se suba el certificado médico escaneado (evidencia) a la aplicación.

Observaciones: Debe estar escaneado el certificado médico y guardado en la computadora.

HISTORIA DE USUARIO	
No: 05	Iteración Asignada: 4 Nombre: Visualizar registro de licencia sin sueldo.
Prioridad del Negocio: Alta.	Nivel de Complejidad: Media.
Punto de Estimación: 1	Riesgo en el desarrollo: Medio.
Descripción: Cuando se hace clic en la opción registro licencias sin sueldo del panel izquierdo de la aplicación se muestra en la parte derecha el registro de licencias sin sueldo con los datos captados: número de expediente, solapín, nombre(s) y apellidos, área, fecha solicitud, fecha salida, vacaciones acumuladas, fecha incorporación y visualizar.	
Observaciones:	

HISTORIA DE USUARIO	
No: 06	Iteración Asignada: 4 Nombre: Visualizar registro de licencia de maternidad.
Prioridad del Negocio: Alta.	Nivel de Complejidad: Media.
Punto de Estimación: 1	Riesgo en el desarrollo: Medio.
Descripción: Cuando se hace clic en la opción registro licencia de maternidad del panel izquierdo de la aplicación se muestra en la parte derecha el registro de licencias de maternidad con los datos captados: número de expediente, solapín, nombre(s) y apellidos, desde y hasta de prenatal, desde y hasta de primera posnatal, desde y hasta de segunda posnatal y promedio.	

Observaciones:

HISTORIA DE USUARIO

No: 09	Iteración Asignada: 1	Nombre: Adicionar trabajador.
---------------	------------------------------	--------------------------------------

Prioridad del Negocio: Alta.	Nivel de Complejidad: Media.
-------------------------------------	-------------------------------------

Punto de Estimación: 1	Riesgo en el desarrollo: Medio.
-------------------------------	--

Descripción: En la interfaz listado de trabajadores se muestra un botón adicionar trabajador que al hacerle clic se pasa a otra interfaz donde se introducen los datos: número de expediente, nombre(s) y apellidos, solapín, cargo y área. Hecho esto se adiciona el trabajador y se agrega al listado de trabajadores.

Observaciones:

HISTORIA DE USUARIO

No: 10	Iteración Asignada: 1	Nombre: Eliminar trabajador.
---------------	------------------------------	-------------------------------------

Prioridad del Negocio: Alta.	Nivel de Complejidad: Media.
-------------------------------------	-------------------------------------

Punto de Estimación: 1	Riesgo en el desarrollo: Medio.
-------------------------------	--

Descripción: En la interfaz listado de trabajadores se muestra en la columna acciones un botón eliminar que al hacerle clic aparece una notificación preguntando si desea eliminarlo o no, en caso afirmativo se elimina del listado y en caso negativo se regresa a la interfaz listado de trabajadores.

Observaciones:

HISTORIA DE USUARIO

No: 11	Iteración Asignada: 2	Nombre: Adicionar incidencia.
---------------	------------------------------	--------------------------------------

Prioridad del Negocio: Alta.	Nivel de Complejidad: Media.
-------------------------------------	-------------------------------------

Punto de Estimación: 1	Riesgo en el desarrollo: Medio.
Descripción: En la interfaz incidencias se muestra un botón adicionar que al hacerle clic se pasa a otra interfaz donde se introducen los datos: clave y descripción. Hecho esto se adiciona la incidencia y se agrega al listado de incidencias.	
Observaciones:	

HISTORIA DE USUARIO		
No: 12	Iteración Asignada: 2	Nombre: Modificar incidencia.
Prioridad del Negocio: Alta.		Nivel de Complejidad: Media.
Punto de Estimación: 1	Riesgo en el desarrollo: Medio.	
Descripción: En la interfaz incidencias se muestra en la columna acciones un botón modificar que al hacerle clic se pasa a otra interfaz donde se introducen los datos: clave y descripción. Hecho esto se actualiza la incidencia y se agrega al listado de incidencias.		
Observaciones:		

HISTORIA DE USUARIO		
No: 13	Iteración Asignada: 2	Nombre: Eliminar incidencia.
Prioridad del Negocio: Alta.		Nivel de Complejidad: Media.
Punto de Estimación: 1	Riesgo en el desarrollo: Medio.	
Descripción: En la interfaz incidencias se muestra en la columna acciones un botón eliminar que al hacerle clic aparece una notificación preguntando si desea eliminarla o no, en caso afirmativo se elimina del listado y en caso negativo se regresa a la interfaz listado de incidencias.		
Observaciones:		

HISTORIA DE USUARIO		
No: 14	Iteración Asignada: 2	Nombre: Registrar incidencia.
Prioridad del Negocio: Alta.		Nivel de Complejidad: Alta.
Punto de Estimación: 3		Riesgo en el desarrollo: Alto.
<p>Descripción: En la interfaz registrar incidencia se muestra un botón adicionar que al hacerle clic se pasa a otra interfaz donde se introducen los datos: trabajador, incidencia, días naturales, doble turno, días feriados, requisito y evaluación; en caso de que sea un certificado médico, licencia sin sueldo o licencia de maternidad se introducen también los datos descritos en las historias de usuario 03, 05 o 06 respectivamente. Hecho esto se adiciona la incidencia y se agrega al registro de incidencias, en caso de que se correspondan con una de las claves dichas anteriormente también se actualizan los registros correspondientes.</p>		
Observaciones:		

HISTORIA DE USUARIO		
No: 15	Iteración Asignada: 3	Nombre: Cargar evidencia de la licencia.
Prioridad del Negocio: Alta.		Nivel de Complejidad: Media.
Punto de Estimación: 1		Riesgo en el desarrollo: Medio.
<p>Descripción: Cuando se adiciona una incidencia con clave 25, 32 o 33 y descripción licencia sin sueldo posnatal (hasta 9 meses), licencia sin sueldo (problemas personales) y licencia sin sueldo (hasta 6 meses si hijo menor de 16) respectivamente, se despliegan los datos que van a ser captados de la licencia, uno de ellos permite que se suba la solicitud de licencia escaneada (evidencia) a la aplicación.</p>		
Observaciones: Debe estar escaneada la solicitud de la licencia y guardada en la computadora.		

HISTORIA DE USUARIO

No: 16	Iteración Asignada: 4	Nombre: Cargar archivo p4.csv.
Prioridad del Negocio: Alta.		Nivel de Complejidad: Alta.
Punto de Estimación: 1		Riesgo en el desarrollo: Alto.
Descripción: En la interfaz conciliación se muestra un botón para cargar el archivo p4.csv, una vez cargado el archivo se hace clic en el botón cargar trabajadores y se pasa a la interfaz listado de trabajadores.		
Observaciones: Debe estar guardado el archivo p4.csv en la computadora.		

HISTORIA DE USUARIO		
No: 17	Iteración Asignada: 4	Nombre: Dar baja a un trabajador.
Prioridad del Negocio: Alta.		Nivel de Complejidad: Media.
Punto de Estimación: 1		Riesgo en el desarrollo: Medio.
Descripción: En la interfaz listado de trabajadores se muestra en la columna acciones un botón dar baja que al hacerle clic se pasa a otra interfaz donde se visualizan los datos obtenidos de la interfaz anterior y se introducen los datos: causa de baja y fecha de baja. Hecho esto se actualiza el registro de bajas y se actualiza el listado de trabajadores.		
Observaciones:		

HISTORIA DE USUARIO		
No: 18	Iteración Asignada: 4	Nombre: Habilitar día hábil.
Prioridad del Negocio: Alta.		Nivel de Complejidad: Alta.
Punto de Estimación: 1		Riesgo en el desarrollo: Alto.
Descripción: Cuando se hace clic en la opción configurar fechas del panel izquierdo de la		

aplicación se muestra en la parte derecha las fechas deshabilitadas, en la columna acciones se muestra un botón habilitar donde al hacer clic aparece una notificación preguntando si desea habilitarla o no, en caso afirmativo se elimina del listado y en caso negativo se regresa a la interfaz fechas deshabilitadas.

Observaciones:

HISTORIA DE USUARIO

No: 19	Iteración Asignada: 4	Nombre: Deshabilitar día hábil.
---------------	------------------------------	--

Prioridad del Negocio: Alta.	Nivel de Complejidad: Alta.
-------------------------------------	------------------------------------

Punto de Estimación: 1	Riesgo en el desarrollo: Alto.
-------------------------------	---------------------------------------

Descripción: En la interfaz fechas deshabilitadas se muestra un botón deshabilitar nueva fecha que al hacerle clic se pasa a otra interfaz donde se introduce la fecha. Hecho esto se adiciona la fecha y se agrega al listado de fechas deshabilitadas.

Observaciones: La aplicación deshabilita de manera automática todos los domingos.

HISTORIA DE USUARIO

No: 20	Iteración Asignada: 4	Nombre: Visualizar registro de bajas.
---------------	------------------------------	--

Prioridad del Negocio: Alta.	Nivel de Complejidad: Media.
-------------------------------------	-------------------------------------

Punto de Estimación: 1	Riesgo en el desarrollo: Medio.
-------------------------------	--

Descripción: Cuando se hace clic en la opción registro de bajas del panel izquierdo de la aplicación se muestra en la parte derecha el registro de bajas con los datos captados: número de expediente, solapín, nombre(s) y apellidos, cargo, área, causa de la baja y fecha de la baja.

Observaciones:

Estimación del esfuerzo por Historias de Usuario y Plan de Iteraciones

En la metodología XP, la creación del sistema se divide en etapas para facilitar su realización. Por lo general, los proyectos de desarrollo constan de más de tres etapas, las cuales toman el nombre de iteraciones; de allí se obtiene el concepto de metodología iterativa. La duración ideal de una iteración es de una a tres semanas.

Para cada iteración se define un módulo o conjuntos de historias que se van a implementar. Al final de cada iteración se obtiene como resultado la entrega del módulo correspondiente, el cual debe de haber superado las pruebas de aceptación que establece el cliente para verificar el cumplimiento de los requerimientos. La siguiente tabla muestra los resultados de estimación obtenidos por cada Historia de Usuario identificada, así como la prioridad, número y fin de iteración.

Núm. de Iteración	Historia de Usuario	Prioridad	Puntos de Estimación		Fin de Iteración
1	Adicionar usuario del sistema.	Media	1	4	1ra Sem. de Feb.
1	Eliminar usuario del sistema.	Media	1		
1	Adicionar trabajador.	Alta	1		
1	Eliminar trabajador.	Alta	1		
2	Adicionar incidencia.	Alta	1	6	3ra Sem. de Marzo
2	Modificar incidencia.	Alta	1		
2	Eliminar incidencia.	Alta	1		
2	Registrar incidencia.	Alta	3		
3	Cargar evidencia de la licencia.	Alta	1	5	4ta Sem. de Abril
3	Cargar evidencia del certificado médico.	Alta	1		
3	Generar documento de prenómina.	Alta	2		
3	Exportar documento de prenómina.	Alta	1		

4	Visualizar registro de certificados médicos.	Alta	1	3	3ra Sem. de Mayo
4	Visualizar registro de licencia sin sueldo.	Alta			
4	Visualizar registro de licencia de maternidad.	Alta			
4	Cargar el archivo p4.csv.	Alta	1		
4	Dar baja a un trabajador.	Alta			
4	Habilitar día hábil.	Alta			
4	Deshabilitar día hábil.	Alta			
4	Visualizar registro de bajas.	Alta			
4	Exportar registro de bajas.	Media	1		
4	Exportar registro de certificados médicos.	Media			
4	Exportar registro de licencia sin sueldo.	Media			
4	Exportar registro de licencia de maternidad.	Media			

2.7. Diseño

A diferencia de las metodologías pesadas, el diseño se realiza durante todo el tiempo de vida del proyecto, siendo constantemente revisado y muy probablemente modificado debido a cambios presentados. XP establece prácticas especializadas, que inciden directamente en la realización y elaboración del diseño de un software, sin embargo, no requiere que la representación del sistema sea mediante diagramas de clases basados en UML (no obstante se consideró importante para un mejor entendimiento mostrar el diagrama de clases persistentes mediante la figura 6), sino que pueden emplearse indistintamente sencillos esquemas descritos en pizarras u otras técnicas como las tarjetas Clase-Responsabilidad-Colaboración (CRC).

Tarjetas CRC

Las Tarjetas CRC son artefactos de diseño que se generan como parte del proceso XP. Permiten visualizar el sistema en términos de objetos que componen la aplicación, identificando clases candidatas, responsabilidades y la forma que colaboran entre ellas. Para la aplicación fueron

especificadas 20 tarjetas CRC, donde cada tarjeta debe corresponder a una o más Historias de Usuario. A continuación se ejemplifica una tarjeta CRC correspondiente a la clase Incidencia.php donde se evidencia la responsabilidad que tiene la clase y la relación que tiene con otra.

TARJETA CRC	
Clase: Incidencia.php	
Responsabilidades:	Colaboraciones
Contiene todas las variables correspondientes a cada una de las incidencias, así como los métodos set y get.	IncidenciaController.php

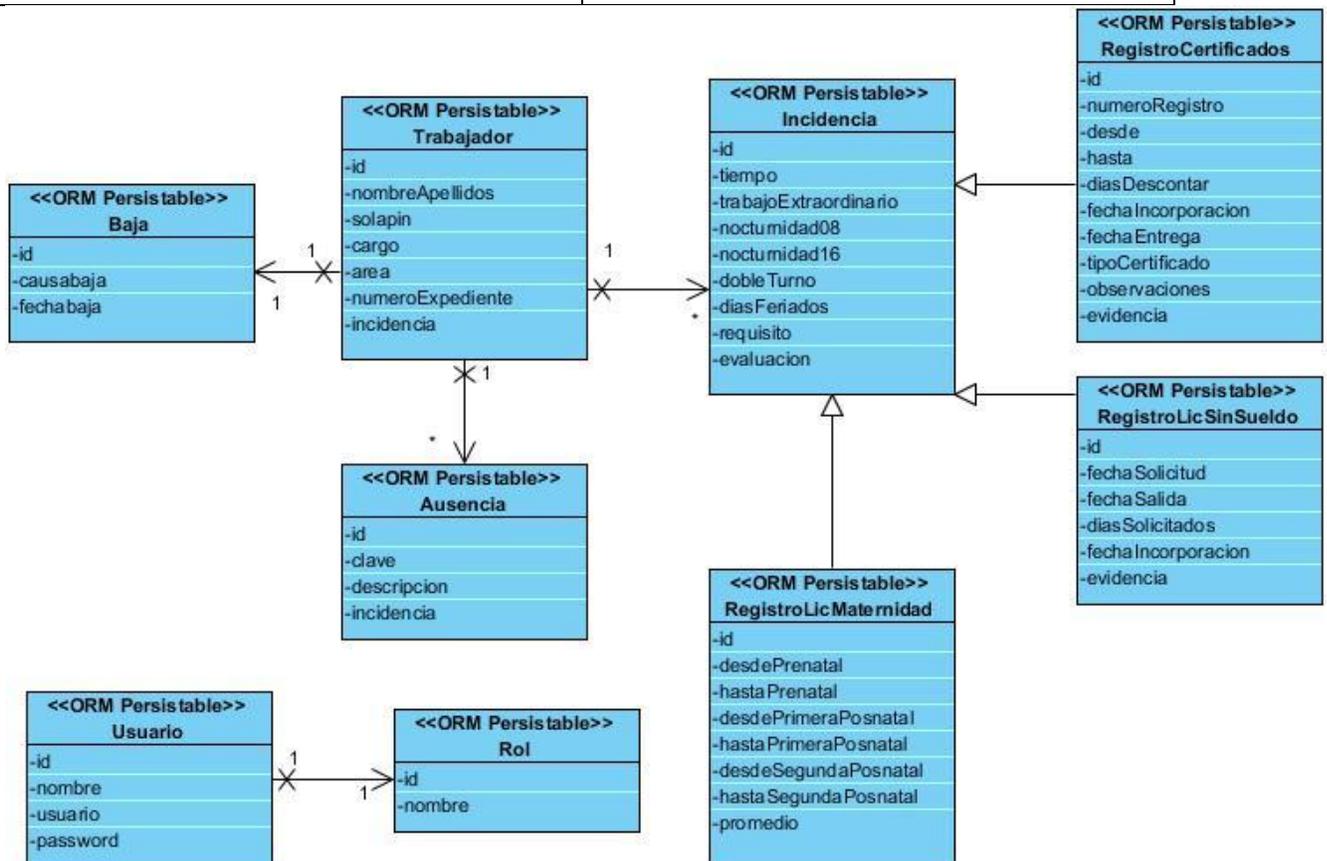


Figura 6: Diagrama de clases persistentes.

Patrón Arquitectónico

Los patrones arquitectónicos, o patrones de arquitectura, son patrones de diseño de software que ofrecen soluciones a problemas de arquitectura de software en ingeniería de software. Dan una descripción de los elementos y el tipo de relación que tienen junto con un conjunto de restricciones sobre cómo pueden ser usados. Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones. En comparación con los patrones de diseño, los patrones arquitectónicos tienen un nivel de abstracción mayor.(Pressman 2005)

El patrón **Modelo-Vista-Controlador** (MVC) 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.

Este patrón arquitectónico presenta varias ventajas:

- Hay una clara separación entre los componentes de un programa; lo cual permite implementarlos por separado.
- Hay un API muy bien definido; cualquiera que use el API, podrá reemplazar el Modelo, la Vista o el Controlador, sin aparente dificultad.
- La conexión entre el Modelo y sus Vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.

Se decidió utilizar este patrón para el desarrollo del sistema en cuestión debido a que el marco de trabajo utilizado, Symfony 2, está basado en el mismo y a que las características y ventajas anteriormente expuestas permiten que el mantenimiento de la aplicación sea más sencillo.

A continuación se muestra una vista transversal a la aplicación en la figura 7.

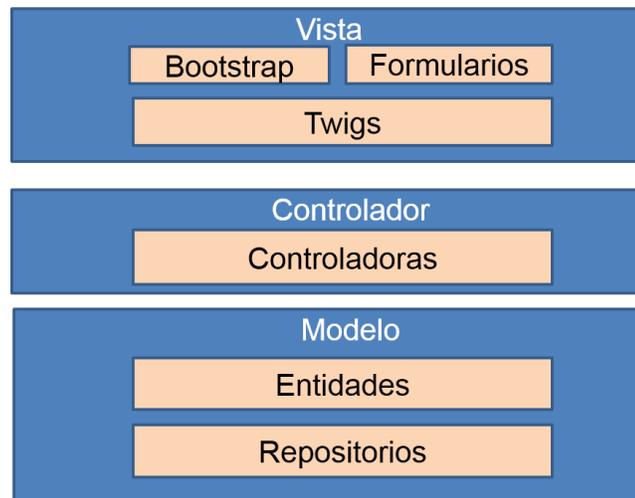


Figura 7: Vista transversal de la aplicación utilizando MVC.

Patrones de Diseño

Un patrón de diseño provee un esquema para refinar los subsistemas o componentes de un sistema de software, o las relaciones entre ellos. Describe la estructura comúnmente recurrente de los componentes en comunicación, que resuelve un problema general de diseño en un contexto particular. Son menores en escala que los patrones arquitectónicos, y tienden a ser independientes de los lenguajes y paradigmas de programación. Su aplicación no tiene efectos en la estructura fundamental del sistema, pero sí sobre la de un subsistema, debido a que especifica a un mayor nivel de detalle, sin llegar a la implementación, el comportamiento de los componentes del subsistema.

Patrones GRASP (Patrones de Principios Generales para Asignar Responsabilidades)

Experto: el uso de este patrón resulta de utilidad en las clases del modelo, las cuales contienen toda la información relacionada con los objetos persistentes que representan, ejemplo la clase Usuario.php.

Creador: este patrón se pone de manifiesto en las clases controladoras del sistema. Brinda soporte a un bajo acoplamiento y mejora la reutilización, ejemplo la clase UsuarioController.php.

Bajo acoplamiento: se evidencia en el marco de trabajo, ya que dentro de la capa modelo las clases de abstracción de datos son las más reutilizables y no tienen asociaciones con las clases de la capa vista ni con el controlador.

Alta cohesión: se evidencia en las clases repositorio, que tienen como única responsabilidad realizar operaciones de acceso a datos solo con la entidad que representa.

Controlador: este patrón se evidencia en las clases controladoras, responsables de implementar las funcionalidades pertenecientes a una interfaz determinada, ejemplo la clase `TrabajadorController.php`.

Patrones GoF (Gang-of-Four, por su traducción al español “grupo de los cuatro”)

Método de Fábrica: En Symfony 2 a la hora de crear un formulario se llama al método `createForm()`, el cual decide el tipo de formulario a crear, así como los tipos de datos correspondientes a los campos de los formularios (`text`, `textarea`, `select`).

Adaptador: La clase `Usuario.php` implementa la interfaz `UserInterface`, el cual es un componente del núcleo de Seguridad de Symfony2. Esta le permite a la clase `Usuario` implementar los métodos `eraseCredentials()`, `getUsername()` y `getRoles()`, necesarias para el correcto funcionamiento de la autenticación en Symfony2.

Decorador: se evidencia en la clase `index.html.twig` el cual permite añadir funcionalidades o responsabilidades a un objeto de forma dinámica evidenciándose en la herencia de plantilla de Twig.

Mediador: Las clases controladoras son mediadoras entre las capas de modelo y vista.

Estándar de codificación

XP enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que sigan ciertos estándares de codificación. Un estándar de codificación es un conjunto de reglas de notación y nomenclatura específicas para cada lenguaje de programación que se usan y siguen durante la fase de codificación de un software. La aplicación de un estándar de programación al código fuente facilita el mantenimiento de un software, además de reducir el riesgo de que los desarrolladores introduzcan errores que no sean detectados por los compiladores, minimizando así el tiempo y coste de las actividades de depuración y pruebas necesarias para la detección y corrección de los mismos.(CHARTE OJEDA 2002)

Para la implementación de la aplicación se tuvieron en cuenta las siguientes reglas de codificación:

- Utilización de nombres descriptivos que ayuden a una mejor comprensión.

- Los nombres de las clases o interfaces tendrán la primera letra de cada palabra en mayúscula, ejemplo: Usuario.php.
- El nombre de los atributos de una clase comienza con la primera letra en minúscula.
- Cada método será público.
- El nombre de cada clase controladora debe terminar con la palabra “Controller”.
- Se utilizará el método Post para las llamadas.
- Los nombres de los métodos reflejan la acción a realizar y siempre comenzando con letra minúscula, en caso de ser un nombre compuesto, todas las palabras que lo componen menos la primera se escriben con letra inicial mayúscula, ejemplo: editAction().
- Cada funcionalidad debe tener un comentario de su funcionamiento.

2.8. Conclusiones parciales

En este capítulo, quedaron definidas las características para el desarrollo de la solución. Además, fueron concebidas las historias de usuario para lograr una planificación en el desarrollo de la aplicación. Las tarjetas CRC fueron elaboradas para las clases de la aplicación, con el objetivo de plasmar sus responsabilidades y colaboradores. Además, se definieron los patrones de diseño utilizados para lograr una correcta estructuración del sistema.

CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA

3.1. Introducción

En el presente capítulo se realizarán las validaciones correspondientes tal como lo propone la metodología de software utilizada. Las pruebas unitarias y de aceptación serán las que se le realizarán a la aplicación web por los métodos de caja blanca con la técnica del camino básico y de caja negra con los casos de pruebas enviados a Calidad de CEGEL. Se utilizará la técnica ladov para medir el grado de satisfacción de los clientes. Se cuantificarán los resultados obtenidos y se graficarán con el objetivo de su apreciación visual.

3.2. Validación de las variables de la investigación

En este epígrafe se define la variable conceptual gestión administrativa de los Recursos Humanos, cuya dimensión es la aplicación, con los indicadores elaboración, registro y control. Dichos indicadores son medidos por la magnitud tiempo y se basa en responder la pregunta, ¿cuánto tiempo se demora la actividad de elaboración, registro de la información y control de evidencias? Para esto se escogen los escenarios: como se realizan en la actualidad (antes) y en la aplicación (después). Se exponen los resultados de la validación de las variables de la investigación en el gráfico 1.

Variable Conceptual	Dimensión	Indicadores
Gestión administrativa de los Recursos Humanos.	Entorno del Sistema Informático para la gestión administrativa de los Recursos Humanos en el CEGEL.	<ul style="list-style-type: none"> • Elaboración • Registro • Control

Indicadores	Antes	Después
Elaboración	Manual: Se registra cada incidencia llenando todos los campos necesarios del modelo de prenomina.	Automático: El sistema genera automáticamente la prenomina a partir de la carga del archivo p4.csv y calcula los tiempos requeridos en los campos del modelo a llenar.
Registro	Manual: Se registran los datos en diferentes documentos Excel,	Automático: El sistema genera automáticamente los registros de bajas,

	llenando todos los campos necesarios de las plantillas.	certificados médicos y licencias una vez que es introducida la incidencia correspondiente.
Control	Manual: Los documentos se escanean y se guardan en una carpeta.	Automático: El sistema permite visualizar una evidencia para corroborar la veracidad de la incidencia.

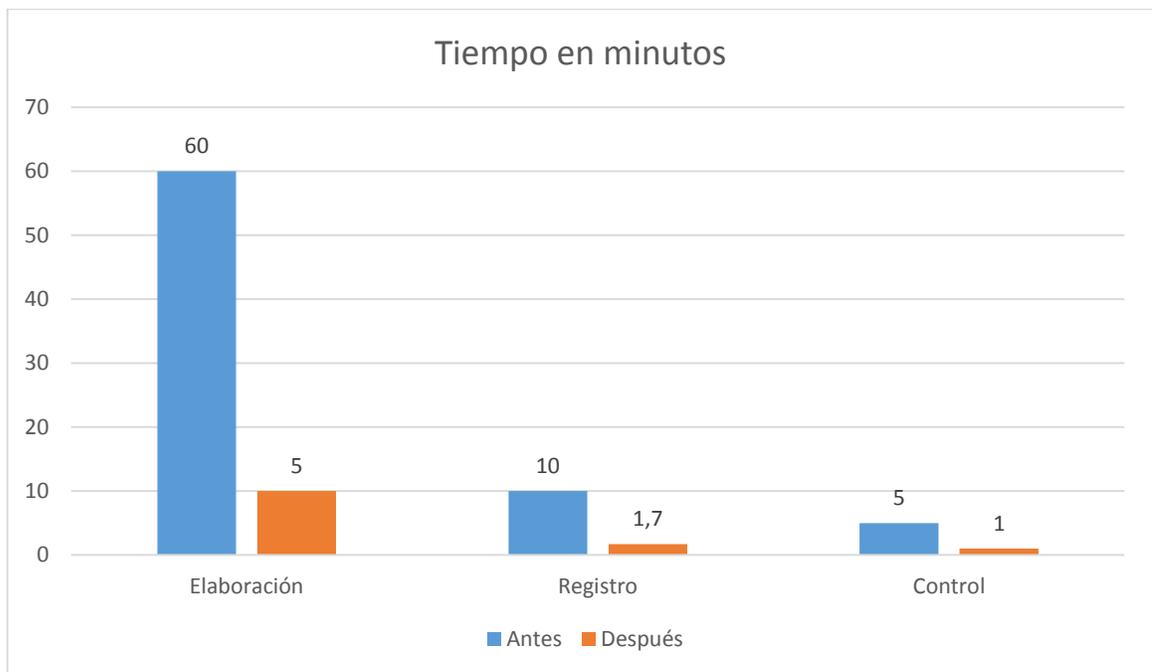


Figura 8: Comparación del tiempo de los indicadores antes y después.

Como se aprecia anteriormente en las tablas y en la figura 8 es notable la disminución del tiempo aun cuando se midió con una carga de trabajo relativamente baja y sin errores en la confección de las actividades.

3.3. Validación de la Fase de Desarrollo

Las pruebas unitarias son desarrolladas por los programadores y se encargan de verificar el código automáticamente y las pruebas de aceptación están destinadas a verificar que al final de cada iteración las HU cumplen con la funcionalidad asignada y satisfagan las necesidades del cliente. El método de pruebas de caja blanca precisa del acceso al código del programa de modo que se pueda comprobar su lógica interna. Al sistema desarrollado se le aplicaron tanto pruebas unitarias como de aceptación, tal y como propone XP.

Para la aplicación de las pruebas de caja blanca se hizo uso de la técnica camino básico, el cual permite obtener una medida de la complejidad de un diseño procedimental, y utilizar esta medida como guía para la definición de una serie de caminos básicos de ejecución, diseñando casos de prueba que garanticen que cada camino se ejecuta al menos una vez y que no existan ciclos infinitos. Se toma de ejemplo el método `updateAction()`, perteneciente a la clase `TrabajadorController` como base para realizar la técnica del camino básico.

El método `updateAction` consiste en mostrar la página `edit.html.twig` enviándole un formulario para modificar los datos del trabajador seleccionado. Este método recibe el id del trabajador como parámetro. Luego se busca la entidad y se crean los formularios de edición y eliminar y se envían a la vista. En caso de que los datos introducidos estén correctos se actualizan en el listado de trabajadores, en caso negativo se redirecciona al listado de trabajadores sin actualizar. Partiendo de la explicación del fragmento de código tomado se obtuvo el siguiente grafo de flujo que muestra la figura 9.

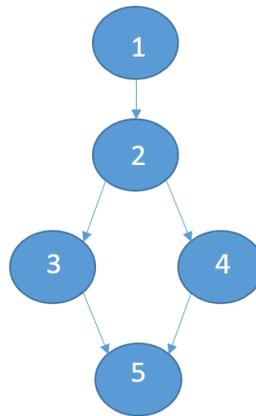


Figura 9: Representación del grafo de flujo del camino básico.

Luego se calculó la complejidad ciclomática $V(G)$ por tres vías para corroborar las respuestas de las fórmulas, obteniéndose como resultado lo siguiente:

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

$$V(G) = 5 - 5 + 2$$

$$V(G) = 2$$

$V(G)$ = número de regiones del grafo.

$$V(G) = 2$$

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

$$V(G) = 2$$

El valor $V(G)$ expresa la cantidad de caminos linealmente independientes de la estructura de control del programa, por lo que se definen los siguientes dos caminos:

Camino básico 1: 1-> 2 -> 3 -> 5

Camino básico 2: 1-> 2 -> 4 -> 5

Cada camino independiente es un caso de prueba a realizar, de forma que los datos señalados causen que se visiten las sentencias vinculadas a cada nodo de camino. En el caso anterior se calcularon dos caminos básicos por tanto surge la necesidad de hacer igual número de casos de prueba, para aplicar las pruebas a este método.

Entrada	Un trabajador a actualizar.
Resultados esperados	Se envían satisfactoriamente todos los datos hacia la página edit.html.twig y se redirige a index.html.twig que muestra el listado de trabajadores con los datos actualizados.
Condiciones	El id del trabajador tiene que ser igual al id de la entidad provisto.

Entrada	Un trabajador a actualizar.
Resultados esperados	No se envían satisfactoriamente todos los datos hacia la página edit.html.twig y se redirige a index.html.twig que muestra el listado de trabajadores con los datos sin modificar.
Condiciones	El id del trabajador tiene que ser igual al id de la entidad provisto.

Una vez ejecutadas las pruebas de caja blanca a través de la técnica del camino básico al método `updateAction()`, no se detectó error alguno, de esta manera se comprobó que el flujo de trabajo de las funciones implicadas en el método es correcto y cumple con las condiciones planteadas anteriormente.

Las pruebas de aceptación se definen junto al cliente y persiguen el objetivo de comprobar el comportamiento del sistema a partir de los requisitos funcionales establecidos. Uno de los artefactos necesarios para realizar adecuadamente las pruebas de aceptación son los diseños de casos de prueba junto a los juegos de datos a probar. La siguiente tabla corresponde al caso de prueba diseñado para la Historia de Usuario 01.

CASO DE PRUEBA DE ACEPTACIÓN 01			
Historia de Usuario 01:		Generar documento de prenomina.	
Descripción:		A partir de cargar el archivo p4.csv se genera la prenomina para que se le inserten las incidencias por trabajador.	
Condiciones de ejecución:		Tiene que estar cargado en la aplicación el archivo p4.csv	
Escenarios de prueba:		Flujo del escenario:	Resultados esperados:
EP1	Se carga el archivo p4.csv para generar la prenomina a rellenar introduciendo parámetros válidos.	En el registro de prenomina se muestra un listado de trabajadores de CEGEL organizados por áreas con su respectiva información personal.	La aplicación muestra el listado organizado por áreas y la información asociada a cada trabajador.
EP2	Se carga el archivo p4.csv para generar la prenomina a rellenar introduciendo parámetros inválidos.	En el registro de prenomina no se muestra un listado de trabajadores de CEGEL organizados por áreas con su respectiva información personal.	Se lanza una excepción mostrando el error.

Luego de confeccionar los casos de pruebas para cada HU se procede a comprobar el funcionamiento del sistema en una primera iteración. Esta comprobación se realiza con el objetivo de encontrar inconformidades en las respuestas esperadas por la aplicación e intentar corregirlas si existen. La figura 10 refleja los resultados obtenidos en la primera iteración realizada.

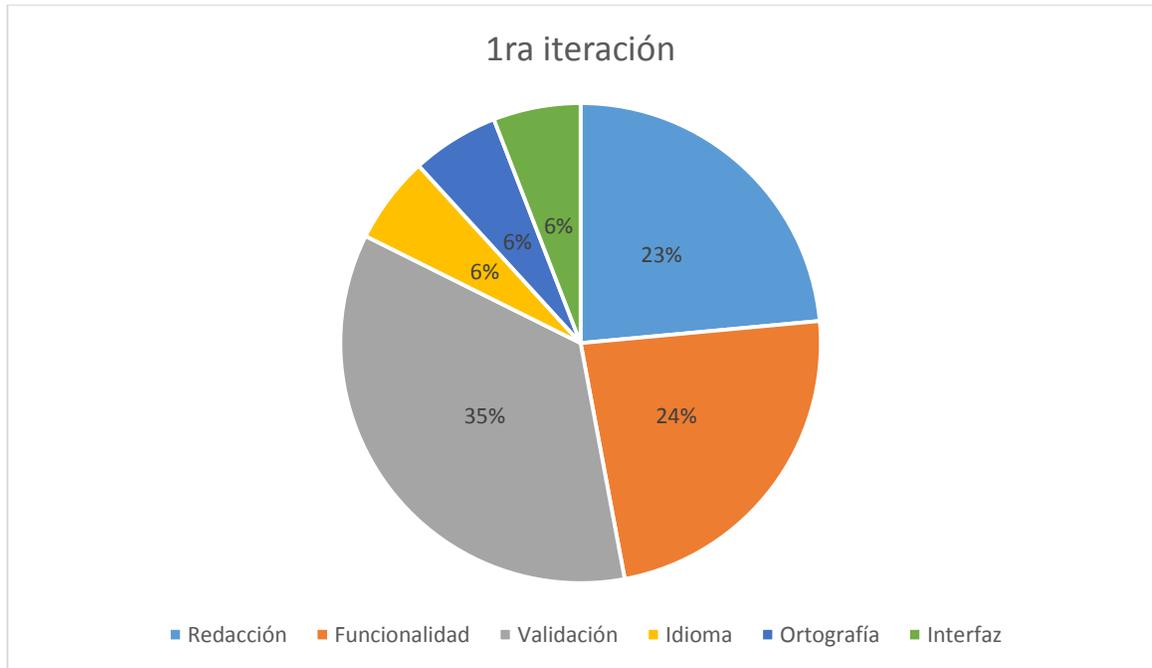


Figura 10: Primera iteración de las Pruebas de Aceptación

Como se puede observar, las pruebas realizadas arrojaron 17 no conformidades que se clasifican en Redacción, Funcionalidad, Validación, Idioma, Ortografía e Interfaz. A raíz de este resultado se procede a realizar un análisis cuidadoso del código para intentar erradicar en su totalidad los principales problemas encontrados. En el interés de comprobar si se suprimieron todos los problemas se realiza una segunda iteración donde se prueban las historias de usuario que presentan errores. En la segunda iteración se detectaron 2 no conformidades clasificadas en Funcionalidad y Ortografía dando lugar a una tercera iteración con el propósito de llegar a 0 no conformidades.

La totalidad de los problemas se eliminan en esta tercera iteración luego de realizado un análisis profundo del código fuente; se identifican los problemas y se procede a darles solución. En el Anexo se puede consultar la carta de aceptación emitida por el cliente y también director de CEGEL Ing. Darián González Ochoa mostrando su conformidad y satisfacción con la propuesta realizada, además de la Carta de Liberación Interna de Productos Software emitida por el Asesor de Calidad de CEGEL Ing. Yordanis García Leyva.

Para valorar el grado de satisfacción de los clientes con la solución desarrollada respecto al control de la información, se aplicó la técnica ladov que permite el estudio del grado de satisfacción del personal involucrado. Durante la valoración fungieron como clientes: el director de CEGEL, la persona encargada de realizar el proceso de nómina y especialistas en el área de los Recursos Humanos, para un total de cinco personas. Como resultado se obtuvo la tabla 2.

Tabla 2: Cuadro lógico de ladov.

¿Le gustaría hacer uso de la aplicación web propuesta para la gestión administrativa de los Recursos Humanos?	¿Considera usted oportuno continuar realizando el proceso de nómina manualmente, a pesar del gran volumen de información que se genera?								
	No			No sé			Sí		
	¿El desarrollo del sistema informático para la gestión administrativa de los Recursos Humanos contribuye a mejorar la elaboración, el registro y el control de la información asociada al proceso de nómina en el CEGEL?								
	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No
Me gustaría mucho	1	2	6	2	2	6	6	6	6
No me gustaría mucho	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgustaría más de lo que me gusta	6	3	6	3	4	4	3	4	4
No me gustaría nada	6	6	6	6	4	4	6	4	6
No sé qué decir	2	3	6	3	3	3	6	3	4

$$ISG = \frac{5 * 1}{5} = 1$$

Los resultados de la satisfacción individual según las categorías empleadas fueron los siguientes:

Nivel de satisfacción	Cantidad	%
Máxima satisfacción	5	100
Más satisfecho que insatisfecho	0	0
No definida	0	0

Al procesar las respuestas a las encuestas en el cuadro lógico de ladov, se obtiene un grado de satisfacción grupal de 1, lo cual se traduce en una clara satisfacción con el uso de la aplicación. En el criterio respecto a mejorar la elaboración, el registro y el control de la información a través del uso de la solución propuesta, hubo una concordancia de un 100% en que contribuye a su mejora. De igual manera el 100% de los participantes manifestó que le gustaría mucho hacer uso de esta aplicación para desarrollar el proceso de prenomina.

Las preguntas abiertas que se formularon fueron:

- ¿Qué valoraciones le sugiere a la aplicación respecto a la elaboración, el registro y el control de la información asociada a la prenomina?
- ¿Qué elemento(s) usted adicionaría a la solución que se propone?

Entre las valoraciones positivas obtenidas como respuestas a las preguntas abiertas, se recopilaron criterios como los siguientes:

- El sistema permite una elaboración, un registro y un control más eficientes del proceso de prenomina.
- Facilita la obtención de reportes para una gestión proactiva desde la dirección de CEGEL.

La aplicación de la técnica de ladov aportó información significativa respecto al grado de satisfacción del cliente. Los resultados obtenidos y los criterios emitidos validan la fortaleza de la propuesta, reflejándose una valoración muy positiva del cliente con la solución.

3.3. Conclusiones parciales

La realización de pruebas de software a la aplicación permitió comprobar la calidad del sistema propuesto. Las pruebas de unidad, enfocadas en verificar el correcto funcionamiento de cada método arrojaron resultados satisfactorios y esperados. Por su parte la realización de pruebas de aceptación,

luego de una tercera iteración, aprobaron el correcto funcionamiento y el cumplimiento de los requisitos de software definidos para la aplicación. Se aplicó la técnica de ladov para medir el grado de satisfacción de los clientes donde se arrojó una respuesta de máxima satisfacción. Todo lo anterior permitió validar la solución propuesta dando lugar al cumplimiento del objetivo general de la investigación.

CONCLUSIONES GENERALES

Luego de finalizada la investigación, presentada la propuesta de solución y realizadas las pruebas de software se puede arribar a las siguientes conclusiones:

- La construcción del marco teórico conceptual de la investigación contribuyó a mejorar los problemas y dificultades detectados en la gestión de la información de los Recursos Humanos.
- La implementación de un sistema informático de gestión administrativa de los Recursos Humanos contribuyó a disminuir las afectaciones existentes en el trabajo de gestión de la información de los procesos así como la elaboración, control y registro de evidencias.
- La realización de pruebas de software, correspondientes a la metodología de desarrollo seleccionada, posibilitó validar las funcionalidades de la aplicación permitiendo catalogar como satisfactoria la propuesta de solución y cumpliendo en su totalidad con el objetivo general de la investigación.

RECOMENDACIONES

Como recomendaciones generales de la investigación se tienen las que se listan a continuación:

1. Utilizar la aplicación con el objetivo de facilitar el trabajo del área de Recursos Humanos del CEGEL.
2. Incorporar nuevos registros para mantener la información asociada a la nómina centralizada en la aplicación.
3. Implementar técnicas de Inteligencia Artificial para validar que las evidencias se correspondan con los modelos autorizados legalmente.

REFERENCIAS BIBLIOGRÁFICAS

Achour, M. B., Friedhelm; Dovgal, Antony; Lopes, Nuno; Olson, Philip; Richter, Georg; Seguy, Damien y Vrana, Jakub (2006). "PHP Manual."

Amaro Calderón, S. D. y. V. R., Jorge Carlos (2007). "Metodologías Ágiles."

Beck, K. (1999). "Extreme Programming Explaine."

Bolton, D. (2014). "Definition of IDE. In."

Canós, J. H. L., Patricio y Penadés, María del Carmen (2004). "Métodologías Ágiles en el Desarrollo de Software."

Carrasco Puebla, Y. A. y. C. G., Enrique (2009).

Contabilidad, D. d. F. y. (2010).

Corporation, O. (2013). "NetBeans IDE. The Smarter and Faster Way to Code."

CHARTE OJEDA, F. (2002). "Visual C# .NET Guía práctica para usuarios."

Datazucar (2012).

Desoft (2015).

Eguiluz, J. (2011). "Desarrollo web ágil con Symfony2."

Foundation, T. A. S. (2011). "Documentación del Servidor HTTP Apache 2.0."

Fowler, M. (2003). "The New Methodology."

García, F. Ó. R. a. S., Crescencio Bravo (2009-2010).

Group, T. P. G. D. (1996-2013).

Hughes, D., et al. (2009). Aplicación para la administración y desarrollo de cursos con la técnica de Metaplan que aporta etapas virtuales mediante la Web. XV Congreso Argentino de Ciencias de la Computación.

IDEA, I. (2012). "Lightweight and smart PHP IDE."

Jeffries, R. A., A. AND Hendrickson, C. (2001). "eXtreme Programming Installed."

Letelier (2006).

Letelier, P. A. P., María del Carmen (2004). "Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). Laboratorio de Sistemas de Información. Departamento de Sistemas Informáticos y Computación. Facultad de Informática. Universidad Politécnica de Valencia."

Loucopulos, P., Karacostas, V. (1995). "System Requirements engineerinuality which will perform effectively."

Mark (2010). "Learning Python, Fourth Edition."

paradigm, V. (2012).

Pilgrim, M. (2010). "HTML5 Up and Running."

Pressman, R. (2005). "La Ingeniería del Software, un enfoque práctico."

Rincón Carrera, J. M. (2011). "Estudio de la Evolución Web y Lenguajes Dinámicos. Universidad Carlos III de Madrid."

Rivadeneira Molina, S. G. (2013). "METODOLOGÍAS ÁGILES ENFOCADAS AL MODELADO DE REQUERIMIENTOS."

Rojas, D. (2012). "Microsoft Developer Network: ¿qué es un unit test?".

Rosales Morales, Y. M. C., Michael Eduardo y Trujillo Oliva, Adrian (2013). "Extensión de la herramienta Visual Paradigm para la generación de clases de acceso a datos con Doctrine 2.0."

S.A.S., S. G. (2015).

Sánchez Asenjo, J. (2013). "Gestión de base de datos."

Sistemas de Gestión de Baleares, S. A. (2011).

Soluciones Corporativas AdaptaPro, C. A. (2010). from www.adaptapro.com.

Sommerville, I. (2005). "Ingeniería del Software: Séptima Edición."

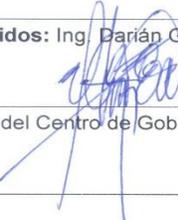
ANEXOS

ACTA DE ACEPTACIÓN

En cumplimiento del desarrollo de la Tesis: **Sistema informático para la gestión administrativa de los RRHH en CEGEL**. Se hace entrega del producto:

- Sistema informático para la gestión administrativa de los RRHH en CEGEL.

La Parte Cliente, luego de haber revisado el software, considera que la solución cumple con cada uno de los requisitos que originaron su desarrollo y constituye una ayuda a la gestión de los recursos humanos en el Centro de Gobierno Electrónico. Conforme a lo anterior se expresa la aceptación del mismo.

Entrega	Recibe
Nombre y apellidos: Gilmar Lázaro Díaz Pérez 	Nombre y apellidos: Ing. Darián González Ochoa 
Cargo: Estudiante	Cargo: Director del Centro de Gobierno Electrónico

Fecha: 08/06/2016

Figura 11: Carta de aceptación emitida por el cliente.



Acta de Liberación Interna de Productos Software

Fecha de emisión del acta: 08/06/2016

Emitida a favor de: Tesis "Sistema informático para la gestión administrativa de los RRHH en CEGEL."

Datos del producto

Artefacto	Versión	Estado final	Cantidad Iteraciones	Tipos de pruebas realizadas	Fecha de liberación
App: Sistema informático para la gestión administrativa de los RRHH en CEGEL	1.0	0	3	Evaluación dinámica Pruebas de Funcionalidad	08/06/2016


Ing. Yordanis Garcia Leiva
Asesor de Calidad CEGEL


Gilmar Lázaro Díaz Pérez
Autor



1

Figura 12: Acta de liberación emitida por el departamento de Calidad del CEGEL.

A continuación el resto de las HU.

HISTORIA DE USUARIO		
No: 07	Iteración Asignada: 1	Nombre: Adicionar usuario del sistema.
Prioridad del Negocio: Media.		Nivel de Complejidad: Medio.
Punto de Estimación: 1		Riesgo en el desarrollo: Medio.
<p>Descripción: En la interfaz usuarios del sistema se muestra un botón adicionar usuario que al hacerle clic se pasa a otra interfaz donde se introducen los datos: nombre, usuario, contraseña, confirmar contraseña y rol. Hecho esto se adiciona el usuario y se agrega al listado de usuarios que se pueden autenticar en la aplicación.</p>		
<p>Observaciones: Sólo se podrá acceder a esta funcionalidad si el usuario autenticado es “administrador”.</p>		

HISTORIA DE USUARIO		
No: 08	Iteración Asignada: 1	Nombre: Eliminar usuario del sistema.
Prioridad del Negocio: Alta.		Nivel de Complejidad: Media.
Punto de Estimación: 1		Riesgo en el desarrollo: Medio.
<p>Descripción: En la interfaz listado de usuarios se muestra en la columna acciones un botón eliminar que al hacerle clic aparece una notificación preguntando si desea eliminarlo o no, en caso afirmativo se elimina del listado y en caso negativo se regresa a la interfaz listado de usuarios.</p>		
<p>Observaciones:</p>		

HISTORIA DE USUARIO		
No: 21	Iteración Asignada: 4	Nombre: Exportar registro de bajas.

Prioridad del Negocio: Media.	Nivel de Complejidad: Media.
Punto de Estimación: 1	Riesgo en el desarrollo: Medio.
<p>Descripción: En la interfaz registro de bajas se muestra en una tabla la información (número de expediente, nombre(s) y apellidos, solapín, área, cargo, causa de la baja y fecha de la baja) asociada a las bajas de los trabajadores. Esta información es captada una vez que se le da de baja a un trabajador. Debajo de la tabla se muestra un botón exportar a pdf donde al hacer clic se muestra una ventana para guardar en la computadora el archivo registro de bajas.pdf.</p>	
Observaciones:	

HISTORIA DE USUARIO		
No: 22	Iteración Asignada: 4	Nombre: Exportar registro de certificados médicos.
Prioridad del Negocio: Media.	Nivel de Complejidad: Media.	
Punto de Estimación: 1	Riesgo en el desarrollo: Medio.	
<p>Descripción: En la interfaz registro de certificados médicos se muestra en una tabla la información (número de expediente, nombre(s) y apellidos, solapín, cargo, número de registro, desde, hasta, días naturales, días a descontar, fecha de incorporación, fecha de entrega, tipo de certificado, observaciones y visualizar) asociada a los certificados médicos de los trabajadores. Esta información es captada una vez que se le adiciona una incidencia a un trabajador con clave 27. Debajo de la tabla se muestra un botón exportar a pdf donde al hacer clic se muestra una ventana para guardar en la computadora el archivo registro de certificados médicos.pdf.</p>		
Observaciones:		

HISTORIA DE USUARIO		
No: 23	Iteración Asignada: 4	Nombre: Exportar registro de licencias sin sueldo.

Prioridad del Negocio: Media.	Nivel de Complejidad: Media.
Punto de Estimación: 1	Riesgo en el desarrollo: Medio.
<p>Descripción: En la interfaz registro de licencias sin sueldo se muestra en una tabla la información (número de expediente, nombre(s) y apellidos, solapín, área, fecha de solicitud, fecha de salida, fecha de incorporación, vacaciones acumuladas y visualizar) asociada a las licencias sin sueldo de los trabajadores. Esta información es captada una vez que se le adiciona una incidencia a un trabajador con claves 25, 29, 32 o 33. Debajo de la tabla se muestra un botón exportar a pdf donde al hacer clic se muestra una ventana para guardar en la computadora el archivo registro de certificados médicos.pdf.</p>	
Observaciones:	

HISTORIA DE USUARIO		
No: 24	Iteración Asignada: 4	Nombre: Exportar registro de licencia de maternidad.
Prioridad del Negocio: Media.	Nivel de Complejidad: Media.	
Punto de Estimación: 1	Riesgo en el desarrollo: Medio.	
<p>Descripción: En la interfaz registro de licencia de maternidad se muestra en una tabla la información (número de expediente, nombre(s) y apellidos, solapín, área, desde y hasta de prenatal, desde y hasta de primera posnatal y desde y hasta de segunda posnatal) asociada a la licencia de maternidad de las trabajadoras. Esta información es captada una vez que se le adiciona una incidencia a una trabajadora con clave 23. Debajo de la tabla se muestra un botón exportar a pdf donde al hacer clic se muestra una ventana para guardar en la computadora el archivo registro de certificados médicos.pdf.</p>		
Observaciones:		

A continuación el listado de las no conformidades detectadas por el departamento de calidad del CEGEL.

No.	No Conformidad	Ubicación de la No Conformidad	Etapas de Detección	Estado NC	Clasificación de las NC
1.	En la interfaz de inicio de sesión se debe utilizar un término más formal para guiar el usuario para registrarse en el sistema.	App/menú de logueo	1era Iteración	Abierta 06/06/2016 Cerrada 08/06/16	Redacción
2.	En el formulario para adicionar un usuario, los campos usuario y contraseña se cargan con valores predeterminados.	App/Usuario del sistema	1era Iteración	Abierta 06/06/2016 Cerrada 08/06/16	Funcionalidad
3.	En todas las interfaces que utilizan campos obligatorios, estos campos deben ser identificados con un asterisco que identifique la obligatoriedad del campo.	General en todas las interfaces del sistema	1era Iteración	Abierta 06/06/2016 Cerrada 08/06/16	Validación
4.	El sistema no da la posibilidad de modificar un usuario.	App/Usuario del sistema	1era Iteración	Abierta 06/06/2016 Cerrada 08/06/16	Funcionalidad
5.	En los mensajes de confirmación que emite el sistema al realizar las acciones, las opciones Sí y No, no deben estar entre signos de exclamación.	General en todas las interfaces del sistema	1era Iteración	Abierta 06/06/2016 Cerrada 08/06/16	Redacción
6.	En adicionar trabajador los campos deben estar validados en correspondencia con los tipos de caracteres que deben admitir.	App/Trabajadores	1era Iteración	Abierta 06/06/2016 Cerrada 08/06/16	Validación
7.	El sistema no da la posibilidad de modificar un trabajador.	App/Trabajadores	1era Iteración	Abierta 06/06/2016 Cerrada 08/06/16	Funcionalidad
8.	En la funcionalidad adicionar incidencia el campo clave no está validado, admite valores negativos.	App/Incidencias	1era Iteración	Abierta 06/06/2016 Cerrada 08/06/16	Validación
9.	En la funcionalidad adicionar incidencia el campo Descripción no está validado, admite cualquier tipo de carácter.	App/Incidencias	1era Iteración	Abierta 06/06/2016 Cerrada 08/06/16	Validación
10.	La funcionalidad registrar incidencia no se ejecuta al al	App/Registrar Incidencias	1era Iteración	Abierta 06/06/2016	Funcionalidad

No.	No Conformidad	Ubicación de la No Conformidad	Etapas de Detección	Estado NC	Clasificación de las NC
	pulsar el botón Adicionar.			Cerrada 08/06/16	
11.	El componente Calendario debe estar en español.	General en todas las interfaces del sistema que hagan uso del componente	1era Iteración	Abierta 06/06/2016 Cerrada 08/06/16	Idioma
12.	En la funcionalidad generar prenomina, el campo Confeccionado por, Validado por y Decano, no están validados, admiten cualquier tipo de caracter.	App/Prenomina	1era Iteración	Abierta 06/06/2016 Cerrada 08/06/16	Validación
13.	En la funcionalidad Registro de bajas, la columna área tiene falta de ortografía, falta la tilde.	App/Registro de bajas	1era Iteración	Abierta 06/06/2016 Cerrada 08/06/16	Ortografía
14.	En la funcionalidad Registrar incidencia, las incidencias de tipo certificado médico, el campo Evidencia no está validado y admite archivos con cualquier tipo de extensión.	App/Registrar Incidencias	1era Iteración	Abierta 06/06/2016 Cerrada 08/06/16	Validación
15.	En la funcionalidad Configurar fecha, cuando se va habilitar una fecha, el mensaje de confirmación que se muestra, es inverso a la acción.	App/Configurar fechas	1era Iteración	Abierta 06/06/2016 Cerrada 08/06/16	Redacción
16.	En la interfaz de algunas funcionalidades, el nombre tiene un punto al final.	General en todas las interfaces del sistema	1era Iteración	Abierta 06/06/2016 Cerrada 08/06/16	Redacción
17.	En la pestaña del navegador no se muestra el nombre del sistema.	General en todo el sistema	1era Iteración	Abierta 06/06/2016 Cerrada 08/06/16	Interfaz
18.	En la funcionalidad Registrar incidencia, los campos Días naturales, Doble turno y Días feriados, admiten números negativos.	App/Registrar Incidencias	2da Iteración	Abierta 08/06/2016 Cerrada 08/06/16	Validación
19.	En la funcionalidad generar prenomina, la columna Días feriados tiene un error ortográfico.	App/Prenomina	2da Iteración	Abierta 08/06/2016 Cerrada 08/06/16	Ortografía