

Universidad de las Ciencias Informáticas
Facultad 4



Módulo de Gestión de Personal para la Suite ZUN del Grupo de Electrónica para el Turismo

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

Autor:

Luis Tomás Flores Ladrón de Guevara

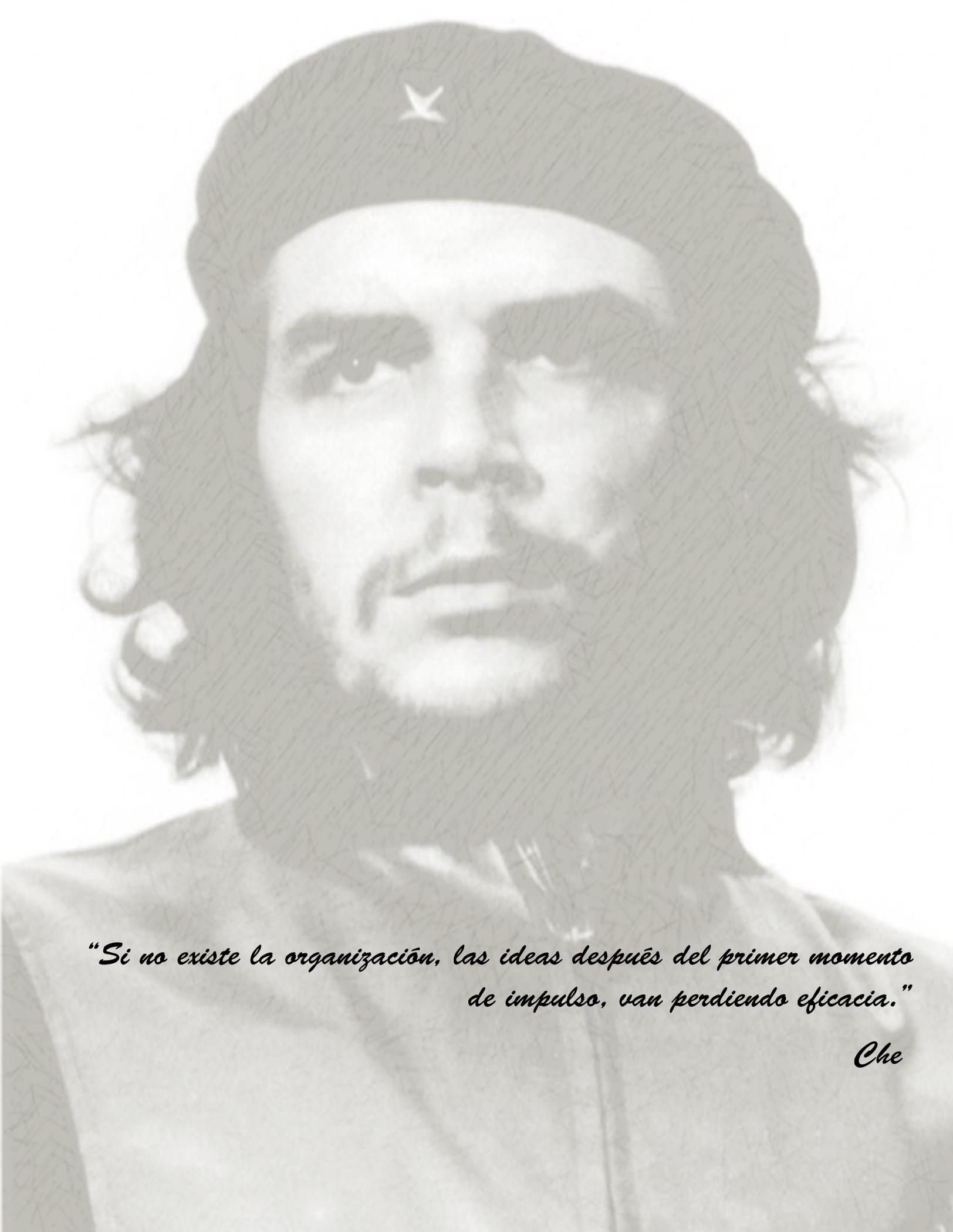
Tutor:

Ing. Adrián García Sánchez

Co-Tutor:

Ing. Lizandra la Hoz Pavón

**La Habana, 2014
“Año 56 de la Revolución”**



"Si no existe la organización, las ideas después del primer momento de impulso, van perdiendo eficacia."

Che

AGRADECIMIENTOS

A la Revolución, por darme la oportunidad de hacerme un profesional sin costo alguno.

A mi mamá, María del Carmen por estar siempre para mí, todo lo que soy es gracias a su amor, consejos, regaños y hasta "etcéteras".

A mi compañera, co-tutora y novia, Lizandra por ser la persona más bella de este mundo, por su entrega incondicional, por sus peleas (justificadas e injustificadas), por su amor que me ha hecho feliz estos últimos 4 años, este resultado es tuyo también mi amor.

A mi padre, Tomás por todo los consejos que tanta falta me han hecho siempre.

A mis hermanos, Nicolay, Lindiana y Luis Enrique (Luisito) por ser mis hermanos del alma, que aunque no hemos estado siempre juntos y tengamos nuestras diferencias, yo los quiero mucho y los tengo siempre muy presentes en mi corazón.

A mis tíos, Ángela y en especial a Luisi, que siempre me han brindado su amor y apoyo incondicional.

A mi abuela, Teresa que no pudo verme graduado y tuvo muchísimo que ver en mi educación. "Mama", te extraño y te quiero un montón, fuiste mi segunda madre.

A mi tutor Adrián que ha sabido guiarme y apoyarme.

A mis colegas de universidad, Andrés, Efraín, Frank, los Manolos, Ovidio, todo el piquete del apartamento por soportarme 5 años y a los del gym.

En fin a toda mi familia, porque este sueño es también parte de ellos.

DEDICATORIA

A mis padres, a mis abuelos que no están presentes y al resto de mi familia.

Muy especialmente a mis hermanos y primos.

A Lizandra, mi inseparable novia y luz de mis ojos.

*A todos esos inolvidables personajes que hicieron de mi paso por la universidad
una historia única.*

A todos los que no se pudieron graduar con nosotros.

RESUMEN

El avance de las tecnologías de la información y las comunicaciones ha cambiado la manera de gestionar los procesos que antes se realizaban de una forma tradicional. Actualmente son cada vez más las organizaciones empresariales que migran hacia la informatización, con el objetivo de elevar el nivel económico y tener un control eficiente sobre los recursos que manejan. Dentro del Ministerio del Turismo existe una empresa nombrada Grupo de Electrónica para el Turismo (GET), encargada de informatizar los procesos empresariales dentro del mismo, en la cual se desarrolla un sistema de gestión hotelera y extra-hotelera conocido como Suite ZUN. La presente investigación tiene como propósito el desarrollo de un módulo que gestione los recursos humanos, con un valor empresarial reutilizable e interoperable con los otros sistemas de la Suite, realizándose un estudio de cómo sistemas similares gestionan los empleados, así como de las herramientas y lenguajes de programación a emplear en el análisis, diseño, implementación y prueba del módulo.

Palabras clave: capital humano, empleados, personal, recursos humanos.

INDICE

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	5
1.1 Introducción del capítulo	5
1.2 Conceptos asociados al dominio del problema	5
1.3 Sistemas similares	6
1.4 Metodologías de desarrollo de software	8
1.5 Lenguaje de modelado.....	14
1.6 Herramientas CASE para el modelado UML	14
1.7 Lenguajes y herramientas de desarrollo.....	16
1.7.1 Lenguaje de desarrollo.....	16
1.7.2 Gestor de bases de datos	17
1.7.3 Entorno de desarrollo integrado	18
1.8 Conclusiones parciales del capítulo	21
CAPÍTULO 2: PROPUESTA DE SOLUCIÓN	22
2.1 Introducción	22
2.2 Modelo de dominio.....	22
2.2.1 Análisis de los conceptos del dominio.....	22
2.2.2 Diagrama del modelo de dominio.....	24
2.3 Descripción del sistema propuesto.....	26
2.4 Modelos de casos de uso del sistema.....	34
2.4.1 Descripción del actor del sistema.....	36
2.4.2 Diagramas de casos de uso del sistema	36
2.4.3 Descripción de los casos de uso del sistema	39
2.5 Modelo de análisis	48
2.5.1 Diagrama de clases del análisis	49
2.5.2 Diagrama de colaboración del análisis	50
2.6 Modelo de diseño.....	51
2.6.1 Patrón de diseño.....	51
2.6.2 Diagrama de clases del diseño	53
2.7 Diseño de la base de datos.....	55
2.7.1 Descripción de las entidades relevantes	57

2.8	Diagrama de despliegue	58
2.9	Conclusiones parciales del capítulo	59
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS		60
3.1	Introducción	60
3.2	Modelo de implementación	60
3.3	Diagrama de componentes	60
3.4	Estándares de código	61
3.5	Pruebas de software	63
3.6	Resultados obtenidos.....	65
3.7	Conclusiones parciales	66
CONCLUSIONES		67
RECOMENDACIONES		68
REFERENCIAS BIBLIOGRÁFICAS		69
GLOSARIO		72
ANEXOS		75

INTRODUCCIÓN

La emergente sociedad de la información, impulsada por un vertiginoso avance científico en un marco socioeconómico neoliberal-globalizador y sustentado por el uso generalizado de las potentes y versátiles Tecnologías de la Información y las Comunicaciones (TIC), conlleva cambios que alcanzan todos los ámbitos de la actividad humana. Sus efectos, se manifiestan de manera muy especial en las actividades laborales. El uso de estas tecnologías puede servir de apoyo para llevar a cabo procesos que antes se realizaban de forma manual, lo que los hacía más engorrosos, lentos e inseguros. Tal es el caso de la mayoría de los Departamentos de Recursos Humanos, que generan numerosos y variados reportes, sin embargo toda la información referente a personal se encuentra en hojas de cálculos de Excel.

El Grupo de Electrónica para el Turismo (GET) es una organización empresarial perteneciente al Ministerio del Turismo (MINTUR), especializada en brindar soluciones informáticas integradoras. En el mismo, se desarrolla un Sistema de Gestión Hotelera y Extra-hotelera conocido como Suite ZUN, el cual está integrado por diversos módulos fácilmente adaptables a cualquier entidad, lo que permite mantener un flujo transparente y consistente de toda la información necesaria entre ellos.

Esta Suite carece de un sistema informático para la gestión integral de los recursos humanos, lo que representa una gran prioridad dentro del MINTUR ya que una inmensa mayoría de sus instalaciones no cuentan con un sistema integral para la gestión y control del personal, dentro de estas el GET. La existencia de un sistema de gestión personalizado y desarrollado a la medida para dicha empresa es muy importante y necesaria, de esta manera se resolverían problemas como: la integridad de la información referente al personal; la generación manual de reportes desde un documento informático en formato Excel que almacena toda la información de personal y la base de datos de la nómina; la automatización y, por ende, la reducción del tiempo que se dedicará en la realización de los procesos referentes a personal en el Departamento de Recursos Humanos de la entidad. Para el caso del procesamiento de las nóminas contaban con diferentes sistemas no certificados ni con la necesaria seguridad y protección de la información, toda vez que los mismos estaban desarrollados para MS-DOS, en los lenguajes de programación Clipper y FoxPro, y las bases de datos eran ficheros con extensión .DBF. Algunos de estos sistemas en

explotación no estaban actualizados de acuerdo con las legislaciones vigentes, lo que se tornaba cada vez más costoso y difícil realizarles el mantenimiento, convirtiéndose en sistemas obsoletos; la necesidad de migrar hacia nuevas tecnologías se hacía cada vez más inminente.

En el proceso de captura de requisitos realizado en la entidad, por sus necesidades y prioridades, y además por la relación del módulo de nóminas en el sistema contable de la empresa, se desarrolló en la primera etapa solo este módulo, quedando el de Gestión del Personal para una etapa posterior.

Al encontrarse en la fase de despliegue en las instituciones del MINTUR, el proceso de nómina certificada se hace imprescindible tener un control exhaustivo del personal, por lo que es necesario el desarrollo del Módulo Gestión de Personal.

De acuerdo con la situación anteriormente descrita, se define el **problema de la investigación** en la siguiente interrogante: ¿Cómo contribuir a agilizar el proceso de gestión de personal dentro de la Suite ZUN?

En consecuencia con lo anterior se determinó como **objeto de estudio** los procesos de gestión de la información vinculada a los recursos humanos.

Enmarcando el **campo de acción** sobre los procesos de gestión de la información vinculada a recursos humanos referentes a personal dentro de la Suite ZUN.

Se declara como **objetivo general** desarrollar un módulo que permita agilizar el proceso de gestión del personal dentro la Suite ZUN integrable al sistema de nóminas ZUNpr.

A partir del objetivo general definido se derivan los siguientes **objetivos específicos**:

- ✓ Construir el marco teórico referencial de la investigación, relacionado con los procesos de gestión de la información vinculada a los recursos humanos.
- ✓ Realizar el análisis y diseño del sistema de gestión del personal.
- ✓ Desarrollar el sistema informático para la gestión del personal teniendo en cuenta el análisis y diseño realizado e integrarlo a la Suite ZUN.
- ✓ Validar mediante pruebas de *software* el funcionamiento del sistema.

Para guiar la investigación se plantea la siguiente **idea a defender**: Si se desarrolla un módulo que facilite el proceso de gestión de personal dentro de la Suite ZUN integrable al sistema de Nóminas, entonces se logrará agilizar el proceso de gestión de la

información inherente a cada empleado, permitiendo la obtención de diversos reportes estadísticos en menos tiempo y con mayor calidad.

Para dar cumplimiento a los objetivos específicos propuestos, se trazaron las siguientes **tareas de investigación:**

- Revisión bibliográfica sobre los sistemas informáticos similares que gestionen Recursos Humanos.
- Estudio del estado del arte del proceso actual de gestión de recursos humanos.
- Investigación sobre las herramientas y metodologías existentes para realizar el análisis, diseño e implementación del sistema.
- Desarrollo del sistema de acuerdo con las herramientas y metodologías escogidas.
- Aplicación de pruebas funcionales al sistema.

Para darle respuesta al problema a resolver y cumplimiento al objetivo general planteado, fueron aplicados los siguientes **métodos científicos** de investigación:

Métodos teóricos

Analítico-sintético: Posibilitó el estudio de fuentes bibliográficas referentes a los procesos de gestión de la información vinculada a los recursos humanos, identificando elementos importantes y necesarios para dar solución al problema planteado.

Métodos Empíricos

Entrevista: Se aplicó al cliente que será beneficiado con el sistema, con el objetivo de identificar sus necesidades y determinar las principales funcionalidades de la aplicación. Algunas de las preguntas que sirvieron de guía se muestran en el [anexo 1](#).

Esta investigación se encuentra estructurada en tres capítulos, ellos son:

Capítulo 1. Fundamentación teórica: presenta el marco teórico referencial relacionado con los procesos de gestión de la información vinculada a los recursos humanos. También se describen herramientas y metodologías a utilizar para el desarrollo del sistema.

Capítulo 2. Propuesta de solución: plasma un análisis de la gestión de información de los recursos humanos dentro del GET, se describe el sistema informático a desarrollar, así como sus funcionalidades, se detalla la estructura de la arquitectura empleada, se define el patrón de diseño, así como el modelo de entidades de datos.

Capítulo 3. Implementación y prueba: muestra las tareas de implementación, donde se detallan los estándares de programación empleados. También se especifican las diferentes pruebas aplicadas al sistema informático desarrollado.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción del capítulo

El empleo de herramientas informáticas aumenta la productividad y competitividad de los proyectos de desarrollo de *software*. En este sentido, la incorporación de nuevas tecnologías a los procesos de producción constituye un desafío. La selección de las mismas requiere de un estudio detallado y crítico, teniendo en cuenta las necesidades del cliente y los recursos disponibles durante el proceso de desarrollo de *software*.

En el presente capítulo se abordan conceptos referentes a gestión de recursos humanos y capital humano. Se analizan diferentes sistemas de gestión de recursos humanos existentes en Cuba y en el mundo, así como la metodología de desarrollo de *software*, herramientas y tecnologías a utilizar para generar la nueva aplicación que se propone.

1.2 Conceptos asociados al dominio del problema

Recursos Humanos:

“Conocido también como Potencial Humano, Capital Humano o Activo Humano, hace referencia al conjunto de trabajadores, empleados o personal que conforman un negocio o empresa.” (Crece Negocios, 2014)

“El término recursos humanos (abreviado como RRHH, RH, RR.HH., y también conocido como capital humano) se originó en el área de economía política y ciencias sociales, donde se utilizaba para identificar a uno de los tres factores de producción, también conocido como trabajo (los otros dos son tierra y capital). Como tales, durante muchos años se consideraba como un recurso más: predecible y poco diferenciable.” (Meltom Technologies, 2013)

En resumen, los recursos humanos no son más que el conjunto de trabajadores o empleados que forman parte de una institución o empresa. Son concebidos como la fuente de riqueza más importante, ya que son los responsables de llevar a cabo la ejecución y el desarrollo de las tareas y actividades necesarias.

Nómina:

“Documento en el que consta el sueldo que una persona recibe regularmente de una empresa y todas las operaciones relacionadas con él.” (WordReference.com, 2014)

“Una nómina es el registro financiero que una empresa realiza sobre los salarios de sus empleados, bonificaciones y deducciones.” (e-conomic, 2014)

En síntesis, las nóminas no son más que los documentos o instrumento que permite realizar de una forma ordenada, el pago a los trabajadores, así como proporcionar la información contable y estadística necesaria, tanto para la misma empresa como para el organismo regulador encargado de estos procesos de pago.

1.3 Sistemas similares

A continuación se muestra el estudio del estado del arte realizado respecto a algunos de los sistemas informáticos que existen, enfocados a informatizar los procesos presentes en la gestión de Recursos Humanos.

ASSETS NS es un Sistema privativo de Gestión Integral estándar y parametrizado desarrollado con la arquitectura cliente-servidor, programada en Visual Basic 6.0 y Microsoft SQL Server 2000, además utiliza adicionalmente el Crystal Reports 7.0 para la generación de reportes. Este sistema cuenta con varios módulos entre los que se encuentran el de control de los procesos de Compras, Ventas, Producción, Taller, Inventario, Finanzas, Contabilidad, Presupuesto, Activos Fijos, Útiles y Herramientas y Recursos Humanos.

“El módulo de Recursos Humanos posibilita el control íntegro de los empleados y la estructura organizativa de una entidad. Asimismo, proporciona opciones de seguridad que le permiten limitar el acceso a los diferentes procesos del sistema de acuerdo con el perfil que tenga cada usuario.” (D´MARCO S.A., 2013)

En este sistema todos sus módulos trabajan en estrecha relación, generando de forma automática los comprobantes de las operaciones realizadas al módulo de la contabilidad.

Pese a las características anteriormente mencionadas, por ser **ASSETS NS** una herramienta diseñada para ser multi-compañía, o sea, muy general en muchos aspectos, no cumple con todas las necesidades del GET, ya que en el organismo al que pertenece dicha empresa existen formas de pagos diferentes y leyes que solo están vigentes dentro del mismo. Además, por el estrecho vínculo que debe haber entre la

gestión del personal y el ZUNpr, y por la cantidad de información en común que se maneja, los dos módulos deben trabajar sobre una misma base de datos; solo así se garantiza la integración con el sistema. Por otro lado, esta herramienta es privativa y extranjera, lo que demanda la compra de los diferentes tipos de licencias para su uso.

RH Expert es un sistema de gestión integral de los recursos humanos 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.

“El sistema está diseñado sobre la base de una estructura jerárquica de entidades que permite el envío de la información de los Recursos Humanos. Las entidades subordinadas envían su información a la entidad superior donde se almacena, procesa y consolida.

El sistema prevé un entorno de trabajo multi-usuario, es desarrollado en Delphi 7 y utiliza como sistema gestor de base de datos SQL Server 2000. Posee una interfaz muy intuitiva y utiliza herramientas sencillas para minimizar la entrada de datos.

RH Expert procesa la información de forma muy eficiente y genera una gran variedad de reportes, que pueden ser diseñados por el propio usuario según sus necesidades. Es una solución adecuada para cualquier empresa moderna, ya que constituye un sistema ágil y preciso para el manejo de los recursos humanos en la actualidad.” (Datazucar, 2014)

Este parte de la necesidad de tener un control centralizado de la información de los recursos humanos, debido a la gran cantidad de información que se tiene que recopilar, organizar y procesar para obtener un consolidado en las empresas del nivel superior.

Si bien el sistema brinda las opciones de evaluar el desempeño laboral de los trabajadores, controlar integralmente al personal, analizar y desarrollar el potencial directivo y controlar las necesidades de capacitación de la organización, su estructura jerárquica de entidades no se adecua a las necesidades del GET, ya que no existen entidades subordinadas, ni flujo de información hacia ninguna entidad superior. Además, como se explica anteriormente, en el organismo al que pertenece dicha empresa existen formas de pagos diferentes y leyes que solo están vigentes dentro del mismo.

Dado al estrecho vínculo que debe existir entre los módulos *gestión del personal* y el de *nóminas (ZUNpr)*, es necesario trabajar sobre una misma base de datos ya que se maneja mucha información en común, solo así se lograría la integración de estos dos sistemas, por lo que no es posible con el RH Expert, que es, además, un *software* privativo.

OrangeHRM es una solución integral, dirigida a pequeñas y medianas empresas ya que es efectiva y de bajo costo para la gestión de recursos humanos. Este sistema presenta una pequeña variedad de funciones para gestionar capital humano. Está desarrollada con el precepto de *software* libre. El lenguaje de programación empleado es Preprocesador de Hipertexto (PHP), como gestor de base de datos *MySQL* y utiliza *Symfony* como framework de desarrollo.

Presenta una serie de módulos, todos en una sola aplicación, que cumple con las necesidades de recursos humanos más importantes. (OrangeHRM, 2013)

Dentro de los módulos que brinda están:

- Módulo de administración
- Gestión de información personal
- Agregar módulo
- Módulo de tiempo
- Módulo de autoservicio del empleado
- Módulo de reclutamiento y seguimiento de candidatos
- Módulo de rendimiento

Aunque el sistema es un *software* libre, no cumple con todas las necesidades del GET, ya que, como se explica anteriormente, en el organismo al que pertenece dicha empresa existen formas de pagos diferentes y leyes que solo están vigentes dentro del mismo, además por el estrecho vínculo que debe haber entre la gestión del personal y el ZUNpr los dos módulos deben trabajar sobre una misma base de datos ya que se maneja mucha información en común; solo así se garantiza la integración con el sistema.

1.4 Metodologías de desarrollo de software

Una metodología es una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de *software* en sus esfuerzos

por implementar nuevos sistemas de información. Una metodología está formada por fases, cada una de las cuales se puede dividir en sub-fases, que guiarán a los desarrolladores de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo. (Tinoco Gómez, 2013)

Las metodologías de desarrollo de *software* se clasifican generalmente en dos grupos, en ágiles, dentro de las más usadas para guiar el proceso de desarrollo se encuentran la Programación Extrema y SCRUM, y en robustas como son el caso de Proceso Unificado de Rational (RUP por sus siglas en inglés) e ICONIX.

RUP

Es un proceso para el desarrollo de un proyecto de *software* que se encarga de unificar a todo el equipo de desarrollo, definiendo claramente quién, cómo, cuándo y qué debe hacerse en el proyecto. Más que un proceso, es un marco de trabajo extensible que puede ser adaptado a organizaciones o proyectos específicos. Por lo general se aplica a grandes proyectos de desarrollo de *software*. Esta aprovecha las mejores funciones y características de los modelos tradicionales de desarrollo, y además se caracteriza por implementar los mejores principios de los modelos ágiles.

Esta metodología reconoce la importancia de la comunicación con el cliente y los métodos simplificados para la descripción de la vista del cliente de un sistema, el Caso de Uso (CU). Sugiere un flujo de proceso que es iterativo e incremental, dirigido por casos de uso y centrado en la arquitectura, proporcionando un ambiente evolutivo que es esencial en el desarrollo de *software* moderno. (S. Pressman, 2010)

El flujo de trabajo propuesto por esta metodología se comprende en las siguientes 5 fases:



Figura 1. Representación del flujo de trabajo de RUP.

La fase de **inicio** abarca las actividades de comunicación con el cliente y la de planificación. Al colaborar con los interesados, se identifican los requisitos de negocio para el *software*, lo que posibilita proponer una arquitectura aproximada para el sistema. Estos requisitos se describen a través de un conjunto de casos de uso preliminares que describen las características y funciones. La arquitectura en este momento no es más que un esquema provisional que en las siguientes fases se perfeccionará y se expandirá a un conjunto de modelos que representarán las diferentes vistas del sistema. En esta fase se identifican los recursos, se evalúan los riesgos principales y se define una planificación que servirá de base para la fase posterior.

La fase de **elaboración** contiene las actividades de planificación y de modelación del proceso de desarrollo. Aquí se refinan y amplían los casos de uso preliminares que se han desarrollado como parte de la fase inicial y se expande la representación arquitectónica generando 5 artefactos fundamentales: el modelo de casos de uso, el modelo de requisitos, el modelo de diseño, el modelo de implementación y el modelo de despliegue. En algunos casos, la elaboración crea una línea de base arquitectónica ejecutable que representa un primer corte del sistema.

En la culminación de esta fase, el plan se revisa cuidadosamente para asegurar que el alcance, los riesgos y las fechas de entrega siguen siendo razonables. Las modificaciones en el plan a menudo se hacen en este momento.

La fase de **construcción** coincide con la actividad de construcción del proceso de desarrollo. Utilizando el modelo arquitectónico como entrada, se desarrollan los componentes de *software* que harán que cada caso de uso sea operativo para los usuarios finales. Para lograr esto, los requisitos y los modelos de diseño que se iniciaron durante la fase de elaboración tienen que haberse completado para que todas las características y funciones necesarias y requeridas, sean implementadas en código fuente, para así reflejar la versión final del incremento del *software*. Al implementar estos componentes se diseñan y ejecutan un conjunto de pruebas unitarias. Además, se llevan a cabo actividades de integración (ensamblaje de componentes y pruebas de integración).

Los casos de uso se utilizan para derivar un conjunto de pruebas de aceptación que se ejecutan antes del inicio de la siguiente fase.

La fase de **transición** abarca las actividades de la construcción y la primera parte de la implementación (entrega y retroalimentación). El *software* se les da a los usuarios para realizarle pruebas beta y reportes de retroalimentación, tanto de defectos como cambios necesarios a realizar. Además, el equipo de *software* crea la información de soporte necesaria (por ejemplo, manuales de usuario, guías de solución de problemas, procedimientos de instalación) que se requiere para la liberación. En la conclusión de la fase de transición, el incremento *software* se convierte en una versión de *software* utilizable.

La fase de **producción** coincide con la actividad de implementación del proceso de desarrollo. Durante esta fase, se vigila el uso continuo del *software*, se proporciona soporte para el entorno operativo, y los reportes de defectos y solicitudes de cambio se presentan y evalúan.

ICONIX

“Es un proceso simplificado en comparación con otros procesos más tradicionales, que unifica un conjunto de métodos de orientación a objetos con el objetivo de abarcar todo el ciclo de vida de un proyecto.” (Hernández Torruco, et al., 2010)

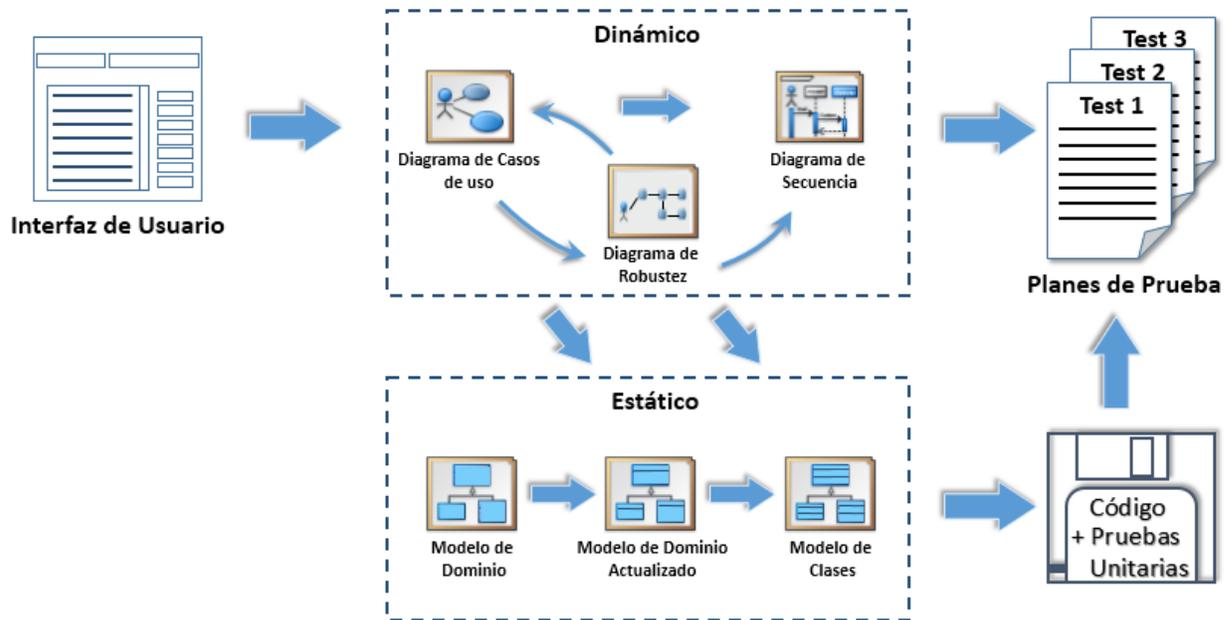


Figura 2. Representación del flujo de trabajo de ICONIX.

Según Rosenberg y Scott en su libro *“Modelado de Objetos dirigido por Casos de Uso con UML: Un enfoque práctico”*, esta metodología consta de 4 etapas que incluyen las siguientes actividades:

Etapas de análisis de requisitos

- Se identifican del “mundo real” los objetos y todas las relaciones de agregación y generalización entre ellos. Para reflejar esto se utiliza un diagrama de clases definido como modelo de dominio.
- Se presenta, si es posible, un prototipo rápido y sencillo de las interfaces del sistema y los diagramas de navegación, de forma que los clientes puedan comprender mejor el sistema propuesto.
- Se identifican los casos de uso del sistema mostrando los actores involucrados mediante la utilización del diagrama de casos de uso.
- Se organizan los casos de uso en grupos, a través de los diagramas de paquetes.

Etapas de análisis y diseño preliminar

- Se describen los casos de uso pudiendo contener los flujos alternativos y los flujos de excepción. Una sugerencia de esta metodología, en esta actividad, es

que se debería usar un estilo consistente que sea adecuado al contexto del proyecto, para no perder mucho tiempo con la descripción textual.

- Se realiza un diagrama de robustez, para ilustrar gráficamente las interacciones entre los objetos participantes de un caso de uso, lo que permite analizar el texto narrativo e identificar un conjunto inicial de objetos participantes de cada caso de uso.

Etapas de diseño detallado

- Se especifica el comportamiento a través del diagrama de secuencia, permitiendo identificar los mensajes entre los diferentes objetos de cada caso de uso.
- Se termina el modelo estático, adicionando los detalles del diseño en el diagrama de clases.
- Se verifica si el diseño satisface todos los requisitos identificados.

Etapas de implementación

- Si fuera necesario se realiza el diagrama de componentes para apoyar el desarrollo, el cual permitirá mostrar la distribución física de los elementos que componen la estructura interna del sistema.
- Se genera el código.
- Se realizan pruebas tales como: Test de unidades; test de casos de uso; test de datos; test de resultados y test de integración con los usuarios para verificar la aceptación de los resultados.

Selección de la metodología

Dado a que ICONIX es una metodología basada en el Proceso Unificado, este es iterativo, incremental, trazable y hace uso del modelado UML (Lenguaje Unificado de Modelado). La principal desventaja de su uso para guiar el desarrollo del sistema propuesto es que esta metodología utiliza solo un subconjunto de diagramas, desechando los diagramas de clases del análisis, colaboración del análisis, clases del diseño, entidad-relación y el de despliegue, siendo estos, parte de los artefactos solicitados por la entidad. Por tanto se selecciona RUP como metodología para guiar el proceso de desarrollo porque es adaptable a las necesidades de la organización y del proyecto, además de ser del dominio del desarrollador. En esencia, esta metodología tiene bien definido y organizado el proceso de desarrollo de *software*, lo que permite

que los involucrados tengan claridad de los pasos a seguir para obtener un producto con calidad. Con su utilización se generarán los artefactos necesarios para el cliente, ya que el módulo que se propone posee un gran alcance y el desarrollador tiene que desempeñar otros roles para realizar las actividades y crear los artefactos necesarios en un corto tiempo.

1.5 Lenguaje de modelado

El lenguaje de modelado para llevar a cabo la representación del sistema es el UML en su versión 2.1. UML es un lenguaje de modelado visual que se emplea en la especificación, visualización, construcción y documentación de artefactos de un sistema. Este captura el conocimiento y las decisiones sobre el sistema que se desea construir. Es usado para entender, diseñar, hojear, configurar, mantener, y controlar la información. Este lenguaje puede usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios, al mismo tiempo pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. (Rumbaugh, et al., 1999)

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- Visualizar: Permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- Especificar: Permite especificar cuáles son las características de un sistema antes de su construcción.
- Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.
- Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden valer para su futura revisión.

1.6 Herramientas CASE para el modelado UML

Existen varias herramientas CASE para el modelado visual de sistemas mediante UML, entre las que se encuentran:

Enterprise Architect es una herramienta UML que abarca todo el ciclo de vida del desarrollo del *software*, provee una estructura competitiva en el modelado del negocio,

diseño del *software*, ingeniería del sistema, arquitectura de la empresa, gestión de los requisitos y prueba.

Permite integrar al equipo de desarrollo proporcionando una visión de trabajo compartido. Enterprise Architect es una herramienta multiusuario, basada en Windows, diseñada para ayudar a construir *software* robusto, fácil de mantener, fácil de usar, rápida y flexible. Ofrece salida de documentación flexible y de alta calidad. Provee una trazabilidad completa desde el análisis de requerimientos hasta los artefactos de análisis y diseño, a través de la implementación y el despliegue. Las bases de Enterprise Architect están construidas sobre la especificación de UML 2.0. (Sparx, 2014)

Rational Rose proporciona un desarrollo iterativo que permite cubrir todo el ciclo de vida de un proyecto, en el que se puede especificar, analizar, diseñar un sistema antes de codificarlo; ayudando a establecer una trazabilidad real entre el modelo (análisis y diseño) y el código ejecutable. Además, permite generar código a partir de los modelos y realizar ingeniería inversa (crear modelo a partir de código).

Sin embargo, cuando estas características la convierten en una herramienta muy utilizada actualmente, su particularidad de ser propietaria (no gratuita, restricciones de licencia, código fuente) la hacen ser rechazada por muchas empresas y desarrolladores de *software*. (González Blanco, 2014)

Visual Paradigm es una herramienta de modelado profesional que soporta el ciclo de vida completo del desarrollo de *software*: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas que sigue el estándar de UML (Diagramas de Clase, Casos de Uso, Comunicación, Secuencia, Estado, Actividad, Componentes). También proporciona abundantes tutoriales, demostraciones interactivas y proyectos UML. Presenta licencia gratuita y comercial, es fácil de instalar y actualizar, compatible entre ediciones y admite ingeniería de ida y vuelta. Además, permite generación de código, documentación y de bases de datos. (Visual Paradigm, 2014)

Selección de la herramienta CASE de modelado UML

Acorde con la metodología de desarrollo y el lenguaje de modelado seleccionados, se optó por utilizar como herramienta CASE el Visual Paradigm para UML. Esta promueve el desarrollo de componentes iterativos e incrementales, e incluye funcionalidades para

la generación automática de código a partir de los modelos, asumiendo a UML como lenguaje de modelado.

1.7 Lenguajes y herramientas de desarrollo

1.7.1 Lenguaje de desarrollo

Un lenguaje de programación es un lenguaje artificial que permite establecer una comunicación con una máquina, especialmente una computadora. Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. (Mi Tecnológico, 2014)

Dentro de la informática, es aquel elemento que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y *software* existentes. (Mi Tecnológico, 2014)

Algunos ejemplos de lenguajes de programación son: ASP¹, PHP², JAVA, Visual Basic, C#, entre otros. Para la creación de aplicaciones de escritorio los más usados y recomendados son C# y JAVA.

El C# es un lenguaje de programación orientado a objetos. Con él se quiso mejorar con respecto a C y C++, de los cuales se deriva, además de incorporarle las ventajas o mejoras de JAVA. Así se consiguió un nuevo lenguaje con las ventajas del C y del C++, pero además con la productividad del lenguaje JAVA, y se le denominó C#.

Algunas de las características del lenguaje de programación C# son: su código se puede tratar íntegramente como un objeto; su sintaxis es muy similar a la del JAVA; es un lenguaje orientado a objetos y a componentes; armoniza la productividad del Visual Basic con el poder y la flexibilidad del C++; se ahorra tiempo en la programación ya que tiene una librería de clases muy completa y bien diseñada.

A pesar de que el lenguaje C# forma parte de la plataforma .NET³, que es una interfaz de programación de aplicaciones, es un lenguaje independiente que originariamente se creó para producir programas sobre esta plataforma.

¹ Es un modelo de desarrollo Web unificado que incluye los servicios necesarios para crear aplicaciones Web empresariales con el código mínimo. ASP forma parte de .NET.

² Es un lenguaje de código abierto muy popular, especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

³ Es una plataforma de desarrollo y ejecución de aplicaciones. Brinda todas las herramientas y servicios que se necesitan para desarrollar modernas aplicaciones empresariales y de misión crítica, además nos provee de mecanismos robustos, seguros y eficientes para asegurar que la ejecución de las mismas sea óptima.

Esta plataforma fue creada entre otras razones porque anteriormente el Visual Basic era uno de los lenguajes de programación que se encargaban de desarrollar aplicaciones. Este es un lenguaje orientado a objetos pero algo pobre, porque se quiso que fuese fácil de aprender para los programadores novatos.

El Visual Basic no tiene algunas de las características necesarias como la herencia, los métodos virtuales, la sobrecarga de operadores, etc. Que se han conseguido con el C# y la plataforma .NET. (La Revista Informática, 2014)

La selección del lenguaje de programación no radica a nivel de lenguaje, porque C# y Java son prácticamente iguales, ambos son lenguajes completamente orientados a objetos, son multiplataforma a nivel binario y código fuente. Se escoge el lenguaje de programación C# porque el módulo tiene que integrarse con el de nóminas y demás aplicaciones de la Suite ZUN desarrolladas en este mismo lenguaje; que es uno de los requisitos de la empresa.

1.7.2 Gestor de bases de datos

Un Sistema Gestor de Bases de Datos (SGBD por sus siglas en español) “*es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Permiten definir, manipular y utilizar la información que contienen las bases de datos, realizar todas las tareas de administración necesarias para mantenerlas operativas, garantizando mantener su integridad, confidencialidad y seguridad.*” (Murillo Alfaro, 2014)

Microsoft SQL Server es un sistema para la gestión de bases de datos relacionales (SGBDR por sus siglas en español) producido por Microsoft. Su lenguaje para consultas es T-SQL⁴, pero también es compatible con ANSI⁵ SQL, que es el lenguaje SQL estándar.

Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle, MySQL, Sybase ASE y PostgreSQL.

Dentro de las características que presenta Microsoft SQL Server podemos encontrar:

- Brinda soporte de transacciones.

⁴ T-SQL o SQL Transaccional, es el principal medio de programación y administración de SQL Server. Expone las palabras clave para las operaciones que pueden realizarse en SQL Server.

⁵ Instituto Nacional Americano de Estándares es una organización sin ánimo de lucro que supervisa el desarrollo de estándares para productos, servicios, procesos y sistemas en los Estados Unidos.

- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL⁶ y DML⁷ gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos.
(Techopedia, 2014)

La herramienta de la interfaz principal de SQL Server es el *SQL Server Management Studio*, es compatible con entornos de 32 bits y 64 bits, cuenta con un entorno amigable y presenta un conjunto importante de funcionalidades que le facilitan la vida al desarrollador.

Se decide escoger como SGBD el Microsoft SQL Server 2005, dado que este presenta cualidades suficientes para realizar trabajos de gran escala, brindando facilidades que van desde la interfaz gráfica hasta el código, permitiendo el autocompletado del mismo. Además, al ser C# el lenguaje de programación seleccionado, se dificulta un tanto la conexión con otros gestores como es el caso de PostgreSQL ya que el mismo es *software* libre, por tanto el proveedor de datos para .NET llamado Npgsql⁸, no está dentro de los proveedores de datos predefinidos en .NET. Por otra parte, la base de datos sobre la que trabajará el sistema está desarrollada en SQL Server 2005.

1.7.3 Entorno de desarrollo integrado

Un Entorno de Desarrollo Integrado (IDE) es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o poder utilizarse para varios. Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, PHP, Python, Java, C#, Delphi, Visual Basic, etc. Un IDE está compuesto por:

- Un editor de texto
- Un compilador

⁶ Lenguaje de definición de datos que permite definir y describir los objetos de la base de datos.

⁷ Lenguaje de manipulación de datos que permite generar consultas.

⁸ Es un proveedor de datos de .Net para PostgreSQL. Permite acceder a la base de datos del servidor a cualquier programa desarrollado en .Net. Se lleva a cabo en el 100% del código C#. Funciona con Postgresql 7.x y superior.

- Un intérprete
- Un depurador
- Un cliente
- Posibilidad de ofrecer un sistema de control de versiones.
- Factibilidad para ayuda en la construcción de interfaces gráficas de usuario.
(unillanosGSW, 2014)

Los IDE ofrecen un marco de trabajo amigable para un gran número de lenguajes de programación. Para algunos de estos, un IDE puede funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto, como es el caso de *Smalltalk*.

IDE (Integrated Development Environment o entorno de desarrollo integrado). Para Ruiz, y es la definición que se asume en esta investigación, un entorno de desarrollo integrado tiene como objetivo integrar en una misma aplicación las herramientas necesarias para realizar todo el proceso de programación: edición, compilación, revisión de errores, depuración y ejecución. El resultado es que utilizando un IDE, compilar y ejecutar un programa requiere únicamente pulsar un botón. (Ruiz, 2009)

Microsoft Visual Studio es un IDE para sistemas operativos Windows con un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros. Asimismo, dichos lenguajes utilizan las funciones de .NET Framework, las cuales ofrecen acceso a tecnologías clave para simplificar el desarrollo de aplicaciones web ASP y Servicios Web XML. (Microsoft, 2010)

Otras de las características que brinda este IDE son:

- Permitir a los desarrolladores crear servicios web en cualquier entorno que soporte la plataforma .NET.
- Crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles.

Para aquellos equipos que trabajen en varias plataformas, Visual Studio proporciona un entorno de colaboración flexible que permite conectar con otras herramientas de desarrollo, como Eclipse y Xcode.

MonoDevelop es un entorno de desarrollo integrado libre y gratuito, diseñado primordialmente para C#, desarrollado para Unix/Linux, entre los que se incluye también al Mac OSX. Permite la creación de proyectos diversos, ya sean simples o complejos, incluye plantillas y permite compilar diversos lenguajes, entre los que se encuentran el C++, Visual Basic, entre otros. (MicroTecnologías, 2009)

Al igual que Visual Studio, dispone de una función de autocompletado que es muy útil a la hora de escribir el código, aunque no dispone de funciones de depuración como la de establecer puntos de ruptura o ejecutar paso a paso, por lo que en algunos casos encontrar un fallo específico puede ser muy complicado.

Incluye tanto las herramientas de desarrollo como la infraestructura necesaria para .NET de cliente y servidor. Está posicionado para convertirse en la opción principal para el desarrollo de aplicaciones de Linux.

Este IDE no depende de la Windows Forms que es una librería privativa, en cambio utiliza la librería GTK#⁹, la cual es equivalente a la Windows Forms. Las aplicaciones construidas usando Gtk# se pueden ejecutar en muchas plataformas, incluyendo Linux, Windows y MacOS X. Gtk# es nativa para herramientas de escritorio en Linux ejecutando GNOME. (Cases, 2014)

Selección del entorno integrado de desarrollo

Aunque el MonoDevelop posee valiosas características y ventajas como es el hecho de compilar diversos lenguajes y ser *software* libre, no dispone de funciones de depuración siendo estas de mucha utilidad a la hora de encontrar errores en el código, y es desarrollado para Linux. Por lo que se decide seleccionar el Microsoft Visual Studio como entorno integrado de desarrollo ya que el sistema a desarrollar se ejecutará en Windows. Este IDE al igual que el MonoDevelop soporta varios lenguajes de programación y autocompleta código, pero presenta otras características relevantes como la función de depuración que permite establecer puntos de ruptura o ejecutar paso a paso la aplicación, además de contener una paleta de componentes muy importantes y de gran utilidad como es el DevExpress, lo que facilita el desarrollo de aplicaciones complejas que requieren de alta calidad.

⁹ Es un conjunto de herramientas de interfaz gráfica de usuario para MonoDevelop y .Net. El proyecto une el conjunto de herramientas gtk+ y varias librerías de GNOME, permitiendo el desarrollo de aplicaciones gráfica Gnome totalmente nativas usando el MonoDevelop y los marcos de desarrollo .Net.

1.8 Conclusiones parciales del capítulo

El estudio del estado del arte sobre herramientas que gestionan recursos humanos existentes en Cuba y en el resto del mundo con características similares al que se quiere desarrollar, arrojó un punto de partida a la solución, definiendo características y métodos a utilizar; se determinó que no deben faltar en la solución características como: generación automática y personalizada de reportes; gestión de los catálogos de nombres de carácter técnico, más conocidos por nomencladores; y opciones de seguridad que permitan limitar el acceso al sistema. Luego de la previa valoración de las ventajas y desventajas, quedó decidido el uso de la metodología RUP para guiar el proceso de desarrollo del sistema, lo que nos permitirá generar detalladamente la documentación solicitada por el cliente con el UML en su versión 2.1 como lenguaje de modelado, para llevar a cabo la modelación del negocio, y será el Visual Paradigm Enterprise Edition en su versión 8.0 la herramienta CASE para dicho lenguaje de modelado; para poder integrar el módulo con el sistema de nóminas existente se seleccionó como entorno de desarrollo para la programación el Visual Studio 2010, como lenguaje de programación, el C#, y como gestor de base de datos, el SQL Server 2005.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

2.1 Introducción

Al diseñar de un sistema informático es necesario aclarar todos los conceptos y sus relaciones, utilizando un lenguaje perceptible, mediante la descripción de la propuesta de solución para facilitar la comprensión de la misma por todos los involucrados. La poca claridad de los procesos en el entorno donde estará el sistema es uno de los motivos para la realización del modelo del dominio en el cual se capturarán los objetos existentes y eventos que suceden en el mismo. En este capítulo se describe la propuesta de solución, para lo cual se modelan los procesos básicos que intervienen en el objeto de estudio. Se definen las características y cualidades que debe tener el sistema a desarrollar, se enumeran los requisitos funcionales y no funcionales, lo que permite identificar mediante Diagramas de Casos de Uso las relaciones existentes con los actores involucrados. También se detalla la estructura de la arquitectura empleada, se describen los principales patrones arquitectónicos y de diseño, así como el modelo entidades de datos.

2.2 Modelo de dominio

Los principales procesos que ocurren en el negocio del sistema a desarrollar no se encuentran bien definidos. Por esta razón, se decidió realizar un modelo de dominio con el objetivo de representar y entender los principales conceptos relacionados con el ambiente donde será desplegado el sistema.

2.2.1 Análisis de los conceptos del dominio

Para lograr una mayor comprensión del diagrama de Modelo de Dominio que se expondrá a continuación, es necesario realizar previamente la definición de los conceptos que involucra dicho modelo:

Grupo Escala: Determina el salario escala del empleado para aquellos casos en que el mismo sea Sueldista o Jornalero (según Régimen Salarial).

Categoría Ocupacional: Describe la naturaleza de los cargos que conforman la plantilla de la entidad y determinan la estructura de la fuerza de trabajo.

Cargo: Contiene todos aquellos en base a los cuales se conformará la plantilla de la entidad. En este nomenclador se determina la escala salarial y la categoría de ocupación que tendrán todos aquellos empleados asociados a los mismos, así como otros datos importantes para la gestión del sistema.

Unidad Organizativa: Tiene especial significado para el departamento de personal pues permite reproducir la estructura en que está conformada la plantilla de la empresa, contiene la estructura organizativa de la entidad

Escala de Antigüedad: Contiene las clasificaciones de las categorías de antigüedad de los empleados.

Categoría de Antigüedad: Son las distintas categorías por las que se rige el pago según la antigüedad de los empleados.

Organizaciones de masas: Son las organizaciones de masas a las que pertenece un empleado.

Profesión u Oficio: Describe el nivel educacional de cada empleado. Se entiende como profesión a un grupo de conocimientos avalados por un título o certificado emitido por el Ministerio de Educación, el Ministerio de Educación Superior u otra entidad debidamente autorizada.

Contrato: Son los distintos tipos de contratos de trabajo que existen en una entidad.

Motivo de Alta/Baja: Contiene indistintamente las causas o motivos por los cuales un trabajador puede ser dado de alta o de baja del centro laboral.

Empleado: Las fichas o expedientes de los empleados almacenan toda la información relativa a los mismos y que concierne tanto al módulo de Personal como de Nómina.

Jornada: Son el fondo de tiempo o jornada de un trabajador en días/mes, horas/días horas/mes.

Conceptos de Pagos: Son los datos de los pagos de un trabajador definidos en la entidad de acuerdo a su estructura salarial.

Descuentos: Son aquellos que, como su nombre indica, representan deducciones del total a pagar al empleado. No se deben confundir con las retenciones que tienen otra connotación en el sistema.

Retenciones: Son los tipos de retenciones definidas en la entidad.

Provincia y Municipio: Relacionan todas las provincias del país con sus municipios.

Ausencias: Son las incidencias que tiene un trabajador de ausencias e impuntualidades durante un período de pago.

Período: Es el período de tiempo por el cual se realiza un pago.

Sanciones: Son los tipos de sanciones aprobadas por la legislatura que se le pueden aprobar a un empleado.

Idioma: Son los idiomas que domina un trabajador, teniendo en cuenta su nivel.

Movimientos: Es el listado de todos los movimientos que se han realizado al empleado comenzando con el alta.

Tipo de Plaza: Es el tipo de plaza que posee el trabajador.

Licencia Conducción: Son los tipos de licencias de conducción que puede tener un empleado.

Franco: Son los días de descansos del trabajador con jornada laboral de 26 días.

Días no laborales: Son los días no laborables durante un período de tiempo.

Calificación Escolar: Es la escolaridad vencida de un empleado.

Grupo de Pago: Es el grupo de pago al que pertenece el empleado.

Datos para el Pago: Son los datos relevantes de un empleado que se tienen en cuenta a la hora de realizar el pago.

2.2.2 Diagrama del modelo de dominio

El Modelo de Dominio es una representación visual y estática del entorno real del proyecto. En otras palabras, es un diagrama con los objetos reales que existen relacionados con el módulo que está en construcción y las relaciones presentes en él. Pero no son clases de *software*, aunque algunos objetos del Modelo de Dominio pueden terminar siéndolo. Se dice que es estática porque no representa la interacción en el tiempo de los objetos, sino que representa una visión "congelada" de las clases y sus interacciones.

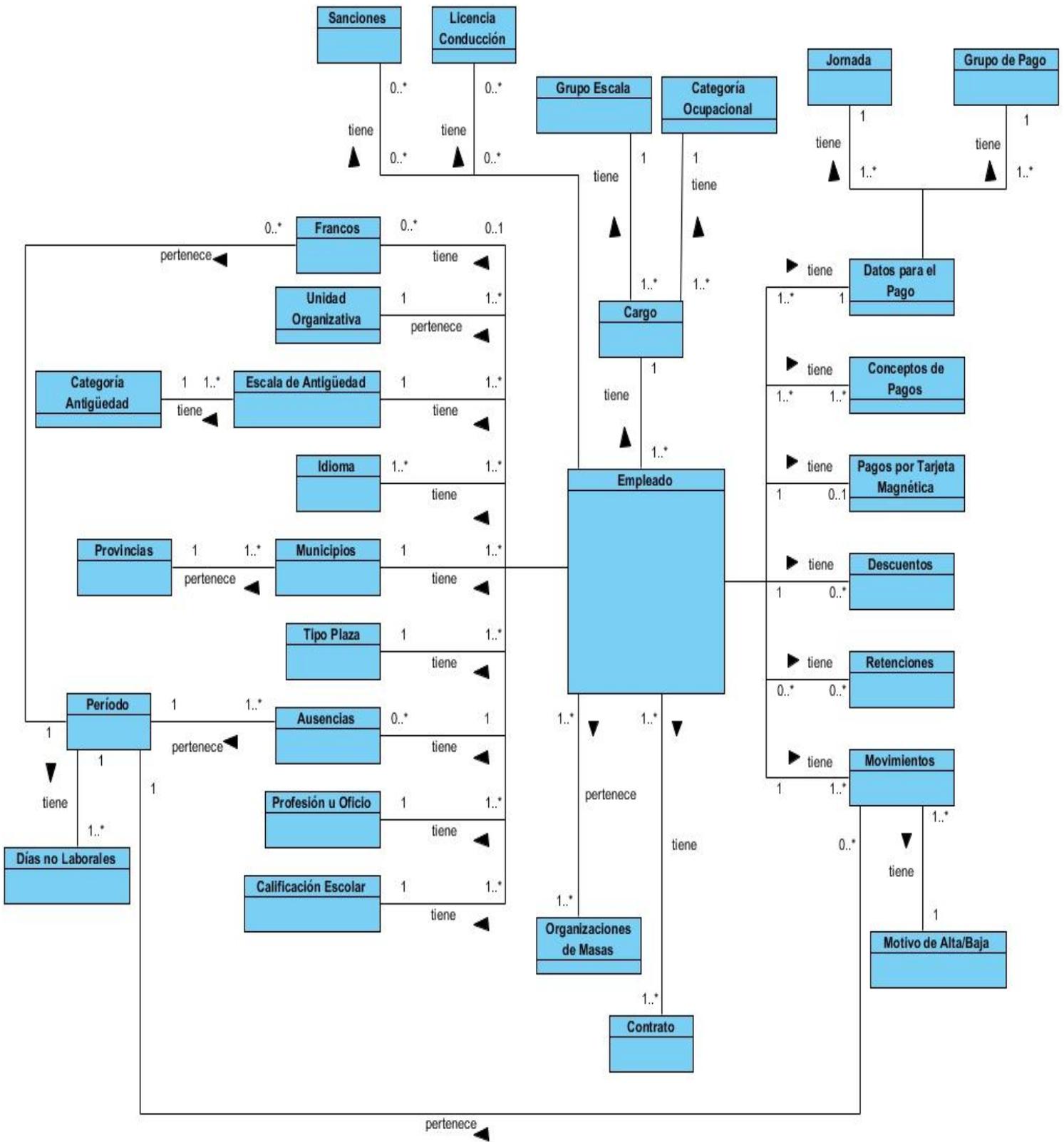


Figura 3. Representación del Modelo de Dominio.

2.3 Descripción del sistema propuesto

El objetivo del módulo propuesto es la gestión del personal, el cual permitirá a los especialistas de recursos humanos llevar a cabo todo el proceso de gestión y control del personal así como generar toda la información estadística necesaria.

Requisitos del software

Los requisitos del *software* son las capacidades y condiciones que el sistema debe cumplir según las necesidades del cliente.

Requisitos funcionales

Los Requisitos Funcionales (RF) son aquellos que, desde el punto de vista de las necesidades del usuario, debe cumplir el sistema, los cuales no varían por sus propiedades o cualidades relacionadas.

Para cumplir con los objetivos propuestos se prevé que el sistema tenga las siguientes funcionalidades:

- RF 1 Autenticarse:** Se le brinda al usuario la posibilidad de introducir sus datos para loguearse en el sistema.
- RF 2 Cambiar contraseña:** Se le brinda la posibilidad al usuario de poder cambiar contraseña al usuario sin necesidad de acceder al Módulo de la Seguridad.
- RF 3 Configurar registro de acceso:** Se le brinda al usuario la posibilidad de configurar a qué servidor se va a conectar en el sistema.
- RF 4 Gestionar empleados**
 - RF 4.1 Insertar empleado:** Se le brinda al usuario la posibilidad de introducir los datos de un nuevo empleado.
 - RF 4.2 Modificar empleado:** Se le permite al usuario la posibilidad de modificar los datos de un empleado.
 - RF 4.3 Eliminar empleado:** Se le permite al usuario la posibilidad de eliminar un empleado.
- RF 5 Gestionar tipos de contratos**
 - RF 5.5 Insertar tipo de contrato:** Se le brinda al usuario la posibilidad de introducir los datos de un nuevo tipo de contrato.

RF 5.6 Modificar tipo de contrato: Se le permite al usuario la posibilidad de modificar los datos de un tipo de contrato.

RF 5.7 Eliminar tipo de contrato: Se le permite al usuario la posibilidad de eliminar un tipo de contrato.

RF 6 Gestionar idiomas

RF 6.1 Insertar idioma: Se le brinda al usuario la posibilidad de introducir los datos de un nuevo idioma.

RF 6.2 Modificar idioma: Se le permite al usuario la posibilidad de modificar los datos de un idioma.

RF 6.3 Eliminar idioma: Se le permite al usuario la posibilidad de eliminar un idioma.

RF 7 Gestionar profesiones

RF 7.1 Insertar profesión: Se le brinda al usuario la posibilidad de introducir los datos de una nueva profesión.

RF 7.2 Modificar profesión: Se le permite al usuario la posibilidad de modificar los datos de una profesión.

RF 7.3 Eliminar profesión: Se le permite al usuario la posibilidad de eliminar una profesión.

RF 8 Gestionar sanciones

RF 8.1 Insertar sanción: Se le brinda al usuario la posibilidad de introducir los datos de una nueva sanción.

RF 8.2 Modificar sanción: Se le permite al usuario la posibilidad de modificar los datos de una sanción.

RF 8.3 Eliminar sanción: Se le permite al usuario la posibilidad de eliminar una sanción.

RF 9 Gestionar calificaciones escolares

RF 9.1 Insertar calificación escolar: Se le brinda al usuario la posibilidad de introducir los datos de una nueva calificación escolar.

RF 9.2 Modificar calificación escolar: Se le permite al usuario la posibilidad de modificar los datos de una calificación escolar.

RF 9.3 Eliminar calificación escolar: Se le permite al usuario la posibilidad de eliminar una calificación escolar.

RF 10 Gestionar tipos de plazas

- RF 10.1 Insertar tipo de plaza:** Se le brinda al usuario la posibilidad de introducir los datos de un nuevo tipo de plaza.
- RF 10.2 Modificar tipo de plaza:** Se le permite al usuario la posibilidad de modificar los datos de un tipo de plaza.
- RF 10.3 Eliminar Tipo de plaza:** Se le permite al usuario la posibilidad de eliminar un tipo de plaza.
- RF 11 Gestionar organizaciones de masas**
- RF 11.1 Insertar organización de masa:** Se le brinda al usuario la posibilidad de introducir los datos de una nueva organización de masa.
- RF 11.2 Modificar organización de masa:** Se le permite al usuario la posibilidad de modificar los datos de una organización de masa.
- RF 11.3 Eliminar organización de masa:** Se le permite al usuario la posibilidad de eliminar una organización de masa.
- RF 12 Gestionar motivos de altas/bajas**
- RF 12.1 Insertar motivo de altas/bajas:** Se le brinda al usuario la posibilidad de introducir los datos de un nuevo motivo de altas/bajas.
- RF 12.2 Modificar motivo de altas/bajas:** Se le permite al usuario la posibilidad de modificar los datos de un motivo de altas/bajas.
- RF 12.3 Eliminar motivo de altas/bajas:** Se le permite al usuario la posibilidad de eliminar un motivo de altas/bajas.
- RF 13 Gestionar categorías de antigüedad**
- RF 13.1 Insertar categoría de antigüedad:** Se le brinda al usuario la posibilidad de introducir los datos de una nueva categoría de antigüedad.
- RF 13.2 Modificar categoría de antigüedad:** Se le permite al usuario la posibilidad de modificar los datos de una categoría de antigüedad.
- RF 13.3 Eliminar categoría de antigüedad:** Se le permite al usuario la posibilidad de eliminar una categoría de antigüedad.
- RF 14 Gestionar escalas salariales**
- RF 14.1 Insertar escala salarial:** Se le brinda al usuario la posibilidad de introducir los datos de una nueva escala salarial.
- RF 14.2 Modificar escala salarial:** Se le permite al usuario la posibilidad de modificar los datos de una escala salarial.
- RF 14.3 Eliminar escala salarial:** Se le permite al usuario de acuerdo a sus permisos, eliminar una escala salarial.

RF 15 Gestionar licencias de conducción

RF 15.1 Insertar licencia de conducción: Se le brinda al usuario la posibilidad de introducir los datos de una nueva licencia de conducción.

RF 15.2 Modificar licencia de conducción: Se le permite al usuario la posibilidad de modificar los datos de una licencia de conducción.

RF 15.3 Eliminar licencia de conducción: Se le permite al usuario la posibilidad de eliminar una licencia de conducción.

RF 16 Gestionar retenciones

RF 16.1 Insertar retención: Se le brinda al usuario la posibilidad de introducir los datos de una nueva retención.

RF 16.2 Modificar retención: Se le permite al usuario la posibilidad de modificar los datos de una retención.

RF 16.3 Eliminar retención: Se le permite al usuario la posibilidad de eliminar una retención.

RF 17 Gestionar cargos

RF 17.1 Insertar cargo: Se le brinda al usuario la posibilidad de introducir los datos de un nuevo cargo.

RF 17.2 Modificar cargo: Se le permite al usuario la posibilidad de modificar los datos de un cargo.

RF 17.3 Eliminar cargo: Se le permite al usuario la posibilidad de eliminar un cargo.

RF 18 Gestionar pagos

RF 18.1 Insertar pago: Se le brinda al usuario la posibilidad de introducir los datos de un nuevo pago.

RF 18.2 Modificar pago: Se le permite al usuario la posibilidad de modificar los datos de un pago.

RF 18.3 Eliminar pago: Se le permite al usuario la posibilidad de eliminar un pago.

RF 19 Gestionar provincias

RF 19.1 Insertar provincia: Se le brinda al usuario la posibilidad de introducir los datos de una nueva provincia.

RF 19.2 Modificar provincia: Se le permite al usuario la posibilidad de modificar los datos de una provincia.

- RF 19.3 Eliminar provincia:** Se le permite al usuario la posibilidad de eliminar una provincia.
- RF 20 Gestionar municipios**
- RF 20.1 Insertar municipio:** Se le brinda al usuario la posibilidad de introducir los datos de un nuevo municipio.
- RF 20.2 Modificar municipio:** Se le permite al usuario la posibilidad de modificar los datos de un municipio.
- RF 20.3 Eliminar municipio:** Se le permite al usuario la posibilidad de eliminar un municipio.
- RF 21 Gestionar unidades organizativas**
- RF 21.1 Insertar unidad organizativa:** Se le brinda al usuario la posibilidad de introducir los datos de una nueva unidad organizativa.
- RF 21.2 Modificar unidad organizativa:** Se le permite al usuario la posibilidad de modificar los datos de una unidad organizativa.
- RF 21.3 Eliminar unidad organizativa:** Se le permite al usuario la posibilidad de eliminar una unidad organizativa.
- RF 22 Gestionar pagos por resultados**
- RF 22.1 Insertar pago por resultado:** Se le brinda al usuario la posibilidad de introducir los datos de un nuevo pago por resultado.
- RF 22.2 Modificar pago por resultado:** Se le permite al usuario la posibilidad de modificar los datos de un pago por resultado.
- RF 22.3 Eliminar pago por resultado:** Se le permite al usuario la posibilidad de eliminar un pago por resultado.
- RF 23 Gestionar ausencias**
- RF 23.1 Insertar ausencia:** Se le brinda al usuario la posibilidad de introducir los datos de una nueva ausencia.
- RF 23.2 Modificar ausencia:** Se le permite al usuario la posibilidad de modificar los datos de una ausencia.
- RF 23.3 Eliminar ausencia:** Se le permite al usuario la posibilidad de eliminar una ausencia.
- RF 24 Gestionar categorías ocupacionales**
- RF 24.1 Insertar categoría ocupacional:** Se le brinda al usuario la posibilidad de introducir los datos de una nueva categoría ocupacional.

- RF 24.2 Modificar categoría ocupacional:** Se le permite al usuario la posibilidad de modificar los datos de una categoría ocupacional.
- RF 24.3 Eliminar categoría ocupacional:** Se le permite al usuario la posibilidad de eliminar una categoría ocupacional.
- RF 25 Gestionar grupos de nómina**
- RF 25.1 Insertar grupo de nómina:** Se le brinda al usuario la posibilidad de introducir los datos de un nuevo grupo de nómina.
- RF 25.2 Modificar grupo de nómina:** Se le permite al usuario la posibilidad de modificar los datos de un grupo de nómina.
- RF 25.3 Eliminar grupo de nómina:** Se le permite al usuario la posibilidad de eliminar un grupo de nómina.
- RF 26 Gestionar jornadas laborales**
- RF 26.1 Insertar jornada laboral:** Se le brinda al usuario la posibilidad de introducir los datos de una nueva jornada laboral.
- RF 26.2 Modificar jornada laboral:** Se le permite al usuario la posibilidad de modificar los datos de una jornada laboral.
- RF 26.3 Eliminar jornada laboral:** Se le permite al usuario la posibilidad de eliminar una jornada laboral.
- RF 27 Gestionar días no laborables**
- RF 27.1 Insertar día no laborable:** Se le brinda al usuario la posibilidad de introducir los datos de un nuevo día no laborable.
- RF 27.2 Eliminar día no laborable:** Se le permite al usuario la posibilidad de eliminar un día no laborable.
- RF 28 Gestionar ausencias de un empleado**
- RF 28.1 Insertar ausencias de un empleado:** Se le brinda al usuario la posibilidad de introducir los datos de una nueva ausencia a un empleado.
- RF 28.2 Eliminar ausencias de un empleado:** Se le permite al usuario la posibilidad de eliminar una ausencia a un empleado.
- RF 29 Gestionar francos de un empleado**
- RF 29.1 Insertar francos de un empleado:** Se le brinda al usuario la posibilidad de introducir un nuevo francos a un empleado con una jornada de 26 días.
- RF 29.2 Eliminar francos de un empleado:** Se le permite al usuario la posibilidad de eliminar un francos a un empleado.

- RF 30 Gestionar conceptos de pagos de un empleado**
- RF 30.1 Insertar concepto de pago de un empleado:** Se le brinda al usuario la posibilidad de introducir los datos de un nuevo concepto de pago a un empleado.
- RF 30.2 Modificar concepto de pago de un empleado:** Se le permite al usuario la posibilidad de modificar los datos de un concepto de pago a un empleado.
- RF 30.3 Eliminar concepto de pagos de un empleado:** Se le permite al usuario la posibilidad de eliminar un concepto de pago a un empleado.
- RF 31 Gestionar retenciones de un empleado**
- RF 31.1 Insertar retención de un empleado:** Se le brinda al usuario la posibilidad de introducir los datos de una nueva retención a un empleado.
- RF 31.2 Modificar retención de un empleado:** Se le permite al usuario la posibilidad de modificar los datos de una retención a un empleado.
- RF 31.3 Eliminar retención de un empleado:** Se le permite al usuario la posibilidad de eliminar una retención a un empleado.
- RF 32 Gestionar descuentos de un empleado**
- RF 32.1 Insertar descuento de un empleado:** Se le brinda al usuario la posibilidad de introducir los datos de un nuevo descuento a un empleado.
- RF 32.2 Modificar descuento de un empleado:** Se le permite al usuario la posibilidad de modificar los datos de un descuento a un empleado.
- RF 32.3 Eliminar descuento de un empleado:** Se le permite al usuario la posibilidad de eliminar un descuento a un empleado.
- RF 33 Gestionar sanciones de un empleado**
- RF 33.1 Insertar sanción de un empleado:** Se le brinda al usuario la posibilidad de introducir los datos de una nueva sanción a un empleado.
- RF 33.2 Modificar sanción de un empleado:** Se le permite al usuario la posibilidad de modificar los datos de una sanción a un empleado.
- RF 33.3 Eliminar sanción de un empleado:** Se le permite al usuario la posibilidad de eliminar una sanción a un empleado.
- RF 34 Gestionar idiomas de un empleado**
- RF 34.1 Insertar idioma de un empleado:** Se le brinda al usuario la posibilidad de introducir los datos de un nuevo idioma a un empleado.

RF 34.2 Modificar idioma de un empleado: Se le permite al usuario la posibilidad de modificar los datos de un idioma a un empleado.

RF 34.3 Eliminar idioma de un empleado: Se le permite al usuario la posibilidad de eliminar un idioma a un empleado.

RF 35 Listar empleados: Muestra un listado con todos los trabajadores.

RF 36 Generar reportes: Se le brinda al usuario la posibilidad de generar uno de los siguientes reportes en forma de documento listo para imprimir o guardar.

- Mostrar análisis periódico de asistencia.
- Mostrar notificación de Vacaciones.
- Mostrar prenómina.
- Mostrar cumpleaños de un rango de fecha.
- Mostrar contratos elaborados.
- Mostrar contratos próximos a vencer.
- Mostrar movimiento de nómina de un empleado.
- Mostrar contrato de un empleado dada la proforma para ese tipo de contrato.
- Mostrar listado de empleados.

Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el sistema debe presentar para hacerlo usable, rápido, confiable y atractivo.

Software

Las computadoras deben tener instalado Windows como sistema operativo y el Framework .Net versión 4.0 o superior.

Hardware

Las computadoras que brindarán el servicio del sistema no deberán presentar potencias menores a las brindadas por una Pentium 4 con al menos 512 MB de RAM, 10 GB de espacio en disco duro y una tarjeta de red.

Restricciones en el diseño y la implementación

- El lenguaje de programación a usar debe ser C#.
- El marco de trabajo base de desarrollo que se utilizará es .Net 4.0.
- Como IDE se empleará Microsoft Visual Studio 2010.
- El SGDB deberá ser Microsoft SQL Server 2005.

Apariencia o interfaz externa

- Debe contener un diseño sencillo, de manera que no sea necesario mucho entrenamiento para utilizar el sistema.

Seguridad y confiabilidad

- El sistema debe proteger el contenido del personal que gestione, de forma que limite el acceso a usuarios no autorizados.
- La seguridad de acceso y administración de usuarios será mediante el otorgamiento de privilegios. Los niveles de acceso están determinados por los diferentes privilegios.

Usabilidad

El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora en sentido general.

Rendimiento

- La ejecución de acciones debe ser de manera rápida para minimizar los pasos a dar en cada proceso.
- Debe suministrarse un producto eficiente, que garantice la consistencia y disponibilidad de la información.

Disponibilidad

El sistema estará disponible para los usuarios registrados por el Sistema Administrador de Seguridad (ZUNsa), desde la institución en la que se labora, manteniendo la integridad de los datos.

2.4 Modelos de casos de uso del sistema

El modelo de casos de uso del sistema representa la forma en que el actor se relaciona con los casos de uso, es decir, los flujos de eventos que ocurren al interactuar un

usuario y el sistema; puede "incluir" o "extender" la funcionalidad de otro con su propio comportamiento. Estos se encuentran vinculados directamente a los requisitos funcionales susceptibles de automatización.

Patrones de casos de uso

Son comportamientos que deben existir en el sistema. Facilitan la descripción de lo que este debe hacer, es decir, describen el uso del sistema y su interacción con los usuarios. Se utilizan generalmente como plantillas para describir la estructura y organización que deberían tener los casos de uso. Los mismos capturan mejores prácticas para la modelación de los casos de uso.

Patrón CRUD

El patrón CRUD completo consiste en un caso de uso para administrar la información, permite modelar las diferentes operaciones para administrar una entidad de información, tales como crear, mostrar, editar y eliminar.

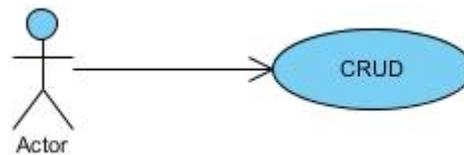


Figura 4. Patrón CRUD.

Incluir (include)

Cuando relacionamos dos casos de uso con un "include", estamos diciendo que el primero (el caso de uso base) incluye al segundo (el caso de uso incluido). Es decir, el segundo es parte esencial del primero. Sin el segundo, el primero no podría funcionar bien; pues no podría cumplir su objetivo. (Sergio Orozco, 2014)

Un ejemplo de su uso se evidencia en la siguiente figura:

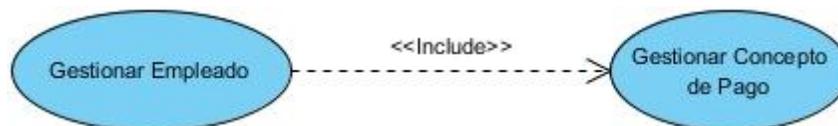


Figura 5. Incluido.

2.4.1 Descripción del actor del sistema

Un actor del sistema es un usuario o sistema que interactúa directamente con el sistema o aplicación que se desarrolla, estas personas suelen corresponderse con trabajadores o actores del negocio que se automatiza, por tanto un actor del sistema tiene el propósito de lograr un beneficio.

Actor	Descripción
Especialista de Recursos Humanos (RH)	Actor encargado de llevar a cabo todo el proceso de gestión y control del personal, así como generar toda la información estadística necesaria y mostrar los reportes.

Tabla 1. Descripción de los actores del sistema.



Figura 6. Diagrama de Actores.

2.4.2 Diagramas de casos de uso del sistema

Los diagramas de casos de uso (DCU) se usan para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. Es decir, un diagrama de casos de uso del sistema representa gráficamente a los procesos y su interacción con los actores. Por la dimensión del sistema, el DCU se realizó por paquetes teniendo en cuenta sus funcionalidades para su agrupación.

Paquete Empleado (PE)

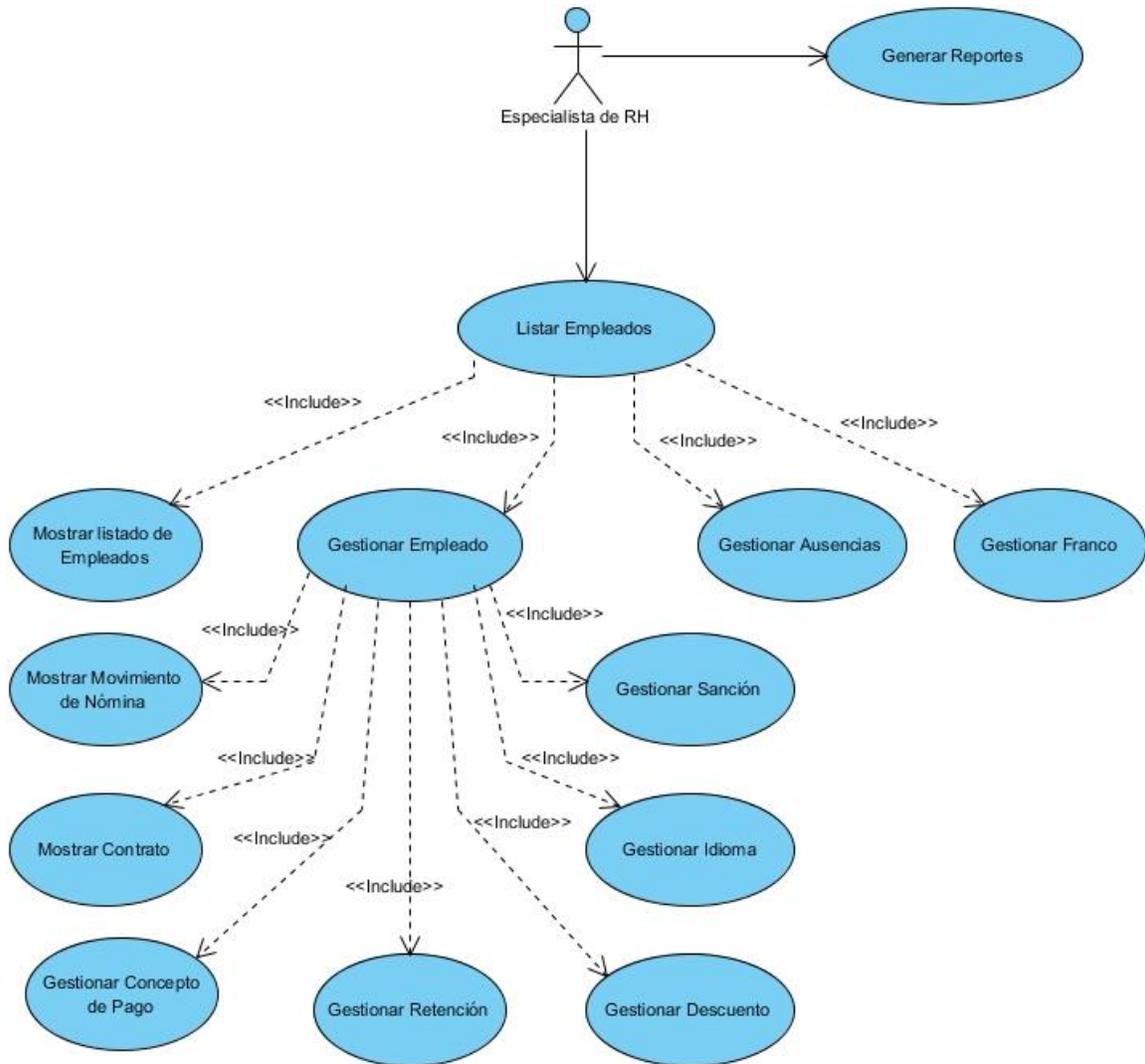


Figura 7. Diagrama de Casos de Uso del Sistema PE.

Paquete Configuración (PC)

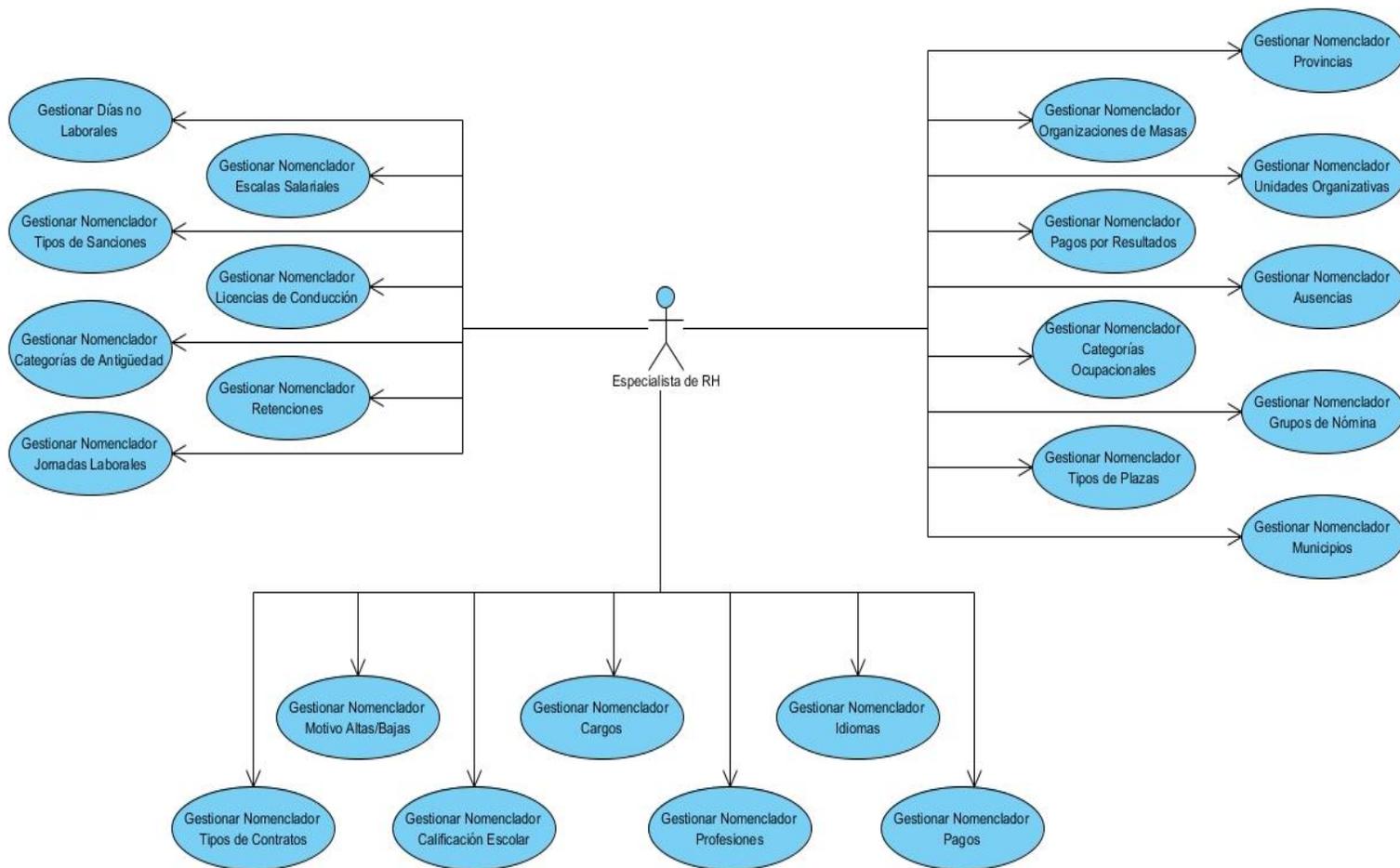


Figura 8. Diagrama de Casos de Uso del Sistema PC.

Paquete Autenticación (PA)

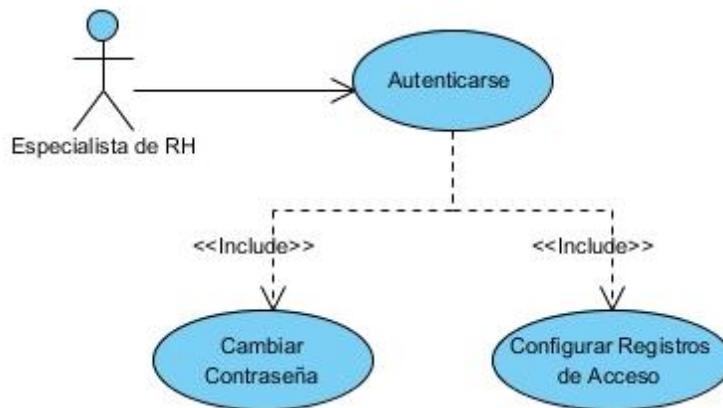


Figura 9. Diagrama de Casos de Uso del Sistema PA.

2.4.3 Descripción de los casos de uso del sistema

La descripción de los casos de uso del sistema no es más que la acción que se realiza durante la interacción entre el actor y el sistema, o sea describe el flujo de actividades que realiza un actor al hacer uso del sistema y las respuestas que emite el mismo.

A continuación se muestra la descripción de los casos de uso del CU Gestionar Empleado. La otra parte se encuentra en el [Anexo 2](#).

CU Gestionar Empleado

Especificación del Caso de Uso

Objetivo	Añadir, modificar y eliminar un empleado.
Actores	Especialista de RH (Inicia): Añade, modifica o elimina un Empleado.
Resumen	El caso de uso se inicia cuando el actor selecciona la opción que le permite realizar una acción sobre los empleados. El actor puede añadir, eliminar y modificar un empleado. En caso de que seleccione la opción de añadir un empleado, el sistema dará la posibilidad de insertar los datos que se necesitan para llenar este formulario, si se culmina el proceso de inserción correctamente el sistema saca una pantalla para insertar los datos de movimiento en caso que el sistema no esté en instalación, de estarlo se generará automáticamente el movimiento de alta y se generará un contrato con los datos insertados. Si el actor elige la opción de modificar el empleado, el sistema mostrará los datos y permitirá modificar los que pueden ser editables dentro del empleado, y una vez realizados los cambios, guardará las modificaciones y si modificó determinados datos del pago el sistema saca una pantalla para insertar los datos de movimiento que generan dichos cambios, manteniendo el movimiento anterior en el listado de movimiento, por otra parte si se modificó el tipo de contrato se generará un nuevo contrato con los datos modificados, manteniendo el contrato anterior en el listado de contrato. El sistema según el estado en que se encuentre, en instalación o no, le permitirá al actor elegir la opción de eliminar el empleado, la cual permitirá eliminar físicamente el empleado del sistema mostrando un mensaje de confirmación, o dar baja al empleado generando un movimiento de baja, esta opción no borrará al empleado físicamente del

	sistema sino que le dará de baja, terminando así el caso de uso.	
Precondiciones	<ul style="list-style-type: none"> • Debe haberse generado el escritorio de trabajo del usuario autenticado. • Para añadir, modificar o eliminar debe haber generado el <i>CU Listar Empleados</i>. • Para modificar Empleado el actor debe ser el responsable temporal del mismo. • Para eliminar Empleado el actor debe tener el permiso de eliminar el elemento. 	
Poscondiciones	Se añadió, modificó o eliminó el Empleado por el actor.	
Flujo de eventos		
Flujo básico		
	Acciones del Actor	Acciones del Sistema
1.	El caso de uso se inicia cuando el actor selecciona la opción de realizar una acción sobre un Empleado.	
2.		Brinda la posibilidad de realizar las acciones: <ul style="list-style-type: none"> • Añadir un nuevo Empleado. • Modificar los datos del Empleado. Ver Sección 1: "<u>Modificar datos del Empleado</u>". • Eliminar el Empleado. Ver Sección 2: "<u>Eliminar Empleado</u>". • Dar baja al empleado. Ver Sección 3: "<u>Dar Baja al Empleado</u>".
3.	Selecciona la opción de Añadir un nuevo Empleado.	
4.		Brinda la posibilidad de introducir o seleccionar los datos del empleado por pestañas: Pestaña Datos Personales

		<ul style="list-style-type: none"> • Nombre. • Primer apellido. • Segundo apellido. • Carnet de identidad. • Fecha de nacimiento. • Sexo. • Color de la piel. • Número de hijos. • Profesión u oficio. • Escala de antigüedad. • Nombre de la madre. • Nombre del padre. • Estado civil. • Estatura. • Calificación escolar. • Provincia. • Municipio. • Categoría docente. • Categoría científica. • Licencia de conducción. • Número de licencia. • Conyugue. • Núcleo familiar. • Militancia. • Organizaciones políticas. • Organizaciones de masa. • Número de teléfono. • Situación para la defensa. • Dirección. <p>Pestaña Datos Principales</p> <ul style="list-style-type: none"> • Número interno. • Fecha de alta.
--	--	--

		<ul style="list-style-type: none"> • Tipo de plaza. • Unidad organizativa. • Centro de costo. • Tipo de contrato. • Fecha de inicio de contrato. • Fecha fin de contrato. • Cargo. • Fecha inicio del cargo. • Grupo escala. • Categoría ocupacional. • Conceptos de Pagos. • Fecha inicio en el sector. • Forma de pago. • Tarifa horaria por defecto. • Trabaja media jornada. • Horas/Mes. • Días/Mes. • Horas/Días. • Tarifa horaria por defecto. • Tarifa diaria por defecto. • Régimen salarial. • Grupo de pago. • Pago por resultado. • Acumula vacaciones. • Pago por vacaciones. • Vacaciones iniciales en días. • Vacaciones iniciales en importe. • Vacaciones acumuladas en días. • Vacaciones acumuladas en importe. • Terminar contable. <p>Pestaña Otros Datos Principales</p>
--	--	--

		<ul style="list-style-type: none"> • Descuentos. • Retenciones. • Pago por tarjeta magnética en CUP. • Cuenta bancaria en CUP. • Nombre en tarjeta magnética. • Pago por tarjeta magnética en CUC. • Cuenta bancaria en CUC. • Tipo de identificación para banco. <p>Pestaña Sanción</p> <ul style="list-style-type: none"> • Tipo de sanción. • Número de resolución. • Fecha de creación. • Fecha de notificación. • Fecha de culminación. <p>Pestaña Idiomas</p> <ul style="list-style-type: none"> • Idioma. • Nivel del idioma. • Está certificado. • Cómo lee. • Cómo escribe. • Cómo habla. • Nombre de la escuela.
5.	Introduce y selecciona los datos del empleado.	
6.	Selecciona en la pestaña <i>Datos Principales</i> la opción Conceptos de Pagos.	
7.		Brinda la posibilidad de añadir, modificar o Eliminar un Pago ver " <u>CU Gestionar Concepto de Pago</u> ".
8.		Permite seleccionar la opción: <ul style="list-style-type: none"> • Aceptar.

		<ul style="list-style-type: none"> • Cancelar.
9.	Selecciona la opción de Aceptar.	
10.		<ul style="list-style-type: none"> • Valida los datos. • Añade el empleado. • Muestra una pantalla para llenar los datos del Movimiento de Nómina permitiendo solo la opción de aceptar después de introducir y seleccionar los siguientes datos: <ul style="list-style-type: none"> • Fecha del movimiento. • Días con salario anterior. • Motivo del movimiento. • Descripción.
11.	Introduce y selecciona los datos del movimiento y selecciona la opción de Aceptar.	
12.		Muestra un mensaje de información. Termina el caso de uso.

Flujo alterno

9. a Selecciona la opción de Cancelar.

	Acciones del Actor	Acciones del Sistema
		*.a.1 Elimina los datos creados. *.a.2 Regresa a la vista anterior. *.a.3 El caso de uso termina.

6. a Selecciona en la pestaña *Otros Datos Principales* la opción de Retenciones

	Acciones del Actor	Acciones del Sistema
		6. a.1 Brinda la posibilidad de añadir, modificar o eliminar una Retención ver " <u>CU Gestionar Retenciones</u> ".

6. b Selecciona en la pestaña *Otros Datos Principales* la opción de Descuentos

	Acciones del Actor	Acciones del Sistema
--	--------------------	----------------------

		6. b.1 Brinda la posibilidad de añadir, modificar o eliminar un Descuento ver " <u>CU Gestionar Descuentos</u> ".
6. c Selecciona la pestaña Movimientos de Nóminas.		
	Acciones del Actor	Acciones del Sistema
		6. c.1 Lista todos los movimientos del trabajador. 6. c.2 Brinda la posibilidad de imprimir el movimiento seleccionado.
6. d Selecciona la pestaña Contratos.		
	Acciones del Actor	Acciones del Sistema
		6. d.1 Lista todos los contratos del trabajador. 6. d.2 Brinda la posibilidad de imprimir el contrato seleccionado.
6. e Selecciona la pestaña Sanciones.		
	Acciones del Actor	Acciones del Sistema
		6. b.1 Brinda la posibilidad de añadir, modificar o eliminar una sanción ver " <u>CU Gestionar Sanciones</u> ".
6. f Selecciona la pestaña Idiomas.		
	Acciones del Actor	Acciones del Sistema
		6. b.1 Brinda la posibilidad de añadir, modificar o eliminar un idioma ver " <u>CU Gestionar Idiomas</u> ".
10. a Existen datos incompletos.		
	Acciones del Actor	Acciones del Sistema
		10. a.1 Muestra un mensaje por cada dato incompleto.
10. b Existen datos incorrectos.		

	Acciones del Actor	Acciones del Sistema
		10. b.1 Muestra un mensaje por cada dato incorrecto.

Sección 1: “Modificar datos del Empleado”.

Flujo Básico		
	Acciones del Actor	Acciones del Sistema
1.	Selecciona la opción de modificar los datos del empleado.	
2.		Muestra los datos del empleado y brinda la posibilidad de cambiar sus valores ya sea introduciendo nuevos o seleccionando diferentes. Permite además: <ul style="list-style-type: none"> • Aceptar los datos. • Cancelar la operación en cualquier momento.
3.	Modifica los datos que necesite y selecciona la opción de Aceptar los mismos.	
4.		<ul style="list-style-type: none"> • Valida los datos. • Actualiza los datos del empleado. • Muestra una pantalla para llenar los datos del movimiento permitiendo solo la opción de aceptar • Muestra un mensaje de información.
5.	Introduce y selecciona los datos del movimiento y selecciona la opción de Aceptar.	
6.		Muestra un mensaje de información. Termina el caso de uso.

Flujo Alterno		
3. a El actor selecciona la opción de Cancelar.		
	Acciones del Actor	Acciones del Sistema
4.		a.1 Elimina los datos creados. a.3 Regresa a la vista anterior. a.4 El caso de uso termina.
4. a Existen datos incompletos.		
	Acciones del Actor	Acciones del Sistema
		a.1 Muestra un mensaje que indica que dato faltó.
		a.2 Regresa al paso 2 del Flujo Básico.

Sección 2: “Eliminar Empleado”.

Flujo Básico		
	Acciones del Actor	Acciones del Sistema
1.	Selecciona la opción de eliminar empleado.	
2.		Muestra el mensaje de advertencia y permite: <ul style="list-style-type: none"> • Aceptar. • Cancelar la operación.
3.	Selecciona la opción de Aceptar.	
4.		<ul style="list-style-type: none"> • Se elimina el Empleado. • Regresa a la vista anterior. • Termina el caso de uso.
Flujo Alterno		
3. a El actor selecciona la opción de Cancelar.		
	Acciones del Actor	Acciones del Sistema
4.		a.1 Regresa a la vista anterior. a.2 El caso de uso termina.

Sección 3: “Dar Baja al Empleado”.

Flujo Básico		
	Acciones del Actor	Acciones del Sistema
1.	Selecciona la opción de eliminar Empleado.	
2.		Muestra el mensaje de advertencia y permite: <ul style="list-style-type: none"> • Aceptar. • Cancelar la operación.
3.	Selecciona la opción de Aceptar.	
4.		<ul style="list-style-type: none"> • Muestra una pantalla para llenar los datos del movimiento de baja permitiendo solo la opción de aceptar. • Se le da de baja al empleado. • Muestra un mensaje de información. • Regresa a la vista anterior.
	Introduce y selecciona los datos del movimiento y selecciona la opción de Aceptar.	
5.		Termina el caso de uso.
Flujo Alternativo		
3. a El actor selecciona la opción de Cancelar.		
	Acciones del Actor	Acciones del Sistema
4.		a.1 Regresa a la vista anterior. a.2 El caso de uso termina.

2.5 Modelo de análisis

El modelo de análisis es una representación conceptual donde se modela una serie de clases y relaciones. Se crea con el objetivo de refinar los casos de uso de una forma más detallada, posibilitando un conocimiento razonable del sistema o producto, permite entender mejor los aspectos internos del sistema y a través de estos diagramas se logra una mayor formalización.

Clase	Estereotipo	Descripción
--------------	--------------------	--------------------

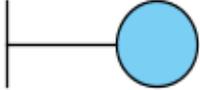
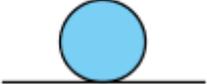
Interfaz	 CI	<ul style="list-style-type: none"> • Se utiliza para modificar la interacción entre el sistema y sus actores. • Es la encargada de recibir y representar informaciones y peticiones de usuarios y de sistemas externos. • Representa la abstracción de ventanas, formularios, paneles e interfaz de comunicación.
Control	 CC	<ul style="list-style-type: none"> • Se encarga de presentar coordinación secuencial, transacciones y control de otros objetos. • Se usa para encapsular el control de un caso de uso en concreto. • Se usa para representar otros procesos complejos.
Entidad	 CE	<ul style="list-style-type: none"> • Se usa para modelar información que posee larga vida, y que es a menudo persistente. • Modela la información y comportamiento asociado de algún fenómeno o concepto (persona, objeto del mundo real o suceso). • Se deriva de una clase de entidad del modelo del dominio.

Tabla 2. Descripción de los estereotipos.

2.5.1 Diagrama de clases del análisis

Un diagrama de clases del análisis es el artefacto que representa los conceptos con el contexto del dominio del problema.

Se muestra una representación del Diagrama de Clases del Análisis (DCA) del CU Gestionar Empleado.

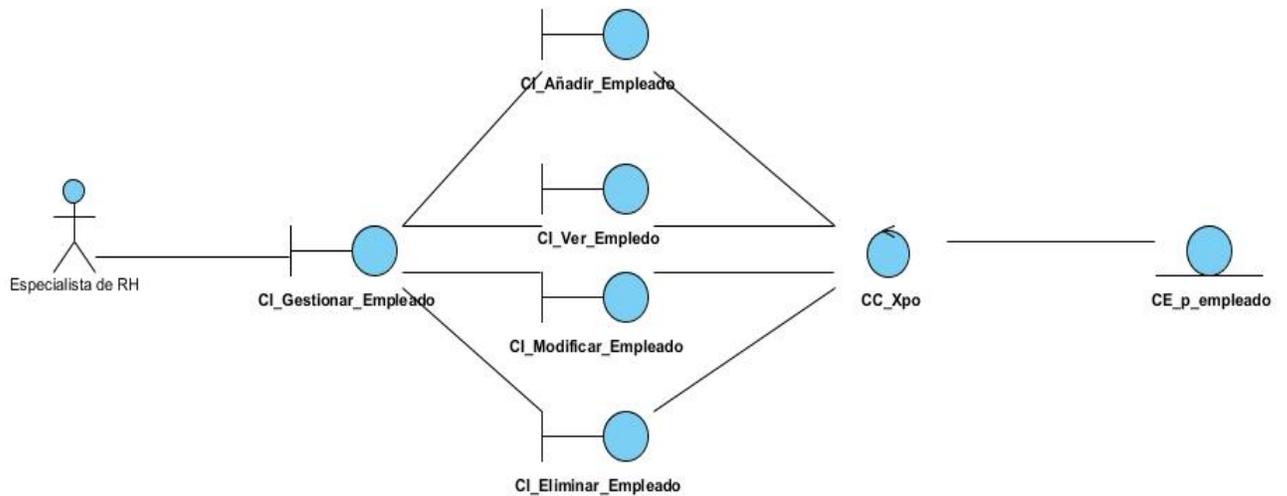


Figura 10. DCA CU Gestionar Empleado.

2.5.2 Diagrama de colaboración del análisis

Un diagrama de colaboración ayuda al descubrimiento de abstracciones pues permite representar los objetos en una disposición próxima a la realidad, dado que permite representar una disposición espacial de la estructura de los objetos en ejecución. La estructura puede ser estática y dinámica, la primera viene dada por los enlaces y la segunda por el envío de mensajes a través de los enlaces.

Se muestra una representación de los Diagramas de Colaboración del análisis (DC) del CU Gestionar Empleado.

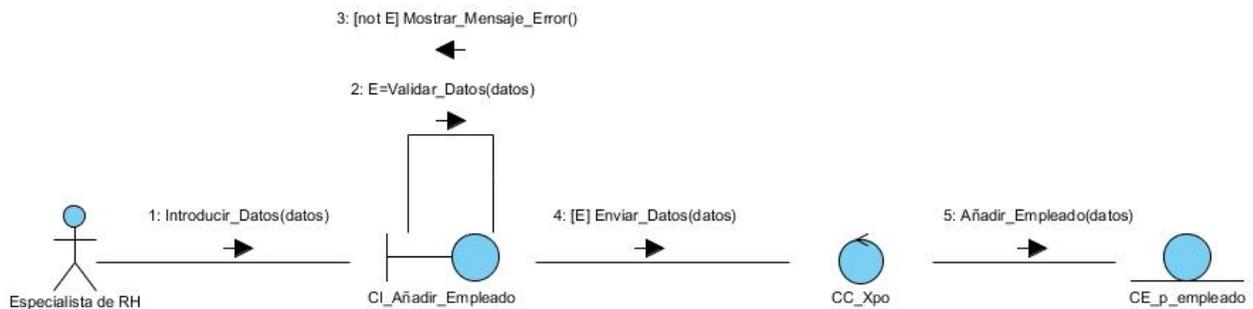


Figura 11. DC CU Gestionar Empleado. Sección Añadir.

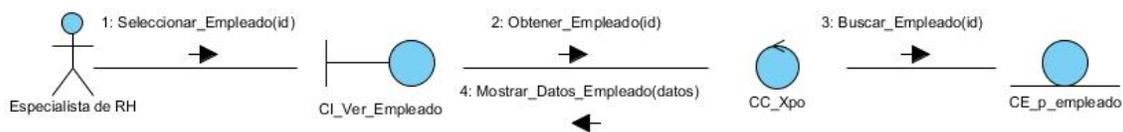


Figura 12. DC CU Gestionar Empleado. Sección Ver.

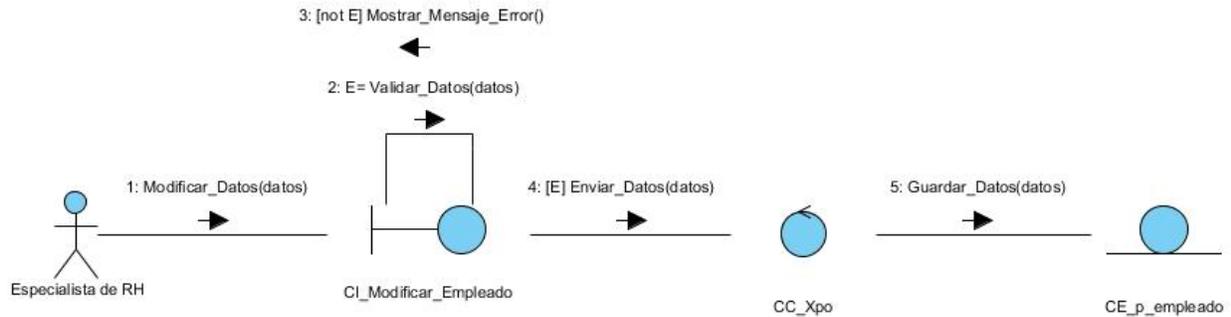


Figura 13. DC CU Gestionar Empleado. Sección Modificar.

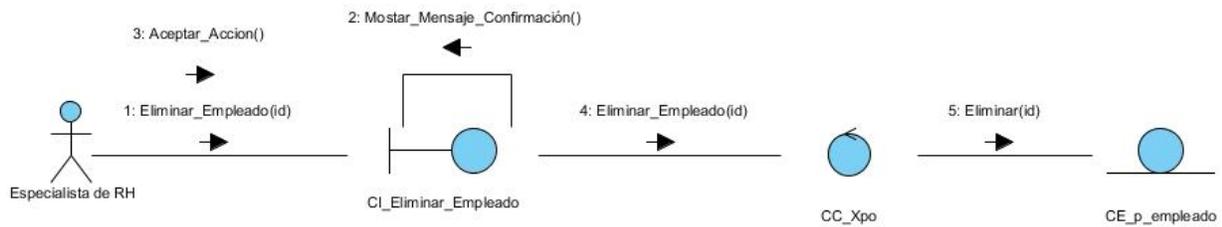


Figura 14. DC CU Gestionar Empleado. Sección Eliminar.

2.6 Modelo de diseño

Es un modelo de objetos que representa las colaboraciones que ocurren entre las páginas, donde cada página lógica puede ser representada como una clase. Esto facilita una abstracción de la implementación del sistema, lo que constituye el artefacto fundamental de entrada a la implementación.

2.6.1 Patrón de diseño

“Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software.” (Tedeschi, 2014)

Estos brindan una solución probada y documentada a problemas de desarrollo de *software* que están sujetos a contextos similares.

En el proceso de desarrollo se empleó el patrón **Modelo Vista Controlador (MVC)**.

Este patrón propone dividir en tres capas las funcionalidades de una aplicación con el objetivo de incrementar la usabilidad y organización de las mismas. Estas capas son:

Modelo: Esta capa es la representación específica de la información con la cual el sistema opera. Este se limita a lo relativo de la vista y su controlador, facilitando las presentaciones visuales complejas. El sistema también puede operar con más datos no relativos a la presentación, haciendo uso integrado de otras lógicas de negocio y de datos afines con el sistema modelado. En otras palabras, es la encargada del acceso a los datos.

Vista: Esta capa presenta el modelo en un formato adecuado para interactuar, generalmente la interfaz de usuario.

Controlador: Esta capa responde a eventos, usualmente acciones del usuario, que invoca peticiones al modelo y, probablemente, a la vista. En otras palabras, es la encargada de llevar la lógica de control del negocio.

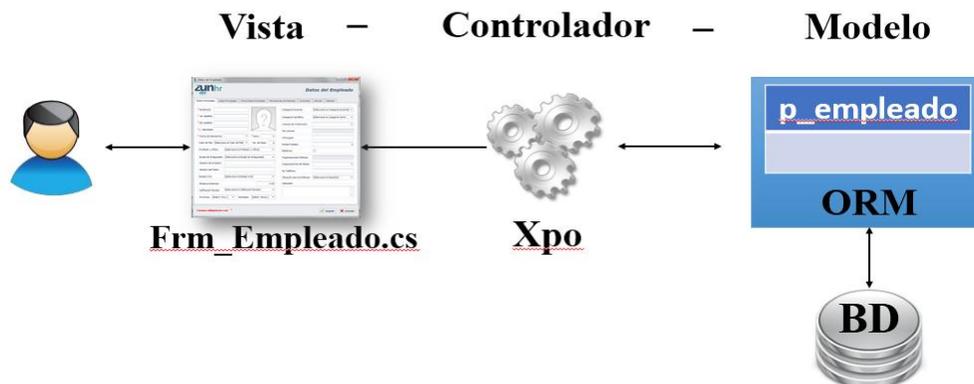


Figura 15. Representación del uso del patrón MVC en el sistema.

En el módulo desarrollado el patrón MVC se manifiesta de la forma siguiente: las clases del componente Xpo¹⁰ son la representación de la capa controlador atendiendo las peticiones realizadas a la aplicación a través de las vistas, las cuales se representan por formularios que serán mostrados al usuario con datos que provienen de una solicitud a

¹⁰ XPO es un componente del paquete DevExpress. Abstrae la capa de base de datos, manteniendo por completo el ámbito orientado a objetos. Resuelve una amplia gama de tareas de la lógica del negocio.

la capa del modelo que está conformada por las clases de acceso a datos del modelo ORM¹¹.

2.6.2 Diagrama de clases del diseño

Describe gráficamente las especificaciones de las clases de *software* y las interfaces en la aplicación. Es una representación visual del interior del módulo. Modela los formularios, sus enlaces. Además de las capas del MVC y su relación con el componente de DevExpress.

Se muestra una representación del diagrama de clases del diseño (DCD) del Paquete Empleado. El otro se encuentra en el [Anexo 4](#).

¹¹ Mapeado de objetos relacionales.

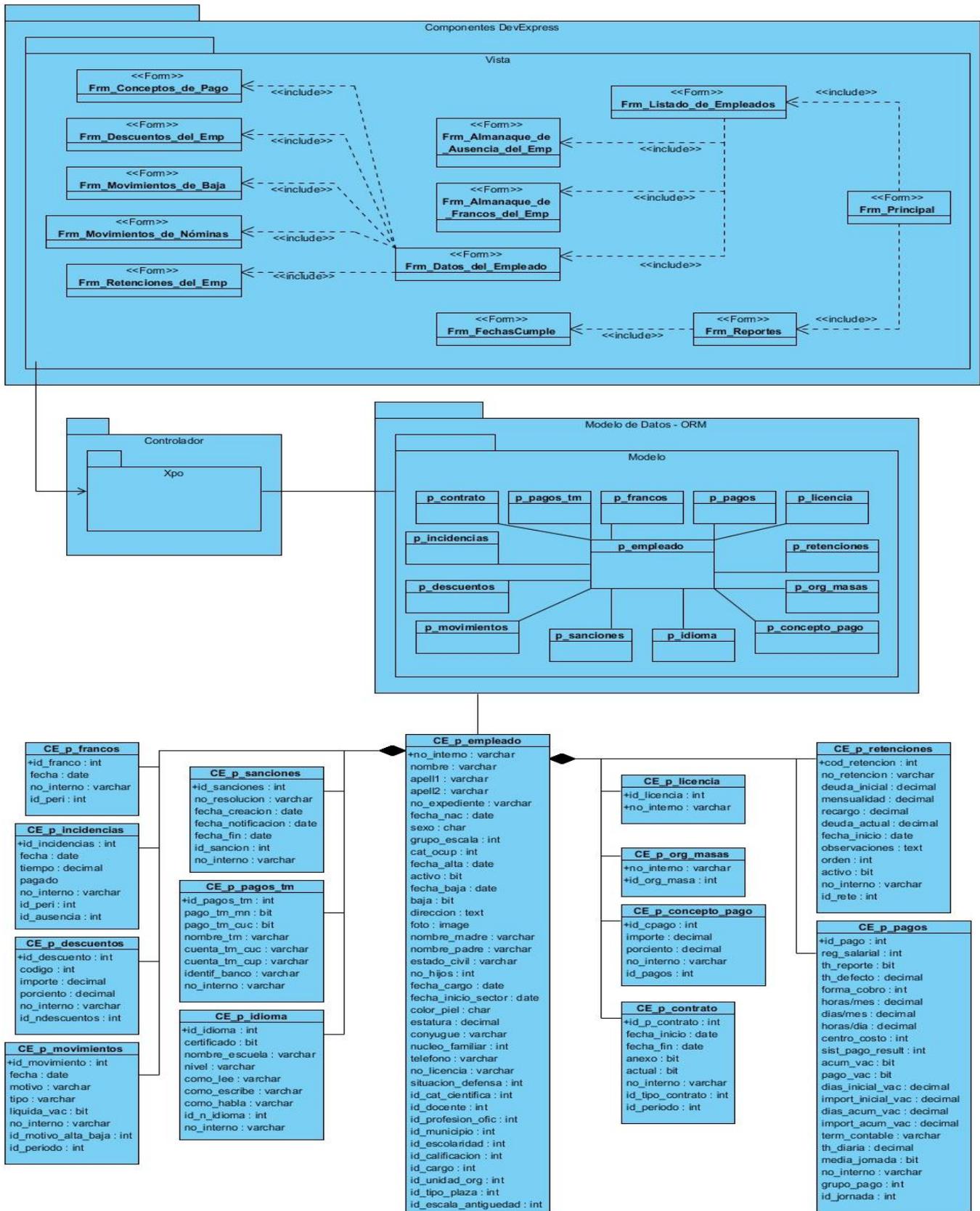


Figura 16. DCD Paquete Empleado.

2.7 Diseño de la base de datos

Una vez obtenido el diseño de todas las clases, se definió el modelo de datos para representar todas las clases persistentes necesarias para acceder a la base de datos y es que precisamente los modelos de datos aportan la base conceptual para diseñar aplicaciones que hacen un uso intensivo de datos. Estos son usados para describir la representación lógica y física de la información persistente manejada por el sistema, por lo que, al desarrollar un correcto diseño de la base de datos se puede garantizar la obtención de un acceso fiel a la información sin ningún tipo de inconveniente y también nos facilita el trabajo a la hora de almacenar los datos, reconocer el contenido y recuperar información. A continuación se presentan las entidades y campos relevantes del modelo de datos propuesto para el sistema de gestión de personal.

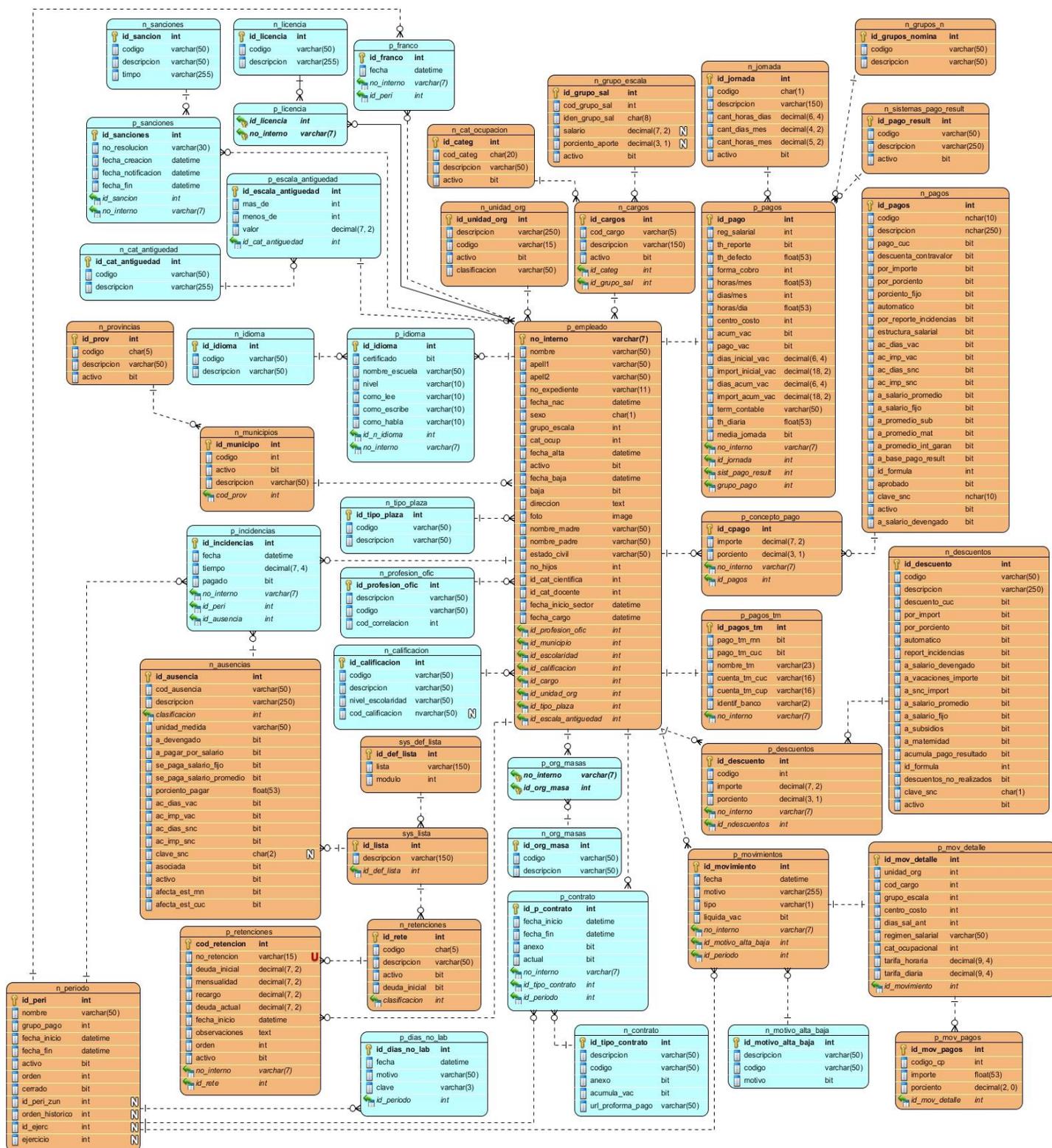


Figura 17. Diagrama de la base de datos del sistema propuesto.

2.7.1 Descripción de las entidades relevantes

Se muestra la descripción de la tabla p_empleado. El resto se encuentra en el [Anexo 3](#).

p_empleado		
Descripción: En la siguiente tabla se agrupa la información existente de un empleado.		
Atributo	Tipo	Descripción
no_interno	VARCHAR(7)	Etiqueta única que identifica al objeto (p_empleado) en la tabla.
nombre	VARCHAR(50)	Almacena el nombre del empleado.
apell1	VARCHAR(50)	Almacena el primer apellido del empleado.
apell2	VARCHAR(50)	Almacena el segundo apellido del empleado.
no_expediente	VARCHAR(11)	Almacena el número del expediente del empleado, el cual no es más que su carnet de identidad.
fecha_nac	DATETIME	Almacena la fecha de nacimiento del empleado.
sexo	CHAR(1)	Almacena el sexo del empleado.
grupo_escal	INT	Almacena el grupo escala del empleado.
cat_ocup	INT	Almacena la categoría ocupacional del empleado.
activo	BIT	Etiqueta que indica el estado del empleado.
fecha_alta	DATETIME	Almacena la fecha de alta del empleado.
baja	BIT	Etiqueta que indica si el empleado es baja o no.
dirección	TEXT	Almacena la dirección particular del empleado.
foto	IMAGE	Almacena la imagen personal del empleado.
nombre_madre	VARCHAR(50)	Almacena el nombre de la madre del empleado.
nombre_padre	VARCHAR(50)	Almacena el nombre del padre del empleado.
estado_civil	VARCHAR(50)	Almacena el estado civil del empleado.
no_hijos	INT	Almacena el número de hijos del empleado.
id_cat_cientifica	INT	Etiqueta extranjera que identifica al objeto (cat_cientifica) en la tabla.
id_cat_docente	INT	Etiqueta extranjera que identifica al objeto (cat_docente) en la tabla.
fecha_inicio_sector	DATETIME	Almacena la fecha de inicio en el sector del empleado.
fecha_cargo	DATETIME	Almacena la fecha en que ocupa determinado cargo el empleado.

id_profesion_ofic	INT	Etiqueta extranjera que identifica al objeto (profesion_ofic) en la tabla.
id_municipio	INT	Etiqueta extranjera que identifica al objeto (municipio) en la tabla.
id_escolaridad	INT	Etiqueta extranjera que identifica al objeto (escolaridad) en la tabla.
id_clasificacion	INT	Etiqueta extranjera que identifica al objeto (clasificacion) en la tabla.
id_cargo	INT	Etiqueta extranjera que identifica al objeto (cargo) en la tabla.
id_unidad_org	INT	Etiqueta extranjera que identifica al objeto (unidad_org) en la tabla.
id_tipo_plaza	INT	Etiqueta extranjera que identifica al objeto (tipo_plaza) en la tabla.
id_escalas_antiguedad	INT	Etiqueta extranjera que identifica al objeto (escalas_antiguedad) en la tabla.

Tabla 3. Descripción de la tabla p_empleado.

2.8 Diagrama de despliegue

El diagrama de despliegue es un tipo de diagrama UML que se emplea para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes, es decir, la arquitectura en tiempo de ejecución del sistema respecto al hardware y al *software*. Los componentes que participan en estos diagramas pueden ser los siguientes:

- Procesadores: nodos que tienen capacidad de procesamiento (computadoras por lo general).
- Dispositivos: nodos que no tienen capacidad de procesamiento (impresoras por lo general).
- Protocolos: estándares que deben estar implementados en la red, para efectuar la comunicación entre los diferentes elementos.

El diagrama de despliegue realizado representa tres nodos principales, el nodo PC Cliente, el nodo Servidor de Bases de Datos y el nodo dispositivo Impresora el cual nos va a permitir imprimir los reportes y demás información que sea necesaria para algún directivo del centro o un usuario del sistema.

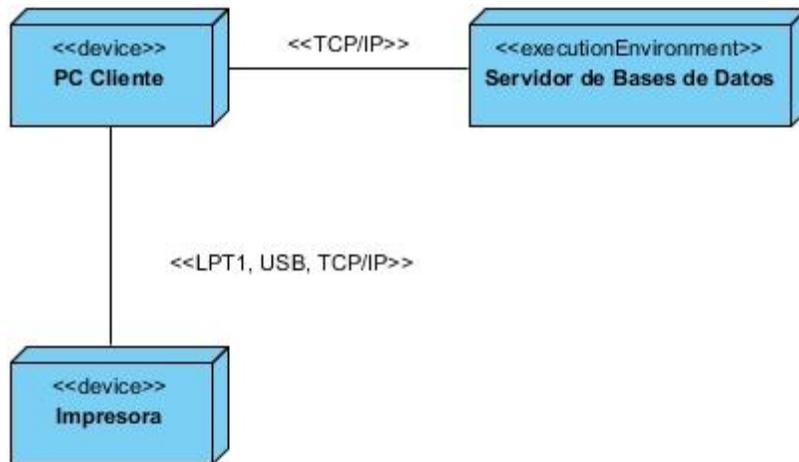


Figura 18. Diagrama de despliegue.

La PC cliente debe contar con los siguientes programas: Sistema Operativo Windows y Framework .Net en su versión 4.0 o superior.

El servidor de la base de datos debe contar con los siguientes programas: Sistema Operativo Windows y Microsoft SQL Server 2005.

2.9 Conclusiones parciales del capítulo

El análisis y diseño detallado de las características del sistema, propició una documentación que mantiene al equipo de desarrollo centrado en los requerimientos funcionales y no funcionales de la aplicación a través de los casos de uso arquitectónicamente significativos, después de obtener los eventos y procesos que ocurren en el entorno donde será implantado el mismo, logrando un lenguaje común entre todos los involucrados mediante la descripción de la propuesta. Además el modelado de los diagramas de clases del análisis ayudó a tener un mayor entendimiento del sistema, el cual dio paso a la realización del diagrama de diseño que es el artefacto principal de salida de esta fase de análisis y diseño. Al iniciar la fase de implementación este le proporciona al programador una visión más clara de lo que se desea realizar en el módulo y estar más enfocado al cumplimiento de los requerimientos funcionales del sistema. Asimismo, el diseño de la base de datos permitió que la información persistente de larga y corta vida tenga un almacén donde se pueda mantener intacta.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

3.1 Introducción

En el proceso de implementación será necesario guiarse por el análisis y diseño realizado al módulo, utilizando como artefacto principal el diagrama de clases del diseño. Durante este proceso el programador realizará todos los requerimientos funcionales y en su desarrollo también tendrá en cuenta los no funcionales para así llegar al producto final. Una vez terminado el *software* se entregará al equipo de calidad del GET para que le realice las pruebas pertinentes al sistema, lo que arrojará su nivel de calidad y permitirá hacerle mejoras al mismo, si es necesario.

3.2 Modelo de implementación

El proceso de implementación es la tarea fundamental durante la fase de construcción, tomando como punto de partida el resultado obtenido durante el flujo de trabajo de análisis y diseño. Durante esta etapa se crea la arquitectura, se estructura el sistema final, se crea el código, se implementan los elementos del diseño, y se integran todos estos para lograr un resultado satisfactorio que responda a lo previamente definido durante las fases anteriores. Al realizar el modelo de implementación se construye la estructura de los elementos de implementación, basándose en las responsabilidades asignadas a los subsistemas de implementación y su contenido.

3.3 Diagrama de componentes

El diagrama de componentes describe la descomposición física del sistema en componentes, a efectos de construcción y funcionamiento. Tal descomposición se realiza en términos de componentes y de relaciones entre los mismos. Estos identifican objetos físicos que hay en tiempos de ejecución, de compilación o de desarrollo, y tienen identidad propia y una interfaz bien definida. Cada componente incorpora la implementación de ciertas clases del diseño del sistema. (Ulises Leon , 2012)

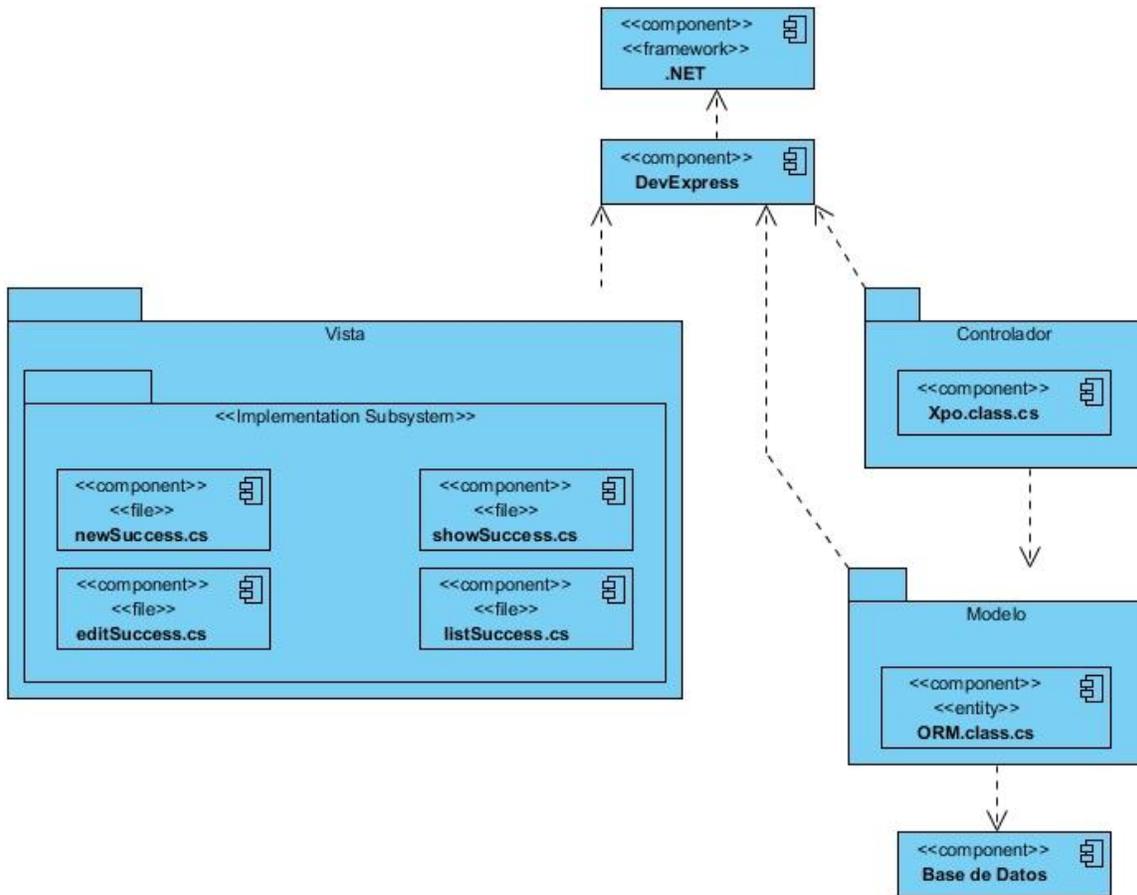


Figura 19. Diagrama de Componentes.

3.4 Estándares de código

Un estándar de codificación comprende todos los aspectos de la generación de código. Durante el desarrollo del sistema se estableció por el desarrollador un estándar de codificación para lograr un estilo armonioso en el código. Este consiste en que el primer carácter de todas las palabras se escribe en mayúsculas y los otros caracteres en minúsculas, anteceditos por un prefijo en caso de referirse a un componente (dígase botón, campo de texto, etc.) según corresponda y separadas por un guion bajo (_). Este estándar se aplicó en la nomenclatura de los componentes usados para conformar la parte visual del sistema de la manera que se muestra a continuación.

Ejemplo: Frm_Datos_del_Empleado (para el caso del formulario datos del empleado), btn_Aceptar (para el caso del botón aceptar), txt_Nombre (para el caso del campo de texto que contendrá el nombre del empleado).

Control	Prefijo
TextBox	txt
DateEdit	dt
ComboBoxEdit	cb
CheckEdit	chk
SimpleButton	btn
GroupBox	gb
XtraTabControl	xtraTabC
XPCollection	xpC
XPDataView	xpDataV
GridControl	gridC
GridView	gridV
XRTableCell	xrTableC
XtraReport	Rpt
XtraForm	Frm
ToolStrip	toolS

Tabla 4. Prefijos usados en la nomenclatura de los componentes.

Para el caso de la nomenclatura de eventos de los componentes y los métodos, quedarían con el nombre del componente seguido de un guion bajo y luego el nombre del evento, tal y como se muestra a continuación.

```
private void txt_No_Interno_Leave(object sender, EventArgs e)...
```

```
private void ckb_Paga_Tarjeta_CUP_CheckedChanged(object sender, EventArgs e)...
```

```
private void ckb_Paga_Tarjeta_CUC_CheckedChanged(object sender, EventArgs e)...
```

```
private void cb_Provincia_EditValueChanged(object sender, EventArgs e)...
```

```
private void ckb_Militancia_CheckedChanged(object sender, EventArgs e)...
```

Figura 20. Representación de la aplicación del estándar de código en los eventos.

```
private void Set_Salario_Escala()...  
private void Conceptos_Pago_Aprobados()...  
private void Habilitar_Campos_Sanciones()...  
private void Validar_CI(string i)...  
private bool Validar_Campos_Obligatorios()...
```

Figura 21. Representación de la aplicación del estándar de código en los métodos.

3.5 Pruebas de software

El proceso de pruebas es el medio apropiado para determinar el nivel de la calidad de un producto de *software*. Durante su transcurso se realizan pruebas dirigidas a sus componentes o al sistema en su totalidad, con el objetivo de evaluar el grado en que la aplicación cumple con los requerimientos. Existen diferentes niveles de pruebas aplicadas para disímiles tipos de objetivos, en diferentes escenarios o niveles de trabajo; entre las utilizadas para probar el módulo desarrollado están:

Prueba de desarrollador: Es la prueba diseñada e implementada por el equipo de desarrollo. Tradicionalmente estas pruebas han sido consideradas solo para la prueba de unidad, aunque en la actualidad, en algunos casos, se pueden ejecutar pruebas de integración.

Prueba de rendimiento: Se centra en determinar la velocidad con la que el sistema realiza una tarea en las condiciones particulares del escenario de pruebas. Este tipo de prueba abarca a su vez un grupo de “*tests*” dentro de los que se encuentran:

- **Test de rendimiento:** El objetivo es determinar si el usuario estará satisfecho con la velocidad de la aplicación, bajo condiciones de uso esperadas durante el “día a día”.
- **Test de carga:** El objetivo de esta prueba es determinar el “rendimiento” necesario para que el sistema funcione en hora.
- **Test de capacidad:** En este caso, se busca dar una estimación de hasta dónde se puede llegar cargando el sistema antes de que sea inutilizable (por ejemplo, un formulario que responde a peticiones en 15 segundos como promedio, funciona correctamente, sin embargo, no es óptimo para el usuario).

- **Test de estrés:** Conocidas como “*stress testing*”, tienen como objetivo obtener datos sobre la carga del sistema, que ayuden a realizar el dimensionamiento del sistema o “*capacity planning*”. Este *test* genera carga en el sistema hasta hacerlo inutilizable. Una vez que la aplicación ha dejado de funcionar, la atención se centra en distintos objetivos, como por ejemplo: verificar la calidad de los mensajes de error del sistema o establecer alertas para poder anticipar un fallo total del sistema. Este tipo de *test* es uno de los últimos que se debe ejecutar, ya que, por su carácter poco realista, podría darse el caso de que la situación de carga simulada nunca se diera en la vida real. (Globetesting S.L.U., 2011)

Pruebas funcionales: Prueba que una funcionalidad esté completa, donde pueden estar implicadas una o varias clases, y además la propia interfaz de usuario.

Pruebas de aceptación: Son pruebas funcionales, pero vistas directamente desde el cliente. Digamos que son aquellas pruebas que demuestran al cliente que la funcionalidad está terminada y funciona correctamente.

Pruebas de función: En esta prueba se fija la atención en la validación de las funciones, métodos, y casos de uso.

Pruebas de seguridad: Esta prueba asegura que los datos y el sistema, solamente sean accedidos por los actores deseados.

Pruebas de caja negra: Este método de pruebas se centra en los requisitos funcionales del *software* y permite obtener entradas que prueben todos los requisitos del sistema. Esta prueba se limita a que se pruebe con “datos” la entrada y se estudie la salida, sin preocupar los procesos internos. (Globetesting S.L.U., 2011)

Con este tipo de pruebas se encuentran:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en la estructura de datos o en el acceso a las bases de datos externas.
- Errores de rendimiento, entre otros.

3.6 Resultados obtenidos

Al módulo desarrollado se le han ejecutado 2 iteraciones de pruebas de **caja negra** guiadas por los diseños de CU, puesto que el grupo de *Testing y Calidad* del cliente realizaría los casos de prueba. Las mismas arrojaron las siguientes No Conformidades (NC) que propiciaron un aumento de su nivel de calidad.

Tipo de no conformidad	Iteración 1	Iteración 2
Formato (A y D)		
Error técnico (D)		
Correspondencia con otro artefacto (A y D)	2	1
Ortografía (A y D)	1	
Redacción (A y D)	2	
Validación (A)	3	1
Opciones que no funcionan (A)	1	
Error de interfaz (A)		
Error idioma (A)		
Funcionalidad (A)	5	1
Excepciones (A)	1	
Otros (A y D)		
Seguridad (A)		
Recomendación (A y D)		
Total	15	3

Tabla 5. Descripción de NC.

Una vez entregado e instalado el módulo desarrollado, se realizó una **prueba de rendimiento**, analizando los factores de tiempo y calidad en la realización de reportes, tomando como muestra el reporte Prenómina, lo que arrojó los siguientes resultados.

La especialista para realizar este reporte de la manera en que lo hacía anteriormente, partiendo del punto en que poseía el análisis periódico de asistencias, tuvo que realizar el cálculo de los tiempos de cada tipo de incidencias para cada empleado, después de totalizarlos y calcular el fondo real trabajado, procedió al llenado del reporte manualmente; el tiempo total empleado en este proceso fue de 4 horas.

Con la utilización del sistema, una vez que tenía el análisis periódico de asistencia, se dirigió a la pestaña de reportes, seleccionó el período y de inmediato generó automáticamente el reporte deseado, tomándole un tiempo de 1 minuto.

Se demuestra, a partir de los resultados obtenidos, que el sistema reduce considerablemente el tiempo de realización del reporte, generado con calidad de ambas formas dada la experiencia que la especialista posee en el tema y el cumplimiento estricto del formato del modelo del reporte.

3.7 Conclusiones parciales

Mediante la construcción del diagrama de componente se incrementó el entendimiento de la descomposición física del sistema en componentes, a efectos de construcción y funcionamiento, lo que permitió que en el proceso de implementación se obtuviera como resultado un módulo para la gestión de personal que garantiza la integración con el módulo de Nóminas y con el resto de los programas de la Suite ZUN, y que responde a las necesidades del cliente. Todo ello fue comprobado durante el proceso de validación del software, lo que contribuyó a incrementar la calidad del mismo al hacerlo más confiable y seguro, una vez eliminadas las fallas detectadas.

CONCLUSIONES

Finalizado el proceso investigativo y el desarrollo de un *software* capaz de gestionar el personal dentro de la Suite ZUN, se puede arribar a una serie de conclusiones que a continuación se exponen:

- El estudio de los diferentes sistemas similares existentes en Cuba y el mundo, así como la sistematización y el análisis integral de los elementos referidos a un *software* para la gestión de recursos humanos, las metodologías y tecnologías seleccionadas, tanto como de las concepciones teóricas que la sustentan, constituyen un punto de partida para llevar a cabo el desarrollo de un *software* capaz de gestionar el personal.
- Se describieron las características del sistema, realizándose las fases de inicio, elaboración y construcción correspondientes a la metodología seleccionada, lo que permite un correcto y organizado desarrollo para el posterior proceso de implementación.
- Luego de realizarse el proceso de implementación se obtuvo un *software* capaz de llevar a cabo la gestión de personal dentro de la Suite, y con la realización de pruebas se comprobó que se cumplieron con todas las necesidades especificadas por el cliente.

RECOMENDACIONES

Tomando como punto de partida los resultados obtenidos con la realización de este trabajo de diploma, se hacen las siguientes recomendaciones:

- Añadir funcionalidad que permita gestionar las etiquetas de las proformas del contrato, para así lograr que el sistema sea más extensible y personalizable.
- Adicionarle a la aplicación los procesos de gestión de la plantilla de cargos de la entidad, para lograr que el sistema trabaje de acuerdo a las disponibilidades de plazas existentes.
- Extender la capacidad de gestión de procesos del sistema, adicionándoles los de Capacitación, Protección y Salud.

REFERENCIAS BIBLIOGRÁFICAS

1. Bligoo. 2014. Bligoo. [En línea] 2014. [Citado el: 31 de Enero de 2014.] <http://programacion3.bligoo.com/content/view/3552616/Que-es-lenguaje-Java-y-sus-caracteristicas.html>.
2. Cases, D. 2014. Programación, diseño web y algo más. [En línea] 2014. [Citado el: 31 de Enero de 2014.] <http://www.dcases.com/61/programacion-en-c-compiladores-y-entornos-de-desarrollo/>.
3. Crece Negocios. 2014. Crece Negocios. [En línea] 2014. [Citado el: 30 de Enero de 2014.] <http://www.crecenegocios.com/concepto-y-funciones-del-area-de-recursos-humanos/>.
4. D'MARCO S.A. 2013. ASSETS: Sistema de Gestión Integral. [En línea] 2013. [Citado el: 13 de Enero de 2014.] <http://www.assets.co.cu/assets.asp>.
5. Datazucar. 2014. Datazucar. Aseguramos su éxito. [En línea] 2014. [Citado el: 2014 de Enero de 14.] <http://www.datazucar.cu/?p=1>.
6. e-conomic. 2014. e-conomic. [En línea] 2014. [Citado el: 10 de Febrero de 2014.] <http://www.e-conomic.es/programa/glosario/definicion-nomina>.
7. Globetesting S.L.U. 2011. Globe Testing. [En línea] 2011. [Citado el: 4 de Mayo de 2014.] <http://www.globetesting.com/>.
8. González Blanco, Rubén. Pérez Tobalina, Sergio. 2014. LESE-2-Introduccion a Rational Rose. [En línea] 2014. [Citado el: 30 de Enero de 2014.] <http://www.essi.upc.edu/~es-e/web/documents/lab/0304Q2/lessons/lese-2/LESE-2%20-%20Introduccion%20a%20Rational%20Rose.ppt>.
9. Hernández Torruco, José , y otros. 2010. *Avances en Informática y Sistemas Computacionales. Tomo V.* 2010. 978-607-7557-71-5.
10. ICE HRM. [En línea] [Citado el: 8 de Diciembre de 2013.] <http://icehrm-blog.gamonoid.com/>.
11. J. Myers, Glenford. 2004. *The Art of Software Testing. Second Edition.* Hoboken, New Jersey : Jhon Wiley & Sons, Inc., 2004.
12. Jacobson, I., Rumbaugh, G. y Booch, J. 2004. *El Proceso Unificado de Desarrollo de Software.* La Habana : Felix Varela, 2004.

13. La Hoz Pavón, Lizandra y García Sánchez, Adrián. 2011. *Desarrollo del módulo Ejercicios para la plataforma educativa ZERA basado en estándares para la interoperabilidad*. La Habana : s.n., 2011.
14. La Revista Informática. 2014. La Revista Informática.com. [En línea] 2014. [Citado el: 16 de Enero de 2014.] <http://www.larevistainformatica.com/LENGUAJES-DE-PROGRAMACION-listado.html>.
15. La Revista Informatica.com. [En línea] [Citado el: 30 de Enero de 2014.] <http://www.larevistainformatica.com/Clipper.htm>.
16. Meltom Technologies. 2013. de Gerencia.com. [En línea] 2013. [Citado el: 10 de Febrero de 2014.] http://www.degerencia.com/tema/recursos_humanos.
17. Mi Tecnológico. 2014. Mi Tecnológico. [En línea] 2014. [Citado el: 31 de Enero de 2014.] <http://www.mitecnologico.com/Main/DefinicionDeLenguajeDeProgramacion..>
18. Microsoft. 2010. Introducción a Visual Studio. [En línea] 2010. [Citado el: 31 de Enero de 2014.] [http://msdn.microsoft.com/es-es/library/6x6bk1f4\(v=vs.90\).aspx](http://msdn.microsoft.com/es-es/library/6x6bk1f4(v=vs.90).aspx).
19. MicroTecnologías. 2009. MicroTecnologías. *Servicios Computacionales*. [En línea] 2009. [Citado el: 31 de Enero de 2014.] <http://microtecnologias.wordpress.com/2009/12/18/novell-presento-mono-2-6-y-monodevelop-2-2/>.
20. Murillo Alfaro, Felix. 2014. Oficina Nacional de Gobierno Electronico e Informática - ONGEI. [En línea] 2014. [Citado el: 31 de Enero de 2014.] <http://www.ongei.gob.pe/publica/metodologias/Lib5083/cap2062.HTM>.
21. Navarro Franco, Jose Angel . 2009. *CONSIDERACIONES SOBRE LA ESTRUCTURACIÓN EN CAPAS DE UN SISTEMA SOFTWARE SEGÚN EL PROCESO UNIFICADO DE RUP*. Facultad Informática CUJAE, La Habana : s.n., 2009.
22. OrangeHRM. 2013. OrangeHRM :: Free Human Resource Management Software. [En línea] 2013. [Citado el: 6 de Diciembre de 2013.] <http://www.orangehrm.com/>.
23. PostgreSQL Global Development. 2014. PostgreSQL:The world's most advanced open source database. [En línea] 2014. <http://www.postgresql.org/about/>.
24. Rosenberg, D. y Scott, K. 2001. *Applying Use Case Driven Object Modeling with UML An Annotated E-Commerce Example*. 2001.
25. —. 1999. *Modelado de Objetos dirigido por Casos de Uso con UML: Un enfoque práctico*. 1999.
26. Rosenberg, D. y Stephens, M. 2007. *Use Case Driven Object Modeling with UML Theory and Practice*. 2007.

27. Ruiz, Alberto. 2009. Observatorio Tecnológico. [En línea] 2009. [Citado el: 31 de Enero de 2014.] <http://recursostic.educacion.es/observatorio/web/eu/software/programacion/911-monografico-java>.
28. Rumbaugh, James, Jacobson, Ivar y Booch, Grady. 1999. *El Lenguaje Unificado de Modelado*. s.l. : Eddison Wesley, 1999.
29. S. Pressman, Roger. 2010. *Software Engineering. A Practitioner's Approach. Seventh Edition*. 2010. 978-0-07-337597-7.
30. Sergio Orozco. 2014. Abiztar. [En línea] 2014. [Citado el: 23 de Abril de 2014.] http://www.milestone.com.mx/articulos/casos_a_incluir_casos_a_extender.htm.
31. Sparx. 2014. Sparx Systems. [En línea] 2014. [Citado el: 31 de Enero de 2014.] <http://www.sparxsystems.com.ar/products/ea.html>.
32. Techopedia. 2014. Techopedia. [En línea] 2014. [Citado el: 31 de Enero de 2014.] <http://www.techopedia.com/definition/1243/sql-server>.
33. Tedeschi, Nicolás. 2014. Microsoft. Developer Network. [En línea] 2014. [Citado el: 5 de Mayo de 2014.] <http://msdn.microsoft.com/es-es/library/bb972240.aspx#XSLTsection122121120120>.
34. Tinoco Gómez, Oscar . Rosales López, Pedro Pablo. Salas Bacalla, Julio. 2013. Criterios de selección de metodologías de desarrollo de software. [En línea] 2013. [Citado el: 15 de Enero de 2014.] <http://www.redalyc.org/articulo.oa?id=81619984009>.
35. Ulises leon . 2012. Buenas Tareas. [En línea] 2012. [Citado el: 20 de Abril de 2014.] <http://www.buenastareas.com/ensayos/Modelo-De-Implementacion/6820192.html>.
36. unillanosGSW. 2014. unillanosGSW. [En línea] 2014. [Citado el: 31 de Enero de 2014.] <http://unillanosgs.wikispaces.com/Entornos+de+desarrollo+de+aplicaciones+web+integrados>.
37. Visual Paradigm. 2014. Visual Paradigm. [En línea] 2014. [Citado el: 31 de Enero de 2014.] <http://www.visual-paradigm.com/product/vpuml/editions/modeler.jsp..>
38. White Town Software. White Town Software. [En línea] [Citado el: 30 de Enero de 2014.] <http://www.whitetown.com/es/misc/dbf-format/>.
39. WordReference.com. 2014. WordReference.com. [En línea] 2014. [Citado el: 30 de Enero de 2014.] <http://es.thefreedictionary.com/n%C3%B3mina>.

GLOSARIO

A

Actor del sistema: Terceros fuera del sistema que interactúan con él.

C

Capacitación: Conjunto de procesos organizados, dirigidos a complementar la preparación de un rol para su correcto desempeño en la organización.

Capital Humano: Conjunto de conocimientos, experiencias, habilidades, sentimientos, actitudes, motivaciones, valores y capacidad para hacer, portados por los trabajadores para crear más riquezas con eficiencia.

Caso de Uso: Descripción narrativa textual de la secuencia de eventos y acciones que ocurren cuando un usuario parte o divide en un diálogo con un sistema durante un proceso significativo.

Clase: Contenedor de uno o más datos (variables o propiedades miembro) junto a las operaciones de manipulación de dichos datos (funciones/métodos).

Cliente-Servidor: Tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro

recurso del grupo de trabajo a través de la organización en múltiples plataformas.

Crystal Reports: Estándar de elaboración de informes de Visual Studio que brinda la posibilidad de crear contenido interactivo con calidad de presentación al entorno de Windows. Permite crear informes complejos y profesionales.

D

DevExpress: Conjunto de componentes para Entornos de Desarrollos Integrados (IDE) que incluyen una amplia gama de componentes visuales que le ayudan a emular interfaces de usuario más populares fácilmente o a producir gráficos o informes de gran calidad.

Dominio: Área de conocimiento o actividad caracterizada por un conjunto de conceptos y terminología.

E

Empleado: es aquella persona que ocupa un puesto de trabajo en la entidad y desempeña una labor determinada por la cual recibe haberes o salarios.

Estándar: Acuerdo documentado que contiene especificaciones técnicas u otros criterios específicos para ser usado como referencia, guía o definición de características, para asegurar qué materiales, productos, procesos y servicios, sean obtenidos o hayan sido realizados de acuerdo a sus propósitos.

F

Fichero: Conjunto de información que se almacena en algún medio de escritura que permita ser leído o accedido por una computadora (pc). Es identificado por un nombre y la descripción de la carpeta o directorio que lo contiene y facilita una manera de organizar los recursos usados para almacenar permanentemente información dentro de una pc.

Foxpro: Lenguaje de programación orientado a objetos, que a la vez es un Sistema Gestor de Bases de Datos (SGBD), publicado originalmente por Fox *Software* y posteriormente por Microsoft, para los sistemas operativos MS-DOS, MS Windows, Mac OS y UNIX.

Framework: Estructura de soporte definida que simplifica el desarrollo de una aplicación mediante la automatización de algunos de los

patrones utilizados para resolver tareas comunes tales como: proporcionar una estructura al código fuente y facilitar la programación de aplicaciones, mediante el encapsamiento de operaciones complejas en instrucciones sencillas.

H

Hipertexto: Escritura no secuencial que, mediante la existencia de una serie de bloques de texto conectados entre sí por nexos (links), permite al usuario establecer una multiplicidad de itinerarios de acceso y ampliar de modo significativo las posibilidades de lectura en una pantalla interactiva.

M

Módulo: Componente auto controlado de un sistema, el cual posee una interfaz bien definida hacia otros componentes. De las varias tareas que debe realizar un programa para cumplir con su función u objetivos, un módulo realizará una parte de dichas tareas (o quizás varias en algún caso).

Movimiento: Operación que se le realiza a la fuerza de trabajo, ya sean altas, bajas o modificaciones. Permite realizar cálculos y estudios de tendencia sobre fluctuación, índices de rotación y las fuentes de reclutamiento de la fuerza de trabajo.

N

No conformidad: Desfasaje entre lo real y lo planificado.

Nomenclador: Es una especie de catálogo que tiene la nomenclatura, o vocabulario de una rama.

O

ORM: (Mapeo de Objetos Relacionales) Técnica de programación que permite generar clases a través de las tablas de la base de datos.

P

Proforma de contrato: Documento de ejemplo que sustenta un formato y describe los lineamientos de un tipo de contrato determinado para la elaboración de contratos similares.

S

Salario: parte del producto nacional que se distribuye a los trabajadores de forma individual, atendiendo a la cantidad y calidad del trabajo aportado, según las condiciones económicas de cada momento histórico. Comprende lo percibido por el trabajador, por rendimiento, unidad de tiempo, pagos adicionales, trabajo extraordinario,

laborar en día de conmemoración nacional y feriados y vacaciones anuales pagadas.

Salario escala: Salario básico que cobra el trabajador sin los pagos adicionales ni otros pagos.

Smalltalk: Sistema informático que permite realizar tareas de computación mediante la interacción con un entorno de objetos virtuales. Metafóricamente, se puede considerar que un Smalltalk es un mundo virtual donde viven objetos que se comunican mediante el envío de mensajes.

Software: Equipamiento lógico o soporte lógico de un ordenador. Comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema (hardware). Suele ser utilizado para referirse a los programas y aplicaciones informáticas.

Symfony: Framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web.

ANEXOS

Anexo 1. Guía de preguntas realizadas en la entrevista.

Objetivo: Identificar las necesidades del cliente para determinar las principales funcionalidades de la aplicación.

Preguntas realizadas:

- ¿Cuántas personas realizan el trabajo de personal?
- ¿Qué roles desempeñan esas personas?
- ¿Qué actividades realiza cada rol?
- ¿Qué datos desean recoger de un empleado?
- ¿Qué nomencladores desean gestionar?
- ¿Qué reportes desean que genere el sistema?
- ¿Qué información del sistema de nóminas utilizan en los procesos de personal?

Anexo 2. Descripciones de los casos de uso.

Paquete Empleados

CU Listar Empleados

Especificación del Caso de Uso

Objetivo	Listar los empleados.
Actores	Usuario (Inicia), generaliza a Especialista de RH: Lista los empleados.
Resumen	El caso de uso se inicia cuando el actor selecciona la opción que le permite listar los empleados, terminando así el caso de uso.
Precondiciones	Debe haberse generado el escritorio de trabajo del usuario autenticado.
Poscondiciones	Se listaron los empleados por el actor.
Flujo de eventos	

Flujo básico		
	Acciones del Actor	Acciones del Sistema
1.	El caso de uso se inicia cuando el actor selecciona la opción de Listar Empleados.	
2.		<p>Permite mostrar el listado de todos los empleados registrados en el sistema y además brinda la posibilidad de:</p> <ul style="list-style-type: none"> • Imprimir el listado de empleados. • Agregar días de ausencias al empleado seleccionado, ver <i>CU Gestionar Ausencias</i>. • Agregar días de franco al empleado seleccionado, ver <i>CU Gestionar Francos</i>. • Permite añadir, modificar y eliminar un empleado, ver <i>CU Gestionar Empleado</i>.
3.		Termina el caso de uso.

CU Gestionar Retenciones

Especificación del Caso de Uso

Objetivo	Añadir, modificar y eliminar una retención al empleado.
Actor	Especialista de RH.
Resumen	El caso de uso se inicia cuando el actor selecciona la opción que le permite realizar una acción sobre una retención del empleado. El actor puede añadir, eliminar y modificar una retención del empleado. En caso de que seleccione la opción de añadir retención, el sistema dará la posibilidad de insertar los datos que se necesitan para llenar este formulario. Si el actor elige la opción de modificar la retención, el sistema mostrará los datos que pueden ser editables dentro de la retención, y una vez realizados los

	cambios, guardará las modificaciones. Si el actor elige la opción de eliminar la retención, el sistema permitirá eliminar la retención mostrando un mensaje de confirmación, terminando así el caso de uso.	
Precondiciones	<ul style="list-style-type: none"> • Debe haberse generado el escritorio de trabajo del usuario autenticado. • Debe haberse generado el <i>CU Gestionar Empleado</i>. • Para modificar la retención el actor debe ser el responsable temporal del mismo. • Para eliminar la retención el actor debe tener el permiso de eliminar el elemento. 	
Poscondiciones	Se añadió, modificó o eliminó la retención del empleado por el actor.	
Flujo de eventos		
Flujo básico		
	Acciones del Actor	Acciones del Sistema
1.	El caso de uso se inicia cuando el actor selecciona la pestaña <i>Otros Datos Principales</i> .	
2.		Permite mostrar el listado de todas las retenciones del empleado registradas en el sistema y además permite realizar varias acciones: <ul style="list-style-type: none"> • Añadir una nueva retención. • Modificar los datos de la entidad. Ver Sección 1: “Modificar datos de la retención”. • Eliminar la entidad. Ver Sección 2: “Eliminar la retención”.

3.	Selecciona la opción de Añadir una nueva retención.	
4.		<p>Brinda la posibilidad de introducir los datos de la retención:</p> <ul style="list-style-type: none"> • Número • Tipo de Retención • Orden de Prioridad • Deuda Inicial • Mensualidad • Recargo • Deuda Actual • Observaciones • Activo
5.	Introduce y selecciona los datos de la retención.	
6.		<p>Permite seleccionar la opción:</p> <ul style="list-style-type: none"> • Aceptar. • Cancelar la operación.
7.	Selecciona la opción de Aceptar.	
8.		<ul style="list-style-type: none"> • Valida los datos. • Crea la retención al empleado.
		Termina el caso de uso.
Flujo Alternativo		
7. a Selecciona la opción de Cancelar.		
	Acciones del Actor	Acciones del Sistema
8.		a.1 Elimina los datos creados.

		a.2 Regresa a la vista anterior. a.3 El caso de uso termina.
8. a Existen datos incompletos.		
	Acciones del Actor	Acciones del Sistema
8.		a.1 Sobre cada campo incompleto muestra el mensaje de error. a.2 Regresa al paso 4 del Flujo Básico.
8. b Existen datos incorrectos.		
	Acciones del Actor	Acciones del Sistema
8.		b.1 Sobre cada campo incorrecto muestra el mensaje de error. b.2 Regresa al paso 4 del Flujo Básico

Sección 1: “Modificar datos de la retención”.

Flujo Básico		
	Acciones del Actor	Acciones del Sistema
1.	Selecciona la opción de modificar los datos de la retención.	
2.		Muestra los datos editables de la retención y brinda la posibilidad de cambiar sus valores. Permite además: <ul style="list-style-type: none"> • Aceptar.

		<ul style="list-style-type: none"> • Cancelar la operación.
3.	Modifica los datos que necesite y selecciona la opción de Aceptar los mismos.	
4.		<ul style="list-style-type: none"> • Valida los datos. • Actualiza los datos de la retención.
5.		El caso de uso termina.
Flujo Alternativo		
3. a El actor selecciona la opción de Cancelar.		
	Acciones del Actor	Acciones del Sistema
4.		a.1 Elimina los datos creados. a.2 Regresa a la vista anterior. a.3 El caso de uso termina.
4. a Existen datos incompletos.		
	Acciones del Actor	Acciones del Sistema
		a.1 Sobre cada campo incompleto muestra el mensaje de información.
		a.2 Regresa al paso 2 del Flujo Básico.
4. b Existen datos incorrectos.		
	Acciones del Actor	Acciones del Sistema
		b.1 Sobre cada campo incorrecto muestra el mensaje de información.
		b.2 Regresa al paso 2 del Flujo Básico.

Sección 2: “Eliminar la retención”.

Flujo Básico		
	Acciones del Actor	Acciones del Sistema
1.	Selecciona la opción de eliminar la retención.	
2.		Muestra el mensaje de advertencia y permite: <ul style="list-style-type: none"> • Aceptar (Si). • Cancelar la operación (No).
3.	Selecciona la opción de Aceptar.	
4.		<ul style="list-style-type: none"> • Se elimina la retención. • Regresa a la vista anterior.
5.		El caso de uso termina.
Flujo Alterno		
3. a El actor selecciona la opción de Cancelar.		
	Acciones del Actor	Acciones del Sistema
4.		a.1 Regresa a la vista anterior. a.2 El caso de uso termina.

CU Gestionar Descuentos

Especificación del Caso de Uso

Objetivo	Añadir, modificar y eliminar un descuento del empleado.
Actor	Especialista de RH.
Resumen	El caso de uso se inicia cuando el actor selecciona la opción que le permite realizar una acción sobre un descuento del empleado. El actor puede

	añadir, eliminar y modificar un descuento del empleado. En caso de que seleccione la opción de añadir descuento, el sistema dará la posibilidad de insertar los datos que se necesitan para llenar este formulario. Si el actor elige la opción de modificar el descuento, el sistema mostrará los datos que pueden ser editables dentro del descuento, y una vez realizados los cambios, guardará las modificaciones. Si el actor elige la opción de eliminar el descuento, el sistema permitirá eliminar el descuento mostrando un mensaje de confirmación, terminando así el caso de uso.	
Precondiciones	<ul style="list-style-type: none"> • Debe haberse generado el escritorio de trabajo del usuario autenticado. • Debe haberse generado el <i>CU Gestionar Empleado</i>. • Para modificar el descuento, el actor debe ser el responsable temporal del mismo. • Para eliminar el descuento, el actor debe tener el permiso de eliminar el elemento. 	
Poscondiciones	Se añadió, modificó o eliminó el descuento del empleado por el actor.	
Flujo de eventos		
Flujo básico		
	Acciones del Actor	Acciones del Sistema
1.	El caso de uso se inicia cuando el actor selecciona la pestaña <i>Otros Datos Principales</i> .	
2.		Permite mostrar el listado de todos los descuentos de empleado registrados en el sistema y además permite realizar varias acciones: <ul style="list-style-type: none"> • Añadir un nuevo descuento.

		<ul style="list-style-type: none"> • Modificar los datos de la entidad. Ver Sección 1: “Modificar datos del descuento”. • Eliminar la entidad. Ver Sección 2: “Eliminar el descuento”.
3.	Selecciona la opción de Añadir un nuevo descuento.	
4.		Brinda la posibilidad de introducir los datos del descuento: <ul style="list-style-type: none"> • Descripción • Porcentaje • Importe
5.	Introduce los datos del descuento.	
6.		Permite seleccionar la opción: <ul style="list-style-type: none"> • Aceptar. • Cancelar la operación.
7.	Selecciona la opción de Aceptar.	
8.		<ul style="list-style-type: none"> • Valida los datos. • Crea el descuento al empleado.
		Termina el caso de uso.
Flujo Alternativo		
7. a Selecciona la opción de Cancelar.		
	Acciones del Actor	Acciones del Sistema
8.		a.1 Elimina los datos creados. a.2 Regresa a la vista anterior. a.3 El caso de uso termina.

8. a Existen datos incompletos.		
	Acciones del Actor	Acciones del Sistema
8.		a.1 Sobre cada campo incompleto muestra el mensaje de error. a.2 Regresa al paso 4 del Flujo Básico.
8. b Existen datos incorrectos.		
	Acciones del Actor	Acciones del Sistema
8.		b.1 Sobre cada campo incorrecto muestra el mensaje de error. b.2 Regresa al paso 4 del Flujo Básico

Sección 1: “Modificar datos del descuento”.

Flujo Básico		
	Acciones del Actor	Acciones del Sistema
1.	Selecciona la opción de modificar los datos del descuento.	
2.		Muestra los datos del descuento y brinda la posibilidad de cambiar sus valores. Permite además: <ul style="list-style-type: none"> • Aceptar. • Cancelar la operación.
3.	Modifica los datos que necesite y selecciona la opción de Aceptar los	

	mismos.	
4.		<ul style="list-style-type: none"> • Valida los datos. • Actualiza los datos del descuento.
5.		El caso de uso termina.
Flujo Alterno		
3. a El actor selecciona la opción de Cancelar.		
	Acciones del Actor	Acciones del Sistema
4.		a.1 Elimina los datos creados. a.2 Regresa a la vista anterior. a.3 El caso de uso termina.
4. a Existen datos incompletos.		
	Acciones del Actor	Acciones del Sistema
		a.1 Sobre cada campo incompleto muestra el mensaje de información.
		a.2 Regresa al paso 2 del Flujo Básico.
4. b Existen datos incorrectos.		
	Acciones del Actor	Acciones del Sistema
		b.1 Sobre cada campo incorrecto muestra el mensaje de información.
		b.2 Regresa al paso 2 del Flujo Básico.

Sección 2: “Eliminar el descuento”.

Flujo Básico		
	Acciones del Actor	Acciones del Sistema
1.	Selecciona la opción de eliminar el descuento.	
2.		Muestra el mensaje de advertencia y permite: <ul style="list-style-type: none"> • Aceptar (Si). • Cancelar la operación (No).
3.	Selecciona la opción de Aceptar.	
4.		<ul style="list-style-type: none"> • Se elimina el descuento. • Regresa a la vista anterior.
5.		El caso de uso termina.
Flujo Alternativo		
3. a El actor selecciona la opción de Cancelar.		
	Acciones del Actor	Acciones del Sistema
4.		a.1 Regresa a la vista anterior. a.2 El caso de uso termina.

CU Gestionar Concepto de Pago

Especificación del Caso de Uso

Objetivo	Añadir, modificar y eliminar un concepto de pago del empleado.
Actor	Especialista de RH.
Resumen	El caso de uso se inicia cuando el actor selecciona la opción que le permite realizar una acción sobre un concepto de pago del empleado. El actor puede añadir, eliminar y modificar un concepto de pago del empleado. En

	<p>caso de que seleccione la opción de añadir concepto de pago, el sistema dará la posibilidad de insertar los datos que se necesitan para llenar este formulario. Si el actor elige la opción de modificar el concepto de pago, el sistema mostrará los datos que pueden ser editables dentro del concepto de pago, y una vez realizados los cambios, guardará las modificaciones. Si el actor elige la opción de eliminar el concepto de pago, el sistema permitirá eliminar el concepto de pago mostrando un mensaje de confirmación, terminando así el caso de uso.</p>	
Precondiciones	<ul style="list-style-type: none"> • Debe haberse generado el escritorio de trabajo del usuario autenticado. • Debe haberse generado el <i>CU Gestionar Empleado</i>. • Para modificar el concepto de pago, el actor debe ser el responsable temporal del mismo. • Para eliminar el concepto de pago, el actor debe tener el permiso de eliminar el elemento. 	
Poscondiciones	<p>Se añadió, modificó o eliminó el concepto de pago del empleado por el actor.</p>	
Flujo de eventos		
Flujo básico		
	Acciones del Actor	Acciones del Sistema
1.	El caso de uso se inicia cuando el actor selecciona la pestaña <i>Otros Datos Principales</i> .	
2.		Permite mostrar el listado de todas los conceptos de pago de empleado registrados en el sistema y además permite realizar varias acciones:

		<ul style="list-style-type: none"> • Añadir un nuevo concepto de pago. • Modificar los datos de la entidad. Ver Sección 1: “Modificar datos del concepto de pago”. • Eliminar la entidad. Ver Sección 2: “Eliminar el concepto de pago”.
3.	Selecciona la opción de Añadir un nuevo concepto de pago.	
4.		Brinda la posibilidad de introducir los datos del concepto de pago: <ul style="list-style-type: none"> • Descripción • Porcentaje • Importe
5.	Introduce los datos del concepto de pago.	
6.		Permite seleccionar la opción: <ul style="list-style-type: none"> • Aceptar. • Cancelar la operación.
7.	Selecciona la opción de Aceptar.	
8.		<ul style="list-style-type: none"> • Valida los datos. • Crea el concepto de pago al empleado.
		Termina el caso de uso.
7. a Selecciona la opción de Cancelar.		
	Acciones del Actor	Acciones del Sistema
8.		a.1 Elimina los datos creados. a.2 Regresa a la vista anterior. a.3 El caso de uso termina.
8. a Existen datos incompletos.		

	Acciones del Actor	Acciones del Sistema
8.		a.1 Sobre cada campo incompleto muestra el mensaje de error. a.2 Regresa al paso 4 del Flujo Básico.
8. b Existen datos incorrectos.		
	Acciones del Actor	Acciones del Sistema
8.		b.1 Sobre cada campo incorrecto muestra el mensaje de error. b.2 Regresa al paso 4 del Flujo Básico

Sección 1: “Modificar datos del concepto de pago”.

Flujo Básico		
	Acciones del Actor	Acciones del Sistema
1.	Selecciona la opción de modificar los datos del concepto de pago.	
2.		Muestra los datos del concepto de pago y brinda la posibilidad de cambiar sus valores. Permite además: <ul style="list-style-type: none"> • Aceptar. • Cancelar la operación.
3.	Modifica los datos que necesite y selecciona la opción de Aceptar los mismos.	

4.		<ul style="list-style-type: none"> • Valida los datos. • Actualiza los datos del concepto de pago.
5.		El caso de uso termina.
Flujo Alterno		
3. a El actor selecciona la opción de Cancelar.		
	Acciones del Actor	Acciones del Sistema
4.		a.1 Elimina los datos creados. a.2 Regresa a la vista anterior. a.3 El caso de uso termina.
4. a Existen datos incompletos.		
	Acciones del Actor	Acciones del Sistema
		a.1 Sobre cada campo incompleto muestra el mensaje de información.
		a.2 Regresa al paso 2 del Flujo Básico.
4. b Existen datos incorrectos.		
	Acciones del Actor	Acciones del Sistema
		b.1 Sobre cada campo incorrecto muestra el mensaje de información.
		b.2 Regresa al paso 2 del Flujo Básico.

Sección 2: “Eliminar el concepto de pago”.

Flujo Básico

	Acciones del Actor	Acciones del Sistema
1.	Selecciona la opción de eliminar el concepto de pago.	
2.		Muestra el mensaje de advertencia y permite: <ul style="list-style-type: none"> • Aceptar (Si). • Cancelar la operación (No).
3.	Selecciona la opción de Aceptar.	
4.		<ul style="list-style-type: none"> • Se elimina el concepto de pago. • Regresa a la vista anterior.
5.		El caso de uso termina.
Flujo Alternativo		
3. a El actor selecciona la opción de Cancelar.		
	Acciones del Actor	Acciones del Sistema
4.		a.1 Regresa a la vista anterior. a.2 El caso de uso termina.

CU Gestionar Idiomas

Especificación del Caso de Uso

Objetivo	Añadir, modificar y eliminar un idioma del empleado.
Actor	Especialista de RH.
Resumen	El caso de uso se inicia cuando el actor selecciona la opción que le permite realizar una acción sobre un idioma del empleado. El actor puede añadir, eliminar y modificar un idioma del empleado. En caso de que seleccione la opción de añadir idioma, el sistema dará la posibilidad de insertar los datos que se necesitan para llenar este formulario. Si el actor elige la opción de modificar el idioma, el sistema mostrará los datos que pueden ser editables

	dentro del idioma, y una vez realizados los cambios, guardará las modificaciones. Si el actor elige la opción de eliminar el idioma, el sistema permitirá eliminar el idioma mostrando un mensaje de confirmación, terminando así el caso de uso.	
Precondiciones	<ul style="list-style-type: none"> • Debe haberse generado el escritorio de trabajo del usuario autenticado. • Debe haberse generado el <i>CU Gestionar Empleado</i>. • Para modificar el idioma, el actor debe ser el responsable temporal del mismo. • Para eliminar el idioma, el actor debe tener el permiso de eliminar el elemento. 	
Poscondiciones	Se añadió, modificó o eliminó el idioma del empleado por el actor.	
Flujo de eventos		
Flujo básico		
	Acciones del Actor	Acciones del Sistema
1.	El caso de uso se inicia cuando el actor selecciona la pestaña <i>Idiomas</i> .	
2.		Permite mostrar el listado de todos los idiomas de empleado registrados en el sistema y además permite realizar varias acciones: <ul style="list-style-type: none"> • Añadir un nuevo idioma. • Modificar los datos de la entidad. Ver Sección 1: "Modificar datos del idioma". • Eliminar la entidad. Ver Sección 2: "Eliminar el idioma".

3.	Selecciona la opción de Añadir un nuevo idioma.	
4.		Brinda la posibilidad de introducir los datos del idioma: <ul style="list-style-type: none"> • Idioma • Está Certificado • Nombre de la Escuela • Nivel • Lee • Escribe • Habla
5.	Introduce los datos del idioma.	
6.		Permite seleccionar la opción: <ul style="list-style-type: none"> • Aceptar. • Cancelar la operación.
7.	Selecciona la opción de Aceptar.	
8.		<ul style="list-style-type: none"> • Valida los datos. • Crea el idioma al empleado.
		Termina el caso de uso.

Flujo Alternativo

7. a Selecciona la opción de Cancelar.

	Acciones del Actor	Acciones del Sistema
8.		a.1 Elimina los datos creados. a.2 Regresa a la vista anterior. a.3 El caso de uso termina.

8. a Existen datos incompletos.		
	Acciones del Actor	Acciones del Sistema
8.		a.1 Sobre cada campo incompleto muestra el mensaje de error. a.2 Regresa al paso 4 del Flujo Básico.
8. b Existen datos incorrectos.		
	Acciones del Actor	Acciones del Sistema
8.		b.1 Sobre cada campo incorrecto muestra el mensaje de error. b.2 Regresa al paso 4 del Flujo Básico

Sección 1: “Modificar datos del idioma”.

Flujo Básico		
	Acciones del Actor	Acciones del Sistema
1.	Selecciona la opción de modificar los datos del idioma.	
2.		Muestra los datos del idioma y brinda la posibilidad de cambiar sus valores. Permite además: <ul style="list-style-type: none"> • Aceptar. • Cancelar la operación.
3.	Modifica los datos que necesite y selecciona la opción de Aceptar los	

	mismos.	
4.		<ul style="list-style-type: none"> • Valida los datos. • Actualiza los datos del idioma.
5.		El caso de uso termina.
Flujo Alterno		
3. a El actor selecciona la opción de Cancelar.		
	Acciones del Actor	Acciones del Sistema
4.		a.1 Elimina los datos creados. a.2 Regresa a la vista anterior. a.3 El caso de uso termina.
4. a Existen datos incompletos.		
	Acciones del Actor	Acciones del Sistema
		a.1 Sobre cada campo incompleto muestra el mensaje de información.
		a.2 Regresa al paso 2 del Flujo Básico.
4. b Existen datos incorrectos.		
	Acciones del Actor	Acciones del Sistema
		b.1 Sobre cada campo incorrecto muestra el mensaje de información.
		b.2 Regresa al paso 2 del Flujo Básico.

Sección 2: “Eliminar el idioma”.

Flujo Básico		
	Acciones del Actor	Acciones del Sistema
1.	Selecciona la opción de eliminar el idioma.	
2.		Muestra el mensaje de advertencia y permite: <ul style="list-style-type: none"> • Aceptar (Si). • Cancelar la operación (No).
3.	Selecciona la opción de Aceptar.	
4.		<ul style="list-style-type: none"> • Se elimina el idioma. • Regresa a la vista anterior.
5.		El caso de uso termina.
Flujo Alternativo		
3. a El actor selecciona la opción de Cancelar.		
	Acciones del Actor	Acciones del Sistema
4.		a.1 Regresa a la vista anterior. a.2 El caso de uso termina.

CU Gestionar Sanción

Especificación del Caso de Uso

Objetivo	Añadir, modificar y eliminar una retención al empleado.
Actor	Especialista de RH.
Resumen	El caso de uso se inicia cuando el actor selecciona la opción que le permite realizar una acción sobre una sanción del empleado. El actor puede añadir, eliminar y modificar una sanción del empleado. En caso de que seleccione la opción de añadir sanción, el sistema dará la posibilidad de insertar los

	<p>datos que se necesitan para llenar este formulario. Si el actor elige la opción de modificar la sanción, el sistema mostrará los datos que pueden ser editables dentro de la sanción, y una vez realizados los cambios, guardará las modificaciones. Si el actor elige la opción de eliminar la sanción, el sistema permitirá eliminar la sanción mostrando un mensaje de confirmación, terminando así el caso de uso.</p>	
Precondiciones	<ul style="list-style-type: none"> • Debe haberse generado el escritorio de trabajo del usuario autenticado. • Debe haberse generado el <i>CU Gestionar Empleado</i>. • Para modificar la sanción el actor debe ser el responsable temporal del mismo. • Para eliminar la sanción el actor debe tener el permiso de eliminar el elemento. 	
Poscondiciones	Se añadió, modificó o eliminó la sanción del empleado por el actor.	
Flujo de eventos		
Flujo básico		
	Acciones del Actor	Acciones del Sistema
1.	El caso de uso se inicia cuando el actor selecciona la pestaña <i>Sanciones</i> .	
2.		<p>Permite mostrar el listado de todas las sanciones del empleado registradas en el sistema y además permite realizar varias acciones:</p> <ul style="list-style-type: none"> • Añadir una nueva sanción. • Modificar los datos de la entidad. Ver Sección 1: “Modificar datos de la sanción”.

		<ul style="list-style-type: none"> • Eliminar la entidad. Ver Sección 2: "Eliminar la sanción".
3.	Selecciona la opción de Añadir una nueva sanción.	
4.		Brinda la posibilidad de introducir los datos de la sanción: <ul style="list-style-type: none"> • Tipo de Sanción • Número de Resolución • Fecha de Creación • Fecha de Comunicación • Fecha de Culminación
5.	Introduce y selecciona los datos de la sanción.	
6.		Permite seleccionar la opción: <ul style="list-style-type: none"> • Aceptar. • Cancelar la operación.
7.	Selecciona la opción de Aceptar.	
8.		<ul style="list-style-type: none"> • Valida los datos. • Crea la sanción al empleado.
		Termina el caso de uso.

Flujo Alternativo

7. a Selecciona la opción de Cancelar.

	Acciones del Actor	Acciones del Sistema
8.		a.1 Elimina los datos creados. a.2 Regresa a la vista anterior. a.3 El caso de uso termina.

8. a Existen datos incompletos.		
	Acciones del Actor	Acciones del Sistema
8.		a.1 Sobre cada campo incompleto muestra el mensaje de error. a.2 Regresa al paso 4 del Flujo Básico.
8. b Existen datos incorrectos.		
	Acciones del Actor	Acciones del Sistema
8.		b.1 Sobre cada campo incorrecto muestra el mensaje de error. b.2 Regresa al paso 4 del Flujo Básico

Sección 1: “Modificar datos de la sanción”.

Flujo Básico		
	Acciones del Actor	Acciones del Sistema
1.	Selecciona la opción de modificar los datos de la sanción.	
2.		Muestra los datos editables de la sanción y brinda la posibilidad de cambiar sus valores. Permite además: <ul style="list-style-type: none"> • Aceptar. • Cancelar la operación.
3.	Modifica los datos que necesite y selecciona la opción de Aceptar los	

	mismos.	
4.		<ul style="list-style-type: none"> • Valida los datos. • Actualiza los datos de la sanción.
5.		El caso de uso termina.
Flujo Alterno		
3. a El actor selecciona la opción de Cancelar.		
	Acciones del Actor	Acciones del Sistema
4.		a.1 Elimina los datos creados. a.2 Regresa a la vista anterior. a.3 El caso de uso termina.
4. a Existen datos incompletos.		
	Acciones del Actor	Acciones del Sistema
		a.1 Sobre cada campo incompleto muestra el mensaje de información.
		a.2 Regresa al paso 2 del Flujo Básico.
4. b Existen datos incorrectos.		
	Acciones del Actor	Acciones del Sistema
		b.1 Sobre cada campo incorrecto muestra el mensaje de información.
		b.2 Regresa al paso 2 del Flujo Básico.

Sección 2: “Eliminar la sanción”.

Flujo Básico		
	Acciones del Actor	Acciones del Sistema
1.	Selecciona la opción de eliminar la sanción.	
2.		Muestra el mensaje de advertencia y permite: <ul style="list-style-type: none"> • Aceptar (Si). • Cancelar la operación (No).
3.	Selecciona la opción de Aceptar.	
4.		<ul style="list-style-type: none"> • Se elimina la sanción. • Regresa a la vista anterior.
5.		El caso de uso termina.
Flujo Alternativo		
3. a El actor selecciona la opción de Cancelar.		
	Acciones del Actor	Acciones del Sistema
4.		a.1 Regresa a la vista anterior. a.2 El caso de uso termina.

CU Generar Reporte

Especificación del Caso de Uso

Objetivo	Generar reportes del sistema.
Actores	Especialista de RH (Inicia): Visualiza Prenómina.
Resumen	El caso de uso se inicia cuando el actor decide visualizar el reporte del sistema. El CU termina con la visualización del reporte seleccionado.

Precondiciones	Debe haberse generado el escritorio de trabajo del usuario autenticado.	
Poscondiciones	El reporte queda visualizado.	
Flujo de eventos		
Flujo básico		
	Acciones del Actor	Acciones del Sistema
1.	El caso de uso se inicia cuando el actor accede a la opción Reporte.	
2.		<p>Permite seleccionar:</p> <ul style="list-style-type: none"> • Período que desea visualizar el reporte • Una unidad organizativa especifica o mostrar todas. • Si desea cambio de página una por unidad organizativa. • Empleado para la visualización de determinados reporte. • Reporte que desea visualizar. <ul style="list-style-type: none"> ○ Mostrar Movimientos Elaborados. ○ Mostrar Contratos Próximos a Vencer. ○ Mostrar Contratos Elaborados. ○ Mostrar Análisis Periódico de Asistencia. ○ Mostrar Prenómina. ○ Mostar Cumpleaños
3.	Selecciona los datos de necesarios y el reporte deseado.	
4.		Muestra el reporte seleccionado.

		Termina el caso de uso.
--	--	-------------------------

Anexo 3. Descripción de las entidades de la base de datos.

p_escala_antigüedad		
Descripción: En la siguiente tabla se agrupa la información existente de una escala de antigüedad.		
Atributo	Tipo	Descripción
id_escala_antigüedad	INT	Etiqueta única que identifica al objeto (p_escala_antigüedad) en la tabla.
mas_de	INT	Almacena el 1er valor del rango de comparación.
menos_de	INT	Almacena el 2do valor del rango de comparación.
valor	DECIMAL(7,2)	Almacena el valor correspondiente al rango.
id_cat_antigüedad	INT	Etiqueta extranjera que identifica al objeto (cat_antigüedad) en la tabla.

Tabla 6. Descripción de la tabla p_escala_antigüedad.

n_cat_antigüedad		
Descripción: En la siguiente tabla se agrupa la información existente de una categoría de antigüedad.		
Atributo	Tipo	Descripción
id_cat_antigüedad	INT	Etiqueta única que identifica al objeto (n_cat_antigüedad) en la tabla.
código	VARCHAR(50)	Almacena el código de la categoría de antigüedad.
descripcion	VARCHAR(255)	Almacena la descripción de la categoría de antigüedad.

Tabla 7. Descripción de la tabla n_cat_antigüedad.

n_idioma		
Descripción: En la siguiente tabla se agrupa la información existente de un idioma.		
Atributo	Tipo	Descripción
id_idioma	INT	Etiqueta única que identifica al objeto (n_idioma)

		en la tabla.
descripcion	VARCHAR(50)	Almacena la descripción del idioma.

Tabla 8. Descripción de la tabla n_idioma.

p_idioma		
Descripción: En la siguiente tabla se agrupa la información de la relación entre n_idioma y p_employado.		
Atributo	Tipo	Descripción
id_idioma	INT	Etiqueta extranjera que identifica al objeto (n_idioma) en la tabla.
no_interno	VARCHAR(7)	Etiqueta extranjera que identifica al objeto (p_employado) en la tabla.

Tabla 9. Descripción de la tabla p_idioma.

p_incidentes		
Descripción: En la siguiente tabla se agrupa la información existente de una incidencia.		
Atributo	Tipo	Descripción
id_incidentes	INT	Etiqueta única que identifica al objeto (p_incidentes) en la tabla.
fecha	DATETIME	Almacena la fecha de la incidencia.
tiempo	DECIMAL(7,2)	Almacena el tiempo de la incidencia.
pagado	BIT	Etiqueta que indica si se pagó o no la incidencia.
no_interno	VARCHAR(7)	Etiqueta extranjera que identifica al objeto (p_employado) en la tabla.
id_peri	INT	Etiqueta extranjera que identifica al objeto (n_periodo) en la tabla.
id_ausencia	INT	Etiqueta extranjera que identifica al objeto (n_ausencias) en la tabla.

Tabla 10. Descripción de la tabla p_incidentes.

ph_incidentes		
Descripción: En la siguiente tabla se agrupa la información histórica existente de una incidencia.		
Atributo	Tipo	Descripción
id_incidentes	INT	Etiqueta única que identifica al objeto

		(ph_incidencias) en la tabla.
fecha	DATETIME	Almacena la fecha de la incidencia.
tiempo	DECIMAL(7,2)	Almacena el tiempo de la incidencia.
pagado	BIT	Etiqueta que indica si se pagó o no la incidencia.
no_interno	VARCHAR(7)	Etiqueta extranjera que identifica al objeto (p_employado) en la tabla.
id_peri	INT	Etiqueta extranjera que identifica al objeto (n_periodo) en la tabla.
id_ausencia	INT	Etiqueta extranjera que identifica al objeto (n_ausencias) en la tabla.

Tabla 11. Descripción de la tabla ph_incidencias.

n_tipo_plaza		
Descripción: En la siguiente tabla se agrupa la información existente de los tipos de plazas.		
Atributo	Tipo	Descripción
id_tipo_plaza	INT	Etiqueta extranjera que identifica al objeto (n_tipo_plaza) en la tabla.
descripcion	VARCHAR(50)	Almacena la descripción del tipo de plaza.

Tabla 12. Descripción de la tabla n_tipo_plaza.

n_profesion_ofic		
Descripción: En la siguiente tabla se agrupa la información existente de las profesiones u los oficios.		
Atributo	Tipo	Descripción
id_profesion_ofic	INT	Etiqueta extranjera que identifica al objeto (n_profesion_ofic) en la tabla.
descripcion	VARCHAR(50)	Almacena la descripción de la profesión u oficio.

Tabla 13. Descripción de la tabla n_profesion_ofic.

n_escolaridad		
Descripción: En la siguiente tabla se agrupa la información existente de la escolaridad.		
Atributo	Tipo	Descripción
id_escolaridad	INT	Etiqueta extranjera que identifica al objeto (n_escolaridad) en la tabla.

cod_escolaridad	VARCHAR(50)	Almacena el código de la escolaridad.
descripcion	VARCHAR(50)	Almacena la descripción de la escolaridad.

Tabla 14. Descripción de la tabla n_ escolaridad.

p_org_masas		
Descripción: En la siguiente tabla se agrupa la información de la relación entre p_ empleado y n_org_masas.		
Atributo	Tipo	Descripción
no_interno	VARCHAR(7)	Etiqueta extranjera que identifica al objeto (p_ empleado) en la tabla.
id_org_masa	INT	Etiqueta extranjera que identifica al objeto (n_org_masas) en la tabla.

Tabla 15. Descripción de la tabla p_org_masas.

n_org_masas		
Descripción: En la siguiente tabla se agrupa la información existente de las organizaciones de masas.		
Atributo	Tipo	Descripción
id_org_masa	INT	Etiqueta única que identifica al objeto (n_org_masas) en la tabla.
codigo	VARCHAR(50)	Almacena el código de la organización de masa.
descripcion	VARCHAR(50)	Almacena la descripción de la organización de masa.

Tabla 16. Descripción de la tabla n_org_masas.

n_calificacion		
Descripción: En la siguiente tabla se agrupa la información existente de la calificación.		
Atributo	Tipo	Descripción
id_calificacion	INT	Etiqueta única que identifica al objeto (n_ calificacion) en la tabla.
cod_calificacion	VARCHAR(50)	Almacena el código de la calificación.
descripcion	VARCHAR(50)	Almacena la descripción de la calificación.

Tabla 17. Descripción de la tabla n_ calificación.

p_contrato

Descripción: En la siguiente tabla se agrupa la información de la relación entre p_ empleado y n_ contrato.

Atributo	Tipo	Descripción
id_ p_ contrato	INT	Etiqueta única que identifica al objeto (p_ contrato) en la tabla.
fecha_inicio	DATETIME	Almacena la fecha de inicio del contrato.
fecha_fin	DATETIME	Almacena la fecha de fin del contrato.
actual	BIT	Etiqueta que indica si es actual.
anexo	BIT	Etiqueta que indica si tiene anexo.
no_interno	VARCHAR(7)	Etiqueta extranjera que identifica al objeto (p_ empleado) en la tabla.
id_tipo_contrato	INT	Etiqueta extranjera que identifica al objeto (n_ contrato) en la tabla.

Tabla 18. Descripción de la tabla p_ contrato.

n_ contrato

Descripción: En la siguiente tabla se agrupa la información existente del contrato.

Atributo	Tipo	Descripción
id_tipo_contrato	INT	Etiqueta única que identifica al objeto (n_ contrato) en la tabla.
descripcion	VARCHAR(50)	Almacena la descripción del contrato.
codigo	VARCHAR(50)	Almacena el código del contrato.
anexo	BIT	Etiqueta que indica si tiene anexo o no.
acumula_vac	BIT	Etiqueta que indica si acumula vacaciones o no.
pago_proforma	VARCHAR(50)	Almacena la url de la proforma de pago.

Tabla 19. Descripción de la tabla n_ contrato.

p_ tiempo_a_prueba

Descripción: En la siguiente tabla se agrupa la información existente del contrato.

Atributo	Tipo	Descripción
id_tiempo_a_prueba	INT	Etiqueta única que identifica al objeto (p_ tiempo_a_prueba) en la tabla.
tiempo_dias	INT	Almacena la descripción del contrato.
Id_categ	INT	Etiqueta extanjera que identifica al objeto

		(n_ cat_ocupacion) en la tabla.
Id_tipo_contrato	INT	Etiqueta extranjera que identifica al objeto (n_contrato) en la tabla.

Tabla 20. Descripción de la tabla p_ tiempo_a_prueba.

Anexo 4. Diagrama de paquete.

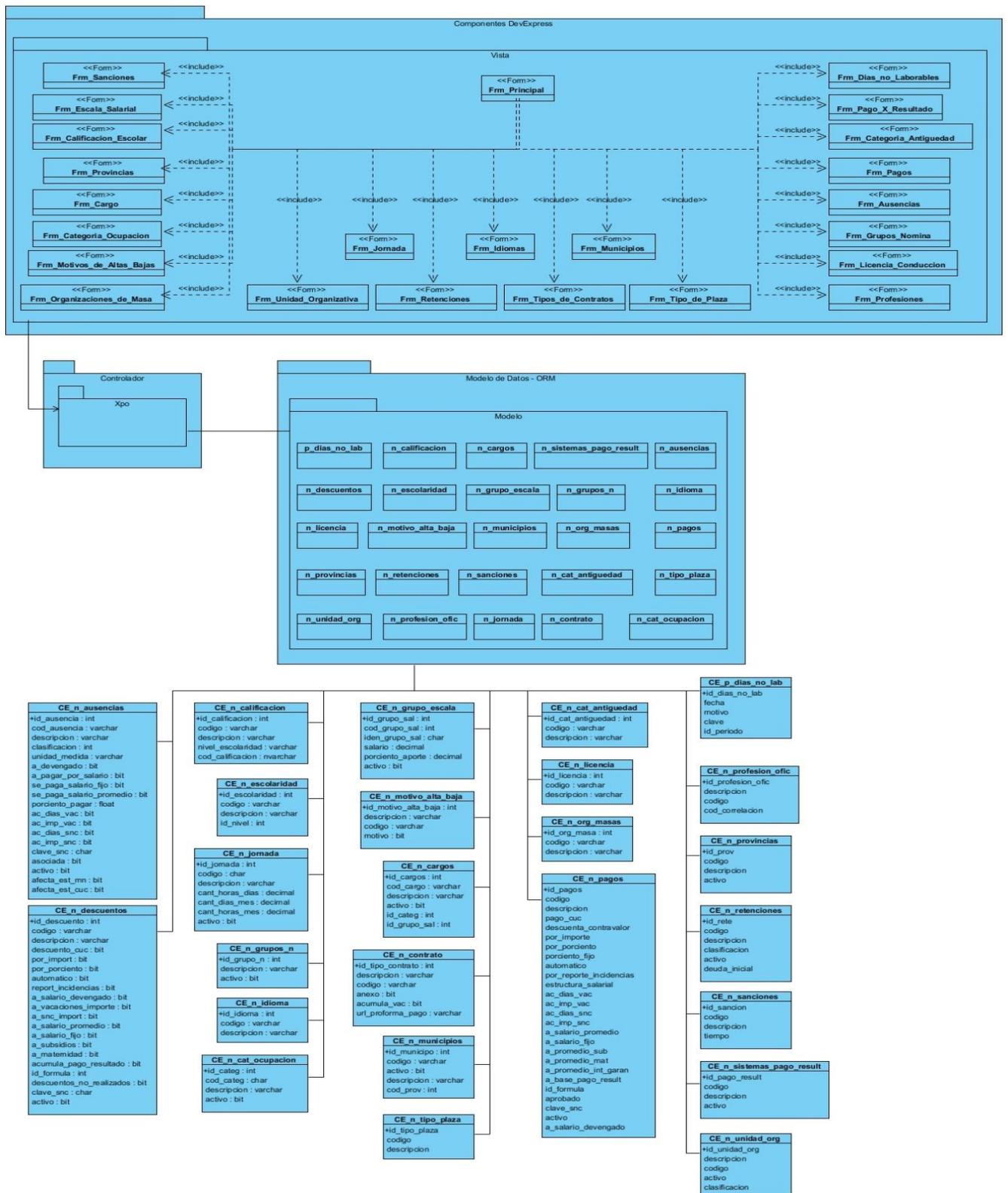


Figura 22. DCD Paquete Configuración.