

Universidad de las Ciencias Informáticas
Facultad 3



**Título: Análisis, Diseño e Implementación del
proceso Administración de Recursos Humanos
del Sistema de Gestión de Mantenimiento
Vehicular v 1.0.**

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

Autor: Alejandro Guerra López

Tutores: Ing. Erich Mario Gómez Pérez

Ing. Pedro Manuel Alas Verdecia

DECLARACIÓN DE AUTORÍA

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

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

Autor:

Alejandro Guerra López

Tutores:

Ing. Erich Mario Gómez Pérez

Ing. Pedro Manuel Alas Verdecia

AGRADECIMIENTOS

Quiero agradecer:

A mi mamá que siempre me ha apoyado y ha estado cuando la he necesitado, además de ser prácticamente una tutora más en el desarrollo de este trabajo y de mi vida.

A mi padrastro que ha sido como un padre para mí y al cual quiero como tal.

A mi hermanita Gabriela, que la quiero mucho.

A Leonor y Débora que las quiero como hermanas.

A mi papá y a mi hermano Alexander, que los quiero mucho.

A mi novia Mailyn que me ha ayudado muchísimo en la realización del trabajo además de ser una persona muy especial con la cual he compartido buenos momentos y a quien quiero mucho.

A Adrian, René, Alexis, Eddy, Yordenys (el Chardo), Máximo, Collado, Jean Pablo, Lisvan, Ariel, Danilo, Ernesto, en fin a todos los amigos del aula, del proyecto y del edificio con los que he compartido en estos años.

A Virtudes y Yusmara, a las cuales he mortificado bastante y viceversa, pero que nos llevamos bien.

A mi amigos de la Lenin que siempre hemos mantenido el contacto, Susel, Yoandry, Yailenis, Diana Rosa, Anaivis, entre otros.

A Gleibis que nos conocemos desde hace muchos años.

A mis tutores Erich y Pedro, muchas gracias por todo.

A todos los profesores que me han enseñado y preparado durante toda mi vida estudiantil, los cuales han sabido transmitir sus conocimientos y me han preparado como futuro profesional.

RESUMEN

En el departamento de Soluciones Empresariales perteneciente al CEIGE¹, se está desarrollando el Sistema de Gestión de Mantenimiento Vehicular para la gestión de los procesos que se realizan en el área de Transporte del CPNB².

En el presente trabajo se realiza el análisis, diseño e implementación del proceso Administración de Recursos Humanos con el objetivo de contribuir a mejorar su gestión y lograr su integración con el resto de los procesos del área de Transporte del CPNB. Puesto que actualmente la información relacionada a los recursos humanos es gestionada de forma manual, pues no existe un control sobre los datos de los familiares, las prendas policiales y herramientas que tienen asociadas. Además la ausencia de integración entre los procesos que se ejecutan en esta área provoca la inexistencia de un registro de las inspecciones técnicas y órdenes de trabajo realizadas por cada persona.

Con la implementación del componente Persona se contribuyó al ahorro de cuantiosos recursos materiales por parte de los clientes, facilitándoles su trabajo con la realización de una serie de reportes que le posibilitan controlar la información referente a los datos de los familiares, las prendas policiales y herramientas asociadas a los trabajadores. Se integraron los procesos Administración de Inspecciones Técnicas, Ejecución de Mantenimiento y Administración de Recursos Humanos, lo cual permitió tener un control sobre la cantidad de horas empleadas en la ejecución de una actividad de mantenimiento determinada.

PALABRAS CLAVE

CEIGE, CPNB, Recursos humanos

1 Centro de Informatización para la Gestión de Entidades

2 Cuerpo de Policía Nacional Bolivariana

TABLA DE CONTENIDOS

| | |
|---|-----|
| AGRADECIMIENTOS..... | I |
| RESUMEN | II |
| TABLA DE CONTENIDOS | III |
| INTRODUCCIÓN | 1 |
| CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA | 4 |
| 1.1 Introducción..... | 4 |
| 1.2 Proceso Administración de Recursos Humanos del área de Transporte del CPNB | 4 |
| 1.3 Gestión de los recursos humanos en los sistemas de gestión de flotas vehiculares..... | 4 |
| 1.4 Modelo de desarrollo orientado a componentes | 11 |
| 1.5 Arquitectura de software | 13 |
| 1.6 Tecnologías..... | 14 |
| 1.7 Herramientas..... | 19 |
| 1.8 Pruebas de Software..... | 23 |
| 1.9 Conclusiones parciales | 24 |
| CAPÍTULO 2 MODELADO DEL NEGOCIO Y REQUERIMIENTOS | 25 |
| 2.1 Introducción..... | 25 |
| 2.2 Modelado del negocio | 25 |
| 2.2.1 Modelado de procesos..... | 25 |
| 2.2.2 Mapa de procesos | 25 |
| 2.2.3 Descripción de procesos de negocio..... | 28 |
| 2.2.4 Validación del modelado de negocio | 29 |
| 2.2.5 Modelo Conceptual..... | 30 |
| 2.3 Requerimientos..... | 31 |
| 2.3.1 Técnicas para la captura de requisitos..... | 31 |
| 2.3.2 Especificaciones de los requisitos funcionales..... | 34 |
| 2.3.3 Administración de los requisitos..... | 38 |
| 2.4 Validación de requisitos | 39 |
| 2.5 Conclusiones parciales | 40 |
| CAPÍTULO 3 DISEÑO, IMPLEMENTACIÓN Y PRUEBAS | 41 |
| 3.1 Introducción..... | 41 |
| 3.2 Diseño..... | 41 |
| 3.2.1 Mecanismos de diseño | 41 |
| 3.2.2 Diagramas de clases del diseño | 43 |
| 3.2.3 Descripción de las clases del diseño..... | 47 |

| | | |
|-------|---|----|
| 3.2.4 | Patrones de diseño..... | 51 |
| 3.2.5 | Métricas para evaluar el diseño propuesto | 53 |
| 3.3 | Modelo de datos | 55 |
| 3.4 | Implementación..... | 57 |
| 3.4.1 | Diagrama de componentes..... | 57 |
| 3.4.2 | Modelo de despliegue | 57 |
| 3.4.3 | Estándares de codificación..... | 58 |
| 3.4.4 | Publicación de servicios entre componentes..... | 60 |
| 3.5 | Pruebas de software | 61 |
| 3.5.1 | Pruebas de caja blanca | 61 |
| 3.5.2 | Pruebas de caja negra..... | 69 |
| 3.6 | Conclusiones parciales | 71 |
| | CONCLUSIONES..... | 72 |
| | RECOMENDACIONES..... | 73 |
| | BIBLIOGRAFÍA..... | 74 |

INTRODUCCIÓN

En la actualidad existe un gran avance tecnológico que ha posibilitado que a través de sistemas informáticos sean realizadas labores que anteriormente se hacían de forma manual. Debido a este avance tecnológico está surgiendo una tendencia por parte de las organizaciones de informatizar sus procesos en aras de tener la información centralizada, controlada y de fácil acceso, contribuyendo en la toma de decisiones.

El CPNB de Venezuela no estando exento a estos cambios ha decidido realizar la informatización de los principales procesos que realizan dentro de todas sus áreas, específicamente del área de Transporte, la cual se encarga de controlar toda la información referente a las unidades policiales que son asignadas a las diferentes Dependencias de la policía, gestionando de estas los accidentes, inspecciones técnicas y mantenimientos tanto preventivos planificados como correctivos. Para lograr un manejo eficiente en la gestión de los recursos humanos de esta área del CPNB es necesario tener toda la información de forma organizada, centralizada e integrada con el resto de los procesos que allí se realizan, con el objetivo de tener un control de las inspecciones técnicas y órdenes de trabajo generadas por cada persona, así como de las prendas policiales y herramientas asociadas a estas.

Actualmente la información relacionada con los recursos humanos de esta área se gestiona de forma manual lo cual trae como consecuencia que no exista un control eficiente sobre las prendas policiales y herramientas asociadas a los mismos. La información referente a los familiares se encuentra archivada en el expediente de cada persona por cuanto la elaboración de relaciones de personal con hijos, cónyuge, no se realiza de forma rápida y eficiente, resultando un proceso muy engorroso para el personal de esta área ya que hay que revisar todos los expedientes para obtener dicha información. Además no existe un registro de las inspecciones técnicas, órdenes de trabajo realizadas por cada persona, lo cual provoca que haya falta de control sobre la cantidad de horas trabajadas en la ejecución de una actividad de mantenimiento determinada. Todo esto debido a que no existe una integración entre los procesos que se realizan en esta área tales como: Administración de Inspecciones Técnicas, Ejecución del Mantenimiento, Organización del Parque Vehicular, Gestión de Accidentes, con el proceso Administración de Recursos Humanos.

La problemática planteada permitió definir el siguiente **problema a resolver**. ¿Cómo contribuir a mejorar la gestión de los recursos humanos y su integración al resto de los procesos del área de Transporte del CPNB?

Se plantea como **objeto de estudio**: Gestión de los recursos humanos en los sistemas de gestión de mantenimiento vehicular.

Y el **campo de acción** se enmarca en: El proceso Administración de Recursos Humanos del área de Transporte del CPNB.

Para la solución del problema planteado se define como **objetivo general**: Realizar el análisis, diseño e implementación del proceso Administración de Recursos Humanos del Sistema de Gestión de Mantenimiento Vehicular.

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

1. Elaborar el marco teórico de la investigación.
2. Realizar el análisis del proceso Administración de Recursos Humanos del área de Transporte del CPNB.
3. Diseñar el proceso Administración de Recursos Humanos del Sistema de Gestión de Mantenimiento Vehicular v 1.0.
4. Implementar el proceso Administración de Recursos Humanos del Sistema de Gestión de Mantenimiento Vehicular v 1.0.
5. Validar los resultados obtenidos.

Se plantea como **idea a defender**. La realización del análisis, diseño e implementación del proceso Administración de Recursos Humanos permitirá mejorar su gestión e integración con el resto de los procesos que se realizan en el área de Transporte del CPNB.

Para el desarrollo del presente trabajo se emplearon los siguientes métodos de la investigación:

Histórico Lógico

Se centró el estudio en la gestión del proceso Administración de Recursos Humanos desde un enfoque histórico lógico, estudiando el desarrollo de este proceso desde el surgimiento del CPNB.

Analítico-Sintético

Permitió separar la información para facilitar su estudio y procesamiento, facilitando el descubrimiento de las características generales y las relaciones esenciales del proceso Administración de Recursos Humanos.

El presente trabajo de diploma se encuentra estructurado de la siguiente forma: introducción, tres capítulos, conclusiones, recomendaciones, anexos y la bibliografía consultada para el completamiento del marco teórico.

Capítulo 1 Fundamentación teórica:

En este capítulo se realiza un estudio sobre el proceso Administración de Recursos Humanos del área de Transporte del CPNB y se expone su importancia dentro de esta área. Se analiza la gestión de los recursos humanos en varios sistemas de gestión de flotas vehiculares existentes en Cuba y Venezuela, determinando a partir de este análisis las razones por las cuales no fueron utilizados. Además se justifica el empleo de cada una de las herramientas, metodologías, lenguaje de desarrollo, frameworks, servidores de base de datos y aplicaciones en la implementación del proceso Administración de Recursos Humanos.

Capítulo 2 Modelado del negocio y requerimientos:

En este capítulo se realiza la modelación del proceso de negocio Administración de Recursos Humanos del área de Transporte del CPNB y se identifican y validan los requisitos funcionales.

Capítulo 3 Diseño, implementación y pruebas de la propuesta solución:

En este capítulo se exponen los artefactos generados durante el diseño y la implementación de la solución, así como las métricas y pruebas utilizadas para su validación, las cuales garantizarán su calidad.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

El presente capítulo comienza abordando el tema de cómo se realiza el proceso Administración de Recursos Humanos dentro del área de Transporte del CPNB, así como la importancia que posee. Se realizará un análisis de la gestión de los recursos humanos en diferentes sistemas de gestión de flotas vehiculares existentes en Cuba y Venezuela. Este capítulo finaliza con la justificación de la selección de cada una de las herramientas, metodología, lenguaje de desarrollo, frameworks, servidores de base de datos y de aplicación web utilizados para la implementación del proceso Administración de Recursos Humanos.

1.2 Proceso Administración de Recursos Humanos del área de Transporte del CPNB

En el área de Transporte del CPNB el objetivo fundamental de la gestión de los recursos humanos es registrar toda la información necesaria para elaborar el expediente de cada persona, dígame datos generales como: nombres, apellidos, cédula de identidad, sexo, rango, cargo que ocupa, fecha de ingreso en el CPNB, cuerpo policial de origen y fecha de ingreso a este, teléfono y dirección particular. Además de datos personales tales como: estado civil, grupo sanguíneo, correo electrónico y fecha de nacimiento. De los familiares de estas personas también manejan algunos datos que son necesarios para cuestiones de seguro de vida, asignación de viviendas, dentro de los que se encuentran: nombres, apellidos, parentesco, si está vivo y cédula de identidad.

El proceso Administración de Recursos Humanos del área de Transporte del CPNB se relaciona específicamente con los procesos de negocio: Administración de Inspecciones Técnicas, Ejecución del Mantenimiento, ya que estos necesitan manejar la información de las personas que realizaron una inspección técnica determinada o que ejecutaron las actividades definidas en una orden de trabajo. Sin embargo actualmente no existe una integración entre ellos, lo cual provoca falta de control sobre los trabajos que realiza una persona en un período determinado y la cantidad de horas empleadas.

1.3 Gestión de los recursos humanos en los sistemas de gestión de flotas vehiculares.

1.3.1 Sistemas internacionales.

En el mundo existen varios sistemas de gestión de flotas vehiculares, los cuales brindan facilidades en el control y administración de todas las tareas que están relacionadas a los vehículos que posea una determinada empresa. Dentro de estos se encuentran:

Software Vehículos Pro V 6.10

El Software Vehículos Pro es una herramienta que permite ejercer un mayor control sobre todas aquellas tareas de mantenimiento que ayudan a prolongar la vida útil del vehículo (Soft, 2001-2010). Presenta los siguientes módulos:

Vehículos: permite añadir nuevos vehículos, modificar, eliminar los datos de los mismos. Imprime una relación de todos los vehículos dados de alta, y los intervalos de mantenimiento de cada uno de ellos. (Soft, 2001-2010)

Mantenimiento: permite controlar por kilómetros, meses, días y horas las tareas de mantenimiento que el usuario defina. Además avisa cuando le toca el próximo mantenimiento al vehículo, controlando el gasto incurrido en la realización de esta tarea. (Soft, 2001-2010)

Averías: permite controlar las averías sufridas por el vehículo, el gasto originado, los días que ha pasado en taller, materiales empleados, garantía de la reparación y las piezas sustituidas. (Soft, 2001-2010)

Neumáticos: permite controlar toda la información sobre los neumáticos que usa el vehículo. (Soft, 2001-2010)

Datos técnicos: en este módulo se registra información referente al combustible y al número de neumático, además permite anotar datos de la ficha técnica de un vehículo determinado, así como todo aquello que el usuario considere de interés, sirviendo estos datos de referencia para otras funciones. (Soft, 2001-2010)

Intervalos de mantenimiento: permite definir el intervalo en kilómetros, meses, días y horas entre un mantenimiento y otro. (Soft, 2001-2010)

Combustible: permite hacer un seguimiento sobre el consumo de un vehículo en litros por cada 100 km y litros por hora, en caso de que se realice por horas trabajadas. (Soft, 2001-2010)

Rutas: permite hacer un seguimiento sobre el movimiento de los vehículos, teniendo de esta forma un control global sobre los kilómetros recorridos por cada vehículo, consumo por 100 km. (Soft, 2001-2010)

Extras: permite controlar todos aquellos repuestos o tareas que no se contemplan en otros módulos. (Soft, 2001-2010)

Controles: permite llevar un control sobre el seguro de los accidentes, tarjetas de sanidad, Inspección Técnica, Tarjeta de Tacógrafo, generando un aviso de renovación con los días de antelación que el usuario defina en la opción de configuración, para prevenir su caducidad. (Soft, 2001-2010)

Accidentes: permite llevar un control de los accidentes sufridos por cada vehículo. (Soft, 2001-2010)

Depósitos de combustible: permite llevar un control exhaustivo de las entradas y salidas del combustible. (Soft, 2001-2010)

Repuestos: permite controlar pedidos de repuestos, recepción de los mismos, estado de los pedidos. (Soft, 2001-2010)

Equipos: permite llevar un control sobre los accesorios que sean montados en los vehículos. (Soft, 2001-2010)

Multas: permite llevar un control de las multas de tráfico impuestas a los conductores. (Soft, 2001-2010)

Software Transportex

Es una aplicación sencilla para la gestión de empresas de transporte de mercancía vía terrestre, la cual presenta los siguientes módulos:

Gestión de Neumáticos: muestra una información detallada de cada neumático. Permite gestionar inventarios de neumáticos de manera personalizada, así como realizar la asignación de estos de forma gráfica, individualmente o en lotes. (Máximo, 2011)

Actividades de Mantenimiento: permite programar y consultar tareas de mantenimiento preventivo o correctivo para vehículos y neumáticos, revisiones técnicas en toda la flota o en alguna unidad en particular. Lleva un registro de gastos asociados, costos de mano de obra, proveedores, materiales y repuestos utilizados; alarmas y tiempos de periodicidad mediante un sistema de aviso inteligente. (Máximo, 2011)

Control de viajes: permite establecer un registro de viajes de despachos de mercancía realizados por los vehículos, controlando eficientemente los adelantos y montos a pagar a cada conductor. Además lleva un

control detallado de los monto del flete y conceptos de gastos tales como: combustible, estadías, comida, peajes, caleta, viáticos y multas. (Máximo, 2011)

Inventario: permite ingresar los datos de la entrada de materiales, repuestos y piezas serializadas al almacén por proveedores, factura, precio; descarga automáticamente ítems del inventario; asigna piezas serializadas a través de las actividades de mantenimiento. Lleva el control de la ubicación actual y anterior de cada pieza serializada bien sea en algún almacén o vehículo de su flota. Crea actividades de mantenimiento para las piezas serializadas, controla los límites mínimos de cantidades de materiales, repuestos y piezas serializadas, además de los estudios generales o detallados de las cantidades en inventario y gastos asociados a estos elementos. (Máximo, 2011)

Sistema de Gestión de Flotas

El Sistema de Gestión de Flotas está diseñado para la sólida y correcta administración de flotas de vehículos (Ltda), presentando los siguientes módulos:

Vehículos: módulo de Mantención, permite crear, modificar, eliminar e imprimir todos los datos de los vehículos del sistema. (Ltda)

Proveedores: módulo de ingreso y mantención de todos los proveedores de insumos y servicios del sistema. (Ltda)

Combustible: módulo de ingreso y mantención de los datos que componen el movimiento de cargas de combustible efectuados a los vehículos. (Ltda)

Neumáticos: módulo de ingreso y mantención de los datos que componen el movimiento de cambio de neumático de los vehículos. (Ltda)

Mantención de Servicios: módulo de mantención de los servicios que componen los movimientos relacionados a los servicios realizados al vehículo. (Ltda)

Orden de Trabajo: módulo de ingreso y mantención de las órdenes de trabajo emitidas a los vehículos para efectuar cierta cantidad de servicios en él. (Ltda)

Cargas de Combustible: módulo de búsqueda de datos que permite ejecutar consultas y obtener resultados acerca de las cargas de combustible registradas en el sistema. Genera un informe sumamente

personalizado según los requerimientos del usuario y permite calcular los datos más importantes por cada vehículo. (Ltda)

Viajes realizados: módulo de búsqueda de datos de los movimientos de viajes realizados por los vehículos en un lapso de tiempo y según las especificaciones establecidas por el usuario. (Ltda)

Reprocesamiento de Cargas de Combustible: módulo que permite reprocesar los datos automáticos generados en el movimiento de cargas de combustible como distancia y rendimiento; generar un informe de errores de los usuarios en las cargas de combustible registradas en el sistema y reparar automáticamente los más usuales. Este módulo evita que los informes de combustible contengan información no válida después de editar o eliminar un movimiento. (Ltda)

Tablas Maestras: módulo de Mantenimiento de los datos generales del Sistema de Gestión de Flotas. (Ltda)

Software GGTT (Gestión Taller)

Es un sistema informático creado especialmente para administrar las Áreas de Taller y Mantenimiento de empresas que poseen grande flotas de vehículos. Cuenta con los siguientes módulos:

Taller: módulo que permite la apertura de Órdenes de Trabajo o Solicitud de Taller; el registro de fecha y hora, tanto de ingreso como de entrega de la unidad. Define los diferentes estados por los que pasa cada Solicitud de Taller y el cierre de cada solicitud. Completa la ficha de cada unidad; define qué tipo de trabajo se va a realizar en el taller, predictivo, correctivo o preventivo según sea el caso. Registra las tareas realizadas a las unidades y administra todas las tareas que realizan los mecánicos en el taller; el trabajo de terceros, identificación del proveedor, el trabajo realizado y el costo. Controla el equipamiento provisto a las unidades; trabaja con los módulos STOCK para la descarga de repuestos y consumibles utilizados en el taller en las distintas unidades. Además define por parte del Administrador del Taller los diferentes estados por los que pasan las unidades. Permite la generación de informes históricos de trabajos realizados a una unidad determinada, detalles del costo de los trabajos que se le realicen, incluso por terceros y de los repuestos más utilizados. (S.A, 2011)

Mantenimiento: este módulo permite el ingreso masivo de las novedades periódicas que la unidad produce; define la frecuencia con que se le va a realizar el mantenimiento ya sea por kilómetros, horas o por kilómetros horas simultáneamente; programa el envío de un mail o msg a un celular cuando una unidad determinada se encuentre en mantenimiento. Ingresar partes de cada unidad desde internet

permitiendo que operadores ubicados en distintos lugares puedan actualizar la base de datos de mantenimiento y realizar consultas de forma dinámica. (S.A, 2011)

STOCK: permite administrar íntegramente los repuestos disponibles en el pañol³. (S.A, 2011)

Gestoría: permite controlar toda la documentación de importancia y con vencimiento que posee una unidad; definiendo la unidad de medida con que se controlará. Digitaliza la documentación de cada unidad; establece que por el vencimiento de determinada documentación, se envíe un mail a una persona determinada y de forma simultánea se dispare un mensaje de texto al celular del responsable. (S.A, 2011)

Amortizaciones: este módulo permite definir nuevos esquemas de amortización con parámetros no considerados en los esquemas básicos, así como proyectar en el tiempo futuras depreciaciones en función de los registros históricos de cada unidad; emitiendo gráficos comparativos de la depreciación de cada unidad en el tiempo. (S.A, 2011)

Digital: este módulo posibilita el registro por parte del mecánico de las operaciones que él mismo realiza. Permite además elevar el nivel de eficiencia en los registros de los trabajos que realizan los operadores del Taller. (S.A, 2011)

Móvil: permite que los encargados de las tareas del taller o mantenimiento, puedan registrar los trabajos realizados a las unidades en lugares remotos. (S.A, 2011)

Usuarios: permite administrar y limitar los accesos que pueden efectuar los usuarios. Presenta las siguientes características: ilimitada cantidad de usuarios; posibilidad de configurar el nombre de los usuarios y la clave de acceso del sistema por parte del administrador; creación de los perfiles de los usuarios. (S.A, 2011)

1.3.2 Sistemas nacionales.

Software SGestMan

Es un sistema informático para recoger y procesar la información especializada de cada área de la empresa que posee relación con la gestión del mantenimiento (CITMATEL, 2006). Este sistema cuenta con 10 módulos de trabajo, los cuales son:

³ Compartimientos que tienen los vehículos para guardar herramientas, víveres y municiones.

Administración: a través de este módulo se garantiza la integridad y fiabilidad de la información que se incluye en el resto del sistema, ya que se encuentra previsto para llevar todo el control del acceso de los usuarios, así como los permisos a las opciones de cada uno de los módulos. (CITMATEL, 2006)

Patrimonio: este módulo permite estructurar toda la información de los diferentes elementos que forman parte de cualquier empresa, permitiendo definir todas las necesidades de mantenimiento que precisa una empresa, en aras de poseer un Patrimonio con un alto nivel de disponibilidad y utilización técnica. (CITMATEL, 2006)

Recursos Humanos: este módulo se encuentra previsto para estructurar toda la información de los empleados que están vinculados a la organización del mantenimiento, y con los cuales se garantiza la disponibilidad de los objetos que conforman el Patrimonio de una empresa. (CITMATEL, 2006)

Órdenes de Servicio: en este módulo se lleva todo el control técnico y económico de las órdenes de servicio que son realizadas por los diferentes ejecutores del mantenimiento, de manera que permite obtener detalladamente todos los históricos de los objetos del Patrimonio. (CITMATEL, 2006)

Contrato: en este módulo se lleva toda la información técnica y económica de las contrataciones realizadas por el departamento de mantenimiento, incluyendo de esta forma también todos aquellos prestadores de servicios que reparan equipamientos de las empresas, así como todos los servicios de mantenimiento que se encuentran tercerizados. (CITMATEL, 2006)

Preventivo: a través de este módulo se prepara la estrategia de proyección, programación y planificación de acciones de mantenimiento preventivo. (CITMATEL, 2006)

Economía: se controla de forma dinámica y en todos los niveles las actividades de mantenimiento, a partir de la valoración de todos los costos obtenidos en las órdenes de servicios desarrolladas por los clientes. (CITMATEL, 2006)

Logística: permite el enlace con sistemas, requisición de materiales, nomencladores de productos, necesidades de compra; permitiendo esto que se puedan incluir todos aquellos productos y/o materiales necesarios para los servicios de mantenimiento que se realicen en la empresa. (CITMATEL, 2006)

Producción: en este módulo se controlan todos los servicios que se realizan a clientes externos a través de órdenes de producción. (CITMATEL, 2006)

Informes: en este módulo se puede obtener toda la información que contiene el sistema en forma impresa. (CITMATEL, 2006)

A partir de la información obtenida sobre el proceso Administración de Recursos Humanos del área de Transporte del CPNB, se realizó un análisis con los sistemas anteriormente mencionados, en aras de conocer cómo estos gestionan los recursos humanos y determinar si era factible su utilización. El análisis arrojó como resultado que ninguno podía ser utilizado en el área de Transporte del CPNB debido a los siguientes inconvenientes encontrados:

- Los sistemas **Transportex, GGTT, Sistema de Gestión de Flotas y Software Vehículos Pro V 6.10** no poseen un módulo a través del cual se pueda gestionar toda la información referente a los recursos humanos que posee el área de Transporte del CPNB.
- **Transportex, GGTT, Sistema de Gestión de Flotas y Software Vehículos Pro V 6.10** son software privativos, además de que utilizan y se ejecutan sobre tecnologías privativas, lo cual impide su utilización ya que el CPNB tiene como política el uso de tecnologías libres.
- El sistema **SGestMan** posee un módulo para el control de los recursos humanos pero a través de este no se gestionan los datos de los familiares, prendas personales, herramientas asociadas a los mismos, información necesaria en el área de Transporte del CPNB.
- Todos los sistemas son aplicaciones de escritorio, lo cual tiene como deficiencia que su acceso se limite al ordenador donde estén instaladas; dependen del sistema operativo y de sus capacidades; requieren de una instalación y actualización personalizada, además de que suelen tener requerimientos especiales de software y librerías.

1.4 Modelo de desarrollo orientado a componentes

Para guiar el desarrollo del Sistema de Gestión de Mantenimiento Vehicular se empleará el **modelo de desarrollo orientado en componentes** creado por el CEIGE, basado en buenas prácticas y principios de varias metodologías, ya sean ágiles como XP⁴ y SCRUM o pesada como RUP⁵. Está orientado a las

4 Programación Extrema del inglés Extreme Programming.

5 Proceso Unificado de Rational del inglés Rational Unified Process.

necesidades y artefactos que son generados durante el desarrollo de cualquier software; define todos los roles involucrados y sus responsabilidades, las diferentes actividades que realizan, junto con el flujo de estas y los artefactos que se deben generar. (Equipo de Producción)

La utilización de este modelo se debe a que es el establecido por el centro para el desarrollo de todos sus productos además de las características que presenta:

➤ **Centrado en la arquitectura**

A través de la arquitectura se orientan las prioridades del desarrollo, se resuelven las necesidades tecnológicas y de soporte, determina la línea base, los elementos de software estructurales a partir de los existentes en la arquitectura de negocio, interviene en la gestión de cambios y diseña la evolución e integración del producto. (Equipo de Producción)

➤ **Orientado a componentes**

Según el nivel de significado arquitectónico de los componentes, son orientadas las iteraciones. (Equipo de Producción)

➤ **Iterativo e incremental**

El equipo de arquitectura, los clientes y la alta gerencia, planifican y coordinan las iteraciones, estas constituyen el desarrollo de componentes, los cuales son integrados al término de la iteración, permitiendo la evolución incremental del producto. (Equipo de Producción)

➤ **Ágil y adaptable al cambio**

Los clientes y funcionales son involucrados en el proyecto, por lo que poseen parte de la responsabilidad del éxito del mismo. Semanalmente se concilian, discuten y aprueban los cambios. El desarrollo de las partes formaliza solo las características principales de la solución, priorizándose de esta manera los talleres y las comunicaciones. (Equipo de Producción)

➤ **Son utilizados solamente los artefactos necesarios para documentar el producto.** (Equipo de Producción)

➤ **Se modela el negocio por procesos y no por casos de uso.** (Equipo de Producción)

1.5 Arquitectura de software

1.5.1 Arquitectura basada en componentes

La Arquitectura de Software se define como la representación de alto nivel de la estructura de un sistema o aplicación, que describe las partes que la integran, las iteraciones entre ellas, los patrones que supervisan su composición y las restricciones a la hora de aplicar esos patrones. De esta forma aparecen las basadas en componentes. Este tipo es completamente modular y favorece la reutilización de todos sus elementos, incluyendo los que definen las distintas relaciones entre ellos (Fuentes, y otros, 2009). Para el desarrollo del Sistema de Gestión de Mantenimiento Vehicular v 1.0 fue definido, por el CEIGE, la utilización de la **Arquitectura basada en componentes**.

La **Arquitectura basada en componentes** se enfoca en la descomposición del software en componentes funcionales, lo cual provee un mayor nivel de abstracción y permite la reutilización de componentes pre-existentes.

1.5.2 Patrones Arquitectónicos

Los patrones arquitectónicos especifican un conjunto predefinido de subsistemas con sus responsabilidades y una serie de recomendaciones para organizar los distintos componentes (Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla). Se pueden ver como la descripción de un problema en particular y recurrente de diseño, que aparece en contextos de diseño arquitectónicos específicos, y representa un esquema genérico demostrado con éxito para su solución (Buschman, 1996).

1.5.2.1 Modelo-Vista-Controlador

El patrón **Modelo-Vista-Controlador** divide una aplicación en tres componentes: el modelo, la vista y el controlador. El modelo se encarga de administrar el comportamiento y los datos del dominio de la aplicación, responder a los requerimientos de información sobre su estado y las instrucciones de cambiar el mismo. Mantiene el conocimiento del sistema y no depende de ninguna vista o controlador. La vista maneja la visualización de la información, mientras que el controlador analiza los mensajes de eventos que recibe el sistema, modifica u obtiene datos del modelo en respuesta a las peticiones del usuario. Este patrón arquitectónico será empleado para el desarrollo del Sistema de Mantenimiento Vehicular v 1.0

debido a que su utilización fue definida por el CEIGE, además de estar implementado dentro del marco de trabajo Sauxe, sobre el cual será desarrollado el sistema.

1.6 Tecnologías

1.6.1 AJAX

El término AJAX es un acrónimo de Asynchronous JavaScript⁶ + XML⁷. Según el artículo “Ajax: A New Approach to Web Applications”, publicado por Jesse James Garret el 18 de febrero de 2005, se definió de la siguiente forma: “Ajax no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes.”. Las tecnologías que componen Ajax son: XHTML⁸, CSS⁹, DOM¹⁰, XML, XSLT¹¹, JSON¹², XMLHttpRequest¹³ y JavaScript (Pérez, 2008). En la implementación del Sistema de Gestión de Mantenimiento Vehicular será utilizada para la comunicación entre la capa de la vista y la controladora, por las ventajas que ofrece:

- Mejorar la interacción del usuario con la aplicación pues evita las recargas constantes de la página, realizando el intercambio con el servidor en un segundo plano.
- Sustituir las peticiones http por javascript que son realizadas al elemento encargado de AJAX, por lo que, las que no necesiten la intervención del servidor obtienen una respuesta inmediata, mientras las otras se realizan de forma asíncrona a través de AJAX.

6 Lenguaje de programación orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

7 Lenguaje de Marcas Extensible del inglés Extensible Markup Language.

8 Lenguaje Extensible de Marcado de Hipertexto del inglés Extensible Hypertext Markup Language.

9 Hojas de estilo en cascada del inglés Cascading Style Sheets.

10 Modelo de Objetos del Documento del inglés Document Object Model..

11 Lenguaje de Hojas Extensibles de Transformación del inglés Extensible Stylesheet Language.

12 Notación de Objetos de JavaScript del inglés JavaScript Object Notation.

13 Lenguaje de Marcas Extensible/Protocolo de Transferencia de Hipertextos del inglés Extensible Markup Language/ Hypertext Transfer Protocol.

- Puede ser utilizada en cualquier plataforma o navegador.

1.6.2 Frameworks

1.6.2.1 EXTJS 3.4

Ext JS es un framework que soporta JavaScript para construir aplicaciones complejas en internet, además de ser la base para Ext. Designer, el cual es una aplicación de escritorio que te permite construir aplicaciones web (Inc, 2011). En la implementación del Sistema de Gestión de Mantenimiento Vehicular será empleado para la construcción de todas las interfaces de la aplicación debido a las ventajas que ofrece:

- Crear aplicaciones RIA ¹⁴ mediante javaScript, utilizando componentes predefinidos.
- Ser utilizado sobre cualquier navegador.
- Proveer un balance entre Cliente-Servidor, ya que distribuye la carga, permitiendo que el servidor gestione más clientes al mismo tiempo.
- Poseer un API ¹⁵fácil de usar.
- Contar con licencias de código abierto y comerciales.
- Desarrollar interfaces parecidas a las aplicaciones de escritorio.
- Obtener información del servidor sin estar sujeto a la acción de un usuario.

1.6.2.2 Zend Framework 1.5.0

Zend Framework es un framework de código abierto para PHP, desarrollado por Zend; en su más bajo nivel es una librería de componentes escritos en PHP5, para facilitar el desarrollo de sitios web (Zend Technologies Ltd., 2006-2012). Será utilizado dado las ventajas que ofrece:

- Es de código abierto por lo cual no hay que preocuparse por cuestiones de derecho de autor o patentes.

14 Aplicaciones de Internet Enriquecidas del inglés Rich Internet Applications

15 Interfaz de Programación de Aplicaciones del inglés Application Programming Interface.

- Se encuentra bajo una licencia BSD¹⁶, lo cual permite su distribución, así como las aplicaciones que se desarrollen con él.
- Posee una amplia documentación, lo más completa y clara posible.
- Está basado en PHP5 por lo cual es completamente orientado a objetos.
- Implementa el patrón modelo-vista-controlador.
- Sus componentes tienen un bajo acoplamiento por lo que se pueden usar de forma independiente.
- Cuenta con soporte para internalización y localización de aplicaciones.
- Contiene adaptadores para gran cantidad de base de datos diferentes.

1.6.2.3 Doctrine Framework 0.11.0

Doctrine es un framework ORM¹⁷ para PHP 5.2 inspirado en Hibernate que permite trabajar con un esquema de base de datos como si fuera un conjunto de objetos, no de tablas y registros. Brinda una capa de abstracción de la base de datos muy completa (Doctrine, 2011). Se empleará debido a sus ventajas:

- La separación total del sistema de base de datos, por lo que si en un futuro se decidiera cambiar el motor de base de datos, esto no afectaría al sistema.
- La utilización de los métodos de un objeto de datos desde distintas partes de la aplicación.
- Posee un sistema para evitar tipos de ataque como pueden ser las inyecciones SQL¹⁸.
- Ordena de forma correcta la capa de datos, por lo que facilita el mantenimiento del código.
- Cuenta con un lenguaje propio para hacer las consultas.
- Encapsula la lógica de los datos permitiendo hacer cambios que afecten toda la aplicación, únicamente modificando una función.

16 Distribución de Software Berkeley del inglés Berkeley Software Distribution.

17 Mapeo Objeto-Relacional del inglés Object-Relational Mapping.

18 Lenguaje de Consulta Estructurado del inglés Structured Query Language.

1.6.2.4 Marco de Trabajo Sauxe

El desarrollo del Sistema de Gestión de Mantenimiento Vehicular será realizado utilizando el marco de trabajo Sauxe, desarrollado por el Departamento de Tecnología del CEIGE. Contiene un conjunto de componentes reutilizables que provee la estructura genérica y el comportamiento para una familia de abstracciones, logrando una mayor estandarización, flexibilidad, integración y agilidad en el proceso de desarrollo. Está desarrollado sobre lenguaje de programación PHP y en la capa de presentación utiliza Java script y HTML. En la capa de presentación utiliza ExtJs por la gran gama de componentes que se pueden reutilizar para agilizar el proceso de desarrollo y mostrarle al usuario una interfaz más amigable y funcional. En la capa de negocio utiliza Zend Framework y en la de acceso a los datos Doctrine Framework. (Morera, junio del 2011)

1.6.2.5 Acaxia

Para garantizar la seguridad del Sistema de Gestión de Mantenimiento Vehicular será empleado Acaxia, el cual fue desarrollado sobre tecnologías libres como el lenguaje PHP, gestor de base de datos Postgres, servidor web Apache. Gestiona las conexiones a la base de datos, las funcionalidades asociadas y las acciones que realizan. Información a partir de la cual se le asignan los permisos a los roles creados en el sistema, o sea, a partir del rol que tenga asignado un usuario, será el nivel de acceso que tendrá este con respecto a las acciones y funcionalidades que podrá realizar. Permite la administración dinámica de perfiles de usuario y consultar todas las acciones realizadas en los sistemas, con toda la información asociada, dígame tiempo, valores, interacción. (Centro de Informatización de la Gestión de Entidades, 2011)

1.6.3 Notación de modelado de procesos de negocio: BPMN

BPMN¹⁹ es una notación gráfica que describe la lógica de los pasos de un proceso de negocio. Esta ha sido diseñada para coordinar la secuencia de los procesos y los mensajes que fluyen entre los participantes de las diferentes actividades (Bizagi, 2011). Será utilizada por las ventajas que ofrece:

- Ser independiente de cualquier metodología de modelado de procesos.

19 Notación para el Modelado de Procesos de Negocio del inglés Business Process Modeling Notation.

- Modelar los procesos de una manera unificada y estandarizada permitiendo un entendimiento entre todas las personas de una organización.
- Crear un puente estandarizado para disminuir la brecha entre los procesos de negocio y la implementación de estos.
- Ser fácil de entender.

1.6.4 Lenguaje Unificado de Modelado: UML

UML²⁰ es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema orientado a objetos. (Lovellette, 1999) Será utilizado por las siguientes ventajas:

- Es una consolidación de muchas notaciones y conceptos más usados orientados a objetos.
- Utiliza un conjunto de símbolos para representar gráficamente los diversos componentes en relación con el sistema.
- Puede ser utilizado para el modelado de negocio de procesamiento y modelado de requisitos.
- Es independiente del lenguaje de programación.
- Realiza la documentación de todos los artefactos que son generados durante el proceso de desarrollo.
- Se aprende con facilidad.
- Es un lenguaje consolidado.

1.6.5 JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlo. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. Este lenguaje permite interactuar con el navegador de manera dinámica y eficaz, proporcionando a las páginas web dinamismo y vida. (Morera, junio del 2011)

²⁰ Lenguaje Unificado de Modelado del inglés Unified Modeling Language.

1.6.6 Lenguaje de programación PHP

PHP es un lenguaje de programación multiplataforma para la creación rápida de contenidos dinámicos de sitios web. Su nombre surge de la abreviación del concepto PHP Hypertext Preprocessor (Bakken, y otros, 2001). Su empleo para el desarrollo del Sistema de Gestión de Mantenimiento Vehicular se debe a que el marco de trabajo Sauxe, sobre el cual se implementará el sistema, fue realizado con este lenguaje de programación, debido a sus ventajas:

- Posee gran cantidad de documentación y librerías.
- Provee diferentes niveles de seguridad, que pueden ser configurados en el archivo .ini.
- Es de código abierto por lo cual no hay que pagar para la obtención de actualizaciones anuales, ni por el uso de este.
- La interacción dinámica entre la página web y el usuario.
- La creación de aplicaciones para servidores e independientes del navegador.
- Utilizando el mismo código fuente, funciona sobre múltiples plataformas.
- Se pueden leer y escribir archivos, crear imágenes, conectarse a servidores remotos, realizar consultas a base de datos.

1.7 Herramientas

1.7.1 Visual Paradigm 2.0

Visual Paradigm es una herramienta CASE²¹. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación (Pressman, 2002). Esta brinda una serie de ventajas que permiten utilizarla:

- Es multiplataforma.
- El diseño es centrado en casos de uso y enfocado al negocio.

21 Ingeniería de Software Asistida por Computadora del inglés Computer Aided Software Engineering.

- Uso de un lenguaje estándar común que facilita la comunicación para todo el equipo de desarrollo.
- Posee la capacidad de realizar ingeniería directa e inversa.
- Permite que el modelo y el código permanezcan sincronizados durante todo el ciclo de desarrollo.
- Cuenta con una licencia gratuita y comercial.
- Las imágenes y reportes generados son de muy buena calidad.
- Contiene soporte para varios idiomas.
- Es fácil de instalar y actualizar.
- La exportación e importación de ficheros XML.

1.7.2 Servidor de aplicaciones

1.7.2.1 Apache 2.2

El servidor Apache HTTP es un servidor web de tecnología código abierto sólido y para uso comercial desarrollado por la Apache Software Foundation (The Apache Software Foundation, 2011). Este servidor de aplicaciones web ofrece una serie de ventajas:

- Es de código abierto, no hay que pagar por utilizarlo.
- Soporta lenguajes como Perl, Python, TCL²² y PHP.
- La configuración de mensajes de error.
- Tiene un alto grado de configuración.
- Posee soporte para SSL²³ y TLS²⁴.
- La autenticación de base de datos basadas en SGBD²⁵.

22 Lenguaje de Herramientas de Comando del inglés Tool Command Language.

23 Capa de Conexión Segura del inglés Secure Sockets Layer.

24 Seguridad en la Capa de Transporte del inglés Transport Layer Security.

25 Sistemas de Gestión de Base de Datos del inglés Database Management System.

1.7.3 Servidor de base de datos

1.7.3.1 PostgreSQL 8.3

PostgreSQL es un potente motor de base de datos, que tiene prestaciones y funcionalidades equivalentes a muchos gestores de base de datos comerciales (PostgreSQL-es, 2009-2012). Este posee una serie de ventajas que permiten utilizarlo:

- Está distribuido bajo licencia BSD.
- Su código fuente está disponible libremente.
- Utiliza un modelo cliente/servidor.
- Funciona correctamente con grandes cantidades de datos y una alta concurrencia de usuarios accediendo al mismo tiempo.
- Posee acceso encriptado vía SSL.
- Tiene múltiples métodos de autenticación.
- Es multiplataforma.
- Cuenta con una amplia y completa documentación.
- Realiza copias de seguridad.
- Contiene numerosos tipos de datos y permite definir algunos nuevos.

1.7.4 Navegador

1.7.4.1 Firefox 4.0

Firefox es un navegador de Internet con interfaz gráfica de usuario desarrollado por la Corporación Mozilla y un gran número de voluntarios (MozillaES, 2009). Se empleará por sus ventajas:

- Es de código libre y multiplataforma. (mozilla.org, 1998-2012)
- Está basado en el poderoso motor de búsqueda Gecko. (mozilla.org, 1998-2012)
- Permite insertar complementos desarrollados por terceros. (mozilla.org, 1998-2012)

- Posee función de autocompletamiento en la barra de navegación, lo cual facilita la navegación por internet. (mozilla.org, 1998-2012)
- Guarda la contraseña para entrar a un sitio determinado. (mozilla.org, 1998-2012)
- La restauración de las ventanas y pestañas que se estuvieran usando antes de cerrar el navegador. (mozilla.org, 1998-2012)
- La navegación ininterrumpida cuando hay un cuelgue de alguno de sus plugins, ya que si uno de estos se congela no afecta al resto del navegador. (mozilla.org, 1998-2012)
- Seguridad avanzada en la navegación pues permite la identificación de algún sitio web de forma instantánea, posibilita navegar de forma privada y sin registro de lo accedido en internet. (mozilla.org, 1998-2012)
- Establece conexiones seguras a sitios web. (mozilla.org, 1998-2012)
- La integración al antivirus que posea la computadora, soporte para varios idiomas y alta configurabilidad. (mozilla.org, 1998-2012)
- Protección en cuanto a malware se refiera, puesto que en caso de acceso a un sitio sospechoso, el navegador le notificará y le dará las razones. (mozilla.org, 1998-2012)

1.7.5 Eclipse-PHP-Galileo

Eclipse es un IDE²⁶ de código abierto basado en Java. Es un desarrollo de IBM cuyo código fuente fue puesto a disposición de los usuarios (Dpto de Informática de la Universidad de Valencia, 2004). Será empleado, para la implementación del proceso Administración de Recursos Humanos del área de Transporte del CPNB por las ventajas que ofrece:

- Es multiplataforma.
- El desarrollo de cualquier lenguaje de programación a través de plugins.
- Es de código abierto y libre, así como sus plugins.
- Altamente configurable.

26 Entorno de Desarrollo Integrado del inglés Integrated Development Environment.

1.8 Pruebas de Software

Las pruebas del software son un elemento crítico para la garantía de calidad del software y representa una visión final de las especificaciones, del diseño y de la codificación. Una vez generado el código fuente, el software debe ser probado para descubrir y corregir, el máximo de errores posibles antes de su entrega al cliente. Glen Myers en su libro estableció varias normas que pueden servir acertadamente como objetivos de las pruebas, estas son (Pressman, 2005):

- La prueba es el proceso de ejecución de un programa con la intención de descubrir un error.
- Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
- Una prueba tiene éxito si descubre un error no detectado hasta entonces.

Los datos que se van recogiendo a medida que se lleva a cabo la prueba proporcionan una buena indicación de la fiabilidad del software y de alguna manera, indican la calidad del software como un todo. (Pressman, 2005) Para realizarle las pruebas al módulo Persona, con el objetivo de descubrir y corregir errores existentes en este, fueron empleadas las pruebas de caja blanca y caja negra.

1.8.1 Pruebas de caja blanca

La prueba de caja blanca, denominada a veces prueba de caja de cristal es un método de diseño de casos de prueba que utiliza la estructura de control del diseño procedimental para obtener los casos de prueba. (Pressman, 2005)

La prueba de caja blanca que se aplicará a la solución desarrollada será la **Prueba del Camino Básico**, que permite obtener una medida de la complejidad lógica del diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución, garantizando con estos que durante la prueba se ejecute por lo menos una vez cada sentencia del programa. (Pressman, 2005)

1.8.2 Pruebas de caja negra

Las pruebas de caja negra, también denominada prueba de comportamiento, se centran en los requisitos funcionales del software. O sea, la prueba de caja negra permite al ingeniero del software obtener

conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. (Pressman, 2005)

Los métodos de caja negra que se utilizaran para asegurar la calidad de la aplicación desarrollada serán:

1.8.2.1 Partición equivalente

La partición equivalente es un método de prueba de caja negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Un caso de prueba ideal descubre de forma inmediata una clase de errores que de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. (Pressman, 2005)

1.8.2.2 Análisis de valores límite

Los errores tienden a darse más en los límites del campo de entrada que en el centro. Por ello, se ha desarrollado el análisis de valores límite (AVL) como técnica de prueba. El análisis de valores límite es una técnica de diseño de casos de prueba que complementa a la partición equivalente. En lugar de seleccionar cualquier elemento de una clase de equivalencia, el AVL lleva a la elección de casos de prueba en los extremos de las clases. En lugar de centrarse solamente en las condiciones de entrada, el AVL obtiene casos de prueba también para el campo de salida. (Pressman, 2005)

1.9 Conclusiones parciales

Después de todo el estudio realizado durante este capítulo se puede concluir que los sistemas existentes en Cuba y Venezuela poseen una serie de inconvenientes, los cuales impiden que sean utilizados para realizar la informatización de los procesos del área de Transporte del CPNB, por lo cual es necesario la creación de un sistema informático capaz de satisfacer las necesidades que actualmente presenta esta área en cada uno de los procesos que realiza, específicamente el de Administración de Recursos Humanos. Además en este capítulo se definieron todas las herramientas, tecnologías y lenguajes de programación, diseño y modelado a utilizar para el desarrollo del módulo Persona del Sistema de Gestión de Mantenimiento Vehicular, a través del cual se realizará la gestión de los recursos humanos del área de Transporte del CPNB.

CAPÍTULO 2 MODELADO DEL NEGOCIO Y REQUERIMIENTOS

2.1 Introducción

En el siguiente capítulo se realiza la descripción del proceso Administración de Recursos Humanos del área de Transporte del CPNB. Se elabora un mapa de procesos en el cual son expuestos los procesos claves y artefactos relacionados con la gestión de los recursos humanos. Se identifican y registran en el Modelo Conceptual los conceptos fundamentales que se manejan en el área de Transporte en cuanto a la administración de recursos humanos. Los requisitos funcionales identificados a partir de la aplicación de técnicas, son especificados y validados.

2.2 Modelado del negocio

2.2.1 Modelado de procesos

El **modelado de procesos de negocio** es la representación de los procesos de negocio de una empresa u organización con objeto de que puedan ser analizados y mejorados. (Kawtar Benghazi, José Luis Garrido Bullejos, Manuel Noguera García, Departamento de Lenguajes y Sistemas Informáticos, Universidad de Granada) Su elaboración brinda una serie de beneficios a las organizaciones ya que les permite reutilizar procesos que sean más eficientes, ventajosos y productivos, detectar tareas que no puedan ser realizadas, así como mejorar de forma general los procesos que se realizan.

La confección de los modelos de procesos de negocio es realizada principalmente por los analistas, labor que les brinda una serie de beneficios como: la agilización del proceso de desarrollo y de la carga de trabajo, la identificación de errores en las fases tempranas, mayor nivel de abstracción, además de la trazabilidad del sistema por identificación de tareas y su asignación a procedimientos manuales o automatizados.

2.2.2 Mapa de procesos

Un **mapa de procesos** es un diagrama que muestra, de manera visual, los procesos que se desarrollan en una organización, así como las relaciones (si las hay) existentes entre ellos. Su utilización permite organizar y documentar los procesos con el fin de facilitar su descripción y entendimiento, por lo que, en su elaboración es necesario tener presente que la expresión gráfica debe ser clara y precisa.

CAPÍTULO 2 MODELADO DEL NEGOCIO Y REQUERIMIENTOS 2012

Para la realización de los mapas de procesos del Sistema de Gestión de Mantenimiento Vehicular fue empleada la plantilla conformada por el CEIGE (consultar documentos entregables CG-SM-DR-001 y CG-SM-DR-002), en la cual se representan los procesos identificados en el área de Transporte del CPNB agrupados por niveles, colocando en un nivel 0 los procesos claves de esta organización y un nivel 1 sus procesos derivados. Dentro de los procesos del nivel 0 se encuentran: Organizar, Actualizar medidores, Ejecución y Control, en el documento CG-SM-DR-001 se especifica información más detallada sobre ellos.

Para los procesos identificados en el nivel 1 se crea un documento que contiene: nombre, breve descripción, nivel y proceso padre al que pertenecen (consultar documento entregable CG-SM-DR-002).

El contenido de este trabajo se enmarca en el proceso Administración de los Recursos Humanos perteneciente al nivel 1 del proceso Organizar.

Se identifican para cada uno de los procesos del nivel 1 los artefactos de entrada y salida involucrados en el proceso y el formato en que se manejan actualmente en el área de Transporte del CPNB.

La matriz de procesos contiene la relación que existe entre los procesos, sistemas o involucrados en el proceso, exponiendo los artefactos de salida de un proceso que constituyen entradas para otros y viceversa (consultar Tabla 1 y documento entregable CG-SM-DR-002).

Tabla 1 Matriz de relación entre los procesos de negocio de nivel 1

| Entradas | | | | | | | |
|-------------------------|----------------------------|------------------------------------|-----------------------------------|---------------------------------|-----------------------------------|----------------------|--|
| | Organizar parque vehicular | Administración de recursos humanos | Definir acciones de mantenimiento | Asignar unidades a dependencias | Control de unidades desaparecidas | Gestionar accidentes | Director de Gestión Administrativa de las dependencias |
| Organizar parque | | | | | | Listado de unidades | |

CAPÍTULO 2 MODELADO DEL NEGOCIO Y REQUERIMIENTOS 2012

| | | | | | | | | | |
|--|------------------------------------|------------------------------|--|--------------------------|--|--|--------------------------|--|--|
| S a l i d a s | vehicular | | | | | | Expediente de la unidad. | | |
| | Administración de recursos humanos | | | | | | Expediente del empleado. | | |
| | Definir acciones de mantenimiento | Grupo de unidades | | | | | | | |
| | Gestionar accidentes | Expediente de la unidad. | | | | | | | |
| | | Registro de accidentes. | | | | | | | |
| Director de Gestión Administrativa de las dependencias | Listado de unidades | Listado de recursos humanos. | Listado de unidades por clasificación. | Listado de dependencias. | | | | | |
| | | | Listado documentos | Memorándum | | | | | |

| | | | | | | | |
|--------------------|--|--|-----------|---------------------------------------|--------------------------|--|--|
| as | | | técnicos. | Estructura del área de mantenimiento. | | | |
| Funcionario | | | | | Copia de la denuncia. | | |
| | | | | | Extracto de novedad. | | |
| | | | | | Informe del funcionario. | | |

2.2.3 Descripción de procesos de negocio

El proceso Administración de Recursos Humanos tiene como objetivo principal registrar toda la información necesaria para la confección del expediente de cada persona. Este proceso comienza con la definición y elaboración del listado de los cargos del área de Transporte del CPNB, a partir de la estructura que esta posee. Luego se realiza el registro de todas las personas que van a trabajar en esta área, quedando conformado así el expediente de la persona. A cada persona se le asigna un cargo y se registran los datos de sus familiares. Además se les hace entrega de las prendas policiales y de vestir, con las cuales van a realizar su trabajo. Si la persona necesita de alguna herramienta de trabajo, se hace entrega de la misma.

A continuación se presentan los artefactos de entrada y de salida del proceso Administración de Recursos Humanos.

Tabla 2 Artefactos de entrada y salida del proceso Administración de Recursos Humanos

| Proceso Administración de Recursos Humanos | |
|--|---|
| Artefactos de Entrada | Artefactos de Salida |
| Organigrama del área de Transporte del CPNB | Listado de cargos del área de Transporte del CPNB |
| Listado de las personas que van a trabajar en el área de Transporte del CPNB | Expediente de la persona |
| Listado de cargos del área de Transporte del CPNB | |

Para obtener mayor información sobre la modelación de este proceso de negocio consultar Anexo 1 y para su descripción consultar documento entregable CG-SM-DR-005.

2.2.4 Validación del modelado de negocio

La validación del modelado de procesos de negocio se realizó aplicando la técnica Revisión técnica formal, donde los clientes revisaron cada documento de descripción de procesos de negocio realizado con el objetivo de validar su correcta elaboración y que el flujo de actividades de cada proceso estuviera en correspondencia con la información brindada.

Calisoft²⁷ validó el modelado de negocio realizado, mediante dos iteraciones, revisando elementos como ortografía, formato y aspectos técnicos, fueron liberados los artefactos Descripción de procesos de negocio y Mapa de procesos de negocio (consultar documento entregable Acta de Liberación PN.pdf).

²⁷ Calisoft, acrónimo de Calidad de Software. Es un centro de referencia de calidad y órgano de certificación de software, conocido y acreditado a nivel nacional, que brinda asesorías, entrenamiento, métodos de medición y servicios de verificación-validación a las diferentes entidades.

2.2.5 Modelo Conceptual

El modelo conceptual es un diagrama conformado con los conceptos que son significativos para el área que se analice, así como las relaciones existentes entre ellos. Permite identificar, organizar y realizar razonamientos sobre los componentes y comportamientos del sistema, siendo la guía para el proceso de diseño del software, pudiéndose usar además como referencia para evaluar un diseño particular y razonar sobre la solución realizada.

El modelo conceptual del proceso Administración de Recursos Humanos fue confeccionado con todas las clases conceptuales identificadas en este proceso junto con los atributos y relaciones existentes entre las mismas (consultar Imagen 1 y documento entregable CG-SM-DE-002).

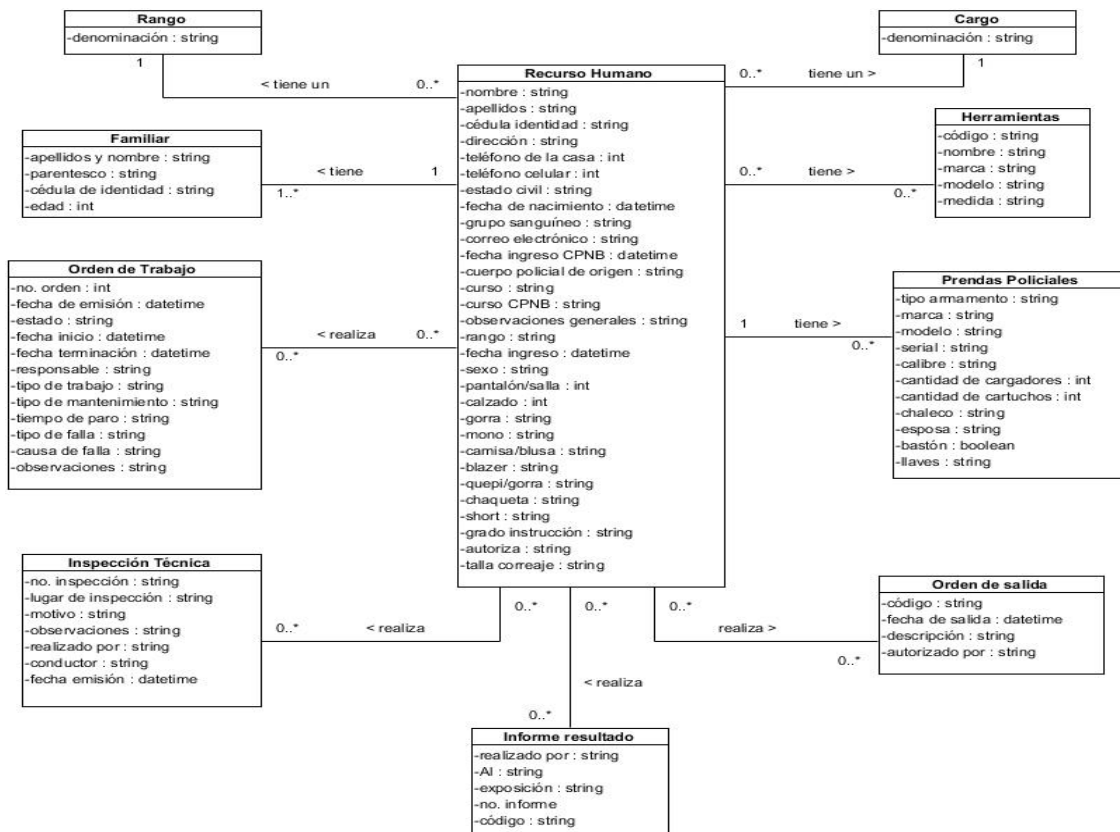


Imagen 1 Modelo conceptual del proceso Administración de Recursos Humanos

2.3 Requerimientos

Un requerimiento es simplemente una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de este. (Sommerville, 2005) Pueden dividirse en dos categorías: requerimientos funcionales y requerimiento no funcionales. Los requerimientos funcionales son los que definen las funciones que el sistema será capaz de realizar, describen las transformaciones que el sistema realiza sobre las entradas para producir salidas. Es importante que se describa el ¿Qué? y no el ¿Cómo? se deben hacer esas transformaciones. (La Ingeniería de Requerimientos y su importancia en el desarrollo de proyectos de software, 2005)

La captura de requisitos es la actividad mediante la cual el equipo de desarrollo de un sistema de software extrae, de cualquier fuente de información disponible, las necesidades que debe cubrir dicho sistema. El proceso de captura de requisitos puede resultar complejo, principalmente si el entorno de trabajo es desconocido para el equipo de analistas, y depende mucho de las personas que participen en él. Por la complejidad que todo esto puede implicar, la ingeniería de requisitos ha trabajado desde hace años en desarrollar técnicas que permitan hacer este proceso de una forma más eficiente y precisa. (Escalona, y otros, 2002)

Antes de comenzar el proceso de captura de los requisitos del proceso Administración de Recursos Humanos se identificaron los proveedores válidos de requisitos teniendo en cuenta algunos criterios como: conocimiento del negocio, disponibilidad de tiempo, habilidades de comunicación, conocimiento de informática (consultar documento CG-SM-DE-i2401) en aras de identificar a las personas con mayor dominio del proceso en el área de Transporte del CPNB y así obtener los requerimientos a implementar.

2.3.1 Técnicas para la captura de requisitos

Se aplicaron las siguientes técnicas para la captura de los requerimientos:

Entrevista: A través de encuentros planificados con los proveedores de requisitos se realizaron preguntas relacionadas con el proceso Administración de Recursos Humanos en función de obtener las necesidades que tenían de informatización. Esto proporcionó como resultado que fueran identificados los requerimientos por los que se guiaría el desarrollo del módulo, a través del cual se realizaría la gestión de este proceso.

JAD²⁸: El equipo de desarrollo junto con los proveedores de requisitos, fueron revisando la documentación de los requerimientos a medida que iban siendo identificados los requisitos, con el objetivo de chequear que las especificaciones realizadas sobre el proceso Administración de Recursos Humanos satisficieran las necesidades de informatización que poseían los proveedores de requisitos, dando como resultado que se documentaran los requisitos que dieran cumplimiento a lo descrito por estos.

La **Lluvia de ideas**: En cada encuentro fueron debatidos los requerimientos identificados, fluyendo variadas ideas acerca de los mismos, lo cual arrojó como resultado que se obtuviera una visión más amplia de lo que se quería implementar, favoreciéndose de esta forma el avance de futuras etapas en el desarrollo del módulo, a través del cual se llevaría a cabo la gestión del proceso Administración de Recursos Humanos.

Los requisitos funcionales identificados del proceso Administración de Recursos Humanos del área de Transporte del CPNB son:

RF 1. Agrupación de requisitos: Gestionar recursos humanos

RF 1.1 Adicionar recurso humano

RF 1.2 Modificar recurso humano

RF 1.3 Eliminar recurso humano

RF 1.4 Consultar recurso humano

RF 1.5 Listar recursos humanos

RF 1.6 Buscar recurso humano

RF 2. Agrupación de requisitos: Gestionar datos de los familiares

RF 2.1 Listar datos de los familiares

RF 2.2 Adicionar datos de los familiares

RF 2.3 Modificar datos de los familiares

²⁸ Desarrollo conjunto de aplicaciones del inglés Joint Application Development.

RF 2.4 Eliminar datos de los familiares

RF 3. Asociar herramientas a los empleados

RF 4. Quitar herramientas a los empleados

RF 5 Realizar búsqueda avanzada de recurso humano

RF 6 Agrupación de requisitos: Gestionar herramientas

RF 6.1 Listar herramientas

RF 6.2 Buscar herramientas

RF 6.3 Listar herramientas de los empleados

RF 6.4 Listar herramientas utilizadas en la Orden de Trabajo

RF 6.5 Adicionar herramienta

RF 6.6 Modificar herramienta

RF 6.7 Eliminar herramienta

RF 7 Mostrar relación de prendas policiales

RF 8 Mostrar relación de prendas policiales de vestir

RF 9 Mostrar relación de personal con hijos

RF 10 Mostrar relación de recursos humanos

RF 11 Mostrar relación de datos de los familiares

RF 12 Mostrar relación de familiares (cónyuge)

RF 13 Mostrar relación de personal con hijos_cónyuge

RF 14 Mostrar relación de prendas policiales (armamento)

RF 15 Mostrar relación de prendas policiales (bastón)

RF 16 Mostrar relación de prendas policiales (chaleco)

RF 17 Mostrar relación de prendas policiales (esposas)

2.3.2 Especificaciones de los requisitos funcionales

La especificación de los requisitos identificados se realizó siguiendo los pasos descritos en la plantilla conformada por el CEIGE.

Tabla 3 Especificación del requisito Adicionar recurso humano

| | |
|-------------------------|--|
| Precondiciones | El usuario ha sido validado. |
| Flujo de eventos | |
| Flujo básico | |
| 1 | Se introducen los datos generales de la persona: Rango Apellidos Nombres Cargo Teléfono de la casa Teléfono celular Cédula Identidad Cuerpo policial de origen Fecha ingreso CPNB Curso CPNB Fecha de ingreso Curso Sexo Dirección |
| 2 | Se introducen los datos personales de la persona: Estado civil Fecha nacimiento Grado de instrucción Años de antigüedad Correo electrónico Grupo sanguíneo |
| 3 | Se introducen los datos de los familiares de la persona: Apellidos y Nombre Parentesco Cédula Identidad Edad |
| 4 | Se introducen las prendas policiales asociadas a la persona: Tipo de armamento Marca Modelo |

Serial
 Calibre
 Cantidad de cargadores
 Cantidad de cartuchos
 Chaleco
 Esposa
 Bastón
 Llaves
 Linterna

5 Se asocian las herramientas de trabajo de la persona y se introduce la cantidad:
 Cantidad

6 Se introducen las observaciones así como las tallas de:
 Camisa
 Pantalón
 Short
 Quepi
 Mono
 Calzado
 Gorra
 Mono
 Blazer
 Chaqueta
 Talla de corraje

7 El sistema valida (ver validación 1) los datos introducidos.

8 Si los datos son correctos el sistema los registra.

9 El sistema confirma el registro de los datos.

10 Concluye el requisito.

Pos-condiciones

1 Se registró en el sistema una nueva persona.

Flujos alternativos

Flujo alternativo 8.a Información errónea.

1 El sistema señala los datos erróneos y permite corregirlos.

2 El usuario corrige los datos.

3 Volver al paso 7 del flujo básico.

Pos-condiciones

1 NA

Flujo alternativo 8.b Información incompleta.

1 El sistema señala los datos vacíos y permite corregirlos.

2 El usuario corrige los datos.

3 Volver al paso 7 del flujo básico.

Pos-condiciones

1 NA

Flujo alternativo *.a El usuario cancela la acción.

1 Concluye el requisito.

Pos-condiciones

1 NA

Validaciones

1 Se validan los datos según lo establecido en el Modelo conceptual CG-SM-DE-002.

| | | |
|-------------------|-----------------------------|---|
| Relaciones | Requisitos Incluidos | Paso 5: Asociar herramientas a los empleados. |
|-------------------|-----------------------------|---|

| | | |
|--|--------------------|----|
| | Extensiones | NA |
|--|--------------------|----|

| | | |
|------------------|-------------------------|--|
| Conceptos | Recursos humanos | Visibles en la interfaz: Rango Apellidos Nombres Cargo Teléfono de la casa Teléfono celular Cédula Identidad Cuerpo policial de origen Fecha ingreso CPNB Curso CPNB Fecha de ingreso Curso Sexo Estado civil Fecha nacimiento Grado de instrucción Años de antigüedad Correo electrónico Grupo sanguíneo Dirección Camisa Pantalón Short Quepi Mono Calzado Gorra Mono Blazer Chaqueta Talla de correa Utilizados internamente: NA |
|------------------|-------------------------|--|

CAPÍTULO 2 MODELADO DEL NEGOCIO Y REQUERIMIENTOS 2012

| | |
|--------------------------------|---|
| Prendas policiales | Visibles en la interfaz: Tipo de armamento Marca Modelo Serial Calibre Cantidad de cargadores Cantidad de cartuchos Chaleco Esposa Bastón Llaves Linterna Utilizados internamente: NA |
| Herramientas de trabajo | Visibles en la interfaz: Código Nombre Marca Modelo Medida Utilizados internamente: NA |
| Familiares | Visibles en la interfaz: Apellidos y Nombre Parentesco Cédula Identidad Edad Utilizados internamente: NA |
| Requisitos especiales | NA |
| Asuntos pendientes | NA |

Prototipo elemental de interfaz gráfica de usuario

The image shows a software window titled "Adicionar recursos humanos" with a tabbed interface. The "Datos generales" tab is active. The form contains the following fields and controls:

- Rango: Dropdown menu with "-Seleccione-" selected.
- Apellidos: Text input field.
- Nombres: Text input field.
- Cargo: Dropdown menu with "-Seleccione-" selected.
- Teléfono de la casa: Text input field.
- Teléfono celular: Text input field.
- Cédula Identidad: Text input field.
- Cuerpo policial de origen: Text input field.
- Fecha de ingreso: Dropdown menu with "-Seleccione-" selected.
- Curso: Text input field.
- Fecha ingreso CPNB: Dropdown menu with "-Seleccione-" selected.
- Curso CPNB: Text input field.
- Sexo: Dropdown menu.
- Dirección: Text input field.

At the bottom right of the window are three buttons: "Cancelar", "Aplicar", and "Aceptar".

Imagen 2 Adicionar recurso humano. Datos generales

Los otros requerimientos de la agrupación de requisitos Gestionar recursos humanos se encuentran especificados en el Anexo 2 y los restantes en los documentos CG-SM-DR-336, CG-SM-DR-032, CG-SM-DR-033, CG-SM-DR-031, CG-SM-DR-131, CG-SM-DR-132, CG-SM-DR-121, CG-SM-DR-133, CG-SM-DR-363, CG-SM-DR-364.

2.3.3 Administración de los requisitos

La administración de los requisitos es un enfoque sistemático para obtener, organizar y documentar los mismos, así como mantener un acuerdo entre el cliente y el equipo del proyecto en los posibles cambios. La clave para una efectiva administración de requisitos es: Mantener un enunciado claro, junto con los

atributos para cada tipo de requisitos y su seguimiento con otros o con elementos del proyecto. (Sandoval Carvajal, y otros)

Para su realización se utilizó la herramienta OSRMT²⁹, la cual permite la gestión de los requisitos y garantiza la trazabilidad entre los mismos, lleva un histórico de quién y cuándo se ha realizado una modificación sobre la documentación y un control de versiones en los documentos de trabajo. Con el empleo de dicha herramienta fue elaborada la matriz de trazabilidad de los requerimientos del proceso de negocio Administración de Recursos Humanos.

2.4 Validación de requisitos

La validación de los requisitos se realiza con la finalidad de comprobar que los requerimientos identificados sean precisos, consistentes, realistas, verificables, definan lo que el usuario desea del producto final, que los errores que hayan sido detectados sean corregidos y el resultado del trabajo cumpla con los estándares establecidos para el proceso, el proyecto y el producto. (Pressman, 2006)

Para la validación de los requisitos fueron aplicadas las siguientes técnicas:

Revisión técnica formal: La revisión de las especificaciones de los requerimientos del proceso Administración de Recursos Humanos fue realizada con los proveedores de requisitos. Con la utilización de esta técnica se validó que no existieran errores en el contenido o malas interpretaciones, información incompleta, inconsistencias y que los requisitos no fueran contradictorios, imposibles o inalcanzables, dando como resultado que fueran aprobados los que estaban descritos de forma correcta, clara y consistente.

Prototipos: A partir de las especificaciones de requisitos fueron conformados prototipos de interfaz de usuario. Con esto se validó que los requerimientos estaban en concordancia con las necesidades plasmadas por los proveedores de requisitos. El empleo de esta técnica ofreció como resultados que el cliente tuviera una idea de la estructura de la interfaz de usuario y se favoreciera la comunicación con el mismo, ya que tenía una visión inicial del módulo a través del cual se gestionaría el proceso Administración de Recursos Humanos.

29 Herramienta Libre para la Gestión de Requisitos del inglés Open Source Requirement Management Tool

Generación de casos de prueba (test de requisitos): Fueron definidos y diseñados casos de pruebas para cada requerimiento especificado, con el objetivo de verificar el cumplimiento de los mismos. La utilización de esta técnica ofreció los siguientes resultados: fueron identificados los posibles escenarios de los requisitos, así como los juegos de datos de los campos determinados en estos, se validaron y aprobaron los que estaban bien enunciados, descritos y consistentes.

Liberación por Calisoft: Los requisitos identificados fueron validados por Calisoft en 3 iteraciones, revisando elementos como ortografía, formato y aspectos técnicos, emitiendo un Acta de liberación de productos de software (consultar documento entregable Acta de Liberación ER-Mantenimiento Vehicular.pdf).

2.5 Conclusiones parciales

Con la modelación del negocio del proceso Administración de Recursos Humanos, fueron generados una serie de artefactos que crearon las condiciones para la captura de los requerimientos funcionales de este proceso, los cuales fueron identificados, especificados y validados, garantizando con esto último que errores existentes que no hubieran sido corregidos pudieran afectar futuras etapas del desarrollo del Sistema de Gestión de Mantenimiento Vehicular, dando paso así, a la realización del diseño de la propuesta de solución.

CAPÍTULO 3 DISEÑO, IMPLEMENTACIÓN Y PRUEBAS

3.1 Introducción

En este capítulo se aborda inicialmente sobre el diseño de la propuesta de solución, son expuestos los mecanismos de diseño utilizados, los diagramas de clases conformados a partir de lo definido en el análisis y los patrones empleados en la modelación. A través de métricas es validado el diseño propuesto, en aras de comprobar su correcta elaboración. Se muestra el modelo de datos, diagrama de componentes y el modelo de despliegue realizado para el sistema. Se tocan los puntos más importantes en la implementación del componente a partir del cual se realizaría la gestión del proceso Administración de Recursos Humanos del área de Transporte del CPNB. Se describen los estándares de codificación empleados para una mayor legibilidad del código. Para finalizar son especificadas las pruebas de software a utilizar para validar que el sistema desarrollado posea la máxima calidad posible.

3.2 Diseño

El diseño es el sitio donde manda la creatividad, donde los requisitos del cliente las necesidades de negocio y las consideraciones técnicas se unen en la formulación de un producto o sistema. Crea una representación o modelo del software, pero a diferencia del modelo de análisis, el modelo de diseño proporciona detalles acerca de las estructuras de datos, las arquitecturas, las interfaces y los componentes del software que son necesarios para implementar el sistema. (Pressman, 2006)

3.2.1 Mecanismos de diseño

Los mecanismos de diseño se utilizan con el objetivo de simplificar los diagramas de clases. Cada diseñador acorde a sus necesidades establece sus propios mecanismos de diseño, teniendo en cuenta los patrones y estilos seleccionados. (Cabrera Pereira., y otros, 2009)

En el módulo Persona del Sistema de Gestión de Mantenimiento Vehicular se definieron los siguientes mecanismos:

3.2.1.1 Mecanismo de diseño para las páginas clientes.

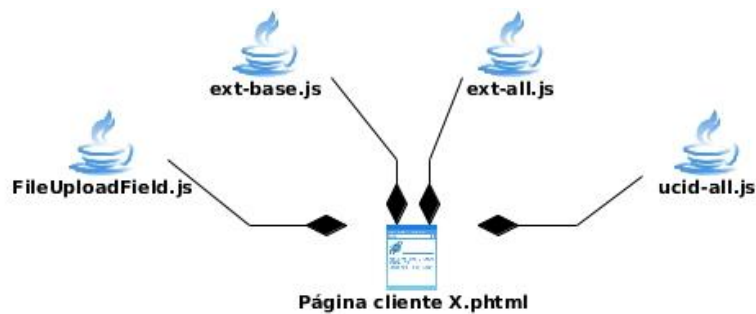


Imagen 3 Mecanismo de diseño para las páginas cliente.

ext-all.js: Se encuentra entre las clases que posee el Ext Js y es la encargada de crear los componentes visuales de la vista.

ucid-all.js: Encargada de mostrar la interfaz estándar.

ext-base.js: Es la encargada de manejar las solicitudes, respuestas y componentes de Ext Js.

FileUploadField.js: Su función es la de cargar los ficheros.

3.2.1.2 Mecanismo de diseño para las clases controladoras.

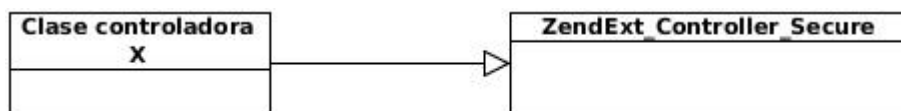


Imagen 4 Mecanismo de diseño para las clases controladoras.

Todas las clases controladoras que se definieron en el diseño del módulo Persona heredan de la clase ZendExt_Controller_Secure, ya que posee funcionalidades comunes para todas.

3.2.1.3 Mecanismo de diseño para el nombre de las clases.

A partir de las agrupaciones de requisitos funcionales que tuvieron relación entre sí fueron elaboradas las clases del diseño del módulo Persona del Sistema de Gestión de Mantenimiento Vehicular. Las clases fueron denominadas según las funcionalidades que estas realizan. Las que agrupan los requisitos adicionar, modificar y eliminar, se nombran AMENombreDeLaClase, las que agrupan listar y buscar,

LBNombreDeLaClase, mientras las que tienen como función mostrar las relaciones son nombradas según los datos que muestran, por ejemplo prendas policiales.

3.2.1.4 Mecanismos de diseño para las clases modelo.



Imagen 5 Mecanismo de diseño para las clases modelo

Todas las clases modelo que fueron definidas en el diseño heredan de la clase ZendExt_Model, la cual posee las principales funciones para el manejo de los datos.

3.2.2 Diagramas de clases del diseño

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación, contiene información como asociaciones, atributos, métodos, y dependencias. (Visconti, y otros)

En las Imágenes 12 y 14 se muestran los diagramas de clases del diseño para las agrupaciones de requisitos Gestionar herramientas, Gestionar recursos humanos, Gestionar datos de los familiares, Asociar herramientas, Quitar herramientas y Realizar búsqueda avanzada de recurso humano. En estos son representados los controladores "nombredelcontroladorController", los que respondiendo a una petición realizada desde la vista principal, a través de una relación <<link>>, contruyen las páginas cliente "nombredelapágina.phtml" mediante la relación <<build>>. Estas páginas cliente contienen uno o varios script "nombredelscript.js", según el caso, pues estos tienen implementado el código de ejecución de la página y a su vez contienen los formularios "nombredelformulario", con una relación <<include>>, los cuales toman los datos que el usuario ingrese en la página y los envía al controlador a través de la relación <<submit>>.

Para dar respuesta a las peticiones realizadas por las páginas cliente, los controladores acceden a las clases del modelo "NombredelaclaseModel", que tienen la responsabilidad de realizar las operaciones sobre la información de las tablas de la base de datos y enviarle los resultados. Estas operaciones son realizadas accediendo a las clases contenidas en el paquete Domain, donde se encuentran mapeadas las

tablas de la base de datos y las clases con el prefijo “Base_”, las que utilizan las funcionalidades brindadas por Doctrine para acceder a las tablas de la base de datos.

En caso de que la información pedida por la página cliente se encuentre en otro componente, el controlador a través de la relación <<dependencia>> accede a la clase IOC o las clases del modelo, las cuales a través de la misma relación, acceden a esta clase y a partir de la información obtenida, realizan las operaciones necesarias y le envían los resultados al controlador.

En las Imágenes 13 y 16 se muestran las clases contenidas en los paquetes del Domain representados en la Imágenes 12, 14 y 15, así como las relaciones entre las mismas.

En la Imagen 15 se muestra el diagrama de clases del diseño para los requisitos mostrar relaciones. En este se representa el controlador RecuperacionespersonaController, el cual respondiendo a la petición realizada desde la vista principal a través de la relación <<link>>, contruye la página cliente recuperacionespersona.phtml mediante la relación <<build>>. Esta página cliente contiene el script recuperacionespersona.js, el cual incluye a su vez, el formulario recuperacionespersona, con una relación <<include>>, siendo el encargado de tomar los datos ingresados por el usuario en la página y enviarlos, a través de la relación <<submit>>, al controlador para procesarlos.

Para dar respuesta a las peticiones realizadas por la página cliente, el controlador accede a la clase del modelo DatTrabajadorModel, que tiene la responsabilidad de realizar las operaciones sobre la información de las tablas de la base de datos y enviarle los resultados. Estas operaciones son realizadas accediendo a las clases contenidas en el paquete Domain, donde se encuentran mapeadas las tablas de la base de datos y las clases con el prefijo “Base_”, las que utilizan las funcionalidades brindadas por Doctrine para acceder a las tablas de la base de datos.

Para mostrar los resultados enviados desde la clase DatTrabajadorModel al controlador RecuperacionespersonaController, este crea una página cliente “nombredelarelación.phtml”, la cual haciendo uso de las librerías TCPDF, muestra los resultados en formato .pdf.

Para mas información acerca de las clases del diseño consultar epígrafe 3.2.3 y el Anexo 3.

A continuación se presentan los diagramas de clases del diseño, así como las relaciones entre estas, elaborados durante el proceso de desarrollo.

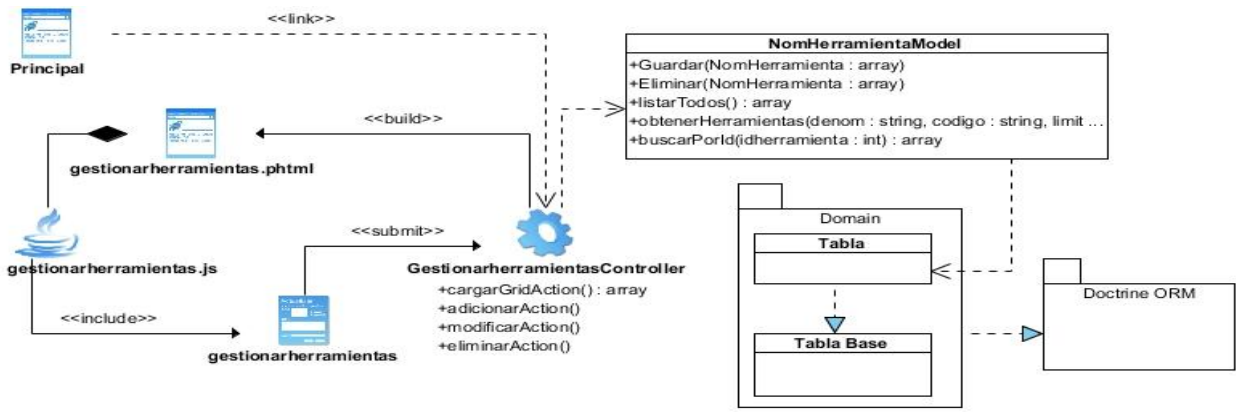


Imagen 6 Diagrama de clases Gestionar herramientas



Imagen 7 Diagrama de clase del paquete de dominio de Gestionar Herramientas

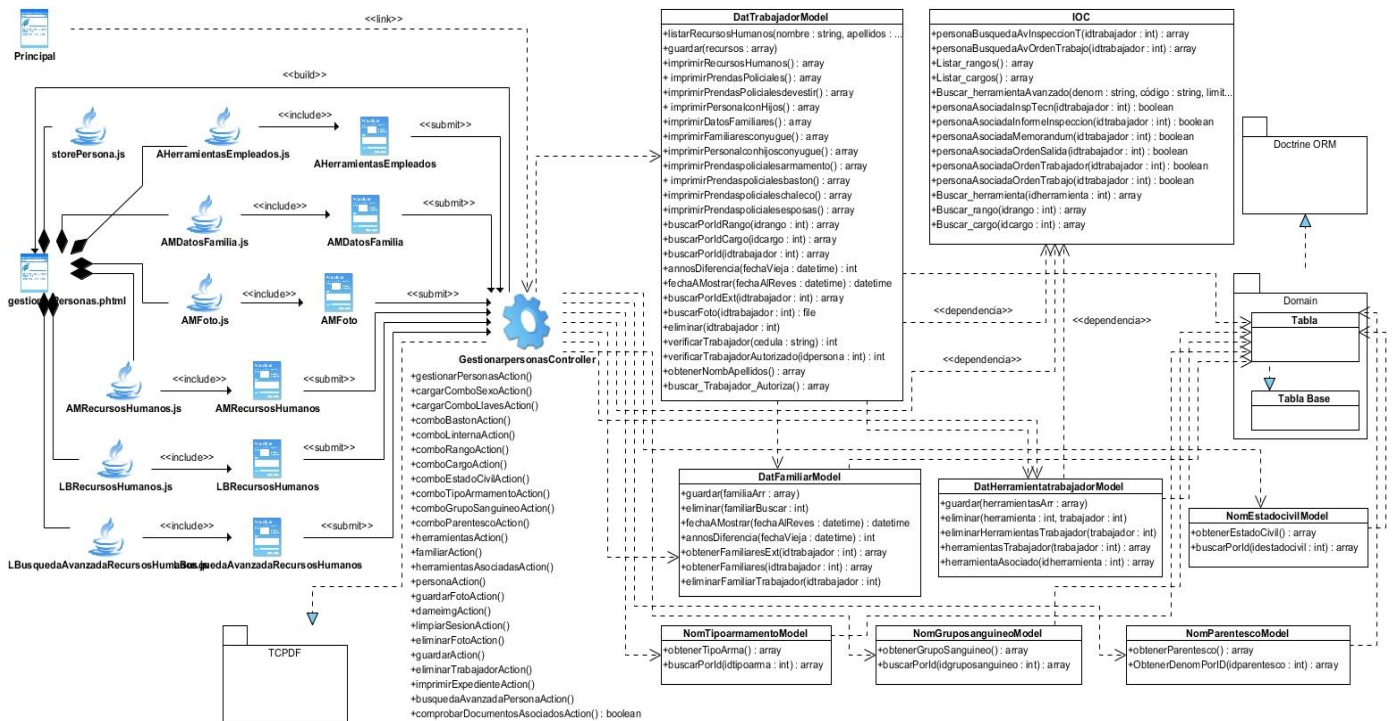


Imagen 8 Diagrama de clases Gestionar Recursos Humanos, datos familiares, asociar y quitar herramientas y realizar búsqueda avanzada

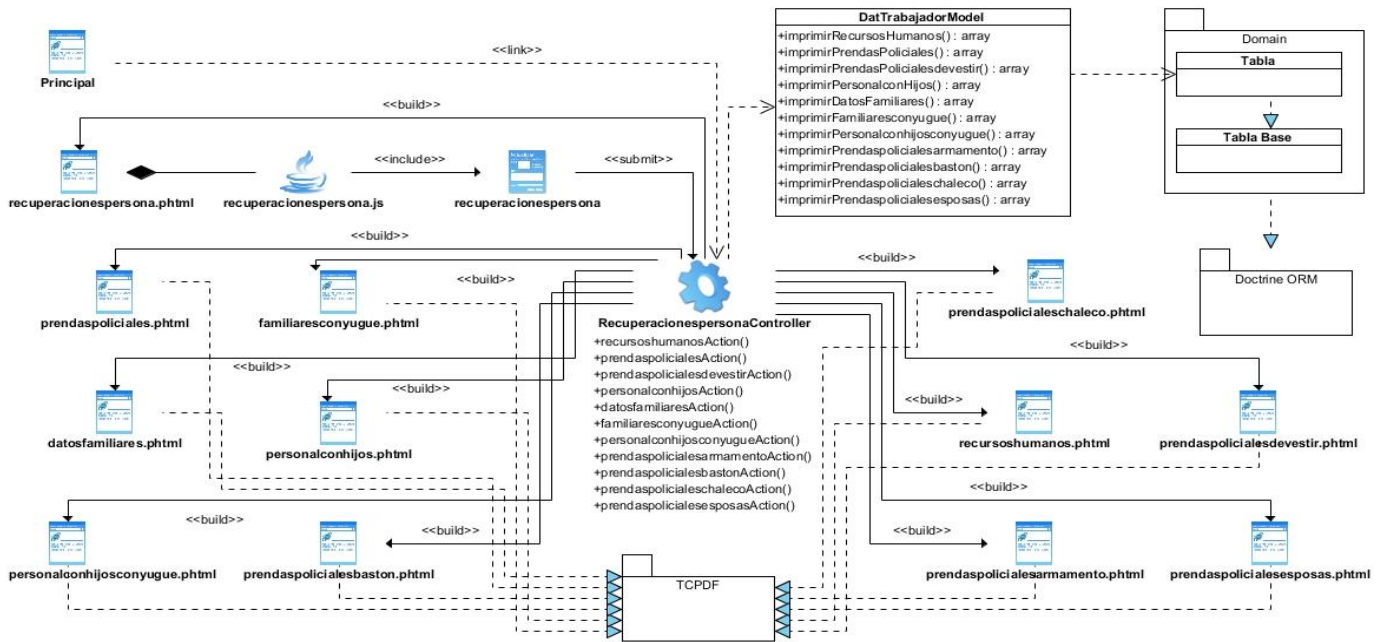


Imagen 9 Diagrama de clases Mostrar relaciones

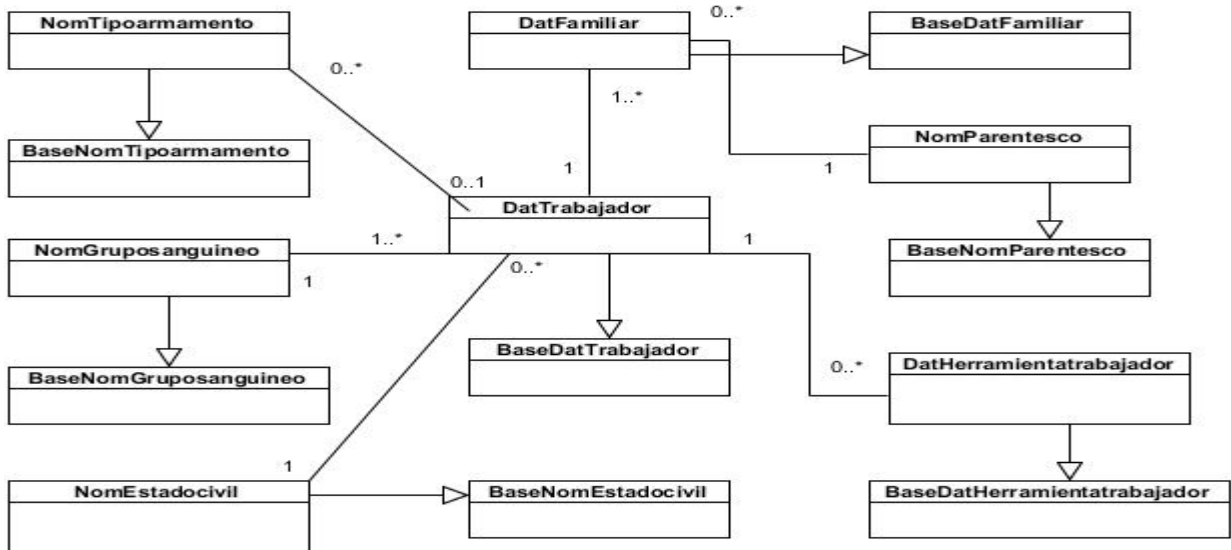


Imagen 10 Diagrama de clases del paquete de dominio del módulo Persona

3.2.3 Descripción de las clases del diseño

A continuación se presentan las descripciones de las clases controladoras y del modelo, lo cual se realiza con el objetivo de poseer un mayor entendimiento de su funcionamiento.

Tabla 4 Descripción de la clase DatTrabajadorModel

| | |
|-----------------------------------|---|
| Nombre: DatTrabajadorModel | |
| Tipos de clase: Modelo | |
| Para cada responsabilidad: | |
| Nombre | listarRecursosHumanos(\$nombre,\$apellidos,\$cedula,\$rango,\$limit,\$start) |
| Descripción | Permite listar los datos de todas las personas que se encuentran en el sistema basándose en los parámetros, que en caso de que estos sean nulos lista todas las personas registradas. |
| Nombre | Guardar(&\$recursos) |
| Descripción | A partir de un arreglo pasado con todos los datos obtenidos de una persona, realiza su adición al sistema, enviando estos a la base de datos utilizando la clase DatTrabajador. |
| Nombre | imprimirRecursosHumanos() |
| Descripción | Obtiene los datos de todas las personas que se encuentran registradas en el sistema. |
| Nombre | imprimirPrendasPoliciales() |
| Descripción | Obtiene un listado con la relación de todas las personas, que se encuentran registradas en el sistema, con las prendas policiales que poseen. |
| Nombre | imprimirPrendasPolicialesDeVestir() |

| | |
|--------------------|---|
| Descripción | Obtiene un listado con la relación de todas las personas, que se encuentran registradas en el sistema, con las prendas policiales de vestir que poseen y hayan sido registradas con anterioridad. |
| Nombre | imprimirPersonalConHijos() |
| Descripción | Obtiene un listado con la relación de todas las personas, que se encuentran registradas en el sistema, con los hijos que poseen y hayan sido insertados anteriormente. |
| Nombre | imprimirDatosFamiliares() |
| Descripción | Obtiene un listado con la relación de todas las personas, que se encuentran registradas en el sistema, con los familiares de estas que hayan sido insertados previamente. |
| Nombre | imprimirFamiliaresConyugue() |
| Descripción | Obtiene un listado con la relación de todas las personas, que se encuentran registradas en el sistema, con los datos de sus cónyuges, que le hayan sido registrados con anterioridad. |
| Nombre | imprimirPersonalConHijosConyugue() |
| Descripción | Obtiene un listado con la relación de todas las personas, que se encuentran registradas en el sistema, con los datos de sus cónyuges/hijos, que le hayan sido registrados anteriormente. |
| Nombre | imprimirPrendasPolicialesArmamento() |
| Descripción | Obtiene un listado con la relación de todas las personas, que se encuentran registradas en el sistema, con los datos de los armamentos que posean, que le hayan sido registrados previamente. |

| | |
|--------------------|--|
| Nombre | imprimirPrendasPolicialesBaston() |
| Descripción | Obtiene un listado con la relación de todas las personas, que se encuentran registradas en el sistema, con la información de si tienen o no bastón. |
| Nombre | imprimirPrendasPolicialesChaleco() |
| Descripción | Obtiene un listado con la relación de todas las personas, que se encuentran registradas en el sistema, con la información de si tienen o no chaleco. |
| Nombre | imprimirPrendasPolicialesEsposas() |
| Descripción | Obtiene un listado con la relación de todas las personas, que se encuentran registradas en el sistema, con la información de si tienen o no esposas. |
| Nombre | buscarPorIdRango(\$idrango) |
| Descripción | A partir del identificador de un rango determinado, se obtiene la cantidad de personas que tengan asociadas. |
| Nombre | buscarPorIdCargo(\$idcargo) |
| Descripción | A partir del identificador de un cargo determinado, se obtiene la cantidad de personas que tengan asociadas. |
| Nombre | buscarPorId(\$idtrabajador) |
| Descripción | A partir del identificador de una persona determinada, se obtienen todos sus datos. |
| Nombre | AnnosDiferencia(\$fechaVieja) |
| Descripción | A partir de una fecha pasada por parámetro, se muestra la diferencia de años entre esta y el día actual. |

| | |
|--------------------|--|
| Nombre | buscarPorIdExt(\$idtrabajador) |
| Descripción | A partir del identificador de una persona determinada, se obtienen todos los datos de la misma. |
| Nombre | buscarFoto(\$idtrabajador) |
| Descripción | A partir del identificador de una persona, se obtiene la foto que se guardó de la misma, del directorio donde se encuentran almacenadas todas las fotos de las personas registradas en el sistema. |
| Nombre | eliminar(\$idtrabajador) |
| Descripción | Permite eliminar los datos de una persona del sistema, recibiendo por parámetro el identificador de la misma. |
| Nombre | verificarTrabajador(\$cedula) |
| Descripción | A partir de una cédula pasada por parámetro comprueba a través del método verificarExistencia(\$cedula) que se encuentra en la clase DatTrabajador, si existe un trabajador con esa misma cédula. |
| Nombre | verificarTrabajadorAutorizado(\$idpersona) |
| Descripción | Permite verificar a partir del identificador de una persona si es quien realiza las autorizaciones en el área de Transporte del CPNB. |
| Nombre | obtenerNombApellidos() |
| Descripción | Permite obtener todos los nombres y apellidos de las personas registradas en el sistema |
| Nombre | buscarTrabajadorAutoriza() |

| | |
|--------------------|---|
| Descripción | Permite obtener los datos de la persona que realiza las autorizaciones en el área de Transporte del CPNB. |
|--------------------|---|

La descripción de las restantes clases del diseño definidas, se encuentra en el Anexo 3.

3.2.4 Patrones de diseño

Un patrón de diseño describe un problema que ocurre frecuentemente en el campo de la construcción de software y su respectiva solución; puede ser empleado muchas veces, en diferentes contextos, sin tener que duplicar el diseño. Se trata de un elemento de diseño que puede ser reutilizado (Facultad de Ingeniería Electrónica y Telecomunicaciones)

En el diseño del módulo Persona del Sistema de Gestión de Mantenimiento Vehicular para la gestión del proceso Administración de Recursos Humanos del área de Transporte del CPNB, fueron empleados los siguientes patrones de diseño:

3.2.4.1 MVC

La implementación de este patrón en dicho módulo se realizó de la siguiente manera:

La Vista engloba la capa y la lógica de presentación, en la misma es manejado todo el flujo web. Le brinda al usuario la posibilidad de interactuar con el Controlador mediante los mensajes de eventos y le permite visualizar las respuestas a sus peticiones.

El Controlador por su parte representa el gestor de eventos, el cual está compuesto por las clases controladoras. Este atiende los pedidos que llegan desde la Vista accediendo al paquete models.

El Modelo se encuentra dividido en los paquetes bussines, el cual encierra las clases donde se realiza la lógica del negocio, y domain, donde están comprendidas las clases del dominio, las que se encargan de leer y/o escribir datos en las tablas de la base de datos.

3.2.4.2 Fachada

La utilización de este patrón está reflejada en la creación y empleo de la clase IOC, ya que por la necesidad de integración entre los módulos del Sistema de Gestión de Mantenimiento Vehicular, era necesaria la implementación de una clase fachada donde fueran publicados los servicios necesarios por estos, facilitando su interacción.

3.2.4.3 Mediador

La aplicación de este patrón fue en la creación de la tabla `DatHerramientatrabajador`, la cual fue necesaria dado que la relación entre las tablas `NomHerramientas` y `DatTrabajador`, era de mucho a mucho, lo cual provocaba que la comunicación entre las mismas fuera compleja y la dependencia que surgía era difícil de entender, quedando de esta forma un comportamiento distribuido entre varias clases.

3.2.4.4 Solitario/Singleton

Este patrón fue utilizado en las clases del dominio, donde los métodos contenidos en estas se hicieron de forma estática, con lo cual se garantizaba que pudieran ser accedidos sin crear una instancia de las mismas. Por ejemplo: para acceder al método `listarTodos()` de la clase del domain `NomHerramienta`, desde la clase `NomHerramientaModel`, del modelo, solo era necesario poner `NomHerramienta::listarTodos()`. También fue usado en las clases contenidas en la Vista ya que por cada petición es realizada una interfaz de usuario.

3.2.4.5 Controlador

El empleo de este se puede observar en la creación de una clase controladora para los diferentes procesos que se realizan en el módulo `Persona` o que están relacionados con este. Para la gestión de las herramientas, los recursos humanos y mostrar todas las relaciones del personal del área de Transporte del CPNB, fueron creadas las clases `GestionarherramientasController`, `GestionarpersonasController` y `RecuperacionespersonaController` respectivamente.

3.2.4.6 Experto

Fue usado en todas las clases definidas ya que estas contaban con la información necesaria para cumplir con la responsabilidad asignada.

3.2.4.7 Alta cohesión

Su empleo está dado en que fueron asignadas responsabilidades a las clases de forma tal que la cohesión siguiera siendo alta, o sea, cada clase se encargará de realizar solamente las funciones que estén en correspondencia con la responsabilidad que posea.

3.2.4.8 Bajo acoplamiento

Un ejemplo donde se puede observar la utilización de este patrón es que la clase GestionarpersonasController, se integra con el módulo Vehículo y no conoce la existencia de la clase DatInspecciontecnica, que se encuentra en dicho módulo y contiene los datos que son solicitados, sino que interactúa con la fachada, encargada de pasarle los valores pedidos.

3.2.5 Métricas para evaluar el diseño propuesto

Las métricas son una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado. Permiten descubrir y corregir problemas potenciales antes de convertirse en defectos catastróficos. Se emplean con el objetivo de llevar un control de la calidad del producto que se está desarrollando, evaluar la efectividad del proceso y mejorar la calidad del trabajo. (Pressman, 2005)

Una vez realizado el diseño del módulo Persona, se dio paso a su evaluación, para esto se utilizaron las métricas orientadas a objetos, específicamente a las clases, ya que las medidas y métricas para una clase individual, la jerarquía y las colaboraciones de estas permiten al ingeniero de software evaluar la calidad del diseño propuesto. (Pressman, 2005)

Métricas empleadas para realizar la evaluación del módulo Persona:

3.2.5.1 Métricas propuestas por Lorenz y Kidd:

En su libro sobre las métricas orientadas a objetos, dividen las métricas basadas en clases en cuatro amplias categorías: tamaño, herencia, valores internos y valores externos. Las orientadas al tamaño para las clases orientadas a objetos se centran en el recuento de atributos y operaciones para cada clase individual y los valores promedio para el sistema orientado a objetos como un todo. (Pressman, 2005).

La métrica empleada fue **Tamaño de clase** (TC), pues se ajusta para evaluar al diseño realizado y es fácil de usar. Debido a la cantidad de clases que posee el módulo, solo se le aplicó a las clases modelos y controladoras de los diagramas realizados.

Para la evaluación de las clases fueron utilizados umbrales para el tamaño general, la responsabilidad, la complejidad y la reutilización de las clases.

Tabla 5 Umbrales para la TC

| Clasificación | Valores de los umbrales |
|----------------|------------------------------------|
| Pequeño | \leq Promedio de operaciones(PO) |
| Medio | $>$ PO y $\leq 2*PO$ |
| Grande | $> 2*PO$ |

Tabla 6 Tamaño de las clases según sus atributos y operaciones

| No. | Nombre | Cantidad de atributos | Cantidad de operaciones | Tamaño |
|-----|---------------------------------|-----------------------|-------------------------|---------|
| 1 | GestionarherramientasController | 0 | 4 | Pequeño |
| 2 | RecuperacionespersonaController | 0 | 12 | Medio |
| 3 | gestionarPersonasController | 1 | 24 | Grande |
| 4 | NomHerramientaModel | 0 | 5 | Pequeño |
| 5 | DatTrabajadorModel | 0 | 25 | Grande |
| 6 | DatFamiliarModel | 0 | 7 | Pequeño |
| 7 | DatHerramientatrabajadorModel | 0 | 5 | Pequeño |
| 8 | NomParentescoModel | 0 | 4 | Pequeño |
| 9 | NomGruposanguineoModel | 0 | 4 | Pequeño |
| 10 | NomTipoarmamentoModel | 0 | 4 | Pequeño |
| 11 | NomEstadocivilModel | 0 | 4 | Pequeño |

Tabla 7 Cantidad de clases por clasificación

| Clasificación | Cantidad de clases | Responsabilidad de las clases | Complejidad de implementación de las clases | Reutilización de las clases |
|---------------|--------------------|-------------------------------|---|-----------------------------|
| Pequeño | 8 | Baja | Baja | Alta |
| Medio | 1 | Medio | Medio | Medio |
| Grande | 2 | Alta | Alta | Baja |

Tabla 8 Resultado general de la métrica

| Cantidad de clases | Cantidad de clases pequeñas | Cantidad de clases grandes | Cantidad de clases medias | Promedio de atributos | Promedio de operaciones |
|--------------------|-----------------------------|----------------------------|---------------------------|-----------------------|-------------------------|
| 11 | 8 | 2 | 1 | 0 | 9 |

Haciendo un análisis de los resultados obtenidos en el empleo de esta técnica, se puede concluir que el diseño realizado es simple, tiene una calidad aceptable, pues la mayoría de las clases que fueron analizadas se encuentran dentro de la categoría de pequeñas, los valores de calidad definidos por la métrica utilizada no fueron afectados, demostrándose en los pequeños valores de TC alcanzados, la implementación de forma general es sencilla, se disminuye en gran medida la responsabilidad de las clases y las pruebas pueden ser realizadas con facilidad.

3.3 Modelo de datos

El modelo de datos es un conjunto de conceptos, reglas y convenciones que permite describir y manipular los datos de un cierto mundo real que deseamos almacenar en la base datos. (Rivera)

En el modelo de datos elaborado para el módulo Persona del Sistema de Gestión de Mantenimiento Vehicular se visualizan las entidades, sus atributos y las relaciones entre ellas. Son las encargadas de almacenar todos los datos necesarios para que el módulo pueda ofrecer las funcionalidades que fueron

definidas en el análisis. Para más información sobre el modelo de datos, consultar documento entregable 010. Diccionario de Datos MV-CPNB. A continuación se presenta el modelo de datos realizado:

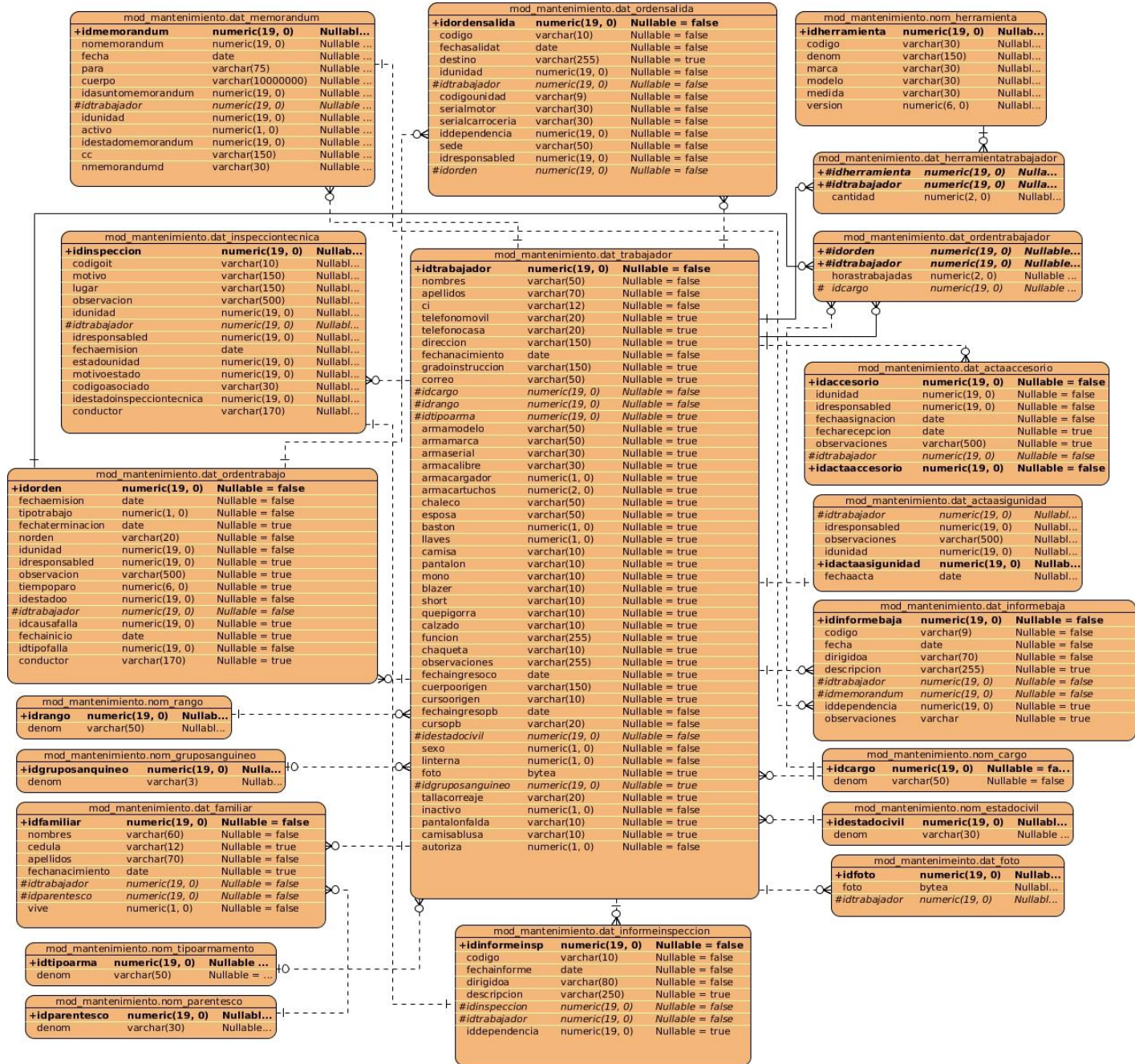


Imagen 11 Modelo de datos

3.4 Implementación

3.4.1 Diagrama de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Se realizan con el objetivo de poseer una vista de forma general del sistema a partir de las dependencias e integraciones de los componentes y módulos. En la Imagen 18 se muestra el diagrama de componentes del Sistema de Gestión de Mantenimiento Vehicular, desde la perspectiva del componente Persona. En este se representa la integración entre los mismos, o sea los servicios que brindan unos y son consumidos por otros y viceversa.

A continuación se presenta el diagrama de componentes elaborado desde la perspectiva del componente Persona:

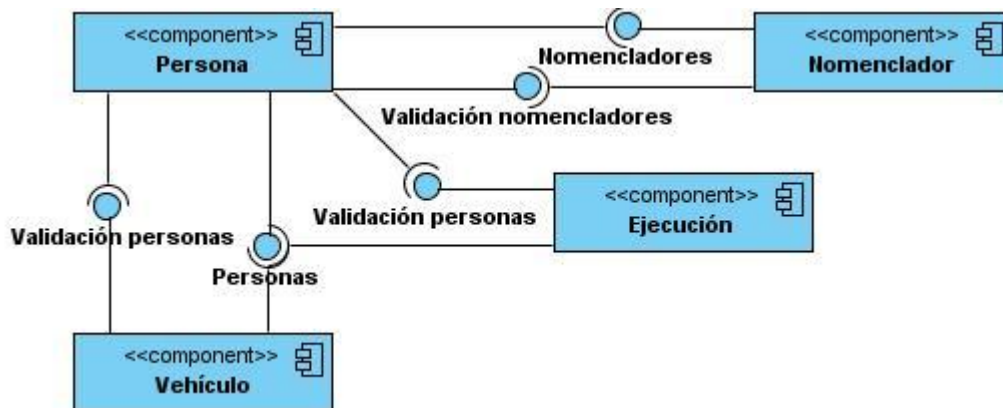


Imagen 12 Diagrama de componentes: Perspectiva del componente Persona

3.4.2 Modelo de despliegue

Los diagramas de despliegue visualizan la disposición física de los distintos recursos computacionales que componen un sistema y el reparto de los componentes sobre dichos nodos. (Universidad de Castilla La Mancha, 2011-2012)

Para el modelo realizado se definió una arquitectura cliente-servidor de la siguiente manera:

Cliente: Accede a la aplicación a través de un computador, donde es ejecutada mediante el navegador Mozilla Firefox, sobre cualquier sistema operativo.

Servidor Web: El servidor de aplicaciones empleado donde radica la lógica de negocio de la aplicación es el Servidor Web Apache2 utilizando el lenguaje PHP.

Servidor de Base de datos: Servidor de Datos PostgreSQL 8.3.

A continuación se presenta el modelo de despliegue elaborado para el Sistema de Gestión de Mantenimiento Vehicular v 1.0:

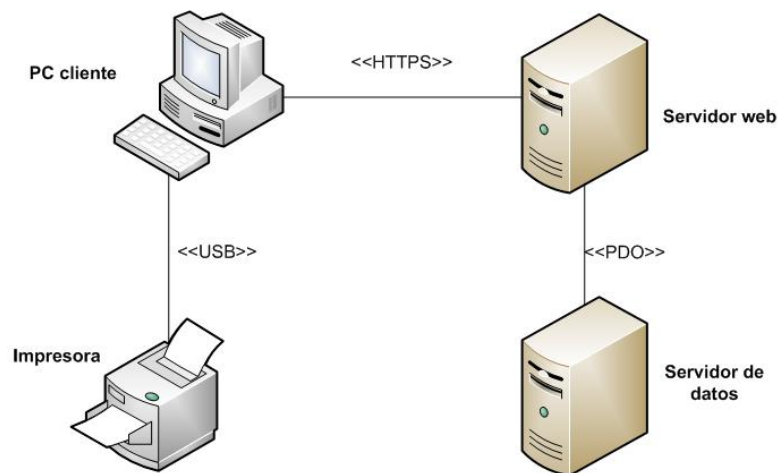


Imagen 13 Modelo de despliegue.

3.4.3 Estándares de codificación

Los estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, la comprensión y mantenimiento del código. (Departamento de Electrónica, Sistemas e Informática Coordinación Docente de Programación)

A continuación se especifican los estándares de codificación que fueron utilizados en el desarrollo del módulo Persona:

3.4.3.1 Nomenclatura de las clases

- Para las clases controladoras el nombre comenzará con la primera letra en mayúscula y el resto en minúscula, además estará dado según la función que realiza y se le adicionará al final "Controller". Ejemplo: GestionarpersonaController.

- El nombre de las clases del modelo que se encuentran dentro de la carpeta “bussines” comenzará con la primera letra en mayúscula y el resto en minúscula. A este nombre se le agregará al final “Model” y al inicio se le colocará “Nom”, en caso de que sea un nomenclador, de lo contrario será Dat, por ejemplo: DatFamiliarModel, NomHerramientaModel.
- Las clases del dominio que se encuentran dentro de la carpeta “domain”, son generadas mediante un mapeo a la base de datos realizado por la herramienta Doctrine Generator v 3.0, la cual fue desarrollada en el CEIGE y son nombradas igual que las tablas de la base de datos. Ejemplo: DatTrabajador.
- Las clases contenidas en la carpeta “generated” son generadas mediante un mapeo a la base de datos realizado por la herramienta Doctrine Generator v 3.0, la cual fue desarrollada en el CEIGE y delante del nombre se le coloca “Base”.
- EL nombre de las clases de la vista comenzará con la primera letra en mayúscula y el resto en minúscula, en caso de que este sea compuesto, se utilizará la notación PascalCasing. A este nombre se le agregaran al principio, las letras en mayúscula de las funciones que se realizarán a través de estas clases. Ejemplo: LBRecursosHumanos(listar y buscar), AMDatosFamilia(adicionar o asociar, y modificar), AHerramientasEmpleados.

3.4.3.2 Nomenclatura de los métodos o funciones

El nombre de los métodos o funciones de una clase comenzará en minúscula, en caso de que sea compuesto se utilizará la notación CamelCasing, o sea, seguido del primer nombre, comenzará el segundo con mayúscula. En caso de que sea una función de una clase controladora se le agregará al final la palabra “Action”. Ejemplo: guardarAction, comprobarDocumentosAsociadosAction, buscar.

3.4.3.3 Nomenclatura de las variables

Las variables serán nombradas comenzando en minúscula, en caso de que el nombre de estas sea compuesto, se utilizará la notación CamelCasing, o sea, seguido del primer nombre, comenzará el segundo en mayúscula. En la capa de la Vista, las variables que contengan componentes que formen parte de la interfaz, se les colocará delante un sufijo que indique el tipo de componente que contiene. Ejemplo: arrayHerramientasAsociadas, btnModificarFamilia, storeDatosFamilia, formularioPersona, tabPersona.

3.4.4 Publicación de servicios entre componentes

Para cada uno de los componentes fueron publicados en sus clases Services, los mismos métodos que son utilizados en estos para la recuperación de los datos en las clases del modelo y del dominio, realizándose de esta manera con el objetivo de reutilizar código ya existente. Los servicios fueron implementados de modo que reciban muchos o ningún parámetro, siempre que fuera posible, disminuyendo con esto la cantidad de servicios a publicar.

El módulo Persona cuenta con las clases Services, DatTrabajadorService y DatHerramientatrabajadorService, las cuales contienen todos los servicios que son consumidos por los restantes componentes del Sistema de Gestión de Mantenimiento Vehicular, con los que se tienen una interrelación.

A continuación se muestran los servicios que son brindados por el componente:

Tabla 9 Servicios que brinda Persona

| Servicios | Componentes que lo utilizan | Descripción |
|-------------------------------|-----------------------------|--|
| rango_Asociado | Nomencladores | A partir del identificador de un rango determinado, obtiene la cantidad de personas que tienen ese rango asociado. |
| cargo_Asociado | Nomencladores | Obtiene la cantidad de personas que tienen asociado un cargo determinado, a partir del identificador de este. |
| listar_RecursosHumanos | Ejecución | Muestra un listado de todas las personas que se encuentran registradas en el sistema si los parámetros son nulos, en caso contrario, realiza una búsqueda de las personas que sus datos concuerden o sean similares a los parámetros |
| | Vehículo | |

| | | |
|-----------------------------------|---------------|--|
| | | pasados. |
| buscar_Trabajador | Vehículo | A partir del identificador de una persona, se obtienen todos los datos de la misma. |
| | Ejecución | |
| buscar_Trabajador_Autoriza | Ejecución | Obtiene todos los datos de la persona encargada de autorizar los trabajos en el área de Transporte del CPNB. |
| | Vehículo | |
| herramientaAsociada | Nomencladores | Obtiene la cantidad de personas que están asociadas a una herramienta determinada, de la cual sea pasado el identificador por parámetro. |

3.5 Pruebas de software

3.5.1 Pruebas de caja blanca

Para utilizar la **Prueba del Camino Básico**, primeramente se hace necesario el calculo de la complejidad ciclomática del algoritmo que vaya a ser analizado, para lo cual se deben enumerar las sentencias de código de este y a partir de ahí elaborar el grafo de flujo de esta funcionalidad. Las sentencias enumeradas del método Guardar(&\$recursos) se encuentran en las imágenes 29 y 30 del Anexo 4.

A continuación en la imagen 20 se muestra el grafo de flujo asociado a la funcionalidad.

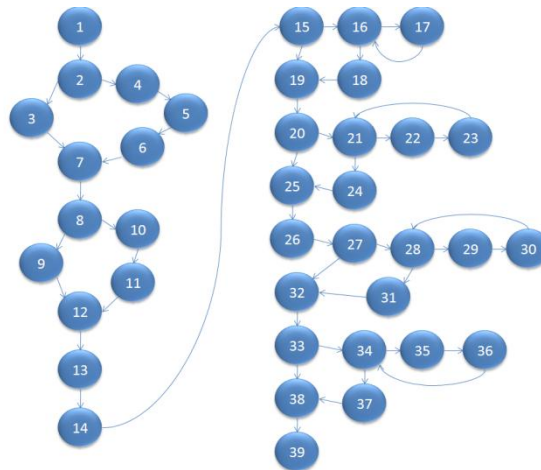


Imagen 14 Grafo de flujo asociado al algoritmo Guardar(&\$recursos).

3.5.1.1 Cálculo de la complejidad ciclomática a partir de un segmento de código

La complejidad ciclomática es una métrica del software que proporciona una medición cuantitativa de la complejidad lógica de un programa. El valor calculado como complejidad ciclomática define el número de caminos independientes del conjunto básico de un programa y ofrece un límite superior para el número de pruebas que se deben realizar para asegurar que sea ejecutada cada sentencia al menos una vez. (Pressman, 2005)

La complejidad ciclomática de un código determinado puede ser calculada de tres maneras diferentes. Al calcular la complejidad del método Guardar(&\$recursos) fueron utilizadas las tres formas posibles con el objetivo de verificar que el calculo se habia realizado correctamente, si los resultados obtenidos en cada una era el mismo. Las fórmulas para realizar dicho cálculo son:

1. $V(G) = (A - N) + 2$, $V(G) = (48 - 39) + 2$, $V(G) = 11$

Siendo A la cantidad total de aristas del grafo y N la cantidad de nodos.

2. $V(G) = P + 1$, $V(G) = 10 + 1$, $V(G) = 11$

Siendo P la cantidad de nodos predicado (son aquellos de los cuales parten dos o más aristas)

3. $V(G) = R$, $V(G) = 11$

Siendo R la cantidad de regiones que posee el grafo.

En cada una de las fórmulas V (G) ha representado el valor del cálculo. A partir de los resultados obtenidos en cada uno, se puede determinar que la complejidad ciclomática del código analizado es 11, que a su vez es el número de caminos posibles a circular el flujo y el límite superior de casos de prueba que se le pueden aplicar a dicho código.

A continuación se muestran los caminos básicos por donde puede circular el flujo:

Camino básico # 1: 1-2-3-7-8-9-12-13-14-15-19-20-25-26-27-32-33-38-39

Camino básico # 2: 1-2-4-5-6-7-8-9-12-13-14-15-19-20-25-26-27-32-33-38-39

Camino básico # 3: ...-8-10-11-12-13-14-15-19-20-25-26-27-32-33-38-39

Camino básico # 4: ...-15-16-18-19-20-25-26-27-32-33-38-39

Camino básico # 5: ...-15-16-17-16-18-19-20-25-26-27-32-33-38-39

Camino básico # 6: ...-20-21-24-25-26-27-32-33-38-39

Camino básico # 7: ...-20-21-22-23-21-24-25-26-27-32-33-38-39

Camino básico # 8: ...-27-28-31-32-33-38-39

Camino básico # 9: ...-27-28-29-30-28-31-32-33-38-39

Camino básico # 10: ...-33-34-37-38-39

Camino básico # 11: ...-33-34-35-36-34-37-38-39

Los tres puntos suspensivos (...) de los caminos del 3 en adelante indican que cualquier vía antes de la estructura de control es aceptable.

Para cada uno de los caminos obtenidos se realiza un caso de prueba. Los casos de prueba realizados son los siguientes:

3.5.1.2 Caso de prueba para el camino básico #1.

Camino: 1-2-3-7-8-9-12-13-14-15-19-20-25-26-27-32-33-38-39

Descripción: El dato de entrada cumplirá con el siguiente requisito:

El parámetro \$recursos, no está vacío, contiene todos los datos de una persona determinada, excepto las herramientas asociadas y las que se le quitan, así como los datos de sus familiares y los eliminados del sistema. Además este parámetro posee el identificador de una persona por lo cual la acción que se va a realizar es la de modificar los datos que hayan sido cambiados.

Entrada: \$recursos = [[idtrabajador] => 9000008, [armacalibre] =>, [armacargador] =>,[armacartuchos] =>,[armamarca] =>,[armamodelo] =>, [armaserial] =>,[baston] => 0,[blazer] =>, [calzado] => 39, [camisa] => M, [chaleco] =>,[chaqueta] =>,[ci] => 15.910.482, [correo] => chicohyuday@hotmail.com, [cuerpoorigen] =>,[cursoorigen] =>,[cursopb] => N/P, [funcion] => MECANICO DE LA CPNB, [direccion] => el valle, urb. San antonio sector III, vereda 7, casa N° 12, [esposa] =>,[fechaingresoco] =>,[fechaingresopb] => 01/04/2011, [fechanacimiento] => 13/06/1980, [gradoinstruccion] => 6 GRADO, [idgruposanguineo] => 2, [idcargo] => 9000005, [idestadocivil] => 102, [idrango] => 1, [idtipoarma] =>,[llave] => 0, [linterna] => 0, [mono] =>,[nombres] => Luis Alberto, [observaciones] =>,[pantalon] => 36, [apellidos] => Garcia Blanco, [quepigorra] =>,[sexo] => 0, [short] =>,[telefonocasa] => 0212-8809482, [telefonomovil] => 0416-1935873, [tallacorrea] =>,[pantalonfalda] =>,[camisablusa] =>,[autoriza] =>,[herramientas] => Array(),[herramientasEliminadas] => Array(),[familiares] => Array(),[familiaresEliminados] => Array()]

Resultados esperados: Se espera que los datos que hayan sido cambiados se modifiquen.

3.5.1.3 Caso de prueba para el camino básico #2.

Camino: 1-2-4-5-6-7-8-9-12-13-14-15-19-20-25-26-27-32-33-38-39

Descripción: El dato de entrada cumplirá con el siguiente requisito:

El parámetro \$recursos, no está vacío, contiene todos los datos de una persona determinada, excepto las herramientas asociadas y las que se le quitan, así como los datos de sus familiares y los que van a ser eliminados del sistema.

Entrada: \$recursos = [[idtrabajador] =>, [armacalibre] => 9mm, [armacargador] => 2,[armacartuchos] => 50,[armamarca] => BERETA,[armamodelo] => PX4, [armaserial] => PX94113,[baston] => 0,[blazer] =>, [calzado] => 39, [camisa] => M, [chaleco] =>,[chaqueta] =>,[ci] => 13.265.365, [correo] => chicohyuday@hotmail.com, [cuerpoorigen] =>,[cursoorigen] => Policía Metropolitana,[cursopb] => N/P, [funcion] => ASISTENTE DE MECANICO DE LA CPNB, [direccion] => el valle, urb. San antonio sector III,

vereda 7, casa N° 23, [esposa] => 244038,[fechaingresoco] => 15/03/1990,[fechaingresopb] => 01/04/2011, [fechanacimiento] => 15/09/1980, [gradoinstruccion] => 6 GRADO, [idgruposanguineo] => 1, [idcargo] => 9000005, [idestadocivil] => 102, [idrango] => 1, [idtipoarma] =>,[llave] => 0, [linterna] => 0, [mono] =>,[nombres] => Ernesto, [observaciones] =>,[pantalon] => 36, [apellidos] => Perez Hernandez, [quepigorra] =>,[sexo] => 0, [short] =>,[telefonocasa] => 0212-8839422, [telefonomovil] => 0416-1955623, [tallacorrea] =>,[pantalonfalda] =>36,[camisablusa] =>,[autoriza] =>,[herramientas] => Array(),[herramientasEliminadas] => Array(),[familiares] => Array(),[familiaresEliminados] => Array()

Resultados esperados: Se espera que sea adicionada una nueva persona en el sistema.

3.5.1.4 Caso de prueba para el camino básico #3.

Camino: ...8-10-11-12-13-14-15-19-20-25-26-27-32-33-38-39

Descripción: El dato de entrada cumplirá con el siguiente requisito:

El parámetro \$recursos, no está vacío, contiene todos los datos de una persona determinada, excepto las herramientas asociadas y las que se le quitan, así como los datos de sus familiares y los eliminados del sistema.

Entrada: \$recursos = [[idtrabajador] => 9000008, [armacalibre] =>, [armacargador] =>,[armacartuchos] =>,[armamarca] =>,[armamodelo] =>, [armaserial] =>,[baston] => 0,[blazer] =>, [calzado] => 39, [camisa] => M, [chaleco] =>,[chaqueta] =>,[ci] => 15.910.482, [correo] => chicohyuday@hotmail.com, [cuerpoorigen] =>,[cursoorigen] =>,[cursopb] => N/P, [funcion] => MECANICO DE LA CPNB, [direccion] => el valle, urb. San antonio sector III, vereda 7, casa N° 12, [esposa] =>,[fechaingresoco] =>,[fechaingresopb] => 01/04/2011, [fechanacimiento] => 13/06/1980, [gradoinstruccion] => 6 GRADO, [idgruposanguineo] => 2, [idcargo] => 9000005, [idestadocivil] => 102, [idrango] => 1, [idtipoarma] =>,[llave] => 0, [linterna] => 0, [mono] =>,[nombres] => Luis Alberto, [observaciones] =>,[pantalon] => 36, [apellidos] => Garcia Blanco, [quepigorra] =>,[sexo] => 0, [short] =>,[telefonocasa] => 0212-8809482, [telefonomovil] => 0416-1935873, [tallacorrea] =>,[pantalonfalda] =>,[camisablusa] =>,[autoriza] =>0,[herramientas] => Array(),[herramientasEliminadas] => Array(),[familiares] => Array(),[familiaresEliminados] => Array()]

Resultados esperados: Se espera que el campo "autoriza" tome valor 0, y que la persona que se esté adicionando o modificando, no sea la responsable de autorizar los documentos que se manejan en el área de Transporte del CPNB.

3.5.1.5 Caso de prueba para el camino básico #5.

Camino: ...-15-16-17-16-18-19-20-25-26-27-32-33-38-39

Descripción: El dato de entrada cumplirá con el siguiente requisito:

El parámetro \$recursos, no está vacío, contiene todos los datos de una persona determinada, así como los identificadores de los familiares que serán eliminados sus datos del sistema, excepto las herramientas asociadas y las que se le quitan, y los datos de los familiares adicionados.

Entrada: \$recursos = [[idtrabajador] =>, [armacalibre] => 9mm, [armacargador] => 2,[armacartuchos] => 50,[armamarca] => BERETA,[armamodelo] => PX4, [armaserial] => PX94113,[baston] => 0,[blazer] =>, [calzado] => 39, [camisa] => M, [chaleco] =>,[chaqueta] =>,[ci] => 13.265.365, [correo] => chicohyuday@hotmail.com, [cuerpoorigen] =>,[cursoorigen] => Policía Metropolitana,[cursopb] => N/P, [funcion] => ASISTENTE DE MECANICO DE LA CPNB, [direccion] => el valle, urb. San antonio sector III, vereda 7, casa N° 23, [esposa] => 244038,[fechaingresoco] => 15/03/1990,[fechaingresopb] => 01/04/2011, [fechanacimiento] => 15/09/1980, [gradoinstruccion] => 6 GRADO, [idgruposanguineo] => 1, [idcargo] => 9000005, [idestadocivil] => 102, [idrango] => 1, [idtipoarma] =>,[llave] => 0, [linterna] => 0, [mono] =>,[nombres] => Ernesto, [observaciones] =>,[pantalon] => 36, [apellidos] => Perez Hernandez, [quepigorra] =>,[sexo] => 0, [short] =>,[telefonocasa] => 0212-8839422, [telefonomovil] => 0416-1955623, [tallacorrea] =>,[pantalonfalda] =>36,[camisablusa] =>,[autoriza] =>,[herramientas] => Array(),[herramientasEliminadas] => Array(),[familiares] => Array(),[familiaresEliminados] => Array([0] => 9000013)]

Resultados esperados: Se espera que sean eliminados los datos de los familiares, que fueron pasados en el parámetro \$recursos, de la persona que se esté adicionando o modificando sus datos en el sistema.

3.5.1.6 Caso de prueba para el camino básico #7.

Camino: ...-20-21-22-23-21-24-25-26-27-32-33-38-39

Descripción: El dato de entrada cumplirá con el siguiente requisito:

El parámetro \$recursos, no está vacío, contiene todos los datos de una persona determinada, así como los de los familiares que le serán adicionados en el sistema, excepto las herramientas asociadas y las que se le quitan, y los datos de los familiares que serán eliminados.

Entrada: \$recursos = [[idtrabajador] => 9000008, [armacalibre] =>, [armacargador] =>,[armacartuchos] =>,[armamarca] =>,[armamodelo] =>, [armaserial] =>,[baston] => 0,[blazer] =>, [calzado] => 39, [camisa] => M, [chaleco] =>,[chaqueta] =>,[ci] => 15.910.482, [correo] => chicohyuday@hotmail.com, [cuerpoorigen] =>,[cursoorigen] =>,[cursopb] => N/P, [funcion] => MECANICO DE LA CPNB, [direccion] => el valle, urb. San antonio sector III, vereda 7, casa N° 12, [esposa] =>,[fechaingresoco] =>,[fechaingresopb] => 01/04/2011, [fechanacimiento] => 13/06/1980, [gradoinstruccion] => 6 GRADO, [idgruposanguineo] => 2, [idcargo] => 9000005, [idestadocivil] => 102, [idrango] => 1, [idtipoarma] =>,[llave] => 0, [linterna] => 0, [mono] =>,[nombres] => Luis Alberto, [observaciones] =>,[pantalon] => 36, [apellidos] => Garcia Blanco, [quepigorra] =>,[sexo] => 0, [short] =>,[telefonocasa] => 0212-8809482, [telefonomovil] => 0416-1935873, [tallacorrea] =>,[pantalonfalda] =>,[camisablusa] =>,[autoriza] =>,[herramientas] => Array(),[herramientasEliminadas] => Array(),[familiares] => Array([0] => Array([id_familiar] => 9000012, [cedula] => 6.585.887, [fechanacimiento] => 1950-07-19, [nombres] => Vicenta Josefina, [apellidos] => Bracho de Cariel, [parentesco] => 2, [denomParentesco] => Madre, [apellidosNombre] => Bracho de Cariel Vicenta Josefina, [vive] => 1)),[familiaresEliminados] => Array())

Resultados esperados: Se espera que sean adicionados los datos de los familiares de la persona que se esté adicionando o modificando.

3.5.1.7 Caso de prueba para el camino básico #9

Camino: ...-27-28-29-30-28-31-32-33-38-39

Descripción: El dato de entrada cumplirá con el siguiente requisito:

El parámetro \$recursos, no está vacío, contiene todos los datos de una persona determinada, así como los identificadores de las herramientas que se van quitar, excepto las herramientas asociadas y los datos de los familiares que se van a registrar y/o eliminar del sistema.

Entrada: \$recursos = [[idtrabajador] => 9000008, [armacalibre] =>, [armacargador] =>,[armacartuchos] =>,[armamarca] =>,[armamodelo] =>, [armaserial] =>,[baston] => 0,[blazer] =>, [calzado] => 39, [camisa] => M, [chaleco] =>,[chaqueta] =>,[ci] => 15.910.482, [correo] => chicohyuday@hotmail.com,

[cuerpoorigen] =>,[cursoorigen] =>,[cursopb] => N/P, [funcion] => MECANICO DE LA CPNB, [direccion] => el valle, urb. San antonio sector III, vereda 7, casa N° 12, [esposa] =>,[fechaingresoco] =>,[fechaingresopb] => 01/04/2011, [fechanacimiento] => 13/06/1980, [gradoinstruccion] => 6 GRADO, [idgruposanguineo] => 2, [idcargo] => 9000005, [idestadocivil] => 102, [idrango] => 1, [idtipoarma] =>,[llave] => 0, [linterna] => 0, [mono] =>,[nombres] => Luis Alberto, [observaciones] =>,[pantalon] => 36, [apellidos] => Garcia Blanco, [quepigorra] =>,[sexo] => 0, [short] =>,[telefonocasa] => 0212-8809482, [telefonomovil] => 0416-1935873, [tallacorrea] =>,[pantalonfalda] =>,[camisablusa] =>,[autoriza] =>,[herramientas] => Array(),[herramientasEliminadas] => Array([0] => 9000015),[familiares] => Array(),[familiaresEliminados] => Array()

Resultados esperados: Se espera que sea eliminada la asociación entre las herramientas pasadas en el parámetro \$recursos y la persona que se esté adicionando o modificando.

3.5.1.8 Caso de prueba para el camino básico #11

Camino: ...-33-34-35-36-34-37-38-39

Descripción: El dato de entrada cumplirá con el siguiente requisito:

El parámetro \$recursos, no está vacío, contiene todos los datos de una persona determinada, así como las herramientas asociadas, excepto aquellas que se quitaron y los datos de los familiares que se van a registrar y/o eliminar del sistema.

Entrada: \$recursos = [[idtrabajador] => 9000008, [armacalibre] =>, [armacargador] =>,[armacartuchos] =>,[armamarca] =>,[armamodelo] =>, [armaserial] =>,[baston] => 0,[blazer] =>, [calzado] => 39, [camisa] => M, [chaleco] =>,[chaqueta] =>,[ci] => 15.910.482, [correo] => chicohyuday@hotmail.com, [cuerpoorigen] =>,[cursoorigen] =>,[cursopb] => N/P, [funcion] => MECANICO DE LA CPNB, [direccion] => el valle, urb. San antonio sector III, vereda 7, casa N° 12, [esposa] =>,[fechaingresoco] =>,[fechaingresopb] => 01/04/2011, [fechanacimiento] => 13/06/1980, [gradoinstruccion] => 6 GRADO, [idgruposanguineo] => 2, [idcargo] => 9000005, [idestadocivil] => 102, [idrango] => 1, [idtipoarma] =>,[llave] => 0, [linterna] => 0, [mono] =>,[nombres] => Luis Alberto, [observaciones] =>,[pantalon] => 36, [apellidos] => Garcia Blanco, [quepigorra] =>,[sexo] => 0, [short] =>,[telefonocasa] => 0212-8809482, [telefonomovil] => 0416-1935873, [tallacorrea] =>,[pantalonfalda] =>,[camisablusa] =>,[autoriza] =>,[herramientas] => Array(),[herramientasEliminadas] => Array(),[familiares] => Array([0] =>

Array([idherramienta] => 9000009, [codigo] => DTM-H-010, [denom] => DESTORNILLADOR, [marca] => DESC, [modelo] => PLANO, [medida] =>)),[familiaresEliminados] => Array())

Resultados esperados: Se espera que se realice la asociación entre las herramientas que fueron pasadas en el parámetro \$recursos y los datos de la persona que se estén adicionando o modificando.

Con la aplicación de los casos de prueba expuestos anteriormente se comprobó que el flujo de trabajo de las funcionalidades es correcto, ya que se probó que cada sentencia es ejecutada al menos una vez, cumpliéndose así las condiciones de la prueba.

3.5.2 Pruebas de caja negra

Para cada uno de los requisitos funcionales fueron definidos casos de prueba, los cuales son los siguientes:

Caso de prueba para el requisito Adicionar recurso humano

Condiciones de ejecución

- Se debe identificar y autenticar ante el sistema, además debe tener los permisos para ejecutar esta acción.
- Se debe seleccionar la opción del menú: **Mantenimiento/Recursos Humanos/ Registro de personas.**

Tabla 10 Descripción del caso de prueba para el requisito Adicionar recurso humano

| Nombre del requisito | Descripción general | Escenarios de pruebas | Flujo del escenario |
|------------------------------|--|---|--|
| 1: Adicionar recurso humano. | El sistema debe permitir adicionar recursos humanos. | EP 1.1: Adicionar recurso humano introduciendo datos válidos. | <ul style="list-style-type: none"> – Se introducen los datos del recurso humano correctamente. – Se presiona el botón Aceptar. – Se muestra un mensaje de información. |

| | | | |
|--|--|--|---|
| | | | <ul style="list-style-type: none"> - Se presiona el botón Aceptar. |
| | | <p>EP 1.2: Adicionar recurso humano introduciendo datos válidos presionando el botón Aplicar.</p> | <ul style="list-style-type: none"> - Se introducen los datos del recurso humano correctamente. - Se presiona el botón Aplicar. - Se muestra un mensaje de información. - Se presiona el botón Aceptar. |
| | | <p>EP 1.3: Adicionar recurso humano introduciendo datos inválidos.</p> | <ul style="list-style-type: none"> - Se introducen los datos inválidos del recurso humano. - Se presiona el botón Aceptar. - Se muestra un mensaje de error. |
| | | <p>EP 1.4: Adicionar recurso humano dejando campos vacíos.</p> | <ul style="list-style-type: none"> - Se introducen los datos dejando algún campo en blanco. - Se presiona el botón Aceptar. - Se muestra un mensaje informando del error. - Se presiona el botón Aceptar. |
| | | <p>EP 1.5: Cancelar.</p> | <ul style="list-style-type: none"> - Se introducen o no los datos del recurso |

| | | | |
|--|--|--|--|
| | | | humano. – Se presiona el botón Cancelar. |
|--|--|--|--|

Para más información sobre el caso de prueba para el requisito Adicionar recurso humano, consultar el documento entregable CG-SM-DR-227. Los casos de prueba elaborados para los restantes requisitos funcionales se encuentran en los documentos entregables: CG-SM-DR-230, CG-SM-DR-229, CG-SM-DR-231, CG-SM-DR-228, CG-SM-DR-225, CG-SM-DR-349, CG-SM-DR-351, CG-SM-DR-352, CG-SM-DR-350, CG-SM-DR-359, CG-SM-DR-362, CG-SM-DR-361, CG-SM-DR-363, CG-SM-DR-360, CG-SM-DR-226, CG-SM-DR-138, CG-SM-DR-144, CG-SM-DR-140, CG-SM-DR-141, CG-SM-DR-139.

La aplicación desarrollada fue probada por el centro Calisoft, donde fue comprobado el correcto funcionamiento de la misma a partir de los diseños de los casos de prueba. Como constancia de los satisfactorios resultados obtenidos, esta entidad emitió un acta de liberación de software (ver Anexo 5, Imágenes 30 y 31, y consultar documento entregable Acta de Liberación Mtto Vehicular-firmado.pdf).

Una vez liberada la aplicación fue presentada ante el cliente, el cual luego de haberla probado estuvo satisfecho con el producto entregado; prueba de esto lo constituyen los avales otorgados por el mismo y por el gerente general del proyecto por parte de Cuba, en los cuales consta que el sistema realizado cumple con las condiciones de calidad necesarias y satisface las necesidades plasmadas por los clientes, contribuyendo a mejorar la gestión de los procesos que se realizan en el área de Transporte del CPNB. Para más información acerca de los avales otorgados consultar Anexo 5, Imágenes 32 y 33.

3.6 Conclusiones parciales

Durante el desarrollo del capítulo fueron expuestos los artefactos generados del diseño del módulo para la gestión del proceso Administración de Recursos Humanos del área de Transporte del CPNB, así como las métricas que fueron empleadas para realizar su validación, las que ofrecieron como resultado que el diseño estaba realizado de forma simple y con una calidad aceptable, permitiendo con esto darle paso a la etapa de implementación. Además de lo anteriormente especificado, se presentaron las pruebas estructurales y funcionales que les fueron aplicadas, las que dieron como resultado que poseía un correcto funcionamiento, contaba con las condiciones de calidad necesarias y satisfacía las necesidades plasmadas por el cliente.

CONCLUSIONES

Tras haber culminado los tres capítulos con los que cuenta el presente trabajo de diploma se puede arribar a las siguientes conclusiones:

- Se logró la integración entre los procesos Administración de Recursos Humanos, Ejecución del mantenimiento y Administración de inspecciones técnicas, la cual no existía debido a la forma en que eran gestionados estos procesos en el área de Transporte del CPNB, permitiendo con esto, llevar un control sobre la cantidad de horas empleadas por un trabajador en la ejecución de una actividad de mantenimiento.
- El diseño realizado para la propuesta de solución fue correcto, siendo este validado a través de métricas que arrojaron resultados satisfactorios.
- El módulo Persona fue implementado correctamente y validado a través de pruebas de software, dándole solución a las necesidades plasmadas por los clientes.
- Con la utilización del sistema se facilitó el trabajo a los clientes, posibilitándoles ahorrar cuantiosos recursos materiales.
- Se contribuyó a mejorar la gestión de los recursos humanos en el área de Transporte del CPNB, lo cual se evidencia en el aval otorgado por los clientes tras haber probado y aceptado la aplicación presentada ante ellos.

RECOMENDACIONES

- Se recomienda que en el desarrollo de futuros sistemas para la gestión de mantenimiento sea utilizado el presente trabajo como referencia en cuanto a la gestión de los recursos humanos.
- Se recomienda que no sea publicada la información que es tratada en el presente trabajo en eventos internacionales pues se maneja información confidencial del Cuerpo de Policía Nacional Bolivariana.

BIBLIOGRAFÍA

- Alfonso, Inna Alfonso. 2011.** Desarrollo del componente para la configuración visual de las excepciones en el marco de trabajo Sauxe. Ciudad de la Habana: s.n., 2011.
- Bizagi. 2011.** [En línea] 2011. <http://bizagi.com/docs/BPMNbyExampleSPA.pdf>.
- Concepto de, Definición de.** [En línea] <http://conceptodefinicion.de/firefox/>.
- doctrine.** [En línea] <http://www.doctrine-project.org/docs/orm/2.1/en/reference/introduction.html>.
- Dpto de Informática de la Universidad de Valencia. 2004.** *Tutorial de Eclipse.* 2004.
- Equipo de Producción. Modelo de Desarrollo Orientado a componentes del proyecto ERP-CUBA.**
- exforsys.com. 2000-2011.** [En línea] 2000-2011. <http://www.exforsys.com/tutorials/uml/uml-introduction.html>.
- . **2000-2011.** Exforsys Inc. [En línea] 2000-2011. <http://www.exforsys.com/tutorials/uml/uml-industry-usage/1.html>.
- . **2000-2011.** Exforsys Inc. [En línea] 2000-2011. <http://www.exforsys.com/tutorials/uml/uml-introduction.html>.
- . **2000-2011.** Exforsys Inc. [En línea] 2000-2011. <http://www.exforsys.com/tutorials/php.html>.
- González, Rolando Alfredo Hernández León y Sayda Coello. 2011.** *El Proceso de Investigación Científica.* Ciudad de la Habana : Editorial Universitaria , 2011. ISBN 978-959-16-.
- Inc, Sencha. 2011.** Sencha. [En línea] 2011. <http://www.sencha.com/learn/extjs/?4x>.
- Ltda, EUG COM.** [En línea] www.eugcom.cl/fichas_tecnicas/Flotas.pdf.
- Máximo, Grupo. 2011.** [En línea] 2011. <http://www.transporex.net/informacion.html>.
- Pérez, Javier Eguíluz. 2008.** *Introducción a AJAX.* 2008.
- PostgreSQL-es. 2009-2011.** PostgreSQL-es. [En línea] 2009-2011. http://www.postgresql.org/es/sobre_postgresql.
- Raj, Martin Owen y Jog. 2003.** *BPMN and Business Process Management, Introduction to the New Business Process Modeling Standard.* 2003.
- S.A, INFT. 2011.** [En línea] 2011. <http://ggtt.com.ar/ggtt/>.
- Soft, Tinita. 2001-2010.** [En línea] 2001-2010. <http://www.tinitasoft.com/programavehi.htm>.
- The Apache Software Foundation. 2011.** Apache HTTP SERVER PROJECT. [En línea] 2011. <http://httpd.apache.org/>.
- ubunte es.** [En línea] http://doc.ubuntu-es.org/HTTPD_Servidor_web_Apache2.
- White, Stephen A.** *Introduction to BPMN.*
- Zend Technologies Ltd. 2006-2011.** Zend Framework. [En línea] 2006-2011. <http://framework.zend.com/manual/en/introduction.overview.html>.
- Gabinete de Recursos Humanos y Organización, Universidad de Granada.** Mapas de Procesos.
- Ivisate, Annerys Armenteros.** Análisis y diseño del Componente Ejecución del subsistema Mantenimiento de Cedrux. Ciudad de la Habana : s.n.
- Kawtar Benghazi, José Luis Garrido Bullejos, Manuel Noguera García, Departamento de Lenguajes y Sistemas Informáticos, Universidad de Granada.** Introducción al Modelado de Procesos de Negocio.
- Miguel Gea, Fco Luis Gutiérrez, José Luis Garrido, José J. Cañas.** Teorías y Modelos Conceptuales para un Diseño basado en Grupos.

LA INGENIERÍA DE REQUERIMIENTOS Y SU IMPORTANCIA EN EL DESARROLLO DE PROYECTOS DE SOFTWARE. **Arias Chaves, Michael.** 2005. 10 y 11, Costa Rica : Redalyc, 2005, Vol. VI. ISSN (Versión impresa): 1409-4746.

Pressman, Roger S. 2006. *Ingeniería del Software Un enfoque práctico.* 2006.

Sommerville. 2005. 2005.

Entorno Virtual de Aprendizaje. [En línea] [Citado el: 20 de 02 de 2012.]

http://eva.uci.cu/file.php/161/Documentos/Materiales_complementarios/UD_2_Ingenieria_de_Requisitos/5_Validacion_de_Requisitos/Manual_Validacion.pdf.

Escalona, María José y Koch, Nora. Entorno Virtual de Aprendizaje. [En línea] [Citado el: 15 de 2 de 2012.]

http://eva.uci.cu/file.php/161/Documentos/Materiales_complementarios/UD_2_Ingenieria_de_Requisitos/5_Validacion_de_Requisitos/Ingenieria_de_Requisitos_para_Aplicaciones_Web.pdf.

Sandoval Carvajal, Maria Marta y García Vargas, Maria Adilia. *LA TRAZABILIDAD EN EL PROCESO DE REQUERIMIENTOS DE SOFTWARE.*

[En línea] [Citado el: 15 de 03 de 2012.] <http://www.dsi.uclm.es/asignaturas/42530/pdf/M2tema12.pdf>.

Cabrera Pereira., Leisniel Ignacio y Hernández Gómez, Maylin. 2009. Análisis y Diseño del Módulo de Cobros y Pagos del sistema integral de gestión CEDRUX. Ciudad de la Habana : s.n., 2009.

DEPARTAMENTO DE ELECTRÓNICA, SISTEMAS E INFORMÁTICA COORDINACIÓN

Facultad de Ingeniería Electrónica y Telecomunicaciones. [En línea]

http://pis.unicauca.edu.co/moodle/file.php/291/Patron_Disenio_MVC.pdf.

Pressman, Roger S. 2005. *Ingeniería de Software, Un enfoque práctico.* 2005.

Rivera, Javier Fernandez. [En línea] [Citado el: 14 de 03 de 2012.] <http://aurea.es/wp-content/uploads/modelodedatos.pdf>.

Visconti, Marcelo y Astudillo, Hernán. Fundamento de Ingeniería de Software.

2011. Magma Soft. [En línea] 05 de 07 de 2011. [Citado el: 03 de 04 de 2012.]

http://www.magma.com.ni/~jorge/upoli_uml/refs/Introduccion_UML.pdf.

Universidad de Castilla La Mancha. 2011-2012. Departamento de Sistemas Informáticos. [En línea] 2011-2012. [Citado el: 03 de 04 de 2012.] <http://www.dsi.uclm.es/asignaturas/42530/pdf/M2tema12.pdf>.

CITMATEL. 2006. *SGestMan.* [multimedia] La Habana : CITMATEL, 2006. ISBN: 959-237-159-8.

Bakken, Stig Saether, y otros. 2001. *Manual de PHP.* 2001.

Buschman. 1996. *Pattern Oriented Software Architecture: A System of Patterns.* 1996. ISBN: 0471958697.

Centro de Informatización de la Gestión de Entidades. 2011. La Habana : s.n., 2011.

Fuentes, Lidia, Troya, José M. y Vallecillo, Antonio. 2009. *Lenguajes y Ciencias de la Computación. Lenguajes y Ciencias de la Computación.* [En línea] 2009. [Citado el: 10 de Octubre de 2011.]

<http://www.lcc.uma.es/~av/Docencia/Doctorado/tema1.pdf>.

Lovelle, Juan Manuel Cueva. 1999. *Introducción a UML Lenguaje para modelar objetos.* Oviedo : s.n., 1999.

Morera, René Hernandez. junio del 2011. Componente para la configuración visual de los servicios de integración en el marco de trabajo Sauxe. La Habana : s.n., junio del 2011.