

Universidad de las Ciencias Informáticas

Facultad 3



Título: "Desarrollo de mejoras funcionales de los módulos

Taller y Recursos humanos de Órbita."

Trabajo de Diploma para optar por el título de
Ingeniero Informático

Autor: Yadirka Verdecia Díaz

Tutor: Pedro Manuel Alás Verdecia

10 de Junio de 2014

DECLARACIÓN DE AUTORÍA

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

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

Yadirka Verdecia Díaz

Pedro Manuel Alás Verdecia

Firma del Autor

Firma del Tutor

DATOS DE CONTACTO

Nombre y apellidos del tutor: Pedro Manuel Alás Verdecía.

Institución: Universidad de las Ciencias Informáticas.

Título: Ingeniero en Ciencias Informáticas.

Correo electrónico: pmalas@uci.cu.

AGRADECIMIENTOS

Le agradezco a mi familia por todo el apoyo que me han brindado.

En especial a mi mamita que ha dado todo de sí para que hoy esté aquí, cumpliendo mi sueño. Eres lo más especial y sabes que sin tí esto no hubiese sido posible.

A pipo, por educarme y ser un padre para mí.

A mi tati, por brindarme soporte en los momentos más difíciles que nos tocó vivir en esta universidad. Porque a pesar de mi carácter siempre estuviste cuando más lo necesité.

A mi tutor por haberse comportado como un amigo y siempre exigirme más, formándome como una buena profesional.

A alí, que es la mejor amiga que he encontrado durante estos cinco años y no hay un minuto en que no piense en ella.

A mis amigos, a Elo, René y Rosa por toda su ayuda.

DEDICATORIA

Este trabajo va dedicado a quienes han hecho de mí una mejor persona. A mamá, a pipo y a mi papá. Siempre los llevo en el corazón y son fuente de inspiración para mí.

RESUMEN

El presente trabajo persigue desarrollar mejoras funcionales a los módulos Taller y Recursos humanos de Órbita para satisfacer las necesidades de integración y notificación de la Dirección de Transporte de la UCI en cuanto a la gestión del mantenimiento de los vehículos y de los recursos humanos.

Se realizó un estudio de varios sistemas de control de flota y mantenimiento lo que propició la reutilización de algunas funcionalidades del Sistema Control de Flota y Mantenimiento v1.0. Se realizó el modelado del negocio donde se enmarca la solución y se aplicaron técnicas para la captura y validación de requisitos del software. Se elaboró el modelo de diseño de software donde se modelaron los requisitos identificados en el análisis. Se hizo una descripción de la implementación del diseño así como las pruebas aplicadas para evaluarla.

Con el desarrollo de la solución se integró el componente Notificaciones al sistema Órbita para el envío de notificaciones automáticas. Además, fue posible la asimilación de la información existente en el Sistema de Capital Humano con los datos generales de los trabajadores del área de transporte de la UCI.

PALABRAS CLAVE

Desarrollo de software, flota, mantenimiento, notificaciones, recursos humanos, taller, vehículos.

CONTENIDO

AGRADECIMIENTOS	1
DEDICATORIA.....	2
RESUMEN	3
CONTENIDO.....	4
INTRODUCCIÓN	5
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	9
1.1. Introducción	9
1.2. Conceptos relacionados con los Sistemas de control de flota y mantenimiento	9
1.3. Sistemas de Control de flota y Mantenimiento	9
1.4. Modelo de desarrollo orientado a componentes (1.1)	18
1.5. Tecnologías y Herramientas.....	18
1.6. Conclusiones parciales.....	24
CAPÍTULO 2: MODELADO DEL NEGOCIO, ANÁLISIS Y DISEÑO	26
2.1. Introducción	26
2.2. Modelado del negocio	26
1.2. Requisitos.....	29
1.3. Diseño	37
1.4. Conclusiones parciales.....	44
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA.....	45
3.1. Introducción	45
3.2. Implementación	45
3.3. Aplicación de técnicas de validación de requisitos y utilización de métricas	55
3.4. Pruebas de software.....	60
3.5. Validación de las variables	69
3.6. Conclusiones parciales.....	71
CONCLUSIONES.....	72
RECOMENDACIONES	73
BIBLIOGRAFÍA	74
ANEXOS	¡ERROR! MARCADOR NO DEFINIDO.

INTRODUCCIÓN

El mantenimiento, por su incidencia significativa sobre la producción y la productividad de las empresas, constituye una de las vías para lograr y mantener mejoras en eficiencia y reducción de costos, mejorando así la competitividad de las empresas que lo implementan. (Prando, 2003)

En aras de favorecer el proceso de mantenimiento y el control de vehículos las Tecnologías de la Información y las Comunicaciones (TIC's) posibilitan a las empresas establecer herramientas que informaticen sus procesos. La UCI no elude este auge tecnológico y en aras de mejorar en este proceso, la rama de la informática ha tomado un papel decisivo para su desarrollo. El uso de los sistemas informáticos en el proceso de Perfeccionamiento Empresarial en Cuba tiene como objetivo fundamental elevar la productividad, la disciplina, la organización, incrementar la eficiencia y como consecuencia elevar el aporte de la empresa a la sociedad. (del Consejo de Estados y de Ministros, 16 de agosto del 2007) Los mismos deben contribuir al cumplimiento de los compromisos contractuales con oportunidad, calidad y ahorro de recursos energéticos y materiales.

La Universidad de las Ciencias Informáticas (UCI) cuenta con un área de transporte donde se aprovechan las ventajas que proporciona el mantenimiento. Se gestiona la información relacionada con los recursos humanos y la ejecución del mantenimiento de forma manual. Normalmente este proceso genera un gran cúmulo de datos que deben ser gestionados sobre formato duro provocando dificultades para la organización. Generalmente conlleva a errores que luego son difíciles de corregir y rectificarlos puede tomar un tiempo considerable. Resultan importantes todos los datos relacionados con la gestión de los recursos humanos, incluyendo no solo a los trabajadores que inciden directamente sobre la actividad de mantenimiento, sino también al resto del personal que trabaja en esta área. Es necesario controlar los datos generales de las personas, las herramientas que les son asignadas y de los conductores la licencia de conducción, los antecedentes, los chequeos médicos realizados y la escuela de recalificación. Además, debe conocerse cuando vence la escuela de recalificación y el chequeo médico de los choferes. En el taller en muchas ocasiones no se conoce con tiempo suficiente cuando debe realizarse un mantenimiento a los vehículos ni los involucrados en esta actividad. De los vehículos debe conocerse con antelación cuando vence los autorizos de parqueo así como somatones.

El departamento de soluciones empresariales del Centro de informatización de la gestión de entidades (CEIGE) de la UCI desarrolló el sistema Mantenimiento Vehicular para los Cuerpos de la Policía Nacional Bolivariana. Este sistema cuenta entre otros con los módulos de Taller y Recursos humanos que responden a la gestión del mantenimiento de los vehículos y de los recursos humanos. Al CEIGE contar con un sistema de mantenimiento identifica la posibilidad de que pueda utilizarse en el área de transporte de la universidad.

A inicios del año 2013 en una reunión entre el jefe de proyecto del Sistema de Mantenimiento Vehicular y el jefe del área de Transporte de la UCI se presentó a este último las funcionalidades del sistema creado. Como resultado de esta reunión se tomó como acuerdo el desarrollo de un sistema informático de gestión que reutilice un grupo de funciones del sistema ya creado e incluya nuevas funciones que respondan a los procesos del área de Transporte.

Después de revisar el módulo Recursos humanos y con vista a la creación de la versión 1.0 de Órbita se adaptaron algunas de las funciones existentes y se determinó que era necesario incluir en la gestión de los datos generales del personal las herramientas y la información de los conductores. Además se identificó que no se requiere del control de las prendas policiales, de los datos de los familiares y de la información relacionada con el ingreso y desempeño en los Cuerpos de la Policía Nacional Bolivariana. Órbita permitirá relacionar los recursos humanos a las diferentes actividades del taller, vehículos y recursos materiales.

Luego de varios talleres y de revisar en el módulo Taller las funciones que se adaptaron, se identificó que las funcionalidades implementadas en este módulo de Órbita 1.0 no satisfacen las necesidades de la Dirección de Transporte de la UCI en cuanto a la ocurrencia de eventos y su conocimiento por parte de los interesados, por lo que la versión 1.1 debe facilitar la notificación de los eventos significativos que ocurran en los procesos del taller.

A partir de las deficiencias planteadas y para dar solución a las necesidades existentes, se plantea el siguiente **problema a resolver**: Las funcionalidades implementadas en los módulos Taller y Recursos humanos del Sistema Control de Flota y Mantenimiento v 1.0 no satisfacen las necesidades de la Dirección de Transporte de la UCI en cuanto a la gestión del mantenimiento de los vehículos y de los recursos humanos.

Se define como **objeto de estudio**: Los sistemas de control de flota y mantenimiento.

Se plantea entonces como **objetivo general**: Desarrollar mejoras funcionales a los módulos Taller y Recursos humanos de Órbita para satisfacer las necesidades de la Dirección de

Transporte de la UCI en cuanto a la gestión del mantenimiento de los vehículos y de los recursos humanos.

El **campo de acción** se enmarca en: El desarrollo de software de sistemas informáticos del área de transporte que involucra los procesos en el Taller y Recursos humanos de la Universidad de las Ciencias Informáticas.

Para el cumplimiento del objetivo se determinan los siguientes **objetivos específicos**:

- Elaborar el marco teórico de la investigación para identificar el estado del arte en la realización de sistemas de control de flota y mantenimiento necesarios como sustento de la investigación.
- Realizar el análisis de las mejoras funcionales identificadas para los módulos Taller y Recursos humanos de Órbita de la Dirección de Transporte de la UCI para validar la propuesta de solución en función de la elicitación de las mejoras funcionales.
- Diseñar las mejoras funcionales identificadas para los módulos Taller y Recursos humanos de Órbita de la Dirección de Transporte de la UCI, para proveer de un plano de desarrollo e implementación efectiva de la solución.
- Realizar la implementación de las mejoras funcionales identificadas para los módulos Taller y Recursos humanos de Órbita de la Dirección de Transporte de la UCI, para la puesta a punto de las funcionalidades y su validación.
- Validar la propuesta de solución con pruebas internas para requisitos funcionales y no funcionales.

Definiéndose como **idea a defender**: Si se desarrollan las mejoras funcionales a los módulos Taller y Recursos humanos de Órbita será posible satisfacer las necesidades de integración y notificación de la Dirección de Transporte de la UCI en cuanto a la gestión del mantenimiento de los vehículos y de los recursos humanos.

En el desarrollo de la investigación se han utilizado los siguientes **métodos científicos**:

- **Histórico lógico**: Se utiliza para determinar el desarrollo tecnológico en Sistemas de Control de flota y Mantenimiento.
- **Análisis y Síntesis**: Este método se aplica para analizar el funcionamiento de diversos sistemas de control de flota y mantenimiento y determinar las características que pueden ser aprovechadas.

- **Inducción y Deducción:** A partir del estudio de diversos sistemas de control de flota y mantenimiento se arribaron a generalidades que luego fueron particularizadas a los procesos de gestión del mantenimiento de vehículos y los recursos humanos en el área de transporte de la UCI.
- **Modelación:** Se utiliza para representar por medio de diagramas el proceso de control de los recursos humanos, teniendo como resultado un mejor entendimiento de la posible solución a implementar. Se usa para la creación de los diagramas del diseño reflejando la lógica del módulo Recursos humanos.

Los **métodos empíricos** empleados son:

- **Entrevista:** Utilizada para obtener información de los proveedores de requisitos con el fin de hacer un levantamiento de requisitos y conocer las posibles características del sistema.
- **Observación:** Utilizado para prestar atención al desarrollo de las actividades cotidianas en el área de transporte. No perder de vista la manera en que se realizan las tareas y si ocurren los pasos específicos como se establecieron.
- **Medición:** Se utiliza con el empleo de las métricas Tamaño operacional de clase y Relaciones entre clases para validar el diseño del módulo Recursos humanos, así como las pruebas para verificar el correcto funcionamiento de la aplicación

El documento se encuentra estructurado en tres capítulos:

Capítulo 1: Fundamentación teórica. Hace alusión a la teoría de la investigación que constituye la base para su posterior desarrollo. Aborda un estudio de varios sistemas de control de flota y mantenimiento, así como una valoración acerca de su posible reutilización. Se definen las herramientas y tecnologías a utilizar.

Capítulo 2: Modelado del negocio, análisis y diseño. Está enfocado en el proceso de negocio donde se enmarca la solución y en consecuencia conocer los elementos que lo caracterizan. Se aplican técnicas para la captura y validación de los requisitos. Se presenta el diseño de la solución donde se modelan los requisitos identificados en el análisis.

Capítulo 3: Implementación y prueba. Se describe la implementación de la solución. Se explican las métricas utilizadas para evaluar el diseño. Se valida la solución aplicando pruebas al software.

Capítulo 1: *Fundamentación teórica*

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

En este capítulo se abordarán los temas fundamentales que sustentan la investigación. Se definen los conceptos relacionados con el dominio del problema. Se hace un estudio sobre los principales sistemas de control de flota y mantenimiento usados tanto en el extranjero como en el territorio nacional. En él se muestran las herramientas empleadas para el desarrollo de la solución, además de definir el modelo de desarrollo, lenguajes y marcos de trabajo a utilizar.

1.2. Conceptos relacionados con los Sistemas de control de flota y mantenimiento

Para beneficiar el entendimiento de los términos tratados en la investigación se definen los conceptos siguientes:

Mantenimiento: Se refiere a tener objetivos definidos y procedimientos para enfrentar las tareas de mantenimiento que indican cómo conseguir el mejor rendimiento de los equipamientos y recursos definidos para esta actividad. (C.W., 1992)

Notificación: Acción y efecto de comunicar formalmente a su destinatario la ocurrencia de un evento.

Sistemas de control de flota y mantenimiento: Software que facilita la administración y control de las flotas a cualquier nivel. Permite minimizar los riesgos asociados con la inversión en vehículos y mejorar su eficiencia y productividad, cumpliendo con la normativa legal.

Una vez definidos los conceptos fundamentales que son utilizados en los sistemas de control de flota y mantenimiento es necesario analizar sus principales funcionalidades, las cuales se exponen a continuación.

1.3. Sistemas de Control de flota y Mantenimiento

A nivel mundial hay diversos sistemas de control de flota y mantenimiento, dedicados a facilitar los procesos relacionados con los vehículos de una determinada entidad. Dentro de estos se encuentran:

- **Sistema de Flotas v. 1.2.209**

El **Sistema de Gestión de Flotas de Informática EUGCOM** ha sido diseñado para poder ejecutar todas las acciones que se generan en torno a una completa Administración y Gestión de Vehículos. (Eugcom, 2011) Está estructurado en:

Capítulo 1: *Fundamentación teórica*

Fichas: Contiene los datos generales relacionados con los vehículos y sus dueños, choferes, proveedores y clientes. (Eugcom, 2011)

Movimientos: Recoge la información de ingreso y mantención de combustibles, viajes y neumáticos. Genera automáticamente el pago de los seguros y las cuotas relacionadas al pago de los vehículos. (Eugcom, 2011)

Servicios: Permite programar, mantener y realizar movimientos de servicio. Controla las órdenes de trabajo. (Eugcom, 2011)

Informes: Permite la búsqueda de datos y genera un informe de Cargas de Combustible, Vehículos, Choferes, Neumáticos, Viajes Realizados y Seguro. (Eugcom, 2011)

Vencimientos: Permite visualizar todos los vencimientos generales y de vehículos. (Eugcom, 2011)

Herramientas del Sistema: Cuenta con el Reprocesamiento de Cargas de Combustible permitiendo crear un informe de errores generados por los usuarios en las cargas de combustible registradas en el sistema. (Eugcom, 2011)

Opciones del Sistema: Permite establecer y configurar los datos de la empresa que posee la licencia del software. Admite la mantención de los datos generales del Sistema de Gestión de Flotas, la configuración de los valores de moneda variables en el sistema y la personalización de los datos de los vehículos que administre la empresa y que no hayan sido incluidos en el Sistema de Gestión de Flotas. (Eugcom, 2011)

Fue desarrollado usando Visual Basic, con un diseño orientado al usuario. Es una aplicación de escritorio que necesita del sistema operativo Windows. Ha sido diseñado para ser instalado en un servidor de datos de una red interna, permitiendo que múltiples usuarios interactúen con el sistema. Sin embargo su reproducción total o parcial está penada por la ley. A pesar de esto permite enviar notificaciones al usuario.

- **Sistema Mantenimiento de Flota y Maquinaria GES-MAQ**

GES-MAQ está desarrollado bajo ciertos criterios de control de Flota y maquinarias relacionado al mantenimiento propiamente tal, como análisis de aceites, control de mantenciones preventivas, control de costos por equipo y control de inventario. (Alejandro Lagos, 2007)

GES-MAQ trabaja en entorno Windows utilizando la herramienta de Microsoft Access 2003. Cuenta con variada información, la que se estructura de la siguiente manera: Ingreso de Flota y

Capítulo 1: *Fundamentación teórica*

Maquinaria, Ingreso de Órdenes de Trabajo, Control de Análisis de Aceite, Control de Mantenciones a la Preventiva, Control de Petróleo (Rendimientos y Consumos) e Histórico de repuestos en Equipos.

Características:

Es posible controlar inventarios, obtener informes de ventas y costos en forma resumida. Se controlan los arriendos mediante el ingreso de arriendos a clientes externos y los aceites usados midiendo cambio a cambio el nivel de partículas que se encuentran en los componentes de cada máquina y camión. (Alejandro Lagos, 2007)

Cada equipo cuenta con información de mantenciones que es entregada por el proveedor del equipo, es necesario crear un manual con toda la información y de esta manera entregar herramientas a los mecánicos para el correcto mantenimiento de los equipos a cargo. (Alejandro Lagos, 2007)

Una importante fuente de información es el control de horas de trabajo de las máquinas y camiones, es posible controlar las horas y obtener un informe sobre producción por obras y mandantes. (Alejandro Lagos, 2007)

No obstante, no permite enviar notificaciones al usuario.

- **Sistema Control de Flota y Mantenimiento V 1.0**

Sistema que permite manejar toda la información del área de transporte del Cuerpo de Policía Nacional Bolivariana vinculada al control y mantenimiento de su flota de vehículos. Permitirá además gestionar el mantenimiento preventivo planificado y correctivo no programado (Gómez Pérez, y otros, Septiembre 2012). Entre sus principales módulos se encuentran:

Clasificadores: A través de este módulo se definen los documentos técnicos, tipos de mantenimientos, tipos de unidad, herramientas, accesorios, causa de fallas, repuestos y unidades de medidas, todas ellas son un conjunto de información a utilizar por los demás módulos. (Gómez Pérez, y otros, Septiembre 2012)

Configuración: En este módulo se definen las dependencias que recibirán un servicio de mantenimiento por parte de transporte. La agrupación de vehículos por tipo de unidad, marca y modelo, el régimen de mantenimiento (fecha o lectura), los tipos de mantenimiento asociados al grupo, la frecuencia, las actividades a realizar por cada uno de los tipos de mantenimiento definidos, así como los tipos de fallas. (Gómez Pérez, y otros, Septiembre 2012)

Capítulo 1: *Fundamentación teórica*

Vehículos: A través de este módulo se gestiona la información de los vehículos a los cuales se le realizarán acciones de mantenimiento, se inicializan los valores del medidor de cada vehículo, se actualiza la fecha de último servicio o mantenimiento, se controlan los vehículos que están asegurados y los accesorios asignados. A partir de este módulo se asignan los vehículos a las dependencias. Se lleva un control de la cantidad de kilómetros recorridos por vehículo. Se gestiona la recepción de vehículos al área de transporte. Cada uno de estos vehículos estará asociado a un grupo que contiene las características generales de estos documentos técnicos, repuestos necesarios para realizar las actividades de mantenimiento, así como las relaciones de accidentes e inspecciones técnicas realizadas en un periodo. (Gómez Pérez, y otros, Septiembre 2012)

Taller: Este módulo maneja toda la información concerniente a la generación de órdenes de trabajo de los vehículos que están próximos a recibir mantenimiento según el umbral de mantenimiento definido en el sistema. Se gestionan las herramientas de trabajo, repuestos y recursos humanos que intervienen en la ejecución del mantenimiento controlando los costos y recursos utilizados. (Gómez Pérez, y otros, Septiembre 2012)

Recursos humanos: En este módulo se gestiona la información referente a todo el personal que labora en el área de transporte logrando realizar un expediente de cada trabajador. (Gómez Pérez, y otros, Septiembre 2012)

Documentos: Módulo de apoyo para la gestión de memorándum e informes registrados por los funcionarios del área de transporte. (Gómez Pérez, y otros, Septiembre 2012)

Cuenta además con los subsistemas complementarios:

Estructura y composición: A través de este módulo se gestiona la organización estructural de las dependencias pertenecientes al Cuerpo de Policía Nacional Bolivariana. (Gómez Pérez, y otros, Septiembre 2012)

Seguridad: Este módulo gestiona la seguridad del sistema mediante la definición de roles y usuarios los cuales presentan o no permisos a las diferentes funcionalidades del sistema. (Gómez Pérez, y otros, Septiembre 2012)

Entre las ventajas que brinda es que es una aplicación web, multiplataforma y fue desarrollado siguiendo las políticas de independencia tecnológica del CEIGE.

Capítulo 1: *Fundamentación teórica*

A pesar de que gestiona todo el personal que labora en el área de transporte no incluye los datos de la escuela de conducción de los choferes, las categorías de la licencia, los antecedentes, los chequeos médicos y la escuela de recalificación de los mismos. Constituyendo esta información sensible para el área de Transporte de la UCI. Además, no permite el envío de notificaciones.

- **Alamo**

Software para Transporte, Logística y Mantenimiento de Flotas simple y fácil de usar (Digital Express, 2013). Cuenta con los Módulos:

Cargas: Trazabilidad de cargas para Expreso y Paquetero. Permite realizar de manera muy simple la trazabilidad completa de la carga almacenando los distintos estados de la misma con la correspondiente documentación desde el retiro en origen hasta su rendición al remitente. (Digital Express, 2013)

Viajes: Control de viajes completos y transportes. Permite el seguimiento de los viajes completos. Es la única solución del mercado que permite administrar múltiples tramos para un mismo viaje, tanto con cargas, como vacíos o de diferentes clientes. (Digital Express, 2013)

Logística: Control de inventario y almacenes. Permite administrar las reservas de sus clientes en su depósito, posibilitando la rápida ubicación, seguimiento, el inventario, la distribución y facturación de la mercadería. (Digital Express, 2013)

Flotas: Mantenimiento de Flotas y Taller. Permite documentar «la Historia clínica » de las unidades de manera integrada y automática. Abarca desde el taller y el control de pañol hasta el seguimiento de cubiertas. (Digital Express, 2013)

Gestión: Control de cajas, Bancos y Cuentas Corrientes de Proveedores. Es un complemento que permite alcanzar el control de la operación integrando las soluciones operativas con Cuentas Corrientes de Proveedores, Cajas y Bancos. (Digital Express, 2013)

El sistema le recordará al usuario el momento en que debe hacer el mantenimiento por medio de alertas. Para adquirir el producto es necesario comprar alguno de los tres tipos de paquetes que ofrece. Estos son Plata, Oro y Platino a elección del cliente en dependencia de la infraestructura de la organización ofreciendo diferentes niveles de servicio desde el más básico hasta la versión completa con el control en todas las áreas de la Empresa. Además, es una aplicación web que opera sobre el sistema operativo Windows.

Capítulo 1: *Fundamentación teórica*

- **Software SGestMan**

SGestMan es una Tecnología Integral acompañada de un sistema informático para la organización y control de la actividad de mantenimiento preventivo planificado, diseñado para reducir costos de mantenimiento, a partir de una correcta estrategia de Mantenimiento Preventivo Planificado. Cuenta con 11 módulos de trabajo, los cuales recogen y procesan toda la información que se realiza en la empresa, por concepto de Mantenimiento. (Desoft, 2013)

Patrimonio: Módulo que está previsto para estructurar toda la información de los diferentes elementos que forman parte de cualquier empresa, sea de producción o de servicios, lo cual garantiza poder tener muy bien definidas todas las necesidades que en materia de mantenimiento precisa una empresa para contar con un Patrimonio con un alto nivel de disponibilidad y utilización técnica. (Desoft, 2013)

Recursos Humanos: Módulo que permite organizar toda la información de los empleados que forman parte de la estructura de mantenimiento o de producción del negocio y con los cuales se garantiza la disponibilidad de los objetos que conforman el Patrimonio de una empresa. (Desoft, 2013)

Solicitudes de Órdenes de Servicios: Módulo diseñado para realizar el proceso de solicitud de las necesidades de mantenimiento de los clientes internos o externos. Permite visualizar todo el seguimiento de atención de las solicitudes realizadas al área de mantenimiento. Garantiza realizar las solicitudes a ejecutores internos o externos. (Desoft, 2013)

Órdenes de Servicios: Módulo diseñado para llevar todo el control técnico - económico de las órdenes de servicio que se realizan por los diferentes ejecutores de mantenimiento tanto internos como externos, de manera que se pueda tener todos los históricos de los objetos del Patrimonio con un alto nivel de detalle. (Desoft, 2013)

Contratos: Módulo dirigido a llevar toda la información técnica y económica de las contrataciones que hace el departamento de mantenimiento para garantizar la disponibilidad de los objetos del Patrimonio. De esta forma podrán ser incluidos todos aquellos prestadores de servicios que brindan mantenimientos y reparaciones a los equipamientos de las empresas. (Desoft, 2013)

Mantenimiento Preventivo: Módulo concebido para preparar toda la estrategia de proyección, programación y planificación de acciones de mantenimiento preventivo, dirigido a garantizar el

Capítulo 1: *Fundamentación teórica*

óptimo desempeño del equipamiento, la máxima disponibilidad y la reducción de costos por concepto de reparaciones y mantenimientos no previstos. (Desoft, 2013)

Producción: Módulo que permite registrar la información necesaria que el mantenimiento precisa de la actividad productiva para su propia gestión. (Desoft, 2013)

Informativo: Reportes de salidas con la información manejada en los módulos del sistema. Análisis y evaluación de indicadores de mantenimiento a partir del uso de herramientas estadísticas y utilización de salidas gráficas. Utilización de índices de clase mundial, diagramas de Pareto, gráficos de tendencia anual móvil, personalización de indicadores a solicitud del cliente. (Desoft, 2013)

Administración y Seguridad: Módulo previsto para llevar todo el control de los accesos a los usuarios del sistema, así como los permisos a las opciones de cada uno de los módulos, lo cual garantiza la integridad y fiabilidad de la información que se incluye en el resto del sistema. Usuarios: Lista de usuarios que tendrán acceso a los módulos del sistema. Accesos: Control de entradas a las diferentes opciones de los módulos. Registro de eventos: Monitoreo de los diferentes procesos que han realizado los usuarios dentro del sistema. Salvas de seguridad: Salvas automáticas y preestablecidas por el usuario administrador del sistema para garantizar copias de respaldo de la información. (Desoft, 2013)

Los requerimientos técnicos que necesita son la Licencia de SQL Server Enterprise y el sistema operativo Windows.

- **Gestión de vehículos y combustibles gCAR**

El sistema **gCar** está destinado al control de vehículos y equipos tecnológicos, en cualquier tipo de empresa u organización. Ofrece un conjunto de funcionalidades y reportes orientados a mejorar la gestión del parque automotriz, los combustibles y lubricantes y otros afines. Brinda la información necesaria a directivos y especialistas. (DATYS, 2011) Permite:

Gestionar los vehículos y equipos tecnológicos, e incluye el control histórico de los cambios realizados en el vehículo. En lo relativo a partes y piezas, el tipo de labor en la que es empleado, el tipo de combustible y lubricantes, entre otras. (DATYS, 2011)

Asegurar el registro y control de las licencias de circulación, licencias operativas, revisiones técnicas, mantenimientos y los gastos de cualquier tipo que defina el usuario. (DATYS, 2011)

Capítulo 1: Fundamentación teórica

Gestionar los combustibles y lubricantes, tanto por vehículos como por equipos tecnológicos. (DATYS, 2011)

Controlar los agregados y accesorios asignados al medio de transporte. (DATYS, 2011)

Realizar las operaciones contables, permitiendo la generación de los comprobantes para el sistema de Contabilidad que se lleve en la organización. (DATYS, 2011)

Obtener reportes y gráficos sobre solicitudes realizadas por el usuario. (DATYS, 2011)

Desarrollado con tecnología Genexus. Necesita del servidor Windows Server 2003 y SQL Server 2005 que son privativos. Sin embargo posee una interfaz web sencilla.

Después de enunciadas las principales funcionalidades de los sistema de control de flota y mantenimiento estudiados se puede establecer una comparación entre ellos que permita determinar sus características positivas. A continuación se manifiestan los resultados obtenidos.

1.3.1. Análisis valorativo

Con el fin de conocer si el uso de algunos de los sistemas aludidos es conveniente se determinan los indicadores: tipo de aplicación (web o escritorio), licencia de tecnologías sobre la que están desarrollados (requerimientos tanto para su implementación como despliegue), la gestión de los recursos humanos (información del personal) y taller (envío de notificaciones).

Tabla 1 Tabla comparativa entre los sistemas.

Sistema	Sistema de Flotas	GES-MAQ	Mantenimiento vehicular v1.0	Alamo	SGestMan	gCar
Indicador						
Licencia	✓	✓	–	✓	✓	✓
Gestión de los recursos humanos	✓	–	✓	✓	✓	–
Gestión del taller	✓	–	–	✓	–	–
Tipo de aplicación	Escritorio	Escritorio	Web	Escritorio	Escritorio	Web

A partir de la información mostrada en la tabla comparativa, el análisis arrojó como resultado:

- Exceptuando al sistema de mantenimiento vehicular v1.0 los sistemas estudiados no pueden utilizarse pues necesitan licencias para las tecnologías usadas en su desarrollo, ya que la UCI tiene como política el uso de tecnologías libres.

Capítulo 1: *Fundamentación teórica*

- A excepción de gCar y GES-MAQ los sistemas analizados gestionan los recursos humanos de sus entidades, sin embargo no se adecuan a las características de la UCI pues se debe manejar la información relacionada con los trabajadores de oficina y los choferes.
- El sistema de mantenimiento vehicular v1.0 gestiona todo el personal que labora en el área de transporte pero no se incluyen los datos de la escuela de conducción de los choferes, las categorías de la licencia, los antecedentes, los chequeos médicos y la escuela de recalificación de los mismos. Constituyendo esta información sensible para el área de Transporte de la UCI.
- Los sistemas analizados proporcionan una gestión adecuada del taller y la generación de reportes. Sistema de Flotas y Alamo permiten enviar notificaciones tanto a choferes como a otros usuarios relacionados a los vehículos. Esta característica reafirma la necesidad de incluir notificaciones en los sistemas de control de flota.
- Exceptuando al sistema de mantenimiento vehicular v1.0 y gCar el resto de 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 Windows y de sus capacidades; requieren de una instalación y actualización personalizada.

Partiendo de los talleres en los que intervino la dirección de transporte y el equipo de desarrollo, así como de la comparación de los sistemas estudiados se decide que se realicen mejoras funcionales al sistema de mantenimiento vehicular v1.0 para el CPNB. Las similitudes que serán aprovechadas son: una aplicación web, uso de tecnologías libres y la gestión de los recursos humanos así como el envío de notificaciones. De este último, se reutilizará parte de él y se agregarán nuevas funcionalidades.

Teniendo en cuenta que el sistema de mantenimiento vehicular v1.0 para el CPNB no permite el envío de notificaciones se identifica la necesidad de añadirlas. El CEIGE cuenta con un componente que permite la gestión de notificaciones y se originan por una acción específica del usuario en el sistema. Fue implementado sobre las mismas tecnologías que el sistema seleccionado, por lo que decide integrarlo a Órbita. Sin embargo las alertas deben ocurrir sistemáticamente sin la necesidad de que un usuario la desencadene, por lo que se determina

Capítulo 1: *Fundamentación teórica*

que se generen automáticamente.

Una vez establecido las mejoras funcionales que serán incorporadas al sistema de mantenimiento vehicular v1.0 se define en el siguiente epígrafe el modelo de desarrollo que guiará la propuesta de solución.

1.4. Modelo de desarrollo orientado a componentes (1.1)

Tomando en consideración el modelo de desarrollo utilizado en el Sistema Control de Flota y Mantenimiento v 1.0 para los Cuerpos de Policía Nacional Bolivariana se decide utilizar el definido por el CEIGE. Consiste en un modelo estandarizado, que establece las distintas fases por las que se debe transitar y el conjunto de artefactos a generar en cada una de ellas.

Este modelo de desarrollo permitirá generar los artefactos modelo de procesos de negocio, descripción de proceso de negocio, modelo conceptual, requisitos funcionales, diagrama de clases del diseño, descripción de las clases del diseño, diagramas de secuencia, modelo de datos, diagrama de componentes, diagrama de despliegue, prototipo de interfaz de usuario, especificación de requisitos y diseños de casos de prueba.

El próximo tema hace referencia a la base tecnológica que será utilizada durante el desarrollo de las mejoras funcionales a los módulos Taller y Recursos humanos.

1.5. Tecnologías y Herramientas

Por la experiencia adquirida en el desarrollo del Sistema Control de Flota y Mantenimiento v1.0 para los Cuerpos de la Policía Nacional Bolivariana se realizó una revisión de las tecnologías y herramientas usadas. Estas permitirán el desarrollo del sistema bajo los principios de independencia tecnológica (Software Libre).

A continuación se hace referencia al marco de trabajo que se empleará así como las tecnologías y herramientas que requiere su utilización.

1.5.1. Marco de Trabajo Sauxe

El desarrollo de las mejoras funcionales de los módulos Taller y Recursos humanos del Sistema de Gestión de Mantenimiento Vehicular utilizará Sauxe como marco de trabajo, desarrollado por el Departamento de Tecnología de CEIGE. Posee las características tecnológicas para la implementación de aplicaciones web de gestión. Cuenta con una arquitectura que permite reutilizar código y promover buenas prácticas como el uso de patrones. En la capa de presentación utiliza Extjs 3.4. En la misma se maneja todo el flujo web permitiendo al cliente

Capítulo 1: *Fundamentación teórica*

visualizar la información. En la capa de negocios utiliza Zend y para el acceso a los datos usa PHP¹ Doctrine.

Sauxe está compuesto por los marcos de trabajo Extjs, Zend, Doctrine y Acaxia, de los que seguidamente serán abordadas sus características principales y donde fueron empleados.

1.5.1.1. Marcos de trabajo que componen Sauxe

EXTJS 3.4

Marco de trabajo que soporta JavaScript² para el desarrollo de aplicaciones Web. Actúa del lado del cliente y se utiliza para la creación de las interfaces de usuario del módulo Recursos humanos y el manejo de sus eventos. Crea todos los objetos HTML (Lenguaje de marcas hipertextuales) a través de la manipulación del Modelo de Objetos del Documento (DOM). Provee un balance entre Cliente-Servidor, ya que distribuye la carga, permitiendo que el servidor gestione más clientes al mismo tiempo. Permite obtener información del servidor sin estar sujeto a la acción de un usuario (Baryolo, 2010)

Zend 1.12

Brinda soluciones para construir aplicaciones web y servicios modernos, robustos y seguros. (Zend Technologies, 2006-2013)Será utilizado dado que se encuentra bajo la licencia Distribución de Software Berkeley (BSD), lo cual permite su distribución, así como las aplicaciones que se desarrollen con él. Proporciona los componentes que forman la estructura del patrón Modelo-Vista-Controlador (MVC). Permite convertir estructuras de datos PHP a la Notación de Objetos de JavaScript (JSON³) y viceversa, para su utilización en aplicaciones AJAX.

Doctrine 1.2.2

Doctrine es un marco de trabajo para el mapeo objeto relacional que está dividido en dos capas principales, la capa de abstracción de base de datos (DBAL) y el mapeador de objetos relacionales (ORM) para PHP 5.2. (phpdoctrine.org)

¹ Preprocesador de hipertextos por sus siglas en inglés.

²Lenguaje de programación orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

³Acrónimo de "JavaScript Object Notation", es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML. La simplicidad de JSON ha dado lugar a la generalización de su uso, especialmente como alternativa a XML en AJAX. Una de las supuestas ventajas de JSON sobre XML como formato de intercambio de datos en este contexto es que es mucho más sencillo escribir un analizador semántico de JSON.

Capítulo 1: *Fundamentación teórica*

Se empleará puesto que Sauxe utiliza en la capa de acceso a datos el Lenguaje de Consulta de Datos (DQL) que implementa Doctrine. Una de sus principales características es la opción de escribir las consultas de base de datos en un objeto con una propiedad orientada al lenguaje SQL, llamada DQL. Tiene la posibilidad de convertir clases a tablas de una base de datos y viceversa, lo que es utilizado para realizar las consultas a la base de datos.

Acaxia

Acaxia es el Sistema de Gestión Integral de Seguridad que se empleará para garantizar la seguridad y tiene como objetivo la gestión dinámica de la autorización a los recursos. Fue desarrollado sobre tecnologías libres como el lenguaje PHP, el gestor de base de datos PostgreSQL y el servidor web Apache. Asigna los permisos a los roles creados en el sistema por lo que cada usuario solo tomará acciones sobre la información que tenga permitida, garantizándose siempre su compartimentación. Permite la administración dinámica de perfiles de usuario y consultar todas las acciones realizadas en el sistema, con toda la información asociada, dígame tiempo, valores e interacción.

1.5.2. Lenguaje y notación de modelado

Notación de modelado de procesos de negocio: BPMN

BPMN⁴ es una notación basada en diagramas de flujo para definir la lógica de los procesos de negocio. Se utiliza para el modelado del proceso de negocio control de los recursos humanos permitiendo la representación de la secuencia de sus actividades y los artefactos que se generan. Se obtiene un Diagrama de Procesos de Negocio que posibilita el entendimiento entre el cliente y el equipo de desarrollo.

Lenguaje Unificado de Modelado: UML 2.0

UML⁵ se utilizará en diferentes etapas del ciclo de vida del desarrollo de los módulos Taller y Recursos humanos. Tecnología orientada a objetos que utiliza un conjunto de símbolos para representar gráficamente los diversos componentes del sistema y soporta extensiones personalizadas a través de elementos estereotipados. Soporta un conjunto rico en elementos de notaciones gráficas permitiendo modelar, construir y documentar los elementos que forman un sistema orientado a objetos. (Object Management Group, 2001)

⁴Notación para el Modelado de Procesos de Negocio del inglés Business Process Modeling Notation

⁵Lenguaje Unificado de Modelado del inglés Unified Modeling Language.

Capítulo 1: *Fundamentación teórica*

1.5.3. Visual Paradigm 8.0

Herramienta de Ingeniería de Software Asistida por Computadora (CASE) que: *“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, Vol. 1. 2005)

Soporta UML y BPMN, definidos para el modelado de la solución, lo que facilita la comunicación para todo el equipo de desarrollo. Se utilizó para la generación de los diagramas del modelado del negocio, la administración de requisitos y el diseño.

1.5.4. Lenguajes de programación

PHP 5.3.2

Definido como: *“lenguaje interpretado, diseñado para la creación de páginas web dinámicas con acceso a información almacenada en bases de datos. La naturaleza de código abierto (open-source) de PHP lo muestra como una alternativa de fácil acceso para todos. PHP contiene funciones de seguridad que previenen la inserción de órdenes dentro de una solicitud de datos, posee manejo de excepciones, posibilita la generación de archivos PDF⁶, flash, así como imágenes en diferentes formatos.”* (The PHP Group, 2013)

Se emplea durante el desarrollo de la solución puesto que es el lenguaje de programación del lado del servidor que utiliza Sauxe. Admite la creación de aplicaciones para servidores e independientes del navegador. Presenta gran capacidad de conexión con la mayoría de los sistemas gestores de base de datos que se utilizan en la actualidad como PostgreSQL.

JavaScript

JavaScript es un lenguaje interpretado en el cliente por el navegador al momento de cargarse la página, es multiplataforma, orientado a eventos con manejo de objetos, cuyo código se incluye directamente en el mismo documento HTML. (FLANAGAN, 2002) Se utilizará como lenguaje de programación del lado del cliente ya que proporciona los medios para controlar las ventanas del navegador y el contenido que muestran, evita depender del servidor Web para capturar los

⁶ Formato de Documento Portable por sus sigla en inglés.

Capítulo 1: *Fundamentación teórica*

eventos generados por el usuario y responder a ellos. Además, se empleará para comprobar los datos que el usuario introduce en un formulario antes de enviarlos.

1.5.5. AJAX

Se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes. (Garret, 2005) La componen: XHTML⁷, CSS⁸, DOM, XML, XSLT⁹, JSON, XMLHttpRequest¹⁰ y JavaScript.

En la implementación del módulo Recursos humanos se utilizará para la comunicación entre la capa de la vista y la controladora ya que mejora la interacción del usuario con la aplicación. Esto es posible puesto que evita las recargas constantes de la página, realizando el intercambio con el servidor en un segundo plano. Sustituye 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.

1.5.6. Bibliotecas de clases

TCPDF (2.1)

Es una clase php para generar documentos pdf. TCPDF es ahora uno de los proyectos de código abierto más activos del mundo. (Nicola Asuni - Tecnick.com LTD) Se utilizará para la generación de los reportes del módulo Recursos humanos ya que no requiere bibliotecas externas para su funcionamiento básico. Contiene métodos para publicar códigos XHTML + CSS, Javascript y formularios. Permite la administración automática de encabezados y título al pie de la hoja.

Strophejs

Es una colección de bibliotecas que se empleará para comunicarse con el protocolo XMPP. Es una biblioteca JavaScript que se basa en flujos bidireccionales sobre HTTP sincrónico (BOSH por sus siglas en inglés) para emular la persistencia con estado de conexión en ambos sentidos a un servidor XMPP. (Moffitt, 2010)

⁷ Lenguaje Extensible de Marcado de Hipertexto del inglés Extensible Hypertext Markup Language.

⁸ Hojas de estilo en cascada del inglés Cascading Style Sheets.

⁹ Lenguaje de Hojas Extensibles de Transformación del inglés Extensible Stylesheet Language.

¹⁰ Lenguaje de Marcas Extensible/Protocolo de Transferencia de Hipertextos del inglés Extensible Markup Language/ Hypertext Transfer Protocol.

Capítulo 1: *Fundamentación teórica*

XMPPPHP

XMPPPHP es una biblioteca que toma ventaja de PHP5 y proporciona una solución elegante con un enfoque directo. Permite conectarse a cualquier servidor XMPP 1.0 (Google Talk, Talk LJ, jabber.org). Posee soporte para la Seguridad en la Capa de Transporte (TLS). (Werdmuller, 2010) Se utilizará para la conexión con el servidor Openfire.

1.5.7. Servidores

Servidor de mensajería instantánea Openfire v3.7.1

Se empleará Openfire como servidor de mensajería para establecer conexiones servidor a servidor que permitan compartir sus usuarios. Utiliza el protocolo Jabber, permite añadir complementos y cuenta con una interfaz web de administración. (REALTIME, Ignite., 2008)

Servidor de aplicaciones Apache 2.2

El servidor Apache HTTP se utilizará para publicar la aplicación y que los usuarios puedan acceder a ella. Se hará uso para la creación de un huésped virtual donde sólo queden públicos los ficheros de presentación (controlador frontal, js, css, imágenes). Soporta el lenguaje PHP que será utilizado para la implementación. Tiene un alto grado de configuración, lo que se pondrá de manifiesto en la activación del módulo mod_proxy. Posee soporte para la Capa de Conexión Segura (SSL) y la Seguridad en la Capa de Transporte que se utilizarán en el envío de correos electrónicos.

Servidor de base de datos PostgreSQL 9.1

PostgreSQL es un potente sistema de gestión de bases de datos objeto-relacional (O-RDBMS), multiusuario, centralizado y de propósito general. Es compatible con los principales Sistemas Operativos: Linux, Unix, Mac OS y Windows. (Lopez, 2013)

PostgreSQL se utilizará para el almacenamiento de los datos del sistema Órbita. Funciona correctamente con grandes cantidades de datos y una alta concurrencia de usuarios accediendo al mismo tiempo. Soporta funciones con privilegios por usuario y realiza copias de seguridad.

1.1.1. Protocolos

Protocolo XMPP/Jabber.

Capítulo 1: *Fundamentación teórica*

El componente Notificaciones tiene definida como técnica de notificación en tiempo real el protocolo XMPP/Jabber. Está basado en un conjunto de tecnología XML para aplicaciones en tiempo real utilizando ficheros en formato XML para hacer las transferencias de mensajes entre sus entidades cliente-servidor o servidor-servidor.

Protocolo Simple de Acceso a Objetos (SOAP).

SOAP es un protocolo basado en XML para acceder a servicios web que permite la comunicación entre aplicaciones. El intercambio de mensajes se produce usando cualquier protocolo que pueda transportar texto y de forma transparente al usuario. (W3Schools, 1999-2014)

Posibilitará hacer uso del servicio web notificacionesMtto que va a brindar Órbita para el envío de notificaciones. En el módulo Recursos humanos se utilizará para consumir de la UDDI¹¹ los servicios web de identificación y del Sistema de Capital Humano.

1.1.2. Cron

En prácticamente todas las distribuciones de Linux se usa el demonio¹² cron (servicio). Permite a usuarios Linux/Unix ejecutar automáticamente comandos o scripts (grupos de comandos) a una hora o fecha específica. Aunque es usado normalmente para comandos de tareas administrativas como respaldos también puede ser usado para ejecutar cualquier cosa. (ADELSTEIN, y otros, 2007)

Solo requiere ser iniciado una vez, generalmente con el mismo arranque del sistema. En la mayoría de las distribuciones el servicio se instala automáticamente. Cron permite controlar qué usuarios pueden o no usar sus servicios. Se utiliza para la ejecución de un archivo escrito en PHP que hace posible la ejecución automática de las notificaciones.

1.2. Conclusiones parciales

Este capítulo permitió analizar algunos software relacionados con los sistemas de control de flota y mantenimiento teniendo en cuenta indicadores que facilitaron la comprensión del estudio y como resultado el desarrollo de mejoras funcionales a los módulos Taller y Recursos humanos de Órbita en función de satisfacer necesidades de la Dirección de Transporte de la

¹¹ De sus siglas en inglés Universal Description, Discovery and Integration.

¹² Un demonio es un tipo especial de proceso informático no interactivo, es decir, que se ejecuta en segundo plano en vez de ser controlado directamente por el usuario.

Capítulo 1: *Fundamentación teórica*

UCI. Se definieron las herramientas, tecnologías y modelo de desarrollo que serán aplicados durante todo el desarrollo del software además de los artefactos que serán obtenidos.

Capítulo 2: Modelado del negocio, análisis y diseño

CAPÍTULO 2: MODELADO DEL NEGOCIO, ANÁLISIS Y DISEÑO

2.1. Introducción

En el siguiente capítulo se describe el proceso control de recursos humanos del área de Transporte de la UCI. Se elabora un mapa de procesos en el cual son expuestos los procesos claves y artefactos relacionados con los mismos. Se identifican y registran en el Modelo Conceptual los conceptos fundamentales que se manejan en el área de Transporte en cuanto a la organización de recursos humanos. Se identifican, especifican y validan tanto los requisitos funcionales como los no funcionales. Se incluyen los diagramas de clases de diseño con los mecanismos y patrones utilizados. La descripción de las clases, los diagramas de secuencia y el modelo de datos. Se explican las métricas usadas para evaluar el diseño.

2.2. Modelado del negocio

El modelado de negocio consiste en representar uno o más elementos de una empresa, tales como su funcionalidad y su lógica de negocios. Esta actividad permite asegurar que los clientes, usuarios finales, desarrolladores y otros involucrados tengan una visión común de la organización.

2.2.1. Mapa de procesos

Para la elaboración del mapa de procesos, como parte del modelado del negocio, fue empleada una plantilla conformada por el CEIGE. Se representan los procesos identificados en el área de Transporte de la UCI agrupados por niveles, colocando en un nivel 0 los procesos claves¹³ de esta organización y en un nivel 1 sus procesos derivados. Dentro de los procesos del nivel 0 se encuentra Disponibilidad del parque automotor, en el documento CIG-CFM-N-MTTO-i1101.xls se especifica información más detalla sobre él.

El contenido de este trabajo se enmarca en el proceso Control de recursos humanos perteneciente al nivel 1 del proceso Disponibilidad del parque automotor. Para su descripción se crea un documento que contiene: nombre del proceso, breve descripción, referencia, nivel y proceso padre al que pertenece (consultar documento entregable CIG-CFM-N-MTTO-i1102.xls). Se identifican los artefactos de entrada y salida involucrados en el proceso y el formato en que se manejan actualmente en el área de Transporte de la UCI.

¹³Procesos fundamentales para alcanzar los objetivos estratégicos de la organización.

Capítulo 2: Modelado del negocio, análisis y diseño

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 documento entregable CIG-CFM-N-MTTO-i1102.doc).

2.2.2. Descripción del proceso de negocio Control de los recursos humanos

El proceso control de los recursos humanos persigue como objetivo guardar los datos de todo el personal del área de transporte para confeccionar el expediente del trabajador, información que será utilizada en varios procesos del área.

Este proceso inicia cuando el especialista decide registrar una persona en el sistema. Primero deben tomarse los datos generales de la persona generándose un expediente. Luego, si es un conductor, se agrega la licencia de conducción y las categorías. Después se introducen los datos de la escuela de recalificación y los chequeos médicos. En caso de tener antecedentes se registran. Si la persona no es un conductor se verifica si necesita de alguna herramienta y se le asocia. Se genera como artefacto de salida el expediente de la persona. (Ver Figura 1)

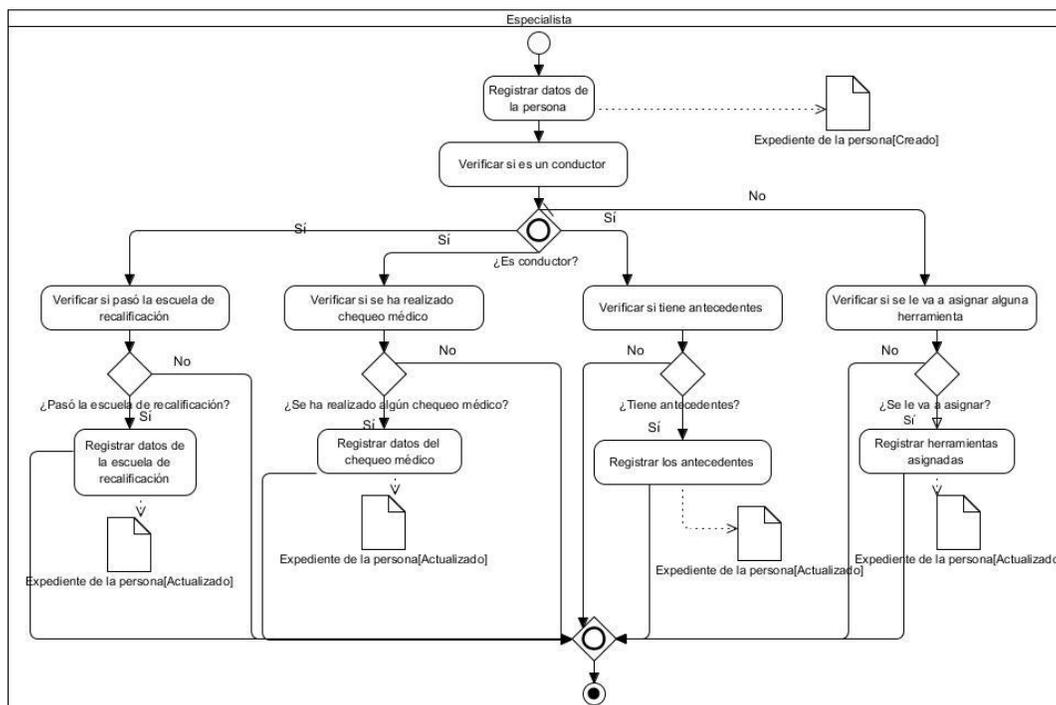


Figura 1 Proceso de negocio Control de los recursos humanos

Este proceso inicia cuando se decide registrar una persona en el sistema. Primero deben tomarse los datos generales de la persona generándose un expediente. Luego, si es un

Capítulo 2: Modelado del negocio, análisis y diseño

conductor, se agregan los antecedentes, los chequeos médicos y la escuela de recalificación. Se registra además las herramientas que necesite. Se genera como artefacto de salida el expediente de la persona.

Para la descripción del proceso de negocio control de los recursos humanos consultar el documento entregable CG-CM-DR-005.doc.

1.1.1. Modelo conceptual

Un modelo conceptual o modelo de dominio, constituye una representación visual para el usuario de los conceptos u objetos significativos del mundo real para un problema o área de interés. Representa conceptos del mundo real, no de los componentes de software, mediante clases conceptuales del dominio del problema, encargándose de capturar los tipos más importantes de objetos y eventos que suceden en el entorno (Pressman, Vol. 1. 2005)

Se representa como un diagrama de clases en el que se muestran conceptos u objetos del dominio del problema, asociaciones entre las clases conceptuales y atributos de las clases conceptuales (Figura 2). Se realiza para favorecer la comunicación entre analistas y clientes, permitiendo comprender las necesidades del usuario y los requisitos del software.

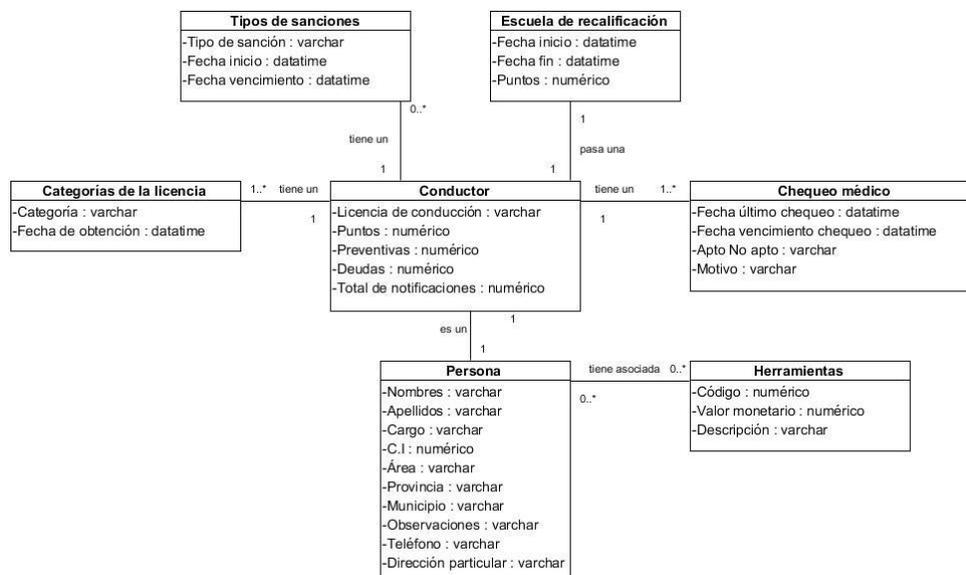


Figura 2 Modelo Conceptual

Cada concepto definido posee un diccionario de datos que expone una breve descripción de los mismos, además de mostrar para cada atributo información relevante como el tipo de datos y

Capítulo 2: Modelado del negocio, análisis y diseño

las restricciones. Para obtener mayor información sobre el modelo conceptual consultar el documento entregable CIG-CFM-N-MTTO-002.doc.

1.1.1. Validación del modelado de negocio

La técnica aplicada para validar el negocio es una Revisión Técnica Formal (RTF), en la cual el cliente tuvo la oportunidad de revisar los artefactos generados. En una primera iteración se identificaron dos No Conformidades (NC) que fueron resueltas para la segunda iteración. Como resultado de la revisión del cliente se emite un Acta de aceptación.

1.2. Requisitos

Un requisito del software es una característica que se debe exhibir por el software desarrollado o adaptado para solucionar un problema particular. (Pressman, Vol. 1. 2005). Estos se dividen en dos grupos: funcionales y no funcionales.

Los requisitos funcionales describen las funciones que el software va a ejecutar. (Pressman, Vol. 1. 2005)

Según Antón los requisitos no funcionales (RNF) *“Describen aspectos del comportamiento de un sistema, capturando las propiedades y restricciones bajo las cuales un sistema debe operar”*. Mientras que Robertson and Robertson los definen como la *“propiedad, o cualidad, que el producto debe tener, como una apariencia, o una propiedad de la velocidad o precisión”*.

Con el fin de comprender lo que desea el cliente y obtener una solución que lo satisfaga, se realiza la captura, especificación, administración, priorización y validación de los requisitos del software. Estas actividades se describen a continuación.

1.2.1. Técnicas para la captura de requisitos

1.2.1.1. Entrevista

Es la técnica de elicitación más utilizada ya que constituye el medio para obtener información sobre el sistema que se desea desarrollar como los requerimientos de usuario. Se utiliza porque el proveedor de requisitos posee dominio del negocio, cuenta con conocimiento general del sistema que se quiere informatizar y domina información de todas las áreas de la Dirección de Transporte de la UCI.

Capítulo 2: Modelado del negocio, análisis y diseño

Se planificó un encuentro con el cliente donde se guió la entrevista usando preguntas abiertas, lo que permitió que surgieran otras preguntas que fueron aportándole mayor claridad al proceso y pudiesen identificarse 5 requisitos funcionales. (Consultar la entrevista en el anexo 1)

1.2.1.2. JAD¹⁴

Como alternativa a las entrevistas individuales esta técnica se basa en cuatro principios: dinámica de grupo, el uso de ayudas visuales para mejorar la comunicación, mantener un proceso organizado y racional y una filosofía de documentación lo que se ve es lo que se obtiene.

Se aplicó en forma de taller, donde se presentaron las funcionalidades del módulo Recursos humanos del Sistema Mantenimiento Vehicular v1.0 a los clientes. Después de observar su funcionamiento se pudo identificar que la gestión de los datos de los familiares no es necesaria, sin embargo es conveniente reutilizar la gestión de las personas, asociar y quitar herramientas a empleados. Como resultado se obtuvo un total de 9 requisitos funcionales.

1.2.1.3. Tormenta de ideas

La Tormenta de ideas es una técnica de grupo que permite la obtención de un gran número de ideas sobre un determinado tema de estudio en un ambiente libre de críticas o juicios. Favorece la participación múltiple de los participantes y la obtención de ideas innovadoras. Frente al JAD tiene la ventaja de que es muy fácil de aprender y requiere poca organización.

Partiendo de las funcionalidades a reutilizar que se identificaron se planteó la posibilidad de generar reportes relacionados con la información del personal. El cliente explicó los informes que realizan de estos datos y otros que pudiesen ser beneficiosos. Además se debatió acerca de la información de los conductores. Se obtuvo como resultado un total de 13 requisitos funcionales.

Como resultado de aplicar las técnicas estudiadas se identificaron los requisitos funcionales que se agruparon de la siguiente forma:

R.F.1. Agrupación de requisitos Gestionar persona

R.F.1.1. Buscar persona.

R.F.1.2. Listar personas.

¹⁴DesarrolloConjunto de Aplicaciones del inglés Joint Application Development

Capítulo 2: Modelado del negocio, análisis y diseño

- R.F.1.3. Listar personas para consumir el servicio de la UCI.
- R.F.1.4. Listar personas para todos los componentes que lo necesiten.
- R.F.1.5. Adicionar persona.
- R.F.1.6. Modificar persona.
- R.F.1.7. Eliminar persona.
- R.F.1.8. Imprimir persona.
- R.F.2. Asociar herramientas a una persona.
- R.F.3. Quitar herramientas a una persona.
- R.F.4. Agrupación de requisitos Gestionar categorías.
 - R.F.4.1. Listar categorías.
 - R.F.4.2. Adicionar categoría.
 - R.F.4.3. Modificar categoría.
 - R.F.4.4. Eliminar categoría.
- R.F.5. Agrupación de requisitos Gestionar tipos de sanciones.
 - R.F.5.1. Listar tipos de sanciones.
 - R.F.5.2. Adicionar tipo de sanción.
 - R.F.5.3. Modificar tipo de sanción.
 - R.F.5.4. Eliminar tipo de sanción.
- R.F.6. Imprimir relación de choferes que se le vence el chequeo médico.
- R.F.7. Imprimir relación de choferes que deben pasar la escuela de recalificación.
- R.F.8. Imprimir relación de personas por área.
- R.F.9. Imprimir relación de personas por cargo.
- R.F.10. Enviar notificación pasar la escuela de recalificación.
- R.F.11. Enviar notificación chequeo médico vencido.
- R.F.12. Enviar notificación vehículo que le corresponde un mantenimiento preventivo.

Capítulo 2: Modelado del negocio, análisis y diseño

R.F.13. Enviar notificación somatón vencido.

R.F.14. Enviar notificación autorizo de parqueo vencido.

Seguidamente se describe la especificación de los requisitos identificados.

1.2.2. Especificación de los requisitos funcionales

A partir de la elicitación de requisitos se realizó su especificación siguiendo la plantilla conformada por el CEIGE. La tabla 2 muestra la especificación del requisito Listar personas para consumir el servicio de la UCI.

Tabla 2 Especificación de requisitos Listar personas para consumir el servicio de la UCI.

Precondiciones		El usuario ha sido validado. Se encuentra disponible el listado de personas del Sistema Capital Humano.
Flujo de eventos		
Flujo básico		
1	El sistema muestra un listado de las personas. Se muestran los siguientes atributos: Nombres Apellidos C.I Cargo Área	
2	Concluye el requisito.	
Pos-condiciones		
1	NA	
Flujos alternativos		
Flujo alternativo		
1	NA	
Pos-condiciones		
1	NA	
Validaciones		
1	NA	
Relaciones	Requisitos	NA
	Incluidos	
	Extensiones	NA

Capítulo 2: Modelado del negocio, análisis y diseño

Conceptos	Persona	Visibles en la interfaz: Nombres Apellidos C.I Cargo Área Utilizados internamente: NA
Requisitos especiales	NA	
Asuntos pendientes	NA	

Prototipo elemental de interfaz gráfica de usuario

El prototipo muestra una ventana titulada "Listado de personas" con los siguientes elementos:

- Campos de entrada para "Nombres:", "Apellidos:" y "Área:" (con menú desplegable).
- Botón de tres puntos "..." a la derecha de los campos.
- Tabla con 5 columnas: "Nombres", "Apellidos", "C.I", "Cargo" y "Área".
- Botones "Cancelar" y "Aceptar" en la parte inferior derecha.

Las especificaciones de requisitos restantes se encuentran en los documentos CIG-CFM-N-MTTO-i2609.doc, CIG-CFM-N-MTTO-i2608.doc, CIG-CFM-N-MTTO-i2607.doc, CIG-CFM-N-MTTO-i2610.doc, CIG-CFM-N-MTTO-i2611.doc, CIG-CFM-N-MTTO-i2904.doc, CIG-CFM-N-MTTO-i2905.doc, CIG-CFM-N-MTTO-i2906.doc, CIG-CFM-N-MTTO-i2907.doc, CIG-CFM-N-MTTO-notificaciones –escuela.doc, CIG-CFM-N-MTTO-notificaciones-chequeo.doc, CIG-CFM-N-MTTO-notificaciones-mtto-preventivo.doc, CIG-CFM-N-MTTO-notificaciones-autorizo de parqueo.doc y CIG-CFM-N-MTTO-notificaciones-somaton.doc.

Una vez realizada la especificación de requisitos de los módulos Taller y Recursos humanos se procede a la administración y priorización de los mismos.

1.2.3. Administración y priorización de requisitos

Capítulo 2: Modelado del negocio, análisis y diseño

La administración de requisitos es una parte esencial para controlar la complejidad y riesgo de un proyecto para un software. La administración de requisitos es efectiva para mejorar los flujos de trabajo a través del ciclo de vida del proyecto. (Sommerville, 2005)

Permitió llevar un control de los requisitos identificados en los módulos Taller y Recursos humanos, lo que favorece conocer el impacto que puede tener un cambio de un requisito sobre el resto. Para su realización se utilizó la herramienta Visual Paradigm y se definieron los elementos de trazabilidad: Requisitos, Modelo conceptual, Componentes y Diseños de casos de pruebas. Como resultado se obtuvo el diagrama de requisitos (Consultar Anexo 2), y las matrices de trazabilidad: requisitos modelo conceptual. (Consultar Anexo 3), requisitos componente (Consultar Anexo 4) y requisitos diseños de casos de pruebas (Consultar Anexo 5).

Después de establecer las relaciones entre los requisitos se realizó la priorización de los mismos haciendo uso de la plantilla Evaluación de requisitos elaborada por el CEIGE. Esta define una serie de indicadores que priorizan los requisitos en dependencia de la complejidad en el orden Alta, Media y Baja. (Consultar documento entregable CIG-CFM-N-MTTO-i6101.xls)

Indicadores definidos:

- **Diferentes comportamientos:** Un mismo requisito se comporta de manera diferente ante determinadas situaciones.
- **Consultas a fuentes de almacenamientos:** Los requisitos pueden presentar diversidad en la cantidad y complejidad de la interacción con la fuente de datos (Base de datos, Ficheros, otros).
- **Restricciones de validación:** Complejidad de todas las validaciones que lleve un requisito, tanto las validaciones en el lado del cliente, como en el servidor.
- **Grado de reutilización:** Complejidad de un requisito, para poder ser reutilizado por otros.
- **Lógica de negocio:** Los requisitos pueden presentar diferentes niveles de complejidad para la implementación de la lógica de negocio que contienen; Ej. Operaciones y métodos matemáticos.

La evaluación de los requisitos teniendo en cuenta los indicadores antes definidos arrojó los siguientes resultados:

Capítulo 2: Modelado del negocio, análisis y diseño

Tabla 3 Complejidad de los requisitos.

Resumen	
Cantidad. de requisitos con complejidad Alta	3
Cantidad de requisitos con complejidad Media	13
Cantidad de requisitos con complejidad Baja	11

Para obtener más información consultar documento entregable CIG-CFM-N-MTTO-i6101.doc.

A continuación se hace referencia a la validación de los requisitos con el objetivo de determinar si existen problemas en su documentación.

1.2.4. Requisitos no funcionales

La identificación de requisitos no funcionales de Órbita para el área de transporte de la UCI la realizó el Analista principal del proyecto. Algunos de los requisitos no funcionales identificados:

Usabilidad

1. El idioma de todas las interfaces de la aplicación será el español puesto que es el idioma básico de los usuarios.
2. El sistema será consistente en el uso de abreviaturas, usará la misma abreviación siempre para la misma palabra y nunca en un elemento de selección o menú. Esto facilita la comprensión de términos al usuario resultando familiar a su lenguaje natural.
3. Los campos de texto tendrán un tamaño estándar de acuerdo con el espacio con que se cuente en el área de la página y en la medida que se llene esa área primaria se agregará la barra de desplazamiento vertical facilitando al usuario la navegación en el sitio.
4. No se utilizarán textos extensos para las etiquetas de la interfaz de usuario, en su lugar se usarán íconos para actividades típicas (Adicionar, Modificar, Eliminar). Los íconos previstos para dichas actividades quedarán plasmados en el Manual de pautas de diseño. Esto permite al usuario familiaridad y conocer el significado de los elementos en cualquier lugar del sistema.
5. Serán notificados los errores cometidos por el usuario y los mensajes incluirán sugerencias de las posibles soluciones permitiendo al usuario solucionar el error.

Capítulo 2: Modelado del negocio, análisis y diseño

6. El sistema expondrá el menú general desde cualquiera de sus páginas facilitando la navegación.
7. La salva de información se hará sólo cuando la información básica del concepto asociado esté completa, de no estarlo el usuario recibirá una notificación de que son necesarios dichos datos y no se continuará el flujo. Esto permite al usuario conocer toda la información necesaria para realizar la operación.
8. El sistema mostrará las opciones desactivadas siempre que no se hayan cumplido las condiciones previas para su activación, evitando así errores del usuario en la gestión.

Fiabilidad

1. El sistema estará disponible durante el horario laboral, los 7 días de la semana, los 365 días del año.
2. El sistema tendrá un respaldo de la información del centro de datos, permitiendo la recuperación ante la pérdida parcial o total de la información. Se realizarán salvadas usando réplicas de datos, las cuales permiten que ciertos datos de la base de datos sean almacenados en más de un sitio. Su principal utilidad es que permite aumentar la disponibilidad de los datos y mejora el funcionamiento de las consultas globales a la base de datos.

SopORTE

1. El sistema será modular, tomando como criterio para la creación de los módulos los macro procesos de negocio identificados, favoreciendo así la incorporación, modificación o eliminación de funcionalidades.
2. La codificación del sistema será estándar y las funcionalidades serán comentadas haciendo uso del estándar definido por el CEIGE.

Seguridad

1. Se garantiza con Acaxia.

Interoperabilidad

1. El sistema permitirá consumir el servicio Web del Sistema Capital Humano para obtener los datos de los trabajadores que son plantilla de la UCI.

Capítulo 2: Modelado del negocio, análisis y diseño

2. El sistema permitirá consumir el servicio Web de identificación para obtener la foto de los trabajadores que son plantilla de la UCI.
3. El sistema permitirá consumir el servicio Web para el envío de notificaciones.

Restricciones de diseño

1. El sistema será una aplicación web centralizada.
2. El sistema se implementará usando el lenguaje PHP.
3. El sistema estará regido por un estilo arquitectónico basado en componentes. Cada uno de los componentes seguirá el patrón Modelo Vista Controlador.
4. El sistema usará el marco de trabajo ExtJS para manejar las vistas.
5. El sistema usará el marco de trabajo Zend para manejar la lógica de negocio.
6. El sistema usará el marco de trabajo Doctrine para manejar la persistencia de los datos.
7. El sistema usará PostgresSql como gestor de base de datos.

Para obtener mayor información sobre el resto de los requisitos no funcionales identificados (consultar documento entregable CIG-CFM-N-MTTO-i2301.doc).

Luego de recolectar toda la información referente a los módulos Taller y Recursos humanos, se da paso a la siguiente etapa, el diseño de la propuesta de solución.

1.3. Diseño

Los contenidos que a continuación se abordan constituyen los elementos esenciales que se realizaron durante el diseño de la solución.

1.3.1. Patrón arquitectónico Modelo-Vista-Controlador (MVC)

Para la implementación de los módulos Taller y Recursos humanos se hace uso del patrón arquitectónico MVC, ya que está implementado dentro de Sauxe, sobre el cual será desarrollada la solución. Permite separar la interfaz de usuario, el control y los datos en el sistema. La Vista visualiza las interfaces de usuario y la respuesta a las peticiones hacia el controlador. El Controlador está constituido por las clases controladoras que permiten la interacción entre la vista y el modelo, administrando los eventos que se generan en el sistema. El Modelo contiene las clases encargadas de la lógica del negocio y las que se dedican al acceso a datos de la base de datos.

Capítulo 2: Modelado del negocio, análisis y diseño

Con el patrón arquitectónico definido es preciso determinar los patrones de diseño a emplearse, los que se describen a continuación.

1.3.2. Patrones de diseño

“Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software.” (Fowler, 1996). Brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares.

En el diseño de la solución fueron empleados los patrones de diseño siguientes:

1. Patrones generales de software para asignar responsabilidades (GRASP):

El patrón **Experto** se encuentra en todas las clases definidas pues las mismas cuentan con la información necesaria para cumplir con la responsabilidad asignada.

El **Bajo acoplamiento** se evidencia en la poca relación existente entre las clases que conforman los módulos.

El empleo de la **Alta cohesión** garantiza que cada clase realice solamente las funciones que estén en correspondencia con la responsabilidad que posea.

El patrón **Controlador** fue empleado para la creación de las clases controladoras que intervienen en los diferentes procesos, ejemplo de ello es GestionarpersonasController.

El patrón **Creador** permite la creación de objetos, donde la nueva clase deberá ser creada por la que tiene toda la información necesaria para realizar la acción. Este patrón se puede observar en la clase GestionarpersonasController, quien es la responsable de crear los objetos del modelo para acceder a las clases correspondientes.

2. Patrones de Grupo de Cuatro (GoF):

El patrón **Fachada** se utiliza para lograr la integración entre los diferentes módulos del sistema, facilitando la comunicación a través del uso de mecanismo de inversión de control (IoC), en el cual son publicados los servicios necesarios.

Singleton garantiza que una clase sólo tenga una instancia y proporciona un punto de acceso global a ella. El mismo puede apreciarse en las clases del dominio y las contenidas en la Vista.

1.3.3. Mecanismos de diseño

Capítulo 2: Modelado del negocio, análisis y diseño

En el diseño de la solución se hizo uso de varios mecanismos de diseño con el objetivo de simplificar los diagramas de clases permitiendo mejorar la comunicación entre los miembros del equipo de desarrollo. Permiten simplificar el tiempo de asimilación del diseño por parte de los desarrolladores. En el módulo Persona y Taller de Órbita se definen los siguientes mecanismos:

Mecanismo de diseño para los nombres de las clases.

Los diagramas de clases del diseño de Órbita se realizaron por agrupaciones de requisitos funcionales. En el caso de las agrupaciones de requisitos conformadas por adicionar, modificar, eliminar, listar y buscar se dividen en dos clases, una que agrupa los requisitos adicionar, modificar y eliminar; a la cual se le denomina AME_Nombre de la clase; y otra clase que agrupa los requisitos listar y buscar a la cual se le pone el nombre LB_Nombre de la clase. Además, los elementos de almacenamiento de datos se agrupan en una clase que se denomina storeNombre de la clase.



Figura 3 Mecanismo de diseño para los nombres de las clases.

Mecanismo de diseño: inversión de control para el consumo de servicios entre componentes desde las clases controladoras y modelos.

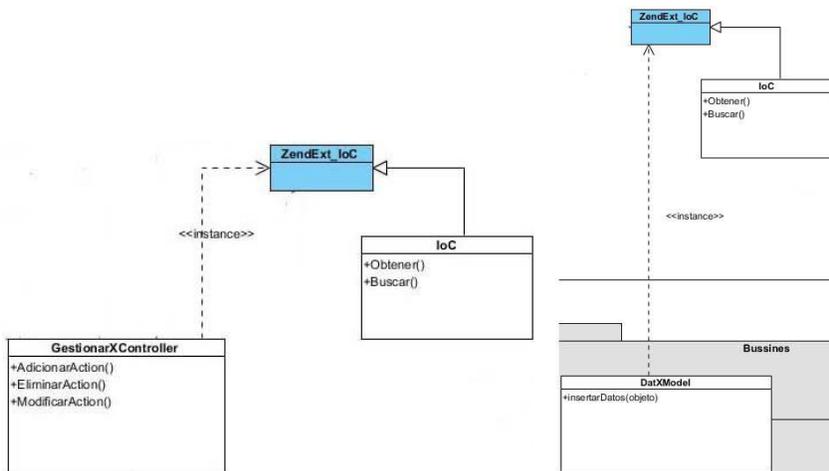


Figura 4 Mecanismo de diseño inversión de control para el consumo de servicios entre componentes desde las clases controladoras y modelos.

Capítulo 2: Modelado del negocio, análisis y diseño

Siempre que se necesite consumir un servicio de otro componente desde una clase controladora o de modelo se instanciará la clase IoC la cual hereda de ZendExt_IoC que permite tener acceso a los servicios de todos los componentes del sistema Control de Flota y Mantenimiento.

Mecanismo de diseño para las clases controladoras.

Todas las clases controladoras definidas en el diseño propuesto heredan de la clase ZendExt_Controller_Secure, ya que en ella se incluyen numerosas funcionalidades comunes en todas las controladoras.



Figura 5 Mecanismo de diseño para las clases controladoras.

El resto de los mecanismos de diseño definidos se encuentran en el documento entregable CIG-CFM-N-MTTO-i3801.doc.

Ya establecidos los elementos significativos del diseño se procede a modelar los diagramas de clases que se describen a continuación.

1.3.4. 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, dependencias. (Visconti, y otros)

Se realizaron los diagramas de clases para la agrupación de requisitos Gestionar Recursos humanos (consultar **Figura 6**) y Recuperaciones de personas (consultar documento entregable Diagramas del diseño.doc).

El diagrama de clases para la agrupación de requisitos Gestionar Recursos humanos cuenta con las principales clases, operaciones y relaciones que se necesitan para darle cumplimiento a los requisitos funcionales relacionados con la gestión de los recursos humanos. La interfaces

Capítulo 2: Modelado del negocio, análisis y diseño

son LBRecursosHumanos.js, LBRechum.js, AMCategorias.js, AHerramientasEmpleados.js, AMSanciones.js, AMRecursosHumanos.js, AMFoto.js y gestionarpersonas.html. Dentro de la lógica de negocio se encuentra gestionarpersonasController.php encargada de establecer la comunicación entre la vista y el modelo. Para el acceso a datos están las clases DatTrabajadorModel.php, DatEscuelaConduccionModel.php, DatCategoriaModel.php, DatTrabajadorTipoSancionModel.php, DatFotoModel.php, DatHerramientaTrabajadorModel.php y DatTrabajadorCategoriaModel.php.

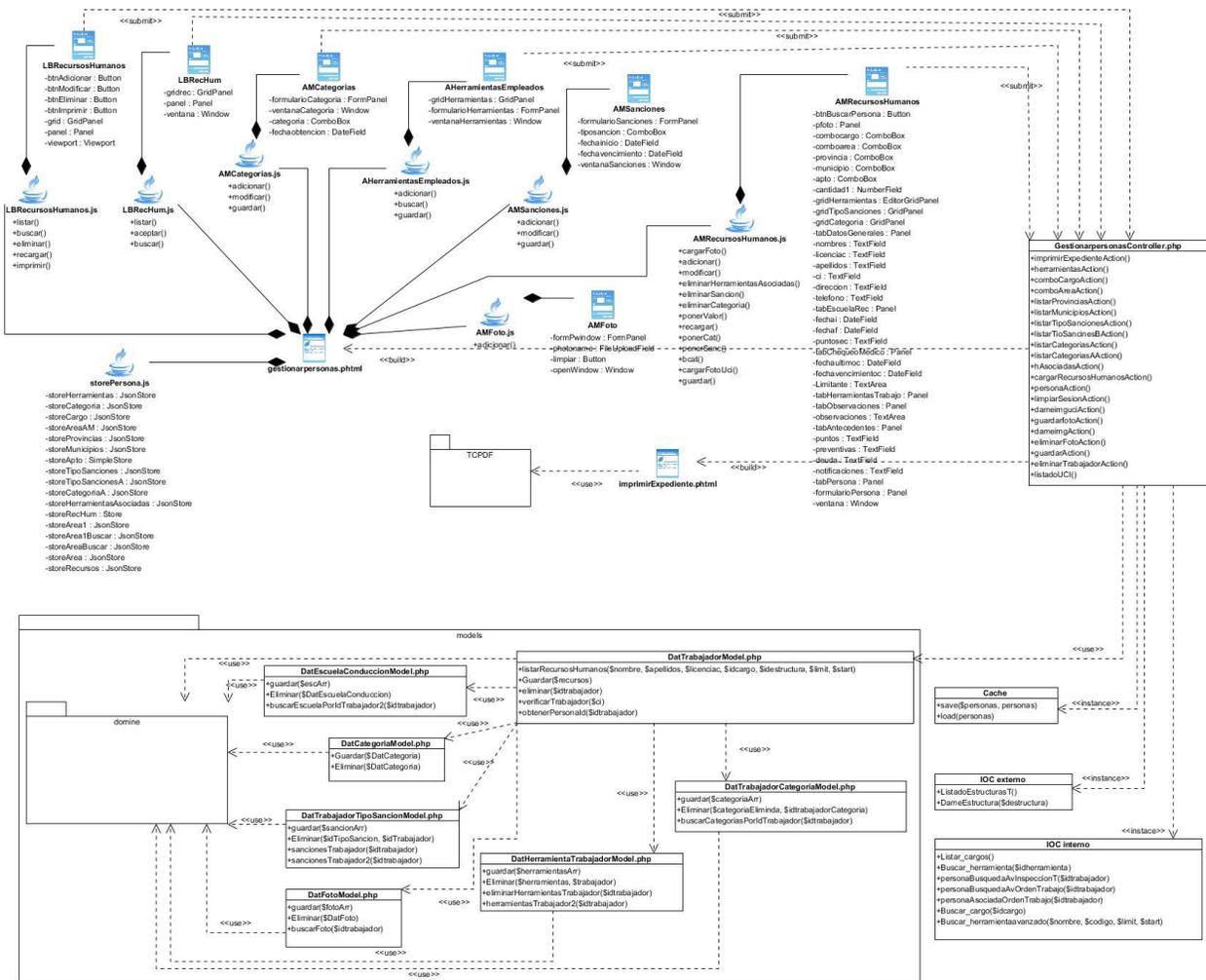


Figura 6 Diagrama de clases del diseño Gestionar Recursos humanos.

El diagrama de clases para la agrupación de requisitos Recuperaciones de personas cuenta con las principales clases, operaciones y relaciones que se necesitan para darle cumplimiento a los requisitos funcionales relacionados con la generación de reportes asociados a los recursos humanos. Las interfaces son recuperacionesperonas.phtml, imptimirPersonasPorCargo.js

Capítulo 2: Modelado del negocio, análisis y diseño

imprimirChoferesChequeoMedicoVencido.js, imprimirChoferesEscuelaVencida.js e imptimirPersonasPorArea.js. Dentro de la lógica de negocio se encuentra Recuperacionespersona.php encargada de establecer la comunicación entre la vista y el modelo. Para el acceso a datos están la clase DatTrabajadorModel.php.

1.3.5. Descripción de las clases del diseño

Partiendo del diseño realizado se describe a continuación un fragmento de la clase **DatTrabajadorModel**, especificando sus funcionalidades y una breve explicación.

Tabla 4 Descripción de la clase **DatTrabajadorModel**.

Nombre: DatTrabajadorModel	
Tipos de clase: Modelo	
Para cada responsabilidad:	
Nombre	Guardar(\$recursos)
Descripción	A partir de un arreglo con los datos de un trabajador realiza su adición al sistema.
Nombre	listarRecursosHumanos(\$nombres,\$apellidos,\$licenciad,\$idcargo,\$idestructura,\$limit,\$start)
Descripción	Permite listar los datos de los recursos humanos que se encuentran en el sistema. Además, puede filtrarse por nombre, apellidos, licencia de conducción, cargo y área.
Nombre	obtenerPersonald(\$idtrabajador)
Descripción	A partir del identificador del trabajador devuelve los datos de la persona.
Nombre	eliminar(\$idtrabajador)
Descripción	Elimina un recurso humano
Nombre	imprimirPersonasPorArea(\$idestructura)
Descripción	Obtiene los datos de las personas que pertenecen a un área determinada.

Consultar el documento entregable Descripción del diseño.doc que contiene la descripción de todas las clases.

Después de haber descrito las clases involucradas en el diseño de la solución se realizaron los diagramas de secuencia que se muestran a continuación.

1.3.6. Diagramas de secuencia

Los diagramas de secuencia modelan el flujo de la lógica dentro del sistema de forma visual, permitiendo documentarla y validarla. Proporcionan una buena base para identificar el

Capítulo 2: Modelado del negocio, análisis y diseño

comportamiento del sistema. Son una buena herramienta para explorar la lógica de una operación compleja o los elementos implicados en la prestación de un servicio.

En la **Figura 7** **Diagrama de secuencia Listar personas para consumir el servicio de la UCI.** **Figura 1** se muestra el diagrama de secuencia que responde al R.F. Listar personas para consumir el servicio de la UCI.

Los diagramas de secuencia generados pueden consultarse en el documento Diagramas del diseño.doc. Después de realizado el diseño de la solución se hace necesario validarlo por lo que a continuación se describen las métricas aplicadas.

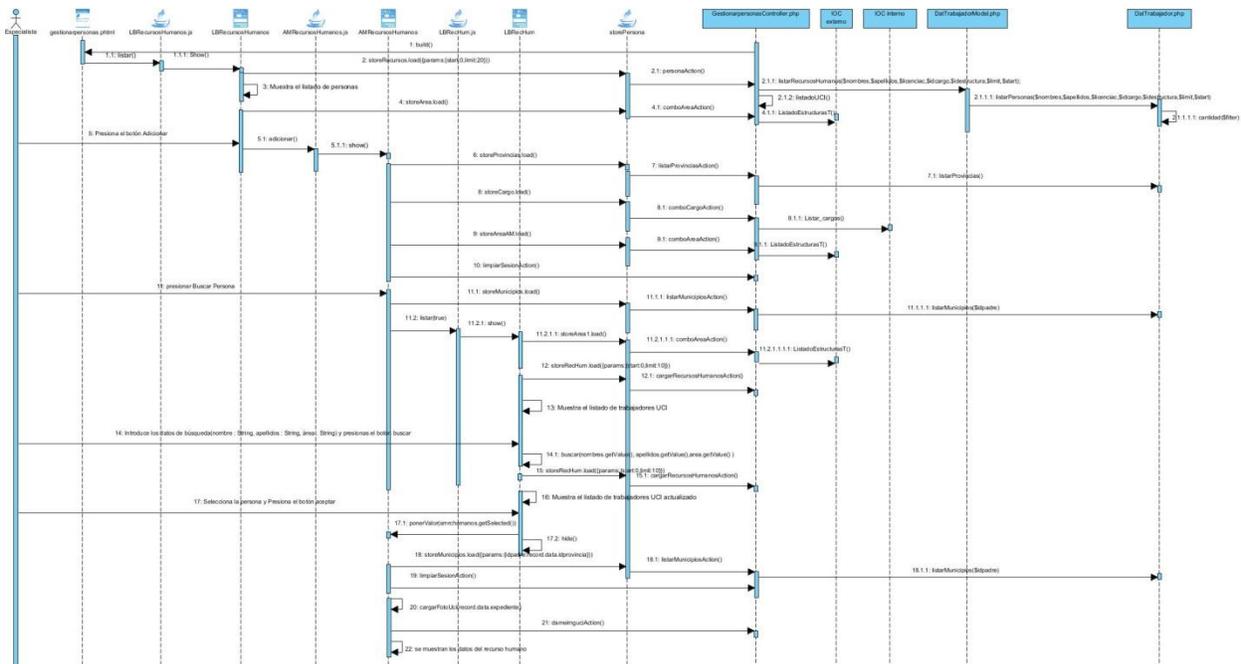


Figura 7 Diagrama de secuencia Listar personas para consumir el servicio de la UCI.

1.3.7. Modelo de datos

Puede definirse un modelo de datos como un conjunto de conceptos, reglas y convenciones que permiten construir una representación organizada de un sistema real. (Lorenzo Castro, 1999-2000)

Para el desarrollo de las mejoras funcionales a los módulos Taller y Recursos humanos se establecieron los conceptos y las relaciones que se establecen entre ellos. La **Figura 8** muestra un fragmento del modelo de datos que los representa. Para más información sobre el modelo de datos (consultar documento entregable CIG-CFM-N-MTTO-i3201.doc).

Capítulo 2: Modelado del negocio, análisis y diseño

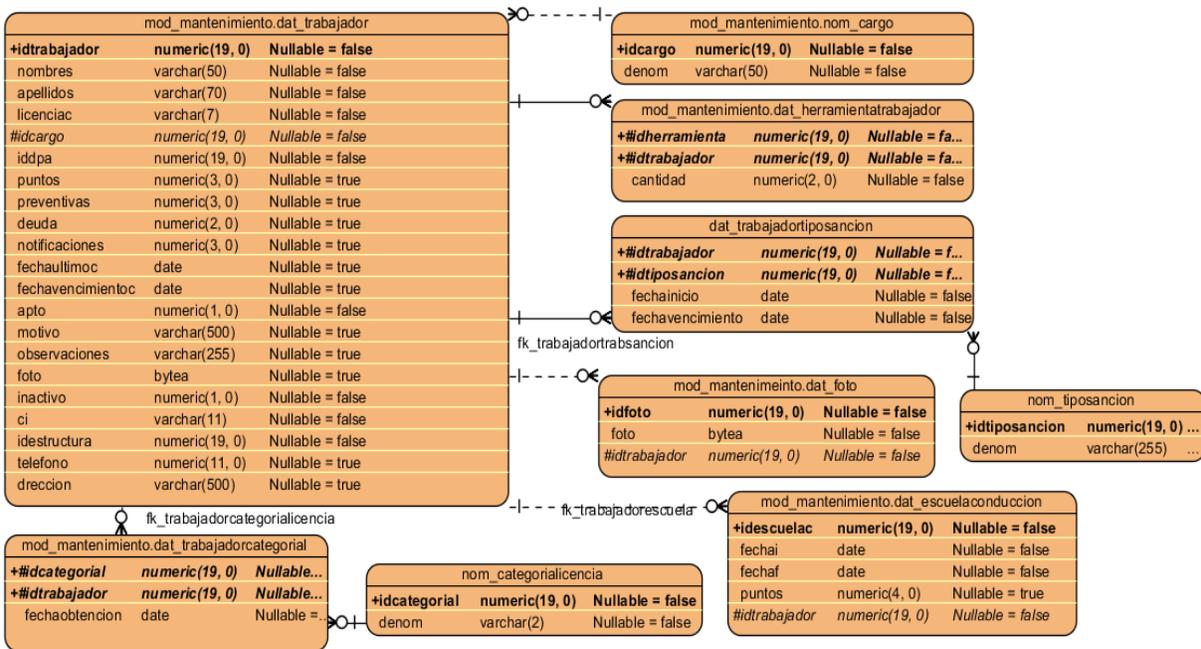


Figura 8 Fragmento del modelo de datos.

1.4. Conclusiones parciales

A partir del modelado del negocio se obtuvo un conocimiento objetivo del área de transporte de la UCI, propiciando definir los conceptos asociados al tema, la comprensión de los procesos involucrados y el funcionamiento apropiado de esta sección de la entidad. Se trabajó en las necesidades del usuario donde se obtuvieron los requisitos que la entidad desea informatizar y que debe poseer el producto final. Se realizó el diseño de la solución.

Capítulo 3: Implementación y prueba

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

3.1. Introducción

Este capítulo aborda elementos que tributan a la obtención de una versión funcional del sistema. Se documenta el diagrama de componentes, el modelo de despliegue y estándares de codificación utilizados. Se realiza una descripción de los elementos significativos del código y el funcionamiento por módulos. Se describen las pruebas aplicadas al software con los resultados obtenidos.

3.2. Implementación

Tomando como punto de partida los artefactos generados en el análisis y diseño, se generan los ficheros de código que respondan a las funcionalidades del sistema solicitadas mediante la lógica de negocio diseñada. (CEIGE, 2013)

A continuación se tratan los aspectos fundamentales que se realizaron durante esta etapa comenzando con el diagrama de componentes.

3.2.1. Diagrama de componentes

El conjunto de componentes y la representación de sus relaciones de manera gráfica constituyen un diagrama de componentes. (Ferré Grau, y otros) Definiéndose componente como la representación física de una parte de un sistema modular y reemplazable que encapsula la implementación. (Object Management Group, 2001)

En la **Figura 9** se muestra el diagrama de componentes de Órbita. La propuesta de solución se enmarca en los elementos resaltados, siendo estos:

- Estructura y Composición que registra las áreas de la UCI.
- Seguridad que maneja la seguridad del sistema y se le integró el componente notificaciones.
- La biblioteca TCPDF que se utiliza para la generación de reportes.
- En Persona, Ejecución y Vehículo se incluyen las funcionalidades identificadas.

Con la distribución física de a solución modelada es preciso definir los elementos necesarios para desplegarla, contenido que se explica a continuación.

Capítulo 3: Implementación y prueba

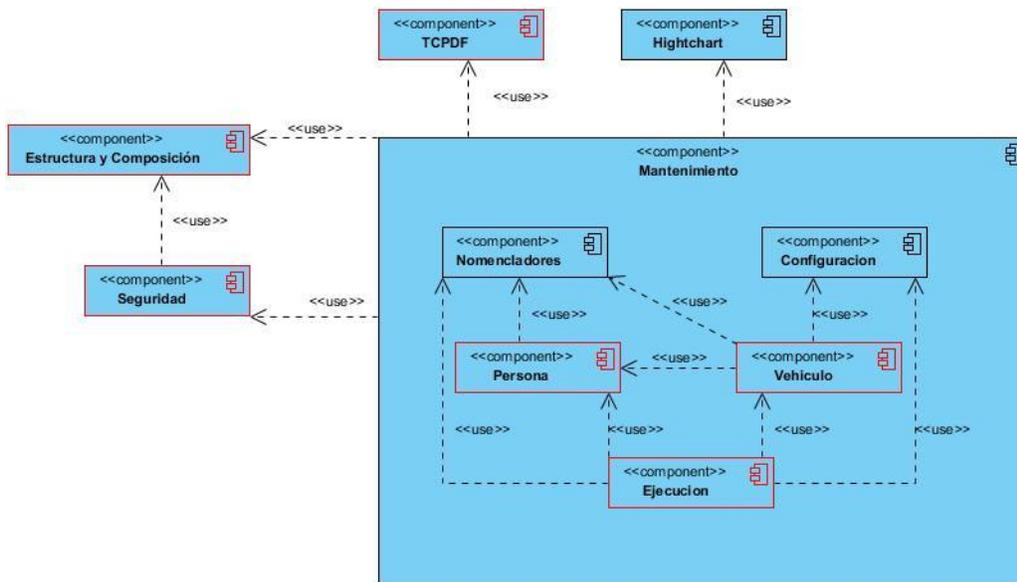


Figura 9 Diagrama de componentes.

3.2.2. Modelo de despliegue

Un diagrama de despliegue muestra cómo se configuran las instancias de los componentes y los procesos para la ejecución en las instancias de los nodos de proceso. (Ferré Grau, y otros) El modelo de despliegue elaborado para Órbita se muestra en la Figura 10. Lo componen:

PC Cliente: Accede a la aplicación a través de un computador, donde es ejecutada mediante el navegador Mozilla Firefox versión 4.0 o superior, sobre cualquier sistema operativo.

Servidor Aplicaciones Web: El servidor de aplicaciones empleado donde radica la lógica de negocio de la aplicación es el Servidor Web Apache 2.2 utilizando biblioteca de clases adicionales: PHP 5.

Servidor de mensajería instantánea: Emplea el Servidor Openfire 3.7.1.

Servidor de Base de datos: Sistema Gestor de Base de Datos: PostgreSQL 9.1 donde se encuentra la base de datos que utiliza el sistema.

Servidor UDDI: Servidor que brinda el servicio para obtener la plantilla de los trabajadores de la UCI y la foto de cada uno.

Servidor de correo electrónico: Radica el servidor de correo electrónico Zimbra de la UCI.

Impresora: Utilizada para imprimir los reportes de Órbita.

Capítulo 3: Implementación y prueba

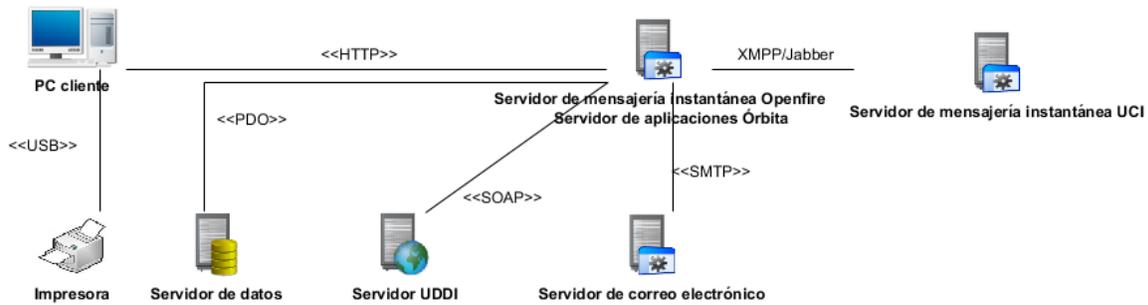


Figura 10 Diagrama de despliegue.

3.2.3. 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, comprensión y mantenimiento del código. Establecerán las pautas que conlleven a lograr un código más legible y reutilizable, de tal forma que pueda aumentar su mantenibilidad a lo largo del tiempo. (CEIGE, 2012)

Seguidamente se especifican los estándares de codificación usados para el desarrollo de la solución:

Nomenclatura de las clases

Los nombres de las clases comienzan con la primera letra en mayúscula y el resto en minúscula.

- En las clases controladoras los nombres irán seguidos de la palabra: “Controller”. Ejemplo: GestionarpersonaController
- Las clases de los modelos que se encuentran dentro de la carpeta del negocio después del nombre llevan la palabra: “Model”. Ejemplo: DatTrabajadorModel.
- Las clases que se encuentran dentro de la carpeta del dominio el nombre que reciben es el de la tabla en la Base de Datos. Ejemplo: DatTrabajador.
- Las clases que se encuentran dentro de la carpeta generado el nombre comienza con la palabra: “Base” y seguido el nombre de la tabla en la Base de Datos. Ejemplo: BaseDatTrabajador.
- En la vista los nombres comienzan con las letras iniciales mayúsculas haciendo

Capítulo 3: Implementación y prueba

referencia a la agrupación de requisitos que se maneja. Agrupación listar, buscar se le pone el nombre LB_Nombre de la clase, una que agrupa los requisitos adicionar, modificar y eliminar; a la cual se le denomina AME_Nombre de la clase y los elementos de almacenamiento de datos se agrupan en una clase que se denomina storeNombre de la clase. Ejemplo: LBRecursosHumanos, AMERecursosHumanos, storePerona.

Nomenclatura de las funciones

El nombre a emplear para las funciones se escribe con la primera palabra en minúscula, en caso de que sea un nombre compuesto a partir de la segunda palabra comienzan con mayúscula. En la clase controladora van seguidas de la palabra "Action". Ejemplo: comboCargoAction

Nomenclatura de las variables

El nombre a emplear para las variables se escribe con la primera palabra en minúscula, en caso de que sea un nombre compuesto se empleará a partir de la segunda palabra letra inicial mayúscula. Además, debe iniciar con el tipo de dato. Ejemplo: arrayPersona

Una vez establecidas las bases para el desarrollo de la solución, se lleva a cabo la implementación de las funcionalidades identificadas. Se describen a continuación los elementos significativos del código y el funcionamiento de cada módulo.

3.2.4. Descripción de la implementación y funcionamiento por módulos

Módulo Recursos humanos:

Se hizo uso del servicio web del Sistema de Capital Humano disponible en la UDDI para establecer una sincronización con los datos primarios de los trabajadores del área de transporte de la UCI.

La función encargada de consumir el servicio del Sistema de Capital Humano es listadoUCI (Figura 11). Primero crea una instancia de la caché a través de un objeto de inversión de control y procede a cargar la de personas. Si no está creada verifica que esté disponible el servicio del Sistema de Capital Humano, en caso positivo se guarda en la caché los datos de las personas plantilla de la UCI, si no termina.

Capítulo 3: Implementación y prueba

```
function listadoUCI() {
    $cache = ZendExt_Cache :: getInstance();
    $personas = $cache->load('personas');
    if (null == $personas) {
        if( fopen("http://assets.uci.cu/servicios/v4/AssetsWS.wsdl", "r") ) {
            $SoapClientPersonas = new SoapClient("http://assets.uci.cu/servicios/v4/AssetsWS.wsdl");
            $personas = $SoapClientPersonas->ObtenerPersonas();
            $cache->save($personas, 'personas');
        }
    }
}
```

Figura 11 Código del método listadoUCI

Para mostrar el listado de personas que son plantilla de la UCI el usuario debe acceder a la interfaz Adicionar persona, una vez mostrada se presiona el botón Buscar persona y este invoca el método cargarRecursosHumanosAction. Primero crea una instancia de la caché a través de un objeto de inversión de control y procede a cargar la de personas. Verifica que esté creada y en caso positivo muestra el listado en caso contrario a través del controlador de excepciones se muestra al usuario un mensaje de error (Figura 13).

```
function cargarRecursosHumanosAction() {
    $cache = ZendExt_Cache :: getInstance();
    $personas = $cache->load('personas');

    if (null == $personas) {
        throw new ZendExt_Exception('MTTOHOT003');
    }
}
```

Figura 12 Código del método cargarRecursosHumanosAction



Figura 13 Mensaje de error

Además, para obtener la foto de los trabajadores se utilizó el servicio de identificación. Cuando el usuario accede al listado de los recursos humanos que son plantilla de la UCI selecciona la persona y presiona el botón aceptar. Se procede a mostrar los datos en la interfaz de los datos generales y se invoca el método dameimguciAction. Entonces se verifica que esté disponible el servicio y si es correcto obtiene la foto de la persona dado su identificador y se muestra en la interfaz. En caso contrario se coloca la imagen por defecto, lo que permite que el usuario pueda

Capítulo 3: Implementación y prueba

agregarla manualmente.

```
function dameimguciAction() {
    $idtrabajador = $_GET['idtrabajador'];
    if( fopen("http://identificacion.uci.cu/servicios/v5/servicios.php?wsdl", "r")){
        $identificacion = new SoapClient("http://identificacion.uci.cu/servicios/v5/servicios.php?wsdl");
        $imag = $identificacion->ObtenerFotoDadoIdExpediente($idtrabajador)->valorFoto;
        $contents = file_get_contents($imag);
        $_SESSION["foto"] = $contents;
        header("Content-type: image/jpg");
        echo $contents;
    } else {
        if (!file_exists($contents)){
            $contents = $this->fotosDir . 'persona.jpg';
            $imag = file_get_contents($contents);
            echo $imag;
        }
        return;
    }
}
```

Figura 14 Código del método dameimguciAction

Se hace uso de los siguientes servicios que son brindados por otros componentes del sistema:

Servicio	Componente que lo brinda	Parámetros	Descripción
ListadoEstructurasT	Metadatos	-	Devuelve el listado de las áreas de la UCI.
Buscar_herramienta avanzado	Nomencladores	Código de la herramienta, descripción de la herramienta, cantidad máxima a mostrar los datos, número desde donde comienza a mostrar los datos	Obtiene el listado de herramientas.
Listar_cargos	Nomencladores	-	Devuelve el listado de los cargos.
personaBusquedaAv InspeccionT	Vehículo	Identificador del trabajador	Devuelve los datos de una inspección técnica asociada a un trabajador
personaBusquedaAv OrdenTrabajo	Ejecución	Identificador del trabajador	Devuelve los datos de una orden de trabajo asociada a un trabajador
Listar_rangos	Nomencladores	-	Devuelve el listado de los rangos.
personaAsociadaOrdenTrabajo	Ejecución	Identificador del trabajador	Devuelve si el trabajador está asociado a una

Capítulo 3: Implementación y prueba

			orden de trabajo
--	--	--	------------------

Se implementaron los siguientes servicios que son utilizados por otros componentes del sistema:

Tabla 5 Servicios que brinda el componente persona.

Servicio	Componentes que lo utilizan	Descripción
listar_RecursosHumanos	Vehículo Ejecución	Realiza una búsqueda de las personas registradas en el sistema que cumplan con los parámetros introducidos, de ser nulos, los devuelve todos.
herramientaAsociada	Nomencladores	Obtiene la cantidad de personas que están asociadas a una herramienta dado su identificador.
buscar_Trabajador	Ejecución Vehículo	Dado el identificador de la persona devuelve sus datos.
buscar_Trabajador_Autoriza	Vehículo Ejecución	Devuelve los datos del trabajador encargado de autorizar.
buscar_TrabajadorConductor	Vehículo	Obtiene los datos de los conductores registrados en el sistema.
cargo_Asociado	Nomencladores	Dado el identificador del cargo devuelve la cantidad de personas asociadas a ese cargo.
rango_Asociado	Nomencladores	Dado el identificador del rango devuelve la cantidad de personas asociadas a él.

El acceso al módulo Recursos humanos se realiza a través del menú vertical a la izquierda del navegador dentro del subsistema mantenimiento. Se despliegan dos elementos:

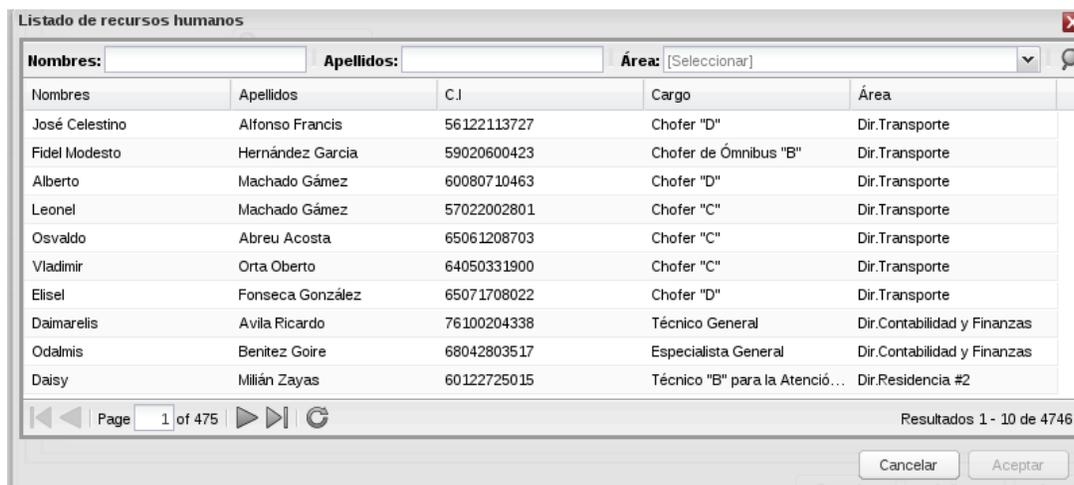
- Recursos humanos, que permite la gestión de la información de los trabajadores del área de transporte de la UCI, creando un expediente digital de cada persona con sus

Capítulo 3: Implementación y prueba

datos generales. Se registran las herramientas de trabajo. En caso de ser conductor se recogen las categorías, sanciones, el chequeo médico y la escuela de recalificación.

- Recuperaciones, que partiendo de la información de los trabajadores existentes en el sistema posibilita la generación de reportes sobre los choferes que se le vence el chequeo médico, los choferes que deben pasar la escuela de recalificación, las personas por área y por cargo.

La Figura 15 muestra la interfaz para listar los trabajadores que son plantilla de la UCI.



Nombres	Apellidos	C.I	Cargo	Área
José Celestino	Alfonso Francis	56122113727	Chofer "D"	Dir.Transporte
Fidel Modesto	Hernández Garcia	59020600423	Chofer de Ómnibus "B"	Dir.Transporte
Alberto	Machado Gámez	60080710463	Chofer "D"	Dir.Transporte
Leonel	Machado Gámez	57022002801	Chofer "C"	Dir.Transporte
Osvaldo	Abreu Acosta	65061208703	Chofer "C"	Dir.Transporte
Vladimir	Orta Obero	64050331900	Chofer "C"	Dir.Transporte
Elisel	Fonseca González	65071708022	Chofer "D"	Dir.Transporte
Daimarelis	Avila Ricardo	76100204338	Técnico General	Dir.Contabilidad y Finanzas
Odalmis	Benitez Goire	68042803517	Especialista General	Dir.Contabilidad y Finanzas
Daisy	Milián Zayas	60122725015	Técnico "B" para la Atención...	Dir.Residencia #2

Figura 15 Interfaz de usuario Listar recursos humanos UCI

Módulo Taller:

El desarrollo de las mejoras funcionales al módulo Taller se realizó en dos etapas:

- Integración del módulo Notificaciones de Sauxe al Sistema Control de Flota y Mantenimiento. (consultar Guía de instalación de Notificaciones.doc).

Requiere la instalación de sun-java6-jre y sun-java6-bin para el servidor de mensajería Openfire, adicionándole el complemento para el manejo de usuarios vía HTTP.

Se necesita configurar el Apache, pues se requiere de activar el módulo mod_proxy.

El componente Notificaciones se integra con el subsistema Seguridad.

- Implementación de la solución.

Las notificaciones ocurren en diferentes módulos que se traducen a componentes. Se determinó el uso de un servicio por componente con la menor cantidad de parámetros posibles

Capítulo 3: Implementación y prueba

para beneficiar la reutilización del código. Se implementaron los servicios llamados `listarNotificacionesPersona`, `listarNotificacionesVehiculo` y `listarNotificacionesEjecucion`. Cada uno de ellos realiza las instrucciones en correspondencia con su dominio de información. Estos verifican si dado una fecha y la cantidad de días de antelación hay posibles notificaciones, generando un listado con los mensajes.

Tabla 6 Servicios de notificaciones.

Servicio	Clase	Componente
<code>listarNotificacionesPersona</code>	<code>DatTrabajadorService</code>	Persona
<code>listarNotificacionesVehiculo</code>	<code>DatVehiculoService</code>	Vehículo
<code>listarNotificacionesEjecucion</code>	<code>DatOrdentrabajoService</code>	Ejecución
<code>listarNotificaciones</code>	<code>NotifEjecucionService</code>	Ejecución

En el componente ejecución se estableció la clase `NotifEjecucionService.php`, que contiene el método `listarNotificaciones`. Este hace uso de servicios para listar las notificaciones y luego enviarlas utilizando el objeto `messagebus`. Este objeto es la entidad que se encarga de gestionar las notificaciones y vías a notificar.

Cuando se ejecutan las notificaciones a través de líneas de comando (el cron ejecuta un archivo que las invoca), no era posible guardarlas en la base de datos puesto que no hay un usuario autenticado en el sistema. Esto provocaba que algunos valores de las variables globales no estuvieran inicializados y el método `save` no funcionara. Para solucionarlo se creó una conexión a la base de datos usando la clase `TransactionManager`, que se encuentra dentro de `lib/ZendExt/Aspect`. Se creó un objeto de esta clase (1), se inició la conexión del módulo seguridad (2), se guarda el registro con esa conexión (3) y se salvan todos los registros creados o modificados. Después se acepta la transacción abierta por cada conexión activa (4).

```
1 $tm = ZendExt_Aspect_TransactionManager::getInstance();
2 $conn = $tm->openConnections('seguridad', true);
3 $datbuzonnotif->save($conn);
4 $tm->commitTransactions();
```

Figura 16 Código para la salva de las notificaciones

Haciendo uso de Soap se creó el servicio web `notificacionesMtto`, que es invocado desde la clase `prueba_service`, con la fecha actual y la cantidad predeterminada de días de antelación con que se quiere recibir la notificación. Este archivo está escrito en lenguaje PHP y se

Capítulo 3: Implementación y prueba

comporta como un cliente que consume los servicios prestados por el sistema. El uso del servicio web permite que pueda ejecutarse desde cualquier lugar en la red o pueda ser invocado por otro sistema.

La emisión de las notificaciones es cíclica, requiriéndose su ejecución diariamente. Para ello se necesita configurar el Cron añadiendo como tarea la ejecución automática del comando `0 0 * * * /usr/bin/php -q /media/Datos/f2/web/mantenimiento/webservices/prueba_service.php`, que establece la ejecución del archivo `prueba_service` diariamente.

Para el envío del correo electrónico se emplea la biblioteca Mail que está incluida dentro de Sauxe y el servidor de la UCI Zimbra. En cuanto a la mensajería instantánea se utiliza Openfire.

La Figura 17 muestra las notificaciones en el sistema y en la Figura 18 las recibidas vía correo electrónico.

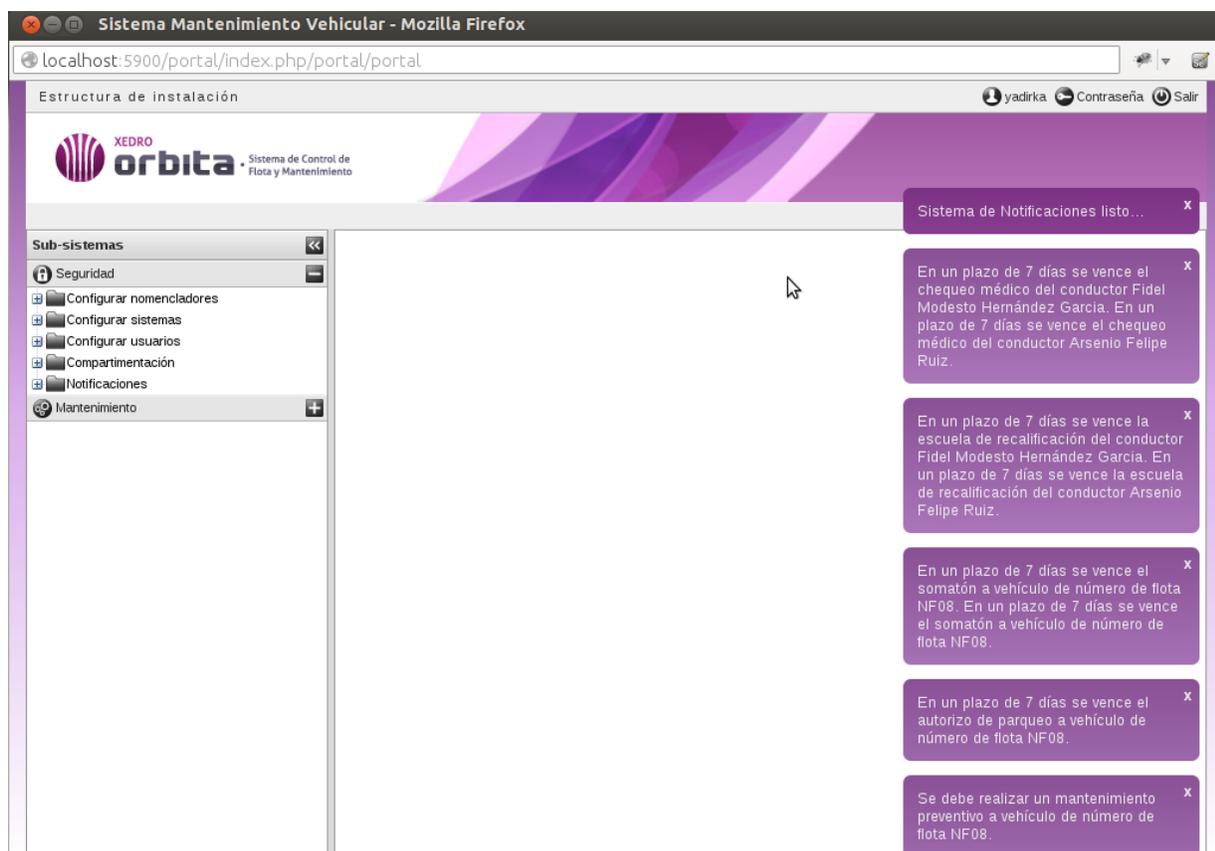


Figura 17 Notificaciones en el sistema.

Capítulo 3: Implementación y prueba

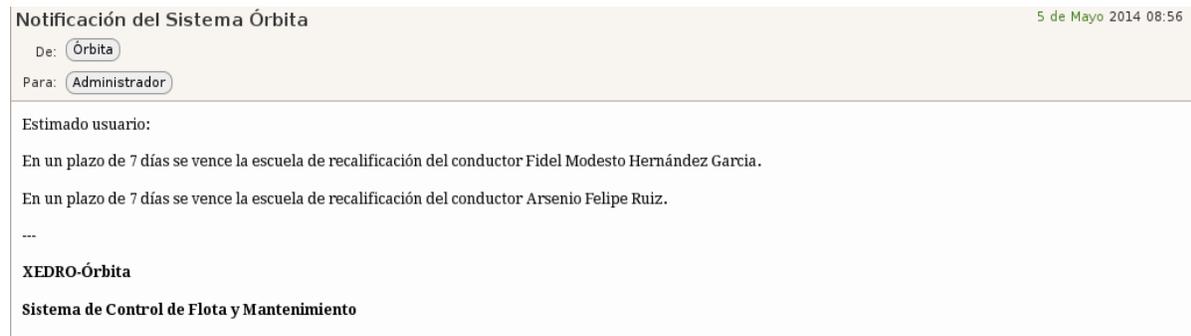


Figura 18 Notificaciones recibidas vía correo electrónico.

A continuación se exponen las técnicas de validación de requisitos aplicadas así como las métricas utilizadas en el diseño.

3.3. Aplicación de técnicas de validación de requisitos y utilización de métricas

3.3.1. Validación de requisitos

El proceso de validación de requisitos debe realizarse o de lo contrario se corre el riesgo de implementar una mala especificación, con el costo que eso conlleva. (Sommerville, 2005) Los tres métodos utilizados son:

Revisión técnica formal (RTF): Tiene el objetivo de verificar la coherencia de la documentación con los requisitos. Una vez aplicada se descubrieron que se redactó de forma incorrecta cinco requisitos en una primera iteración, los que fueron solucionados para la segunda.

Prototipado: Se construyeron los prototipos de interfaz de usuario que permitieron al cliente hacerse una idea de la estructura de la interfaz y comprobar si estaban correctos. Se le explicó toda la información que no se encontraba explícita en el prototipo. Se detectó que había información que no estaba ubicada en el lugar adecuado corrigiéndose los problemas señalados

Generación de diseños de casos de prueba: Tiene como objetivo comprobar la verificabilidad de los requisitos. Se define para cada requisito funcional describiendo perfectamente los datos de entrada, las tareas a realizar y los resultados esperados. (Sommerville, 2005)

Como resultado se obtuvo el acta de aceptación de requisitos.

3.3.2. Métricas para evaluar el diseño propuesto

Capítulo 3: Implementación y prueba

El Standard Glossary of Software Engineering Terms (IEEE) define métrica como una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado. Una vez realizado el diseño del software permiten al ingeniero evaluar la calidad antes de iniciar la implementación del sistema.

Lorenz y Kidd para medir las características estáticas del diseño del software dividen las métricas basadas en clases en cuatro categorías, cada una con un diseño a nivel de componentes: tamaño, herencia, valores internos y valores externos. De esta manera, estas métricas contribuyen a la mantenibilidad de los productos de software.

Aplicación de la métrica Tamaño operacional de clase (TOC):

Esta métrica está dada por el número de métodos asignados a una clase, se clasifica en baja, media, alta y evalúa los siguientes atributos de calidad:

Responsabilidad: Un aumento del TOC implica un aumento de la responsabilidad asignada a la clase.

Complejidad de implementación: Un aumento del TOC implica un aumento de la complejidad de implementación de la clase.

Reutilización: Un aumento del TOC implica una disminución del grado de reutilización de la clase.

Tabla 7 Rango de valores para los criterios de evaluación de la métrica TOC.

Atributo	Clasificación	Criterio
Responsabilidad	Baja	\leq Promedio.
	Media	Entre Promedio y $2 \times$ Promedio.
	Alta	$> 2 \times$ Promedio.
Complejidad de implementación	Baja	\leq Promedio.
	Media	Entre Promedio y $2 \times$ Promedio.
	Alta	$> 2 \times$ Promedio.
Reutilización	Baja	$> 2 \times$ Promedio.
	Media	Entre Promedio y $2 \times$ Promedio.
	Alta	\leq Promedio.

Tabla 8 Resultados de la evaluación de la métrica TOC.

Clase	Cantidad de	Responsabilidad	Complejidad	Reutilización
-------	-------------	-----------------	-------------	---------------

Capítulo 3: Implementación y prueba

	Procedimientos			
GestionarpersonasController	21	Alta	Alta	Baja
RecuperacionespersonaController	6	Media	Media	Media
DatCategoriaModel	2	Baja	Baja	Alta
DatEscuelaConduccionModel	3	Baja	Baja	Alta
DatFotoModel	3	Baja	Baja	Alta
DatHerramientaTrabajadorModel	4	Baja	Baja	Alta
DatTrabajadorCategoriaModel	3	Baja	Baja	Alta
DatTrabajadorModel	9	Alta	Alta	Baja
DatTrabajadorTipoSancionModel	4	Baja	Baja	Alta
DatEscuelaConduccion	1	Baja	Baja	Alta
DatFoto	1	Baja	Baja	Alta
DatHerramientaTrabajador	3	Baja	Baja	Alta
DatTrabajador	11	Alta	Alta	Baja
DatTrabajadorCategorialicencia	2	Baja	Baja	Alta
DatTrabajadorTipoSancion	2	Baja	Baja	Alta
NomCategorialicencia	1	Baja	Baja	Alta
NomTipoSancion	1	Baja	Baja	Alta

Partiendo del análisis de los resultados obtenidos al aplicar la métrica TOC se obtuvo un promedio de procedimientos de 4.52 para un total de 17 clases y 77 procedimientos. La mayoría de las clases poseen entre 1 y 5 procedimientos. (Figura 19) El diseño propuesto posee una elevada reutilización ya que el 76% de las clases se clasifican en altas. En cuanto a complejidad de implementación y responsabilidad el 76% de las clases son bajas lo que indica que se necesitarán pocos cambios en caso de usarlas en otro sistema y la información que manejan está de acuerdo con las responsabilidades asignadas. (Figura 20) Por lo que se concluye que los resultados obtenidos según esta métrica son positivos.

Capítulo 3: Implementación y prueba

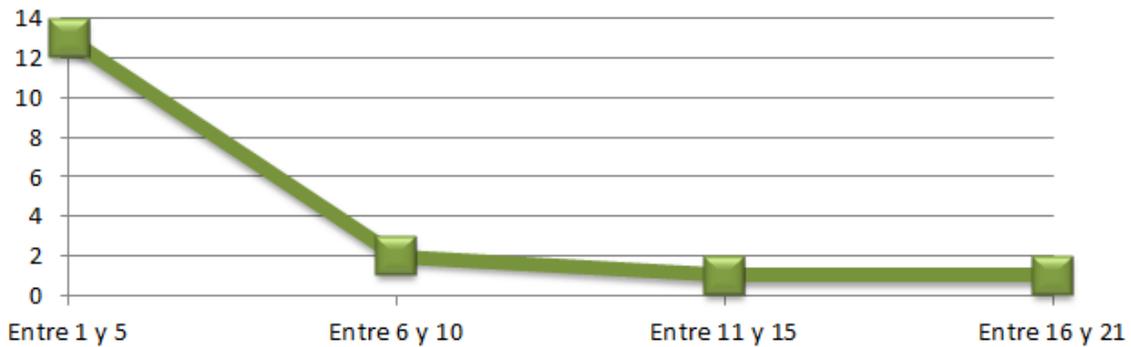


Figura 19 Cantidad de procedimientos por clase

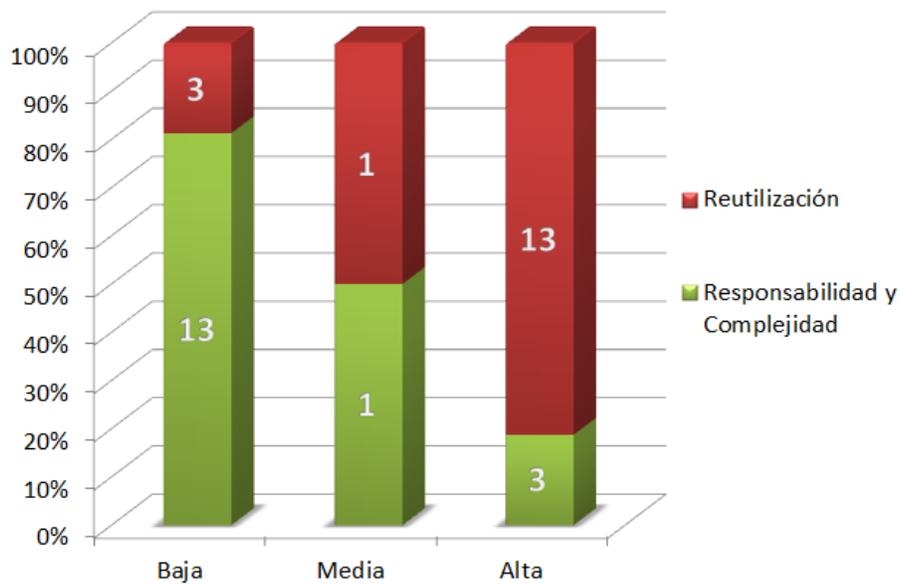


Figura 20 Clasificación de las clases por indicadores

Aplicación de la métrica Relaciones entre clases (RC):

Está dada por el número de relaciones de una clase con otra, se clasifica en ninguna, baja, media, alta y evalúa los siguientes atributos de calidad:

Acoplamiento: Un aumento del RC implica un aumento del Acoplamiento de la clase.

Complejidad de mantenimiento: Un aumento del RC implica un aumento de la complejidad del mantenimiento de la clase.

Reutilización: Un aumento del RC implica una disminución en el grado de reutilización de la clase.

Capítulo 3: Implementación y prueba

Cantidad de pruebas: Un aumento del RC implica un aumento de la Cantidad de pruebas de unidad necesarias para probar una clase.

Tabla 9 Rango de valores para los criterios de evaluación de la métrica RC.

Atributo	Categoría	Criterio
Acoplamiento	Ninguno	0
	Bajo	1
	Medio	2
	Alto	>2
Complejidad de Mantenimiento	Baja	\leq Promedio.
	Media	Entre Promedio y 2^* Promedio.
	Alta	$> 2^*$ Promedio.
Reutilización	Baja	$>2^*$ Promedio.
	Media	Entre Promedio y 2^* Promedio.
	Alta	\leq Promedio.
Cantidad de Pruebas	Baja	\leq Promedio.
	Media	Entre Promedio y 2^* Promedio.
	Alta	$> 2^*$ Promedio.

Tabla 10 Resultados de la evaluación de la métrica RC

Acoplamiento	Cantidad de clases	Promedio
Ninguno	8	47
Bajo	7	41
Medio	0	0
Alto	2	12
Complejidad de Mantenimiento	Cantidad de clases	Promedio
Baja	15	88
Media	0	0
Alta	2	12
Cantidad de Pruebas	Cantidad de clases	Promedio
Baja	15	88
Media	0	0
Alta	2	12
Reutilización	Cantidad de clases	Promedio
Baja	2	12

Capítulo 3: Implementación y prueba

Media	0	0
Alta	15	88

Una vez aplicada la métrica RC se obtuvo un promedio de 1,23 relaciones de uso entre las clases. En cuanto al acoplamiento el 47% y 41% se clasifican en ninguno y bajo respectivamente, demostrando que las clases usan pocos métodos de las otras. Los atributos complejidad de mantenimiento y cantidad de pruebas alcanzaron un 88% en la categoría baja, lo que demuestra que no es necesario un elevado esfuerzo a la hora de realizar modificaciones, reparaciones y pruebas al software. La reutilización de las clases es favorable ya que se obtuvo un 88% de ellas en la clasificación de alta.

3.4. Pruebas de software

Las pruebas de software sirven para evaluar la calidad de un producto software o para mejorarlo, mediante la identificación de sus defectos y problemas. Consiste en la verificación dinámica del comportamiento real de un programa frente al comportamiento esperado, para un conjunto finito de casos de prueba (convenientemente seleccionados entre las usualmente infinitas posibilidades de ejecución). (Ruiz, 2009)

Con el objetivo de lograr un producto con calidad se verifica que la implementación esté en concordancia con los requisitos identificados. Esto permite descubrir errores que luego son corregidos. Las pruebas realizadas a la solución se encuentran en los niveles de unidad e integración, las cuales se describen a continuación.

3.4.1. Pruebas de unidad

Las pruebas de unidad están enfocadas a aislar cada parte del programa y mostrar que es correcto. Se utilizó para probar cada módulo de forma independiente y los métodos de prueba utilizados son pruebas de caja blanca y pruebas de caja negra.

Pruebas de Caja Blanca

La prueba de caja blanca se centra en el estudio minucioso de la operatividad de una parte del sistema considerando los detalles procedurales. (Departamento de Informática, 2007) Esta prueba permite garantizar que en el código fuente:

- Se ejercita por lo menos una vez todos los caminos independientes de cada módulo.
- Se ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa.
- Se ejecuten todos los bucles en sus límites operacionales.

Capítulo 3: Implementación y prueba

- Se ejerciten las estructuras internas de datos para asegurar su validez.

Entre las técnicas para obtener casos de prueba de caja blanca se encuentra la del camino básico. Este método permite obtener una medida de la complejidad lógica de un diseño procedimental que posibilita definir un conjunto básico de caminos de ejecución. Para la obtención de la complejidad ciclomática se emplea una representación del flujo de control en forma de grafo. (Departamento de Informática, 2007)

La complejidad ciclomática define el número de caminos independientes de un programa. Por camino independiente se entiende aquel que introduce una arista que no haya sido recorrida antes. Da una cota o límite superior para el número de casos de prueba necesarios para ejercitar cada sentencia del código. (Departamento de Informática, 2007)

Desarrollo del método:

Se cuenta con el requisito de alta complejidad Listar personas para consumir el servicio de la UCI del que se muestra el camino del método listadoUCI, el cual es parte de la ejecución de este requisito.

El siguiente fragmento de código corresponde al método:

```
function listadoUCI(){
    $cache = ZendExt_Cache :: getInstance(); //1
    $personas = $cache->load('personas');
    if (null == $personas) {//2
        if( fopen("http://assets.uci.cu/servicios/v4/AssetsWS.wsdl", "r")){//3
            $SoapClientPersonas = new SoapClient("http://assets.uci.cu/servicios/v4/AssetsWS.wsdl");//4
            $personas = $SoapClientPersonas->ObtenerPersonas();
            $cache->save($personas, 'personas');
        }
    }
} //5
```

Figura 21 Código del método listadoUCI.

Dado el código se hace el grafo que se muestra en la figura siguiente:



Figura 22 Grafo del método listadoUCI.

Capítulo 3: Implementación y prueba

Se calcula la complejidad ciclomática utilizando las fórmulas que se muestran a continuación, siendo A la cantidad de aristas, N la cantidad de nodos, P la cantidad de nodos predicados y R la cantidad de regiones contenidas en el grafo.

$$V(G) = A - N + 2, V(G) = 6 - 5 + 2, V(G) = 3$$

$$V(G) = P + 1, V(G) = 2 + 1, V(G) = 3$$

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

Para cada una de ellas el resultado es de 3, determinando la cantidad de casos de prueba.

Camino 1: 1-2-3-4-5

Camino 2: 1-2-3-5

Camino 3: 1-2-5

Caso de prueba para el camino básico # 1

Camino: 1-2-3-4-5

Descripción: Está disponible el servicio del Sistema de Capital Humano.

Los datos de entrada cumplirán con el siguiente requisito:

El parámetro \$personas está vacío.

Entrada: \$personas = [].

Resultados esperados: Se espera que cree un archivo caché de personas con el listado de trabajadores plantilla de la UCI en el directorio /cache.

Resultados obtenidos: Satisfactorio.

Caso de prueba para el camino básico # 2

Camino: 1-2-3-5

Descripción: No está disponible el servicio del Sistema de Capital Humano.

Los datos de entrada cumplirán con los siguientes requisitos:

El parámetro \$personas está vacío.

Entrada: \$personas = [].

Capítulo 3: Implementación y prueba

Resultados esperados: Se espera que termine el flujo sin crear un archivo caché de personas en el directorio /cache.

Resultados obtenidos: Satisfactorio.

Caso de prueba para el camino básico # 3

Camino: 1-2-5

Descripción: Los datos de entrada cumplirán con los siguientes requisitos:

El parámetro \$personas no puede estar vacío.

Entrada: \$personas = [a:4731:{i:0; O:8:"stdClass":17:{s:9:"Apellidos"; s:15:"Alfonso Francis"; s:4:"Area";O:8:"stdClass":2:{s:6:"IdArea";s:6:"063000";s:10:"NombreArea"; s:14:"Dir.Transporte"; } s:5:"Cargo"; O:8:"stdClass":2:{s:7:"IdCargo";s:4:"8055";s:11:"NombreCargo"; s:10:"Chofer "D"; } s:2:"CI"; s:11:"56122113727"; s:9:"Direccion"; s:56:"Calle 3ra. # 214 e/ F y D, Rpto. Rosario,Arroyo Naranjo"; s:6:"EsAlta"; b:1; s:6:"EsBaja"; b:0; s:9:"EsDocente"; b:0; s:9:"FechaAlta"; s:19:"2002-09-02T00:00:00"; s:15:"FechaNacimiento"; s:19:"1956-12-21T00:00:00"; s:12:"IdExpediente"; s:5:"06014";s:9:"Militante";i:2; s:9:"Municipio"; O:8:"stdClass" :3:{s:11:"IdMunicipio"; s:4:"2314"; s:15:"NombreMunicipio"; s:14:"ARROYO NARANJO";s:9:"Provincia";O:8:"stdClass":2:{s:11:"IdProvincia";s:2:"23";s:15:"NombreProvincia" ; s:9:"LA HABANA";}}s:16:"NivelEscolaridad"; s:44:"Educación Técnica Profesional Nivel Medio "; s:7:"Nombres"; s:15:"José Celestino";s:4:"Sexo"; b:1; s:8:"Telefono"; s:1:" ";}] .

Resultados esperados: Se espera que termine el flujo sin crear un archivo caché de personas en el directorio /cache.

Resultados obtenidos: Satisfactorio.

Después de aplicar los casos de prueba la respuesta de las funcionalidades coincidió con los resultados esperados. Esto demuestra que internamente funciona de forma correcta la solución. Permitted chequear que son ejecutadas las diferentes vertientes del código.

Pruebas de Caja Negra

Las pruebas de caja negra consideran la función específica para la cual fue creado el producto, es decir lo que hace. Se llevan a cabo sobre la interfaz del sistema y el número de casos de prueba se reduce mediante la elección de condiciones de entrada y salida válidas y no válidas que ejercitan toda la funcionalidad. (Departamento de Informática, 2007)

Capítulo 3: Implementación y prueba

Para el desarrollo de dicha prueba se aplicó la técnica de partición de equivalencia que consiste en determinar a cada elemento de entrada los valores válidos y no válidos. Fueron generados un total de 27 diseños de casos de prueba.

Se necesita que un usuario se identifique y autentique ante el sistema, además de tener los permisos para ejecutar esta acción. Se debe seleccionar en el menú **Mantenimiento/Recursos Humanos/Registro de personas**. Se debe presionar el botón adicionar y una vez se muestre la ventana, presionar el botón buscar persona. A continuación se muestran los escenarios de prueba para el requisito Listar personas para consumir el servicio de la UCI.

Tabla 11 Requisito a probar Listar personas para consumir el servicio de la UCI.

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1: Listar persona para consumir el servicio de la UCI.	El sistema debe mostrar el listado de las personas.	EP 1.1: Listar persona con el servicio de la UCI disponible.	– NA
		EP 1.2: Listar persona con el servicio de la UCI no disponible.	– Se muestra un mensaje informando del error. – Se presiona el botón Aceptar .

Tabla 12 Juego de datos a probar

Id del escenario	Escenario	Respuesta del sistema	Resultado de la prueba
EP 1.1	EP 1.1: Listar persona con el servicio de la UCI disponible.	El sistema muestra un listado de las personas.	NA
EP 1.2	EP 1.2: Listar persona con el servicio de la UCI no disponible.	El sistema muestra el mensaje de error: "No está disponible la lista de trabajadores del Sistema Capital Humano. Introduzca los datos manualmente." El sistema mantiene la interfaz abierta.	NA

Capítulo 3: Implementación y prueba

Los diseños de casos de prueba se encuentran en los documentos CIG-CFM-N-MTTO-i5126.doc, CIG-CFM-N-MTTO-i5144.doc, CIG-CFM-N-MTTO-i5143.doc, CIG-CFM-N-MTTO-i5142.doc, CIG-CFM-N-MTTO-i5145.doc, CIG-CFM-N-MTTO-i5128.doc, CIG-CFM-N-MTTO-i5130.doc, CIG-CFM-N-MTTO-i5131.doc, CIG-CFM-N-MTTO-i5129.doc, CIG-CFM-N-MTTO-i5132.doc, CIG-CFM-N-MTTO-i51130.doc, CIG-CFM-N-MTTO-i5134.doc, CIG-CFM-N-MTTO-i5137.doc, CIG-CFM-N-MTTO-i5135.doc, CIG-CFM-N-MTTO-i5136.doc, CIG-CFM-N-MTTO-i5133.doc, CIG-CFM-N-MTTO-i5138.doc, CIG-CFM-N-MTTO-i5140.doc, CIG-CFM-N-MTTO-i5141.doc, CIG-CFM-N-MTTO-i5139.doc, CIG-CFM-N-MTTO-i5127.doc, CIG-CFM-N-MTTO-notificaciones-A.doc, CIG-CFM-N-MTTO-notificaciones-C.doc, CIG-CFM-N-MTTO-notificaciones-E.doc, CIG-CFM-N-MTTO-notificaciones-M.doc y CIG-CFM-N-MTTO-notificaciones-S.doc.

Tabla 13 Total de NC detectadas en las pruebas de calidad interna.

Módulo	Iteración 1	Iteración 2	Iteración 3
Recursos humanos	16	2	0
Taller	2	0	0

A los módulos Taller y Recursos humanos se le realizaron pruebas internas donde las NC identificadas estaban relacionadas con la validación de los datos que se introducen en el sistema y nombres de interfaces incorrectos. Para la segunda iteración estos errores fueron corregidos sin embargo se detectaron 2 nuevas NC de faltas de ortografía. En la tercera iteración todas fueron resueltas demostrando que la aplicación funciona correctamente.

Luego fue probado el módulo Recursos humanos por la entidad Calisoft, donde fue comprobado el correcto funcionamiento del mismo a partir de los diseños de los casos de prueba con cero NC.

3.4.2. Pruebas de integración

Utilizada para probar la combinación de la aplicación con otros sistemas y determinar si funcionan correctamente en conjunto. Fue aplicada la técnica de integración incremental ascendente de Órbita con el Sistema de Capital Humano e identificación.

Los RNF del sistema Órbita ya fueron validados por el arquitecto del proyecto, sin embargo fueron agregados para el desarrollo de la solución cuatro que resonden al atributo de calidad de interoperabilidad. Con el fin de validar los nuevos RNF se utilizaron los escenarios que se describen a continuación como parte de la integración entre los sistemas.

Capítulo 3: Implementación y prueba

Nombre del Escenario: Consumir servicios web del Sistema de Capital Humano.

- **Objetivo del negocio:** Probar la interoperabilidad del sistema Órbita con el servicio web del Sistema de Capital Humano.
- **Descripción:** Es la capacidad del sistema para poder obtener la colección de objetos Persona con todos los trabajadores que conforman la plantilla de la UCI Sede Central. Un usuario accede al módulo Recursos Humanos del sistema Órbita. Se invocará la llamada del servicio a través de la funcionalidad Registro de personas.
- **Ambiente:** Plataforma del sistema Órbita y del Directorio de Servicios Web UCI disponibles, debe de existir conexión de red y usuario con permisos.
- **Respuesta:** Si está disponible el servicio se guarda en la caché el listado de trabajadores, si no debe mostrar un mensaje de error donde informa al usuario que debe introducir los datos manualmente.
- **Medida de la Respuesta:** Tiempo máximo de 5 segundos.
- **Atributo de calidad que impacta:** Interoperabilidad.

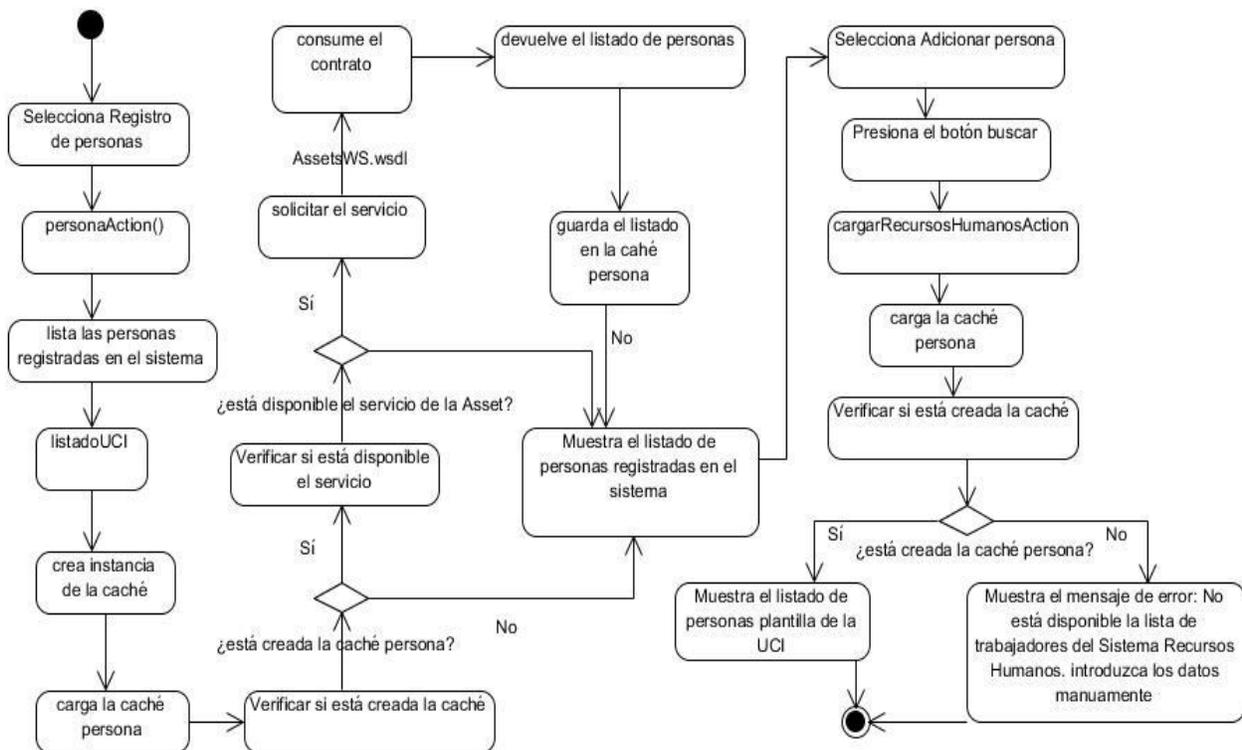


Figura 23 Descripción del escenario Consumir servicios web del Sistema de Capital Humano.

Capítulo 3: Implementación y prueba

Nombre del Escenario: Consumir servicio web de identificación.

- **Objetivo del negocio:** Probar la interoperabilidad del sistema Órbita con el servicio web de identificación.
- **Descripción:** Es la capacidad del sistema para poder obtener la foto de los trabajadores que conforman la plantilla de la UCI Sede Central. Un usuario accede al módulo Recursos Humanos del sistema Órbita y luego a la funcionalidad Registro de personas. Después de la búsqueda del listado de trabajadores de la UCI en Adicionar persona, se selecciona la persona y se presiona aceptar donde se invocará la llamada del servicio.
- **Ambiente:** Plataforma del sistema Órbita y del Directorio de Servicios Web UCI disponibles, debe de existir conexión de red y usuario con permisos.
- **Respuesta:** Si el servicio está disponible se muestra la foto del trabajador, si no la predeterminada.
- **Medida de la Respuesta:** Tiempo máximo de 5 segundos.
- **Atributo de calidad que impacta:** Interoperabilidad.

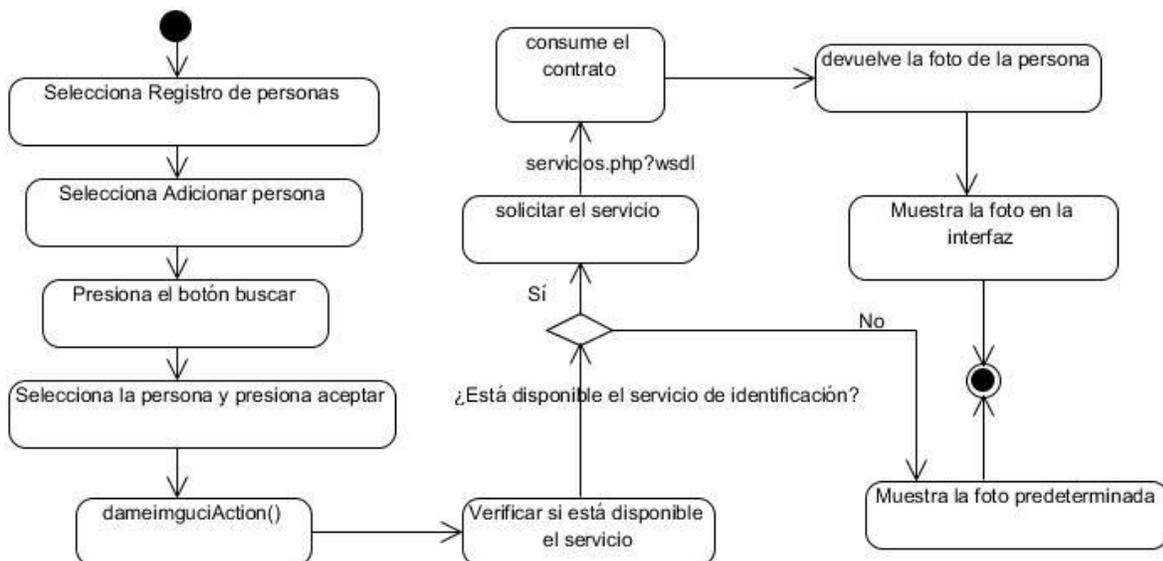


Figura 24 Descripción del escenario Consumir servicio web de identificación.

Nombre del Escenario: Consumir servicio web para el envío de notificaciones.

- **Objetivo del negocio:** Probar la interoperabilidad del sistema Órbita.

Capítulo 3: Implementación y prueba

- **Descripción:** Es la capacidad del sistema Órbita para brindar el servicio de envío de notificaciones. El cliente solicita el servicio que se encuentra publicado en el sistema Órbita.
- **Ambiente:** Debe de existir conexión de red, el cron debe estar funcionando con la tarea de ejecución programada, el wsdl debe encontrarse público y debe existir el archivo prueba_service.php.
- **Respuesta:** Se muestra en el sistema un mensaje de cada notificación.
- **Medida de la Respuesta:** Tiempo máximo de 5 segundos si el usuario está autenticado en el sistema, si no, cuando inicia sesión en el sistema.
- **Atributo de calidad que impacta:** Interoperabilidad.

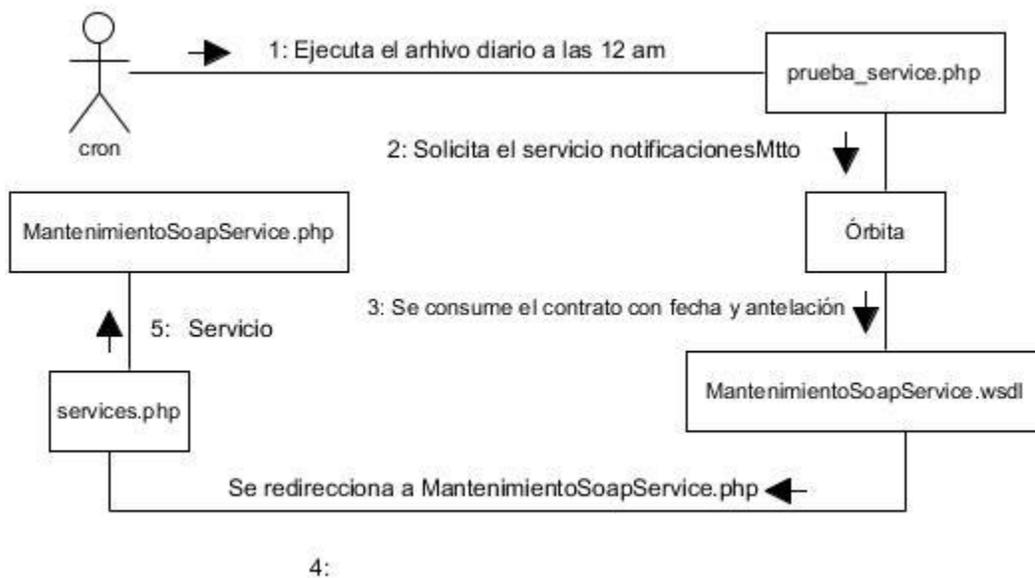


Figura 25 Descripción del escenario Consumir servicio web para el envío de notificaciones.

Nombre del Escenario: Consumir servicio web para el envío de notificaciones.

- **Objetivo del negocio:** Probar la interoperabilidad del sistema Órbita.
- **Descripción:** Es la capacidad del sistema Órbita para brindar el servicio de envío de notificaciones. El cliente solicita el servicio que se encuentra publicado en el sistema Órbita.

Capítulo 3: Implementación y prueba

- **Ambiente:** Debe de existir conexión de red, el cron debe estar funcionando con la tarea de ejecución programada, el wsdl debe encontrarse público y debe existir el archivo prueba_service.php.
- **Respuesta:** Se recibe el correo electrónico de cada notificación.
- **Medida de la Respuesta:** Tiempo máximo de 5 segundos si el usuario está autenticado en el sistema, si no, cuando inicia sesión en el sistema.
- **Atributo de calidad que impacta:** Interoperabilidad.

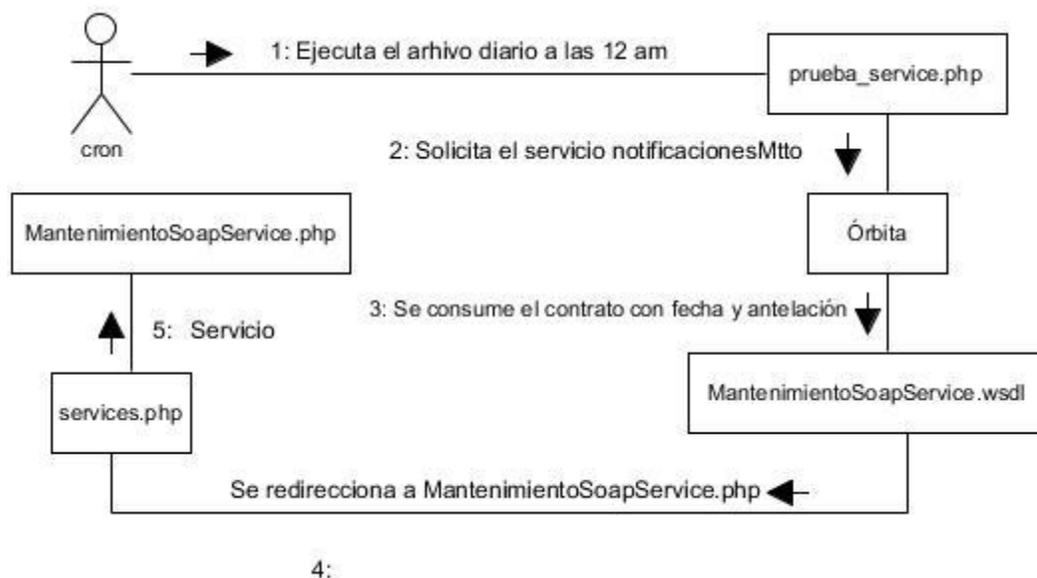


Figura 26 Descripción del escenario Consumir servicio web para el envío de notificaciones.

Una vez aplicados los escenarios de prueba se identificaron 7 NC las cuales fueron resueltas y en la segunda iteración ya no se encontraron NC, por lo tanto las pruebas de integración fueron realizadas satisfactoriamente.

3.5. Validación de las variables

Con el desarrollo de mejoras funcionales a los módulos Recursos humanos y Taller se satisfacen las necesidades de integración y notificación de la Dirección de Transporte de la UCI en cuanto a la gestión del mantenimiento de los vehículos y de los recursos humanos. Esto se evidencia con los elementos que a continuación se describen.

- Errores en el registro de los datos

Capítulo 3: Implementación y prueba

Antes de contar con el sistema Órbita la información de los trabajadores se registraba de forma manual por lo que los datos eran propensos a los errores que pudiera cometer el especialista y no percatarse en ese momento. Proceso que traía consigo que fuese necesario verificar la información que está registrada en la Dirección de Capital Humano dirigiéndose hasta su oficina.

Usando el módulo Recursos humanos de Órbita se notifica al usuario sobre los errores cometidos cuando introduce los datos. Se le muestran mensajes que incluyen sugerencias de las posibles soluciones. Hasta que la información no sea correcta no permite registrarla en el sistema. Mediante el uso del sistema se realiza el registro de los datos de los trabajadores y permite asimilar la información existente en el Sistema de Capital Humano. El especialista debe autenticarse en el sistema, acceder al menú Recursos humanos y dentro de este a Registro de personas. Se presiona el botón adicionar, se muestra la ventana y se hace la búsqueda de personas. Esta funcionalidad permite mostrar el listado de trabajadores de la UCI y seleccionar el que se desea registrar en el sistema presionando el botón aceptar.

- Los mantenimientos sorprenden a los involucrados

Antes en el taller la información se almacenaba en documentos y hojas de cálculo por lo que en muchas ocasiones no se conocía con tiempo suficiente cuando debía realizarse un mantenimiento ni quiénes eran los involucrados. En el periodo del 1 al 30 de abril se identificaron un total de 3 mantenimientos que no se realizaron en la fecha correspondiente porque los mecánicos no estaban al tanto de esa actividad.

Mediante el uso de las notificaciones en el periodo del 1 al 31 de mayo fue posible informar con anterioridad sobre 8 mantenimientos que debían realizarse.

- Identificación tardía del vencimiento de las recalificaciones, chequeos médicos, somatones y autorizos de parqueo.

Antes se estaba al tanto del vencimiento de la escuela de recalificación y el chequeo médico de los conductores cuando expiraban. De igual forma ocurría con los somatones y autorizos de parqueo de los vehículos. En el periodo del 1 al 30 de abril se identificaron un total de 7 eventos de los que no se tuvo conocimiento previo de su vencimiento.

Mediante el uso de las notificaciones en el periodo del 1 al 31 de mayo fue posible enviar una alerta a los usuarios correspondientes ante la proximidad del vencimiento de 5 eventos relacionados con las recalificaciones, chequeos médicos, somatones y autorizos de parqueo.

Capítulo 3: Implementación y prueba

Por los elementos antes expuestos se demuestra el cumplimiento del problema de investigación planteado, constatando que con el desarrollo de las mejoras funcionales a los módulos Taller y Recursos Humanos realizado, se satisfacen las necesidades de la Dirección de Transporte de la UCI en cuanto a la gestión del mantenimiento de los vehículos y de los recursos humanos.

3.6. Conclusiones parciales

Se describió el proceso de implementación, enfatizando en el desarrollo de las mejoras funcionales y su funcionamiento. La aplicación de pruebas permitió corroborar que la aplicación tiene un correcto funcionamiento, en correspondencia con el diseño elaborado y las necesidades identificadas.

Conclusiones

CONCLUSIONES

El desarrollo de mejoras funcionales a los módulos Taller y Recursos humanos permitió suplir las necesidades de integración y notificación de la Dirección de Transporte de la UCI en cuanto a la gestión del mantenimiento de los vehículos y de los recursos humanos. Fue posible puesto que:

- Se estudiaron algunos software relacionados con los sistemas de control de flota y mantenimiento decidiéndose realizar mejoras funcionales a los módulos Taller y Recursos humanos de Órbita.
- El uso del modelo de desarrollo junto a las tecnologías y herramientas permitió obtener un producto que cumpla con el principio de independencia tecnológica.
- Se realizó el análisis y diseño de las mejoras funcionales lo cual facilitó el proceso de implementación.
- Se implementaron las funcionalidades identificadas lo que permitió la gestión de los datos de los trabajadores del área de transporte de UCI.
- Se pudo establecer una sincronización entre los datos primarios de los trabajadores del área de transporte de la UCI, registrados en el Sistema de Capital Humano, con los datos correspondientes a estas mismas personas en el sistema Órbita.
- Se logró la integración del componente Notificaciones con Órbita generándose una guía para su instalación.
- Se incluyó las notificaciones automáticas, lo que posibilitó su envío diariamente.
- Con la aplicación de las pruebas al software se demostró que la implementación está en correspondencia con el diseño realizado y funciona de forma correcta.

Recomendaciones

RECOMENDACIONES

Con el fin de incrementar las funcionalidades de Órbita se recomienda como mejoras a la solución:

- Realizar la gestión de la recepción de las notificaciones a nivel de usuario.
- Ampliar la elicitación de requisitos de las notificaciones hacia el resto de los componentes del sistema.
- Si se necesitara gestionar más información de las personas debe valorarse la posibilidad de dividirla en varias ventanas para disminuir la carga de datos.

BIBLIOGRAFÍA

- ADELSTEIN, Tom y LUBANOVIC, Bill. 2007.** *Administración de sistemas Linux*. 2007.
- Alejandro Lagos. 2007.** Control de Mantenimiento de Flota y Maquinaria. [En línea] 2007. [Citado el: 15 de enero de 2014.] <http://ges-maq.blogspot.com/2007/08/sistema-de-control-y-gestion-gesmaq-r.html>.
- Baryolo, Oiner Gómez, Borbón, Yoandry Morejón y Tejo, Darien Garcia. 2010.** ARQUITECTURA TECNOLÓGICA PARA EL DESARROLLO DE SOFTWARE. Ciudad de la Habana : s.n., 2010.
- C.W., Gits. 1992.** *Design of maintenance concepts*. s.l. : Int. Journal of Production Economics, 24, pp.217-226., 1992.
- CEIGE. 2012.** *Estándares de codificación Proyectos con el marco de trabajo Sauxe del CEIGE*. 2012.
- . **2013.** *Modelo de desarrollo de software*. 2013.
- DATYS. 2011.** DATYS Tecnología y Sistemas. *DATYS Tecnología y Sistemas*. [En línea] 2011. [Citado el: 5 de diciembre de 2013.] <http://www.datys.cu/>.
- del Consejo de Estados y de Ministros, Comité Ejecutivo. 16 de agosto del 2007.** Decreto No. 281. *Reglamento para la implantación y consolidación del Sistema de Dirección y Gestión Empresarial Estatal*. 16 de agosto del 2007.
- Departamento de Informática. 2007.** *Tema 7: Validación*. s.l. : Universidad de Valladolid, 2007.
- Desoft. 2013.** Sgestman. *Sgestman*. [En línea] Desoft, 2013. [Citado el: 3 de diciembre de 2013.] <http://www.sgestman.cu/>.
- Digital Express. 2013.** Alamo Software para Transporte, Logística y Mantenimiento de flotas. *Alamo Software para Transporte, Logística y Mantenimiento de flotas*. [En línea] digital express, 2013. <http://www.mialamo.com.ar>.
- Eugcom. 2011.** *Descripción del producto, Sistema de Flotas Version 1.2.209*. Santiago : s.n., 2011.
- Ferré Grau, Xavier y Sánchez Segura, María Isabel.** Desarrollo Orientado a Objetos con UML. s.l. : Facultad de Informática – UPM.
- FLANAGAN, David. 2002.** *JavaScript: The definitive guide*. s.l. : O'Reilly Media, Inc, 2002.
- Fowler, Martin. 1996.** *“Analysis Patterns: Reusable Object Models”*. s.l. : Addison-Wesley Professional, 1996.
- Garret, Jesse James. 2005.** 18 de febrero de 2005.
- GERARDO, GABRIEL SILVA BAUTISTA.** Sistema de Gestión de Flota Vehicular. [En línea] http://www.uruman.org/material_tecnico/X%20Autores/GSilva.pdf.
- Gómez Pérez, Erich Mario, y otros. Septiembre 2012.** *Desarrollo de Aplicaciones Informáticas en la sociedad*. Ciudad de la Habana : s.n., Septiembre 2012.
- Lopez, Felipe. 2013.** Tutorial de PostgreSQL, 9.1.0. [En línea] 2013. [Citado el: 9 de 12 de 2013.] <http://pgsqtutorial.readthedocs.org>.
- Lorenzo Castro, Fernando . 1999-2000.** Modelo de datos, Conceptos y clasificación. 1999-2000.
- Máximo, Grupo. 2013.** TranposrteX. *TranposrteX-Software de Gestión para Empresas de*. [En línea] Grupo Máximo, 2013. [Citado el: 7 de noviembre de 2013.] <http://www.transportex.net>.
- Moffitt, Jack. 2010.** XMPP Programming with Javascript® and jQuery. Indianapolis : s.n., 2010. 978-0-470-54071-8..
- Nicola Asuni - Tecnick.com LTD . TCPDF.** [En línea] 2013. [Citado el: 8 de diciembre de 2013.] <http://www.tcpdf.org/>.
- Object Management Group. 2001.** OMG Unified Modeling Language Specification. [En línea] 2001. [Citado el: 17 de abril de 2014.] <http://www.omg.org>.

Bibliografía

- phpdoctrine.org** . Doctrine. [En línea] [Citado el: 8 de diciembre de 2013.]
<http://www.phpdoctrine.org>.
- Prando, Raul R. 2003.** *Manual de Gestión de Mantenimiento a la Medida*. Piedra Santa, 1996 : Publisher, 2003. ISBN, 8483773996, 9788483773994.
- Pressman, R. Vol. 1. 2005.** *Ingeniería de Software, un enfoque práctico*. La Habana, Cuba. : E.F. Varela, Vol. 1. 2005.
- REALTIME, Ignite. 2008.** Openfire Server. [En línea] 2008. [Citado el: 22 de febrero de 2014.]
<http://www.igniterealtime.org/projects/openfire>.
- Ruiz, Francisco. 2009.** CTR - Computadores y Tiempo real. Universidad de Cantabria.
INGENIERÍA DEL SOFTWARE I. [En línea] 2009. [Citado el: 24 de abril de 2014.]
<http://www.ctr.unican.es/asignaturas/is1/is1-t01-trans.pdf>.
- Soft, Tinita. 2012.** Software de Mantenimiento de Vehículos. *Software de Vehículos Pro V 6.10*. [En línea] Tinita Soft, 2012. [Citado el: 07 de Noviembre de 2013.]
<http://www.tinitasoft.com/programavehi.htm>.
- Sommerville, Ian. 2005.** *Ingeniería del Software*. s.l. : Miguel Martín Romo , 2005.
- The PHP Group. 2013.** PHP: Hypertext Preprocessor. [En línea] 2013. [Citado el: 8 de diciembre de 2013.] <http://php.net/>.
- W3Schools. 1999-2014.** SOAP Introduction. *W3Schools.com*. [En línea] 1999-2014. [Citado el: 11 de abril de 2014.] http://www.w3schools.com/Webservices/ws_soap_intro.asp.
- Werdmuller, Ben. 2010.** Build a web-based notification tool with XMPP Write real-time web applications with XMPP, PHP, and JavaScript. 2010.
- Zend Technologies. 2006-2013.** Zend Framework. *Zend Framework*. [En línea] Zend Technologies Ltd, 2006-2013. [Citado el: 8 de diciembre de 2013.]
<http://framework.zend.com/manual/1.5/en/introduction.html>.

