



UNIVERSIDAD DE CHILE

FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

**SISTEMA DE GESTIÓN Y CICLO DE VIDA DE MATERIAL MAYOR PARA
BOMBEROS DE CHILE**

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

VIJAY KUMAR KHEMLANI COBAISE

ABRIL 2012



UNIVERSIDAD DE CHILE

FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

**SISTEMA DE GESTIÓN Y CICLO DE VIDA DE MATERIAL MAYOR PARA
BOMBEROS DE CHILE**

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

VIJAY KUMAR KHEMLANI COBAISE

PROFESOR GUÍA:

SERGIO OCHOA DELORENZI

MIEMBROS DE LA COMISIÓN:

ROMAIN ROBBES

SANDRA DE LA FUENTE GONZÁLEZ

SANTIAGO DE CHILE

ABRIL 2012

Resumen

Bomberos de Chile es una institución con una profunda significancia en nuestro país. Su labor en el combate de incendios y otros incidentes los pone constantemente en riesgo, sin recibir por ello remuneración alguna. A pesar de esto, los bomberos de Chile continuamente muestran una formación y temple que es ejemplo para otros países de la región, y que está a la altura de naciones desarrolladas.

Si bien esta organización muestra una capacidad técnica y operativa excelente, sus procesos de gestión aún distan mucho de ser los apropiados. Por ejemplo, no existen registros formales y actualizados de sus voluntarios, material menor (por ejemplo mangueras, pitones, etc.) y material mayor, correspondiente a su flota de vehículos.

Al no tener esta información, Bomberos opera con incerteza respecto a sus recursos, lo que aumenta los gastos para la organización y el riesgo para sus voluntarios. Por ejemplo, el no conocer el estado de mantenimiento de un carro bomba aumenta la probabilidad de que este tenga desperfectos técnicos graves y requiera reparaciones costosas.

Por otra parte, la reciente promulgación de la Ley Marco de Bomberos obliga a la institución a informar al Estado, de manera detallada, acerca de su patrimonio y operaciones; información que actualmente está incompleta y es engorrosa de recopilar.

Para apoyar a Bomberos en el cumplimiento de estas necesidades este trabajo de memoria ha desarrollado un sistema de información para la Junta Nacional de Cuerpos de Bomberos de Chile (JNCB). Dicho software es capaz de gestionar el ciclo de vida de todo el material mayor de la JNCB y los cuerpos bomberiles del país. Para cada vehículo, este ciclo incluye su dada de alta con sus características técnicas y documentación digitalizada, sus asignaciones a cuerpos y compañías, sus mantenciones técnicas, y eventualmente su dada de baja.

La funcionalidad principal del sistema es el hacer una revisión constante del estado de cada uno de los vehículos de la JNCB. Si el software detecta que le corresponde mantenimiento programado a alguno de estos carros, emite automáticamente las órdenes de mantención apropiadas. Dichas órdenes y su posterior ejecución son almacenadas permanentemente en la hoja de vida del vehículo para su futura referencia, manteniendo así un registro actualizado y permanente de toda la flota.

Este sistema, actualmente en producción, está siendo alimentado con la información de los cerca de 2.500 carros con los que cuenta Bomberos de Chile actualmente. Esto permitirá a la Junta Nacional de Bomberos hacer seguimiento de cada uno de ellos y, eventualmente, obtener reportes y estadísticas sobre esta información, apoyando con métricas concretas sus procesos de toma de decisiones.

A mi abuelito, a quien no alcance a cumplirle esta promesa

Agradecimientos

Esta memoria no habría sido posible sin las personas que de una u otra forma me han apoyado durante mi vida.

A mis padres y hermana, por su amor y paciencia infinitos con todas las horas que paso frente al computador y lejos de ellos, por su sacrificio constante que me permitió estudiar lo que más deseaba, y por su apoyo incondicional en todas las decisiones que he tomado en la vida.

A mis abuelos, por tomar una vez más el rol de padres y permitirme vivir con ellos durante seis años brindándome su cariño durante un proceso lleno de desafíos.

A mis amigos, que en el colegio me enseñaron que había cosas más allá del computador, y que en la universidad me motivan a ser mejor persona.

A mis profesores, en particular a mi profesor guía Sergio Ochoa, por su dedicación a la enseñanza, su compromiso con la formación de profesionales íntegros y su inspiración constante para aprender algo nuevo cada día.

Al equipo del Programa de Movilidad Estudiantil de la Universidad de Chile, por todo su apoyo en mi estadía en California, uno de los periodos de mayor claridad e inspiración de mi vida.

Finalmente, a mis mascotas, Lisa, Roni y Flucke, por el cariño y apego especial que tienen los perros y que me motivaron en los momentos más complicados de este viaje.

Contenido

| | | |
|--------|--|----|
| 1. | Introducción | 8 |
| 1.1. | Problema a Resolver | 9 |
| 1.2. | Objetivos de la Memoria | 12 |
| 2. | Requisitos de la Solución..... | 13 |
| 2.1. | Requisitos funcionales..... | 13 |
| 2.2. | Decisiones iniciales..... | 16 |
| 2.3. | Usuarios del sistema | 17 |
| 2.3.1. | Usuarios dentro de la JNCB | 17 |
| 2.3.2. | Usuarios dentro de los cuerpos de Bomberos | 18 |
| 2.4. | Casos de Uso | 19 |
| 2.5. | Procesos involucrados..... | 23 |
| 2.5.1. | Dar de alta material mayor | 23 |
| 3. | Diseño arquitectónico de la solución | 34 |
| 3.1. | Arquitectura física | 34 |
| 3.2. | Arquitectura del Proveedor de Servicios | 36 |
| 3.3. | Arquitectura de la aplicación de gestión de material mayor | 40 |
| 3.4. | Modelo de Datos..... | 41 |
| 4. | Diseño detallado de la aplicación de gestión de material mayor | 52 |
| 5. | Implementación de la solución | 55 |
| 5.1. | Implementación del Proveedor de servicios..... | 55 |
| 5.1.1. | Estadísticas de código | 56 |
| 5.2. | Implementación de la aplicación de gestión de material mayor | 56 |
| 5.2.1. | Estadísticas de código | 58 |
| 5.3. | Flujo de uso | 59 |
| 6. | Resultados esperados y obtenidos..... | 68 |
| 7. | Conclusiones y trabajo futuro | 69 |
| 8. | Bibliografía | 70 |

Listado de Ilustraciones

| | |
|--|----|
| Ilustración 1 - Ciclo de vida del material mayor de Bomberos de Chile | 10 |
| Ilustración 2 - Caso de Uso de Ingreso de Material Mayor | 19 |
| Ilustración 3 - Caso de Uso de Carga de pauta de mantenimiento de chasis o carrosado | 20 |
| Ilustración 4 – Caso de Uso de asignación de material mayor a cuerpo..... | 21 |
| Ilustración 5 - Caso de Uso de Asignación de material mayor a compañía | 22 |
| Ilustración 6 - Caso de Uso de Generación y resolución de órdenes de mantenimiento | 23 |
| Ilustración 7 - Proceso de Dar de alta material mayor..... | 24 |
| Ilustración 8 - Ejemplo de pauta de mantención | 25 |
| Ilustración 9 - Proceso de crear pauta de mantención | 26 |
| Ilustración 10 - Proceso de asignar y reasignar material mayor a cuerpo | 28 |
| Ilustración 11 - Proceso de asignar y reasignar material mayor a compañía | 30 |
| Ilustración 12 - Proceso de generar orden de mantención | 32 |
| Ilustración 13 - Proceso de resolver orden de mantención | 33 |
| Ilustración 14 - Arquitectura física | 36 |
| Ilustración 15 – Ejemplo de documento XML usado por la aplicación | 38 |
| Ilustración 16 - Descripción a alto nivel del Proveedor de Servicios..... | 38 |
| Ilustración 17 - Diseño interno del Proveedor de Servicios | 39 |
| Ilustración 18 - Arquitectura de la aplicación de gestión de material mayor | 40 |
| Ilustración 19 - Modelo de Datos de tablas básicas..... | 43 |
| Ilustración 20 – Modelo de datos de información técnica del material mayor | 45 |
| Ilustración 21 – Modelo de datos de información de adquisición del material mayor | 47 |
| Ilustración 22 - Modelo de datos de mantenciones de material mayor | 49 |
| Ilustración 23 - Modelo de datos de procesos secundarios..... | 51 |
| Ilustración 24 - Diseño detallado de la aplicación de gestión de material mayor | 53 |
| Ilustración 25 - Página de autenticación de usuarios..... | 59 |
| Ilustración 26 - Página de inicio del sistema | 60 |
| Ilustración 27 - Lista de pautas de mantención de carrosado antes de subir una nueva pauta | 60 |
| Ilustración 28 - Interfaz para agregar una nueva pauta de mantención de carrosado | 61 |
| Ilustración 29 - Lista de pautas de mantención de carrosado después de subir una nueva pauta | 61 |
| Ilustración 30 - Interfaz para editar pautas de mantención | 62 |
| Ilustración 31 - Interfaz para dar de alta un nuevo material mayor | 63 |
| Ilustración 32 - Navegador de material mayor | 64 |
| Ilustración 33 - Detalles de material mayor | 65 |
| Ilustración 34 - Listado de mantenciones pendientes | 66 |
| Ilustración 35 - Detalle de mantención pendiente | 66 |
| Ilustración 36 - Resolución de mantención programada pendiente..... | 67 |

1. Introducción

Bomberos de Chile, formalmente conocidos como la Junta Nacional de Cuerpos de Bomberos de Chile (JNCB), es una organización que agrupa administrativamente a los 320 cuerpos bomberiles a lo largo del país. Cada cuerpo se dedica al combate de incendios urbanos y forestales, el rescate de personas en ciudades y accidentes de tránsito, el manejo de incidentes que involucran el control de materiales considerados peligrosos, entre otras labores de la misma índole. Dichas tareas son llevadas a cabo por las ramas operativas de cada cuerpo, denominadas compañías, de las cuales existen (aproximadamente) 1.200 en todo el país. La primera compañía de Bomberos fue creada en 1851 en la ciudad de Valparaíso. Desde entonces la totalidad de los integrantes operativos de cada cuerpo han actuado en la calidad de “voluntarios”, sin recibir remuneración alguna por su trabajo, e incluso teniendo que pagar una cuota para poder ejercer esta vocación.

Legalmente cada Cuerpo de Bomberos es una entidad jurídica independiente, conformada como una corporación, y por tanto no existe una relación jerárquica estricta con respecto a la Junta Nacional. Sin embargo ésta actúa como medio coordinador de la acción de los diversos cuerpos, apoya sus necesidades administrativas y económicas, y actúa como canal de comunicación expedito con el Gobierno y los otros Cuerpos de Bomberos. Además, la Junta Nacional mantiene la Academia Nacional de Bomberos, encargada de realizar “docencia, investigación y extensión en todas las materias relacionadas con las actividades bomberiles”, además de realizar cursos de capacitación y perfeccionamiento para los voluntarios.

Como parte vital de su trabajo Bomberos de Chile debe comprar y mantener material mayor para cada cuerpo del país, correspondiente a una flota de cerca de 2.500 carros bomba, de rescate, portaescalas, etc., y cuyo precio puede alcanzar los 1.200.000 dólares por unidad. En este sentido Bomberos de Chile maneja en su conjunto la flota de vehículos mayores más grande del país, aun considerando a las empresas de transporte interurbano.

A pesar de las cifras anteriores, la gestión que se hace de estos vehículos es muy pobre. La JNCB no tiene conocimiento oportuno acerca de qué vehículos posee y de sus características, además de su ubicación y condiciones técnicas actuales. Esta información es crítica al momento de negociar acuerdos de mantenimiento con sus proveedores, evaluar la necesidad de adquirir nuevos carros, solicitar recursos al Gobierno, y gestionar su flota actual para distribuir de manera óptima los vehículos entre los distintos cuerpos bomberiles del país.

Una segunda familia de problemas que nace de esta falta de información es que los carros de bomberos son vehículos que requieren de mantención constante. Este mantenimiento es responsabilidad del cuerpo al que el vehículo está asignado actualmente, y garantiza su correcto funcionamiento ante emergencias. En la práctica estas mantenciones no se ejecutan, y los equipos permanecen relegados sin sus cuidados mínimos por largos periodos de tiempo. Esto genera una pérdida de recursos importante para la organización y, peor aún, aumenta innecesariamente el riesgo para voluntarios y víctimas de accidentes.

1.1. Problema a Resolver

Como se mencionó anteriormente, Bomberos de Chile cuenta con la flota de vehículos mayores más grande del país, pero cuya gestión es insuficiente para las necesidades administrativas de la JNCB y para atender efectivamente los siniestros que ocurren día a día, en los que en muchos casos hay personas en riesgo vital involucradas.

A través de entrevistas con el personal del departamento de Operaciones Bomberiles de la JNCB y con el departamento de Asistencia Técnica de Bomberos de Renault Truck International (principal proveedor de material mayor de la JNCB) se procuró tener una imagen del problema concreto que tienen los carros de Bomberos actualmente.

De dichas entrevistas se concluyó que el problema operacional a resolver es que muchos carros tienen desperfectos técnicos durante el trayecto al siniestro que van a combatir o simplemente no están disponibles cuando ocurre uno. Estos problemas son en su mayoría por falta de mantenimiento periódico, a pesar de que la Junta Nacional pone a disposición de cada cuerpo un sistema de envío gratuito de refacciones. Esta situación retrasa la llegada al lugar del incidente y aumenta el riesgo para las víctimas, además de incrementar los gastos de cada cuerpo, pues la reparación de un vehículo es más cara que su mantención. También es menester mencionar que un carro puede pasar meses fuera de servicio en espera de ser reparado tras uno de estos desperfectos, por lo que el cuerpo pierde parte importante de su capacidad operativa por un periodo prolongado.

En consecuencia, el problema de fondo a abordar en la memoria es de optimizar la vida útil de los vehículos de Bomberos de Chile, minimizando sus tiempos fuera de servicio y garantizando dentro de lo posible que no tendrán problemas técnicos durante sus horas de operación. Esto permitirá a los voluntarios trabajar de manera más eficiente, proveyendo ayuda oportuna a las víctimas y minimizando las consecuencias de incendios y otros incidentes.

Este problema de gestión será abordado a través de un sistema informático que formalice los procesos más importantes involucrados en el ciclo de vida de cada vehículo, y que además sea capaz de emitir alertas tanto para el cuerpo donde reside dicho vehículo como a nivel central en la Junta Nacional. Este ciclo de vida a grandes rasgos se puede visualizar en la Figura 1.

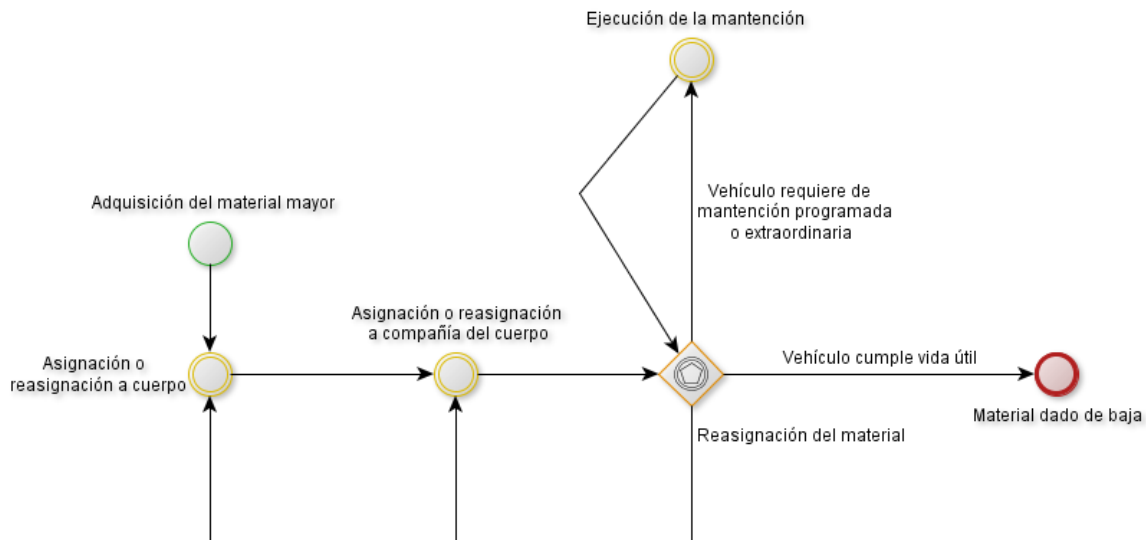


Ilustración 1 - Ciclo de vida del material mayor de Bomberos de Chile

Este proceso a gran escala empieza el momento en que la JNBC adquiere el vehículo en particular y lo asigna a un Cuerpo de bomberos en específico. Posteriormente el material es asignado a las operaciones de una compañía y desde ese momento el carro entra a un ciclo de mantención de su chasis (componentes mecánicas comunes a todo vehículo como el motor, caja de cambios, neumáticos, etc.) y de su carrosado (equipamiento bomberil como lo son mangueras, pitones, escaleras, etc.). Además el vehículo puede ser reasignado a un nuevo cuerpo cuando el carro cumple cierto número de años o el cuerpo original renueva su flota. Finalmente el carro es dado de baja cuando cumple su vida útil, sufre un accidente importante o cuando la JNCB estima conveniente.

El ciclo de vida también incluye a grandes rasgos los eventos de mantención extraordinaria, correspondientes a las reparaciones después de accidentes de tránsito o durante un acto de servicio. Dado que estos procesos son sumamente específicos su implementación no es incluida en la memoria. Asimismo el diagrama no muestra los procesos más simples por los que puede pasar un vehículo, como lo son la actualización de sus datos técnicos o su cambio de número de motor o chasis.

El desarrollo de este sistema informático propone desafíos técnicos y de ingeniería en varios niveles. Con respecto a su labor final, se debe hacer un trabajo de ingeniería cuidadoso para modelar de manera adecuada los procesos involucrados y los pasos de cada uno, de manera tal que resulte ser una herramienta útil y valiosa para Bomberos. En particular el sistema modela procesos que se dan a lo largo de más de 20 años, por lo que deberá ser usado por un periodo extremadamente largo.

En términos de interoperabilidad, el sistema depende de un esquema de autenticación de usuarios y permisos basados en cargos ya presente en la Junta Nacional de Bomberos, además de cargar información preexistente de carácter geográfico, como lo son las comunas, provincias y regiones donde opera Bomberos.

Detrás de este esquema de desarrollo e interoperabilidad hay problemas subyacentes importantes en la arquitectura de sistemas que maneja Bomberos hoy en día, que también deberán ser abordados con cuidado, y que se describen en los próximos párrafos.

Actualmente los sistemas informáticos de la JNCB no siguen una política consistente con respecto a sus fuentes comunes de información, de cómo y dónde almacenar datos, de interoperabilidad, de calidad de código y de documentación. Esto los convierte en desarrollos únicos y monolíticos, complejos de mantener en el tiempo e imposibles de extender con nuevas funcionalidades.

La falta de interoperabilidad con los servicios centrales de la JNBC (como la autenticación de usuarios y la obtención de información básica de cuerpos y compañías) hace que cada sistema deba comunicarse directamente con la base de datos central de Bomberos utilizando SQL. En particular cabe destacar que las relaciones entre tablas en la base de datos central de la JNCB no son formales (pueden ser "llaves" numéricas o en texto) y no tienen restricción de integridad, por lo que las consultas directas son en sí mismas volátiles y peligrosas. Asimismo, muchos campos de las tablas no están en uso o mantenimiento e inducen a errores por parte de los desarrolladores de nuevas aplicaciones.

Estas situaciones introducen graves problemas para las aplicaciones actuales y futuras pues, por ejemplo, si se cambia la estructura de una tabla o se decide migrar el motor de base de datos a una nueva plataforma, no hay garantías de la continuidad en las operaciones de cada sistema. Además este esquema no define claramente qué servicios provee la infraestructura central de la JNCB, violando los principios de delimitación y definición de responsabilidades de cada componente en una arquitectura de software.

Los nuevos desarrollos también se ven afectados por este esquema, pues deben repetir las mismas funcionalidades ya implementadas por otros sistemas (esquema de autenticación de usuarios, obtención de permisos, etc.) y nuevamente imposibilitan cambios posteriores a la estructura de la base de datos o a la infraestructura física de los servidores de bomberos.

Finalmente, la falta de interoperabilidad entre los distintos sistemas limita algunas de las funcionalidades que desea implementar la JNCB en el mediano y largo plazo. Por ejemplo, se desea que cada usuario sólo deba autenticarse una vez contra el sistema como un todo y poder acceder libremente a cada subsistema sin tener que ingresar sus credenciales nuevamente, lo que resulta complejo o poco viable para la arquitectura actual.

1.2. Objetivos de la Memoria

El objetivo general de la memoria es desarrollar un sistema informático de gestión y ciclo de vida de material mayor para la Junta Nacional de Cuerpos de Bomberos de Chile. Dicho sistema deberá permitir administrar y visualizar varios de los procesos de negocio asociados al mantenimiento de los vehículos desde el momento en que son adquiridos y asignados a un cuerpo hasta que son dados de baja.

El más importante y complejo de estos procesos es el de mantención programada de material mayor. Este subsistema debe monitorear cada vehículo y, en función de su chasis (componentes mecánicas) y carrozado (mangueras, escaleras, etc.) generar órdenes periódicas de mantención técnica. Este proceso incluye recordatorios permanentes a la Junta Nacional y a los encargados locales de cada vehículo para evitar problemas técnicos a futuro.

Para levantar este proceso se requiere de una serie de desarrollos previos que permitan disponer de la información necesaria para generar las ordenes de mantención, que incluyen la implementación del proceso de dar de alta un vehículo, su asignación o reasignación a un cuerpo bomberil en particular, el ingreso de las pautas generales que rigen su ciclo de mantenciones, entre otros.

Como solución a la falta de fuentes comunes de información evaluada en la sección anterior, este levantamiento requiere el despliegue de una solución de interoperabilidad a nivel de la infraestructura básica de Bomberos. Esto permitirá que el sistema acceda a la información de autenticación, permisos de usuario e información básica de cada cuerpo y compañía utilizando servicios remotos de alto nivel e independientes del motor de procesamiento y de la estructura interna de la base de datos subyacente.

Los objetivos específicos que se desprenden del objetivo general son los siguientes:

- Diseñar y levantar una arquitectura de servicios y acceso a datos para la interoperabilidad de nuevas aplicaciones con la infraestructura actual de la Junta Nacional de Bomberos.
- Diseñar e implementar un sistema informático que permita la gestión del ciclo de vida del material mayor de bomberos, en particular de su proceso de mantención programada de vehículos.
- Diseñar e implementar un modelo de datos y flujo de procesos acorde a las necesidades del sistema anterior y apegado a un conjunto de convenciones y mejores prácticas.
- Documentar el modelo de datos, proceso de negocios, levantamiento del ambiente de desarrollo e instalación de la aplicación en producción.

2. Requisitos de la Solución

En esta sección se presentan los requisitos de alto nivel del sistema informático propuesto, correspondientes a las necesidades principales que el software debe abordar para resolver los problemas expuestos anteriormente.

Posteriormente se detallan los tipos de usuario emanados de dichos requisitos, sus casos de uso y la formalización de los procesos de negocio involucrados.

2.1. Requisitos funcionales

A continuación se describen los requisitos funcionales que debe cumplir el sistema.

Poder cargar pautas de mantención generales de chasis y carrosado

El subsistema más importante de la aplicación es aquel que gestiona el mantenimiento programado de cada vehículo. Para lograr esto el área de Operaciones Bomberiles de la JNCB debe ser capaz de cargar las pautas de mantenimiento asociadas a variados modelos de chasis y carrosado. Estas pautas describen qué mantenciones se deben realizar (“Cambio de aceite”, por ejemplo) y su frecuencia.

Estas pautas pueden constar de cientos de operaciones, por lo que rellenarlas directamente en un formulario web involucra riesgos que se desean evitar. Estos riesgos están relacionados directamente con lo fácil que es perder información al momento de rellenar un formulario directamente en un navegador de Internet. Por ejemplo, si se cierra accidentalmente la ventana del navegador mientras se realiza el proceso entonces se perderá toda la información ingresada hasta ese momento sin advertencia. También se puede dar el caso que un error en la conexión a Internet o de parte del servidor resulte en una petición HTTP parcial o corrupta que haga que el formulario no sea procesado adecuadamente, dejando al navegador en un estado inconsistente, en el que probablemente también se perderá la información ingresada.

Como contraparte, llenar los formularios localmente en un formato intermedio resulta más seguro, pues muchos de los programas del mercado hacen guardados progresivos de los cambios hechos y alertan al usuario antes de cerrar la aplicación si es que hay cambios sin guardar.

Existe una variedad de formatos que son útiles para esta tarea, cuya edición es sencilla con herramientas gratuitas y cuyo procesamiento por parte del servidor web es igualmente simple. Ejemplos de estos son los formatos CSV (Comma-separated values), ODS (OpenDocument Spreadsheet) y XLS (Microsoft Excel Spreadsheet).

Permitir dar de alta material mayor

Todos los procesos dentro de la aplicación dependen de la capacidad de “dar de alta” vehículos al sistema informático, esto a través de un formulario web que recabe los datos principales de cada vehículo y los ingrese formalmente a la flota, además de asignarles las pautas de mantenimiento de carrocería y chasis que les corresponden.

Contrario a la operación de cargar pautas de mantención, el dar de alta un vehículo no involucra un formulario tan extenso, y además dicho formulario incluye campos de selección múltiple dinámicos, cuyas opciones son cargadas desde la base de datos del sistema, haciendo compleja la posibilidad de rellenarse fuera del navegador web.

La capacidad de dar de alta material mayor debe estar disponible tanto para el área de operaciones bomberiles de la JNCB como para los comandantes de cada cuerpo, pero en este último caso requiere de una validación posterior por parte de la Junta Nacional.

Poder asignar y reasignar material mayor entre distintos cuerpos / compañías

Parte fundamental del ciclo de vida de cada carro de Bomberos es su asignación a servicio como parte de la flota de un cuerpo bomberil. Este proceso se puede dar múltiples veces durante la vida útil del vehículo, y hacerle seguimiento permite saber dónde está cada vehículo y quien es el encargado de las mantenciones del mismo.

Por este motivo, el área de operaciones de la JNCB debe ser capaz de asignar y reasignar material mayor entre los distintos cuerpos a nivel nacional, mientras que los encargados de cada cuerpo deben poder distribuir sus flotas entre las compañías que les corresponden.

Generar automáticamente instrucciones de mantención programada para cada vehículo

A partir de las pautas de mantenimiento asociadas a cada carro y la fecha en que fue dado de alta, el sistema informático debe ser capaz de generar las órdenes de mantención de cada vehículo en el momento que le corresponda. Estas pautas deben ser confeccionadas automáticamente a partir de los datos existentes y ser notificadas a los encargados correspondientes en la JNCB y el cuerpo encargado del vehículo en cuestión.

Permitir resolver las órdenes de mantención programadas

Una vez emitidas las órdenes de mantención de cada vehículo, estas deben poder ser atendidas por el encargado correspondiente de cada cuerpo.

Esto implica poder marcar cada operación individual como ejecutada, posponerla en caso de falta de recursos, poder subir archivos asociados a su ejecución (cotizaciones, facturas, etc.) y permitir dejar comentarios acerca de cada una.

Mantener la hoja de vida de cada vehículo

Una de las necesidades más importantes del sistema es poder archivar automáticamente todos los eventos por los que ha pasado un vehículo. Esto incluye el momento en que entra a la flota, sus asignaciones y reasignaciones a cuerpos y compañías, sus mantenciones programadas, sus cambios de características técnicas, entre otros.

Poder dar de baja material mayor

Todos los vehículos de la flota de Bomberos eventualmente son dados de baja por diversos motivos, ya sea por un accidente gravísimo, por una enajenación o simplemente porque ya se cumplió con su vida útil.

Esta capacidad debe estar disponible para los encargados de operaciones bomberiles de la Junta Nacional y debe incluir los documentos y antecedentes que se consideren necesarios para el caso.

Permitir actualizar datos técnicos

El proceso de dar de alta un vehículo no requiere detallar en gran medida todas sus características técnicas, pero estas eventualmente pueden ser útiles para los fines estadísticos de la Junta Nacional.

En este sentido las características de un vehículo deben poder ser actualizadas por el personal encargado en su cuerpo de Bomberos correspondiente o directamente por la Junta Nacional.

Implementar procesos secundarios del ciclo de vida de cada vehículo

Además de sus grandes eventos como reasignaciones o mantenciones programadas un vehículo también pasa por procesos más acotados, pero que en su conjunto son igualmente importantes para los fines de gestión de la Junta Nacional.

Por ejemplo, se debe proveer la capacidad de asignarle una patente vehicular a cada carro. Esta facultad es importante pues la JNCB está en un proceso de normalizar la situación legal de todos sus vehículos, y disponer de la información de cuales aún no tienen placa patente es muy útil.

Ofrecer un sistema de almacenamiento de documentos

Muchas de las operaciones antes mencionadas involucran documentos binarios en formatos como PDF, Word o Excel que son complejos de convertir a otros formatos y engorrosos para digitar. Por este motivo se debe ofrecer la facilidad para subirlos directamente al sistema y que después se puedan descargar cuando se estime conveniente.

Estos documentos deben estar bajo un sistema de permisos estricto que sólo permita su acceso al personal de la JNCB y a los encargados del cuerpo en que reside el vehículo correspondiente.

2.2. Decisiones iniciales

La arquitectura y posterior aplicación a desarrollar deben cumplir con una serie de requisitos técnicos que se detallan a continuación.

Con respecto a la arquitectura de interoperabilidad de los nuevos sistemas de la JNCB con su infraestructura actual, se debe seleccionar cuidadosamente un protocolo de comunicación adecuado (SOAP, JSON-RPC, etc.) que tenga buen soporte en múltiples ambientes de desarrollo. Además, el diseño de esta arquitectura debe permitir agregar nuevas funcionalidades de manera simple, lo que también exige una documentación exhaustiva de su modo de operación. El desarrollo de este “proveedor de servicios” (como será denominado de ahora en adelante) exige el uso de prácticas que garanticen una alta disponibilidad para el sistema y un mecanismo de registro de eventos completo acerca de sus operaciones diarias para identificar puntos de error.

Por su parte, la aplicación de gestión del ciclo de vida de material mayor será utilizada por usuarios finales, por lo que tiene requisitos más específicos. En primer lugar, el sistema debe poder autenticar usuarios usando sus credenciales ya existentes en la JNCB y dar permisos dependiendo del cargo que ocupen dentro de la organización. Este proceso se debe llevar a cabo usando el proveedor de servicios descrito anteriormente.

La aplicación será utilizada en los 150 cuerpos de bomberos más grandes de Chile, distribuidos a lo largo de todo el país. Por este motivo la solución tiene que estar disponible de forma centralizada, requiriendo que funcione a través de una interfaz web. Dicha interacción debe tener un alto nivel de usabilidad pues el grado de capacitación digital dentro de los cuerpos de bomberos varía considerablemente.

En el plano técnico, la aplicación se debe apegar a las mejores prácticas de desarrollo de aplicaciones web, esto es, promover el uso de conexiones *stateless*, tener consideraciones de seguridad en cada interfaz pública, usar un esquema de desarrollo Modelo Vista Controlador (MVC), aplicar patrones de diseño en donde se estime adecuado, promover la reutilización y refactorización de código, y dentro de lo posible hacer optimizaciones para mejorar el desempeño y rapidez de respuesta del sistema.

En particular la aplicación depende de un sistema de *scheduling* para monitorear las mantenciones programadas de cada carro de Bomberos. La implementación de este sistema introduce aspectos considerablemente complejos al tratarse de un servicio que corre en *background* de forma independiente al sistema que resuelve las consultas web.

2.3. Usuarios del sistema

Los usuarios que hacen uso del sistema se pueden dividir en dos grandes grupos: Aquellos que trabajan directamente en la Junta Nacional de Cuerpos de Bomberos y aquellos que trabajan en alguno de los 320 cuerpos bomberiles a lo largo de Chile, por este motivo resulta conveniente definirlos por separado.

2.3.1. Usuarios dentro de la JNCB

Dentro de la JNCB se distinguen tres tipos de usuarios en función del departamento en el que trabajan dentro de la Junta Nacional.

Usuario del departamento de Operaciones Bomberiles

Este usuario es el encargado principal de mantener los aspectos administrativos del sistema, contando con permisos para subir nuevas pautas de mantención, dar de alta vehículos inmediatamente, monitorear a cada uno y correr sus procesos cuando corresponda.

En el plano administrativo Operaciones Bomberiles es el departamento dentro de la JNCB que hace uso más frecuente e intensivo del sistema, por lo que consecuentemente requiere de todos los permisos disponibles.

Este usuario está involucrado en los casos de uso mostrados en las ilustraciones 2, 3, 4, 5, y 6.

Usuario del departamento de Adquisiciones

Los integrantes del departamento de adquisiciones son capaces de dar de alta material mayor y ver sus detalles técnicos. En particular están interesados en tener acceso a los documentos asociados a la compra o donación del vehículo a la JNCB para los diversos trámites que tienen que realizar en conjunto con Aduanas o el Servicio de Impuestos Internos, entre otros organismos.

Este usuario está involucrado en el caso de uso mostrado en la ilustración 2.

Usuario del departamento de Jurídica

El usuario de este departamento sólo requiere tener acceso de lectura a los datos del vehículo y en particular a sus documentos históricos, como por ejemplo su solicitud de primera inscripción y permiso de circulación. Estos datos son importantes en las disputas legales que ocasionalmente involucran a un vehículo de Bomberos, como por ejemplo en el caso de un choque con un vehículo particular durante el trayecto a un acto de servicio.

2.3.2. Usuarios dentro de los cuerpos de Bomberos

A nivel de cuerpo se distinguen los usuarios en función de los cargos que ocupan dentro de la organización de Bomberos. En ese sentido se identifican los siguientes usuarios.

Comandante de cuerpo

Encargado principal de las operaciones del cuerpo. Para los efectos del sistema informático el comandante es capaz de dar de alta nuevos vehículos (aunque sujeto a validación de parte de la JNCB), asignar sus vehículos entre las distintas compañías que conforman el cuerpo bomberil, y revisar sus hojas de vida.

Además el comandante puede ver las mantenciones programadas pasadas y actuales por las que ha pasado cada uno de sus vehículos.

Este usuario está involucrado en los casos de uso mostrados en las ilustraciones 2, 4, 5, y 6.

Encargado de material mayor

En principio cada cuerpo tiene dentro de su personal a un encargado de material mayor, definido por el comandante y a cargo de la mantención técnica de la flota de vehículos del cuerpo.

Para los efectos del sistema este usuario puede ver los vehículos actualmente asignados a su cuerpo y resolver las órdenes de mantenimiento que el sistema genera.

En particular este usuario recibe notificaciones constantes si alguno de sus vehículos tiene mantenciones programadas pendientes.

Este usuario está involucrado en el caso de uso mostrado en la ilustración 6.

2.4. Casos de Uso

A continuación se muestran los casos de uso más importantes de la aplicación, en donde los actores corresponden a los usuarios con roles y cargos definidos anteriormente.

Ingreso de material mayor

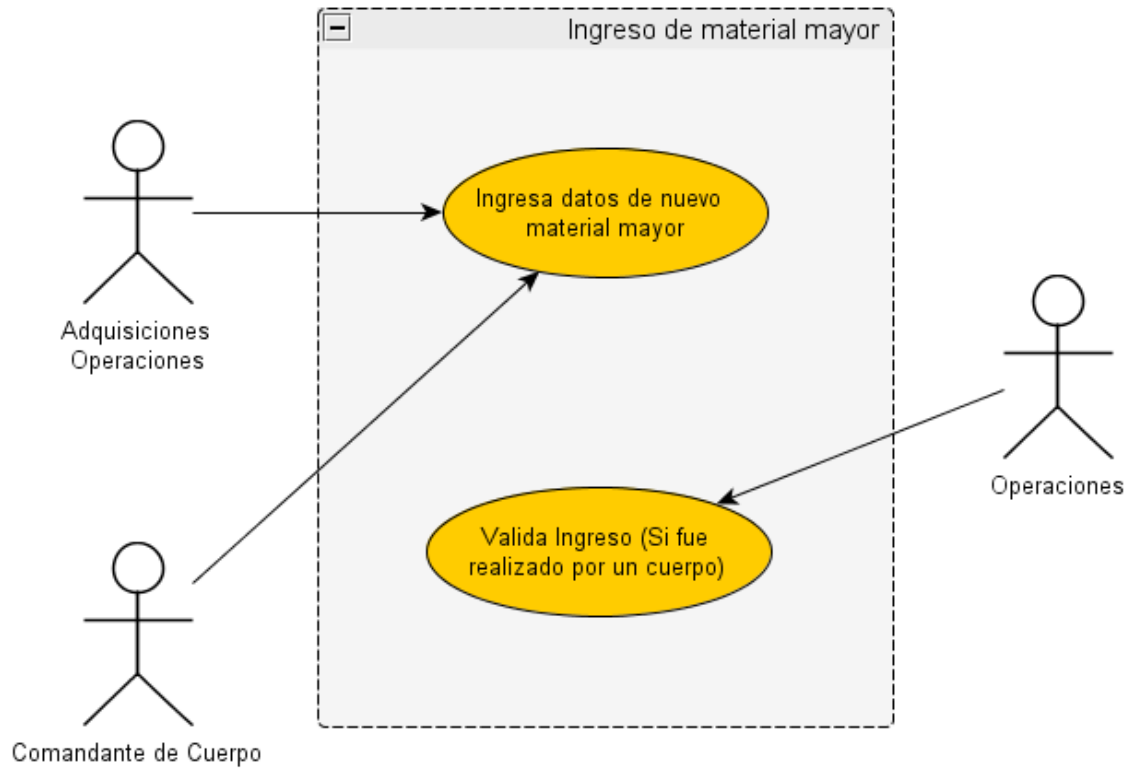


Ilustración 2 - Caso de Uso de Ingreso de Material Mayor

Carga de pauta de mantenimiento de chasis o carrosado

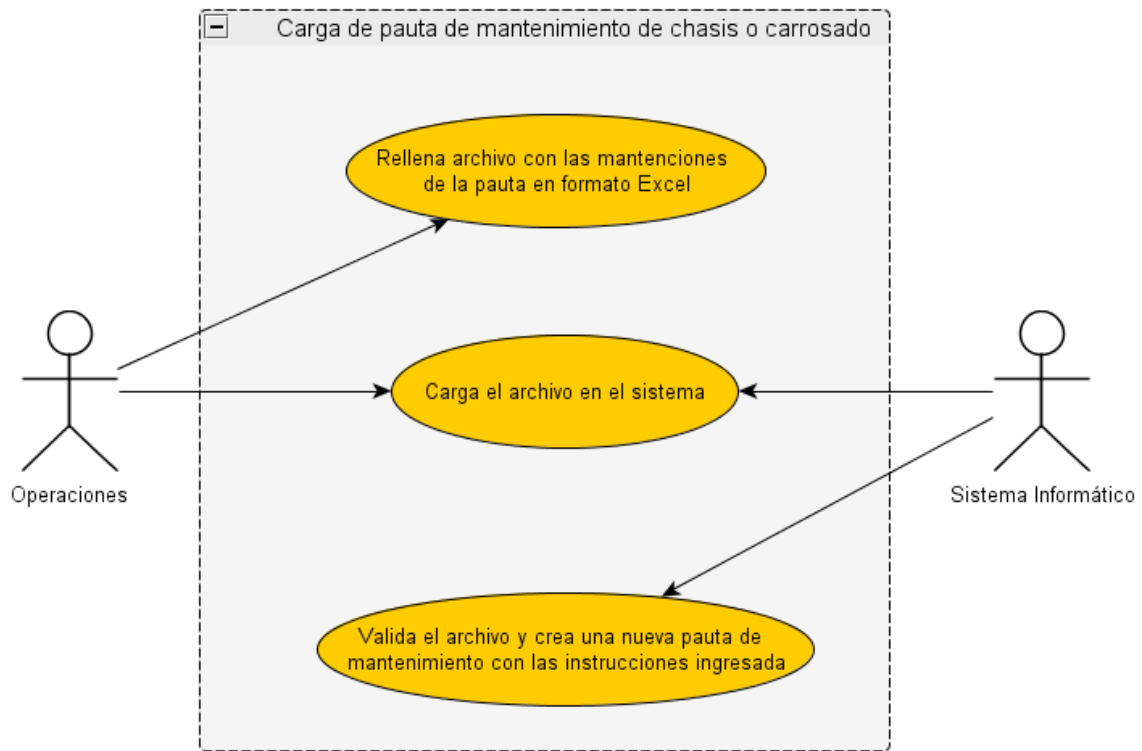


Ilustración 3 - Caso de Uso de Carga de pauta de mantenimiento de chasis o carrosado

Asignación de material mayor a cuerpo

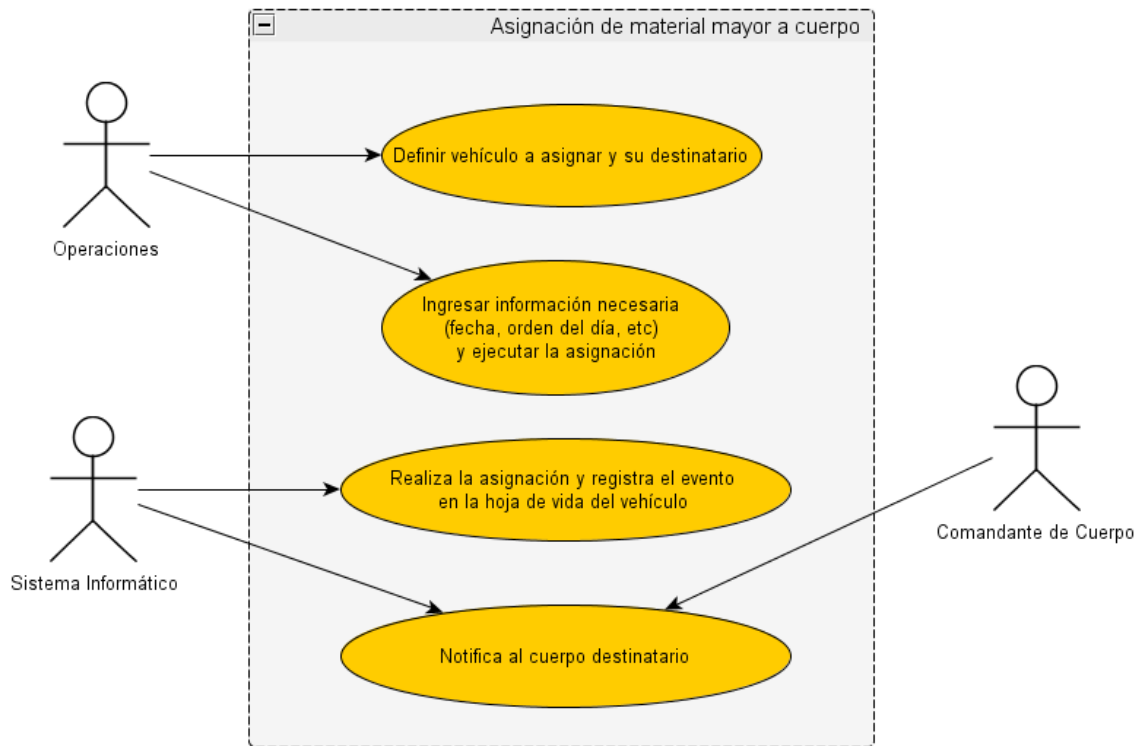


Ilustración 4 – Caso de Uso de asignación de material mayor a cuerpo

Asignación de material mayor a compañía

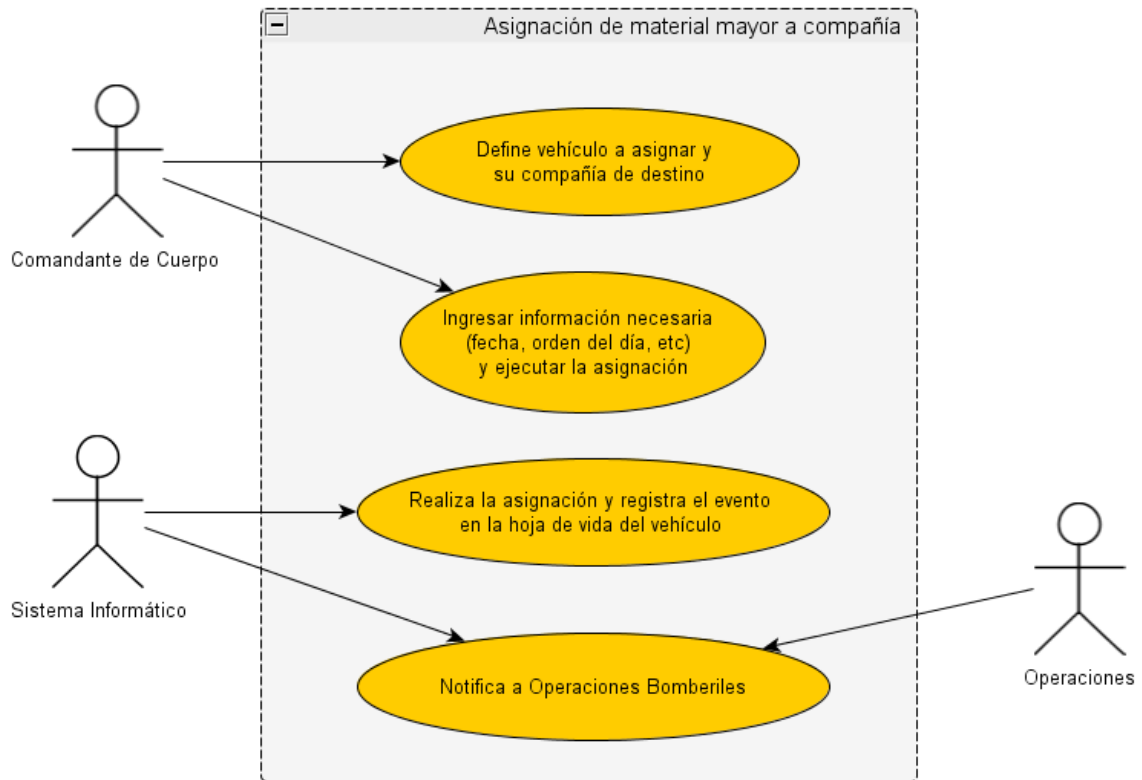


Ilustración 5 - Caso de Uso de Asignación de material mayor a compañía

Generación y resolución de órdenes de mantenimiento

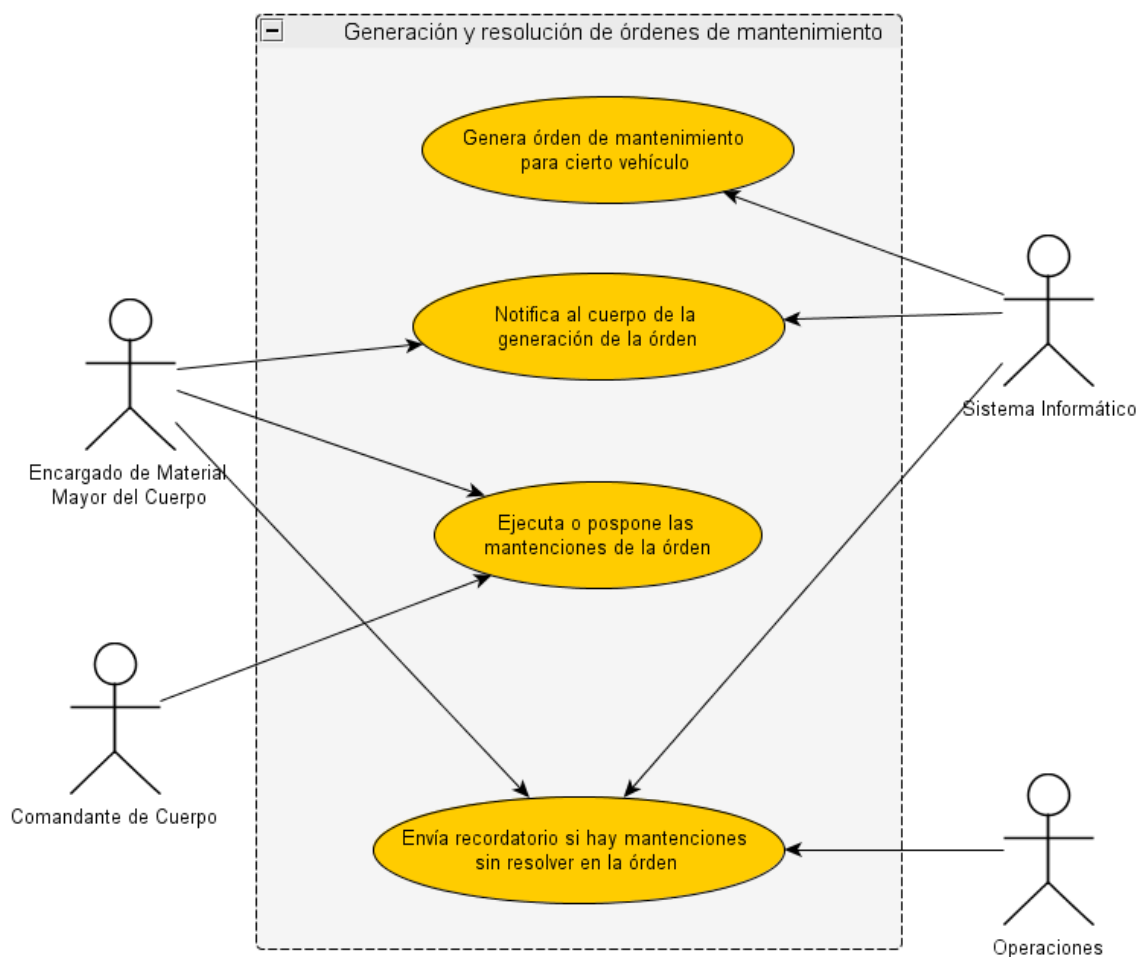


Ilustración 6 - Caso de Uso de Generación y resolución de órdenes de mantenimiento

2.5. Procesos involucrados

A continuación se muestran en formato BPMN el detalle de los procesos más importantes involucrados en la herramienta desarrollada.

2.5.1. Dar de alta material mayor

El proceso de dar de alta material mayor se ejecuta cuando el personal de la JNCB o de un cuerpo bomberil desea agregar un nuevo vehículo a la flota. El usuario debe definir el origen de este (compra o donación) y, dependiendo de la opción seleccionada, debe rellenar distintos formularios. Finalmente, si el vehículo fue dado de alta por un cuerpo de Bomberos, este debe ser validado por el personal de la Junta Nacional antes de considerarse como parte del listado de vehículos.

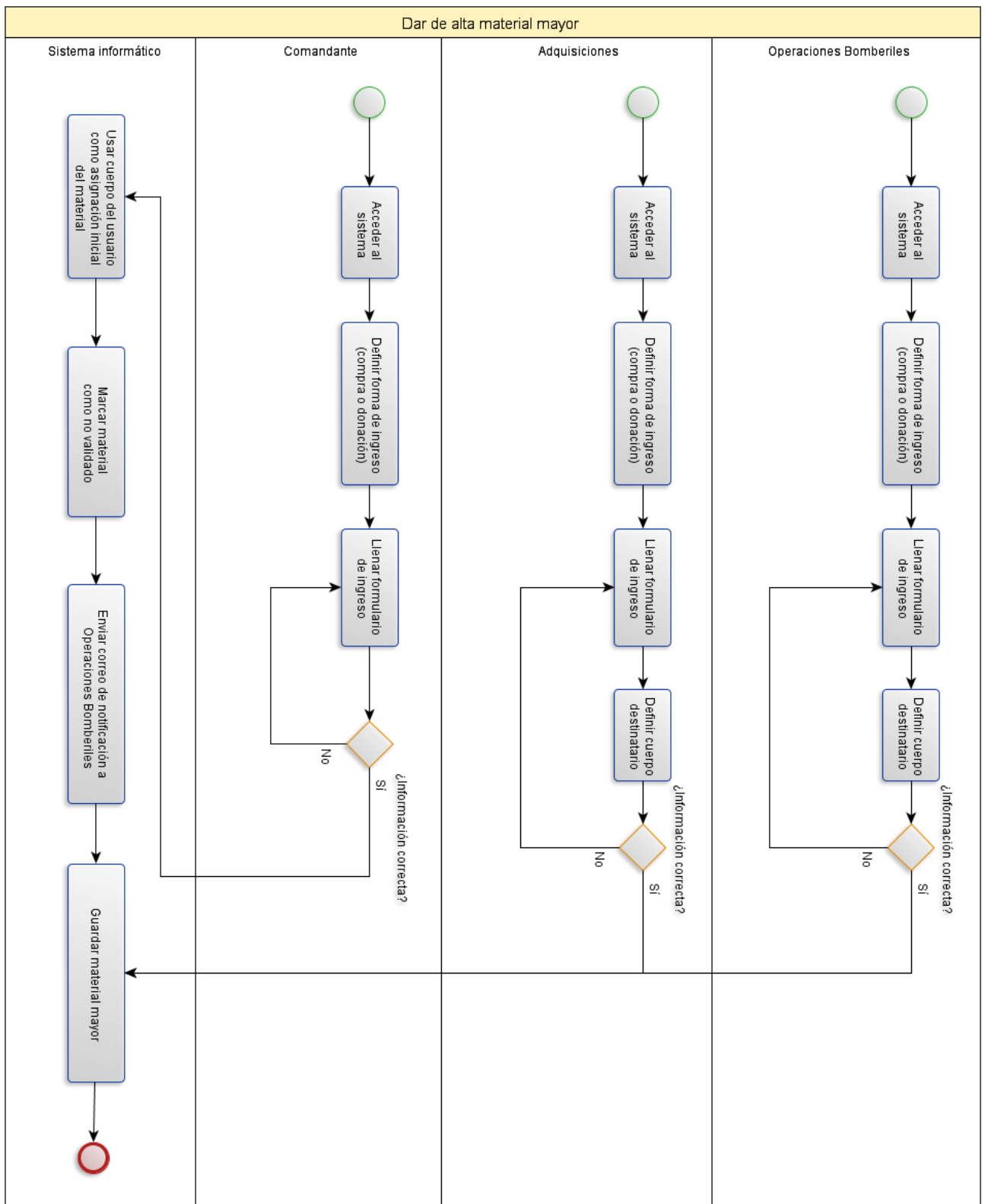


Ilustración 7 - Proceso de Dar de alta material mayor

2.5.2. Crear pauta de mantenimiento (de chasis o carrosado)

Todos los vehículos activos en el sistema deben tener asignadas dos pautas de mantenimiento, la primera asociada a su modelo de chasis y la segunda al equipo bomberil que lleva (carrosado).

Ambos tipos de pauta son generados por personal de Operaciones Bomberiles rellorando un archivo XLS (Excel) cuya estructura se ejemplifica en la Ilustración 8. Este archivo es posteriormente subido y analizado por el sistema informático. Si este no contiene errores entonces queda disponible para ser asignado a uno o varios carros.

Se optó por usar archivos de Microsoft Excel (XLS) pues el personal de la JNCB se siente cómodo con su uso y ya cuenta con las licencias de software para su manipulación, además de ser sencillos de procesar por parte de la aplicación.

| | A | B |
|----|---|-------------------|
| 1 | Nombre de la pauta | |
| 2 | Pauta de mantenimiento de carrosado para vehículos Renault | |
| 3 | | |
| 4 | Mantenición | Frecuencia |
| 5 | Vaciar motor | 12 meses |
| 6 | Vaciar circuito refrigerante y cambio de tapon presión/depresión | 24 meses |
| 7 | Cambio de cartucho del filtro de aceite del motor y limpieza de la cuva | 12 meses |
| 8 | Cambio de cartucho(s) de filtro(s) de combustible y limpieza de la cuva | 12 meses |
| 9 | Cambio de prefiltro de combustible | 12 meses |
| 10 | Cambio filtro de aire y limpieza de la cuba | 24 meses |
| 11 | Cambio de correa(s) | 36 meses |
| 12 | Cambio rodillo tensor del alternador | 12 meses |
| 13 | Limpiar radiador(es) soplando aire comprimido (o agua caliente) a baja presión | 12 meses |
| 14 | Verificar estanqueidad del circuito de de admision de aire | 12 meses |
| 15 | Verificar funcionamiento del cilindro de mando del ralentizador en el escape | 24 meses |
| 16 | Verificar estanquidad del motor (aceite, líquido refrigerante, combustible) | 12 meses |
| 17 | Verificar todos los niveles | 12 meses |
| 18 | Verificar estado y tensión de las correas | 12 meses |
| 19 | Verificar rejilla de entrada del filtro de aire (límpiela si es necesario) | 12 meses |
| 20 | Verificar limpieza exterior de las tuberías del (los) radiador (es) | 12 meses |
| 21 | Verificar juego de las válvulas | 24 meses |
| 22 | Verificar estado y fijaciones del cableado | 12 meses |
| 23 | Verificar estado y apriete de los silent block de fijación del motor y radiador | 12 meses |
| 24 | Verificar grado de proteccion del circuito refrigerante | 12 meses |
| 25 | | |

Ilustración 8 - Ejemplo de pauta de mantenimiento

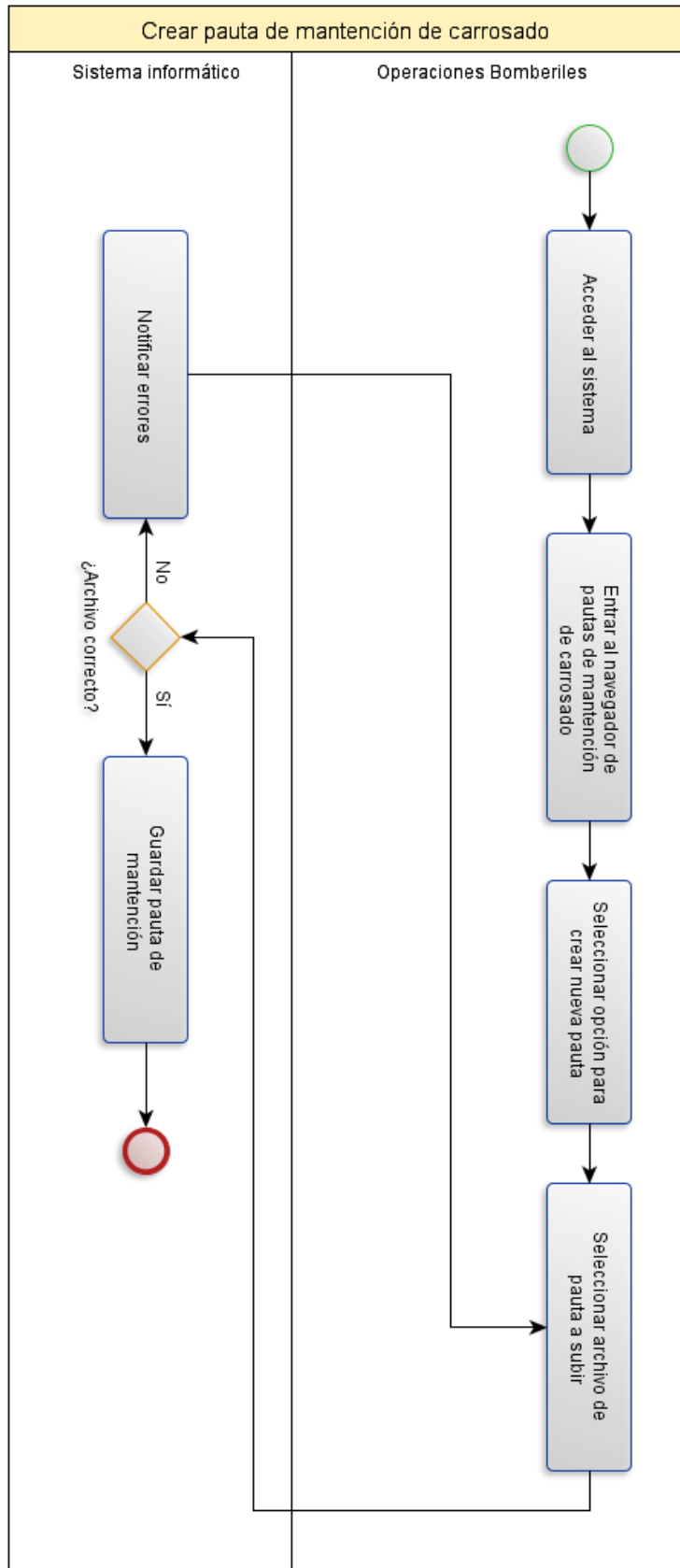


Ilustración 9 - Proceso de crear pauta de mantenimiento

2.5.3. Asignar y reasignar material mayor a cuerpo

Cuando un carro es dado de alta al sistema se debe definir su primer cuerpo bomberil donde será utilizado para el combate de incendios, rescate u otras tareas. Este proceso tiene el nombre de “asignación” y se realiza siempre al formalizar la adquisición de un vehículo.

Además, un carro puede ser “reassignado” de un cuerpo a otro por una variedad de motivos. Razones para esto pueden ser la renovación de flota del cuerpo original, una necesidad de emergencia del cuerpo destinatario, entre otros.

Si bien los procesos de asignación y reasignación son formalmente distintos, acá mostramos sólo el más complejo de ambos (el de reasignación) pues para los efectos del sistema informático son relativamente similares en sus pasos.

2.5.4. Asignar y reasignar material mayor a compañía

Una vez que un vehículo ha sido asignado a un cuerpo, el comandante del mismo debe nuevamente asignarlo, pero esta vez a alguna de las compañías que conforman el cuerpo. De la misma manera, un vehículo puede ser reasignado a una compañía distinta si la situación lo amerita. Este proceso es independiente de la gestión de la Junta Nacional, pues esta principalmente se preocupa de la administración de Bomberos hasta el nivel de cuerpos y no de compañías.

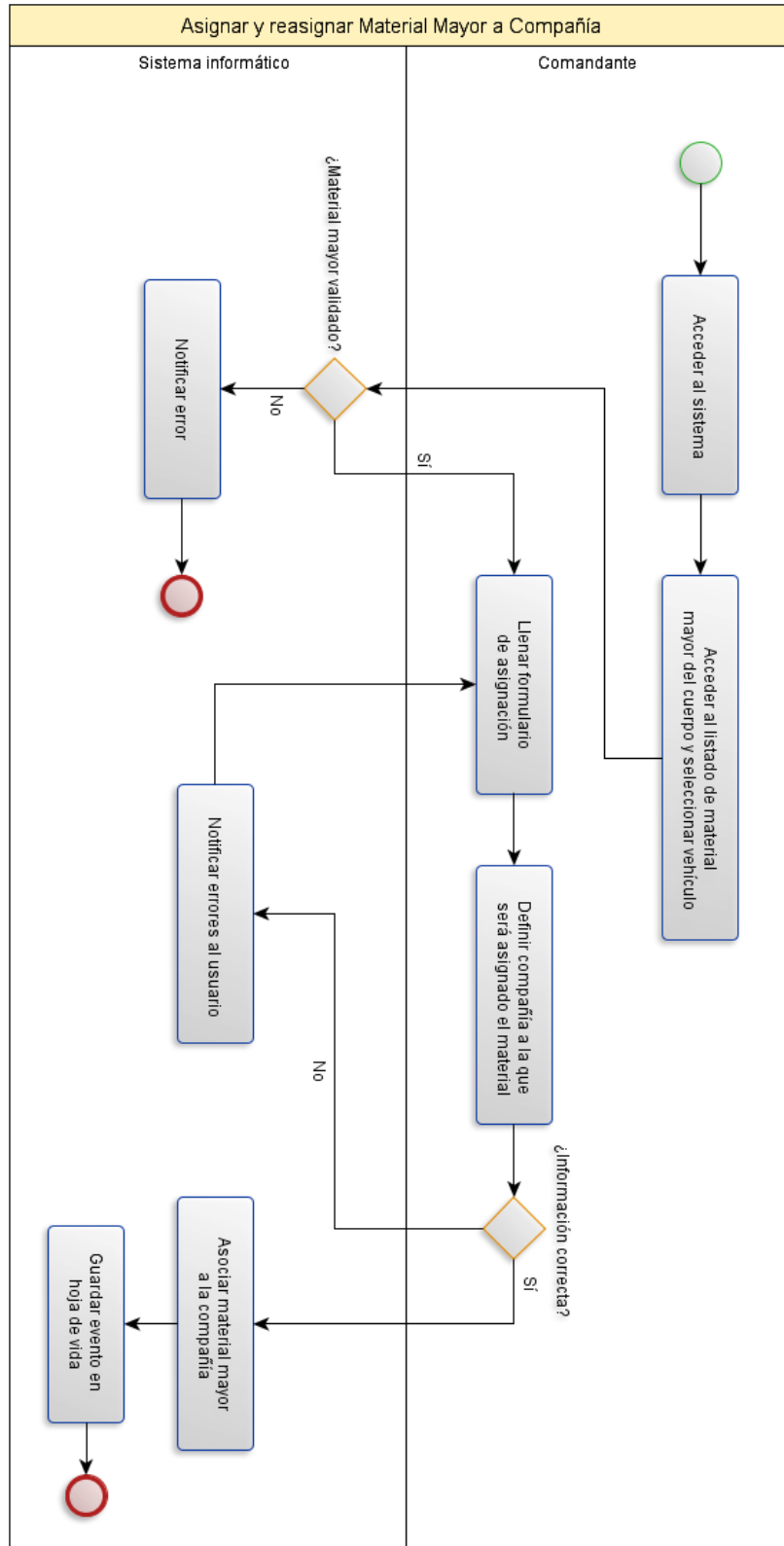


Ilustración 11 - Proceso de asignar y reasignar material mayor a compañía

2.5.5. Generación y resolución de órdenes de mantención programada

Uno de los objetivos principales del sistema es automatizar el proceso mediante el cual se generan las órdenes de mantención programadas para cada uno de los cerca de 2.500 vehículos que posee Bomberos en la actualidad.

Para conseguir esto el sistema informático periódicamente revisa todos los carros ingresados en el sistema. Para cada uno de ellos se obtienen sus pautas de mantención de chasis y carrosado y se confirma si es que le corresponde algún procedimiento. De ser este el caso se genera una orden de mantenimiento con todas las instrucciones y se notifica al cuerpo correspondiente de esta situación.

Para resolver la orden de mantención, el encargado de material mayor o comandante del cuerpo en cuestión debe entrar al sistema, ver la orden, y marcar cada instrucción como ejecutada o pospuesta, como se muestra en el detalle del proceso.

Si el cuerpo bomberil no resuelve por completo la orden, continuará recibiendo recordatorios hasta que la termine.

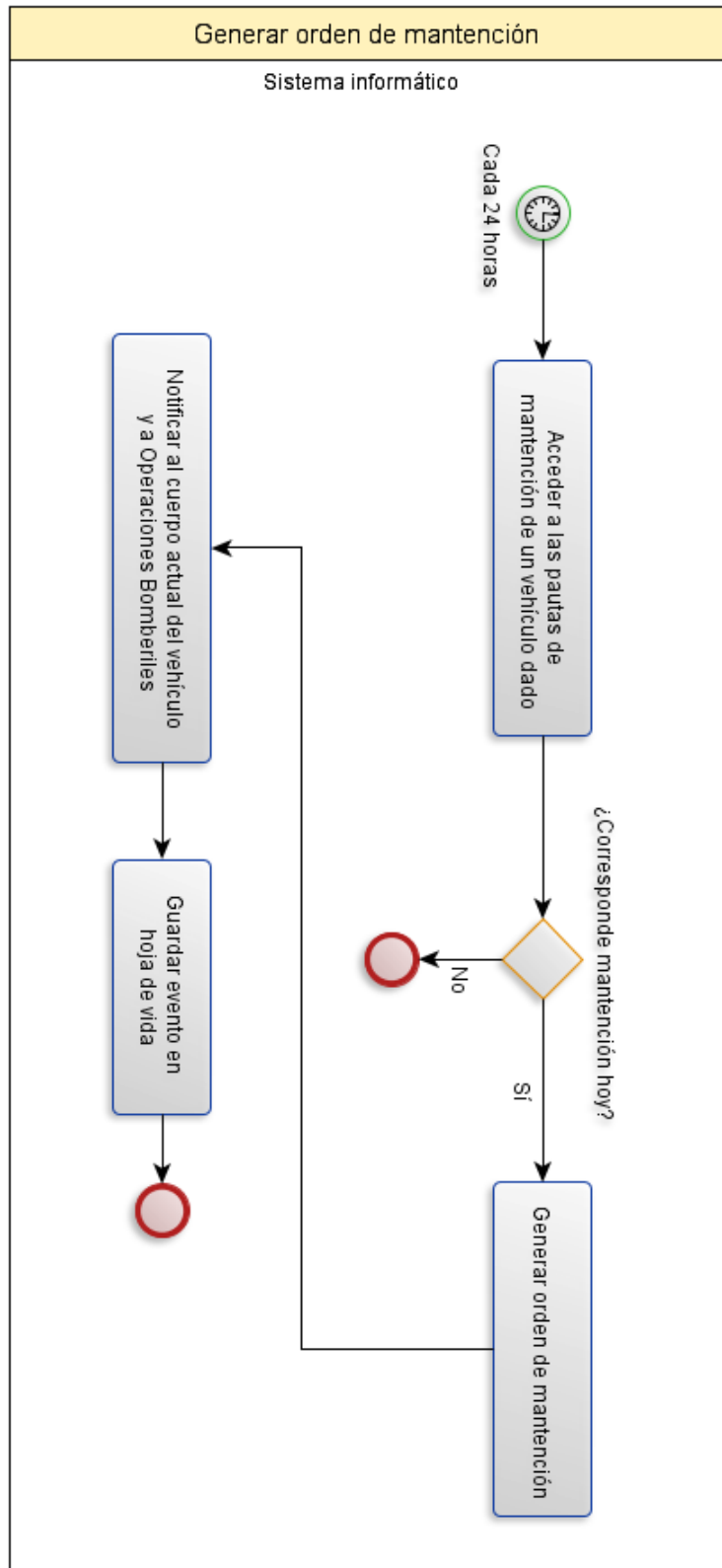


Ilustración 12 - Proceso de generar orden de mantención

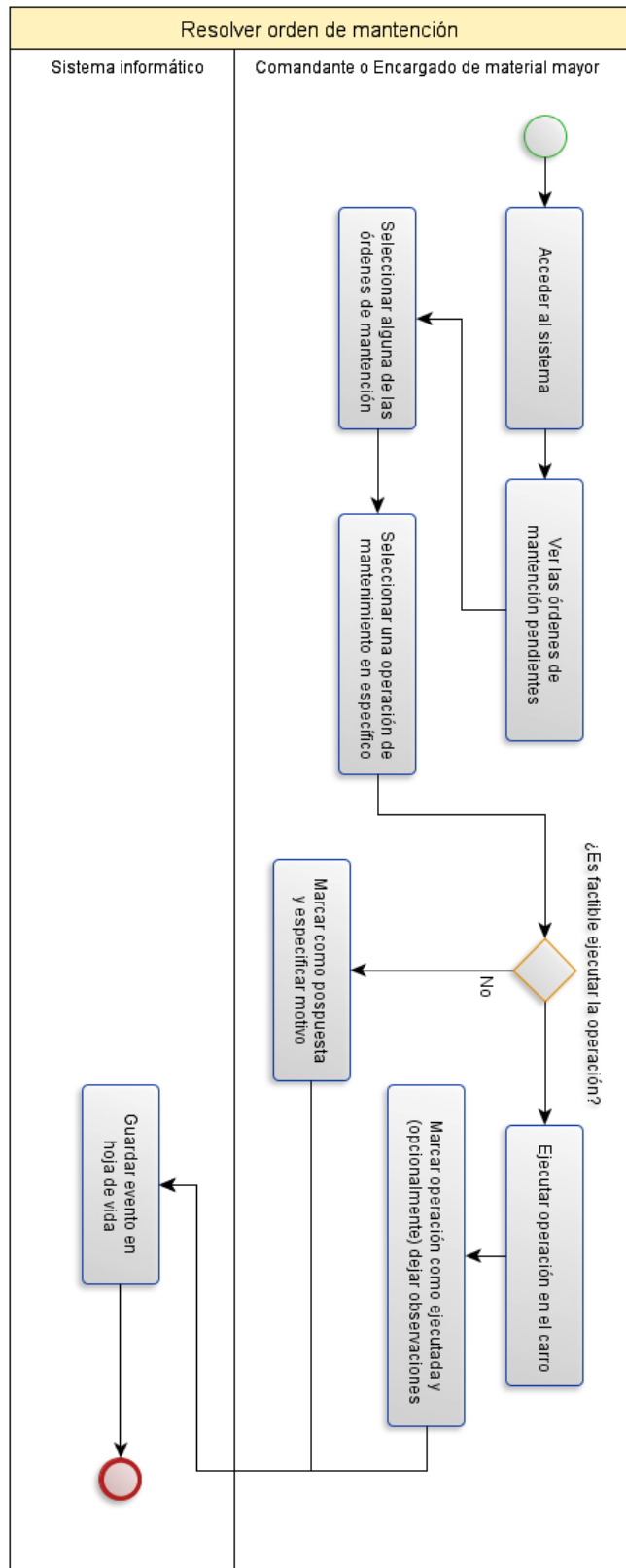


Ilustración 13 - Proceso de resolver orden de mantención

3. Diseño arquitectónico de la solución

En esta sección se presentan las componentes macro más importantes que conforman el sistema. Dichas partes corresponden a la infraestructura física sobre la que corre el software (servidores y bases de datos), a la elección de protocolos y formatos de comunicación entre dichas partes, y a las capas y patrones más importantes dentro del software propiamente tal.

También se detalla el modelo de datos elaborado para representar toda la información que desea albergar el sistema en función de los requerimientos y procesos mostrados anteriormente.

3.1. Arquitectura física

Las aplicaciones ya existentes en los sistemas de la JNCB se rigen por una arquitectura física bastante estándar, descrita en el recuadro “Arquitectura legada” de la Ilustración 14 - Arquitectura física.

Bajo este esquema una aplicación accede a la información básica de Bomberos (datos de regiones, provincias, comunas, cuerpos, compañías y voluntarios) a través de consultas SQL directas al servidor de base de datos donde están guardadas estas entradas. En principio esta estructura es razonable y suficiente para la mayoría de las necesidades, sin embargo existe una serie de consideraciones que la hacen poco recomendable para nuevos desarrollos:

- La información está dispersa entre dos bases de datos lógicas distintas, denominadas “principal” y “postfix”.
- No es directo qué información útil se puede extraer de estas bases de datos sin conocer su estructura interna.
- Existen múltiples campos sin mantenimiento en tablas importantes que inducen a los desarrolladores a cometer errores que no son evidentes a primera vista.
- Los nombres de los campos y tablas no siguen una nomenclatura consistente. Por ejemplo algunas tablas llevan nombres en singular y otras en plural.
- Existen copias parciales o incompletas de algunas tablas, lo que induce a error
- Las llaves foráneas entre tablas relacionadas entre sí no tienen restricción de integridad
- Las llaves foráneas pueden consistir en texto o en campos numéricos
- Algunas consultas son innecesariamente complejas. Por ejemplo el proceso de autenticar un usuario y obtener su información básica involucra cuatro tablas distintas entre las dos bases de datos existentes.

En vista de esto, se concluyó que la arquitectura actual rompe los principios de delegación y encapsulación de responsabilidades que se busca en cualquier desarrollo de software. Debido a esto se propuso extender este sistema, añadiendo estas características deseables y manteniendo retro compatibilidad con las aplicaciones existentes.

Para esto se optó por incluir un nuevo agente, denominado “Proveedor de servicios”, y que abstrae el acceso a las bases de datos “principal” y “postfix” a través del uso de servicios web (WebServices) libremente disponibles para cualquier aplicación cliente que desee utilizarlos.

Un ejemplo del uso de este agente es al momento de autenticar usuarios para poder ingresar al sistema. Para eso el usuario debe proveer su correo de la organización y su contraseña, el sistema debe verificar que el usuario existe, que su contraseña es la correcta y, de ser ese el caso, obtener los cargos del usuario dentro de la organización para comprobar si su cargo tiene permiso de acceso o no.

Bajo el diseño actual de la arquitectura de Bomberos este caso (típico entre sus aplicaciones internas) se lleva a cabo a través de un proceso engorroso sobre dos bases de datos diferentes:

1. Comprobar nombre de usuario y contraseña en la tabla “mailbox” de la base de datos “postfix”
2. Si el nombre de usuario y contraseña coinciden con las de algún usuario, obtener el campo “correopk” de esa fila.
3. Consultar la tabla “usu_mail” en la base de datos “postfix” por todas aquellas filas con el campo “correopk” obtenido anteriormente (pueden ser cero, una, o varias)
4. Si sólo una fila en “usu_mail” coincide con el campo “correopk” obtenido, obtener el campo “id_usu” de esa misma fila.
5. Consultar la tabla “usuarios” en la base de datos “principal” por la información básica del usuario (nombre, apellido, etc.).
6. Consultar las tablas “usu_cargo” y “cargos” en la base de datos “principal” para obtener los cargos del usuario en cuestión.

Bajo la nueva arquitectura basta con realizar una petición HTTP al proveedor de servicios a una URL sencilla (por ejemplo “https://webservices.bomberos.cl/login”), pasar el nombre de usuario y contraseña del usuario dentro del cuerpo del mensaje, y como respuesta se obtiene un documento XML con los datos y cargos del usuario, o bien un documento vacío en el caso de que las credenciales de autenticación sean incorrectas. Un ejemplo de esta clase de respuesta se puede ver en la Ilustración 15.

El servidor de la aplicación de gestión de material mayor propiamente tal hace uso de este proveedor de servicios para acceder a la información común de Bomberos y mantiene su propia base de datos con la información estrictamente interna del sistema a desarrollar.

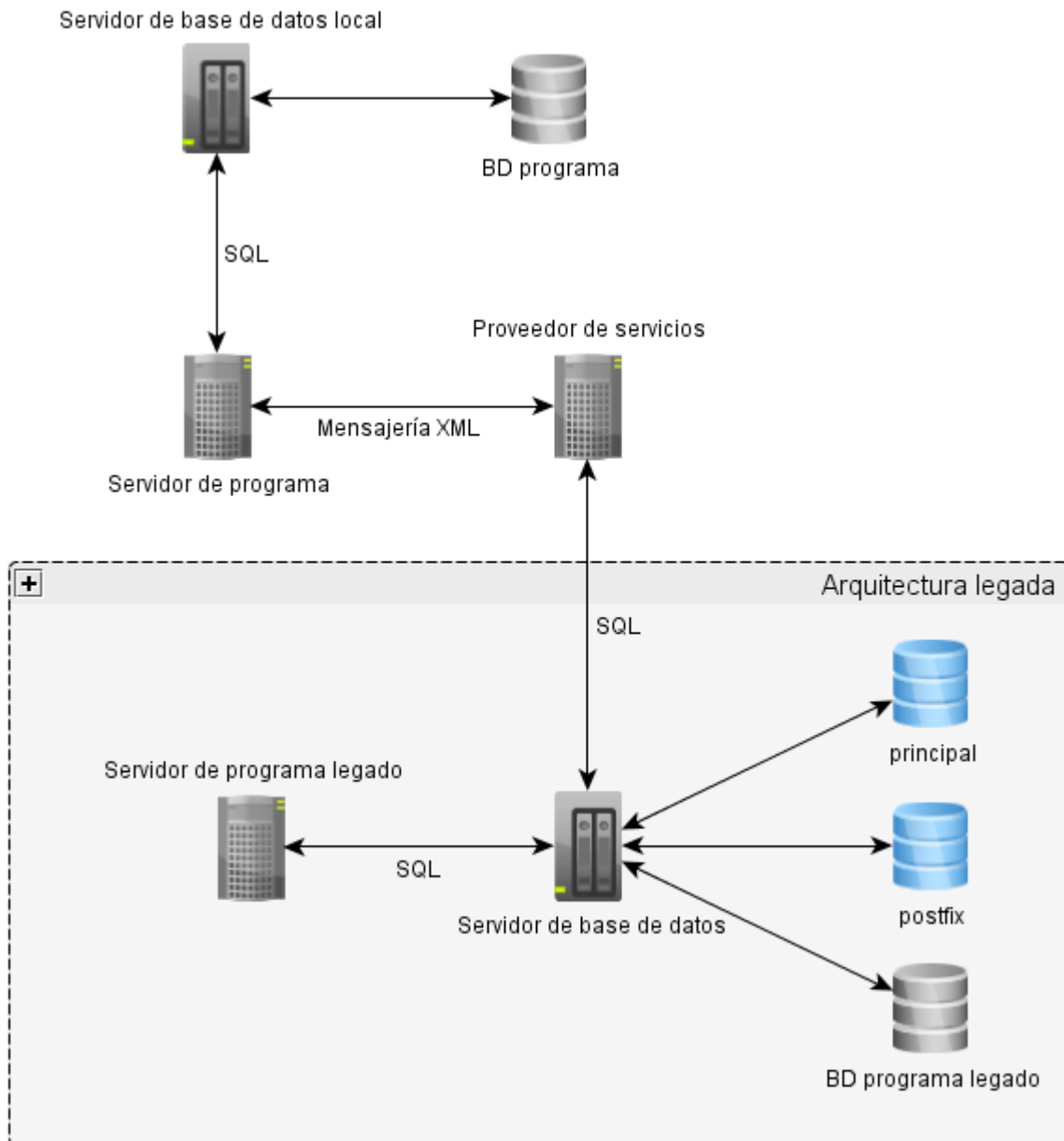


Ilustración 14 - Arquitectura física

3.2. Arquitectura del Proveedor de Servicios

Requerimientos funcionales y técnicos

El proveedor de servicios desarrollado debía cumplir con una serie de requerimientos funcionales y técnicos que definieron en gran medida las tecnologías que finalmente se usaron en su implementación. Los requerimientos funcionales más importantes fueron los siguientes:

- Ser capaz de publicar servicios web a través de un esquema sencillo de utilizar por las aplicaciones clientes.
- Abstractar por completo las complejidades y limitaciones de las bases de datos descritas anteriormente de manera transparente.

Mientras que los requisitos técnicos más importantes fueron:

- Usar protocolos de comunicación y formatos de archivo amigables con la mayor cantidad posible de entornos de desarrollo y lenguajes de programación.
- Ser fácil de probar por herramientas comunes de desarrollo para los efectos de depuración del proveedor mismo o de sus aplicaciones cliente.
- Ser extensible para permitir agregar nuevos servicios sin perjudicar a las aplicaciones clientes ya existentes.

Todas las decisiones técnicas presentadas a continuación responden a cumplir estos requerimientos de la mejor forma posible.

Protocolo y formato de comunicación

Con respecto a la comunicación entre el Servidor del programa de gestión de material mayor, el Proveedor de servicios, y el Servidor de base de datos de la JNCB (Ilustración 16 - Descripción a alto nivel del Proveedor de Servicios), éstas fueron las decisiones técnicas tomadas:

- El protocolo de comunicación entre las aplicaciones cliente y el proveedor de servicios fue HTTP, esto gracias a su uso masificado en Internet, su facilidad de implementación en todos los ambientes de programación actuales, y a la gran cantidad de herramientas que existen para publicar WebServices mediante este protocolo. Además es sencillo probar su funcionamiento con software como RESTClient (extensión para Firefox) y soapUI (aplicación independiente).
- Con respecto a las peticiones del cliente al proveedor, se optó por que se realizaran usando el esquema REST. Sin entrar en detalles acerca de sus principios de diseño, REST permite publicar los servicios deseados mediante URLs con sentido semántico y usando los métodos que ya existen en el protocolo HTTP.

Por dar un ejemplo hipotético, una petición GET en la URL `"/usuarios/3465/"` retornará los datos del usuario con identificador 3465, mientras que una petición POST en esa misma URL actualizará los datos de ese mismo usuario con los que se incluyen en el cuerpo de la petición.

En este sentido, REST es un esquema muy limpio comparado con otras tecnologías como WSDL, que generan automáticamente URLs que actúan como "endpoints" de procedimientos remotos (RPC), complejos de depurar y utilizar en aplicaciones cliente.

- Con respecto al formato de las respuestas, se decidió que XML era la opción ideal para estos fines, con un soporte masificado para su lectura en todos los lenguajes de programación y la suficiente rigurosidad técnica como para garantizar la consistencia del mensaje a enviar en todo

el proceso. XML puede ser visto como una solución engorrosa y con mala fama en algunos círculos, pero esto se debe a que usualmente se ocupa en el contexto de un contenedor de mensajes como SOAP, que es altamente verboso y complejo de manejar. En cambio, para los efectos de los servicios del proveedor se utilizó una forma muy básica de XML, fácil de interpretar y analizar, y del que se muestra un ejemplo en la Ilustración 15.

```

1  <?xml version="1.0"?>
2  <user>
3      <first_name>Tomás</first_name>
4      <last_name>Bernales</last_name>
5      <cargos>
6          <cargo>Voluntario</cargo>
7          <cargo>Comandante</cargo>
8      </cargos>
9  </user>

```

Ilustración 15 – Ejemplo de documento XML usado por la aplicación

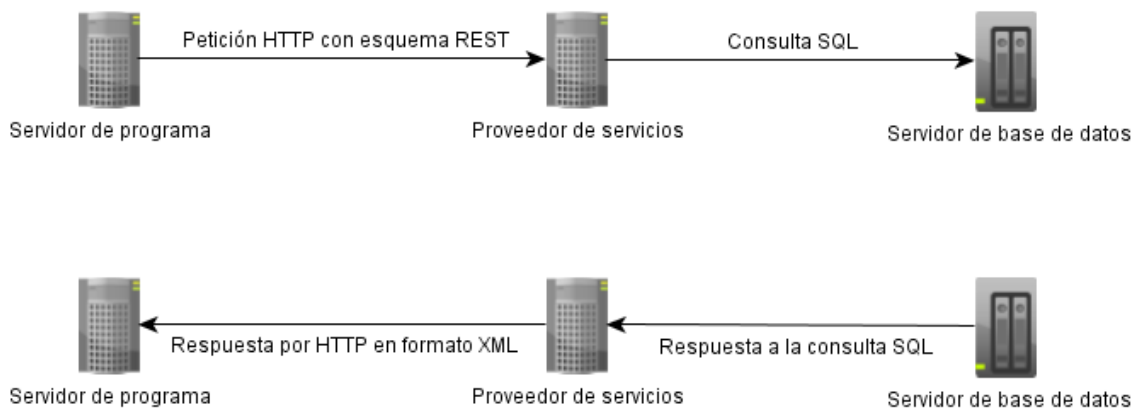


Ilustración 16 - Descripción a alto nivel del Proveedor de Servicios

Software de implementación

La implementación de los Webservice necesarios se debía apegar a las elecciones de protocolo (HTTP – REST) y de formato (XML) definidas anteriormente, además de cumplir los requerimientos técnicos expuestos al comienzo de la sección.

En este sentido el principal desafío consistió en poder abstraer una base de datos con claros problemas técnicos y poder manipularla de manera programática sin usar consultas SQL manuales dentro del Proveedor de Servicios.

Los frameworks de desarrollo modernos y de alto nivel (Ruby on Rails, Django, etc.) no son apropiados para esta tarea pues no están pensados para hacer “ingeniería reversa” sobre bases de datos ya existentes, mientras que lenguajes de programación web de más bajo nivel como PHP son engorrosos de

manejar al no estar pensados para hacer desarrollos ordenados con funcionalidades separadas en clases, además que involucraría ejecutar todas las consultas SQL manualmente.

Finalmente la decisión óptima fue usar un “stack” de software basado en Java, capaz de abstraer la base de datos utilizando el ORM Hibernate a través del patrón ActiveRecord. Bajo este patrón cada tabla en la base de datos es abstraída como una clase en una relación 1:1, en donde los campos de cada tabla representan los atributos de la clase. Hibernate en particular otorga la capacidad de abstraer sólo los atributos relevantes a la aplicación, además de corregir automáticamente la falta de llaves foráneas entre muchas de las tablas de la bases de datos “principal” y “postfix”.

El “stack” de software escogido también permite publicar servicios web usando componentes del framework “Spring”, como se muestra en la ilustración.

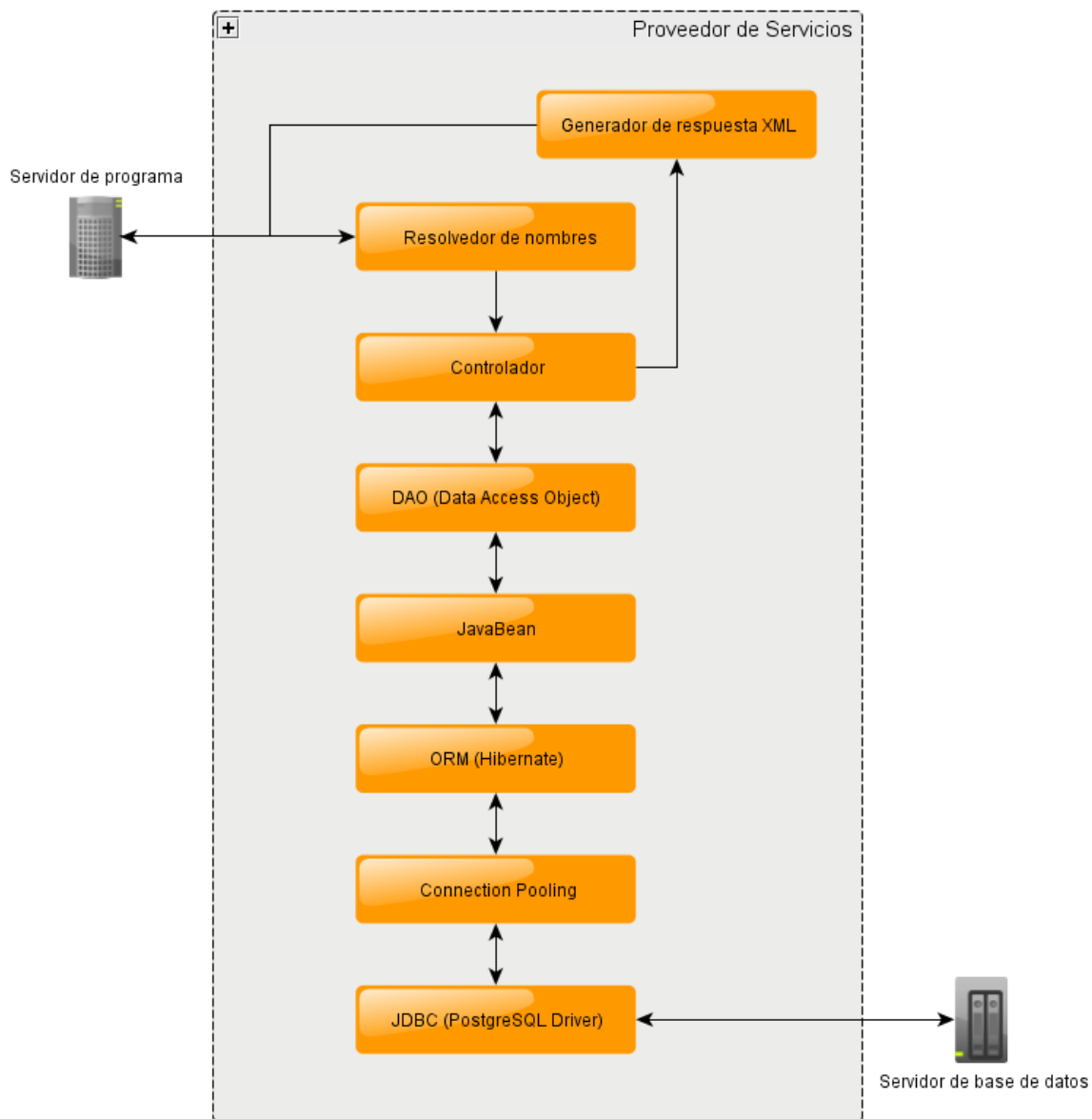


Ilustración 17 - Diseño interno del Proveedor de Servicios

Contrario a otras aplicaciones Java, la solución desarrollada no requiere de un servidor de aplicación como JBoss o WebSphere y se puede instalar fácilmente en un contenedor de servlets como Tomcat a través de un archivo WAR estándar.

El diseño expuesto anteriormente cumple con todos los requerimientos expuestos inicialmente. Por ejemplo, el ORM se encarga de abstraer la conexión con las dos bases de datos existentes mientras que los JavaBean “mapean” directamente las tablas a sus respectivas clases con nombres estandarizados y llaves foráneas estrictas.

Por su parte los servicios se exponen como métodos en los controladores que uno desee a través de un sistema de anotaciones fácil de entender para quien desee extender la aplicación.

Finalmente la generación de archivos XML es automática dependiendo del JavaBean que se desee serializar, aunque permite cierto grado de personalización si se tienen requerimientos particulares en la generación de la respuesta.

3.3. Arquitectura de la aplicación de gestión de material mayor

El diseño de la aplicación de gestión de material mayor se basa en la implementación del patrón Modelo – Vista – Controlador (MVC) tal y como se muestra en la siguiente ilustración.

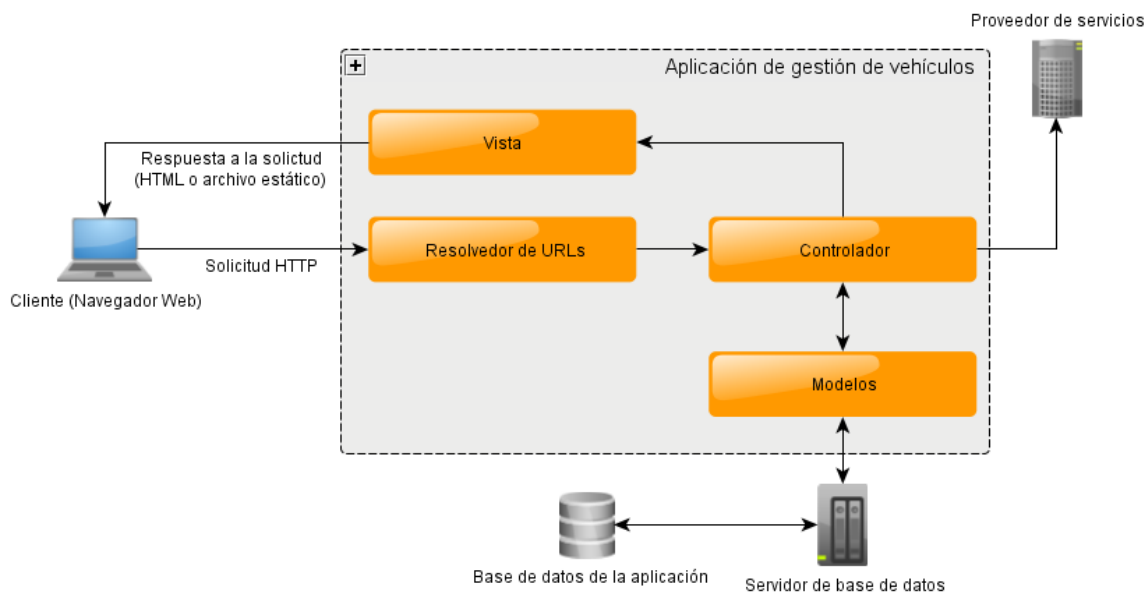


Ilustración 18 - Arquitectura de la aplicación de gestión de material mayor

En esta arquitectura, los clientes usan exclusivamente un navegador web para interactuar con la aplicación, enviando solicitudes estándar por HTTP. La aplicación recibe esta solicitud y, dependiendo de

la URL de destino, “mapea” esta solicitud a un método del controlador de la aplicación, el que la ejecuta y retorna la respuesta a través de una vista, que usualmente genera un archivo HTML para ser visualizado en el navegador del cliente.

Para resolver la solicitud, el controlador puede necesitar realizar sus propias peticiones al proveedor de servicios, además de hacer consultas a la base de datos interna de la aplicación abstraída por los modelos de la aplicación (capa de persistencia).

Finalmente, uno de los requerimientos del sistema es permitir subir y tener controles de acceso a los múltiples archivos asociados a cada material mayor. Por ejemplo, si se sube un documento asociado a un vehículo se desea que sólo el personal de la JNCB y del cuerpo de Bomberos del carro sea capaz de acceder a dicho documento.

Normalmente esto es un inconveniente pues en la mayoría de los casos el servidor web (Apache o IIS, por ejemplo) sirve directamente los archivos binarios sin pasar por la lógica de la aplicación, por lo que en principio son accesibles por cualquier persona (con o sin autenticación) que conozca la ruta del archivo.

Para solucionar esto se optó por resolver las solicitudes de archivos directamente por la aplicación. De esta manera las peticiones de archivos binarios (fotografías y documentos) son redirigidas a la aplicación, esta ejecuta la lógica de verificar si el usuario puede o no acceder a dicho archivo, y lo retorna si la autenticación es exitosa.

3.4. Modelo de Datos

El modelo de datos subyacente a la aplicación de gestión de material mayor es complejo e involucra un número considerable de tablas, pero es lo suficientemente modular como para ser analizado por partes.

Antes de detallar cada módulo es necesario considerar que todos comparten una única tabla “interface_materialmayor”, que guarda los datos básicos de los vehículos ingresados al sistema, es por este motivo que dicha tabla aparece en varios de los diagramas subsecuentes.

Tablas básicas

Contienen la información básica de la base de datos, lo que incluye:

- Información geográfica (regiones, provincias, comunas)
- Información de los cuerpos y compañías ya disponible en los sistemas de Bomberos
- Información de los usuarios, sus cargos dentro de la jerarquía de Bomberos (Capitanes, Comandantes, etc.) y sus roles dentro de la JNCB si corresponde (Operaciones Bomberiles, Jurídica, Adquisiciones, etc.).

Toda esta información (exceptuando los roles) proviene de los sistemas actuales de la JNCB a través del proveedor de servicios descrito anteriormente.

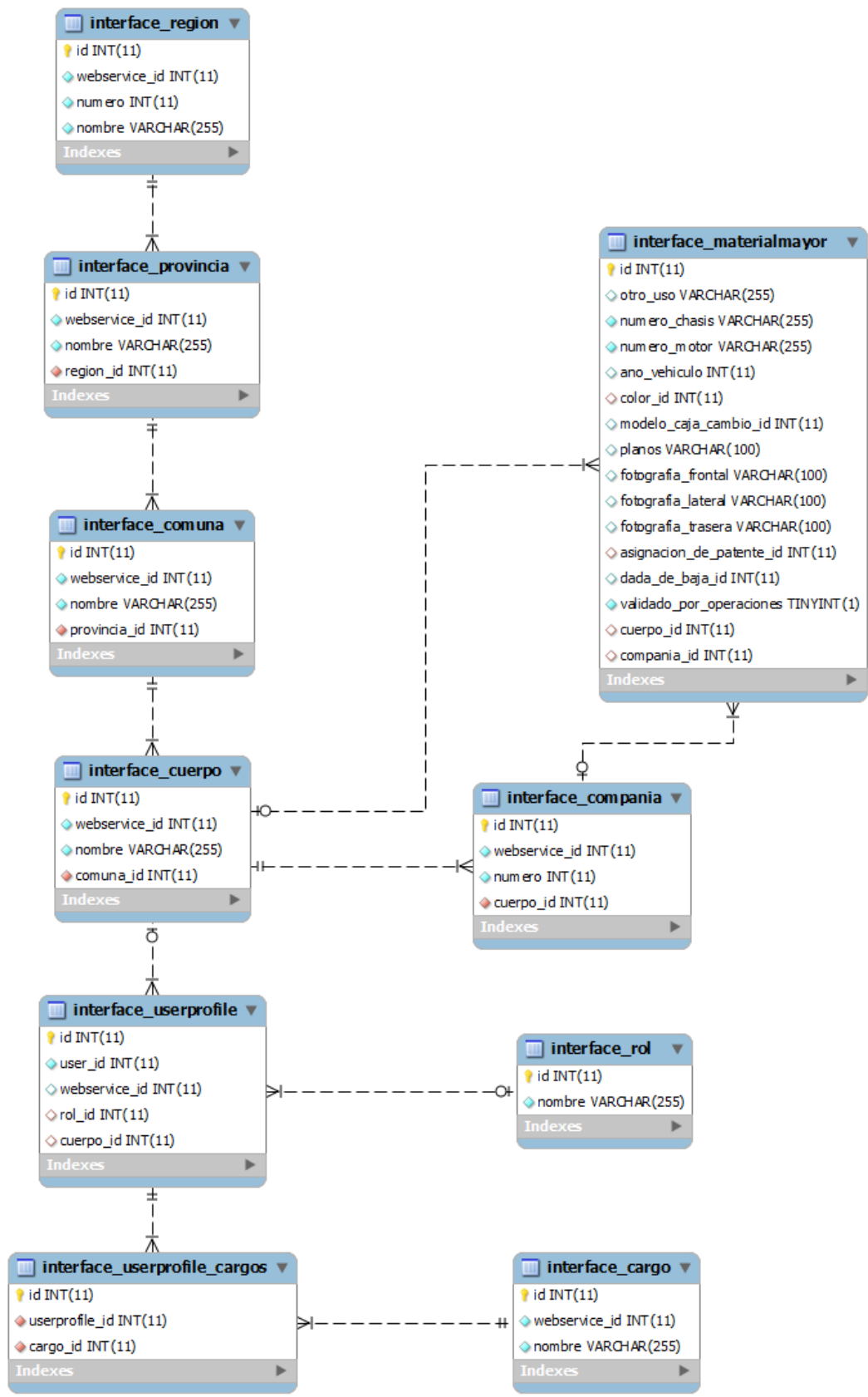


Ilustración 19 - Modelo de Datos de tablas básicas

Información de Alta

Cuando un material mayor es dado de alta se debe rellenar un formulario para definir sus características técnicas, como lo son su marca de chasis, el uso que se le va a dar, etc. El almacenamiento de esta información se detalla en la Ilustración 20 – Modelo de datos de información técnica del material mayor, mientras que un ejemplo del formulario que se rellena para estos fines se puede ver en la Ilustración 31 - Interfaz para dar de alta un nuevo material mayor.

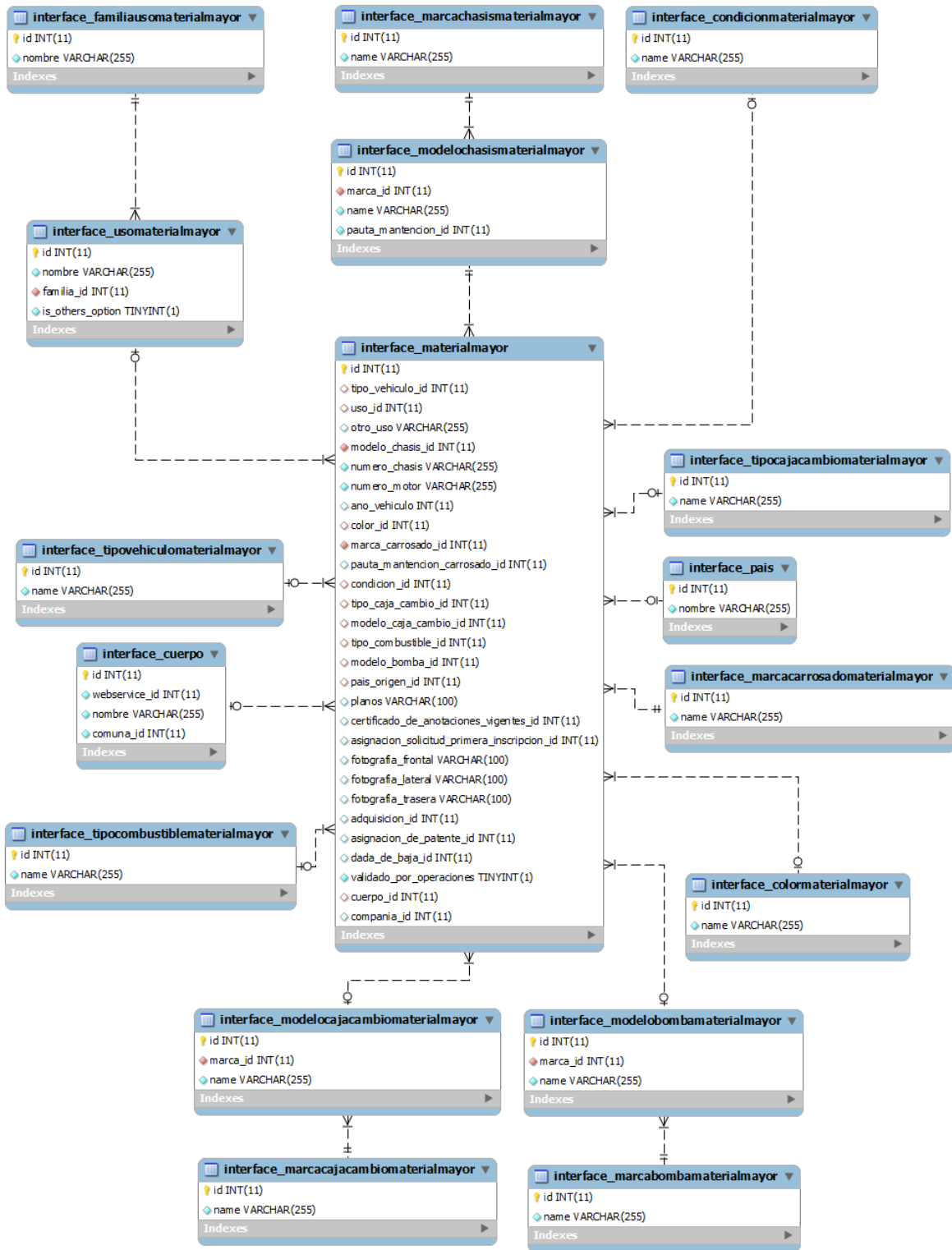


Ilustración 20 – Modelo de datos de información técnica del material mayor

Información de Adquisición

Como parte del proceso de dar de alta un vehículo se debe especificar si el carro fue adquirido mediante una compra o una donación. Para representar esta dualidad en la estructura de la base de datos se implementó una aproximación de *polimorfismo*, una habilidad de la programación orientada a objetos (OOP).

Esta propiedad es muy útil para la aplicación pues permite modelar una situación compleja como la presentada en este caso (un vehículo tiene una adquisición, pero no se sabe *a priori* si es por compra o por donación) manteniendo los beneficios de usar una base de datos relacional con restricciones estrictas.

Sin la abstracción de polimorfismo no se podría tener una única llave foránea desde la tabla "materialmayor", pues existen dos tablas independientes que representan distintos medios de adquisición. En ese caso se debería optar por eliminar la llave foránea y realizar un control a nivel de aplicación de a qué tabla ("adquisicioncompra" o "adquisiciondonacion") se refiere cada entrada en la tabla "materialmayor". Este control es muy propenso a errores y no garantiza integridad de la información. Una segunda alternativa es tener dos llaves foráneas en "materialmayor" (una para cada tabla de formas de adquisición) con la posibilidad de tener valores nulos, sin embargo esta opción es propensa a errores de integridad (por ejemplo, que las dos llaves foráneas sean valores nulos)

Utilizando la abstracción de polimorfismo es posible mantener todas las llaves foráneas del modelo de datos y representar de manera más intuitiva la información. La aplicación aún tiene que ejecutar un control (y una consulta SQL adicional) para obtener los datos concretos de adquisición de un vehículo, pero a un nivel mucho más básico que el necesario en el caso de no usar polimorfismo.

Para conseguir este comportamiento se agregó una tabla intermedia común para los dos tipos de adquisiciones ("interface_adquisicionmaterialmayor") y se agregaron llaves foráneas hacia esta nueva tabla desde la que representa al material mayor y las que representan los tipos concretos de adquisición.

Como las bases de datos relacionales no tienen inherentemente la capacidad de hacer polimorfismo de tablas se debe agregar una última tabla que funcione como índice ("interface_modoadquisicionmaterialmayor") y referenciada por la tabla intermedia.

Con esta estructura se puede obtener programáticamente la adquisición concreta asociada a un vehículo manteniendo la integridad estructural de la base de datos y sin agregar complejidad considerable al algoritmo de búsqueda o al modelo de datos. Esta técnica es usada en múltiples otras oportunidades en el diseño de la base de datos.

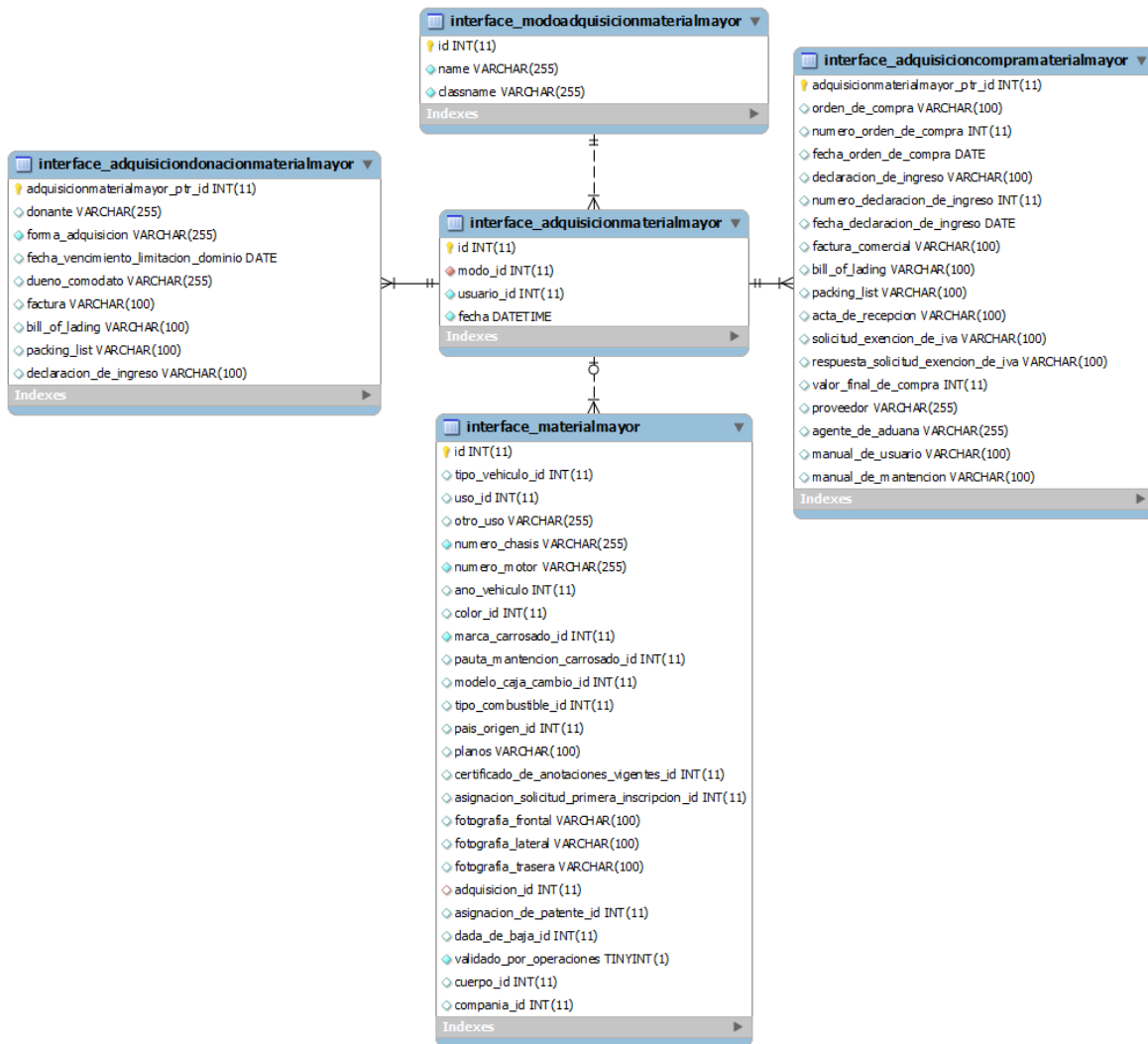


Ilustración 21 – Modelo de datos de información de adquisición del material mayor

Información de mantención de material mayor

El sistema de mantenciones de material mayor está separado en dos partes y será abordado de esta misma manera.

La primera parte (tablas en la parte superior de la Ilustración 22 - Modelo de datos de mantenciones de material mayor) definen a qué pautas de mantención se somete cada vehículo del sistema. En este sentido un carro siempre tiene dos pautas de mantención asociadas, la primera a su carrosado y la segunda a su chasis, dependiendo del modelo de chasis del vehículo.

Para los efectos del sistema una pauta de mantención no es más que un conjunto de operaciones de mantención, en donde cada operación es una instrucción en texto plano y una frecuencia que define cada cuanto tiempo se tiene que realizar (“cada 6 meses”, por ejemplo).

La segunda parte (tablas en la parte inferior) corresponde a la ejecución de una orden de mantención en particular y funciona de la siguiente manera.

Día a día el sistema informático monitorea los vehículos de la flota de Bomberos, y cuando detecta que corresponde una o más operaciones de mantención sobre alguno de ellos se genera una orden mantención, compuesta por el conjunto de operaciones de mantención que corresponde hacerle al vehículo en ese momento en particular.

El sistema archiva el estado de ejecución de cada una de estas operaciones (“ejecutada”, “pospuesta” o “sin resolver”) y además permite asociar archivos binarios a la orden de mantención, lo que es útil por si se desean subir cotizaciones, boletas o facturas asociadas al mantenimiento.

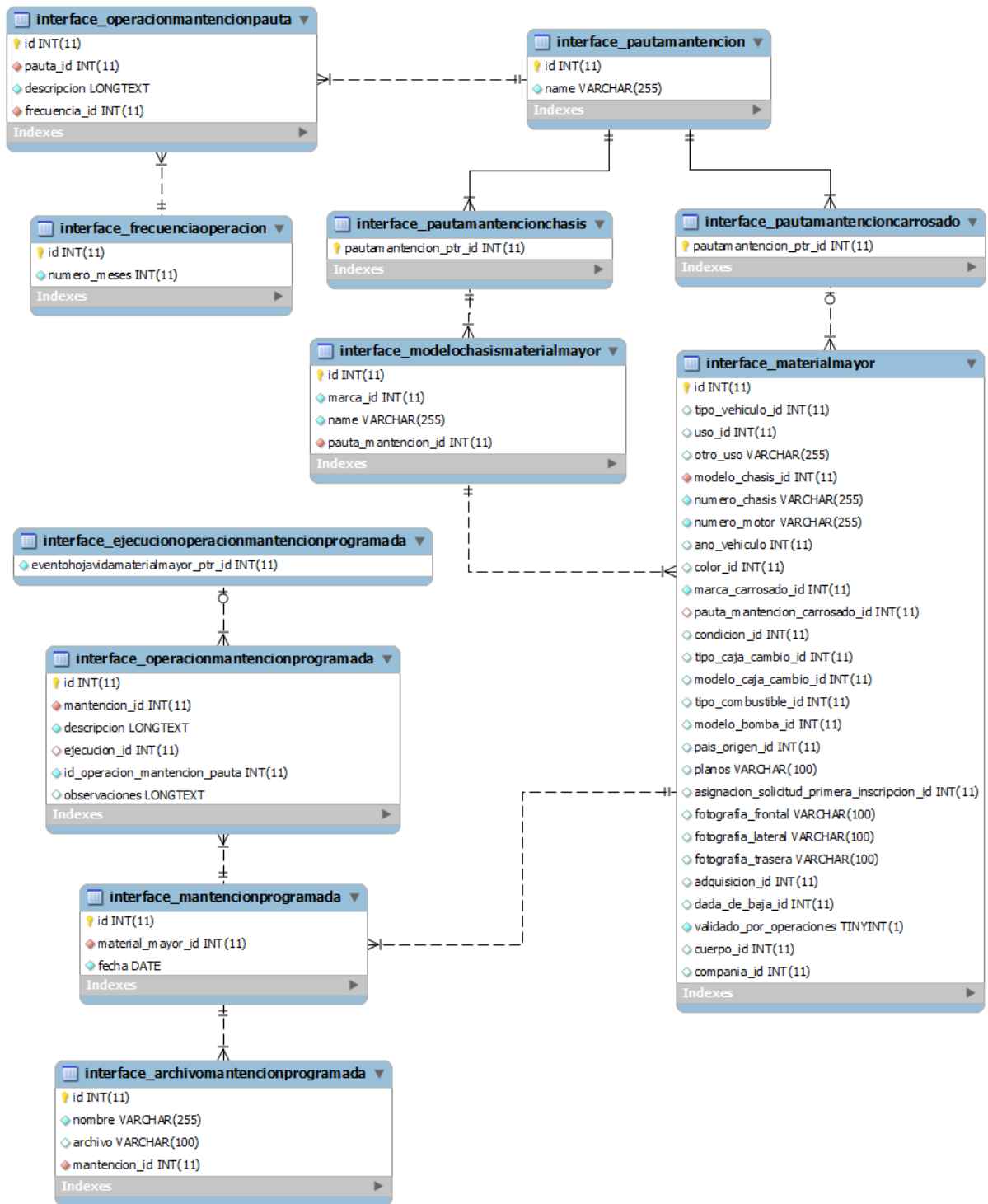


Ilustración 22 - Modelo de datos de manteniones de material mayor

Procesos secundarios sobre el material mayor

Además de los procesos anteriormente descritos, un vehículo de Bomberos puede pasar por varios otros procedimientos de menor importancia, pero que igualmente son implementados en el sistema informático.

A continuación se listan estos procesos y se muestran en la Ilustración 23 - Modelo de datos de procesos secundarios.

- Cambio de número de chasis
- Cambio de número de motor
- Asignación de solicitud de primera inscripción
- Asignación de placa patente
- Asignación/reasignación a cuerpo
- Asignación/reasignación a compañía
- Cambio de certificado de anotaciones vigentes
- Dada de baja

Por lo general estos procesos sólo involucran un formulario web y poca información adicional, por lo que no se entra en detalles sobre cada uno.

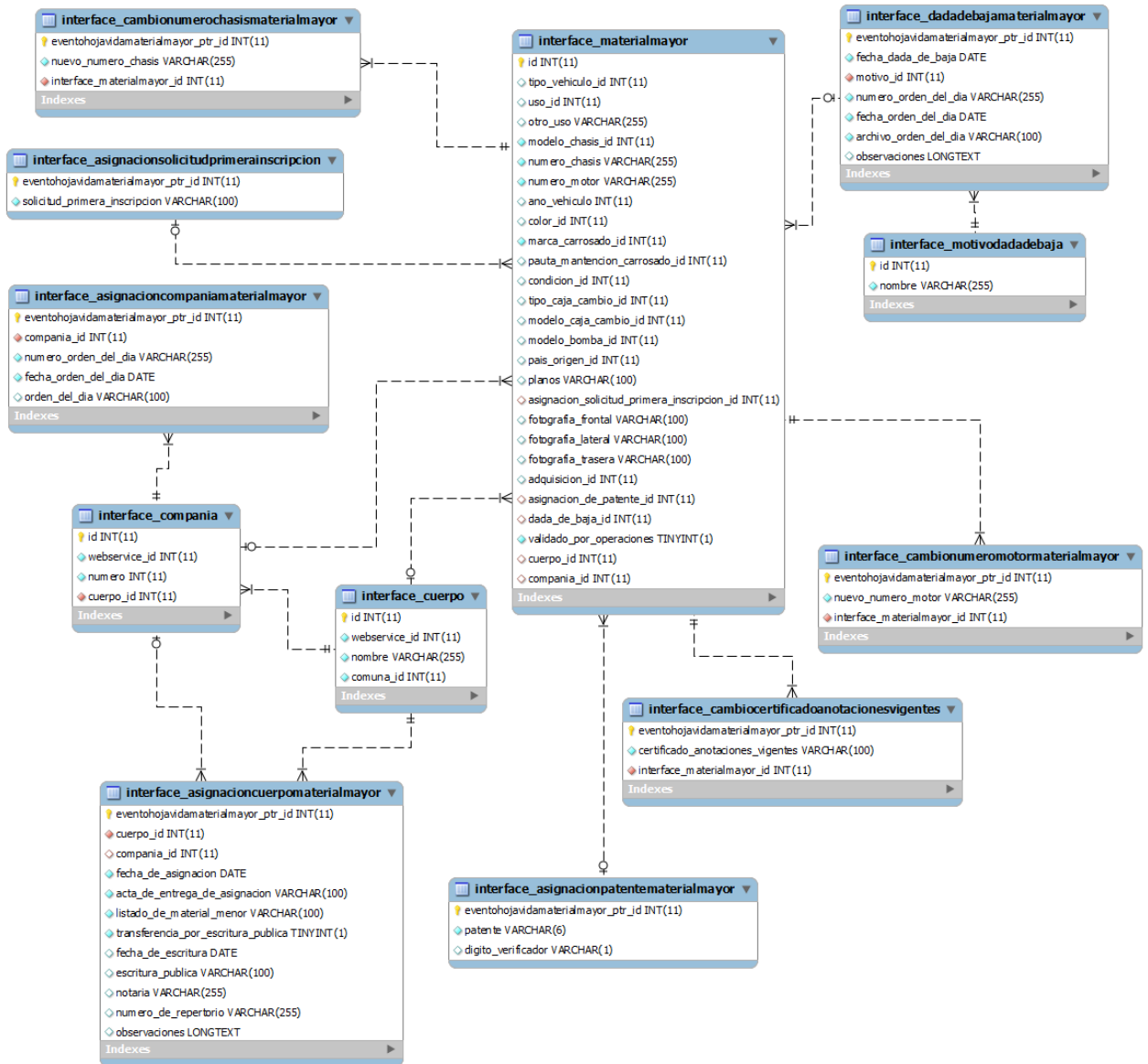


Ilustración 23 - Modelo de datos de procesos secundarios

4. Diseño detallado de la aplicación de gestión de material mayor

En la sección anterior se presentó a grandes rasgos la arquitectura de la aplicación de la aplicación de gestión de material mayor y sus componentes principales, basada en el patrón modelo / vista / controlador (MVC).

En esta sección se detallan con más cuidado los componentes con los que se implementará este patrón y las relaciones que existen entre ellos. Primero se presenta un diagrama parcial con las clases que conforman al sistema.

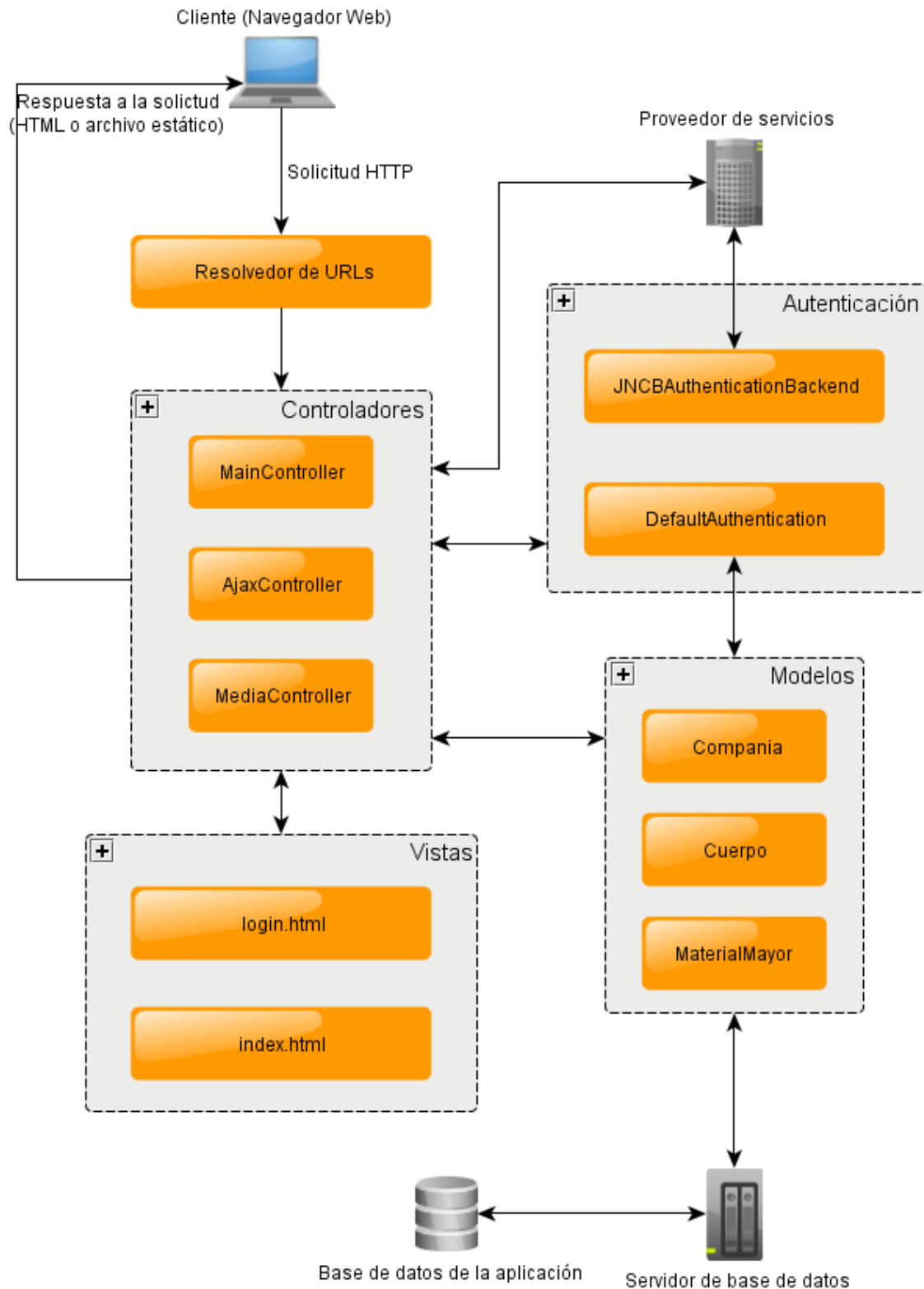


Ilustración 24 - Diseño detallado de la aplicación de gestión de material mayor

Bajo esta arquitectura, cada petición realizada por el cliente es convertida por el Resolvedor de URLs (provisto por el framework de desarrollo) en una llamada a función en alguno de los controladores provistos. En el contexto de la aplicación de gestión de material mayor se consideraron tres tipos de controladores:

MainController: Controlador que responde todas las consultas por páginas web (contenido HTML) y las peticiones de persistir cambios (guardar nuevos vehículos, generar pautas de mantención, etc.)

AjaxController: Controlador que responde la solicitud de recursos asíncronos (en formato JSON) por parte del navegador cuando se implementa dinamismo en las páginas web usando JavaScript. Por ejemplo un formulario dinámico puede cargar en tiempo real las compañías que corresponden al cuerpo seleccionado en un campo específico para facilitarle al usuario la entrada de información.

Se escogió el formato JSON para esta comunicación pues es muy fácil de procesar por JavaScript, al contrario de XML.

MediaController: Controlador que se encarga de resolver todas las peticiones de contenido binario (fotografías y documentos en formato PDF) asociados a cada vehículo en el sistema. Esta capa es necesaria para implementar un control de acceso flexible a dicha información.

Todos los controladores necesitan poder realizar consultas a la base de datos. Estas consultas son abstraídas por los Modelos, correspondientes a clases que encapsulan cada una de las tablas y campos de la base de datos usando el patrón ActiveRecord (detallado en la sección 3.2), que permite hacer las consultas más frecuentes (SELECT, INSERT, UPDATE y DELETE) a través de métodos de alto nivel. En el diagrama solo se muestran tres de estas clases, pero en la implementación final existen cerca de cincuenta.

El control de autenticación de usuarios se realiza a través de dos clases principales. DefaultAuthentication corresponde a la implementación tradicional incluida con la mayoría de los frameworks de desarrollo, en la cual éste crea su propio registro interno de usuarios. En la aplicación de gestión de material mayor esta clase es usada para mantener a los usuarios a nivel de JNCB y sus permisos particulares.

Paralelamente existe la clase de autenticación JNCBAuthenticationBackend, que se encarga de validar a los usuarios que ya existen en los registros de la JNCB a través de los servicios web provistos por la aplicación de webservices descrita anteriormente.

Finalmente, el controlador principal debe generar un gran número de páginas en formato HTML. Para evitar mezclar lógica del dominio del problema con la de generación de documentos se recurre al uso de vistas (o “templates”) en formato HTML, que corresponden básicamente a documentos estáticos con espacios definidos para que el controlador inserte el contenido que corresponde en ellos.

5. Implementación de la solución

La implementación de la memoria se separa en dos componentes esenciales, el proveedor de servicios y el sistema de gestión de material mayor propiamente tal. En consecuencia se detalla la implementación de la solución para cada uno por separado.

5.1. Implementación del Proveedor de servicios

En la sección de Arquitectura del Software ya se discutió en gran detalle acerca de la decisión de tecnologías para la implementación del proveedor de servicios. En función de esta discusión se optó por un “stack” de software basado en el entorno Java compuesto por los siguientes elementos:

- Lenguaje de programación: Java 1.6
- Framework de desarrollo: Spring 3.0.5
- ORM de conexión con la base de datos: Hibernate 3.5
- Conector con la base de datos: Driver JDBC oficial para PostgreSQL
- Plataforma de producción objetivo: Tomcat 7.0.24

Las versiones del software utilizadas responden a que varias de estas componentes ya estaban instaladas en los servidores de la JNCB (Java y Tomcat), mientras que para el resto se optó por aquella más reciente, con mejor soporte, documentación y/o apoyo de la comunidad.

Uno de los requisitos técnicos fundamentales de la solución fue que el sistema debiese ser fácil de instalar, esto con tal de evitar los clásicos problemas de la industria para desplegar aplicaciones Java. Estas dificultades se dan pues todas las aplicaciones Java deben ser instaladas en un servidor apropiado (por ejemplo Tomcat, Glassfish o Weblogic) y además que todas las bibliotecas de las que depende la aplicación (Spring, Hibernate, etc.) deben estar disponibles en el *classpath* del proyecto en sus versiones específicas. Este proceso es más complejo que la configuración para levantar, por ejemplo, un servicio web hecho en PHP o Perl.

Con este fin se apuntó a poder correr la aplicación sobre un contenedor de servlets relativamente sencillo (Tomcat) sin configuraciones adicionales, y además se tomó la decisión de empaquetar la aplicación como un archivo WAR que contuviese la aplicación misma y todas sus dependencias. Este archivo se puede instalar fácilmente desde el entorno web gráfico que viene incluido con Tomcat, y después de su instalación basta con definir el nombre de usuario y contraseña para el acceso a las bases de datos del sistema.

Para comprobar el proceso de instalación en producción del sistema se levantó una máquina virtual con el mismo sistema operativo utilizado en los servidores Linux de la JNBC (Debian 6.0 “Squeeze”) y se documentaron los pasos concretos a seguir para el despliegue del proveedor de servicios desde una instalación limpia del sistema operativo.

En la práctica la instalación y configuración de la aplicación fueron directos, sin embargo se detectaron problemas en la configuración de la instalación de Tomcat existente que debieron ser corregidos antes de poder continuar. Además se debió configurar con cuidado un nuevo usuario de PostgreSQL con permisos de lectura exclusivamente sobre las tablas utilizadas por la aplicación, que además pudiese acceder a dichas tablas de forma externa desde un servidor distinto al de la base de datos.

5.1.1. Estadísticas de código

A continuación se presentan los datos cuantitativos más importantes del código utilizado para implementar el proveedor de servicios.

Número de clases

JavaBeans (abstracción de la base de datos): 16 clases

DAO (Data Access Object): 9 clases

Controladores: 7 clases

Adaptadores (usados en la generación de documentos XML): 4 clases

Número de líneas de código

Java (lógica de la aplicación): 1416 líneas

XML (configuración): 252 líneas

5.2. Implementación de la aplicación de gestión de material mayor

En las secciones anteriores ya se discutió en extenso acerca de los requisitos, arquitectura y diseño de la aplicación de gestión de material mayor propiamente tal.

Como resultado de este proceso, se definieron los siguientes puntos de complejidad críticos de la aplicación:

- Una base de datos extensa (más de 50 tablas) y que aplica técnicas modernas de desarrollo (polimorfismo en bases de datos relacionales)
- Un extenso número de procesos de variada complejidad, que involucraban decenas de formularios, cada uno con validación independiente.
- La necesidad de manejar la subida de múltiples archivos binarios (documentos y fotografías) y someterlos a un sistema de permisos de acceso.
- La necesidad de poder incluir tareas programables a nivel de sistema operativo.
- La necesidad de leer y generar archivos en formato XLS (Excel)

En particular para el último requisito, es necesario poder leer archivos XLS para procesar las pautas de mantención de material mayor subidas por el personal de la JNCB, y también es necesario poder generar archivos XLS pues se solicitó la capacidad de poder exportar algunas de las interfaces web (por ejemplo el resultado de la búsqueda de vehículos) como archivos Excel.

Estas observaciones permitieron concluir que el uso de un lenguaje de bajo nivel como PHP, incluso con un framework como CakePHP o CodeIgniter, sería muy lento para los efectos de desarrollo, además de no abordar de manera transparente el manejo de formularios.

Las soluciones basadas en Java y .NET, si bien son muy estables y son las únicas plataformas que ofrecen un desarrollo en lenguajes fuertemente tipados, en la práctica son engorrosas para el desarrollo de aplicaciones orientadas al usuario final y aún no adoptan los paradigmas más recientes de programación, haciendo complejo implementar algunas de las características deseadas.

Como resultado de este proceso de descarte se concluyó que la alternativa más apropiada para abordar la implementación del sistema era usar un framework de desarrollo de alto nivel basado en un lenguaje de programación dinámico. En este sentido las alternativas más conocidas son aquellas basadas en Ruby (con el framework Ruby on Rails) y en Python (con el framework Django).

Ambos entornos de desarrollo se adaptaban de forma adecuada a los requisitos técnicos impuestos para la implementación de la solución, por lo que la decisión final quedó a criterio del programador.

Considerando la experiencia previa usando Django en proyectos personales y en una aplicación previa de la JNCB (Catastro de Cuerpos y Compañías de Bomberos de Chile), finalmente se optó por este framework, al conocer de antemano la manera concreta de abordar la mayoría de los puntos críticos del listado anterior. En particular el framework soluciona de manera muy elegante los problemas de polimorfismo en la base de datos y de representación y validación de formularios.

Para la implementación propiamente tal se seleccionó el siguiente conjunto de herramientas y bibliotecas, nuevamente un punto medio entre el software ya disponible en los servidores de la JNCB y aquel propuesto por el desarrollador.

- Lenguaje de programación: Python 2.5
- Framework de desarrollo: Django 1.3.1
- Base de datos: PostgreSQL 8.3
- Bibliotecas para el manejo de imágenes (fotografías): PIL 1.1.7 y sorl-thumbnail 11.09
- Bibliotecas para el manejo de archivos XLS: xlrd 0.7.1 y xlwt 0.7.2

Finalmente, el despliegue se realizó sobre un servidor web Apache utilizando la biblioteca mod_wsgi para permitirle correr una aplicación Django, por lo que no se debieron instalar servicios adicionales en el servidor.

Como Python es en particular un lenguaje de scripting, resultó sencillo implementar un “script” que ejecuta el proceso de generación de nuevas órdenes de mantención. Este script es llamado desde un crontab (administrador de procesos a intervalos regulares de UNIX) una vez al día de manera muy sencilla y transparente.

Este programa revisa la fecha en que se dio de alta cada vehículo de la flota de Bomberos y la compara con la fecha de hoy. Posteriormente verifica las pautas de mantención del vehículo en cuestión, y si determina que existen mantenciones que deben realizarse sobre el carro entonces genera las órdenes de mantención y alertas correspondientes.

Por ejemplo, si un vehículo fue dado de alta el 1 de febrero de 2012 y se ejecuta el programa el 1 de agosto de 2012, el sistema determinará que han pasado seis meses desde que se dio de alta el vehículo, y que por consiguiente corresponde ejecutar las mantenciones que le corresponden cada dos, tres y seis meses.

5.2.1. Estadísticas de código

A continuación se presentan los datos cuantitativos más importantes del código utilizado para implementar la aplicación de gestión de material mayor.

Número de clases

Modelos: 53 clases

Formularios: 40 clases

Controladores: 54 clases

Número de líneas de código

Python (lógica de la aplicación): 3460 líneas

HTML (“templates” para la generación de la interfaz web): 1954 líneas

CSS (Hojas de estilo para la interfaz web): 127 líneas

JavaScript (lógica sencilla ejecutada en el navegador del usuario): 697 líneas

5.3. Flujo de uso

A continuación se muestra la ejecución de algunos de los procesos más importantes involucrados en el sistema tal y como fueron implementados en la aplicación final.

Login

Al entrar por primera vez a la aplicación el sistema automáticamente redirige al usuario a la página de autenticación. En ésta debe ingresar su nombre de usuario y contraseña típicos dentro de la organización (para el caso de los usuarios dentro de los cuerpos) o bien el nombre de usuario y contraseñas de un usuario específico del sistema (para el caso de usuarios de la JNCB).



Ingreso al sistema

Nombre de usuario:

Contraseña:

(c) 2011 Junta Nacional de Cuerpos de Bomberos de Chile
Esquema de templates provisto por [Blueprint](#)

Ilustración 25 - Página de autenticación de usuarios

En este caso se entra al sistema como un usuario con el rol de personal del departamento de Operaciones Bomberiles, haciendo notar que la navegación dentro de la página es contextual al cargo o rol que tiene el usuario.



Sistema de Gestión de Material mayor

Bienvenido al sistema de gestión de material mayor de Bomberos de Chile.

(c) 2011 Junta Nacional de Cuerpos de Bomberos de Chile
Esquema de templates provisto por [Blueprint](#)

Ilustración 26 - Página de inicio del sistema

Subir pauta de mantención de chasis o carrosado

Para subir una nueva pauta de mantención de chasis o carrosado, el usuario navega a la sección correspondiente y selecciona “Agregar nueva pauta”. En este ejemplo se subirá una pauta de mantención de carrosado basada en el mismo archivo que el mostrado en la Ilustración 8 - Ejemplo de pauta de mantención.



Pautas de mantención de carrosado

Pautas de mantención de carrosado

[Agregar nueva pauta](#)

(c) 2011 Junta Nacional de Cuerpos de Bomberos de Chile
Esquema de templates provisto por [Blueprint](#)

Ilustración 27 - Lista de pautas de mantención de carrosado antes de subir una nueva pauta



[Pautas de mantención de carrosado](#) » [Agregar pauta](#)

Agregar pauta de mantención de carrosado

Pauta de mantención (en formato Excel):

(c) 2011 Junta Nacional de Cuerpos de Bomberos de Chile
Esquema de templates provisto por [Blueprint](#)

Ilustración 28 - Interfaz para agregar una nueva pauta de mantención de carrosado

Una vez subida, la pauta es agregada al listado y se puede editar o eliminar.



[Pautas de mantención de carrosado](#)

Pautas de mantención de carrosado

[Agregar nueva pauta](#)

[Pauta de mantención de carrosado para vehículos Renault](#)

(c) 2011 Junta Nacional de Cuerpos de Bomberos de Chile
Esquema de templates provisto por [Blueprint](#)

Ilustración 29 - Lista de pautas de mantención de carrosado después de subir una nueva pauta



Pautas de mantención de carrosado » Pauta de mantención de carrosado para vehículos Renault

Editar pauta de mantención de carrosado

Nombre:

Operaciones de la pauta

| Nombre | Frecuencia | ¿Eliminar? |
|--|---------------------------------------|--------------------------|
| <input type="text" value="Vaciar motor"/> | <input type="text" value="12 meses"/> | <input type="checkbox"/> |
| <input type="text" value="Vaciar circuito refrigerante y cambio de tapon presión/depresión"/> | <input type="text" value="24 meses"/> | <input type="checkbox"/> |
| <input type="text" value="Cambio de cartucho del filtro de aceite del motor y limpieza de la cuva"/> | <input type="text" value="12 meses"/> | <input type="checkbox"/> |
| <input type="text" value="Cambio de cartucho(s) de filtro(s) de combustible y"/> | <input type="text" value="12 meses"/> | <input type="checkbox"/> |

Ilustración 30 - Interfaz para editar pautas de mantención

Dar de alta vehículo

Para dar de alta un vehículo se debe escoger el formulario adecuado a su origen (compra o donación) y llenar la información solicitada.














[Dar de alta](#) » [Alta de vehículo \(compra\)](#)

Alta de material mayor (Compra)

Los campos marcados con * son obligatorios

Datos del vehículo

| | |
|----------------------|---|
| Tipo de vehículo | <input type="text" value="-----"/>   |
| Uso | <input type="text" value="-----"/>   |
| Marca chasis (*) | <input type="text" value="-----"/>   |
| Modelo de chasis (*) | <input type="text" value="Por favor seleccione una marca"/>   |
| Número de chasis (*) | <input type="text"/> |
| Número de motor (*) | <input type="text"/> |
| Año del vehículo | <input type="text" value="-----"/>  |
| Color | <input type="text" value="-----"/>   |

Información Adicional

| | |
|------------------------|--|
| Marca de carrosado (*) | <input type="text" value="-----"/>   |
|------------------------|--|

Ilustración 31 - Interfaz para dar de alta un nuevo material mayor

Una vez dado de alta, el vehículo puede ser ubicado a través del navegador de material mayor y acceder a todos sus datos técnicos, archivos asociados y procesos (reasignación a cuerpo, dar de baja, etc.).



Material mayor

Navegador de material mayor

Región Cuerpo

¿Incluir material dado de baja?

[Descargar esta búsqueda como Excel](#)

| Marca chasis | Modelo chasis | Marca carrosado | N° chasis | N° motor | Cuerpo | Compañía | Editar |
|--------------|---------------|-----------------|-----------|------------------|-------------|-------------|------------------------|
| Renault | MCB80 | Camiva | AB3518 | 1749279293485738 | No asignado | No asignado | Editar |

(c) 2011 Junta Nacional de Cuerpos de Bomberos de Chile
Esquema de templates provisto por [Blueprint](#)

Ilustración 32 - Navegador de material mayor



Material mayor » Renault MCB90 - N° chasis AB3518

Editar material mayor

Información general

| | |
|-------------------------------------|---|
| Fecha de ingreso al sistema | 23 de enero de 2012 a las 21:27 |
| Estado actual | En servicio |
| Cuerpo destinatario original | Talca (Región del Maule) |
| Asignación actual | Nivel central JNBC (Asignar) |
| Patente | Sin patente (Asignar patente) |
| Certificado de anotaciones vigentes | No posee certificado (Actualizar certificado) |
| Solicitud de primera inscripción | No posee solicitud (Asignar) |
| Pauta de mantenimiento de carrosado | Pauta de mantención de carrosado para vehículos Renault (Cambiar pauta) |
| Ingresado por | Alejandro Mansilla |
| Forma de adquisición | Compra (Detalles) |
| Hoja de vida | Ver hoja de vida |
| Mantenciones programadas | Ver mantenciones programadas |

[Eliminar material mayor](#)

[Dar de baja](#)

Ilustración 33 - Detalles de material mayor

Resolver orden de mantención

Cuando se genera una nueva orden de mantención, esta aparecerá en la sección de “Mantenciones programadas pendientes”, desde donde se puede acceder al listado concreto de operaciones que se le deben realizar al vehículo en particular.



Mantención » Mantenciones programadas pendientes

Mantenciones programadas pendientes

| Vehículo | Cuerpo / compañía | Fecha de la mantención | Acciones |
|----------------------------------|----------------------|------------------------|--------------------------|
| Renault MCB90 - N° chasis AB3518 | JNBC - Nivel central | 2013/01/23 | Resolver |

1/1 10

(c) 2011 Junta Nacional de Cuerpos de Bomberos de Chile
Esquema de templates provisto por [Blueprint](#)

Ilustración 34 - Listado de mantenciones pendientes



Material mayor » Renault MCB90 - N° chasis AB3518 » Mantenciones programadas » Mantenimiento de 2013-01-23

Mantenimiento de 2013-01-23

[Descargar esta pauta como Excel](#)

| Operación | Estado | |
|---|-----------|--|
| Vaciar motor | Pendiente | Marcar como ejecutada Posponer |
| Cambio de cartucho del filtro de aceite del motor y limpieza de la cuva | Pendiente | Marcar como ejecutada Posponer |
| Cambio de cartucho(s) de filtro(s) de combustible y limpieza de la cuva | Pendiente | Marcar como ejecutada Posponer |
| Cambio de prefiltro de combustible | Pendiente | Marcar como ejecutada Posponer |
| Cambio rodillo tensor del alternador | Pendiente | Marcar como ejecutada Posponer |

Ilustración 35 - Detalle de mantención pendiente

Desde esta interfaz las operaciones se pueden marcar como ejecutadas o posponerlas, además de subir archivos asociados a todo el proceso.



Material mayor » Renault MCB90 - N° chasis AB3518 » Mantenciones programadas » Mantenimiento de 2013-01-23

Mantenimiento de 2013-01-23

[Descargar esta pauta como Excel](#)

| Operación | Estado | |
|---|-----------|--|
| Vaciar motor Observaciones: Operación realizada exitosamente | Ejecutada | Ver detalles |
| Cambio de cartucho del filtro de aceite del motor y limpieza de la cuva Observaciones: Mantención pospuesta por falta de recursos | Pospuesta | Marcar como ejecutada |
| Cambio de cartucho(s) de filtro(s) de combustible y limpieza de la cuva | Pendiente | Marcar como ejecutada Posponer |
| Cambio de prefiltro de combustible | Pendiente | Marcar como ejecutada Posponer |
| Cambio rodillo tensor del alternador | Pendiente | Marcar como ejecutada Posponer |

Ilustración 36 - Resolución de mantención programada pendiente

6. Resultados esperados y obtenidos

Al momento de iniciar el trabajo de memoria se planteó un objetivo central ambicioso: darle a Bomberos de Chile una herramienta de administración útil y poderosa para gestionar una flota de más de 2.500 vehículos de los más diversos tipos. Este sistema además permitiría administrar virtualmente todos los eventos de la vida útil de un carro, desde el momento en que se da de alta hasta que se da de baja, pasando por todas sus mantenciones programadas y extraordinarias.

Detrás de este sistema se esperaba un compromiso de la JNCB y Bomberos de Chile en general para darle uso, cargar los datos necesarios y mantener la herramienta actualizada día tras día, además de considerar útil el sistema automatizado de generación de órdenes de mantención.

Finalmente se esperaba que todo el proceso fuese un ejemplo para los futuros desarrollos informáticos de la JNCB con respecto a su calidad de código, documentación y proceso de despliegue, de manera tal que sirva como referencia para formalizar la estrategia informática de la organización.

Con la implementación del trabajo finalizada, y al hacer una comparación entre los resultados esperados y los obtenidos, se pueden ver varios contrastes. El objetivo principal de modelar todos los procesos de los carros de Bomberos de Chile probó ser muy extenso para los fines de un trabajo de memoria de tres meses, por lo que se debió acotar a los eventos más importantes en la vida del vehículo y sus mantenciones programadas, dejando de lado sus eventos extraordinarios como choques o volcamientos.

Como contrapeso, el compromiso percibido desde la JNCB de sacar el sistema adelante es notable, y ya se está ejecutando el proceso de digitación mediante el cual se ingresarán todos los vehículos actuales de Bomberos a la herramienta para empezar a darle uso. Este esfuerzo es acompañado por un trabajo en conjunto con los principales proveedores de Bomberos (Camiva, Crimson Fire, etc.) para estandarizar las pautas de mantención de sus vehículos para apegarse al formato utilizado por el sistema. Finalmente es menester mencionar la contratación de un encargado de material mayor como parte del departamento de Operaciones Bomberiles, y que va a monitorear el uso del sistema a nivel nacional.

Con el proceso de ingreso de datos en marcha se ha podido ver como los usuarios de la aplicación interactúan con el sistema. Inicialmente mostraban cierta molestia cuando se encontraban con algún error pequeño que les impedía trabajar de manera continua, pero a medida que estos fueron corregidos se llegó a un punto en el que no se notificaron más inconvenientes en el software. Un ejemplo de estos errores es que los campos dinámicos del formulario para dar de alta material mayor en ocasiones no se actualizaban correctamente, esto por un “bug” en la lógica implementada en JavaScript para esa interfaz.

Los encargados administrativos de la JNCB se han mostrado conformes con los objetivos conseguidos, confiando en que es una herramienta a la que se le dará utilidad por un largo periodo. Finalmente existe la plena confianza de que el diseño y arquitectura de la herramienta permitirá que se le puedan agregar las funcionalidades que quedaron fuera del alcance de esta memoria sin implicar un impacto importante para los datos que están siendo ingresados en el sistema.

7. Conclusiones y trabajo futuro

Antes de abordar el tema de memoria, y como parte de otro proyecto de desarrollo para la JNCB, se tuvo la oportunidad de conocer de primera mano la contradictoria situación que vive Bomberos de Chile. Por una parte entrega un servicio importantísimo al país y a sus habitantes, ganándose el respeto de la sociedad por su entrega y abnegación a una labor peligrosa. Por otra parte, viven de recursos relativamente escasos para los enormes gastos que significa mantener las operaciones de Bomberos a nivel nacional, lo que limita enormemente las inversiones que pueden hacer en áreas que no están relacionadas directamente con el combate de incendios y rescate de personas.

Esta dualidad ha generado dentro de la JNCB una serie de problemas administrativos, en particular aquel de gestión de material mayor abordado en esta memoria, y que le han costado millones de pesos a la organización o, peor aún, la vida de una víctima de un accidente que no pudo ser socorrida oportunamente o la de un voluntario que no contaba con un equipamiento adecuado.

Para ayudar a paliar esta situación este trabajo de memoria desarrolló un sistema que permite hacer gestión sobre cada uno de los 2.500 vehículos de Bomberos de Chile a lo largo del país, desde que son adquiridos hasta que son dados de baja, apoyando en particular sus procesos de reparación y mantenimiento técnico.

A través del uso de este sistema la Junta Nacional de Bomberos podrá en el mediano plazo mejorar notoriamente la disponibilidad de sus carros para atender actos de servicio, y en el largo plazo extender su vida útil considerablemente. En último término todos estos beneficios mejorarán el uso de recursos por parte de la JNCB y permitirán que Bomberos realicen un trabajo más eficiente y seguro.

El diseño de la aplicación también permitirá extraer reportes importantes con respecto a las fallas técnicas de sus vehículos y a los gastos incurridos por ellas, lo que le permitirá a la Junta Nacional contar con datos concretos al momento de negociar con sus proveedores de material mayor, en particular con respecto a sus plazos de garantía con servicio técnico garantizado.

Al finalizar esta memoria sólo puedo esperar que Bomberos de Chile considere útil y valiosa la herramienta que acá se entrega, y que esta vaya más allá de ser simplemente el camino para obtener un título profesional.

Como se planteó anteriormente, quedaron varios objetivos en el camino para favorecer la calidad del sistema entregado, pero que será necesario abordarlos en el futuro. El primero de ellos es permitir la consideración de eventos extraordinarios en la vida útil de un carro (accidentes de tránsito, volcamientos, etc.), pues estos requieren de financiamiento especial de parte de la JNCB, que muchas veces no es coordinado por la falta de comunicación entre el cuerpo de Bomberos correspondiente y la Junta Nacional.

El segundo objetivo, y como parte de un sistema de más amplio alcance, es permitir gestionar el material menor (mangueras, pitones, etc.) a bordo de cada vehículo, lo que le permitirá a la central de alarmas de cada cuerpo despachar rápidamente el carro óptimo para las necesidades de un incidente en particular en función del equipamiento que lleva a bordo.

8. Bibliografía

1. Asociación de Bomberos de Chile, URL: <http://www.abomach.cl/>. Última visita: Febrero de 2012.
2. Bomberos de Chile, URL: <http://www.bomberos.cl/>. Última visita: Febrero de 2012.
3. Jakob Nielsen and Marie Tahir. Homepage Usability: 50 Websites Deconstructed. NewRiders Publishing. 2001.
4. Wikipedia. Junta Nacional de Cuerpos de Bomberos de Chile, URL: http://es.wikipedia.org/wiki/Bomberos_de_chile/. Última visita: Febrero de 2012.
5. Roger Pressman. Software Engineering: A Practitioner's Approach (Hardcover), Seventh Edition, McGraw-Hill. January 2009.
6. Cuerpo de Bomberos de Ñuñoa. Sistema de clasificación de emergencias. Documento Interno, 2011.