UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

TEMA:

"ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL VEHICULAR EN INSTITUCIONES PÚBLICAS"

Autor: Egda. Patricia Alejandra Terán Sevilla

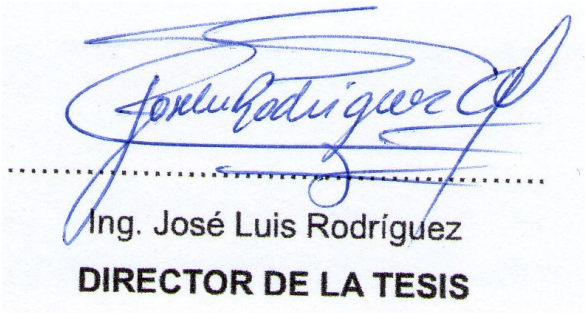
Director: Ing. José Luis Rodríguez

Ibarra – Ecuador

2012

CERTIFICACIÓN

Certifico que la Tesis **"ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL VEHICULAR EN INSTITUCIONES PÚBLICAS"**, ha sido realizada en su totalidad por la Señorita Patricia Alejandra Terán Sevilla, portadora de la cédula de identidad número 1002558060.



Ing. José Luis Rodríguez

DIRECTOR DE LA TESIS

EMAPSA-E- S- 2012 –67
EL Ángel, 15 de Julio de 2012

CERTIFICACION

La Srta. Terán Sevilla Patricia Alejandra, con Nro. de cédula de Ciudadanía 100255806-0 estudiante de la **Universidad Técnica del Norte, UTN-** Facultad de Ingeniería en Ciencias Aplicadas, de la carrera Ingeniería en Sistemas de décimo semestre; ha realizado su Trabajo de Grado, Desarrollo del Sistema del Control Vehicular, en la EMPRESA MUNICIPAL DE AGUA POTABLE Y SANEAMIENTO AMBIENTAL DEL CANTÓN ESPEJO EMAPSA-E; cumpliendo con todos los requisitos reglamentarios de aprobación de la Institución, con cualidades de responsabilidad y profesionalismo.

A tal efecto, se extiende el presente CERTIFICADO DE CULMINACIÓN DE TRABAJO DE GRADO, en la ciudad de El Ángel a los 15 días del mes de Julio de 2012.

Atentamente

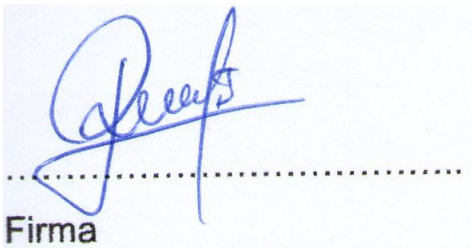
Lcdo. Alejandro Jojoa
GERENTE GENERAL DE EMAPSA-E

EMPRESA MUNICIPAL DE AGUA POTABLE Y SANEAMIENTO AMBIENTAL - ESPEJO
GERENCIA

UNIVERSIDAD TÉCNICA DEL NORTE

CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, Patricia Alejandra Terán Sevilla, con cédula de identidad Nro. 1002558060, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículos 4, 5 y 6, en calidad de autor del trabajo de grado denominado: **"ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL VEHICULAR EN INSTITUCIONES PÚBLICAS"**, que ha sido desarrollado para optar por el título de Ingeniería en Sistemas Computacionales en la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En mi condición de autor me reserva los derechos morales de la obra antes citada. En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Técnica del Norte.



Nombre: Patricia Alejandra Terán Sevilla

Cédula: 1002558060

Ibarra, a los doce días del mes de Noviembre del 2012

UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN
A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

La Universidad Técnica del Norte dentro del proyecto Repositorio Digital Institucional, determinó la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

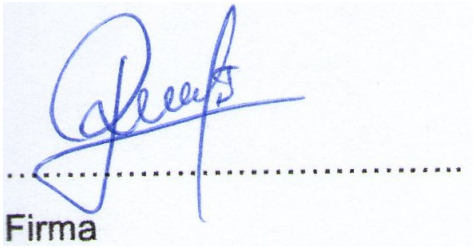
Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1002558060		
APELLIDOS Y NOMBRES:	TERÁN SEVILLA PATRICIA ALEJANDRA		
DIRECCIÓN:	GRAL. JULIO ANDRADE 4-95 Y JUAN F. BONILLA		
EMAIL:	brujitapats@hotmail.com		
TELÉFONO FIJO:	062953680	**TELÉFONO MOVIL:**	0984472550

DATOS DE LA OBRA	
TÍTULO:	"ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL VEHICULAR EN INSTITUCIONES PÚBLICAS"
AUTOR:	EGDA. PATRICIA ALEJANDRA TERÁN SEVILLA
FECHA:	2012/11/12
PROGRAMA:	PREGRADO
TITULO POR EL QUE OPTA:	INGENIERÍA EN SISTEMAS COMPUTACIONALES
DIRECTOR:	ING. JOSÉ LUIS RODRÍGUEZ

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Patricia Alejandra Terán Sevilla, con cédula de identidad Nro. 1002558060, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión, en concordancia con la Ley de Educación Superior Artículo 144.



Nombre: Patricia Alejandra Terán Sevilla
Cédula: 1002558060
Ibarra, a los doce días del mes de Noviembre del 2012

DEDICATORIA

Dedico este trabajo a mis padres y mis hermanos por su incondicional apoyo para la culminación de mi carrera.

Y de manera especial a mi novio Pablo por estar siempre ahí en los momentos buenos y malos y siempre incentivarme para conseguir la titulación de mi carrera.

AGRADECIMIENTO

Agradezco a Dios en primer lugar por darme la sabiduría y el conocimiento para llegar a la culminación de mi carrera, a mi madre por ser el apoyo en los momentos más difíciles, a todos los profesores que a lo largo de mi carrera me fueron llenando de conocimiento para llegar a ser una profesional y por último a mi querida Universidad Técnica del Norte por abrirme sus puertas para formarme en tan prestigiosa Institución.

TABLA DE CONTENIDOS

ÍNDICE DE CUADROS

ÍNDICE DE GRÁFICOS

RESUMEN

La Empresa Municipal de Agua Potable y Saneamiento Ambiental del Cantón Espejo es una institución del sector público encargada de la prestación de servicios de agua potable y saneamiento ambiental, encaminada a preservar la salud de los habitantes y obtener una rentabilidad social y económica en sus inversiones.

La empresa es responsable de la administración, planificación, diseño, construcción, control y mantenimiento de los sistemas para producción, distribución, comercialización de Agua Potable y Alcantarillado; así como el tratamiento de las aguas residuales.

Los vehículos pertenecientes al sector público, y a las entidades de derecho privado que disponen de recursos públicos, están destinados exclusivamente para uso oficial, es decir, para el desempeño de funciones públicas, en los días y horas laborables, y no podrán ser utilizados en fines personales, ni familiares, ajenos al sector público, ni en actividades electorales y políticas.

Para el desarrollo de la aplicación se utilizaron herramientas y tecnologías de software libre como son: PostgreSQL, Symfony Framework, PHP y Java IReports con lo cuál se beneficia directamente la empresa en cuanto a la reducción del costo de licencias.

La arquitectura que se utilizó es el Modelo Vista Controlador; en el lado del servidor se utilizó Debian GNU/Linux, se dispone tanto del servidor de base de datos como el de aplicaciones y se cuenta con toda la infraestructura necesaria para su correcto funcionamiento; por su parte el cliente únicamente necesita un navegador web.

Para la movilización de los vehículos oficiales, fuera del cantón Espejo, las Órdenes de Movilización serán emitidas por el señor Gerente de la empresa, y tendrá una vigencia no mayor de 5 días hábiles.

El sistema informático permite conocer la disponibilidad de un vehículo en una determinada fecha, disponer de la cantidad adecuada de lubricantes y combustibles en los vehículos, mantenimientos oportunos, mejorando la vida útil de los vehículos.

SUMMARY

The “Empresa Muncipal de Agua Potable y Saneamiento Ambiental” is a public sector institution responsible for the provision of safe water and sanitation, aimed at preserving the health of people and obtain social and economic return on their investments.

The company is responsible for administration, planning, design, construction, inspection and maintenance of systems for production, distribution, marketing of Water and Wastewater, and the treatment of wastewater.

Vehicles belonging to the public sector, and private law entities that have public resources, are intended exclusively for official use, ie, for the performance of public functions, in the days and hours, and may not be used for purposes personal, family or, outside the public sector, or in political and electoral activities.

For application development tools and technologies were used a free software such as: PostgreSQL, Symfony Framework, PHP and Java iReports thereby directly benefit the company in terms of reducing the cost of licenses.

The architecture used is Model View Controller, on the server side was used Debian GNU / Linux, you have both the database server and the application and has all the necessary infrastructure for proper operation; by Meanwhile the client only needs a web browser.

For the mobilization of official vehicles, out of the corner mirror, mobilization orders will be issued by the Manager of the company, and will have been issued within 5 working days.

The computer system enables the availability of a vehicle on a given date, to have the right amount of lubricants and fuels in vehicles, timely maintenance, improving the life of the vehicles.

CAPÍTULO I

1 INTRODUCCIÓN

1.1 ANTECEDENTES

En la ciudad de El Ángel, Cantón Espejo, Provincia del Carchi, se constituyó el 15 de febrero del 2005, la Empresa Municipal de Agua Potable y Saneamiento Ambiental del Cantón Espejo, como persona jurídica de derecho público con autonomía administrativa, operativa, financiera y patrimonial, la misma que se regirá por las normas de la Ley Orgánica de Régimen Municipal, la presente Ordenanza que regula la prestación de los servicios de Agua Potable y Saneamiento Ambiental, las disposiciones de los reglamentos internos generales y específicos que se expidan y demás normas jurídicas aplicables[1].

La empresa se constituye por la presente ordenanza y se denominará Empresa Municipal de Agua Potable y Saneamiento Ambiental del Cantón Espejo, cuyas siglas son EMAPSA-E, y por ello, con este mismo nombre se identificará en todos los actos públicos, privados, judiciales, extrajudiciales y administrativos.

La EMAPSA-E ejercerá su acción en el cantón Espejo, provincia del Carchi, teniendo competencia para todo lo relacionado con la prestación de los servicios de Agua Potable, Alcantarillado y Saneamiento Ambiental, dentro del plan cantonal de desarrollo. Estos servicios que se inician en la ciudad de El Ángel, podrán extenderse a las parroquias rurales del cantón y otras jurisdicciones del régimen autónomo y entidades públicas o privadas, dedicadas a la prestación de ellos. Todo esto a través de convenios legalmente suscritos.

La Empresa tiene como objetivo la prestación de los servicios de Agua Potable y Saneamiento Ambiental, tendiente a preservar la salud de los habitantes y obtener una rentabilidad social y económica en sus inversiones.

---

1 Tomado del Registro Oficial No. 032 de fecha Lunes 6 de Junio de 2005.

1.1.1 ORGANIGRAMA ESTRUCTURAL DE EMAPSA-E

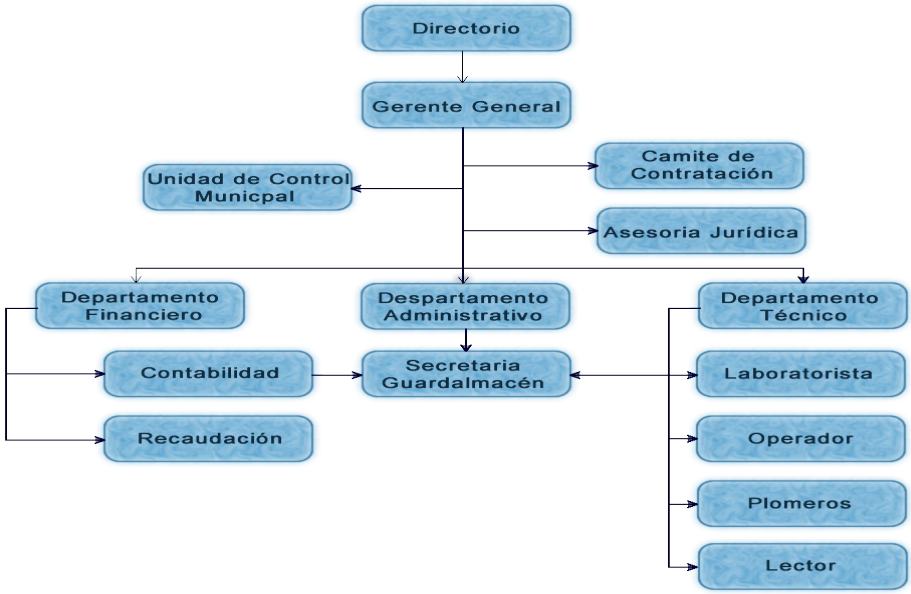


Ilustración 1: *Organigrama Estructural EMAPSA-E*

Fuente: *EMAPSA-E*

1.1.2 VISIÓN Y MISIÓN DE LA EMAPSA-E

1.1.2.1 VISIÓN DE EMAPSA-E

La Empresa Municipal de Agua Potable y Saneamiento Ambiental del Cantón Espejo, prevee en cinco años, bajo el concepto de calidad total convertirse en una Empresa Líder de la Región Norte del País.

1.1.2.2 MISIÓN DE EMAPSA-E.

Contribuir al mejoramiento de la calidad de vida de la población del Cantón Espejo, a través de la prestación eficiente del Servicio de Agua Potable.

1.2 PROBLEMA

La Empresa Municipal de Agua Potable y Saneamiento Ambiental (EMAPSA-E), tiene como finalidad la captación, tratamiento, distribución, producción y venta de agua potable y la prestación de los servicios de alcantarillado a la ciudad El Ángel y las parroquias rurales (27 de Septiembre, La Libertad, San Isidro, El Goaltal) del cantón Espejo, garantizando eficiencia y eficacia, con criterio de equidad y justicia, comprometida con una concepción ecológica que preserve las cuencas hidrográficas y proteja el medio ambiente de todo el Cantón.

EMAPSA-E es una empresa, que brinda servicios de agua potable y alcantarillado a la población del cantón Espejo, proyectándose siempre a la satisfacción del cliente.

También tiene como finalidad dar el debido mantenimiento al sistema regional así como disponer de un sistema de agua potable que garantice cantidad y calidad las veinte y cuatro horas de servicio.

La Empresa Municipal de Agua Potable y Saneamiento Ambiental cuenta con el Departamento de Proyectos, Departamento de Medio Ambiente y Laboratorio; estos departamentos son los encargados de desarrollar proyectos para la implementación de nuevas redes de agua potable como de alcantarillado, juntamente con los otros departamentos se encargan de desarrollar plantas de tratamiento de aguas residuales así como del manejo de todos los residuos sólidos a la vez el Laboratorista es el encargado de realizar un monitoreo diario del agua para realizar el análisis correspondiente para saber cuál es el estado del agua.

Por ser una empresa con apenas cinco años, esta no cuenta con todos los sistemas informáticos necesarios que ayuden a optimizar sus procesos y cumplir con las normas requeridas por las entidades de control gubernamentales.

La Empresa Municipal de Agua Potable y Saneamiento Ambiental en la actualidad el control vehicular lo realiza de una manera informal y no acorde a las exigencias de las entidades de control gubernamentales; y lleva los procesos de control vehicular de forma manual.

El deficiente manejo de los inventarios de vehículos sus accesorios y herramientas provoca la falta de información en tiempo real, difícil acceso a registros históricos y menor control de existencias posibilitando la generación de artículos obsoletos.

Las órdenes de movilización, partes de novedades y accidentes se realizan de forma manual y en hojas pre impresas, lo cual conlleva a un registro desordenado e inexacto de la información.

La emisión de las órdenes de provisión de combustibles y lubricantes se lo realiza de forma manual y en hojas pre impresas dificultando llevar un control, lo cual puede provocar un desmesurado o escaso pedido de suministros de los mismos.

El inadecuado manejo de la administración de los vehículos de la institución, dificulta el registro de entrada / salida de los mismos; y su entrega – recepción a los conductores.

Las consecuencias de no disponer a futuro de una solución informática para el control vehicular es que al seguir llevando un manejo manual esto podría causar pérdida, inconsistencia y confusión al momento de requerir la información.

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

- Analizar, Diseñar e Implementar un Sistema de Control Vehicular para la Empresa Municipal de Agua Potable y Saneamiento Ambiental del Cantón Espejo utilizando herramientas de software libre.

1.3.2 OBJETIVOS ESPECÍFICOS

1. Recolectar los requerimientos funcionales y no funcionales.

2. Realizar la planificación del desarrollo del proyecto.

3. Definir el modelo del negocio, modelo de objetos del negocio y modelo del dominio.

4. Implementar la solución desarrollada en la Empresa Municipal de Agua Potable y Saneamiento Ambiental del Cantón Espejo.

5. Controlar mediante código de barras la emisión de solicitudes.

6. Realizar las pruebas y estabilización de la aplicación informática.

7. Realizar la documentación necesaria del sistema.

1.4 JUSTIFICACIÓN

Los vehículos pertenecientes al sector público, y a las entidades de derecho privado que disponen de recursos públicos, están destinados exclusivamente para uso oficial, es decir para el desempeño de funciones públicas, en los días y horas laborables, y no podrán ser utilizados en fines personales, ni familiares, ajenas al sector público, ni en actividades electorales y políticas.

Para la movilización de los vehículos oficiales, fuera de la sede donde los funcionarios ejercen habitualmente sus funciones, las Órdenes de Movilización serán emitidas por la máxima autoridad o el servidor delegado para el efecto que podrá ser el Jefe de Transportes y tendrán una vigencia no mayor de 5 días hábiles.

Al contar con un sistema informático de control vehicular se podrá conocer la disponibilidad de un vehículo en una determinada fecha, disponer de la cantidad adecuada de lubricantes y combustibles en los vehículos, mantenimientos oportunos, mejorando la vida útil de los vehículos con un gran beneficio para la empresa. Además, al tener un mantenimiento adecuado de los vehículos se reduce de manera significativa las emisiones de contaminantes atmosféricos de origen vehicular.

Por lo antes expuesto se hace imperiosa la necesidad de contar con una solución tecnológica para la realización de dicho control.

1.5 ALCANCE

Para el desarrollo del sistema de control vehicular se utilizarán las siguientes tecnologías:

Servidor Web: Apache HTTP Server 2.2
Lenguaje de Programación: PHP 5.3
Sistema de Gestión de Base de Datos Relacional: PostgreSQL 9.1
Framework: Symfony PHP Web 1.4
Entorno de Desarrollo: NetBeans 7.0

1.5.1 ARQUITECTURA MODULAR

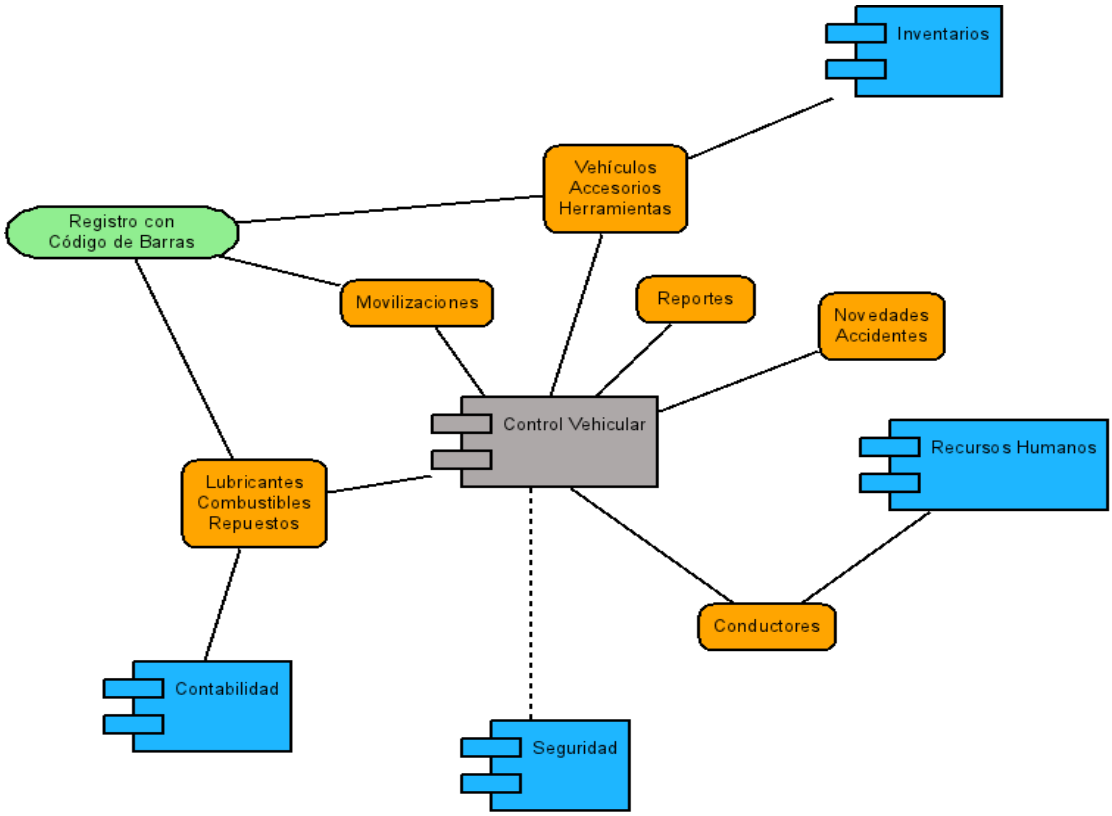


Ilustración 2: *Arquitectura Modular*

Fuente: *Propia*

ROLES DE USUARIO

Administrador: Encargado del control de los servidores, y del módulo de seguridad.

Gerente: Acceso a todos reportes.

Secretaria: Ingreso de vehículos, accesorios y herramientas; y emisión de las órdenes de movilización.

Contadora: Emisión de órdenes de consumo de lubricantes, combustibles y repuestos.

Recursos Humanos: Ingreso de Conductores.

1.5.2 ARQUITECTURA TECNOLÓGICA

La arquitectura que se utilizará es Modelo Vista Controlador (MVC); en el lado del servidor se utilizará Linux y se tendrá tanto el servidor de base de datos como el de aplicaciones, y contará con toda la infraestructura necesaria para su correcto funcionamiento; por su parte del lado del cliente se necesitará únicamente que estos cuenten con un Navegador Web compatible.

LIMITACIONES:

- La aplicación informática será implementada en la Empresa Municipal de Agua Potable y Saneamiento Ambiental del Cantón Espejo.

- Para el desarrollo de la aplicación se utilizarán las herramientas y tecnologías que los encargados de la parte informática de la institución lo dispongan.

- La aplicación será parte del Sistema de Planificación de Recursos Empresariales (Sistema Integrado) que se encuentra en la etapa de desarrollo en la institución.

BENEFICIOS DE LA IMPLEMENTACIÓN DEL SISTEMA

Al contar con un sistema informático de control vehicular se podrá conocer la disponibilidad de un vehículo en una determinada fecha, disponer de la cantidad adecuada de lubricantes y combustibles en los vehículos, mantenimientos oportunos, mejorando la vida útil de los vehículos con un gran beneficio para la empresa.

Además, al tener un mantenimiento adecuado de los vehículos se reduce de manera significativa las emisiones de contaminantes atmosféricos de origen vehicular.

CAPÍTULO II

2 MARCO TEÓRICO

2.1 SERVIDOR WEB: APACHE HTTP SERVER

Ilustración 3: *Apache HTTP Server*

Fuente: *(The Apache Software Foundation, 2012)*

Apache HTTP[2] Server es uno de los más robustos y rápidos servidores web multiplataforma que existen, ha sido creado desde la idea de open-source[3] demostrando una vez más que ir asociado a esta idea no es signo de fracaso y lo avala el ser el servidor web más utilizado en todo el mundo (desde pequeñas y grandes empresas, pasando por instituciones y universidades) (The Apache Software Foundation, 2012).

Con Apache HTTP Server se puede ejecutar CGI, Perl, PHP más Bases de datos, SSL, soporte para host virtuales, soporte IPv6, en fin, casi todo lo que le pidamos a cualquier servidor web.

---

2 HTTP es un protocolo sin estado, es decir, no recuerda nada relativo a conexiones anteriores a la actual. La conexión sólo tiene la duración correspondiente a la transmisión de la página solicitada si la encuentra, y si no la encuentra devuelve un código de error (HTTP, 2012).

3 Código Abierto es el software que está licenciado de tal manera que los usuarios pueden estudiar, modificar y mejorar su diseño mediante la disponibilidad de su código fuente (Código abierto, 2012).

Apache es un servidor web de software libre desarrollado por la Apache Software Foundation[4], cuyo objetivo es servir o suministrar páginas web (en general, hipertextos) a los clientes web o navegadores que las solicitan.

La arquitectura utilizada es cliente/servidor, es decir, el equipo cliente hace una solicitud o petición al equipo servidor y éste la atiende.

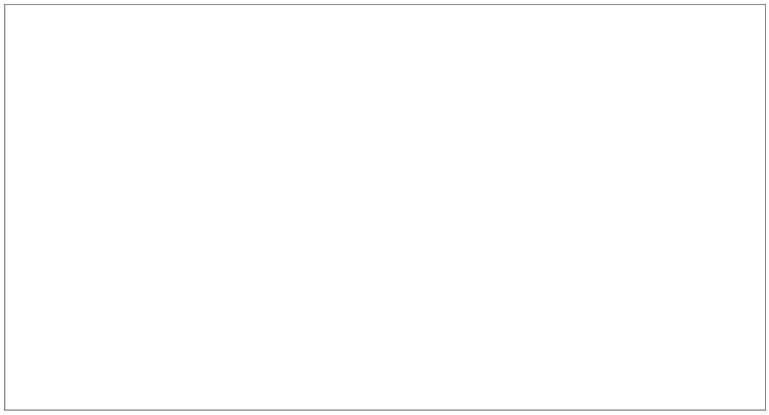


Ilustración 4: *Arquitectura Cliente/Servidor*

Fuente: *(Camara. A, 2011)*

En el equipo cliente se ejecuta una aplicación llamada “navegador o cliente web” que:

- Sirve de interfaz con el usuario: atiende sus peticiones, muestra los resultados de las consultas y proporciona al usuario un conjunto de herramientas que facilitan su comunicación con el servidor.

- Se comunica con el servidor web: transmite las peticiones de los usuarios.

El protocolo utilizado para la transferencia de hipertexto es HTTP (HiperText Transfer Protocol) que está basado en el envío de mensajes y establece el conjunto de normas mediante las cuales se envían las peticiones de acceso a una web y la respuesta de la misma.

[4] Apache Software Foundation es una comunidad de desarrolladores descentralizada. Los proyectos se desarrollan de forma colaborativa y consensuada. Está basada en la meritocracia, es decir, sus miembros deben ganarse méritos colaborando activamente en los proyectos (The Apache Software Foundation, 2012).

Ilustración 5: *Programación en el Cliente y en el Servidor*

Fuente: *(AyCarg, 2011)*

El servidor web Apache proporciona contenidos al cliente web o navegador como:

1. Páginas estáticas: es el uso más generalizado que se hace de un servidor web. De esta forma se transfieren archivos HTML, imágenes, etc y no se requiere un servidor muy potente en lo que al hardware y software se refiere.

2. Páginas dinámicas: la información que muestran las páginas que sirve Apache cambia ya que se obtiene a partir de consultas a bases de datos u otras fuentes de datos. Son, por tanto, páginas con contenido dinámico, cambiante.

Su curioso nombre hace referencia a sus orígenes. Cuando el proyecto inicial (Centro Nacional de Actividades de Supercomputación, NCSA, Universidad de Illinois)[5] fue abandonado por su principal desarrollador, Rob McCool[6], diferentes webmasters[7] comenzaron a desarrollar “parches” para el código fuente de este servidor inicial y mediante el correo electrónico sincronizaban sus aportaciones.
De esta forma apareció el proyecto Apache, cuyo nombre se debe a: A PAtCHy server (un servidor "parcheado"). La primera versión de Apache es la 0.6 en 1995, y en la actualidad la versión disponible es la 2.2.

Y su nombre de Apache se debe a que Behelendorf[8] quería que tuviese la connotación de algo que es firme y enérgico pero no agresivo, y la tribu Apache fue la última en rendirse al que pronto se convertiría en gobierno de EEUU, y en esos momentos la preocupación de su grupo era que llegasen las empresas y "civilizasen" el paisaje que habían creado los primeros ingenieros de internet.

Apache HTTP Server es un esfuerzo para desarrollar y mantener un servidor HTTP de código abierto para sistemas operativos como UNIX[9] y Windows NT[10] (Softpedia, 2012).

---

5 NCSA es la sigla del *National Center for Supercomputing Applications* (Centro Nacional de Aplicaciones de Supercomputación). Es un organismo estadounidense relacionado con la investigación en el campo de la Informática y las Telecomunicaciones (NCSA, 2012).

6 Rob McCool Autor del Servidor Apache HTTP. Él escribió la primera versión (Robert McCool, 2012).

7 Webmasters persona responsable de mantenimiento o programación de un sitio web, en ciertos casos es el responsable de los contenidos del sitio mientras que en otros es el encargado de la operatibilidad, programación y mantenimiento de la disponibilidad de un sitio web sin que necesariamente intervenga en la creación de contenidos (Webmaster, 2012).

8 Behelendorf Pionero del movimiento internacional de software libre de código abierto, creador del servidor HTTP Apache (Brian, 2012).

9 Unix Es una familia de sistemas operativos tanto para ordenadores personales como para mainframes. Soporta gran número de usuarios y posibilita la ejecución de distintas tareas de forma simultánea (multiusuario y multitarea) (Unix, 2012).

10 Windows NT (Windows New Technology) Nueva tecnología de Windows. Sistema operativo avanzado de 32 bits para redes, de Microsoft, que ejecuta aplicaciones DOS y Windows. incluyen redes par a par, multitarea preferente, multilectura, multiprocesamiento, tolerancia a fallas y soporte para el conjunto de caracteres Unicode (Windows NT, 2012).

El objetivo principal de Apache HTTP Server es el de proporcionar un servidor eficaz, seguro y extensible con servicios HTTP en sincronización con los estándares de HTTP. Apache ha sido el servidor web más popular en Internet desde 1996.

Apache HTTP Server es un esfuerzo de desarrollo de software colaborativo encaminado a crear una implementación de código abierto, sólida, de calidad comercial, completa y disponible libremente de un servidor HTTP (web).

Está administrado conjuntamente por un grupo de voluntarios de todo el mundo, utilizando el Internet y la web para comunicar, planificar y desarrollar el servidor y su documentación relacionada.

Tal como sucede en los proyectos de software libre: cientos de usuarios han contribuido con ideas, código y documentación, a través de la comunidad.

Apache se ejecuta en millones de servidores de Internet. Ha sido probado exhaustivamente por desarrolladores y usuarios.

Apache HTTP Server mantiene rigurosos estándares antes de lanzar nuevas versiones del servidor y el servidor se ejecuta sin problemas en más de 70% de todos los servidores WWW disponibles en Internet.

Cuando se muestran errores, se publican ajustes y nuevas versiones tan pronto como estén disponibles.

2.1.1 ALGUNAS DE LAS CARACTERÍSTICAS DE APACHE HTTP SERVER

- Es un servidor web potente, flexible y compatible.
- Es muy configurable y extensible con módulos de terceros.
- Puede personalizarse escribiendo "módulos" utilizando el módulo Apache API. Proporciona código fuente completo y viene con una licencia ilimitada. Es multiplataforma es decir, se puede ejecutar en cualquier sistema operativo. Cuenta con una gran comunidad.

2.1.2 MOD JK

mod_jk es un conector que permite al contenedor de Java Server Pages (JSP) Jakarta Tomcat interactuar con servidores web como Apache, Netscape, iPlanet, SunOne e incluso IIS usando el protocolo AJP (Apache, 2012).

La principal funcionalidad de este módulo es permitir a servidores de aplicaciones o al servidor Jakarta Tomcat enlazarse con un servidor web. Este servidor web, típicamente el servidor HTTP Apache, introduce una mayor gestión en las conexiones de los clientes y mayor la seguridad en las transacciones del sistema. Así mismo se puede enlazar varias instancias al servidor web permitiendo así una mayor tolerancia a errores y aligerar la carga en los servidores Java.

2.2 SYMFONY PHP WEB FRAMEWORK

Ilustración 6: *Symfony PHP Web Framework*

Fuente:*(SensioLabs, 2012)*

En el año 2003, Fabien Potencier[11], creador de symfony y actual CEO[12] de Sensio Labs, investigó acerca de las herramientas open source existentes para el desarrollo de aplicaciones web en PHP, pero ninguna de las existentes cumplió con sus expectativas. Cuando PHP 5 fue liberado, consideró que las herramientas que existían en ese momento habían madurado lo suficiente para ser integradas en un solo framework[13] (SensioLabs, 2012).

Le llevó un año desarrollar el núcleo de symfony. Basó su trabajo en el Modelo Vista Controlador, el ORM[14] de Propel[15] y el ayudante para realizar plantillas de Ruby on Rails[16].

---

11 Fabien Potencier Creador de symfony y actual CEO de Sensio Labs (Potencier, 2012).

12 CEO (Chief Executive Officer) o Director Ejecutivo, Presidente Ejecutivo o Principal Oficial Ejecutivo, es la persona que tiene a su cargo la máxima autoridad de la gestión y dirección administrativa en una empresa, organismo, asociación o institución (Director ejecutivo, 2012).

13 Framework (plataforma, entorno, marco de trabajo). Desde el punto de vista del desarrollo de software, un framework es una estructura de soporte definida, en la cual otro proyecto de software puede ser organizado y desarrollado (Framework, 2012).

14 ORM (Mapeo Objeto Relacional) Es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, utilizando un motor de persistencia (ORM, 2012).

15 Propel es un kit de mapeo objeto-relacional (ORM) de código abierto escrito en PHP. Es además una parte integral del framework Symfony (Propel, 2012).

16 Ruby on Rails Es un framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby, siguiendo el paradigma de la arquitectura Modelo Vista Controlador (MVC) (37signals)

La primera versión de symfony fue lanzada en octubre de 2005, por Fabien Potencier. Originalmente fue creado para el desarrollo de las aplicaciones de Sensio. Luego, tras el éxito que tuvo en el desarrollo de una página web para comercio electrónico y algunos otros proyectos, decidió liberarlo bajo una licencia open source.

Symfony es patrocinado por Sensio Labs, una compañía francesa que provee consultoría, servicios, formación sobre tecnologías open source. Inicialmente fue nombrado Sensio Framework, y a todas sus clases se les aplicó el prefijo sf. Tiempo después, cuando se decidió lanzarlo como un framework open source, se acordó llamarle Symfony.

Symfony es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web mediante algunas de sus principales características. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web.

Symfony está desarrollado completamente en PHP 5.3. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL[17], PostgreSQL, Oracle[18] y Microsoft SQL Server[19]. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows.

---

[17] MySQL Es un sistema de gestión de bases de datos (SGBD) multiusuario, multiplataforma y de código abierto (MySQL).

[18] Oracle es un sistema de gestión de base de datos objeto-relacional desarrollado por Oracle Corporation (Oracle, 2012).

[19] Microsoft SQL Server Es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional.

2.2.1 CARACTERÍSTICAS

Symfony fue diseñado para ajustarse a los siguientes requisitos:

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares).

- Independiente del sistema gestor de bases de datos. Su capa de abstracción y el uso de Propel, permiten cambiar con facilidad de SGBDR[20] en cualquier fase del proyecto.

- Utiliza programación orientada a objetos, de ahí que sea imprescindible PHP 5.

- Sencillo de usar en la mayoría de casos, aunque es preferible para el desarrollo de grandes aplicaciones Web que para pequeños proyectos.

- Aunque utiliza MVC (Modelo Vista Controlador), tiene su propia forma de trabajo en este punto, con variantes del MVC clásico como la capa de abstracción de base de datos, el controlador frontal y las acciones.

- Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador sólo debe configurar aquello que no es convencional.

- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.

- Preparado para aplicaciones empresariales y adaptable a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.

[20] SGBDR Los sistemas de gestión de bases de datos relacional son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan (SGBD).

- Código fácil de leer que incluye comentarios de phpDoc[21] y que permite un mantenimiento muy sencillo.

- Fácil de extender, lo que permite su integración con las bibliotecas de otros fabricantes.

- Una potente línea de comandos que facilitan generación de código, lo cual contribuye a ahorrar tiempo de trabajo.

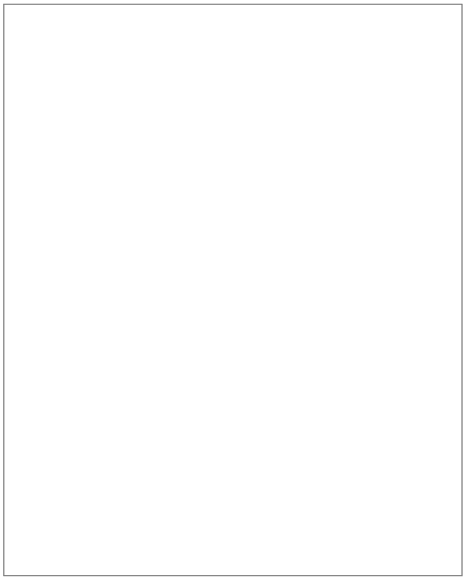


Ilustración 7: *Estructura de Symfony*

Fuente: *(SensioLabs, 2012)*

Symfony emplea el tradicional patrón de diseño MVC (modelo-vista-controlador) para separar las distintas partes que forman una aplicación web. El modelo representa la información con la que trabaja la aplicación y se encarga de acceder a los datos.
La vista transforma la información obtenida por el modelo en las páginas web a las que acceden los usuarios. El controlador es el encargado de coordinar todos los demás elementos y transformar las peticiones del usuario en operaciones sobre el modelo y la vista (Potencier, 2010).

[21] phpDoc Es una adaptación de javadoc para php que define un estándar oficial para comentar código php (phpDoc, 2012).

2.2.2 AUTOMATIZACIÓN DE CARACTERÍSTICAS DE PROYECTOS WEB

Symfony automatiza la mayoría de elementos comunes de los proyectos web, como:

- La capa de internacionalización[22] que incluye Symfony permite la traducción de los datos y de la interfaz, así como la adaptación local de los contenidos.

- La capa de presentación utiliza plantillas y *layouts*[23] que pueden ser creados por diseñadores HTML sin ningún tipo de conocimiento del framework. Los *helpers* incluidos permiten minimizar el código utilizado en la presentación, ya que encapsulan grandes bloques de código en llamadas simples a funciones.

- Los formularios incluyen validación automatizada y relleno automático de datos (*"repopulation"*), lo que asegura la obtención de datos correctos y mejora la experiencia de usuario.

- Los datos incluyen mecanismos de escape que permiten una mejor protección contra los ataques producidos por datos corruptos.

- La gestión de la caché[24] reduce el ancho de banda utilizado y la carga del servidor.

- La autenticación[25] y la gestión de credenciales simplifican la creación de secciones restringidas y la gestión de la seguridad de usuario.

---

[22] Internacionalizacion Es el proceso de diseñar software de manera tal que pueda adaptarse a diferentes idiomas y regiones sin la necesidad de realizar cambios de ingeniería ni en el código (Intern, 2012).

[23] Layouts Etiqueta que se utiliza para identificar independientemente a cada uno de los componentes (SensioLabs, 2012).

[24] Caché Componente que almacena datos para que los futuros requerimientos a esos datos puedan ser servidos más rápidamente. Generalmente son datos temporales (SensioLabs, 2012).

[25] Autenticación En la seguridad de ordenador, la autenticación es el proceso de intento de verificar la identidad digital del remitente de una comunicación como una petición para conectarse. El remitente siendo autenticado puede ser una persona que usa un ordenador, un ordenador por sí mismo o un programa del ordenador (SensioLabs, 2012).

- El sistema de enrutamiento[26] y las URL[27] *limpias* permiten considerar a las direcciones de las páginas como parte de la interfaz, además de estar optimizadas para los buscadores.

- El soporte de correo electrónico incluido y la gestión de APIs[28] permiten a las aplicaciones web interactuar más allá de los navegadores.

- Los listados son más fáciles de utilizar debido a la paginación automatizada, el filtrado y la ordenación de datos.
- Los plugins[29], las factorías (patrón de diseño *"Factory"*) y los *"mixin"* permiten realizar extensiones a medida de Symfony.

- Las interacciones con Ajax[30] son muy fáciles de implementar mediante los *helpers* que permiten encapsular los efectos JavaScript[31] compatibles con todos los navegadores en una única línea de código.

---

26 Enrutamiento Es la forma de buscar un camino entre todos los posibles en una red de paquetes cuyas topologías poseen una gran conectividad (SensioLabs, 2012).

27 URL Es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet para su localización o identificación (URL, 2012).

28 API's (Application Programming Interface - Interfaz de Programación de Aplicaciones) es un conjunto de convenciones internacionales que definen cómo debe invocarse una determinada función de un programa desde una aplicación (API, 2012).

29 Plugins Programa que puede anexarse a otro para aumentar sus funcionalidades (generalmente sin afectar otras funciones ni afectar la aplicación principal). No se trata de un parche ni de una actualización, es un módulo aparte que se incluye opcionalmente en una aplicación (SensioLabs, 2012).

30 Ajax Acrónimo de *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (*Rich Internet Applications*) (AJAX, 2012).

31 JavaScript Es un lenguaje de programación interpretado, Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico (JavaScript, 2012).

2.2.3 ENTORNO DE DESARROLLO Y HERRAMIENTAS

Symfony puede ser completamente personalizado para cumplir con los requisitos de las empresas que disponen de sus propias políticas y reglas para la gestión de proyectos y la programación de aplicaciones. Por defecto incorpora varios entornos de desarrollo diferentes e incluye varias herramientas que permiten automatizar las tareas más comunes de la ingeniería del software:

- Las herramientas que generan automáticamente código han sido diseñadas para hacer prototipos de aplicaciones y para crear fácilmente la parte de gestión de las aplicaciones.

- El framework de desarrollo de pruebas unitarias y funcionales proporciona las herramientas ideales para el desarrollo basado en pruebas (*"test-driven development"*).

- La barra de depuración web simplifica la depuración de las aplicaciones, ya que muestra toda la información que los programadores necesitan sobre la página en la que están trabajando.

- La interfaz de línea de comandos automatiza la instalación de las aplicaciones entre servidores.

- Es posible realizar cambios *"en caliente"* de la configuración (sin necesidad de reiniciar el servidor).

- El completo sistema de log permite a los administradores acceder hasta el último detalle de las actividades que realiza la aplicación.

2.2.4 ¿POR QUÉ LO LLAMARON *"SYMFONY"* ?

Porque Fabien quería una nombre corto que tuviera una letra 's' (de Sensio) y una letra 'f' (de framework), que fuera fácil de recordar y que no estuviera asociado a otra herramienta de desarrollo. Además, no le gustan las mayúsculas. "Symfony" era muy parecido a lo que estaba buscando, aunque no es una palabra correcta en el idioma inglés (la palabra correcta es *"symphony"*), y además estaba libre como nombre de proyecto. La otra alternativa era *"baguette"*.

2.2.5 LA COMUNIDAD SYMFONY

En cuanto se abrió al público el sitio web de Symfony (http://www.symfony-project.org/) muchos desarrolladores de todo el mundo se descargaron e instalaron el framework, comenzaron a leer la documentación y construyeron sus primeras aplicaciones con Symfony, aumentando poco a poco la popularidad de Symfony.

En ese momento, los frameworks para el desarrollo de aplicaciones web estaban en pleno apogeo, y era muy necesario disponer de un completo framework realizado con PHP. Symfony proporcionaba una solución irresistible a esa carencia, debido a la calidad de su código fuente y a la gran cantidad de documentación disponible, dos ventajas muy importantes sobre otros frameworks disponibles. Los colaboradores aparecieron en seguida proponiendo parches y mejoras, detectando los errores de la documentación y realizando otras tareas muy importantes.

El repositorio público de código fuente y el sistema de notificación de errores y mejoras mediante tickets permite varias formas de contribuir al proyecto y todos los voluntarios son bienvenidos. Fabien continua siendo el mayor contribuidor de código al repositorio y se encarga de garantizar la calidad del código.

Actualmente, el foro de Symfony, las listas de correo y el IRC[32] ofrecen otras alternativas válidas para el soporte del framework, con el que cada pregunta suele obtener una media de cuatro respuestas. Cada día nuevos usuarios instalan Symfony y el wiki y la sección de fragmentos de código almacenan una gran cantidad de documentación generada por los usuarios. Cada semana el número de aplicaciones conocidas desarrolladas con Symfony se incrementa en cinco y el aumento continua.

2.3 SISTEMA DE GESTIÓN DE BASE DE DATOS RELACIONAL: POSTGRESQL

Ilustración 8: *PostgreSQL*

Fuente: *(PGDG, 2012)*

PostgreSQL es un poderoso sistema de base de datos relacional de código abierto con el desarrollo activo y una arquitectura probada se ha ganado una sólida reputación de fiabilidad, integridad de datos y corrección. Funciona en todos los principales sistemas operativos, entre ellos: Linux, Unix (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows. Es totalmente compatible con ACID[33], tiene soporte completo para claves foráneas, uniones, vistas, disparadores y procedimientos almacenados (en varios lenguajes).

---

[32] IRC (Internet Relay Chat). Protocolo de comunicación en tiempo real basado en texto (chatear). Permite conversar con personas en forma de texto dentro de canales (salones de chat) o de forma privada (IRC, 2012).

[33] ACID En bases de datos se denomina ACID a un conjunto de características necesarias para que una serie de instrucciones puedan ser consideradas como una transacción (ACID, 2012).

Incluye la mayoría de tipos de datos de SQL92 y SQL99 entre ellos: INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL y TIMESTAMP. También soporta almacenamiento de objetos binarios grandes como imágenes, sonidos y video. Dispone de interfaces nativas de programación de C/C++, Java, .NET, Perl, Python, Ruby, Tcl, ODBC, entre otros; y una documentación de carácter excepcional (PGDG, 2012).

Como muchos otros proyectos de software libre, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyados por organizaciones comerciales. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

El uso de caracteres en mayúscula en el nombre PostgreSQL puede confundir a algunas personas a primera vista. Las distintas pronunciaciones de "SQL" pueden llevar a confusión. Los desarrolladores de PostgreSQL lo pronuncian /poːst gɹɛs kjuː ɛl/;. Es también común oír abreviadamente como simplemente "Postgres", el que fue su nombre original.

Debido a su soporte del estándar SQL entre la mayor parte de bases de datos relacionales, la comunidad consideró cambiar el nombre al anterior Postgres. Sin embargo, el PostgreSQL Core Team[34] anunció en 2007 que el producto seguiría llamándose PostgreSQL. El nombre hace referencia a los orígenes del proyecto como la base de datos "post-Ingres", y los autores originales también desarrollaron la base de datos Ingres.

PostgreSQL ha tenido una larga evolución, la cual se inicia en 1982 con el proyecto Ingres en la Universidad de Berkeley. Este proyecto, liderado por Michael Stonebraker[35], fue uno de los primeros intentos en implementar un motor de base de datos relacional.

---

[34] PostgreSQL Core Team Equipo encargado de la estandarización de PostgreSQL (PGDG, 2012).

[35] Michael Stonebraker Es un científico especializado en la base de datos de investigación y desarrollo. Su carrera abarca, y ayudó a crear, la mayoría de la base de datos relacional del mercado existente hoy en día (Michael, 2012).

Después de haber trabajado un largo tiempo en Ingres y de haber tenido una experiencia comercial con él mismo, Michael decidió volver a la Universidad en 1985 para trabajar en un nuevo proyecto sobre la experiencia de Ingres, dicho proyecto fue llamado post-ingres o simplemente POSTGRES.

El proyecto post-ingres pretendía resolver los problemas con el modelo de base de datos relacional que habían sido aclarados a comienzos de los años 1980. El principal de estos problemas era la incapacidad del modelo relacional de comprender "tipos", es decir, combinaciones de datos simples que conforman una única unidad. Actualmente estos son llamados objetos. Se esforzaron en introducir la menor cantidad posible de funcionalidades para completar el soporte de tipos.

Estas funcionalidades incluían la habilidad de definir tipos, pero también la habilidad de describir relaciones las cuales hasta ese momento eran ampliamente utilizadas pero mantenidas completamente por el usuario. En Postgres la base de datos «comprendía» las relaciones y podía obtener información de tablas relacionadas utilizando reglas. Postgres usó muchas ideas de Ingres pero no su código.

La siguiente lista muestra los hitos más importantes en la vida del proyecto Postgres.

- 1986: se publicaron varios papers (papeles) que describían las bases del sistema.
- 1988: ya se contaba con una versión utilizable.
- 1989: el grupo publicaba la versión uno para una pequeña comunidad de usuarios.
- 1990: se publicaba la versión dos la cual tenía prácticamente reescrito el sistema de reglas.
- 1991: publicación de la versión tres, esta añadía la capacidad de múltiples motores de almacenamiento.

- 1993: crecimiento importante de la comunidad de usuarios, la cual demandaba más características.

- 1994: después de la publicación de la versión cuatro, el proyecto terminó y el grupo se disolvió.

El proyecto PostgreSQL continúa haciendo lanzamientos principales anualmente y lanzamientos menores de reparación de bugs, todos disponibles bajo la licencia BSD[36], y basados en contribuciones de proveedores comerciales, empresas aportantes y programadores de código abierto mayormente.

2.3.1 CARACTERÍSTICAS

Algunas de sus principales características son, entre otras:

ALTA CONCURRENCIA

Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo commit[37]. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos (PGDG, 2011).

---

[36] BSD Es la licencia de software otorgada principalmente para los sistemas BSD (*Berkeley Software Distribution*). Es una licencia de software libre permisiva como la licencia de OpenSSL o la MIT License (BSD, 2012).

[37] Commit (acción de comprometer) se refiere a la idea de consignar un conjunto de cambios "tentativos, o no permanentes". Un uso popular es al final de una transacción de base de datos (Commit, 2012).

AMPLIA VARIEDAD DE TIPOS NATIVOS

PostgreSQL provee nativamente soporte para:

- Números de precisión arbitraria.
- Texto de largo ilimitado.
- Figuras geométricas (con una variedad de funciones asociadas).
- Direcciones IP (IPv4 e IPv6).
- Bloques de direcciones estilo CIDR[38].
- Direcciones MAC[39].
- Arrays.

Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables gracias a la infraestructura GiST de PostgreSQL. Algunos ejemplos son los tipos de datos GIS creados por el proyecto PostGIS.

Otras características

- Claves ajenas también denominadas Llaves ajenas o Claves Foráneas (foreign keys).

- Disparadores (triggers): Un disparador o trigger se define como una acción específica que se realiza de acuerdo a un evento, cuando éste ocurra dentro de la base de datos. En PostgreSQL esto significa la ejecución de un procedimiento almacenado basado en una determinada acción sobre una tabla específica.

---

38 CIDR *Classless Inter-Domain Routing* o CIDR (en español «enrutamiento entre dominios sin clases») Representa la última mejora en el modo de interpretar las direcciones IP (CIDR, 2012).

39 MAC (siglas en inglés de *media access control*; en español "control de acceso al medio") corresponde de forma única a una tarjeta o dispositivo de red. Se conoce también como dirección física, y es única para cada dispositivo (MAC, 2012).

- Los disparadores se definen por seis características:
    - El nombre del disparador o trigger
    - El momento en que el disparador debe arrancar
    - El evento del disparador deberá activarse sobre
    - La tabla donde el disparador sé activará
    - La frecuencia de la ejecución
    - La función que podría ser llamada

Entonces combinando estas seis características, PostgreSQL le permitirá crear una amplia funcionalidad a través de su sistema de activación de disparadores (triggers).

- Vistas.
- Integridad transaccional.
- Herencia de tablas.
- Tipos de datos y operaciones geométricas.
- Soporte para transacciones distribuidas. Permite a PostgreSQL integrase en un sistema distribuido formado por varios recursos (p.ej, una base de datos PostgreSQL, otra Oracle, gestionado por un servidor de aplicaciones donde el éxito ("commit") de la transacción globlal es el resultado del éxito de las transacciones locales.

Ilustración 9: *Características de PostgreSQL*

Fuente: *(PGDG, 2012)*

2.4 LENGUAJE DE PROGRAMACIÓN: PHP

Ilustración 10: *Lenguaje de Programación PHP*

Fuente: *(phpDoc, 2012)*

PHP es un lenguaje de programación interpretado (Lenguaje de alto rendimiento), diseñado originalmente para la creación de páginas web dinámicas. Se usa principalmente para la interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica (PHP Group, 2012).

PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools). Fue creado originalmente por Rasmus Lerdorf[40] en 1994; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. El lenguaje PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores, el número de sitios en PHP ha compartido algo de su preponderante dominio con otros nuevos lenguajes no tan poderosos desde agosto de 2005. El sitio web de Wikipedia está desarrollado en PHP.

El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.

Aunque todo en su diseño está orientado a facilitar la creación de sitios webs, es posible crear aplicaciones con una interfaz gráfica para el usuario, utilizando la extensión PHP-Qt o PHP-GTK. También puede ser usado desde la línea de órdenes, de la misma manera como Perl o Python pueden hacerlo; a esta versión de PHP se la llama PHP-CLI (Command Line Interface).

Cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Éste procesa el script solicitado que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente. Mediante extensiones es también posible la generación de archivos PDF, Flash, así como imágenes en diferentes formatos.

[40] Rasmus Lerdorf Es un programador informático nacido en Groenlandia creador de la primera versión del lenguaje de programación PHP (Rasmus, 2012).

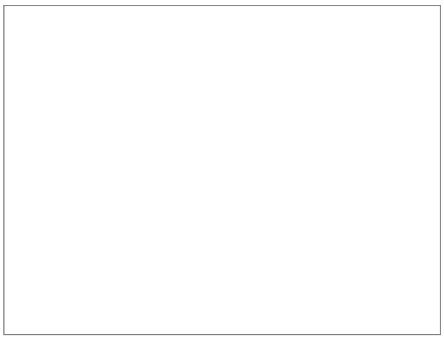


Ilustración 11: *Petición Cliente/Servidor*

Fuente: *(Tutty Morán, 2011)*

Permite la conexión a diferentes tipos de servidores de bases de datos.

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, y puede interactuar con los servidores de web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI.

PHP tiene las ventajas que provee el nuevo Zend Engine 2 como:

- Mejor soporte para la programación orientada a objetos.
- Mejoras de rendimiento.
- Mejor soporte para MySQL con extensión completamente reescrita.
- Mejor soporte a XML (XPath, DOM, etc.).
- Soporte nativo para SQLite.

- Soporte integrado para SOAP.
- Iteradores de datos.
- Manejo de excepciones.
- Mejoras con la implementación con Oracle.

2.4.1 CARACTERÍSTICAS DE PHP

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- El código fuente escrito en PHP es invisible al navegador web y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es Software Libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.

- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.

- PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar (muchos otros lenguajes tampoco lo hacen), aun haciéndolo, el programador puede aplicar en su trabajo cualquier técnica de programación o de desarrollo que le permita escribir código ordenado, estructurado y manejable.

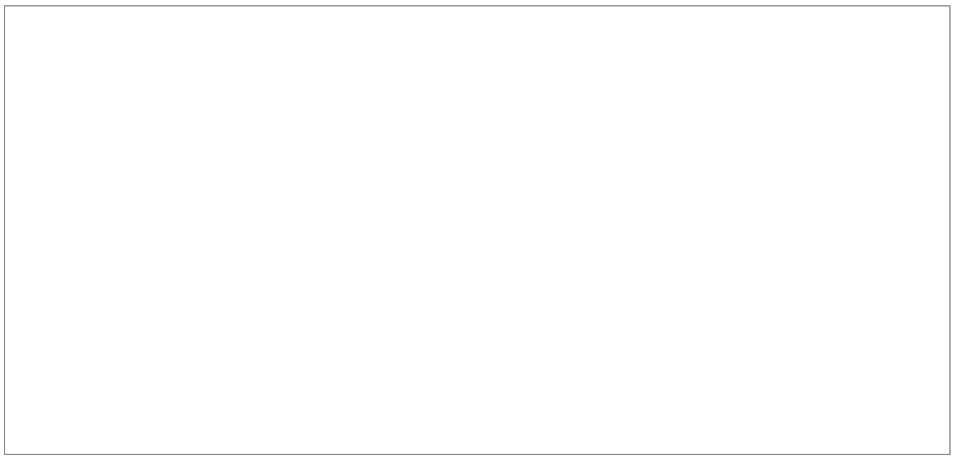


Ilustración 12: *Características de PHP*

Fuente: *(PrograWeb.net, 2012)*

Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador (MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.
Como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparan rápidamente.

El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP. Es utilizado en aplicaciones Web relacionadas por algunas de las organizaciones más prominentes tales como Mitsubishi[41], Redhat[42], Ericsson[43] y NASA[44].

La forma de usar php es insertando código php dentro del código html de un sitio web. Cuando un cliente (cualquier persona en la web) visita la página web que contiene éste código, el servidor lo ejecuta y el cliente sólo recibe el resultado. Su ejecución, es por tanto en el servidor, a diferencia de otros lenguajes de programación que se ejecutan en el navegador.

Generalmente los scripts en PHP se embeben en otros códigos como HTML, ampliando las posibilidades del diseñador de páginas web enormemente.

La interpretación y ejecución de los scripts PHP se hacen en el servidor, el cliente (un navegador que pide una página web) sólo recibe el resultado de la ejecución y jamás ve el código PHP.

2.5 PATRÓN DE ARQUITECTURA DE SOFTWARE: MODELO VISTA CONTROLADOR

La arquitectura MVC (Model/View/Controller) fue introducida como parte de la versión Smalltalk-80 del lenguaje de programación Smalltalk (Molina, 2012).

---

41 Mitsubishi Es una de las mayores compañías de Japón. Fue fundada el 13 de mayo de 1870 por Yatarō Iwasaki.

42 Redhat Es la compañía responsable de la creación y mantenimiento de una distribución del sistema operativo GNU/Linux.

43 Ericcson Es una compañía multinacional de origen sueco dedicada a ofrecer equipos y soluciones de telecomunicaciones, principalmente en los campos de la telefonía, la telefonía móvil las comunicaciones multimedia e internet.

44 NASA es la Administración Nacional de Aeronáutica y del Espacio (National Aeronautics and Space Administration) de los Estados Unidos, que es la agencia gubernamental responsable de los programas espaciales.

Fue diseñada para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. Sus características principales son que el Modelo, las Vistas y los Controladores se tratan como entidades separadas; esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas.

Este modelo de arquitectura se puede emplear en sistemas de representación gráfica de datos o en sistemas CAD, en donde se presentan partes del diseño con diferente escala de aumento, en ventanas separadas.

Este modelo de arquitectura presenta varias ventajas:

- Hay una clara separación entre los componentes de un programa; lo cual permite implementarlos por separado.

- Hay un API muy bien definido; cualquiera que use el API, podrá reemplazar el Modelo, la Vista o el Controlador, sin aparente dificultad.

- La conexión entre el Modelo y sus Vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.

Al incorporar el modelo de arquitectura MVC a un diseño, las piezas de un programa se pueden construir por separado y luego unirlas en tiempo de ejecución. Si uno de los componentes, posteriormente, se observa que funciona mal, puede reemplazarse sin que las otras piezas se vean afectadas.

2.5.1 DEFINICIÓN DE LAS PARTES

El Modelo es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.

La Vista es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Modelo a través de una referencia al propio Modelo.

El Controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo.

Para el diseño de aplicaciones con sofisticados interfaces se utiliza el patrón de diseño Modelo-Vista-Controlador. La lógica de un interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio. Si realizamos un diseño ofuscado, es decir, un pastiche que mezcle los componentes de interfaz y de negocio, entonces la consecuencia será que, cuando necesitemos cambiar el interfaz, tendremos que modificar trabajosamente los componentes de negocio. Mayor trabajo y más riesgo de error.

Se trata de realizar un diseño que desacople la vista del modelo, con la finalidad de mejorar la reusabilidad. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos.

2.5.2 ELEMENTOS DEL PATRÓN

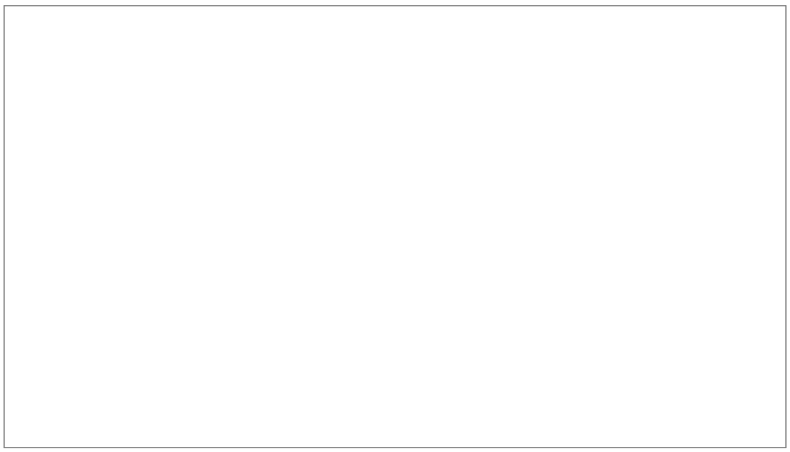


Ilustración 13: *Elementos del Patrón de Arquitectura de Software: MVC*

Fuente: *(WL, 2012)*

- Modelo: datos y reglas de negocio.

- Vista: muestra la información del modelo al usuario.

- Controlador: gestiona las entradas del usuario.

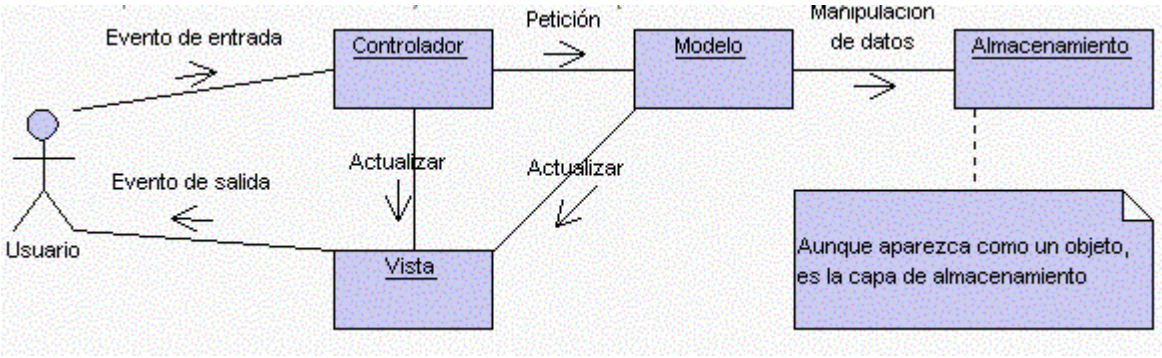


Ilustración 14: *Elementos del Patrón*

Fuente: *Propia*

Un modelo puede tener diversas vistas, cada una con su correspondiente controlador. Un ejemplo clásico es el de la información de una base de datos, que se puede presentar de diversas formas: diagrama de tarta, de barras, tabular, etc. A continuación se detalla cada componente:

1. El modelo es el responsable de:

    - Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.

    - Define las reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: "Si la mercancía pedida no está en el almacén, consultar el tiempo de entrega estándar del proveedor".

    - Lleva un registro de las vistas y controladores del sistema.

- Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo (por ejemplo, un fichero bath que actualiza los datos, un temporizador que desencadena una inserción, etc).

2. El controlador es responsable de:

    - Recibe los eventos de entrada (un clic, un cambio en un campo de texto, etc.).

    - Contiene reglas de gestión de eventos, del tipo "SI Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada al método "Actualizar()". Una petición al modelo puede ser "Obtener_tiempo_de_entrega(nueva_orden_de_venta)".

3. Las vistas son responsables de:

    - Recibir datos del modelo e interactuar con el usuario.

    - Tienen un registro de su controlador asociado (normalmente porque además lo instancia).

    - Pueden dar el servicio de "Actualización()", para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

Un ejemplo de MVC con un modelo pasivo (aquel que no notifica cambios en los datos) es la navegación web, que responde a las entradas del usuario, pero no detecta los cambios en datos del servidor.

El diagrama de secuencia:

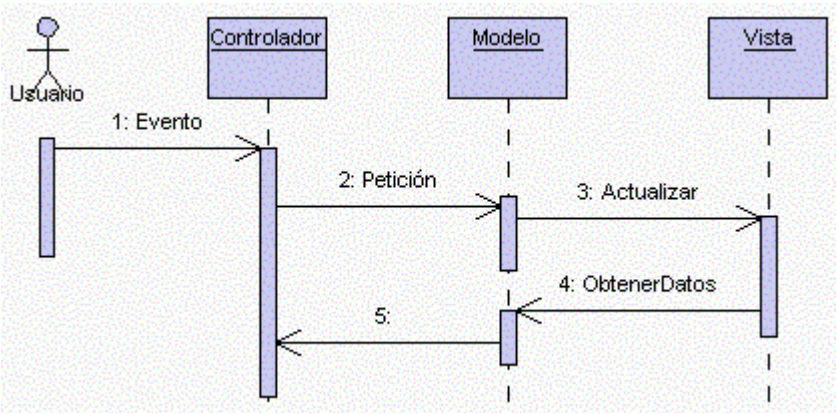


Ilustración 15: *Diagrama de Secuencia de MVC*

Fuente: *Propia*

Pasos:

1. El usuario introduce el evento.

2. El Controlador recibe el evento y lo traduce en una petición al Modelo (aunque también puede llamar directamente a la vista).

3. El modelo (si es necesario) llama a la vista para su actualización.

4. Para cumplir con la actualización la Vista puede solicitar datos al Modelo.

5. El Controlador recibe el control.

El Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos. El patrón de llamada y retorno MVC (según CMU), se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.

2.5.2.1 ¿QUÉ VENTAJAS TRAE UTILIZAR EL MVC?

1. Es posible tener diferentes vistas para un mismo modelo (ejemplo: representación de un conjunto de datos como una tabla o como un diagrama de barras).

2. Es posible construir nuevas vistas sin necesidad de modificar el modelo subyacente.

3. Proporciona un mecanismo de configuración a componentes complejos muchos más tratable que el puramente basado en eventos (el modelo puede verse como una representación estructurada del estado de la interacción).

2.5.3 FLUJO QUE SIGUE EL CONTROL EN UNA IMPLEMENTACIÓN GENERAL DE UN MVC

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente:

1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace).

2. El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.

3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.

4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra.

5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

Ilustración 16: *Diagrama de Flujo de un Framework MVC*

Fuente: *(Molina, 2012)*

CAPÍTULO III

3 DESCRIPCIÓN Y FUNCIONAMIENTO DEL SISTEMA

3.1 GESTIÓN DEL PROYECTO

En esta sección se detalla la planificación inicial del proyecto para la fase de inicio y la fase de elaboración (según la definición de la metodología RUP).

3.1.1 VISTA GENERAL DEL PROYECTO

La información que a continuación se incluye ha sido extraída de las diferentes reuniones que se han celebrado con el stakeholder[45] de la empresa desde el inicio del proyecto.

El proyecto debe proporcionar una respuesta para el desarrollo de todos los módulos implicados en el "Análisis, Diseño e Implementación de un Sistema de Control Vehicular en Instituciones Públicas". Estos módulos son los siguientes:

- **Vehículos, Accesorios y Herramientas:** Inventario de vehículos, accesorios y herramientas. Entrega y recepción. Registro de entrada y salida. Ingreso de fotografías de los vehículos.

- **Movilizaciones:** Emisión de las órdenes de movilización, informe diario de movilización de cada vehículo.

- **Novedades y Accidentes:** Parte de novedades y accidentes. Libro de novedades.

- **Lubricantes, combustibles y repuestos:** Control y orden de provisión de los lubricantes, combustibles y repuestos.

[45] Stakeholder es aquella persona o entidad que está interesada en la realización de un proyecto o tarea, auspiciando el mismo ya sea mediante su poder de decisión o de financiamiento, o a través de su propio esfuerzo.

- **Conductores:** Registro de conductores de vehículos. Ingreso de fotografías de los conductores.

- **Seguridad:** Manejo de usuarios, roles y permisos de acceso.

- **Registro con Código de Barras**

- **Reportes:** Orden de movilización, Listado de Conductores, Listado de Vehículos, Listado de Accesorios, Listado de Herramientas, Informes de movilización, Informes de Mantenimientos de Vehículos, Informes de Próximos Chequeos.

3.1.1.1 SUPOSICIONES Y RESTRICCIONES

Las suposiciones y restricciones respecto del sistema, y que se derivan directamente de las entrevistas con el stakeholder de la empresa son:

a) Debe contemplarse las implicaciones de los siguientes puntos críticos:
   a.a) Sistemas seguros: protección de información, seguridad en las transmisiones de datos, etc.
   a.b) Gestión de flujos de trabajo, seguridad de las transacciones e intercambio de información.

b) La automatización de la gestión interna del registro debe ajustarse a la legislación vigente.

Como es natural, la lista de suposiciones y restricciones se incrementará durante el desarrollo del proyecto, particularmente una vez establecido el artefacto “Visión”.

3.1.1.2 ENTREGABLES DEL PROYECTO

A continuación se indican y describen cada uno de los artefactos que serán generados y utilizados por el proyecto y que constituyen los entregables. Esta lista constituye en la configuración de RUP desde la perspectiva de artefactos, y que se propone para este proyecto.

Es preciso destacar que de acuerdo a la filosofía de RUP (y de todo proceso iterativo e incremental), todos los artefactos son objetos de modificaciones a lo largo del proceso de desarrollo, con lo cual, sólo al término del proceso podríamos tener una versión definitiva y completa de cada uno de ellos.

Sin embargo, el resultado de cada iteración y los hitos del proyecto están enfocados a conseguir un cierto grado de complejidad y estabilidad de los artefactos. Esto será indicado más adelante cuando se presenten los objetivos de cada iteración.

Modelo de Casos de Uso del Negocio

Es un modelo de las funciones de negocio vistas desde la perspectiva de los actores externos (Agentes de registro, solicitantes finales, otros sistemas, etc.) permite situar al sistema en el contexto organizacional haciendo énfasis en los objetivos en este ámbito. Este modelo se representa con un Diagrama de Casos de Uso usando estereotipos específicos para este modelo.

Modelo de Objetos del Negocio

Es un modelo que describe la realización de cada caso del negocio, estableciendo los actores internos, la información que en términos generales manipulan y los flujos de trabajo asociados al caso de uso del negocio.

Para la representación de este modelo se utilizan Diagramas de Colaboración (para mostrar actores externos, internos y las entidades que manipulan), un Diagrama de Clases para mostrar gráficamente las entidades del sistema y sus relaciones, y un Diagrama de Actividades para mostrar los flujos de trabajo.

Glosario

Es un documento que define los principales términos usados en el proyecto.

Modelo de Casos de Uso

El modelo de Casos de Uso presenta las funciones del sistema y los actores que hacen uso de ellas, con Diagramas de Casos de Uso.

Visión

Este documento define la visión del producto desde la perspectiva del cliente, especificando las necesidades y características del producto. Constituye una base de acuerdo a los requisitos del sistema.

Especificaciones de Casos de Uso

Para los casos de uso que se requieran (cuya funcionalidad no sea evidente o que no baste con una simple descripción narrativa) se realiza una descripción detallada utilizando una plantilla de documento, donde se incluyen: pre – condiciones, post – condiciones, flujos de eventos, requisitos no – funcionales asociados. También para casos de uso cuyo flujo de eventos sea complejo podrá adjuntarse una representación gráfica mediante un Diagrama de Actividad.

Especificaciones Adicionales

Este documento capturará todos los requisitos que no han sido incluidos como parte de los casos de uso y se refieren a requisitos no – funcionales globales. Dichos requisitos incluyen: requisitos legales o normas, aplicación de estándares, requisitos de calidad del producto, tales como: confiabilidad, desempeño, etc., u otros requisitos de ambiente, tales como: sistema operativo, requisitos de compatibilidad, etc.

Prototipos de Interfaces de Usuario

Se trata de prototipos que permiten al usuario hacerse una idea más o menos precisa de las interfaces que proveerá el sistema, y así conseguir una retroalimentación de su parte respecto a los requisitos del sistema. Estos prototipos se realizarán como: dibujos a mano en papel, dibujos con alguna herramienta gráfica o prototipos ejecutables interactivos, siguiendo ese orden de acuerdo al avance del proyecto.

Sólo los de este último tipo serán entregados al final de esta fase de Elaboración, los otros serán desechados. Así mismo, este artefacto, será desechado en la fase final de Construcción en la medida que el resultado de las iteraciones vayan desarrollando el producto final.

Modelo de Análisis y Diseño

Este modelo establece la realización de casos de uso en clases y pasando desde una representación en términos de análisis (sin incluir aspectos de implementación) hacia una de diseño (incluyendo una orientación hacia el entorno de implementación), de acuerdo al avance del proyecto.

Modelo de Datos

Previendo que la persistencia de la información del sistema será soportada por una base de datos relacional, este modelo describe la presentación lógica de los datos persistentes, de acuerdo con el enfoque para modelado relacional de daos. Para expresar este modelo se utiliza un Diagrama de Clases.

Modelo de Despliegue

Este modelo muestra el despliegue la configuración de tipos de nodos del sistema, en los cuales se hará el despliegue de los componentes.

Modelo de Implementación

Este modelo es una colección de componentes y los módulos que los contienen. Estos componentes incluyen: ficheros ejecutables, ficheros de código fuente, y todo otro tipo de ficheros necesarios para la implantación y despliegue del sistema.

Casos de Prueba

Cada prueba es especificada mediante un documento que establece las condiciones de ejecución, las entradas de la prueba, y los resultados esperados. Estos casos de prueba son aplicados como pruebas de regresión en cada iteración. Cada caso de prueba llevará asociado un procedimiento de prueba con las instrucciones para realizar la prueba.

Solicitud de Cambio

Los cambios propuestos para los artefactos se formalizarán mediante este documento. Mediante este documento se hace un seguimiento de los defectos, solicitud de mejoras o cambios en los requisitos del producto. Así se provee un registro de decisiones de cambios, de su evaluación e impacto, y se asegura que éstos sean conocidos por el equipo de desarrollo.

Los cambios se establecen respecto de la última baseline (el estado del conjunto de los artefactos en un momento determinado del proyecto) establecida. En nuestro caso al final de cada iteración se establecerá una baseline.

Plan de Iteración

Es un conjunto de actividades y tareas ordenadas temporalmente, con recursos asignados, dependencias entre ellas.

Evaluación de Iteración

Este documento incluye la evaluación de los resultados de cada iteración, el grado en el cual se han conseguido los objetivos de la iteración, las lecciones aprendidas y los cambios a ser realizados.

Lista de Riesgos

Este documento incluye una lista de los riesgos conocidos y vigentes en el proyecto, ordenados en orden decreciente de importancia y con acciones específicas de contingencia o para su mitigación.

Manual de Instalación

Este documento incluye las instrucciones para realizar la instalación del producto.

Material de Apoyo al Usuario Final

Corresponde a un conjunto de documentos y facilidades de uso del sistema, incluyendo: Guías del Usuario, Guías de Operación, Guías de Mantenimiento y Sistema de Ayuda en línea.

Producto

Los ficheros del producto empaquetados y almacenados en un medio digital con los mecanismos apropiados para facilitar su instalación. El producto, a partir de la primera iteración de la fase de Construcción es desarrollado incremental e iterativamente, obteniéndose una nueva release al final de cada iteración.

3.1.2 ORGANIZACIÓN DEL PROYECTO

3.1.2.1 PARTICIPANTES EN EL PROYECTO

El personal participante en el proyecto está formado por los siguientes puestos de trabajo y personal asociado:

Jefe de Proyecto: Ing. José Luis Rodríguez
Arquitecto de Software: Srta. Patricia Terán Sevilla
Ingeniero de Software: Srta. Patricia Terán Sevilla
Programador: Srta. Patricia Terán Sevilla
Gerente: Lcdo. Alejandro Jojoa
Secretaria: María Augusta Fajardo
Contador: Gabriela Remache

3.1.2.2 INTERFACES EXTERNAS

La empresa definirá los participantes del proyecto que proporcionarán los requisitos del sistema, y entre ellos quiénes serán los encargados de evaluar los artefactos de acuerdo a cada módulo y según el plan establecido.

El equipo de desarrollo interactuará activamente con los participantes de la empresa para especificación y validación de los artefactos generados.

3.1.2.3 ROLES Y RESPONSABILIDADES

A continuación se describen las principales responsabilidades de cada uno de los puestos en el equipo de desarrollo, de acuerdo con los roles que desempeñan en RUP.

Puesto	Responsabilidad
Jefe de Proyecto	El jefe de proyecto asigna los recursos, gestiona las prioridades, coordina las interacciones con los clientes y usuarios, y mantiene al equipo del proyecto enfocado en los objetivos. El jefe de proyecto también establece un conjunto de prácticas que aseguran la integridad y calidad de los artefactos del proyecto. Gestión de riesgos, Planificación y control del proyecto.
Arquitecto de Software	El arquitecto de software se encarga de supervisar el establecimiento de la Arquitectura del Sistema, es decir, definir la vista arquitectónica, los estilos arquitectónicos, el patrón de arquitectura y la arquitectura tecnológica a utilizar. Establecer la conectividad entre las diferentes sucursales de la empresa.
Ingeniero de Software	El ingeniero de software se encarga de la gestión de requisitos, gestión de configuración y cambios, elaboración del modelo de datos, preparación de las pruebas funcionales, elaboración de la documentación. Elaborar modelos de implementación y despliegue. Captura, especificación y validación de requisitos, interactuando con el cliente y los usuarios mediante entrevistas. Elaboración del Modelo de Análisis y Diseño. Colaboración en la elaboración de las pruebas funcionales y el modelo de datos. Encargado además de la puesta en producción de la empresa.
Programador	El programador se encarga de la construcción de prototipos. Colaboración en la elaboración de las pruebas funcionales, modelo de datos y en las validaciones con el usuario.
Tester	El tester se encarga de la realización de las pruebas funcionales, de conectividad y rendimiento del sistema.

Tabla 1: *Roles y Responsabilidades*

Fuente: *Propia*

3.1.3 GESTIÓN DEL PROCESO

3.1.3.1 PLAN DEL PROYECTO

En esta sección se presenta la organización en fases e iteraciones y el calendario del proyecto.

PLAN DE FASES

El desarrollo se llevará a cabo en base a fases con una o más iteraciones en cada una de ellas. La siguiente tabla muestra la distribución de tiempos y el número de iteraciones de cada fase.

Fase	Nro. Iteraciones	Duración
Fase de Inicio	1	5 semanas
Fase de Elaboración	2	4 semanas
Fase de Construcción	3	12 semanas
Fase de Transición	2	4 semanas

Tabla 2: *Plan de Fases*

Fuente: *Propia*

Los hitos que marcan el final de cada fase se describen en la siguiente tabla:

Descripción	Hito
Fase de Inicio	Se desarrolla los requisitos del producto desde la perspectiva del usuario, los cuales serán establecidos en el artefacto Visión. Los principales casos de uso serán identificados y se hará un refinamiento al Plan de Desarrollo de Software. La aceptación del cliente/usuario del artefacto Visión y el Plan de Desarrollo de Software marcan el final de esta fase.
Fase de Elaboración	Se analizan los requisitos y se desarrolla un prototipo de arquitectura. Al final de esta fase, todos los casos de uso correspondientes a requisitos que serán implementados en la primera release de la fase de Construcción deben ser analizados y diseñados (en el Modelo de Análisis/Diseño). La revisión y aceptación del prototipo de la arquitectura del sistema marca el final de esta fase.
Fase de Construcción	Se terminan de analizar y diseñar todos los casos de uso, refinando el Modelo de Análisis/Diseño. El producto se construye en base a dos iteraciones, cada una produciendo un release a la cual se le aplican las pruebas y se valida con el cliente/usuario. Se comienza la elaboración del material de apoyo al usuario. El hito que marca el fin de esta fase es la versión de la release 3.0, con la capacidad operacional parcial del producto que se haya considerado como crítica, lista para ser entregada a los usuarios.
Fase de Transición	Se preparan dos release para distribución, asegurando una implantación y cambio del sistema previo de manera adecuada, incluyendo el entrenamiento de los usuarios. El hito que marca el fin de esta fase incluye, la entrega de toda la documentación del proyecto con los manuales de instalación y todo el material de apoyo al usuario, la finalización del entrenamiento de los usuarios y el empaquetado del producto.

Tabla 3: *Hitos de las Fases del Proyecto*

Fuente: *Propia*

CALENDARIO DEL PROYECTO

A continuación se presenta un calendario de las principales tareas del proyecto. Como se ha comentado, el proceso iterativo e incremental de RUP está caracterizado por la realización en paralelo de todas las disciplinas de desarrollo a lo largo del proyecto, con lo cual la mayoría de los artefactos son generados muy tempranamente en el proyecto pero van desarrollándose en mayor o menor grado de acuerdo a la fase e iteración del proyecto.

La siguiente figura ilustra este enfoque, en ella lo ensombrecido marca el énfasis de cada disciplina (workflow) en un momento determinado del desarrollo.

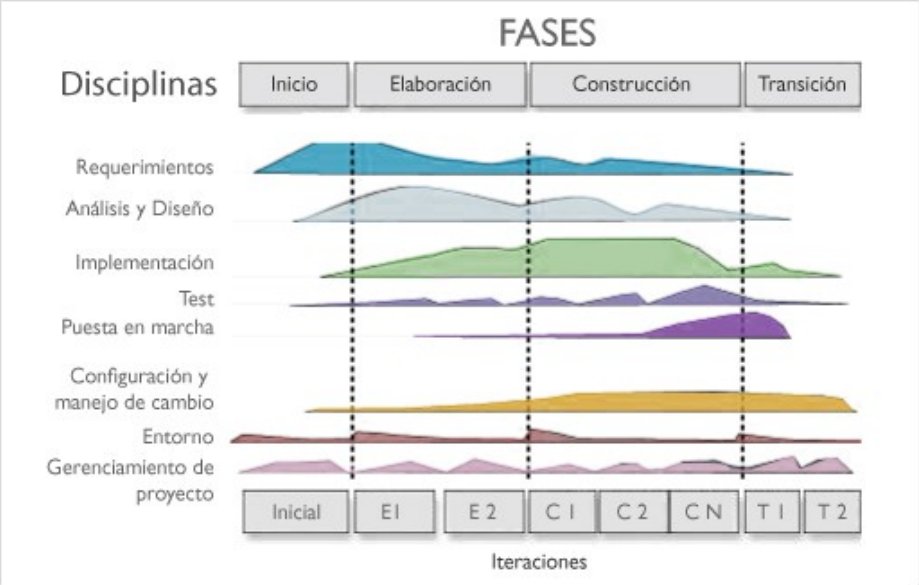


Ilustración 17: *Iteraciones del Proyecto*

Fuente: *Propia*

SEGUIMIENTO Y CONTROL DEL PROYECTO

Gestión de Requisitos

Los requisitos del sistema son especificados en el artefacto Visión. Cada requisito tendrá una serie de atributos tales como importancia, estado, iteración, donde se implementa, etc. Estos atributos permitirán realizar un efectivo seguimiento de cada requisito. Los cambios en los requisitos serán gestionados mediante una Solicitud de Cambio, las cuales serán evaluadas y distribuidas para asegurar la integridad del sistema y el correcto proceso de gestión y configuración de cambios.

Control de Plazos

El calendario del proyecto tendrá un seguimiento y evaluación semanal por el jefe de proyecto.

Control de Calidad

Los defectos detectados en las revisiones y formalizados también en una Solicitud de Cambio tendrán un seguimiento para asegurar la conformidad al respecto de la solución de dichas deficiencias. Para la revisión de cada artefacto y su correspondiente garantía se utilizará las guías de revisión y checklist (listas de verificación) incluidas en RUP.

Gestión de Riesgos

A partir de la fase de Inicio se mantendrá una lista de riesgos asociados al proyecto y de las acciones establecidas como estrategia para mitigarlos o acciones de contingencia. Esta lista será evaluada al menos una vez cada iteración.

Gestión de Configuración

Se realizará una gestión de configuración para llevar un registro de los artefactos generados y sus versiones. También se incluirá la gestión de las Solicitudes de Cambio y de las modificaciones que éstas produzcan, informando y publicando dichos cambios para que sean accesibles a todos los participantes en el proyecto.

Al final de cada iteración se establecerá un baseline (un registro del estado de cada artefacto, estableciendo una versión), la cual podrá ser modificada solo por una Solicitud de Cambio aprobada.

3.2 MODELADO DEL NEGOCIO

A continuación se presentan los modelos definidos en RUP como modelo del negocio, modelo de datos y modelo de análisis y diseño.

3.2.1 DIAGRAMAS DE CASOS DE USO

3.2.1.1 DIAGRAMA DE CASO DE USO: INVENTARIO DE VEHÍCULOS, ACCESORIOS Y HERRAMIENTAS

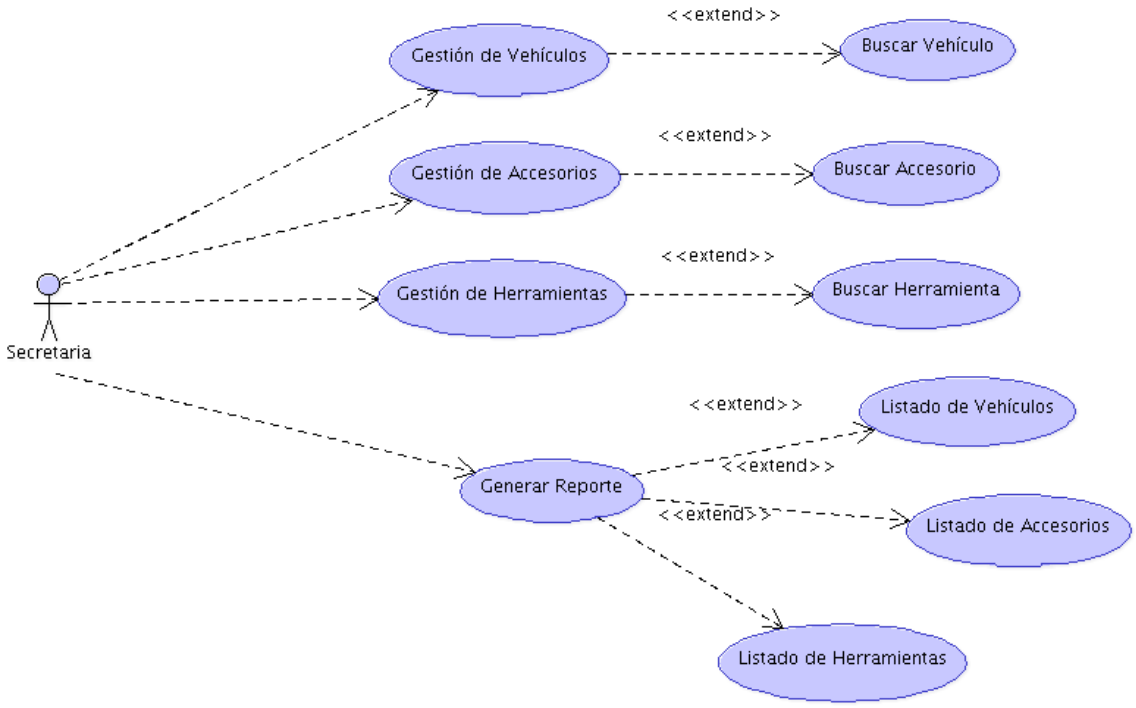


Ilustración 18: *Diagrama de Caso de Uso: Inventario de Vehículos, Accesorios y Herramientas*

Fuente: *Propia*

3.2.1.2 DIAGRAMA DE CASO DE USO: CONTROL DE MANTENIMIENTO

Ilustración 19: *Diagrama de Caso de Uso: Control de Mantenimiento*

Fuente: *Propia*

3.2.1.3 DIAGRAMA DE CASO DE USO: ORDEN DE MOVILIZACIÓN

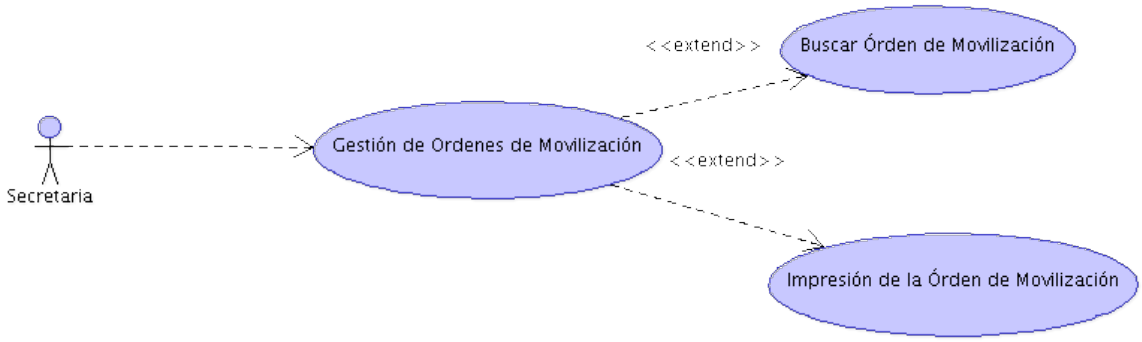


Ilustración 20: *Diagrama de Caso de Uso: Órden de Movilización*

Fuente: *Propia*

3.2.1.4 DIAGRAMA DE CASO DE USO: INFORME DIARIO DE MOVILIZACIÓN DE CADA VEHÍCULO

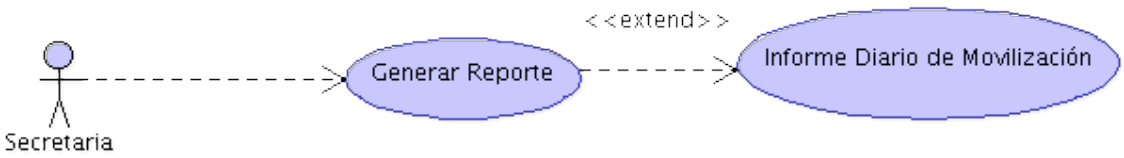


Ilustración 21: *Diagrama de Caso de Uso: Informe Diario de Movilización de cada Vehículo*

Fuente: *Propia*

3.2.1.5 DIAGRAMA DE CASO DE USO: CONTROL DE LUBRICANTES, COMBUSTIBLES Y REPUESTOS

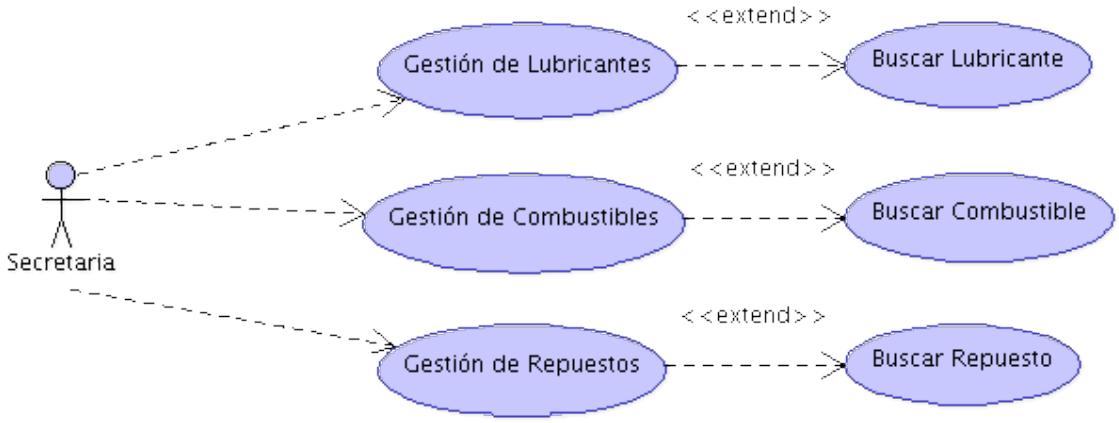


Ilustración 22: *Diagrama de Caso de Uso: Control de Lubricantes, Combustibles y Repuestos*

Fuente: *Propia*

3.2.1.6 DIAGRAMA DE CASO DE USO: ORDEN DE PROVISIÓN DE COMBUSTIBLE Y LUBRICANTES

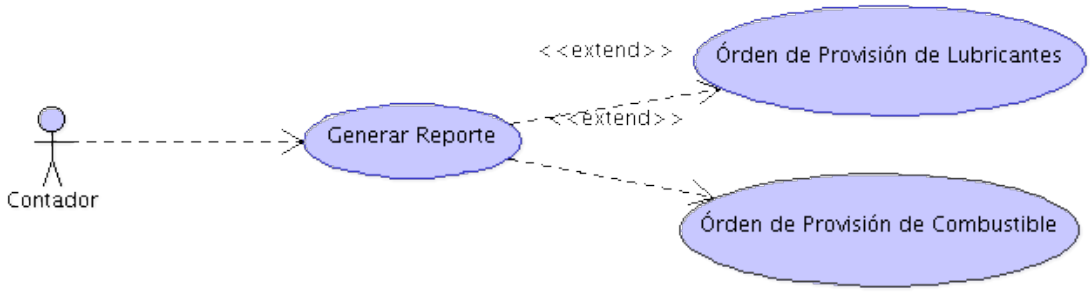


Ilustración 23: *Diagrama de Caso de Uso: Órden de Provisión de Combustible y Lubricantes*

Fuente: *Propia*

3.2.1.7 DIAGRAMA DE CASO DE USO: REGISTRO DE ENTRADA Y SALIDA DE VEHÍCULOS

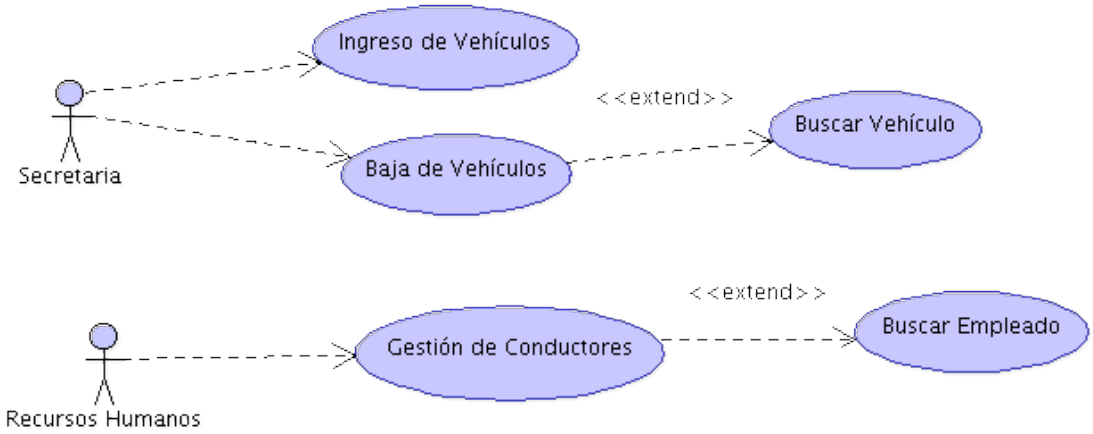


Ilustración 24: *Diagrama de Caso de Uso: Registro de Entrada y Salida de Vehículos*

Fuente: *Propia*

3.2.1.8 DIAGRAMA DE CASO DE USO: LIBRO DE NOVEDADES

Ilustración 25: *Diagrama de Caso de Uso: Libro de Novedades*

Fuente: *Propia*

3.2.1.9 DIAGRAMA DE CASO DE USO: ACTA DE ENTREGA RECEPCIÓN DE VEHÍCULOS

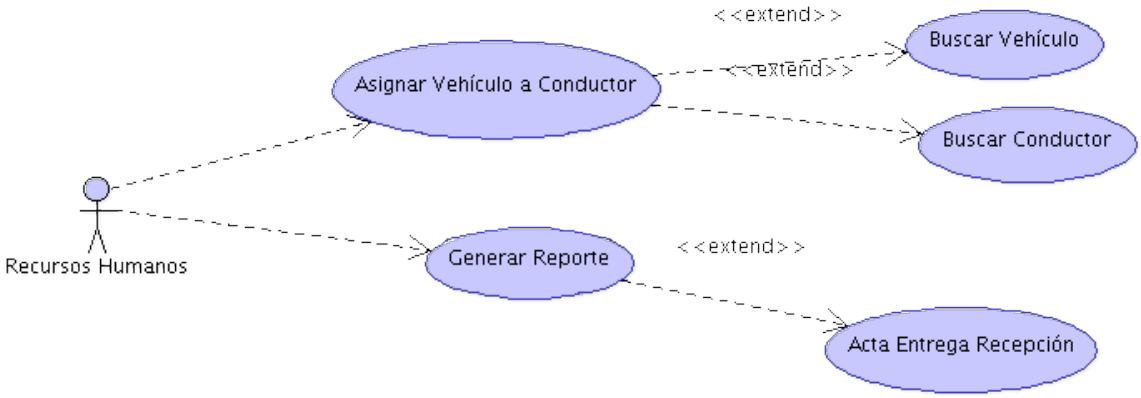


Ilustración 26: *Diagrama de Caso de Uso: Acta Entrega Recepción de Vehículos*

Fuente: *Propia*

3.2.2 ESPECIFICACIÓN DEL CASO DE USO: "INVENTARIO DE VEHÍCULOS, ACCESORIOS Y HERRAMIENTAS"

Este caso de uso permite a la secretaria tener una gestión de vehículos, para luego poder administrarlos en el funcionamiento del sistema, la secretaria podrá crear, editar y dar de baja los vehículos según sea el criterio de la Institución.

FLUJO DE EVENTOS

FLUJOS BÁSICOS

El sistema muestra la interfaz "Gestión de Vehículos" con los criterios de búsqueda que son los campos: placa, marca, modelo, tipo; además de las opciones buscar, editar, nuevo, eliminar.

CREAR VEHÍCULO

1. El caso de uso comienza cuando la secretaria solicita “Crear” en la interfaz “Gestión de Vehículos”.

2. El sistema muestra la interfaz “Crear Vehículo”.

3. La secretaria ingresa todos los datos del vehículo.

4. La secretaria elige la opción “Guardar” para guardar el nuevo vehículo creado.

5. El sistema guarda registro nuevo del vehículo ingresado.

6. El sistema genera mensaje de confirmación de creación de vehículo.

7. El sistema regresa a la interfaz “Gestión de Vehículos”.

EDITAR VEHÍCULO

1. La secretaria ingresa el criterio de búsqueda del vehículo en la interfaz “Gestión de Vehículos”.

2. La secretaria selecciona el botón “Buscar”.

3. Se ejecuta el caso de uso extendido “Buscar Vehículo”.

4. El sistema muestra los vehículos y sus datos en una tabla en la interfaz “Gestión de Vehículos”.

5. La secretaria presiona “Editar” en el vehículo que desea modificar.

6. La secretaria modifica los datos del vehículo según criterio.

7. La secretaria selecciona “Guardar”.

8. El sistema guarda los datos modificados y genera un mensaje “Actualización Completa”.

9. La secretaria presiona “Aceptar”.

10. El sistema regresa a la interfaz “Gestión de Vehículos”.

DAR DE BAJA VEHÍCULO

1. La secretaria ingresa al criterio de búsqueda del vehículo en la interfaz “Gestión de Vehículos”.

2. La secretaria selecciona buscar.

3. Se ejecuta el caso de uso extendido “Buscar Vehículo”.

4. El sistema muestra los vehículos y sus datos en una tabla en la interfaz “Gestión de Vehículos”.

5. La secretaria selecciona el vehículo que desea dar de baja.

6. La secretaria presiona el botón “Dar de Baja”.

7. El sistema muestra mensaje “Seguro que desea dar de baja el vehículo seleccionado”.

8. La secretaria confirma presionando el botón “Aceptar”.

9. El sistema da de baja el vehículo seleccionado de la base de datos.

FLUJOS ALTERNATIVOS

ERROR FALTA DE INGRESAR DATOS OBLIGATORIOS

En el punto cuatro de crear vehículo, cuando el usuario selecciona “Guardar” sin haber llenado todos los campos requeridos, el sistema muestra un mensaje de error “Campo requerido” en cada uno de los campos que son necesarios.

NO COLOCA CRITERIO DE BÚSQUEDA

Si en editar vehículo y dar de baja vehículo la secretaria presiona “Buscar” sin ingresar un criterio de búsqueda, el sistema mostrará en la tabla todos los vehículos ingresados hasta el momento.

PRECONDICIONES

La secretaria tiene que estar logeada en el sistema.
Que existan vehículos en la base de datos.

POSTCONDICIONES

El sistema ha actualizado la lista de vehículos.

PUNTOS DE EXTENSIÓN

El sistema llama al caso de uso “Buscar Vehículo”.

3.2.3 ESPECIFICACIÓN DEL CASO DE USO: “CONTROL DE MANTENIMIENTO”

Este caso de uso permite a la secretaria tener una gestión de mantenimiento, para luego poder administrarlos en el funcionamiento del sistema, la secretaria podrá crear y editar los mantenimientos según sea el criterio de la Institución.

FLUJO DE EVENTOS

FLUJOS BÁSICOS

El sistema muestra la interfaz “Gestión de Mantenimiento” los criterios de búsqueda que son los campos: fecha, vehículo, tipo de mantenimiento, factura, valor; además de las opciones buscar, editar, nuevo.

CREAR MANTENIMIENTO

1. El caso de uso comienza cuando la secretaria solicita “Crear” en la interfaz “Gestión Mantenimiento”.

2. El sistema muestra la interfaz “Crear Mantenimiento”.

3. La secretaria ingresa todos los datos del mantenimiento.

4. La secretaria elige la opción “Guardar” para guardar el nuevo mantenimiento creado.

5. El sistema guarda registro nuevo del mantenimiento ingresado.

6. El sistema genera mensaje de confirmación de creación de mantenimiento.

7. El sistema regresa a la interfaz “Gestión de Mantenimiento”.

EDITAR MANTENIMIENTO

1. La secretaria ingresa el criterio de búsqueda del mantenimiento en la interfaz “Gestión de Mantenimiento”.

2. La secretaria selecciona el botón “Buscar”.

3. Se ejecuta el caso de uso extendido “Buscar Mantenimiento”.

4. El sistema muestra los mantenimientos y sus datos en una tabla en la interfaz “Gestión de Mantenimiento”.

5. La secretaria presiona “Editar” en el mantenimiento que desea modificar.

6. La secretaria modifica los datos del mantenimiento según criterio.

7. La secretaria selecciona “Guardar”.

8. El sistema guarda los datos modificados y genera un mensaje “Actualización Completa”.

9. La secretaria presiona “Aceptar”.

10. El sistema regresa a la interfaz “Gestión de Mantenimiento”.

FLUJOS ALTERNATIVOS

ERROR FALTA DE INGRESAR DATOS OBLIGATORIOS

En el punto cuatro de crear mantenimiento, cuando el usuario selecciona “Guardar” sin haber llenado todos los campos requeridos, el sistema muestra un mensaje de error “Campo requerido” en cada uno de los campos que son necesarios.

NO COLOCA CRITERIO DE BÚSQUEDA

Si en editar mantenimiento la secretaria presiona “Buscar” sin ingresar un criterio de búsqueda, el sistema mostrará en la tabla todos los mantenimientos ingresados hasta el momento.

PRECONDICIONES

La secretaria tiene que estar logeada en el sistema.
Que existan mantenimientos en la base de datos.

POSTCONDICIONES

El sistema ha actualizado la lista de mantenimientos.

PUNTOS DE EXTENSIÓN

El sistema llama al caso de uso “Buscar Mantenimiento”.

3.2.4 ESPECIFICACIÓN DEL CASO DE USO: “ÓRDEN DE MOVILIZACIÓN”

Este caso de uso permite a la secretaria tener una gestión de órdenes de movilización, para luego poder administrarlos en el funcionamiento del sistema, la secretaria podrá crear y editar las órdenes de movilización según sea el criterio de la Institución.

FLUJO DE EVENTOS

FLUJOS BÁSICOS

El sistema muestra la interfaz “Gestión de Órdenes de Movilización” con los criterios de búsqueda que son los campos: fecha, Conductor, Empleado Responsable, Destino, Objeto de Salida; además de las opciones buscar, editar, nuevo.

CREAR ÓRDEN DE MOVILIZACIÓN

1. El caso de uso comienza cuando la secretaria solicita “Crear” en la interfaz “Gestión de Órdenes de Movilización”.

2. El sistema muestra la interfaz “Crear Órden de Movilización”.

3. La secretaria ingresa todos los datos de la Órden de Movilización.

4. La secretaria elige la opción “Guardar” para guardar la nueva órden de movilización creada.

5. El sistema guarda registro nuevo de la órden de movilización ingresada.

6. El sistema genera mensaje de confirmación de creación de órden de movilización.

7. El sistema regresa a la interfaz “Gestión de Órdenes de Movilización”.

EDITAR ÓRDEN DE MOVILIZACIÓN

1. La secretaria ingresa el criterio de búsqueda de la órden de movilización en la interfaz “Gestión de Órdenes de Movilización”.

2. La secretaria selecciona el botón “Buscar”.

3. Se ejecuta el caso de uso extendido “Buscar Órden de Movilización”.

4. El sistema muestra las órdenes de movilización y sus datos en una tabla en la interfaz “Gestión de Órdenes de Movilización”.

5. La secretaria presiona “Editar” en la Órden de Movilización que desea modificar.

6. La secretaria modifica los datos de Órden de Movilización según criterio.

7. La secretaria selecciona “Guardar”.

8. El sistema guarda los datos modificados y genera un mensaje “Actualización Completa”.

9. La secretaria presiona “Aceptar”.

10. El sistema regresa a la interfaz “Gestión de Órdenes de Movilización”.

FLUJOS ALTERNATIVOS

ERROR FALTA DE INGRESAR DATOS OBLIGATORIOS

En el punto cuatro de crear Órden de Movilización, cuando el usuario selecciona “Guardar” sin haber llenado todos los campos requeridos, el sistema muestra un mensaje de error “Campo requerido” en cada uno de los campos que son necesarios.

NO COLOCA CRITERIO DE BÚSQUEDA

Si en editar Órden de Movilización la secretaria presiona “Buscar” sin ingresar un criterio de búsqueda, el sistema mostrará en la tabla todos los vehículos ingresados hasta el momento.

PRECONDICIONES

La secretaria tiene que estar logeada en el sistema.
Que existan Órdenes de Movilización en la base de datos.

POSTCONDICIONES

El sistema ha actualizado la lista de Órdenes de Movilización.

PUNTOS DE EXTENSIÓN

El sistema llama al caso de uso “Buscar Órden de Movilización”.

3.2.5 ESPECIFICACIÓN DEL CASO DE USO: “INFORME DIARIO DE MOVILIZACIÓN”

Este caso de uso permite a la secretaria generar el informe diario de movilización, para luego poder visualizar e imprimir dicho informe.

FLUJO DE EVENTOS

FLUJOS BÁSICOS

El sistema muestra la interfaz “Informe diario de movilización” con sus parámetros que son los campos: fecha y vehículo.

VISUALIZAR INFORME DIARIO DE MOVILIZACIÓN

1. El caso de uso comienza cuando la secretaria ingresa todos los parámetros necesarios para la visualización del informe.

2. La secretaria elige la opción “Imprimir” para visuzalizar el informe.

3. El sistema genera el informe y lo muestra en formato PDF.

4. El sistema genera mensaje de confirmación de generación del informe.

FLUJOS ALTERNATIVOS

ERROR FALTA DE INGRESAR DATOS OBLIGATORIOS

En el punto dos cuando el usuario selecciona “Imprimir” sin haber llenado todos los campos requeridos, el sistema muestra un mensaje de error “Campo requerido” en cada uno de los campos que son necesarios.

PRECONDICIONES

La secretaria tiene que estar logeada en el sistema.
Que existan Órdenes de Movilización en la base de datos.

PUNTOS DE EXTENSIÓN

El sistema llama al caso de uso “Informe Diario de movilización”.

3.2.6 ESPECIFICACIÓN DEL CASO DE USO: “CONTROL DE LUBRICANTES, COMBUSTIBLES Y REPUESTOS”

Este caso de uso permite a la secretaria tener una gestión de lubricantes para luego poder administrarlos en el funcionamiento del sistema, la secretaria podrá crear y editar los lubricantes según sea el criterio de la Institución.

FLUJO DE EVENTOS

FLUJOS BÁSICOS

El sistema muestra la interfaz “Gestión de Lubricantes” con los criterios de búsqueda que son los campos: vehículo, tipo, número, valor; además de las opciones buscar, editar, nuevo.

CREAR CONTROL LUBRICANTE

1. El caso de uso comienza cuando la secretaria solicita “Crear” en la interfaz “Gestión de Lubricantes”.

2. El sistema muestra la interfaz “Crear Lubricante”.

3. La secretaria ingresa todos los datos del Lubricante.

4. La secretaria elige la opción “Guardar” para guardar el nuevo lubricante creado.

5. El sistema guarda registro nuevo del lubricante ingresado.

6. El sistema genera mensaje de confirmación de creación de lubricante.

7. El sistema regresa a la interfaz “Gestión de Lubricantes”.

EDITAR LUBRICANTE

1. La secretaria ingresa el criterio de búsqueda del lubricante en la interfaz “Gestión de Lubricantes”.

2. La secretaria selecciona el botón “Buscar”.

3. Se ejecuta el caso de uso extendido “Buscar Lubricante”.

4. El sistema muestra los lubricantes y sus datos en una tabla en la interfaz “Gestión de Lubricantes”.

5. La secretaria presiona “Editar” en el lubricante que desea modificar.

6. La secretaria modifica los datos del lubricante según criterio.

7. La secretaria selecciona “Guardar”.

8. El sistema guarda los datos modificados y genera un mensaje “Actualización Completa”.

9. La secretaria presiona “Aceptar”.

10. El sistema regresa a la interfaz “Gestión de Lubricantes”.

FLUJOS ALTERNATIVOS

ERROR FALTA DE INGRESAR DATOS OBLIGATORIOS

En el punto cuatro de crear lubricante, cuando el usuario selecciona “Guardar” sin haber llenado todos los campos requeridos, el sistema muestra un mensaje de error “Campo requerido” en cada uno de los campos que son necesarios.

NO COLOCA CRITERIO DE BÚSQUEDA

Si en editar lubricante la secretaria presiona “Buscar” sin ingresar un criterio de búsqueda, el sistema mostrará en la tabla todos los lubricantes ingresados hasta el momento.

PRECONDICIONES

La secretaria tiene que estar logeada en el sistema.
Que existan lubricantes en la base de datos.

POSTCONDICIONES

El sistema ha actualizado la lista de lubricantes.

PUNTOS DE EXTENSIÓN

El sistema llama al caso de uso “Buscar Lubricante”.

Orden de provisión de combustible y lubricantes

3.2.7 ESPECIFICACIÓN DEL CASO DE USO: “ORDEN DE PROVISIÓN DE COMBUSTIBLE Y LUBRICANTES

Este caso de uso permite al contador generar la Orden de provisión de combustible y lubricantes, para luego poder visualizar e imprimir dicho informe.

FLUJO DE EVENTOS

FLUJOS BÁSICOS

El sistema muestra la interfaz “Orden de provisión de combustible y lubricantes” con sus parámetros que son los campos: fecha y vehículo.

VISUALIZAR ORDEN DE PROVISIÓN DE COMBUSTIBLE Y LUBRICANTES

1. El caso de uso comienza cuando el contador ingresa todos los parámetros necesarios para la visualización del informe.

2. El contador elige la opción “Imprimir” para visuzalizar el informe.

3. El sistema genera el informe y lo muestra en formato PDF.

4. El sistema genera mensaje de confirmación de generación del informe.

FLUJOS ALTERNATIVOS

ERROR FALTA DE INGRESAR DATOS OBLIGATORIOS

En el punto dos cuando el usuario selecciona “Imprimir” sin haber llenado todos los campos requeridos, el sistema muestra un mensaje de error “Campo requerido” en cada uno de los campos que son necesarios.

PRECONDICIONES

El contador tiene que estar logeado en el sistema.
Que existan Órdenes de provisión de combustible y lubricantes en la base de datos.

PUNTOS DE EXTENSIÓN

El sistema llama al caso de uso “Orden de provisión de combustible y lubricantes”.

3.2.8 ESPECIFICACIÓN DEL CASO DE USO: “REGISTRO DE ENTRADA Y SALIDA DE VEHÍCULOS”

Este caso de uso permite a la persona de recursos humanos tener una gestión de conductores, para luego poder administrarlos en el funcionamiento del sistema, la persona de recursos humanos podrá crear, editar y dar de baja los conductores según sea el criterio de la Institución.

FLUJO DE EVENTOS

FLUJOS BÁSICOS

El sistema muestra la interfaz “Gestión de Conductores” con los criterios de búsqueda que son los campos: nombres, apellidos, tipo de licencia, modelo, tipo; además de las opciones buscar, editar, nuevo, dar de baja.

CREAR CONDUCTOR

1. El caso de uso comienza cuando la persona de recursos humanos solicita “Crear” en la interfaz “Gestión de Conductores”.

2. El sistema muestra la interfaz “Crear Conductor”.

3. La persona de recursos humanos ingresa todos los datos del conductor.

4. La persona de recursos humanos elige la opción “Guardar” para guardar el nuevo conductor creado.

5. El sistema guarda registro nuevo del conductor ingresado.

6. El sistema genera mensaje de confirmación de creación de conductor.

7. El sistema regresa a la interfaz “Gestión de Conductores”.

EDITAR CONDUCTOR

1. La persona de recursos humanos ingresa el criterio de búsqueda del conductor en la interfaz “Gestión de Conductores”.

2. La persona de recursos humanos selecciona el botón “Buscar”.

3. Se ejecuta el caso de uso extendido “Buscar Empleado”.

4. El sistema muestra los empleados y sus datos en una tabla en la interfaz “Gestión de Conductores”.

5. La persona de recursos humanos presiona “Editar” en el conductor que desea modificar.

6. La persona de recursos humanos modifica los datos del conductor según criterio.

7. La persona de recursos humanos selecciona “Guardar”.

8. El sistema guarda los datos modificados y genera un mensaje “Actualización Completa”.

9. La persona de recursos humanos presiona “Aceptar”.

10. El sistema regresa a la interfaz “Gestión de Conductores”.

DAR DE BAJA CONDUCTOR

1. La persona de recursos humanos ingresa al criterio de búsqueda del conductor en la interfaz “Gestión de Vehículos”.

2. La persona de recursos humanos selecciona buscar.

3. Se ejecuta el caso de uso extendido “Buscar Empleado”.

4. El sistema muestra los conductores y sus datos en una tabla en la interfaz “Gestión de Conductores”.

5. La persona de recursos humanos selecciona el conductor que desea dar de baja.

6. La persona de recursos humanos presiona el botón “Dar de Baja”.

7. El sistema muestra mensaje “Seguro que desea dar de baja el conductor seleccionado”.

8. La persona de recursos humanos confirma presionando el botón “Aceptar”.

9. El sistema da de baja el conductor seleccionado de la base de datos.

FLUJOS ALTERNATIVOS

ERROR FALTA DE INGRESAR DATOS OBLIGATORIOS

En el punto cuatro de crear conductor, cuando el usuario selecciona “Guardar” sin haber llenado todos los campos requeridos, el sistema muestra un mensaje de error “Campo requerido” en cada uno de los campos que son necesarios.

NO COLOCA CRITERIO DE BÚSQUEDA

Si en editar conductor y dar de baja conductor la persona de recursos humanos presiona "Buscar" sin ingresar un criterio de búsqueda, el sistema mostrará en la tabla todos los conductores ingresados hasta el momento.

PRECONDICIONES

La persona de recursos humanos tiene que estar logeada en el sistema.
Que existan conductores en la base de datos.

POSTCONDICIONES

El sistema ha actualizado la lista de conductores.

PUNTOS DE EXTENSIÓN

El sistema llama al caso de uso "Buscar Empleado".

3.2.9 ESPECIFICACIÓN DEL CASO DE USO: "LIBRO DE NOVEDADES"

Este caso de uso permite a la secretaria generar reporte, para luego poder visualizar e imprimir dicho informe.

FLUJO DE EVENTOS

FLUJOS BÁSICOS

El sistema muestra la interfaz "Libro de novedades" con sus parámetros que son los campos: empleado, fecha, hora .

VISUALIZAR LIBRO DE NOVEDADES

1. El caso de uso comienza cuando la secretaria ingresa todos los parámetros necesarios para la visualización del informe.

2. La secretaria elige la opción “Imprimir” para visuzalizar el informe.

3. El sistema genera el informe y lo muestra en formato PDF.

4. El sistema genera mensaje de confirmación de generación del informe.

FLUJOS ALTERNATIVOS

ERROR FALTA DE INGRESAR DATOS OBLIGATORIOS

En el punto dos cuando el usuario selecciona “Imprimir” sin haber llenado todos los campos requeridos, el sistema muestra un mensaje de error “Campo requerido” en cada uno de los campos que son necesarios.

PRECONDICIONES

La secretaria tiene que estar logeada en el sistema.
Que exista Novedades en la base de datos.

PUNTOS DE EXTENSIÓN

El sistema llama al caso de uso “Libro de Novedades”.

3.2.9.1 ESPECIFICACIÓN DEL CASO DE USO: “ACTA DE ENTREGA RECEPCIÓN DE LOS VEHÍCULOS”

Este caso de uso permite a la persona de recursos humanos asignar vehículo a conductor, para luego poder administrarlos en el funcionamiento del sistema, la persona de recursos humano podrá crear, editar y dar de baja a las personas asignadas a los vehículos según sea el criterio de la Institución.

FLUJO DE EVENTOS

FLUJOS BÁSICOS

El sistema muestra la interfaz “Asignar vehículo a conductor” con los criterios de búsqueda que son los campos: placa, marca, modelo, tipo; además de las opciones buscar, editar, nuevo, dar de baja.

ASIGNAR VEHÍCULO A CONDUCTOR

1. El caso de uso comienza cuando la persona de recursos humanos solicita “Buscar” en la interfaz “Asignar vehículo a conductor”.

2. El sistema muestra la interfaz “Asignar vehículo a conductor”.

3. La persona de recursos humanos busca todos los datos de conductores disponibles.

4. La persona de recursos humanos busca todos los datos de vehiculos disponibles.

5. La persona de recursos humanos elige la opción “Asignar” para guardar la asignación de conductor al vehículo .

6. El sistema guarda registro nuevo del vehículo asignado a conductor.

7. El sistema genera mensaje de confirmación de asignación de vehículo.

EDITAR ASIGNACIÓN DE VEHÍCULO A CONDUCTOR

1. La persona de recursos humanos ingresa el criterio de búsqueda de la asignación de vehículo en la interfaz “Asignar vehículo a conductor”.

2. La persona de recursos humanos selecciona el botón “Buscar”.

3. Se ejecuta los casos de usos extendido “Buscar Vehículo” y “Buscar Conductor”.

4. El sistema muestra los vehículos y sus datos y los conductores y sus datos en una tabla en la interfaz “Asignar vehículo a conductor”.

5. La persona de recursos humanos presiona “Editar” en el vehículo y/o conductor que desea modificar.

6. La persona de recursos humanos modifica los datos del vehículo y/o conductor según criterio.

7. La persona de recursos humanos selecciona “Guardar”.

8. El sistema guarda los datos modificados y genera un mensaje “Actualización Completa”.

9. La persona de recursos humanos presiona “Aceptar”.

10. El sistema regresa a la interfaz “Asignar vehículo a conductor”.

FLUJOS ALTERNATIVOS

ERROR FALTA DE INGRESAR DATOS OBLIGATORIOS

En el punto cuatro de asignar vehículo a conductor, cuando el usuario selecciona “Guardar” sin haber llenado todos los campos requeridos, el sistema muestra un mensaje de error “Campo requerido” en cada uno de los campos que son necesarios.

NO COLOCA CRITERIO DE BÚSQUEDA

Si en editar asignar vehículo a conductor y editar asignar vehículo a conductor la persona de recursos humanos presiona “Buscar” sin ingresar un criterio de búsqueda, el sistema mostrará en la tabla todos los vehículos asignados a conductor ingresados hasta el momento.

PRECONDICIONES

La persona de recursos humanos tiene que estar logeada en el sistema.
Que existan vehículos asignados a conductores en la base de datos.

POSTCONDICIONES

El sistema ha actualizado la lista de vehículos asignados a conductores.

PUNTOS DE EXTENSIÓN

El sistema llama al caso de uso “Buscar Vehículo” y “Buscar Conductor”.

3.3 REQUISITOS

3.3.1 POSICIONAMIENTO

3.3.1.1 OPORTUNIDAD DE NEGOCIO

Este sistema permitirá a la empresa informatizar el control de sus actividades (orden de movilización, control de lubricantes combustible y herramientas, etc), lo cual supondrá un acceso rápido y sencillo a los datos, gracias a interfaces gráficas sencillas y amigables. Además los datos accedidos estarán siempre actualizados, lo cual es un factor muy importante para poder llevar un control centralizado de la información.

3.3.1.2 SENTENCIA QUE DEFINE EL PROBLEMA

El problema de	El Control vehicular se lo realiza de una manera informal. Las órdenes de movilización se realizan de forma manual. La emisión de las órdenes de provisión de combustibles y lubricantes se lo realiza de forma manual. El registro de entrada / salida de los vehículos y su entrega – recepción a los conductores se lo realiza de forma manual.
Afecta a	Contador, Bodegueros, Vendedores, Departamento de Recursos Humanos.
El impacto asociado es	Almacenar toda la información referente a las compras y ventas que realiza la empresa, y que esta información este al instante y actualizada en lugares físicamente muy distantes es un proceso prácticamente imposible de realizar en el caso que no este informatizado.
Una solución adecuada sería	Informatizar todo el proceso, usando una red local con una base de datos accesible desde los distintos nodos de la red y generar interfaces amigables y sencillas con las que acceder a dicha base de datos.
Para	Gerente Contador, Secretaria, Persona de Recursos Humanos.
Quienes	Controlan los procesos de la empresa.
El nombre del producto	Es una herramienta de software.
Que	Almacena la información necesaria para gestionar los procesos de control vehicular.
No como	El sistema actual.
Nuestro producto	Permite gestionar las distintas actividades de la empresa mediante una interfaz gráfica sencilla y amigable. Además, proporciona un acceso rápido y actualizado a la información desde cualquier punto que tenga acceso a la base de datos.

Tabla 4: Sentencia que define el problema del proyecto

Fuente: *Propia*

3.3.2 DESCRIPCIÓN DE STAKEHOLDERS (PARTICIPANTES EN EL PROYECTO) Y USUARIOS

Para proveer de forma efectiva productos y servicios que se ajusten a las necesidades de los usuarios, es necesario identificar e involucrar a todos los participantes en el proyecto como parte del modelado de requerimientos. También es necesario identificar a los usuarios del sistema y asegurarse de que el conjunto de participantes en el proyecto los representa adecuadamente.

Esta sección muestra un perfil de los participantes y de los usuarios involucrados en el proyecto, así como los problemas más importantes que estos perciben para enfocar la solución propuesta hacia ellos. No describe sus requisitos específicos ya que éstos se capturan mediante otro artefacto. En lugar de esto proporciona la justificación de por que éstos requisitos son necesarios.

3.3.2.1 RESUMEN DE LOS STAKEHOLDERS

Nombre	Descripción	Responsabilidades
Gerente de la Institución	Representante global de la Institución.	El Stakeholder realiza: • Representa a todos los usuarios posibles del sistema. • Seguimiento del desarrollo del proyecto. • Aprueba requisitos y funcionalidades.

Tabla 5: *Resumen de los Stakeholders del Proyecto*

Fuente: *Propia*

3.3.2.2 RESUMEN DE LOS USUARIOS

ACT1 Secretaria	Encargada del Ingreso de vehículos, accesorios y herramientas; y emisión de las órdenes de movilización.	STK1 Secretaria
ACT2 Contador	Encargado de Emisión de órdenes de consumo de lubricantes, combustibles y repuestos.	STK2 Contador
ACT3 Recursos Humanos	Encargado de Ingreso de Conductores.	STK3 Recursos Humanos
ACT4 Gerente	Encargado de la Visualización de todos los reportes generados	STK4 Gerente

Tabla 6: *Resumen de los Usuarios*

Fuente: *Propia*

3.3.2.3 ENTORNOS DE USUARIOS

Los usuarios necesitan disponer de un navegador web únicamente. Los reportes pueden ser generados tanto en pdf, html, procesadores de texto y hojas de cálculo

3.3.2.4 PERFIL DE LOS STAKEHOLDERS

REPRESENTANTE DEL ÁREA TÉCNICA Y SISTEMAS DE INFORMACIÓN

Representante	Gerente de la Institución.
Descripción	Representante global de la Institución.
Tipo	Experto en negocios.
Responsabilidades	Encargado de mostrar las necesidades de cada usuario del sistema. Además, lleva a cabo un seguimiento del desarrollo del proyecto y aprobación de los requisitos y funcionalidades del sistema.
Criterio de éxito	A definir por el cliente.
Grado de participación	Revisión de requerimientos, estructura del sistema.

Tabla 7: *Representante del Área Técnica del Proyecto*

Fuente: *Propia*

3.3.2.5 PERFILES DE USUARIO

SECRETARIA

Representante	STK1 SECRETARIA.
Descripción	Secretaria.
Tipo	Usuario experto.
Responsabilidades	Responsable del Ingreso de vehículos, accesorios y herramientas; y emisión de las órdenes de movilización.
Criterio de éxito	A definir por el cliente.
Grado de participación	A definir por el cliente.

Tabla 8: *Usuario: Secretaria*

Fuente: *Propia*

CONTADOR

Representante	STK2 CONTADOR.
Descripción	Contador.
Tipo	Usuario experto.
Responsabilidades	Encargado de Emisión de órdenes de consumo de lubricantes, combustibles y repuestos.
Criterio de éxito	A definir por el cliente.
Grado de participación	A definir por el cliente.

Tabla 9: *Usuario: Contador*

Fuente: *Propia*

RECURSOS HUMANOS

Representante	STK3 RECURSOS HUMANOS.
Descripción	RECURSOS HUMANOS.
Tipo	Usuario experto.
Responsabilidades	Encargado de Ingreso de Conductores.
Criterio de éxito	A definir por el cliente.
Grado de participación	A definir por el cliente.

Tabla 10: *Usuario: Recursos Humanos*

Fuente: *Propia*

3.3.3 DESCRIPCIÓN GLOBAL DEL PRODUCTO

3.3.3.1 PERSPECTIVA DEL PRODUCTO

El producto a desarrollar es un “Sistema de Control Vehicular” para la Institución, con la intención de agilizar su funcionamiento. Las áreas a tratar por el sistema son: secretaría contabilidad, recursos humanos

3.3.3.2 RESUMEN DE LAS CARACTERÍSTICAS

A continuación se mostrará un listado de los beneficios que obtendrá el cliente a partir del producto.

Beneficio del cliente	Características que lo apoyan
Gestión automatizada del Ingreso de vehículos, accesorios y herramientas; y emisión de las órdenes de movilización.	Sistema de optimización del control vehicular
Mayor agilidad en la Emisión de órdenes de consumo de lubricantes, combustibles y repuestos.	Aplicación que le permite poder realizar la emisión de las órdenes.
Mayor facilidad para la gestión de recursos humanos.	Base de datos centralizada con la información de todo el personal.
Mayor control de los productos.	El sistema gestionará los productos del almacén, generará reportes.
Seguridad.	El ingreso del sistema se controla por medio de un usuario y contraseña, se controla el acceso a las opciones a través de permisos de roles, y demás seguridades que proveen los marcos de trabajo.

Tabla 11: *Resumen de las Características del Proyecto*

Fuente: *Propia*

3.3.3.3 SUPOSICIONES Y DEPENDENCIAS

Para garantizar el término de una transacción de reserva se pretende minimizar el peso y uso de gráficos que impidan el uso ágil de la página. Todo término de transacción satisfactorio o insatisfactorio será notificado inmediatamente al usuario.

3.3.3.4 DESCRIPCIÓN GLOBAL DEL PRODUCTO

CSW1: SECRETARÍA

El departamento de secretaría tendrá acceso al Ingreso de vehículos, accesorios y herramientas; y emisión de las órdenes de movilización.

CSW2: CONTABILIDAD

El departamento de contabilidad tendrá acceso a la de Emisión de órdenes de consumo de lubricantes, combustibles y repuestos.

CSW4: RECURSOS HUMANOS

El departamento de recursos humanos tendrá acceso al Ingreso de Conductores.

CSW5: GERENTE

El gerente tendrá acceso a la visualización de todos los reportes generados.

CSW6: SEGURIDAD

El administrador del sistema será encargado del manejo del módulo de Seguridad y de la administración de los servidores donde se encontrará el sistema.

3.3.4 RESTRICCIONES

Las restricciones que el sistema presenta y advierte a sus usuarios es la necesidad de contar una adecuada conectividad en la Institución y un navegador web (firefox, opera, safari, internet explorer, etc).

El hardware requerido por la Institución es el adecuado y además se ha efectuado una inversión de un servidor para el alojamiento del servidor de aplicaciones y del sistema de gestión de base de datos relacional.

3.3.5 OTROS REQUISITOS DEL PROYECTO

3.3.5.1 ESTÁNDARES APLICABLES

Lenguaje para el diseño de páginas WEB: HTML avanzado (compatible desde cualquier navegador web).
Sistema de Gestión de Base de Datos Relacional: PostgreSQL.
Protocolo de Comunicación: TCP/IP versión 4.

3.3.5.2 REQUISITOS DE SISTEMA

Como ya se mencionó anteriormente el requerimiento esencial para los usuarios es contar con una conexión adecuada de internet o un canal de datos, y un navegador web (firefox, opera, safari, internet explorer, etc).

3.3.5.3 REQUISITOS DE DESEMPEÑO

El requerimiento mayor de rendimiento estará dado por la facilidad de acceso al sistema de la Institución. Se ha considerado un diseño vistoso pero ligero de peso.

3.3.5.4 REQUISITOS DE ENTORNO

Los dispositivos de red y servidores, tendrán que ser fijados en un rack para cumplir con algunos estándares de cableado estructurado. Para los productos se necesita un ambiente con temperaturas: -5<15<+5 °C.

3.3.6 REQUISITOS DE DOCUMENTACIÓN

3.3.6.1 MANUAL DE USUARIO

Los manuales de usuario podrán ser descargados directamente desde el sistema. El manual de usuario contendrá toda la documentación de la instalación, uso y mantenimiento del sistema. Será enfocado a cada rol de usuario, ya que cada uno maneja diferentes tipos de información.

3.3.6.2 AYUDA EN LÍNEA

La ayuda en línea podrá ser accedida de dos formas:

- Usando los diferentes hipervínculos situados cerca de las opciones relevantes del sistema.

- Mediante el acceso a nuestra página web, con la posibilidad de consultas y sugerencias.

3.3.6.3 GUÍAS DE INSTALACIÓN, CONFIGURACIÓN Y FICHERO LÉAME

Se encuentra en los anexos digitales del proyecto.

CAPÍTULO IV

4 CONSTRUCCIÓN DE LA APLICACIÓN

4.1 ANÁLISIS/DISEÑO

A continuación se presentan los modelos definidos en RUP como modelo de datos y modelo de análisis/diseño (Diagrama de Clases, Modelo Relacional).

4.1.1 DIAGRAMA DE CLASES

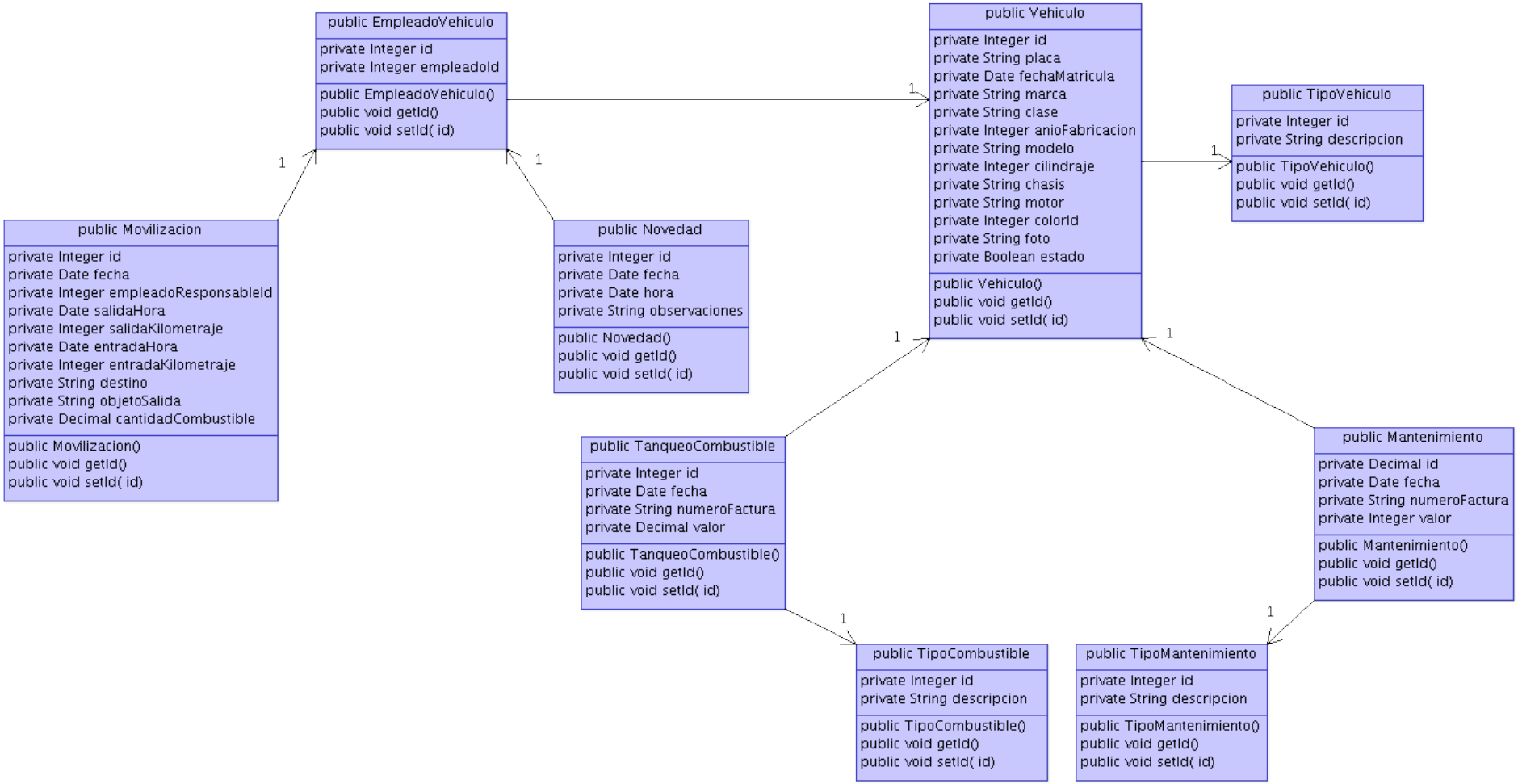


Ilustración 27: *Diagrama de Clases*

Fuente: *Propia*

4.1.2 MODELO ENTIDAD RELACIÓN

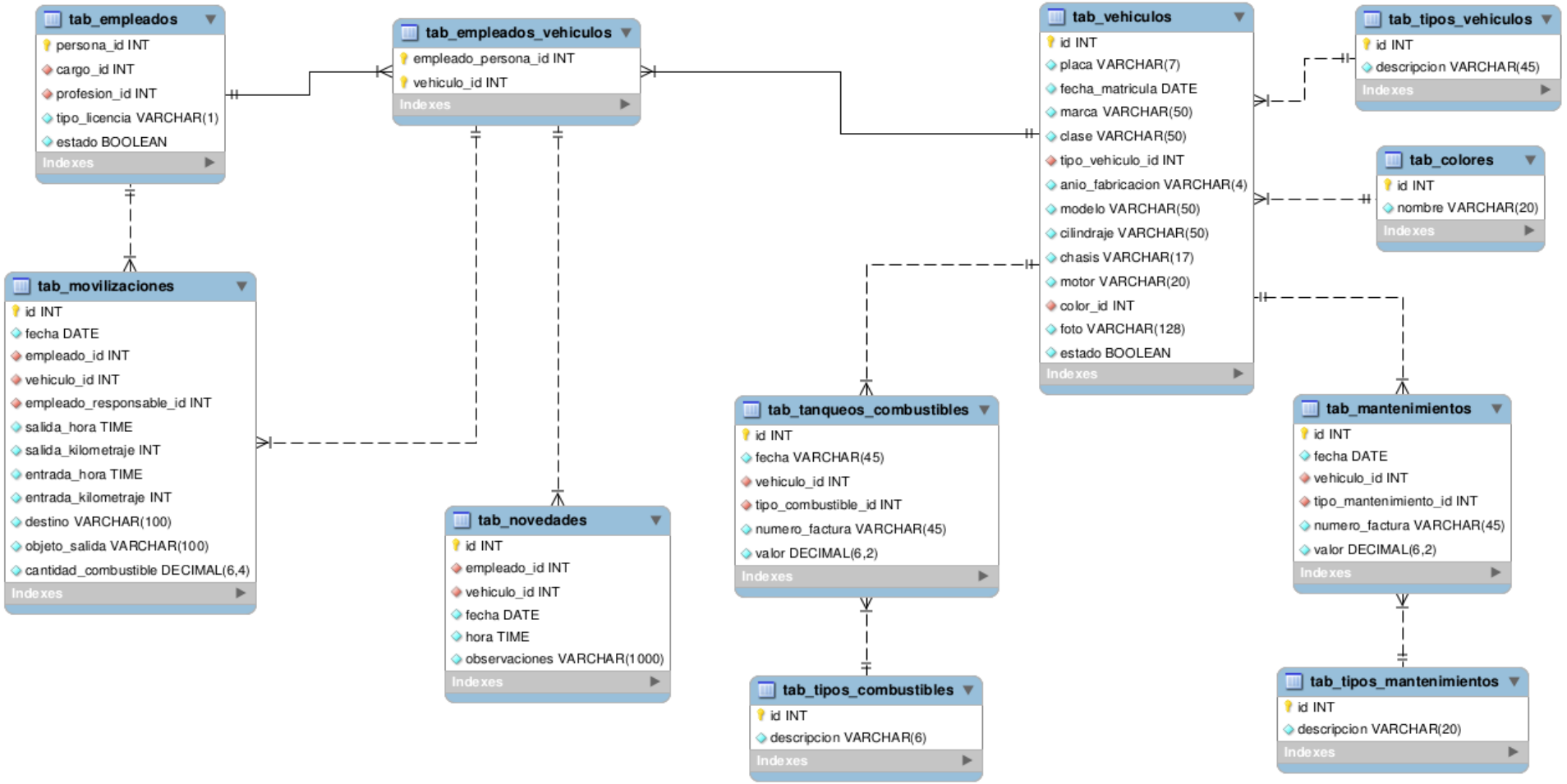


Ilustración 28: *Modelo Entidad Relación*

Fuente: *Propia*

4.2 IMPLEMENTACIÓN

4.2.1 PROTOTIPOS DE INTERFACES DE USUARIO

A continuación se presentan los prototipos de interfaces gráficas de usuario diseñadas para la aplicación final.

PRINCIPAL

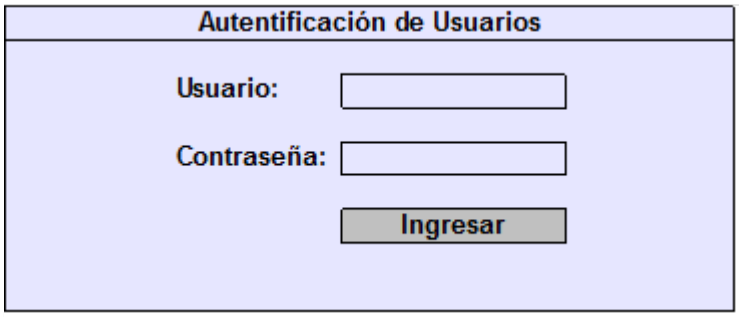


Ilustración 29: *Interfaz de Usuario: Autentificación de Usuarios*

Fuente: *Propia*

MENÚ PRINCIPAL

Ilustración 30: *Interfaz de Usuario: Menú Principal*

Fuente: *Propia*

MENÚ DEL MÓDULO DE CONTROL VEHICULAR

LISTADO DE ÍTEM

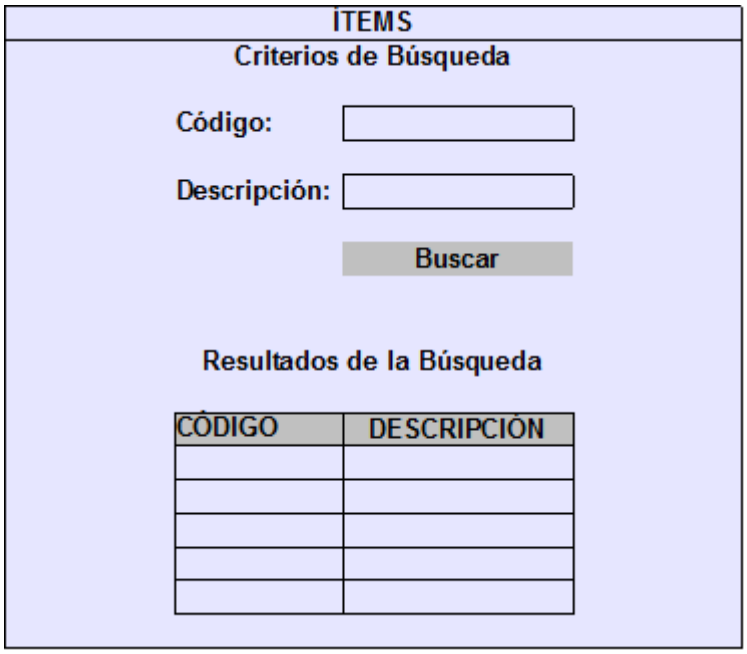


Ilustración 31: *Interfaz de Usuario: Listado de Ítem*

Fuente: *Propia*

CREAR Y/O EDITAR ÍTEM

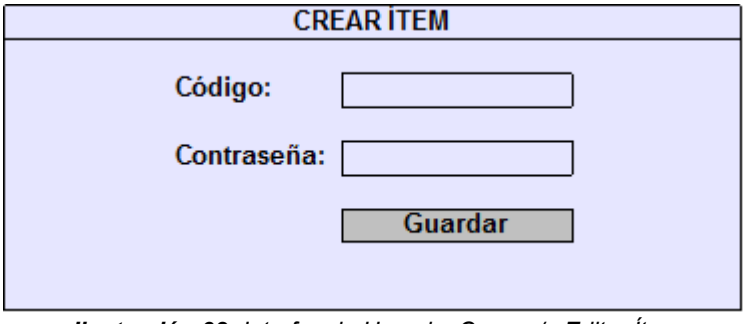


Ilustración 32: *Interfaz de Usuario: Crear y/o Editar Ítem*

Fuente: *Propia*

4.2.2 DIAGRAMA DE COMPONENTES

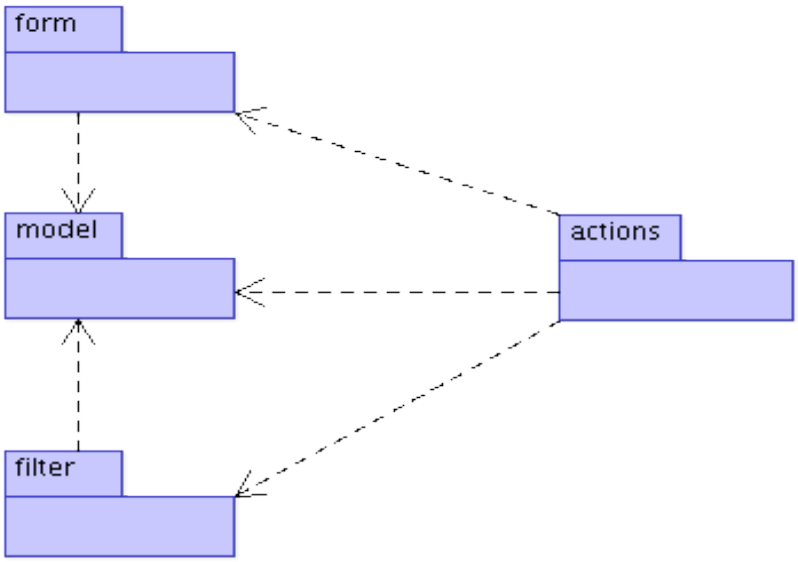


Ilustración 33: *Diagrama de Componentes*

Fuente: *Propia*

4.2.3 DIAGRAMA DE DESPLIEGUE

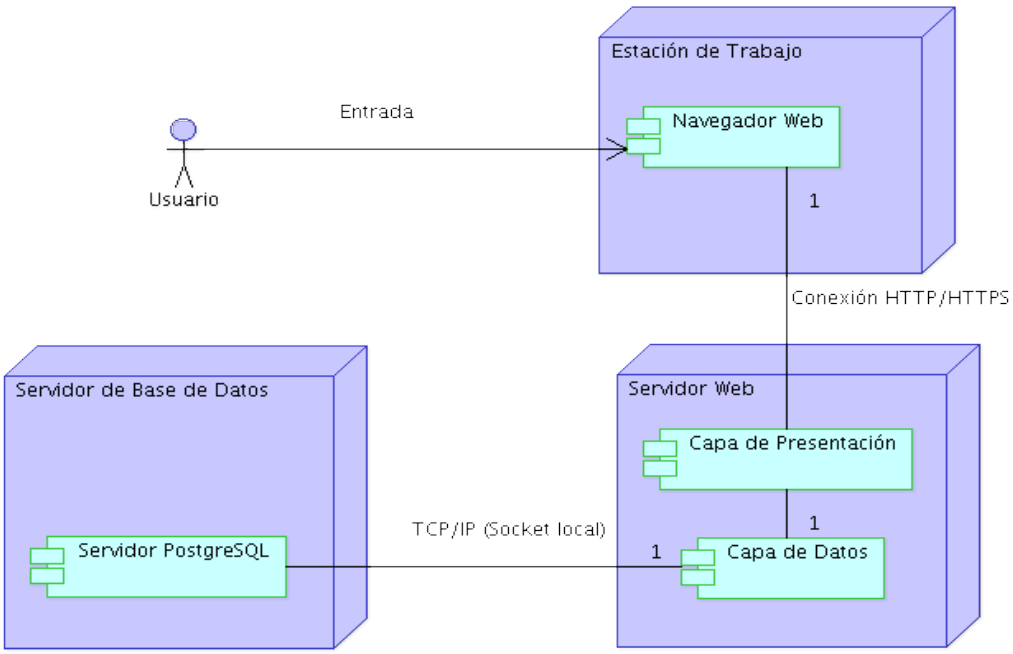


Ilustración 34: *Diagrama de Despliegue*

Fuente: *Propia*

4.2.4 ARQUITECTURA DE SOFTWARE

4.2.4.1 PROPÓSITO

Este documento tiene como propósito describir la vista general de la arquitectura de software del Sistema de Control Vehicular para lo cual se utilizan vistas arquitectónicas para representar los diferentes aspectos del sistema.

4.2.4.2 ALCANCE

Este documento de arquitectura de software proporciona una descripción del “Sistema de Control Vehicular”. Este documento ha sido generado directamente del análisis de los casos de uso a automatizar y del modelo de diseño.

4.2.4.3 REPRESENTACIÓN ARQUITECTÓNICA

Modelo de Caso de Uso: Describe los procesos que brindarán al negocio la funcionalidad automatizada deseada y cómo funcionan internamente, contiene el modelo de caso de uso.

Modelo de Análisis: Describe un primer bosquejo de las clases de análisis que servirán de soporte para el diseño.

Modelo de la Experiencia del Usuario: Describe las pantallas del sistema, el contenido dinámico de pantallas y como el usuario navega a través de las pantallas para ejecutar las funcionalidades del sistema.

Modelo de Diseño: Describe las partes arquitectónicas significativas del modelo de diseño, tales como su descomposición en módulos y paquetes.

En el desarrollo del sistema se ha utilizado el patrón de arquitectura Modelo Vista Controlador (MVC).

4.2.4.4 OBJETIVOS ARQUITECTÓNICOS Y COACCIONES

Los principales objetivos del “Sistema de Control Vehicular” es la automatización de todos los procesos de control vehicular para un mejor control de los mismos, además se podrá obtener reportes actualizados de estos procesos.

Todos los requerimientos estipulados en el documento de la Visión deben ser tomados en consideración durante el desarrollo de la arquitectura.

La construcción principal del diseño y de la implementación ha sido que la aplicación debe funcionar bajo una plataforma que consiste en los componentes siguientes:

Servidor Web: Apache HTTP Server 2.2
Lenguaje de Programación: PHP 5.3
Sistema de Gestión de Base de Datos Relacional: PostgreSQL 9.1
Framework: Symfony PHP Web 1.4
Entorno de Desarrollo: NetBeans 7.0

4.2.4.5 VISTA DE CASOS DE USO

Nos presenta una vista de los casos de uso de la arquitectura de software. La vista de casos de uso es la entrada importante de la selección de contexto y/o los casos de uso que son el punto de una interacción. Los casos de uso del “Sistema de Control Vehicular”.

4.2.4.6 CASOS DE USO ARQUITECTÓNICAMENTE SIGNIFICATIVOS

Los caso de uso arquitectónicamente significativos son aquellos que representan las partes más críticas de la arquitectura del sistema y demuestran la funcionalidad del sistema. Además de lo ya mencionado, para el “Sistema de Control Vehicular” los casos de uso significativos han sido priorizados en base al soporte que brindan a las metas del negocio.

Inventario de vehículos, accesorios y herramientas

Permitirá a la secretaria tener una gestión de vehículos, para luego poder administrarlos en el funcionamiento del sistema, la secretaria podrá crear, editar y dar de baja los vehículos según sea el criterio de la Institución.

Orden de Movilización

Permitirá a la secretaria tener una gestión de órdenes de movilización, para luego poder administrarlos en el funcionamiento del sistema, la secretaria podrá crear y editar las órdenes de movilización según sea el criterio de la Institución.

4.2.4.7 VISTA LÓGICA

Esta sección describe la estructura lógica del sistema. Empieza con la descripción de la arquitectura y después presenta sus elementos estructurales y del comportamiento dominantes.

4.2.4.8 ELEMENTOS DEL MODELO ARQUITECTURALMENTE SIGNIFICATIVO

El “Sistema de Control Vehicular” se ha descompuesto en lo siguientes módulos:
Vehículos, Accesorios y Herramientas, Movilizaciones, Novedades y Accidentes, Lubricantes, combustibles y repuestos, Conductores, Seguridad, Registro con Código de Barras, Reportes.

Cada uno de los componentes del negocio se divide más a fondo en las tres capas del patrón de arquitectura Modelo Vista Controlador (MVC).

1. Lógica de la presentación,
2. Lógica del negocio, y
3. Lógica de la integración.

Es decir la arquitectura descompone los sistemas a la largo de dos dimensiones:

1. La primera dimensión está a lo largo de las líneas de la funcionalidad del sistema.

2. La segunda dimensión está a lo largo de las capas comúnmente reconocidas que separan tres clases de preocupaciones:

    a) Preocupaciones de la presentación, o cómo manejar la comunicación con el usuario y controlar su acceso a los servicios y a los recursos de sistema.

    b) Preocupaciones de negocio, o cómo organizar los elementos del sistema que realizan funciones de los servicios del negocio y del sistema, y

    c) Preocupaciones de la integración, o cómo conectar los elementos del sistema con el mecanismo de la persistencia, otros sistemas, los dispositivos físicos, etc.

4.2.5 ESPECIFICACIÓN DEL CASO DE PRUEBA: INVENTARIO DE VEHÍCULOS, ACCESORIOS Y HERRAMIENTAS

ESCENARIO: FLUJO BÁSICO

Validar que el usuario pueda ingresar un vehículo correctamente.

Descripción: Es el escenario ideal del caso de uso, no deberían presentarse errores.

PRECONDICIONES

El usuario debe tener su sesión activa en el sistema.

DATOS DE ENTRADA

Se accederá al sistema con el usuario “prueba” cuya contraseña es “123”. Su perfil es Secretaria.

FLUJO DE ACTIVIDADES

Paso	Instrucción	Resultado esperado
1	El caso de uso comienza cuando la secretaria solicita “Crear” en la interfaz “Gestión de Vehículos”.	El sistema muestra la interfaz de “Gestión de Vehículos”.
2	El sistema muestra la interfaz “Crear Vehículo”.	El sistema muestra la interfaz “Crear Vehículos”
3	La secretaria ingresa todos los datos del vehículo.	Campos ingresados correctamente.
4	La secretaria elige la opción “Guardar” para guardar el nuevo vehículo creado.	Preparar el almacenamiento del vehículo en el sistema.
5	El sistema guarda registro nuevo del vehículo ingresado.	Guardado con éxito el nuevo vehículo en el sistema.
6	El sistema genera mensaje de confirmación de creación de vehículo.	Registro satisfactorio
7	El sistema regresa a la interfaz “Gestión de Vehículos”.	El sistema muestra la interfaz de inicio de Gestión de Vehículos.

Tabla 12: *Flujo de Actividades del escenario Flujo Básico del Caso de Prueba Inventario de vehículos, accesorios y herramientas del Proyecto*

Fuente: *Propia*

PUNTOS DE REVISIÓN

Paso	Punto de Control	Validación a realizar
1	Carga la interfaz de “Crear Vehículo”.	Al momento de intentar acceder a la interfaz de “Crear Vehículo” el sistema verifica si el usuario logeado tiene permiso de acceder a dicha interfaz.
2	Guardar el vehículo creado.	Verificar si el sistema almacena correctamente el vehículo creado en la base de datos.
	El sistema regresa a la interfaz “Gestión de Vehículos”.	El sistema debe mostrar la interfaz “Gestión de Vehículos”.

Tabla 13: *Punto de Revisión del escenario Flujo Básico del Caso de Prueba Inventario de vehículos, accesorios y herramientas del proyecto*

Fuente: *Propia*

4.2.6 ESPECIFICACIÓN DEL CASO DE PRUEBA: CONTROL DE MANTENIMIENTO

ESCENARIO: FLUJO BÁSICO

El sistema muestra la interfaz “Gestión de Mantenimiento”.

Descripción: Es el escenario ideal del caso de uso, no deberían presentarse errores.

PRECONDICIONES

La secretaria tiene que estar logeada en el sistema.

DATOS DE ENTRADA

Se accederá al sistema con el usuario “prueba” cuya contraseña es “123”. Su perfil es Secretaria.

FLUJO DE ACTIVIDADES

Paso	Instrucción	Resultado esperado
1	El caso de uso comienza cuando la secretaria solicita “Crear” en la interfaz “Gestión Mantenimiento”.	El sistema muestra la interfaz de “Gestión Mantenimiento”.
2	El sistema muestra la interfaz “Crear Mantenimiento”.	El sistema muestra la interfaz “Crear Mantenimiento”
3	La secretaria ingresa todos los datos del mantenimiento.	Campos ingresados correctamente.
4	La secretaria elige la opción “Guardar” para guardar el nuevo mantenimiento creado.	Preparar el almacenamiento del mantenimiento en el sistema.
5	El sistema guarda registro nuevo del mantenimiento ingresado.	Guardado con éxito el nuevo mantenimiento en el sistema.
6	El sistema genera mensaje de confirmación de creación de mantenimiento.	Registro satisfactorio
7	El sistema regresa a la interfaz “Gestión de Mantenimiento”.	El sistema muestra la interfaz de inicio de Gestión de Mantenimiento.

Tabla 14: *Flujo de Actividades del escenario Flujo Básico del Caso de Prueba Control de Mantenimiento del Proyecto*

Fuente: *Propia*

PUNTOS DE REVISIÓN

Paso	Punto de Control	Validación a realizar
1	Carga la interfaz de "Crear Mantenimiento".	Al momento de intentar acceder a la interfaz de "Crear Mantenimiento" el sistema verifica si el usuario logeado tiene permiso de acceder a dicha interfaz.
2	Guardar el mantenimiento creado.	Verificar si el sistema almacena correctamente el mantenimiento creado en la base de datos.
	El sistema regresa a la interfaz "Gestión de Mantenimiento".	El sistema debe mostrar la interfaz "Gestión de Mantenimiento".

Tabla 15: *Punto de Revisión del escenario Flujo Básico del Caso de Prueba Inventario de vehículos, accesorios y herramientas del Proyecto*

Fuente: *Propia*

4.2.7 ESPECIFICACIÓN DEL CASO DE PRUEBA: ORDEN DE MOVILIZACIÓN

ESCENARIO: FLUJO BÁSICO

El sistema muestra la interfaz "Gestión de Órdenes de Movilización".

Descripción: Es el escenario ideal del caso de uso, no deberían presentarse errores.

PRECONDICIONES

La secretaria tiene que estar logeada en el sistema.

DATOS DE ENTRADA

Se accederá al sistema con el usuario "prueba" cuya contraseña es "123". Su perfil es Secretaria.

FLUJO DE ACTIVIDADES

Paso	Instrucción	Resultado esperado
1	El caso de uso comienza cuando la secretaria solicita “Crear” en la interfaz “Gestión de Órdenes de Movilización”.	El sistema muestra la interfaz de “Gestión de Órdenes de Movilización”.
2	El sistema muestra la interfaz “Crear Orden de Movilización”.	El sistema muestra la interfaz “Crear Orden de Movilización”
3	La secretaria ingresa todos los datos de la Orden de Movilización.	Campos ingresados correctamente.
4	La secretaria elige la opción “Guardar” para guardar la nueva orden de movilización creada.	Preparar el almacenamiento de la orden de movilización en el sistema.
5	El sistema guarda registro nuevo de la orden de movilización ingresada.	Guardado con éxito de la nueva orden de movilización en el sistema.
6	El sistema genera mensaje de confirmación de creación de orden de movilización.	Registro satisfactorio.
7	El sistema regresa a la interfaz “Gestión de Órdenes de Movilización”.	El sistema muestra la interfaz de inicio de Gestión de Órdenes de Movilización.

Tabla 16: *Flujo de Actividades del escenario Flujo Básico del Caso de Prueba Orden de Movilización del Proyecto*

Fuente: *Propia*

PUNTOS DE REVISIÓN

Paso	Punto de Control	Validación a realizar
1	Carga la interfaz de “Crear Orden de Movilización”.	Al momento de intentar acceder a la interfaz de “Crear Orden de Movilización” el sistema verifica si el usuario logeado tiene permiso de acceder a dicha interfaz.
2	Guardar la Orden de Movilización creada.	Verificar si el sistema almacena correctamente la Orden de Movilización creada en la base de datos.
	El sistema regresa a la interfaz “Gestión de Órdenes de Movilización”.	El sistema debe mostrar la interfaz “Gestión de Órdenes de Movilización”.

Tabla 17: *Punto de Revisión del escenario Flujo Básico del Caso de Prueba Orden de Movilización del Proyecto*

Fuente: *Propia*

4.2.8 ESPECIFICACIÓN DEL CASO DE PRUEBA: INFORME DIARIO DE MOVILIZACIÓN

ESCENARIO: FLUJO BÁSICO

El sistema muestra la interfaz “Informe diario de movilización”.

Descripción: Es el escenario ideal del caso de uso, no deberían presentarse errores.

PRECONDICIONES

La secretaria tiene que estar logeada en el sistema.

DATOS DE ENTRADA

Se accederá al sistema con el usuario “prueba” cuya contraseña es “123”. Su perfil es Secretaria.

FLUJO DE ACTIVIDADES

Paso	Instrucción	Resultado esperado
1	El caso de uso comienza cuando la secretaria ingresa todos los parámetros necesarios para la visualización del informe.	El sistema muestra la interfaz de "Informe diario de movilización".
2	La secretaria elige la opción "Imprimir" para visualizar el informe.	Inicio de generación de informe.
3	El sistema genera el informe y lo muestra en formato PDF.	Generación de informe satisfactorio.
4	El sistema genera mensaje de confirmación de generación del informe.	Informe listo para impresión.

Tabla 18: *Flujo de Actividades del escenario Flujo Básico del Caso de Prueba Informe Diario de Movilización del Proyecto*

Fuente: *Propia*

PUNTOS DE REVISIÓN

Paso	Punto de Control	Validación a realizar
1	Carga la interfaz “Informe diario de movilización”.	Al momento de intentar acceder a la interfaz de “Informe diario de movilización” el sistema verifica si el usuario logeado tiene permiso de acceder a dicha interfaz.
2	El sistema genera el informe y lo muestra en formato PDF. .	Verificar si el sistema genera correctamente el Informe diario de movilización creado en la base de datos.
	El sistema regresa a la interfaz “Informe diario de movilización”.	El sistema debe mostrar la interfaz “Informe diario de movilización”.

Tabla 19: *Punto de Revisión del escenario Flujo Básico del Caso de Prueba Informe Diario de Movilización*

Fuente: *Propia*

4.2.9 ESPECIFICACIÓN DEL CASO DE PRUEBA: CONTROL DE LUBRICANTES, COMBUSTIBLES Y REPUESTOS

ESCENARIO: FLUJO BÁSICO

El sistema muestra la interfaz “Gestión de Lubricantes”.

Descripción: Es el escenario ideal del caso de uso, no deberían presentarse errores.

PRECONDICIONES

La secretaria tiene que estar logeada en el sistema.

DATOS DE ENTRADA

Se accederá al sistema con el usuario “prueba” cuya contraseña es “123”. Su perfil es Secretaria.

FLUJO DE ACTIVIDADES

Paso	Instrucción	Resultado esperado
1	El caso de uso comienza cuando la secretaria solicita “Crear” en la interfaz “Gestión de Lubricantes”.	El sistema muestra la interfaz de “Gestión de Lubricantes”.
2	El sistema muestra la interfaz “Crear Lubricante”.	El sistema muestra la interfaz “Crear Lubricante”
3	La secretaria ingresa todos los datos del Lubricante.	Campos ingresados correctamente.
4	La secretaria elige la opción “Guardar” para guardar el nuevo lubricante creado.	Preparar el almacenamiento del lubricante en el sistema.
5	El sistema guarda registro nuevo del lubricante ingresado.	Guardado con éxito el nuevo lubricante en el sistema.
6	El sistema genera mensaje de confirmación de creación de lubricante.	Registro satisfactorio
7	El sistema regresa a la interfaz “Gestión de Lubricantes”.	El sistema muestra la interfaz de inicio de “Gestión de Lubricantes”.

Tabla 20: *Flujo de Actividades del escenario Flujo Básico del Caso de Prueba Control de lubricantes, combustibles y repuestos del Proyecto*

Fuente: *Propia*

PUNTOS DE REVISIÓN

Paso	Punto de Control	Validación a realizar
1	Carga la interfaz de “Crear Lubricante”.	Al momento de intentar acceder a la interfaz de “Crear Lubricante” el sistema verifica si el usuario logeado tiene permiso de acceder a dicha interfaz.
2	Guardar el lubricante creado.	Verificar si el sistema almacena correctamente el lubricante creado en la base de datos.
3	El sistema regresa a la interfaz “Gestión de Lubricantes”.	El sistema debe mostrar la interfaz “Gestión de Lubricantes”.

Tabla 21: Punto de revisión del escenario Flujo Básico del Caso de Prueba Control de lubricantes, combustibles y repuestos del Proyecto

Fuente: *Propia*

4.2.10 ESPECIFICACIÓN DEL CASO DE PRUEBA: ORDEN DE PROVISIÓN DE COMBUSTIBLE Y LUBRICANTES

ESCENARIO: FLUJO BÁSICO

El sistema muestra la interfaz “Orden de provisión de combustible y lubricantes”.

Descripción: Es el escenario ideal del caso de uso, no deberían presentarse errores.

PRECONDICIONES

El contador tiene que estar logeado en el sistema.

DATOS DE ENTRADA

Se accederá al sistema con el usuario “prueba” cuya contraseña es “123”. Su perfil es Secretaria.

FLUJO DE ACTIVIDADES

Paso	Instrucción	Resultado esperado
1	El caso de uso comienza cuando el contador ingresa todos los parámetros necesarios para la visualización del informe.	El sistema muestra la interfaz de "Orden de provisión de combustible y lubricantes".
2	El contador elige la opción "Imprimir" para visualizar el informe.	Inicio de generación de informe.
3	El sistema genera el informe y lo muestra en formato PDF.	Generación de informe satisfactorio.
4	El sistema genera mensaje de confirmación de generación del informe.	Informe listo para impresión.

Tabla 22: *Flujo de Actividades del escenario Flujo Básico del Caso de Prueba Orden de provisión de combustible y lubricantes del Proyecto*

Fuente: *Propia*

PUNTOS DE REVISIÓN

Paso	Punto de Control	Validación a realizar
1	Carga la interfaz "Orden de provisión de combustible y lubricantes".	Al momento de intentar acceder a la interfaz de "Orden de provisión de combustible y lubricantes" el sistema verifica si el usuario logeado tiene permiso de acceder a dicha interfaz.
2	El sistema genera el informe y lo muestra en formato PDF. .	Verificar si el sistema genera correctamente la Orden de provisión de combustible y lubricantes creado en la base de datos.
3	El sistema regresa a la interfaz "Orden de provisión de combustible y lubricantes".	El sistema debe mostrar la interfaz "Orden de provisión de combustible y lubricantes".

Tabla 23: *Punto de revisión del escenario Flujo Básico del Caso de Prueba Orden de provisión de combustible y lubricantes del Proyecto*

Fuente: *Propia*

4.2.11 ESPECIFICACIÓN DEL CASO DE PRUEBA: REGISTRO DE ENTRADA Y SALIDA DE VEHÍCULOS

ESCENARIO: FLUJO BÁSICO

El sistema muestra la interfaz “Gestión de Conductores”.

Descripción: Es el escenario ideal del caso de uso, no deberían presentarse errores.

PRECONDICIONES

La persona de recursos humanos tiene que estar logeada en el sistema.

DATOS DE ENTRADA

Se accederá al sistema con el usuario “prueba” cuya contraseña es “123”. Su perfil es Secretaria.

FLUJO DE ACTIVIDADES

Paso	Instrucción	Resultado esperado
1	El caso de uso comienza cuando la persona de recursos humanos solicita "Crear" en la interfaz "Gestión de Conductores".	El sistema muestra la interfaz de "Gestión de Conductores".
2	El sistema muestra la interfaz "Crear Conductor".	El sistema muestra la interfaz "Crear Conductor"
3	La persona de recursos humanos ingresa todos los datos del conductor.	Campos ingresados correctamente.
4	La persona de recursos humanos elige la opción "Guardar" para guardar el nuevo conductor creado.	Preparar el almacenamiento del conductor en el sistema.
5	El sistema guarda registro nuevo del conductor ingresado.	Guardado con éxito el nuevo conductor en el sistema.
6	El sistema genera mensaje de confirmación de creación de conductor.	Registro satisfactorio
7	El sistema regresa a la interfaz "Gestión de Conductores".	El sistema muestra la interfaz de inicio de "Gestión de Conductores".

Tabla 24: *Flujo de Actividades del escenario Flujo Básico del Caso de Prueba Registro de Entrada y Salida de Vehículos del Proyecto*

Fuente: *Propia*

PUNTOS DE REVISIÓN

Paso	Punto de Control	Validación a realizar
1	Carga la interfaz de "Crear Conductor".	Al momento de intentar acceder a la interfaz de "Crear Conductor" el sistema verifica si el usuario logeado tiene permiso de acceder a dicha interfaz.
2	Guardar el conductor creado.	Verificar si el sistema almacena correctamente el conductor creado en la base de datos.
3	El sistema regresa a la interfaz "Gestión de Conductores".	El sistema debe mostrar la interfaz "Gestión de Conductores".

Tabla 25: *Punto de Revisión del escenario Flujo Básico del Caso de Prueba Registro de Entrada y Salida de*

Fuente: *Propia*

4.2.12 ESPECIFICACIÓN DEL CASO DE PRUEBA: LIBRO DE NOVEDADES

ESCENARIO: FLUJO BÁSICO

El sistema muestra la interfaz “Libro de novedades”.

Descripción: Es el escenario ideal del caso de uso, no deberían presentarse errores.

PRECONDICIONES

La secretaria tiene que estar logeada en el sistema.

DATOS DE ENTRADA

Se accederá al sistema con el usuario “prueba” cuya contraseña es “123”. Su perfil es Secretaria.

FLUJO DE ACTIVIDADES

Paso	Instrucción	Resultado esperado
1	El caso de uso comienza cuando la secretaria ingresa todos los parámetros necesarios para la visualización del informe.	El sistema muestra la interfaz de “Libro de novedades”.
2	La secretaria elige la opción “Imprimir” para visualizar el informe.	Inicio de generación de informe.
3	El sistema genera el informe y lo muestra en formato PDF.	Generación de informe satisfactorio.
4	El sistema genera mensaje de confirmación de generación del informe.	Informe listo para impresión.

Tabla 26: *Flujo de Actividades del escenario Flujo Básico del Caso de Prueba Libro de Novedades del Proyecto*

Fuente: *Propia*

PUNTOS DE REVISIÓN

Paso	Punto de Control	Validación a realizar
1	Carga la interfaz “Libro de novedades”.	Al momento de intentar acceder a la interfaz de “Libro de novedades” el sistema verifica si el usuario logeado tiene permiso de acceder a dicha interfaz.
2	El sistema genera el informe y lo muestra en formato PDF. .	Verificar si el sistema genera correctamente el Libro de novedades creado en la base de datos.
3	El sistema regresa a la interfaz “Libro de novedades”.	El sistema debe mostrar la interfaz “Libro de novedades”.

Tabla 27: *Punto de Revisión del escenario Flujo Básico del Caso de Prueba Libro de Novedades*

Fuente: *Propia*

4.2.13 ESPECIFICACIÓN DEL CASO DE PRUEBA: ACTA DE ENTREGA RECEPCIÓN DE LOS VEHÍCULOS

ESCENARIO: FLUJO BÁSICO

El sistema muestra la interfaz “Asignar vehículo a conductor”.

Descripción: Es el escenario ideal del caso de uso, no deberían presentarse errores.

PRECONDICIONES

La secretaria tiene que estar logeada en el sistema.

DATOS DE ENTRADA

Se accederá al sistema con el usuario “prueba” cuya contraseña es “123”. Su perfil es Secretaria.

FLUJO DE ACTIVIDADES

Paso	Instrucción	Resultado esperado
1	El caso de uso comienza cuando la persona de recursos humanos solicita "Buscar" en la interfaz "Asignar vehículo a conductor".	El sistema muestra la interfaz de "Asignar vehículo a conductor".
2	El sistema muestra la interfaz "Asignar vehículo a conductor".	El sistema muestra la interfaz "Asignar"
3	La persona de recursos humanos busca todos los datos de conductores disponibles.	Listar datos de conductores disponibles.
4	La persona de recursos humanos busca todos los datos de vehículos disponibles.	Listar datos de vehículos disponibles.
5	La persona de recursos humanos elige la opción "Asignar" para guardar la asignación de conductor al vehículo.	Guardado con éxito el nuevo conductor asignado a vehículo en el sistema.
6	El sistema guarda registro nuevo del vehículo asignado a conductor.	Registro satisfactorio
7	El sistema genera mensaje de confirmación de asignación de vehículo.	El sistema muestra la interfaz de inicio de "Asignar vehículo a conductor".

Tabla 28: *Flujo de Actividades del escenario Flujo Básico del Caso de Prueba Acta de Entrega Recepción de los vehículos del Proyecto*

Fuente: *Propia*

PUNTOS DE REVISIÓN

Paso	Punto de Control	Validación a realizar
1	Carga la interfaz de “Asignar vehículo a conductor”.	Al momento de intentar acceder a la interfaz de “Asignar vehículo a conductor” el sistema verifica si el usuario logeado tiene permiso de acceder a dicha interfaz.
2	Guardar vehículo asignado a conductor.	Verificar si el sistema almacena correctamente el vehículo asignado a conductor creado en la base de datos.
3	El sistema regresa a la interfaz “Asignar vehículo a conductor”.	El sistema debe mostrar la interfaz “Asignar vehículo a conductor”.

Tabla 29: *Punto de Revisión del escenario Flujo Básico del Caso de Prueba Acta de Entrega de Recepción de los vehículos del Proyecto*

Fuente: *Propia*

CAPÍTULO V

5 CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

1. Los objetivos planteados en mi trabajo de grado fueron comprobados ya que se logró un software de altísima calidad, que cumplió con los objetivos planteados.

2. El contar con este aplicativo realizado en aplicación web nos permite acceder a la información extrayendo reportes en línea de manera rápida, atendiendo a solicitudes realizadas por la Institución.

3. Implementar el software permitió contar con un sistema de calidad para el desempeño y productividad de sus empleados, y simplificó el control de acceso a la información del control vehicular (Órdenes de Movilización, Control de Mantenimientos, Órdenes de Lubricantes y Combustibles, Registro de Conductores, etc), contando con información organizada, centralizada y de fácil acceso.

4. Este proyecto permitió un cambio organizacional previo y durante la implementación, atendiendo los requerimientos en los servicios de secretaria, contabilidad, recursos humanos y gerencia.

5. Con la Implementación del registro con código de barras se pudo realizar de manera más ágil las búsquedas de las Órdenes de Movilización y de los Mantenimientos de los vehículos.

6. Symfony es un completo framework que gracias a sus características optimiza el desarrollo de aplicaciones web. Separa el desarrollo en tres capas (Modelo Vista Controlador) y su curva de aprendizaje es bastante corta.

7. PostgreSQL es un motor de base de datos de software libre su instalación es ilimitada, provee estabilidad y confiabilidad, es multiplataforma y está diseñado para ambientes de alto volumen.

5.2 RECOMENDACIONES

1. La Institución debe hacer uso constante de la aplicación entregada ya que el uso de la misma sirve para un mejor desempeño de las funciones inherentes al control vehicular.

2. Los funcionarios inmersos en el manejo del sistema deben realizar un uso adecuado del mismo.

3. Se debe realizar una constante alimentación de información a la base de datos del aplicativo para obtener un correcto funcionamiento del mismo.

4. Realizar constantes mejoras y actualizaciones a los diversos módulos del Aplicativo Web.

5. Es recomendable el mantenimiento, actualización de software y hardware en los computadores y en algunos casos el cambio de los mismos, pertenecientes a los diferentes departamentos de la Institución para evitar problemas en la utilización de Aplicativos instalados a nivel de red y de terminal.

REFERENCIAS BIBLIOGRÁFICAS

LIBROS:

Fabien Potencier . (2010). Practical symfony Create professional web applications with PHP and symfony 1.3 & 1.4, Doctrine 1.2. : Sensio SA.

The PostgreSQL Global Development Group . (2011). PostgreSQL 9.0 Official Documentation Volumen I The SQL Langu. : Fultus Corporation.

PUBLICACIONES EN LÍNEA

The Apache Software Foundation. (2012). Apache HTTP Server. Recuperado de:http://httpd.apache.org.

Wikipedia. (2012). Código abierto. Recuperado de: http://es.wikipedia.org/wiki/C%C3%B3digo_abierto.

Wikipedia. (2012). Hypertext Transfer Protocol. Recuperado de: http://es.wikipedia.org/wiki/Hypertext_Transfer_Protocol.

Camara. A. (2011). Apache: El servidor de Internet. Recuperado de: http://queessoftwarelibre.blogspot.com/2011/03/apache-el-servidor-de-internet.html.

Aycarg. (2011). Apache. Recuperado de: http://informaticamascomputacion.blogspot.com/2012/07/apache.html.

Wikipedia. (2012). Robert McCool. Recuperado de: http://es.wikipedia.org/wiki/Robert_McCool.

Wikipedia. (2012). National Center for Supercomputing Applications. Recuperado de: http://es.wikipedia.org/wiki/National_Center_for_Supercomputing_Applications.

Wikipedia. (2012). Webmaster. Recuperado de: http://es.wikipedia.org/wiki/Webmaster.

Wikipedia. (2012). Brian Behlendorf. Recuperado de: http://es.wikipedia.org/wiki/Brian_Behlendorf.

Softpedia. (2012). Descripción de Apache HTTP Server. Recuperado de:http://www.softpedia.es/programa-Apache-HTTP-Server-for-Windows-12027.html.

Wikipedia. (2012). Unix. Recuperado de: http://es.wikipedia.org/wiki/Unix.

Wikipedia. (2012). Windows NT. Recuperado de: http://es.wikipedia.org/wiki/Windows_NT.

The Apache Software Foundation. (2012). The Apache Tomcat Connector. Recuperado de: http://tomcat.apache.org/connectors-doc/.

SensioLabs. (2012). Open-Source PHP Web Framework. Recuperado de: http://www.symfony-project.org/.

Fabien Potencier. (2012). Fabien Potencier. Recuperado de: http://fabien.potencier.org/about.

Wikipedia. (2012). Framework. Recuperado de: http://es.wikipedia.org/wiki/Framework.

Wikipedia. (2012). Director ejecutivo. Recuperado de: http://es.wikipedia.org/wiki/Director_ejecutivo.

Wikipedia. (2012). Propel (PHP). Recuperado de: http://es.wikipedia.org/wiki/Propel_%28PHP%29.

37signals. (2012). Ruby on Rails. Recuperado de: http://rubyonrails.org/ .

Wikipedia. (2012). Mapeo objeto-relacional. Recuperado de: http://es.wikipedia.org/wiki/Mapeo_objeto-relacional.

Wikipedia. (2012). Oracle Database. Recuperado de: http://es.wikipedia.org/wiki/Oracle_Database.

Wikipedia. (2012). MySQL. Recuperado de: http://es.wikipedia.org/wiki/MySQL.

Wikipedia. (2012). Sistema de Gestión de Base de Datos. Recuperado de: http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_bases_de_datos.

phpDoc. (2012). phpDocumentor. Recuperado de: http://www.phpdoc.org/.

Wikipedia. (2012). Internacionalización y localización. Recuperado de: http://es.wikipedia.org/wiki/Internacionalizaci%C3%B3n_y_localizaci%C3%B3n.

Wikipedia. (2012). Localizador uniforme de recursos. Recuperado de: http://es.wikipedia.org/wiki/Localizador_uniforme_de_recursos.

Wikipedia. (2012). Interfaz de programación de aplicaciones. Recuperado de: http://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones.

Wikipedia. (2012). JavaScript. Recuperado de: http://es.wikipedia.org/wiki/JavaScript.

Wikipedia. (2012). AJAX. Recuperado de: http://es.wikipedia.org/wiki/AJAX.

Wikipedia. (2012). Internet Relay Chat. Recuperado de: http://es.wikipedia.org/wiki/Internet_Relay_Chat.

PostgreSQL Global Development Group. (2012). PostgreSQL. Recuperado de: http://www.postgresql.org/.

Wikipedia. (2012). ACID. Recuperado de: http://es.wikipedia.org/wiki/ACID.

Wikipedia. (2012). Michael Stonebraker. Recuperado de: http://es.wikipedia.org/wiki/Michael_Stonebraker.

Wikipedia. (2012). Berkeley Software Distribution. Recuperado de: http://es.wikipedia.org/wiki/Berkeley_Software_Distribution.

Wikipedia. (2012). Commit. Recuperado de: http://es.wikipedia.org/wiki/Commit.

Wikipedia. (2012). Class Inter-Domain Routing. Recuperado de: http://es.wikipedia.org/wiki/Classless_Inter-Domain_Routing.

Wikipedia. (2012). Dirección MAC. Recuperado de: http://es.wikipedia.org/wiki/Direcci%C3%B3n_MAC.

The PHP Group. (2012). PHP: Hypertext Preprocessor. Recuperado de: http://php.net/.

Wikipedia. (2012). Rasmus Lerdorf. Recuperado de: http://es.wikipedia.org/wiki/Rasmus_Lerdorf.

Tutty Morán. (2011). PHP Algunas definiciones. Recuperado de: http://www.tuttymoran.com/publico/definiciones.php?cid=006.

ProgramacionWeb.net. (2012). Introducción al PHP . Recuperado de: http://www.programacionweb.net/articulos/articulo/?num=182.

Molina, Juan Carlos. (2012). MVC. Recuperado de: http://es.scribd.com/doc/92266623/mvc.

WikiLibros. (2012). Implementación Modelo Vista Controlador (MVC). Recuperado de: http://es.wikibooks.org/wiki/Aplicaciones_Distribuidas:_Un_enfoque_pr%C3%A1ctico#Implementaci.C3.B3n_Modelo_Vista_Controlador_.28MVC.29.