

**UNIVERSIDAD POLITECNICA SALESIANA
SEDE CUENCA**

CARRERA DE INGENIERIA DE SISTEMAS

Tesis previa a la obtención del Título de:
Ingeniero de Sistemas

TEMA:

*“ANALISIS, DISEÑO E IMPLEMENTACION DE UN SISTEMA DE
INFORMACION WEB PARA EL CONTROL Y GESTIÓN DE RECURSOS PARA LA
DIRECCIÓN REGIONAL 2 DE LA CONTRALORÍA GENERAL DEL ESTADO”*

AUTOR:

Oscar Steve Castro Arteaga

DIRECTORA:

Ing. Bertha Tacuri

CUENCA – ECUADOR

2013

DECLARATORIA DE RESPONSABILIDAD

Los conceptos desarrollados, analizados, realizados y las conclusiones son de exclusiva responsabilidad del autor.

Cuenca, 30 de enero de 2013



Oscar Steve Castro Arteaga

CERTIFICACIÓN

Yo, Bertha Katerine Tacuri Capelo, Docente de la carrera de Ingeniería de Sistemas de la Universidad Politécnica Salesiana.

CERTIFICO

Que he leído íntegramente el presente trabajo de grado, comprobando que se cumplen los objetivos planteados en el proyecto. Así mismo declaro que el autor Oscar Steve Castro Arteaga es el único responsable de la originalidad del trabajo desarrollado.



Ing. Bertha Tacuri C.

AUTORIZACION

El autor del trabajo de tesis “ANALISIS, DISEÑO E IMPLEMENTACION DE UN SISTEMA DE INFORMACION WEB PARA EL CONTROL Y GESTIÓN DE RECURSOS PARA LA DIRECCIÓN REGIONAL 2 DE LA CONTRALORÍA GENERAL DEL ESTADO” Oscar Steve Castro Arteaga, autoriza a la Universidad Politécnica Salesiana el uso de la misma para fines académicos.



Oscar Steve Castro Arteaga

DEDICATORIA

Este proyecto de tesis va dedicado en primer lugar a Dios por haberme dado la oportunidad de crecer como persona y permitirme cumplir la más grande de mis metas.

A mis padres y a mi hermana Michelle, ya que sin su apoyo no hubiese podido alcanzar tan anhelado triunfo.

A mi abuelita Michi que desde la gloria del Señor siempre me acompañará y desde lo alto del cielo me llenará de bendiciones.

A toda mi familia y a cada una de las personas que siguieron día tras día el desarrollo del proyecto brindándome todo su apoyo incondicional y fueron un pilar y una fuente de energía en los momentos más difíciles.

Y para aquellas personas que hoy ya no se encuentran a nuestro lado pero que en su momento me brindaron su cariño, amistad y todo su apoyo.

Oscar Steve Castro Arteaga.

AGRADECIMIENTO

Agradezco a Dios por haberme dado fuerzas en los momentos más difíciles y la sabiduría para tomar las decisiones correctas.

A mis padres, a mi hermana por su cariño e incondicional apoyo y dedicación.

A toda mi familia que me apoyaron y me dieron fuerzas para seguir adelante.

A todos mis profesores, que fueron los pilares fundamentales en mi desarrollo académico y que gracias a sus exigencias me ayudaron a crecer profesionalmente y sobre todo me ayudaron a ser una mejor persona.

A mis amigos y amigas que siempre estuvieron a mi lado brindándome su ayuda, apoyo y confianza.

Oscar Steve Castro Arteaga.

CONTENIDO

INDICE DE IMÁGENES	11
INDICE DE EJEMPLOS	13
INDICE DE TABLAS	15
DIAGRAMAS DE NAVEGACIÓN	16
INTRODUCCIÓN	17
1. CAPÍTULO I - SITUACION ACTUAL	20
1.1. HISTORIA DE LA CONTRALORÍA GENERAL DEL ESTADO	20
1.1. MISIÓN	21
1.2. VISIÓN	21
2. CAPÍTULO II - MARCO TEORICO.....	23
2.1. ¿QUÉ SON LAS APLICACIONES WEB?	23
2.1.1. EL MODELO DE TRES CAPAS.....	25
2.1.2. COMPARACIÓN DE LAS APLICACIONES WEB Y LAS APLICACIONES DE ESCRITORIO.....	26
2.2. ¿QUÉ ES JEE?	27
2.2.1. EL MODELO DE APLICACIÓN JAVA EE	29
2.2.2. COMPONENTES JEE.....	29
2.2.3. JAVA EE APIS	30
2.3. JAVA PERSISTENCE API	32
2.3.1. QUÉ ES JAVA PERSISTENCE API	32
2.3.2. ARQUITECTURA JPA	34
2.3.3. CREACION DE UNA ENTIDAD.....	34
2.3.4. MANEJO DE RELACIONES EN JPA	38
2.3.5. LENGUAJE DE CONSULTA JPQL (Java Persistence Query Lenguaje).....	42

2.4.	JAVA JSF FRAMEWORK.....	49
2.4.1.	¿QUE ES JSF?	49
2.4.2.	CARACTERÍSTICAS DE JSF	49
2.4.3.	JSF Y EL MODELO MVC.....	50
2.4.4.	COMPONENTES DE UNA APLICACION JSF	50
2.4.5.	MANEJO DE CONTROLADORES BEAN	51
2.4.6.	PRINCIPALES ETIQUETAS JSF	52
2.4.7.	CONVERTIDORES Y VALIDADORES	53
2.5.	FRAMEWORKS JSF	60
2.5.1.	COMPARACION DE FRAMEWORKS.....	60
2.5.2.	INSTALACIÓN DE PRIMEFACES.....	64
2.6.	SERVIDORES WEB	68
2.6.1.	¿QUÉ ES UN SERVIDOR WEB?.....	68
2.6.2.	INSTALACION APACHE TOMCAT SERVER.....	68
2.7.	BASES DE DATOS	75
2.7.1.	¿QUE ES UNA BASE DE DATOS?.....	75
2.7.2.	INSTALACIÓN DE POSTGRESQL	76
2.8.	IREPORT	83
2.8.1.	¿QUE ES IREPORT?.....	83
2.8.2.	CARACTERÍSTICAS DE IREPORT	83
2.8.3.	INSTALACION IREPORT PLUGIN PARA NETBEANS.....	83
3.	CAPÍTULO III – ALCANCE DEL PROYECTO Y FASE DE ANALISIS DE REQUERIMIENTOS.....	94
3.1.	ALCANCE DEL PROYECTO	94
3.2.	ROLES DE USUARIO	106
3.3.	FASE DE ANÁLISIS DE REQUERIMIENTOS	106
3.3.1.	ANÁLISIS DE REQUERIMIENTOS DEL USUARIO	106

3.3.2.	ANÁLISIS DE REQUERIMIENTOS NO FUNCIONALES	140
3.4.	REQUERIMIENTOS DE HARDWARE	141
3.5.	REQUERIMIENTOS DE SOFTWARE	141
3.5.1.	SERVIDOR DE APLICACIÓN	141
3.5.2.	SERVIDOR DE BASES DE DATOS	142
3.5.3.	ENTORNO DE DESARROLLO.....	142
4.	CAPÍTULO IV – ANALISIS DE RIESGOS	143
4.1.	ANÁLISIS DE RIESGOS.....	143
4.1.1.	RIESGOS DEL PRODUCTO.....	144
4.1.2.	RIESGOS DE NEGOCIO.....	145
4.1.3.	RIESGOS DEL SISTEMA	146
5.	CAPÍTULO V - FASE DE DISEÑO.....	148
5.1.	IDENTIFICACIÓN DE ACTORES	148
5.2.	IDENTIFICACIÓN DE CASOS DE USO	149
5.2.1.	MÓDULO USUARIOS	149
5.2.2.	MÓDULO EMPLEADOS	160
5.2.3.	MÓDULO PRODUCTOS	168
5.2.4.	MÓDULO CURSOS.....	178
5.2.5.	MÓDULO VEHÍCULOS	187
5.2.6.	MÓDULO CONTROL MEDICO	198
5.3.	DIAGRAMAS DE SECUENCIA	205
5.4.	DIAGRAMAS DE PROCESOS	211
5.4.1.	MÓDULO USUARIOS	211
5.4.2.	MÓDULO EMPLEADOS	212
5.4.3.	MÓDULO PRODUCTOS	214
5.4.4.	MÓDULO CURSOS.....	215
5.4.5.	MÓDULO VEHÍCULOS	217

5.4.6.	MÓDULO CONTROL MEDICO	219
5.5.	MODELO CONCEPTUAL.....	220
5.6.	DIAGRAMA DE CLASES.....	221
5.7.	MODELO DE COMPONENTES	222
5.8.	DIAGRAMA DE NAVEGACIÓN	223
5.9.	DICCIONARIO DE DATOS.....	236
5.9.1.	MÓDULO USUARIOS	236
5.9.2.	MÓDULO EMPLEADOS	237
5.9.3.	MÓDULO PRODUCTOS	239
5.9.4.	MÓDULO CURSOS.....	241
5.9.5.	MÓDULO VEHICULOS	243
5.9.6.	MÓDULO CONTROL MÉDICO	245
5.10.	IDENTIFICACIÓN DE ROLES Y PRIVILEGIOS.....	248
5.11.	INTERFACES DE USUARIO	252
5.11.1.	AUTENTICACIÓN DEL SISTEMA	252
5.11.2.	PANTALLA PRINCIPAL.....	252
5.11.3.	MENUS DEL SISTEMA.....	252
5.11.4.	GESTIÓN DE ROLES.....	253
5.11.5.	GESTIÓN DE USUARIOS	254
5.11.6.	GESTIÓN DE EMPLEADOS	255
5.11.7.	GESTIÓN DE PRODUCTOS.....	255
5.11.8.	GESTIÓN DE ALMACÉN PRODUCTOS.....	256
5.11.9.	CONTROL MÉDICO	258
5.11.10.	GESTIÓN DE CURSO	260
5.11.11.	GESTIÓN MANTENIMIENTOS VEHÍCULO	262
5.11.12.	GESTIÓN VEHÍCULOS	262
5.11.13.	GESTIÓN HISTORIAL MANTENIMIENTOS VEHÍCULO	263

5.11.14.	GESTIÓN SOLICITUD VEHÍCULO	264
5.11.15.	GESTIÓN ORDENES ED MOVILIZACIÓN.....	264
6.	CAPÍTULO VI – FASE DE DESARROLLO	266
6.1.	ARCHIVOS DE CONFIGURACION	266
6.1.1.	FACES-CONFIG.XML	266
6.1.2.	PERSISTENCE.XML.....	268
6.1.3.	WEB.XML.....	269
7.	CAPÍTULO VII – PRUEBAS DEL SISTEMA	272
7.1.	PRUEBAS DEL SISTEMA	272
7.1.1.	PRUEBAS MÓDULO USUARIO	272
7.1.2.	PRUEBAS MÓDULO EMPLEADOS	276
7.1.3.	PRUEBAS MÓDULO PRODUCTOS	281
7.1.4.	PRUEBAS MÓDULO CURSOS.....	288
7.1.5.	PRUEBAS MÓDULO VEHÍCULOS	290
7.1.6.	PRUEBAS MÓDULO CONTROL MEDICO	293
	CONCLUSIONES	296
	RECOMENDACIONES.....	297
	ANEXOS	298
	BIBLIOGRAFIA	299

INDICE DE IMÁGENES

Imagen 1: Evolución Web.....	17
Imagen 2: Páginas web estáticas, están enfocadas principalmente a mostrar una información permanente.....	23
Imagen 3: Funcionamiento web básico.....	24
Imagen 4: Ejemplo de páginas web actuales.....	25
Imagen 5: Modelo de tres capas.....	26
Imagen 6: Esquema General de JEE.....	29
Imagen 7: Componentes de aplicación JEE.....	29
Imagen 8: Java EE APIs.....	31
Imagen 9: Componentes principales de JPA.....	34
Imagen 10: Modelo MVC de JSF.....	50
Imagen 11: Componentes principales de una aplicación JSF.....	51
Imagen 12: Popularidad entre frameworks JSF.....	61
Imagen 13: Ranking Web Frameworks.....	61
Imagen 14: Definición del namespace de Primefaces en una página web.....	67
Imagen 15: Interfaz de autenticación.....	252
Imagen 16: Pantalla principal del sistema.....	252
Imagen 17: Menú del sistema.....	253
Imagen 18: Interfaz para la gestión de roles.....	253
Imagen 19: Formulario para el registro de roles.....	253
Imagen 20: Formulario para asignar acceso y permisos a módulos.....	254
Imagen 21: Interfaz para la gestión de usuarios.....	254
Imagen 22: Formulario para el registro de usuarios.....	254
Imagen 23: Interfaz para la gestión de empleados.....	255
Imagen 24: Formulario para el registro de empleados.....	255
Imagen 25: Interfaz para la gestión de productos.....	255
Imagen 26: Formulario para el registro de productos.....	256
Imagen 27: Interfaz para la gestión almacén productos.....	256
Imagen 28: Formulario para el registro de un producto en un almacén específico.....	256
Imagen 29: Interfaz para el ingreso de stock de productos.....	257
Imagen 30: Formulario para el registro de stock de productos.....	257
Imagen 31: Interfaz para la consulta de kardex.....	257

Imagen 32: Graficas de los movimientos del producto.....	258
Imagen 33: Tabla de movimientos del producto.....	258
Imagen 34: Formulario para el registro del historial clínico.....	258
Imagen 35: Formulario para registro de consultas médicas (parte 1).....	259
Imagen 36: Formulario para registro de consultas médicas (parte 2).....	259
Imagen 37: Formulario para registro de consultas médicas (parte 3.1).....	259
Imagen 38: Formulario para registro de consultas médicas (parte 3.2).....	260
Imagen 39: Formulario para registro de consultas médicas (parte 4).....	260
Imagen 40: Interfaz para la gestión de cursos.....	260
Imagen 41: Formulario para el registro de cursos.....	261
Imagen 42: Interfaz para la gestión de alumnos.....	261
Imagen 43: Formulario para el registro de alumnos.....	261
Imagen 44: Interfaz para la gestión de mantenimientos de vehículo.....	262
Imagen 45: Formulario para el registro de mantenimientos de vehiculo.....	262
Imagen 46: Interfaz para la gestión de vehículos.....	262
Imagen 47: Formulario para el registro de vehículos.....	262
Imagen 48: Interfaz para la gestión de historia de mantenimientos de vehículo.....	263
Imagen 49: Formulario para el registro de mantenimientos de vehículo (parte 1).....	263
Imagen 50: Formulario para el registro de mantenimientos de vehículo (parte 2).....	263
Imagen 51: Interfaz para la gestión de solicitudes de vehículos.....	264
Imagen 52: Formulario para el registro de solicitudes vehículo.....	264
Imagen 53: Interfaz para la gestión de órdenes de movilización.....	264
Imagen 54: Formulario para el registro de orden de movilización.....	265
Imagen 55: Interfaz para la gestión de las órdenes de movilización.....	265
Imagen 56: Formulario para terminación o cancelación de orden de movilización.....	265
Imagen 57: Ubicación de archivos de configuración en Netbeans.....	266

INDICE DE EJEMPLOS

Ejemplo 1: Ejemplo de anotación	28
Ejemplo 2: Clase PERSONA transformada a una entidad.....	36
Ejemplo 3: Anotación @Entity.....	36
Ejemplo 4: Declaración de atributos y métodos en una entidad	37
Ejemplo 5: Declaración de un constructor de una entidad.....	37
Ejemplo 6: Mapeo tabla PERSONAS / entidad PERSONA.....	38
Ejemplo 7: Declaración de nombre de una entidad por defecto.	38
Ejemplo 8: Ejemplo relación uno a uno	39
Ejemplo 9: Declaración de relación uno a uno.	40
Ejemplo 10: Relación de uno a muchos.....	40
Ejemplo 11: Declaración relación uno a muchos.....	40
Ejemplo 12: Declaración relación muchos a uno.....	41
Ejemplo 13: Relación de muchos a muchos.	41
Ejemplo 14: Declaración relación muchos a muchos.	42
Ejemplo 15: Sintaxis consulta en JPQL.....	43
Ejemplo 16: Ejemplo de consulta en JPQL.....	43
Ejemplo 17: Consulta con alias de una tabla en JPQL.....	43
Ejemplo 18: Visualización de atributos en una entidad.....	43
Ejemplo 19: Consulta de atributos a varios niveles.	44
Ejemplo 20: Funciones agregadas en JPQL.....	44
Ejemplo 21: Consulta condicional en JPQL.	44
Ejemplo 22: Consulta de varias condiciones en JPQL.	45
Ejemplo 23: Consulta condicional con rangos en JPQL.....	45
Ejemplo 24: Consulta con operadores para cadenas de texto en JPQL.	45
Ejemplo 25: Declaración de parámetros por posición.	46
Ejemplo 26: Declaración de parámetros por nombre.....	46
Ejemplo 27: Declaración de una consulta dentro de una clase.....	47
Ejemplo 28: Consulta con resultado simple en JPQL.....	47
Ejemplo 29: Consulta con múltiples resultados en JPQL.....	47
Ejemplo 30: Declaración de una consulta utilizando parámetros por posición.	47
Ejemplo 31: Declaración de una consulta utilizando parámetros por nombre.....	48
Ejemplo 32: Declaración de una consulta con nombre en una entidad.....	48

Ejemplo 33: Utilización de una consulta con nombre.	48
Ejemplo 34: Declaración de una consulta nativa.	49
Ejemplo 35: Definición de un controlador bean dentro del archivo faces-config.xml.	51
Ejemplo 36: Declaración de un controlador bean en una clase.	52
Ejemplo 37: Utilización de un convertidor estándar JSF.....	54
Ejemplo 38: Definición de un convertidor personalizado en JSF.....	57
Ejemplo 39: Declaración de un convertidor personalizado en archivo faces- config.xml.	57
Ejemplo 40: Utilización de un convertidor personalizado en JSF.....	57
Ejemplo 41: Definición de un validador personalizado en JSF.	59
Ejemplo 42: Declaración de un validador personalizado en archivo faces-config.xml.	59
Ejemplo 43: Utilización de un validador personalizado en JSF.	60
Ejemplo 44 : Implementación de componentes Primefaces en una pagina web.....	68

INDICE DE TABLAS

Tabla 1: Comparación entre aplicaciones de escritorio y web.....	27
Tabla 2: Lista de anotaciones JPA por categorías.....	35
Tabla 3: Palabras reservadas de JPQL.....	44
Tabla 4: Principales etiquetas de JSF.....	53
Tabla 5: Convertidores JSF estándar.....	54
Tabla 6: Atributos para convertidores numéricos.....	55
Tabla 7: Atributos para convertidores de fecha.....	55
Tabla 8: Validadores estándar de JSF.....	58
Tabla 9: Matriz de análisis de probabilidad e impacto.....	143
Tabla 10: Resultados caja blanca creación nombre usuario.....	274
Tabla 11: Resultados pruebas caja negra registro usuario.....	276
Tabla 12: Resultados prueba caja blanca validación cédula.....	278
Tabla 13: Resultados caja negra registro empleado.....	281
Tabla 14: Resultados prueba caja blanca kardex producto.....	286
Tabla 15: Resultados caja negra registró stock producto.....	287
Tabla 16: Resultados pruebas caja negra registro curso.....	290
Tabla 17: Resultados pruebas caja blanca validación placa vehículos.....	291
Tabla 18: Resultados pruebas caja negra registro vehículos.....	293
Tabla 19: Resultados pruebas caja negra registro vehículos.....	295

DIAGRAMAS DE NAVEGACIÓN

Diagrama Navegación 1: Gestión roles.....	223
Diagrama Navegación 2: Gestión usuarios.....	224
Diagrama Navegación 3: Gestión departamentos.....	224
Diagrama Navegación 4: Gestión cargos.....	225
Diagrama Navegación 5: Gestión empleados.....	225
Diagrama Navegación 6: Gestión cargas familiares.....	226
Diagrama Navegación 7: Gestión historial cargos.....	226
Diagrama Navegación 8: Gestión unidades.....	227
Diagrama Navegación 9: Gestión categorías.....	227
Diagrama Navegación 10: Gestión Productos.....	228
Diagrama Navegación 11: Gestión almacén productos.....	228
Diagrama Navegación 12: Gestión stock almacén producto.....	229
Diagrama Navegación 13: Detalles movimientos producto.....	229
Diagrama Navegación 14: Gestión solicitudes productos.....	230
Diagrama Navegación 15: Gestión medicamentos.....	230
Diagrama Navegación 16: Gestión tipos de alergias.....	231
Diagrama Navegación 17: Gestión tipos enfermedades.....	231
Diagrama Navegación 18: Gestión historiales clínicos.....	232
Diagrama Navegación 19: Gestión consultas médicas.....	232
Diagrama Navegación 20: Gestión mantenimientos vehículos.....	233
Diagrama Navegación 21: Gestión vehículos.....	233
Diagrama Navegación 22: Gestión historial mantenimientos.....	234
Diagrama Navegación 23: Gestión de solicitudes vehículo y registro de ordenes de movilización.....	234
Diagrama Navegación 24: Gestión ordenes movilización.....	235
Diagrama Navegación 25: Gestión cursos.....	235

INTRODUCCIÓN

La evolución de la Web y el nacimiento de nuevas tecnologías han permitido generar sistemas y servicios basados en web más dinámicos y que permiten una mayor interacción con el usuario.

Al principio se contaba con páginas web estáticas, a esta se la conoce como la primera generación web; luego aparecen portales con contenido dinámica, estos ya permitían al usuario interactuar con el sistema por medio de formularios, es aquí donde los usuarios comienzan a interactuar de forma colaborativa con otros usuarios, de esta manera se establece una red de conocimiento; la tercera generación permite acceder a grandes bases de datos por medio de aplicaciones basadas en Web y para ello es necesario varias tecnologías (servidores web, navegadores web, servidores de aplicación, etc.) para que el usuario pueda acceder a la aplicación, a este modelo se lo conoce como modelo cliente servidor; un usuario solicita el servicio, el servidor responde a la petición mediante un programa que se encuentra ejecutándose dentro de este, dichos programas sirven como un medio de comunicación entre el usuario y el servidor a más de permitir el acceso a los servicios.

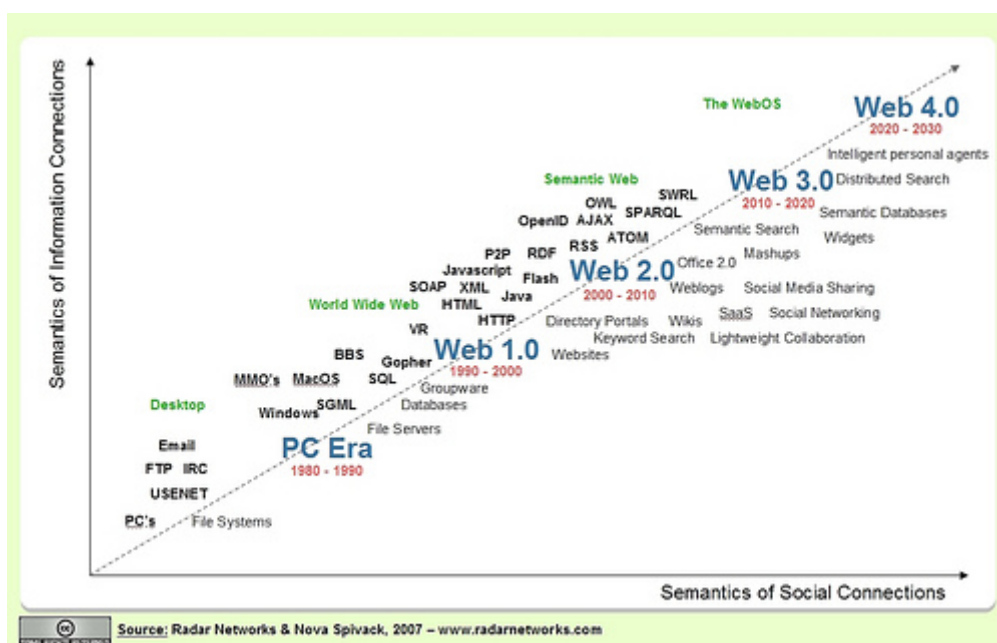


Imagen 1: Evolución Web.¹

¹ Tomado de la página web: moduloweb.wikidot.com/evolucion-de-la-web

El objetivo principal de este tema, es el de desarrollar una aplicación Web con la ayuda de tecnologías nuevas y que permitan a los usuarios un mejor manejo de los datos, reduciendo posibles errores al momento de manipular la información, y de esta manera disminuir el tiempo de búsqueda de información para la toma de decisiones.

Las aplicaciones Web utilizan un modelo o patrón de desarrollo denominado MVC (Modelo Vista Controlador), el cual no es otra cosa que separar por capas la aplicación: la primera (vista) se encargara de mostrara la información y la que estar a en contacto directo con el usuario como páginas web y formularios; la segunda capa (controlador) será la encargada de la parte lógica del sistema, validaciones, operaciones, etc. y una tercera (modelo) que se encargara de la gestión de la base de datos; aplicar este patrón permitirá un mayor control y facilitará el mantenimiento de la aplicación.

Generar sistemas basados en web tiene grandes ventajas: entre una de ellas mostrar al usuario datos en tiempo real y en cualquier lugar, permite un mejor mantenimiento de la aplicación, no requieren de grandes prestaciones de hardware y son independientes de los sistemas operativos, entre otras.

Hablemos un poco de las formas de almacenar información; en muchas empresas o instituciones, es muy común que la información se las guarde en hojas de cálculo, esta forma de almacenamiento tienes algunos inconvenientes en cuanto a seguridad, pues son archivos fácilmente manipulables y de fácil acceso, implementa un sistema que integre el almacenamiento de información en una base de datos es mucho más seguro, pues solo personas con suficiente privilegios o a los cuales se les de permisos para su manipulación podrán acceder a la información almacenada, esta es una gran ventaja frente a las hojas de cálculo o cualquier otro tipo de archivo.

Gracias a algunas librerías o herramientas existentes es posible importar la información almacenada en las bases de datos a hojas de cálculo para ser transportada o manipulada, para ello podemos utilizar IReport, que es una

herramienta para el diseño de informes en forma visual, es poderoso e intuitivo y fácil de usar.

Diseñar la interfaz grafica parece ser la parte más tedioso y difícil del desarrollo, pues este debe de ser muy amigable con el usuario, deberá seguir un patrón y sobre todo tener uniformidad, el manejo de colores es muy importante, pues este puede influenciar en la percepción y conducta de los usuarios. Es por ellos que se deberá utilizar colores armónicos y de bajo impacto visual.

Es necesario también contar con librerías de componentes para las interfaces que faciliten el desarrollo de las páginas, existen varias opciones en el mercado como IceFaces, Primefaces, Richfaces, entre otros. Cada uno de ellos proporcionan varios componentes visuales para la construcción de páginas web, pero hay que tener en cuenta sobre cuál de todas se acopla a nuestras necesidades, si son de fácil implementación, si son gratuitas y si son compatibles con otras tecnología.

En base a estas características, será de utilidad la búsqueda de herramientas y tecnologías que permitan un rápido desarrollo y a la vez que sean robustas y estables, también será de vital importancia el desarrollo de una interfaz de usuario que permita la interacción de forma amigable con el sistema y que permitan a usuarios no expertos un rápido manejo y familiarización con la aplicación.

A continuación se presentará la situación actual de la institución y los problemas que se pretenden eliminar; se analizarán los requerimientos por parte de los usuarios y el alcance del mismo, luego se procederá a un análisis de riesgos del proyecto, su diseño y desarrollo y concluiremos con las pruebas del sistema.

También se describirán las principales características y funcionalidades de las diferentes tecnologías involucradas en el desarrollo del proyecto.

1. CAPÍTULO I - SITUACION ACTUAL

1.1. HISTORIA DE LA CONTRALORÍA GENERAL DEL ESTADO

El 2 de diciembre de 1927, como consecuencia del asesoramiento prestado al país, por la Misión Kemmerer, entró en vigencia la Ley Orgánica de Hacienda, publicada en el Registro Oficial N° 448 de 16 de noviembre de 1927, que crea la Contraloría General de la Nación, como departamento independiente del gobierno, con el objeto de realizar el control fiscal, compilar cuentas y constituir la oficina central de la Contabilidad del Estado.

Con el Decreto Ley 1065-A se reforma a la ley Orgánica de Hacienda de 1927; tales enmiendas son codificadas en 1960, estableciendo que la Contraloría debe tener un sistema moderno de control (Sistema Integrado de Administración Financiera y Control) con el fin de precautelar la economía, efectiva y eficiente administración de los recursos humanos, materiales y financieros, para controlar oportunamente los fondos fiscales y satisfacer las necesidades del servicio público y del desarrollo del país.

En la constitución de 1967 se le asignaron, además, funciones de fiscalización y se cambió la denominación de "Contraloría General de la Nación" por la de Contraloría General del Estado.

En 1977 se expide la ley Orgánica de Administración Financiera y Control (LOAFYC) publicada en el Registro Oficial 337, de 16 de mayo de ese año, con la cual se sustituyó a la Ley Orgánica de Hacienda Codificada. La LOAFYC, contiene las normas fundamentales que rigen la estructura y funcionamiento de la Contraloría General del Estado; en ella se plasmó el cambio de esta Institución dejando atrás el sistema del simple Control Fiscal para instaurar el sistema de Control Gubernamental Moderno de los recursos públicos.

Al cambiar la estructura legal y administrativa de la Contraloría, dejó de ser la oficina de Contabilidad e Intervención Fiscal, se constituyó como un Organismo

Superior de Control de los recursos de las entidades del sector público. Este control lo efectuará mediante un examen posterior a las operaciones financieras y administrativas de cada entidad a través de exámenes especiales, auditorías financieras y auditorías operacionales.²

1.1. MISIÓN

“Somos el Organismo Técnico encargado del control de la utilización de los recursos estatales y la consecución de los objetivos de las instituciones del Estado y de las demás personas jurídicas de derecho privado que dispongan de recursos públicos.”³

1.2. VISIÓN

“En el 2012 la Contraloría General del Estado será: Un organismo modelo de gestión pública, moderno y confiable, fundamentado en principios de ética, transparencia, calidad y enfoque hacia los resultados, que garanticen a la ciudadanía ecuatoriana el eficiente control de los recursos públicos”.⁴

1.3. SITUACIÓN ACTUAL

En la Dirección Regional 2, de la Contraloría General del Estado es complejo obtener información de inventarios de manera inmediata, oportuna y actualizada para tomar decisiones, pues la misma es llevada de manera inadecuada (forma manual y hojas de cálculo) ya que no se cuenta con un sistema computarizado que almacena los datos; lo que hace posible que esta se convierta en un medio manipulable dentro de la empresa y aun peor pueda extraviarse.

Al carecer de un sistema de información, desembocan varios inconvenientes, entre ellos, no se dispone de un proceso automático sobre el inventario de suministros de oficina o el movimiento de los productos; así como registrar el mantenimiento de los

² Tomado de la página web: http://www.contraloria.gob.ec/la_institucion.asp?id_SubSeccion=1

^{3,4} Tomado de la página web: http://www.contraloria.gob.ec/la_institucion.asp?id_SubSeccion=2

vehículos y su detalle se ha vuelto un proceso inadecuado al tener que guardar facturas de manera física y los datos en hojas de cálculo, con formatos no adecuados y cada vez adaptados a las necesidades de los mismos; de igual manera la asignación de un vehículo para la movilización de los funcionarios es algo tedioso al tener que llenar un formulario físico y tener contrariedades al momento del traslado, ya que por falta de coordinación en la hora de salida y regreso o en obtener el recurso móvil a tiempo, lo que genera una demanda de tiempo y esfuerzo por parte el usuario innecesario que puede ser optimizado con un sistema.

Por otro lado las consultas médicas que se realizan a los empleados, de igual manera son registradas en hojas de cálculo y con respaldo de manera física en hojas impresas, lo que impide contar con un historial completo de enfermedades y alergias y disponer de un registro completo sobre los medicamentos recetados en cada sesión, generando inconvenientes al no contar con un conocimiento clínico de toda la entidad lo que impide llevar un reporte de enfermedades comunes y recurrentes, información que apoyará a la toma de decisiones a las autoridades de la institución.

Sin embargo, como en toda empresa se hace necesario el proceso de capacitación del personal, para mejorar su desarrollo y desempeño diario, el mismo que actualiza los conocimientos no solo del capacitando sino también de la entidad involucrada, estos datos son almacenados con la copia de las certificaciones de cada uno de los involucrados, este procedimiento no se considera apto ya que si se desea obtener un reporte de los cursos, mucha de la información podría ser omitida (calificaciones, lugar, duración, etc...)

Dentro de este marco de análisis se puede observar que la institución cuenta con un sistema de registro de hojas de cálculo que no es muy óptimo a la hora de tomar decisiones, pues estas se encuentran distribuidas en varios departamentos y no están consolidadas en una base de datos única, lo que puede provocar pérdida de información con el paso del tiempo y no es nada eficaz al momento de analizar la misma.

2. CAPÍTULO II - MARCO TEORICO

2.1. ¿QUÉ SON LAS APLICACIONES WEB?

Una aplicación web es un sistema que corre bajo el protocolo HTTP (protocolo de transferencia de hipertexto) el cual es ampliamente utilizado por la web. Este tipo de aplicaciones se encuentran alojadas en un servidor web y los usuarios las utilizan por medio un navegador.

A principio las páginas web a las que los usuarios accedían presentaban únicamente información estática y éstos no tenían ningún tipo de interacción con las mismas.



Imagen 2: Páginas web estáticas, están enfocadas principalmente a mostrar una información permanente.

Los sistemas basados en web tiene una arquitectura cliente/servidor, y como se puede ver en la Imagen 1, el usuario/cliente por medio del navegador envía un REQUEST (petición) por medio del protocolo HTTP a un servidor web, el cual se encargará de procesar la petición del usuario y a su vez enviará un RESPONSE (respuesta) al cliente, en este caso en forma de una página web.

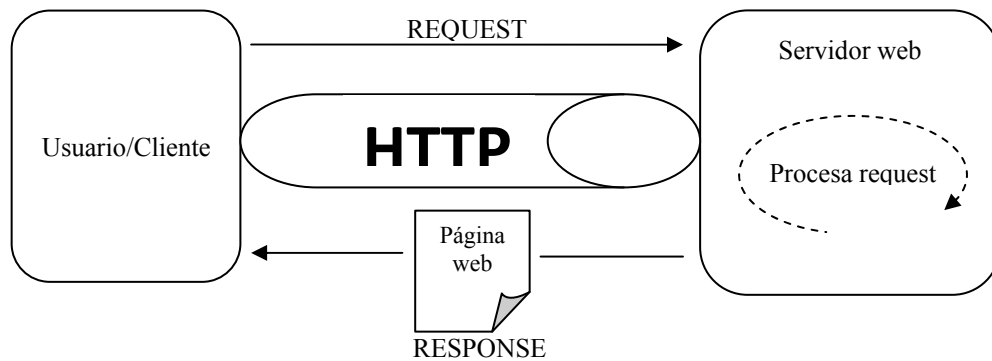


Imagen 3: Funcionamiento web básico.

Cabe mencionar que no necesariamente la respuesta del servidor debe de ser una página web, sino que puede ser de cualquier otro tipo o una salida de un programa; normalmente esto se consigue mediante la utilización de una interfaz de entrada común o CGI (Common Gateway Interface).

CGI especifica un estándar para la transferencia de datos entre el cliente y el programa y cuyo resultado final son objetos MIME (Multipurpose Internet Mail Extensions, en español Extensiones Multipropósito de Correo de Internet), que no son más que una serie de especificaciones que hace posible la transferencia de varios tipos de archivos a través de internet, estos pueden ser archivos de texto, audio, vídeo, etc.

Los CGI fueron los primeros pasos para el inicio de páginas web dinámicas.

Los navegadores de hoy y las tecnologías de desarrollo web han permitido desarrollar páginas web mucho más dinámicas y han conseguido que el usuario interactúe de forma más directa con dichas páginas, así aparecen las aplicaciones web enriquecidas o RIA (Rich Internet Applications), que son aplicaciones web que tiene la gran mayoría de las características de las aplicaciones de escritorio.

Para conseguir nuevas funcionalidades o efectos dinámicos, muchos desarrolladores utilizan lenguajes interpretados como AJAX, JavaScript, PHP, Perl, etc.

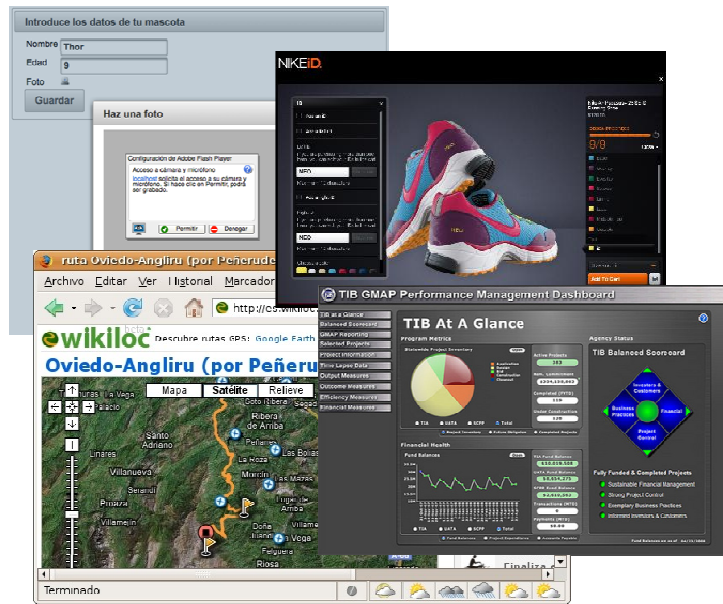


Imagen 4: Ejemplo de páginas web actuales.

Un lenguaje interpretado es aquel que no necesita ser compilado o traducido a lenguaje de máquina, sino que se ejecutan a medida que son necesarios y son analizados y ejecutados por un programa intérprete. A estos lenguajes se los conoce también como scripts.

2.1.1. EL MODELO DE TRES CAPAS

Hoy en día las aplicaciones web se basan en el modelo de tres capas, esto quiere decir, que su lógica de negocio está separada de su lógica de diseño. La ventaja de seguir este modelo es que si desea realizar algún cambio o modificación, se ataca únicamente el nivel requerido sin interferir con los demás.

En la imagen 4 se muestra el esquema de un modelo de tres capas, la idea principal del modelo es separar los componentes de la aplicación para un mejor manejo y control.



Imagen 5: Modelo de tres capas

La primera capa es la Capa de Presentación, la cual se encargará de presentar al usuario la información, capturar la información por medio de formularios, etc.; a esta capa se la conoce también como interfaz gráfica y debe ser entendible y fácil de usar. La siguiente capa es la Capa de Negocio, aquí es donde se encuentran las aplicaciones que se encargan de recibir, procesar y enviar la información al usuario, es aquí donde se establecen las diferentes reglas de la aplicación, y por último tenemos la Capa de Datos que es donde residen los datos y puede estar formado por un gestor de base de datos. En la actualidad este es el modelo más común de desarrollo web.

2.1.2. COMPARACIÓN DE LAS APLICACIONES WEB Y LAS APLICACIONES DE ESCRITORIO

APLICACIONES WEB	APLICACIONES ESCRITORIO
Instalación solo en el servidor.	Instalación por máquina.
Multiplataforma.	Dependientes del sistema operativo.
Se ejecutan en cualquier ordenador.	Ejecución de forma local.
Necesitan de conexión a internet.	No necesitan una conexión a internet.
Depende de la calidad y velocidad de la conexión.	Más robustas, estables y tiempo de respuesta más rápido.
Demandan de mayor seguridad.	Son más seguras.
Solo necesita de un navegador.	Requieren especificaciones técnicas.
Fácilmente actualizables.	Actualizaciones personalizadas.
No se requiere de mucho espacio en disco duro.	Demanda mayor capacidad de disco.
Funcionalidades limitadas.	Sistemas complejos.
Problemas de privacidad, se puede rastrear las actividades del usuario.	Mayor privacidad, se accede únicamente desde el computador local.

Mayor número de aplicaciones de código libre.	Se requiere de licencia por instalación.
---	--

Tabla 1: Comparación entre aplicaciones de escritorio y web.

2.2. ¿QUÉ ES JEE?

Actualmente es muy importante para un desarrollador contar con herramientas que faciliten el trabajo del desarrollo, que permitan reducir costos tanto monetarios como de recursos y que a su vez, el tiempo de desarrollo sea el menor posible, sin dejar de considerar que las aplicaciones a desarrollar deberán ser rápidas al ejecutar los procesos, ser seguras y ser confiables.

Tras el desarrollo de la plataforma de programación Java EE o JEE (Java Platform, Enterprise Edition, conocida anteriormente como J2EE hasta la versión 1.4) desarrollada por Sun Microsystems, ha permitido desarrollar aplicaciones Java empresariales de manera más fácil y rápida. El objetivo de JEE es proporcionar un conjunto de API's que ayuden a reducir el tiempo de desarrollo, la complejidad y mejoren el rendimiento de las aplicaciones, permitiendo que los desarrolladores se centren más en la lógica de negocio en lugar de las tareas de mantenimiento de bajo nivel.

JEE ofrece un framework para el desarrollo de aplicaciones distribuidas multicapa basadas en Web, definiendo una infraestructura común para el acceso a bases de datos, gestión de persistencias, control de seguridad, gestión de transacciones y una arquitectura modelo, vista, controlador que no es más que la separación entre la interfaz de usuario, la lógica de negocio y los datos.

Entre las características de Java EE es la introducción de un modelo de programación simplificado, permitiendo al desarrollador introducir información por medio de anotaciones dentro del código fuente, el servidor Java EE se encargará de configurar el componente al momento del despliegue y del tiempo de ejecución.

Las anotaciones son una alternativa a la tecnología XML para introducir metadatos dentro del código fuente; estas pueden añadirse a clases, métodos, campos, parámetros, variables locales, y paquetes.

Las anotaciones se diferencian del resto de código porque siempre vienen precedidas por una arroba (@), el siguiente es un ejemplo de anotación:

```
@Override  
public String toString() {...}
```

Ejemplo 1: Ejemplo de anotación

En el ejemplo anterior, la anotación @Override indica que el método está siendo sobrescrito, si no se sobrescribe el método este generará un error; el encargado de determinar dicho comportamiento será la JVM (Máquina Virtual de Java), la cual buscará los metadatos en los archivos de clases.

Otra de las características de JEE, es que permite inyecciones de dependencias, estas pueden ser utilizadas en contenedores EJB, contenedores Web y clientes de aplicación. La inyección de dependencias permite al contenedor Java EE automáticamente insertar referencias a otros componentes requeridos o recursos, en otras palabras, suministra objetos a una clase en lugar de ser la propia clase quien cree el objeto.

Por último JEE, permite realizar un mapeo entre datos relacionales y objetos mediante JPA (Java Persistence API), el objetivo de JPA es el de no perder las ventajas de la orientación a objetos al interactuar con base de datos, mediante la persistencia de datos. JPA también se puede utilizar en aplicaciones JSE (Java Standar Edition).

2.2.1. EL MODELO DE APLICACIÓN JAVA EE

La plataforma Java EE utiliza un modelo de aplicación de múltiples capas, la imagen 6 muestra un esquema general de Java EE:

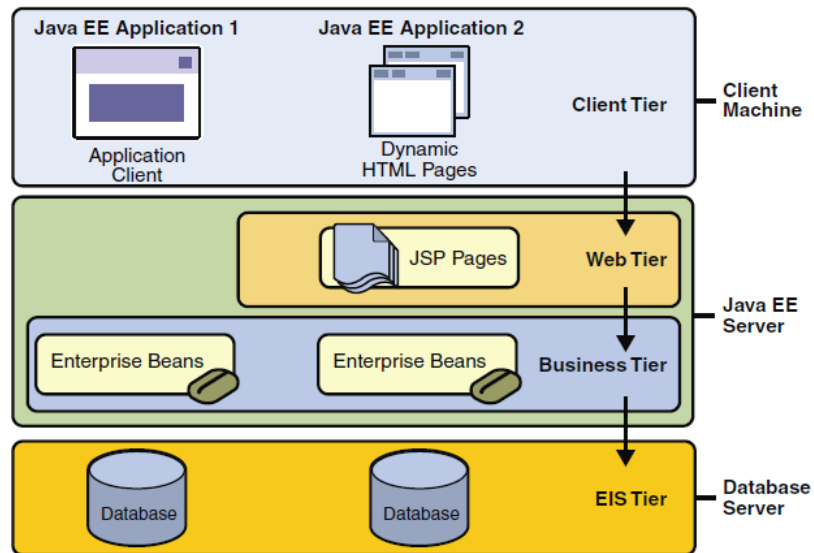


Imagen 6: Esquema General de JEE⁵

2.2.2. COMPONENTES JEE

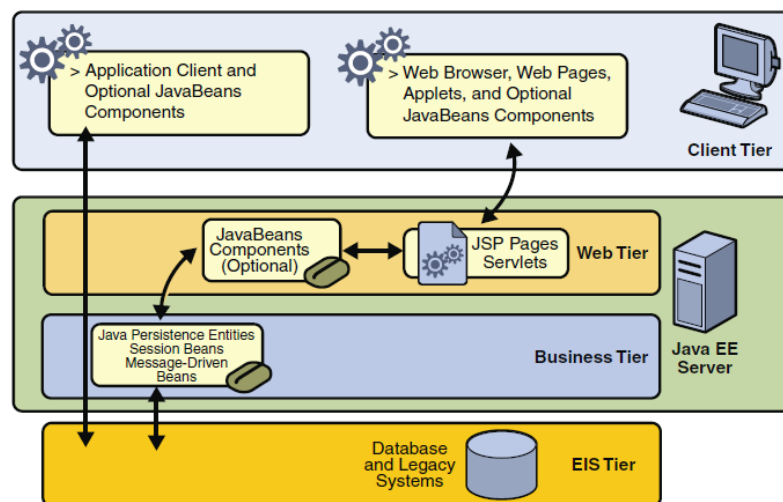


Imagen 7: Componentes de aplicación JEE⁶

A continuación una descripción de cada una de las capas:

^{5,6} Tomado de la página web: <http://docs.oracle.com/javaee/5/tutorial/doc/bnaay.html>

- Los componentes de la capa de cliente: se ejecutan en la máquina del cliente, ejemplo: navegador, aplicaciones de escritorio, etc.
- Los componentes de la capa Web: se ejecutan en el servidor Java EE y son los responsables de generar la presentación de los datos en formato HTML, ejemplo: servlets, páginas JSP (Java Server Pages), JSF (Java Server Faces), etc. Estos suelen apoyarse en el uso de componentes Java Beans los cuales se usan para encapsular varios objetos en un único objeto, para hacer uso de un solo objeto en lugar de varios más simples.
- Los componentes de la capa de negocios: se ejecutan en el servidor Java EE y son los responsables de la lógica de la aplicación.
- La capa de Sistemas de Información Empresariales (EIS): se ejecutan en el servidor EIS y es aquí donde se gestiona la información del sistema que se encuentra almacenada en base de datos.

2.2.3. JAVA EE APIS

Java EE se basa en Java SE, por lo tanto todas las APIs de Java SE están disponibles, claro que Java EE también define algunas APIs específicas para el desarrollo de sistemas multicapas.

En la imagen 8 se muestra todo el conjunto de Apis de Java EE:

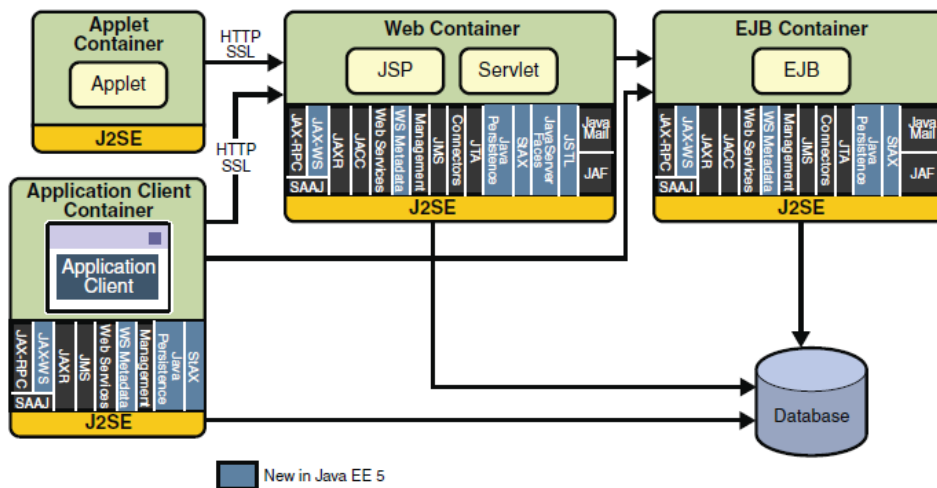


Imagen 8: Java EE APIs⁷

Las principales APIs de Java EE son:

- Enterprise Java Beans: Componentes EJB que ejecutan los procesos de negocio.
- Java Persistence API (JPA): Ofrece un soporte estándar para persistencia de objetos, esto quiere decir que permite definir un mapeo objeto-relacional para tipos de entidades mediante anotaciones o ficheros XML. JPA se encarga de gestionar las consultas, cargas, modificaciones y escritura de entidades. Se conoce como entidad a un objeto Java que representa los datos almacenados en una tupla de una base de datos.
- Java Database Connectivity (JDBC): soporte para acceso a sistemas gestores de base de datos e invocación de sentencias SQL.
- API de Servlets: servlets son objetos Java que manejan peticiones HTTP y se ejecutan en un servidor o contenedor JEE.
- Java Server Pages (JSP): lenguaje de marcado que combina Java y HTML. Las JSP's permiten la utilización de código Java mediante scripts y se consideran como una manera alternativa y simplificada de construir servlets.
- JSP Standard Tag Library (JSTL): juego de etiquetas estándar para definir páginas JSP a alto nivel.

⁷ Tomado de la página web: <http://docs.oracle.com/javaee/5/tutorial/doc/bnadj.html>

- Java Server Faces (JSF): es una tecnología y framework para aplicaciones Java basadas en web que integra un conjunto de componentes que simplifica el desarrollo de interfaces de usuario.

2.3. JAVA PERSISTENCE API

2.3.1. QUÉ ES JAVA PERSISTENCE API

Java Persistence API (JPA) provee a los programadores un mapeo objeto/relacional de manera más fácil al momento de gestionar datos relacionales en las aplicaciones y se lo puede utilizar tanto en plataformas JEE como en JSE.

Un mapeo objeto/relacional conocido también como ORM (Object Relational Mapping – Mapeo Relacional de Objetos), es la correlación entre objetos (entidades) Java y tablas de la base de datos. Esta relación se logra mediante anotaciones en las propias clases de entidad.

Las entidades son objetos que representan a tablas relacionales y cada instancia representa un registro en esa tabla.

JPA está compuesto por tres partes:

- La propia Api de Java Persistence (`javax.persistence.package`)
- El lenguaje de consulta conocido como Java Persistence Query Language (JPQL)
- Metadatos mapeo objeto/relacional.

Los tipos de datos que se pueden asignar a los campos o propiedades de una entidad son:

- Tipos primitivos de Java

- Tipo String
- Objeto serializable (BigInteger, BigDecimal, Date, Calendar, Date, etc)
- Tipos enumerados
- Otras entidades y/o colección de entidades
- Clases embebidas

Las entidades pueden utilizar campos o propiedades persistentes, dependerá únicamente del lugar donde se coloquen las anotaciones, por ejemplo, si las anotaciones de mapeo se aplican a las instancias de las entidades, la entidad utiliza campos persistentes, en cambio, si se aplican a los métodos getters de la entidad, se utilizan propiedades persistentes.

Las anotaciones de JPA se clasifican en:

- Mapeo Lógico: Permite describir el modelo de objeto, asociaciones de clase, etc.
 - @OneToOne
 - @OneToMany
 - ...
- Mapeo Físico: Describen esquemas físicos de base de datos, tablas, columnas, índices, etc.
 - @Column
 - @JoinColumn
 - ...

2.3.2. ARQUITECTURA JPA

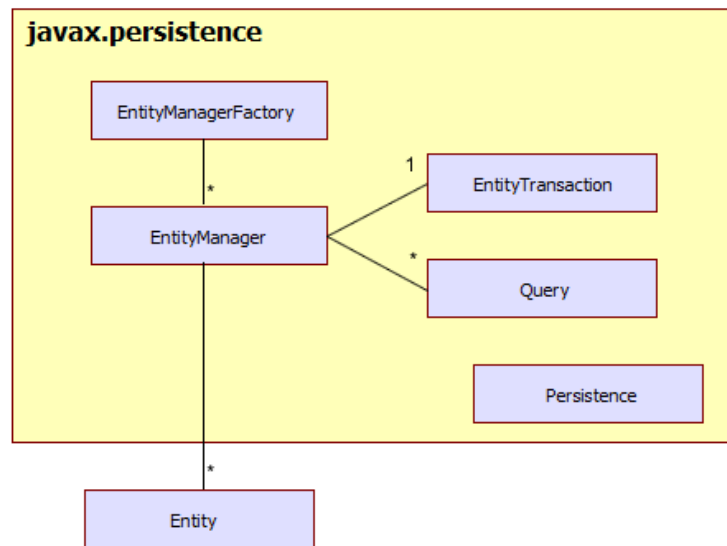


Imagen 9: Componentes principales de JPA⁸

2.3.3. CREACION DE UNA ENTIDAD

Las entidades son las piezas claves de la capa de persistencia, los valores que se almacenan en los campos de un objeto de una clase de entidad, representan la información existente en la base de datos.

Una entidad está asociada a la tabla de una base de datos y dentro de esta se utilizan distintos tipo de anotaciones para crear la asociación entre el objeto y la tabla en la base de datos. A continuación se presenta los tipos de anotaciones según su categoría:

CATEGORÍA	ANOTACIÓN
Entidad	@Entity
Atributos de esquema de base de datos	@Table
	@Column
	@JoinColumn
Identidad	@Id

⁸ Tomado de la página web: <http://www.roseindia.net/jpa/jpa-architecture.shtml>

	@GeneratedValue
Asignación de relaciones	@OneToOne
	@ManyToOne
	@OneToMany
	@ManyToMany
	@OrderBy
Manejador de entidad	@PersistenceUnit
	@PersistenceContext
	@PersistenceProperty
Consultas	@NamedQuery
	@NamedNativeQuery

Tabla 2: Lista de anotaciones JPA por categorías.

Ahora, una clase entidad deberá cumplir con los siguientes requerimientos:

- La clase no debe ser declarada como *final*.
- La clase debe tener definido un constructor sin parámetros.
- Ninguno de los componentes de la clase (atributos y métodos) puede estar definido como *final*.
- Los atributos de la entidad no pueden ser públicos.
- Es obligatorio que toda entidad cuente con una clave primaria.

Ejemplo de creación de una entidad:

Persona
-id: Long -nombre: String -apellido: String
+setId(id: Long): void +getId(): Long +setNombre(nombre: String): void +getNombre(): String +setApellido(apellido: String): void +getApellido(): String


```

@Entity
public class Persona implements Serializable {
    private Long id;
    private String nombre;
    private String apellido;
    public void setId(Long id){...}
    public Long getId(){...}
    public void setNombre(String nombre){...}
    public String getNombre(){...}
    public void setApellido(String apellido){...}
    public String getApellido(){...}
    public Persona(){...}
}

```

Ejemplo 2: Clase PERSONA transformada a una entidad.

En el Ejemplo 2, se presenta la estructura de una clase entidad. Como se mencionó anteriormente, para que la clase sea una entidad debe cumplir con ciertos requerimientos:

1. La clase debe tener como encabezado la anotación `@Entity` (`javax.persistence.Entity`) y no debe ser declarada como *final*.

```

@Entity
public class Persona implements Serializable {
    ...
}

```

Ejemplo 3: Anotación `@Entity`.

Si una instancia de una entidad es invocada de manera remota, debe implementar la interfaz `SERIALIZABLE`, caso contrario no es necesario su implementación.

2. Ninguno de los componentes de la clase (atributos y métodos) puede estar definido como *final* y los atributos de la entidad no deben ser públicos.

```

private Long id;
private String nombre;
private String apellido;
public void setId(Long id){...}
public Long getId(){...}
public void setNombre(String nombre){...}
public String getNombre(){...}
public void setApellido(String apellido){...}
public String getApellido(){...}

```

Ejemplo 4: Declaración de atributos y métodos en una entidad

3. La clase debe tener definido un constructor sin parámetros.

```

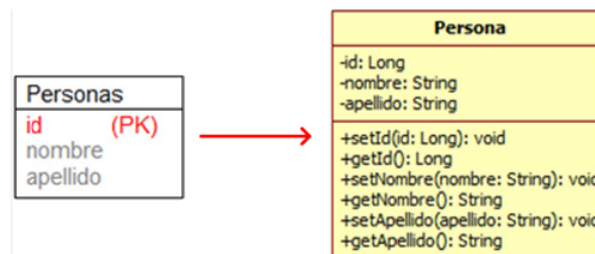
public Persona(){...}

```

Ejemplo 5: Declaración de un constructor de una entidad

Como se observa en el Ejemplo 2, la clase cumple con ciertos requerimientos, lo que falta es declarar un identificador o clave primaria dentro de la entidad (veremos más adelante como declarar un atributo como clave primaria), pero hasta este punto podemos decir que nuestra clase está definida como una entidad.

Para mapear una tabla de base de datos con una entidad, utilizamos la anotación `@Table`. Utilizando el ejemplo anterior tendríamos:



```

@Entity

```

```

@Table(name="Personas")

```

```

public class Persona implements Serializable {

```

```

private Long id;
private String nombre;
private String apellido;
public void setId(Long id){...}
public Long getId(){...}
public void setNombre(String nombre){...}
public String getNombre(){...}
public void setApellido(String apellido){...}
public String getApellido(){...}
public Persona(){...}
}

```

Ejemplo 6: Mapeo tabla PERSONAS / entidad PERSONA

En el Ejemplo 6, podemos ver que la anotación `@Table` nos permite definir el nombre de la tabla a la cual se está haciendo referencia, si no se utiliza `name` dentro de la anotación el nombre de la tabla por defecto será el nombre de la clase y en este caso JPA buscará una tabla con el mismo nombre que la clase (Véase Ejemplo 7).

```

@Entity
@Table
public class Ciudades implements Serializable {
...
}

```

Ejemplo 7: Declaración de nombre de una entidad por defecto.

2.3.4. MANEJO DE RELACIONES EN JPA

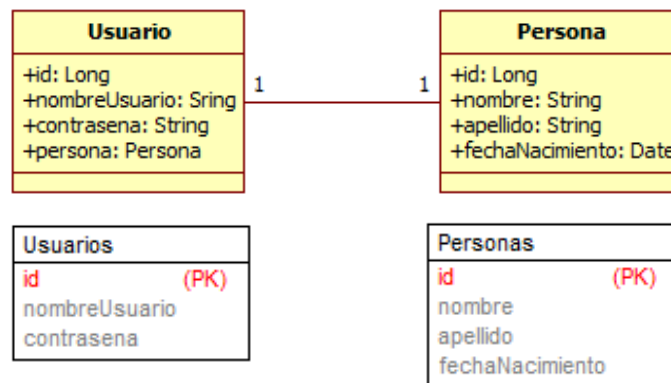
Las relaciones son similares a las que se hacen con las tablas de las bases de datos y para representar el tipo de relación entre entidades se utilizan anotaciones que representan relaciones unidireccionales o bidireccionales.

- **Relación Unidireccional:** La información solo puede viajar en una dirección y solo una entidad es la propietaria de la relación, por lo cual solo la entidad propietaria hace referencia a la otra.

- **Relación Bidireccional:** La información puede viajar en ambas direcciones. Las entidades se referencian mutuamente, pero de igual manera que en el caso anterior debe existir una entidad propietaria de la relación.

2.3.4.1. TIPOS DE RELACIONES JPA

- Relación Uno a Uno (`@OneToOne`)



Ejemplo 8: Ejemplo relación uno a uno

En el ejemplo anterior se puede observar la relación entre Usuario y Persona, esta relación es de tipo uno a uno, pues una persona es única dueña de un usuario. Para representar la relación entre las entidades usamos la anotación `@JoinColumn`:

```

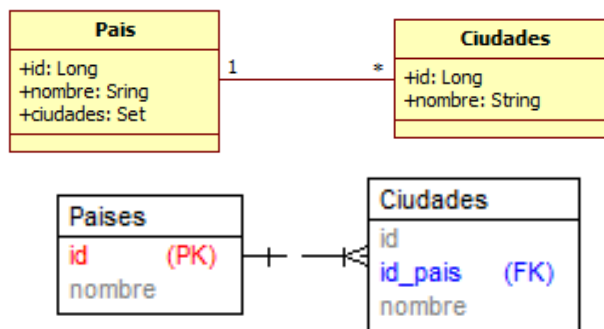
@Entity
public class Usuario implements Serializable {
    ...
    @OneToOne
    @JoinColumn
    private Persona persona;
    ...
}

@Entity
public class Persona implements Serializable {
    ...
}
  
```

Ejemplo 9: Declaración de relación uno a uno.

En esta relación la entidad *Usuario* tiene 1 referencia a *Persona* mientras que esta no tiene referencia a *Usuario*, por lo tanto *Usuario* es el dueño de la relación y se hará referencia a la clave foránea de la tabla.

- Relación Uno a Muchos / Muchos a Uno (@OneToMany / @ManyToOne)



Ejemplo 10: Relación de uno a muchos.

Este ejemplo nos muestra la relación entre Países y Ciudades, esta relación es de tipo uno a muchos. Como se puede observar en la entidad *Pais*, el atributo *ciudades* es de tipo Set (puede también ser de tipo Collection o List), esto nos indica que la entidad *Pais* contiene un set, una lista o colección de ciudades. Para configurar la relación uno a muchos se utilizan las anotaciones `@OneToMany`, `@JoinTable` con el atributo `joinColumn`:

```
@Entity
public class Pais implements Serializable{
    ...
    @OneToMany(mappedBy="idPais")
    private Set ciudades;
    ...
}
```

Ejemplo 11: Declaración relación uno a muchos.

Para la relación muchos a uno bidireccional entre *Ciudades* y *Pais*, JPA define que el extremo muchos será el dueño de la relación, por lo tanto, el atributo *pais* de la

entidad *Ciudades* será el dueño de dicha relación. Mediante la anotación `@JoinColumn` indicamos que el nombre de la clave foránea de la tabla *Pais* será *idPais* y a su vez hará referencia a la clave primaria de la tabla *Pais* llamada *id*.

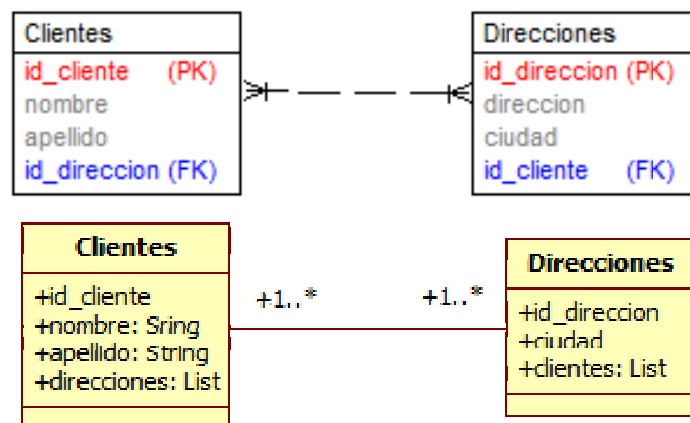
```

@Entity
public class Ciudades implements Serializable{
    ...
    @ManyToOne
    @JoinColumn(name="idPais",referencedColumnName="id")
    private Pais pais;
    ...
}

```

Ejemplo 12: Declaración relación muchos a uno.

- Relación Muchos a Muchos (`@ManyToMany`)



Ejemplo 13: Relación de muchos a muchos.

La entidad *Cientes* tiene una relación de tipo muchos a muchos con la entidad *Direcciones* y estará determinada por medio de la anotación `@ManyToMany`.

En este tipo de relaciones, JPA creará una entidad intermedia que contendrá los datos referenciales de ambas tablas y se deberá elegir qué entidad será la propietaria de la relación para que JPA realice el mapeo correctamente.

```

@Entity
public class Empleados implements Serializable{
    ...
    @ManyToOne
    @JoinTable(name="EMPLEADO_PROYECTO",
        joinColumns=@JoinColumn(name="ID_EMPLEADO"),
        inverseJoinColumns=@JoinColumn(name="ID_PROYECTO"))
    private List proyectos;
    ...
}

@Entity
public class Proyectos implements Serializable{
    ...
    @ManyToOne(mappedBy="proyectos")
    private List clientes;
    ...
}

```

Ejemplo 14: Declaración relación muchos a muchos.

La opción **mappedBy** nos indica el atributo de la entidad propietaria (EMPLEADOS) y permitirá a través de una instancia de la entidad *proyectos* podemos recuperar la instancia de la entidad Empleado asociada.

2.3.5. LENGUAJE DE CONSULTA JPQL (Java Persistence Query Lenguaje)

Dentro de Java Persistence API se define un lenguaje de consulta conocido como Java Persistence Query Language (JPQL), este se utiliza para hacer consultas a las entidades para obtener los datos almacenados en las bases de datos relacionales. Sus consultas son muy similares a la sintaxis de las consultas SQL, la diferencia es que operan sobre objetos (entidades) JPA en lugar de hacerlo directamente con las tablas de base de datos.

JPQL permite realizar consultas basándose en múltiples criterios (propiedades de una entidad o condiciones booleanas) y también posibilita recuperar más de un objeto por consulta. Dichas consultas se definen en una sola cadena y deben seguir el siguiente patrón:

```
SELECT [<resultado a obtener>]  
[FROM <entidad(es)>]  
[WHERE <filtro>]  
[GROUP BY <agrupación>]  
[HAVING <condición>]  
[ORDER BY <orden>]
```

Ejemplo 15: Sintaxis consulta en JPQL.

Ejemplo de una consulta:

```
SELECT e FROM Empleados e
```

Ejemplo 16: Ejemplo de consulta en JPQL.

Como se puede observar en el ejemplo anterior, la sintaxis de consulta es similar a las utilizadas en Oracle, Mysql, PostgreSQL, etc.

Si deseamos obtener el valor de un atributo especial, utilizamos un *alias* de la entidad:

```
SELECT e.nroCedula FROM Empleados e
```

Ejemplo 17: Consulta con alias de una tabla en JPQL.

Con JPQL también podemos obtener resultados referentes a más de una atributo y a distinto nivel:

```
SELECT e.nroCedula,e.fechaNacimiento,  
e.idDependencia.dependencia  
FROM Empleados e
```

Ejemplo 18: Visualización de atributos en una entidad.

Para recuperar datos de atributos de distintos niveles utilizamos el *punto*:

```
SELECT c.atributo1.atributo2.....atributoN
FROM Clase c
```

Ejemplo 19: Consulta de atributos a varios niveles.

También se pueden realizar consultas de una función agregada (count, avg, max, min, sum, etc):

```
SELECT COUNT(e)
FROM Empleados e
```

Ejemplo 20: Funciones agregadas en JPQL.

A continuación se presenta una tabla de las palabras reservadas de JPQL:

Select	From	Mod	Any	Delete
Join	Outer	Inner	Order	Bit_Length
By	Having	Fetch	Distinct	Object
Null	True	False	Not	And
Exists	As	Like	In	Between
Unknowm	Empty	Some	Of	Is
Avg	Max	Min	Sum	Count
Left	By	Asc	Desc	Where
Upper	Lower	Trim	Position	Character_Length
Group	Current_Time	Current_Timestamp	New	Or
All	Update	Member		

Tabla 3: Palabras reservadas de JPQL.

Para consultas condicionales utilizamos la clausula **WHERE** que nos servirá para filtrar los resultados devueltos por la consulta:

```
SELECT ap
FROM AlmacenProductos ap
WHERE ap.totalProducto < 50
```

Ejemplo 21: Consulta condicional en JPQL.

Las sentencias condicionales también pueden contener varias condiciones:

```
SELECT kp
FROM KardexProducto kp
WHERE kp.cantidadSalida > 5
AND kp.fecha = '01/08/2012'
```

Ejemplo 22: Consulta de varias condiciones en JPQL.

También se puede utilizar la condición OR si se lo requiere.

Si se necesita obtener resultados dentro de un rango de valores se utiliza el operador BETWEEN (también se puede usar NOT BETWEEN):

```
SELECT e
FROM Empleados e
WHERE e.fechaIngreso
BETWEEN '01/01/2012' AND '01/08/2012'
```

Ejemplo 23: Consulta condicional con rangos en JPQL.

Otro operador comparativo es LIKE o NOT LIKE, este operador nos permite comparar una cadena de texto:

```
SELECT e
FROM Empleado e
WHERE e.apellidoPaterno LIKE 'cast%'
```

Ejemplo 24: Consulta con operadores para cadenas de texto en JPQL.

Dentro de las consultas es muy común utilizar parámetros y en JPQL se las puede hacer de dos maneras:

- Parámetros por posición. Ejemplo:

```
SELECT e
FROM Empleados e
WHERE e.codigoEmpleado = ?1
```

Ejemplo 25: Declaración de parámetros por posición.

- Parámetros por nombre. Ejemplo:

```
SELECT e
FROM Empleados e
WHERE e.codigoEmpleado = :código
```

Ejemplo 26: Declaración de parámetros por nombre.

A continuación se mostrará la forma en que se implementan las consultas JPQL en una clase, para ellos necesitaremos las siguientes interfaces:

- Persistence: contiene métodos estáticos donde obtendremos una instancia de EntityManagerFactory.
- EntityManagerFactory: nos ayuda a crear objetos de EntityManager.
- EntityManager: es la interfaz principal de JPA utilizada para la persistencia de las aplicaciones.
- Query: implementación para encontrar objetos persistentes manejando cierto criterio de búsqueda.

Desde una instancia EntityManager es posible obtener una instancia Query. Para ello podemos existen tres formas distintas:

- createQuery(String cadena); para consultas JPQL.
- createNamedQuery(String cadena); para consulta con nombre.
- createNativeQuery(String cadena); para consultas SQL.

Código de la estructura de consultas JPQL:

```
import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Query;
import javax.persistence.Persistence;

public class Empleados implements Serializable
```

```

{
    ...
    EntityManagerFactory emf =
    Persistence.createEntityManagerFactory("PersistenceUnitName");
    EntityManager em = emf.createEntityManager();
    Query q = em.createQuery("Consulta_JPQL");
    List<Empleados> resultados = q.getResultList();
    ...
}

```

Ejemplo 27: Declaración de una consulta dentro de una clase.

Hay dos maneras de obtener resultados en una consulta JPQL:

- Si el resultado devolverá 0 o 1 resultados:

```

Object resultado = query.getSingleResult();
Ejemplo 28: Consulta con resultado simple en JPQL.

```

- Si la consulta devuelve varios resultados o simplemente se desconoce:

```

List results = query.getResultList();
Ejemplo 29: Consulta con múltiples resultados en JPQL.

```

Una consulta con parámetros funcionaria de la siguiente manera:

```

Query q = em.createQuery("SELECT e FROM Empleado e WHERE
e.apellido = ?1");
q.setParameter(1, "castro");
List<Empleados> resultado = q.getResultList();

```

Ejemplo 30: Declaración de una consulta utilizando parámetros por posición.

Utilizamos `setParameter(int, Object)` para insertar el valor correspondiente al parámetro de la posición ?1 que utilizaremos en la consulta; si utilizamos parámetros por nombre utilizaremos `setParameter(String, Object)` y nuestra la consulta o sentencia JPQL sería:

```

Query q = em.createQuery("SELECT e FROM Empleado e WHERE
e.nroCedula = :cedula");

```

```
q.setParameter("cedula","0101010101");
```

```
Empleados resultado = q.getSingleResult();
```

Ejemplo 31: Declaración de una consulta utilizando parámetros por nombre.

Las consultas con nombre son muy diferentes de las sentencias dinámicas anteriormente descritas, estas consultas son más eficientes y ofrecen un mejor rendimiento. Las consultas con nombre son definidas por metadatos definidos en la propia entidad mediante *@NamedQuery*:

```
@Entity
```

```
@Table(name = "EMPLEADOS")
```

```
@NamedQuery(name = "Empleados.buscarTodos", query = "SELECT e  
FROM Empleados e")
```

```
public class Empleados implements Serializable {
```

```
...
```

```
}
```

Ejemplo 32: Declaración de una consulta con nombre en una entidad.

Donde *name* es el nombre de la consulta, y *query* es la sentencia JPQL a ejecutar.

Para ejecutar una consulta con nombre debemos hacerla mediante

createNamedQuery():

```
Query q = em.createNamedQuery("Empleado.buscarTodos");
```

```
List<Empleados> resultado = q.getResultList();
```

Ejemplo 33: Utilización de una consulta con nombre.

createNamedQuery(name="...") requiere de un parámetro de tipo String que será el nombre de la consulta, dicho nombre está definido en *@NamedQuery(name="...")*.

Si necesitáramos realizar consultas más complejas y directamente desde la tablas de la base de datos utilizamos SQL nativo, y para ello deberíamos hacer es utilizar *createNativeQuery(String cadena)*:

```
Query q = em.createNativeQuery("SELECT * FROM  
EMPLEADOS");
```

```
List<Empleados> resultado = q.getResultList();
```

Ejemplo 34: Declaración de una consulta nativa.

2.4. JAVA JSF FRAMEWORK

2.4.1. ¿QUE ES JSF?

Esencialmente, JavaServerFaces permite la creación de interfaces de usuario para aplicaciones Web.

Al establecer un marco de desarrollo definido, permite construir interfaces de usuario mediante patrones establecidos simplificando el desarrollo de dichas interfaces.

2.4.2. CARACTERÍSTICAS DE JSF

- JavaServerFaces proporciona componentes centrados en el cliente lo que mejora la productividad al momento de desarrollar las aplicaciones, además es fácil de usar.
- Simplifica el acceso y manejo de los datos del usuario.
- Gestión automática del estado de la interfaz de usuario entre múltiples solicitudes y múltiples clientes de una manera sencilla y discreta.
- Proporciona un marco de desarrollo amigable.

2.4.3. JSF Y EL MODELO MVC

JSF sigue el patrón de diseño Modelo-Vista-Controlador (MVC), esto quiere decir que nuestra aplicación estará segmentada en tres componentes de aplicación distintos:

- El Modelo, que contiene la lógica de negocio.
- La Vista, representa la interfaz de usuario. En esta caso las páginas web.
- El Controlador, que es un agente de front-end que directamente gestiona las peticiones de los usuarios y los despachos de la vista correspondiente.

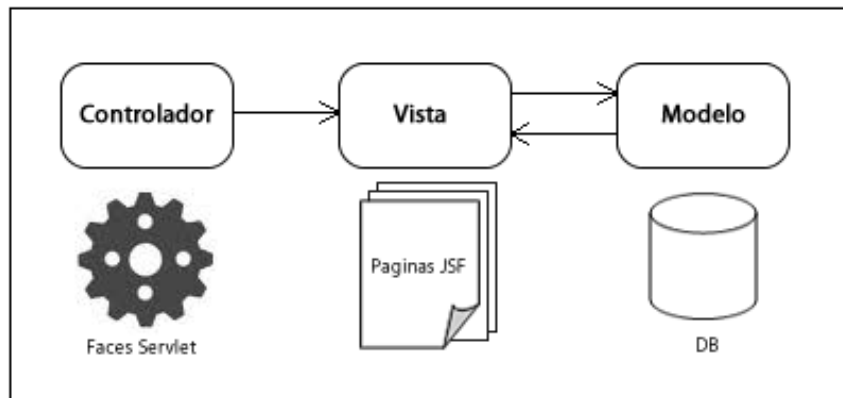


Imagen 10: Modelo MVC de JSF.

2.4.4. COMPONENTES DE UNA APLICACION JSF

Todas las aplicaciones JSF están formadas por un conjunto específico de archivos de configuración, los controladores beans (son las clases que actuarán como controladores de las acciones) y los archivos de contenido Web (HTML, CSS, etc). Entre los archivos de configuración más importantes que se requieren son:

- faces-config.xml: el archivo de configuración maestro requerido por todas las aplicaciones JSF, contiene una referencia para todas las partes de una aplicación.
- web.xml: es el descriptor de despliegue J2EE Web y el archivo maestro de configuración requerida por cualquier aplicación web J2EE.

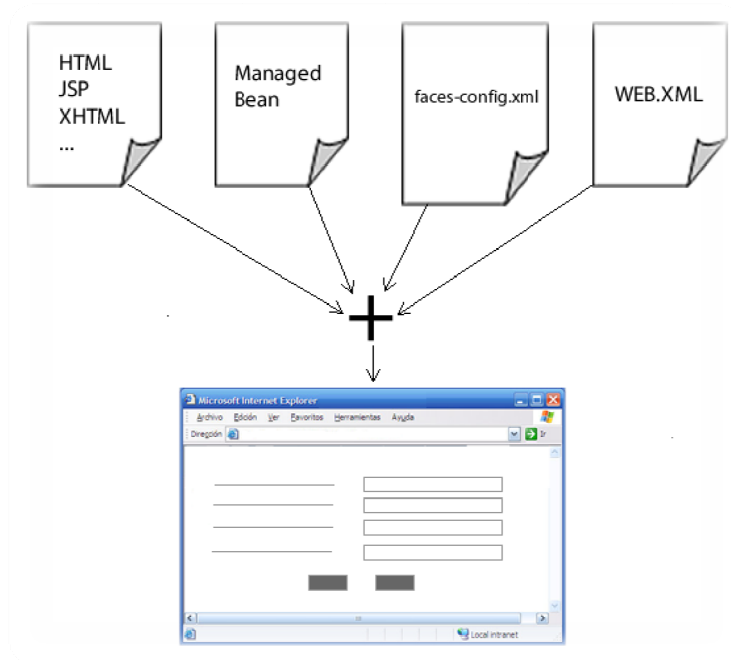


Imagen 11: Componentes principales de una aplicación JSF.

2.4.5. MANEJO DE CONTROLADORES BEAN

Los controladores beans son esencialmente POJOs que almacenan los datos de la aplicación y son gestionados por el contenedor. Ellos están registrados con el tiempo de ejecución de JSF a través de un archivo XML y se inicializa cuando son necesarios por la aplicación.

Ejemplo de un controlador bean definido dentro de faces-config.xml:

```

<managed-bean>
  <managed-bean-name>personasBean</managed-bean-name>
  <managed-bean-class>
    org.controlador.PersonaBean
  </managed-bean-class>
  <managed-bean-scope>session</managed-bean-scope>
</managed-bean>

```

Ejemplo 35: Definición de un controlador bean dentro del archivo faces-config.xml.

Se puede definir un controlador bean utilizando anotaciones dentro las clases:


```

@ManagedBean(name = "personaBean")
@SessionScoped
public class PersonaBean {
...
}

```

Ejemplo 36: Declaración de un controlador bean en una clase.

2.4.5.1. TIPOS DE BEAN SCOPES

Sirven para definir el grado de accesibilidad del bean dentro del repositorio virtual. Los valores que pueden tomar un bean son:

- Request: se crea una nueva instancia por cada petición (request) HTTP.
- Session: Se crea una instancia por cada sesión HTTP.
- Application: Se mantienen los valores de la instancia creada durante la duración o tiempo de vida de la aplicación.

2.4.6. PRINCIPALES ETIQUETAS JSF

ETIQUETA	DESCRIPCION
h:commandButton	Crea un botón
h:commandLink	Un enlace hipertexto
h:dataTable	Crea una tabla de datos dinámica.
h:form	Define un formulario en la página.
h:graphicImage	Muestra una imagen(jpg, gif, png, etc)
h:inputHidden	Campo oculto
h:inputText	Campo de texto normal
h:inputSecret	Campo de tipo contraseña.
h:inputTextarea	Campo de texto multi línea.
h:message	Imprime mensaje de error en la página
h:messages	Imprime varios mensajes de error
h:outputFormat	Muestra texto parametrizado.

h:outputLabel	Muestra un texto fijo.
h:outputLink	Crea un enlace hipertexto.
h:outputText	Formatea un texto de salida
h:panelGrid	Panel cuadrículado
h:panelGroup	Agrupar varios componentes.
h:selectBooleanCheckbox	Checkbox booleanos
h:selectManyCheckbox	Conjunto de casillas activables.
h:selectManyListbox	Lista que permite seleccionar múltiples elementos.
h:selectManyMenu	Lista desplegable de selección múltiple.
h:selectOneListbox	Lista en la que se puede seleccionar un único elemento.
h:selectOneMenu	Lista desplegable de selección.
h:selectOneRadio	Lista de radio buttons

Tabla 4: Principales etiquetas de JSF.

2.4.7. CONVERTIDORES Y VALIDADORES

Dos temas importantes que se deben considerar en el desarrollo de las aplicaciones web son las conversiones y las validaciones de datos. JSF dispone de componentes que permiten manejar estos aspectos al momento de desarrollar las aplicaciones.

La conversión de datos es manejada por los convertidores que aseguran que la aplicación de la lógica de negocio solo haga referencia a datos del tipo de lenguaje de Java.

La validación es manejada por los validadores donde su principal función es el de asegurar que los datos escritos no se encuentren en contra de limitaciones o reglas establecidas por la aplicación.

La conversión y validación se puede dar mediante 2 tipos: estándar y personalizada.

Las conversiones y validaciones son aceptadas únicamente por componentes del tipo `UIInput` (`inputText`, `inputSecret`, `inputHidden`, etc) y `outputText`.

2.4.7.1. CONVERTIDORES

Debemos tomar en cuenta que JSF dispone de dos tipos de convertidores: tipos numéricos (`NumberConverter`) y tipos Fecha (`DateTimeConverter`).

TIPO	IDENTIFICADOR
<code>BigDecimalConverter</code>	<code>BigDecimal</code>
<code>BigIntegerConverter</code>	<code>BigInteger</code>
<code>NumberConverter</code>	<code>Number</code>
<code>IntegerConverter</code>	<code>Integer</code>
<code>ShortConverter</code>	<code>Short</code>
<code>ByteConverter</code>	<code>Byte</code>
<code>CharacterConverter</code>	<code>Character</code>
<code>FloatConverter</code>	<code>Float</code>
<code>DoubleConverter</code>	<code>Double</code>
<code>BooleanConverter</code>	<code>Boolean</code>
<code>DateTimeConverter</code>	<code>DateTime</code>

Tabla 5: Convertidores JSF estándar.

2.4.7.1.1. CONVETIDOR ESTANDAR

Ejemplo aplicado con un convertidor numérico y de fecha estándar:

```
<h:inputText id="saldo" value="#{personaBean.saldo}">
  <f:convertNumber type="currency"/>
</h:inputText>
<h:inputText id="fechaNac" value="#{personaBean.fechaNacimiento}">
  <f:convertDateTime pattern="dd/MM/yyyy"/>
</h:inputText>
```

Ejemplo 37: Utilización de un convertidor estándar JSF.

En las siguientes tablas, se muestran los atributos que pueden tomar los convertidores numéricos y de fecha:

ATRIBUTO	TIPO
currencyCode	String
currencySymbol	String
groupingUsed	boolean
integerOnly	boolean
locale	java.util.Locale
maxFractionDigits	int
maxIntegerDigits	Int
minFractionDigits	Int
minIntegerDigits	int
pattern	String
type	String

Tabla 6: Atributos para convertidores numéricos.

ATRIBUTO	TIPO
dateStyle	String
parseLocale	String / Locale
pattern	String
timeStyle	String
timeZone	String / TimeZone
type	String

Tabla 7: Atributos para convertidores de fecha.

2.4.7.1.2. CONVERTIDORES PERSONALIZADOS

Al momento de crear nuestro convertidor debemos tener en cuenta las siguientes observaciones:

- Tener en cuenta que deseamos que nuestro converso realice
- Crear una clase que implemente la interfaz *javax.faces.convert.Converter*.
- Creamos los métodos *getAsObject* y *getAsString*.
 - *getAsObject ()* analiza una cadena en una subclase *java.lang*.
 - *getAsString ()* convierte el objeto de modelo de nivel de una cadena para la representación.
- Luego registramos el nuevo convertidor en el contexto de JSF (*faces-config.xml*)
- Y por último para utilizarlo dentro de la página los insertamos dentro del componente que necesitemos convertir tag `<f:converter>`.

En el siguiente ejemplo se muestra como se crea una clase que convertirá el contenido de un input en mayúsculas:

```
import javax.faces.component.UIComponent;
import javax.faces.context.FacesContext;
import javax.faces.convert.Converter;

public class UppercaseConverter implements Converter {
    public UppercaseConverter() {}
    public Object getAsObject(FacesContext facesContext, UIComponent
component, String value) {
        return value.toString().toUpperCase();
    }
}
```

```

    public String getAsString(FacesContext facesContext, UIComponent
component, Object value) {
        return value.toString().toUpperCase();
    }
}

```

Ejemplo 38: Definición de un convertidor personalizado en JSF.

Registro del convertidor en faces-config.xml:

```

<converter>
<converter-id>uppercaseConverter</converter-id>
<converter-class>org.com.converter.UppercaseConverter</converter-class>
</converter>

```

Ejemplo 39: Declaración de un convertidor personalizado en archivo faces-config.xml.

Aplicando el convertidor a un componente:

```

<h:inputTextarea id="dirDom" value="#{empleados.direccionDomicilio}">
<b:f:converter converterId="uppercaseConverter" for="dirDom"/>
</h:inputTextarea>

```

Ejemplo 40: Utilización de un convertidor personalizado en JSF.

2.4.7.2. VALIDACIONES

Las validaciones son utilizadas para que los datos de uno o más componentes sigan las reglas de la aplicación.

Los validadores se pueden especificar usando el atributo de validación de un componente o mediante la anidación de etiquetas JSF. La validación se puede utilizar únicamente en componentes de tipo UIInput o componentes que extienden la clase UIInput.

2.4.7.2.1. VALIDADORES ESTANDAR

Al utilizar las validaciones estándar, no es necesario escribir ningún código para realizar la validación, pero si se desea validar según un criterio específico, se puede hacer referencia a un controlador bean y anidar la etiqueta de validación al componente.

El conjunto de validadores estándar disponibles son:

ATRIBUTO	EJEMPLO	DESCRIPCION
ValidateDoubleRange	<code><f:validateDoubleRange minimum= 0 maximum="10" /></code>	El valor del componente deberá estar un valor mínimo y un valor máximo.
ValidateLength	<code><f:validateLength minimum= 0 maximum="10" /></code>	Valida que la longitud del valor del componente esté entre un valor mínimo y un valor máximo.
ValidateLongRange	<code><f:validateLongRange minimum= 0 maximum="10" /></code>	El valor del componente debe estar entre un valor mínimo y un valor máximo.

Tabla 8: Validadores estándar de JSF.

2.4.7.2.2. VALIDACIONES PERSONALIZADAS

Al igual que los convertidores, crear un validador personalizado es muy fácil, solamente es necesario implementar la interfaz *javax.faces.validator.Validator* en la clase y registrar la misma en el faces-config.

Ejemplo de validador personalizado:

```
import javax.faces.application.FacesMessage;
import javax.faces.component.UIComponent;
import javax.faces.context.FacesContext;
import javax.faces.validator.*;
```

```

public class ValidarEmail implements Validator {
private static final String EMAIL_REGEX = "[_A-Za-z0-9-]+(\\.[_A-Za-
z0-9-]+)*@[A-Za-z0-9-]+(\\.[A-Za-z0-9-]+)*(\\.[A-Za-z]{2,})$";

public boolean validarEmail(String email)
{
    Pattern mascara = null;
    mascara = Pattern.compile(EMAIL_REGEX);
    Matcher coincidencia = mascara.matcher(email);
    return coincidencia.matches();
}

@Override
public void validate(FacesContext context, UIComponent component,
Object value) throws ValidatorException {
    if (validarEmail((String) value)) {
        FacesMessage message = new FacesMessage();
        message.setDetail("Ingrese un correo electronico valido");
        message.setSummary(component.getId()+" no valido");
        message.setSeverity(FacesMessage.SEVERITY_ERROR);
        throw new ValidatorException(message);
    }
}
}
}

```

Ejemplo 41: Definición de un validador personalizado en JSF.

Registrando la clase en faces-config.xml:

```

<validator>
<validator-id>validarEmail</validator-id>
<validator-class>org.com.validadores.ValidarEmail</validator-class>
</validator>

```

Ejemplo 42: Declaración de un validador personalizado en archivo faces-config.xml.

Aplicamos el validador a un componente:


```
<h:inputText id="emailPersonal" size="50" value="#{empleados.emailPersonal}">
<f.validator validatorId="validarEmail"/>
</h:inputText>
```

Ejemplo 43: Utilización de un validador personalizado en JSF.

2.5. FRAMEWORKS JSF

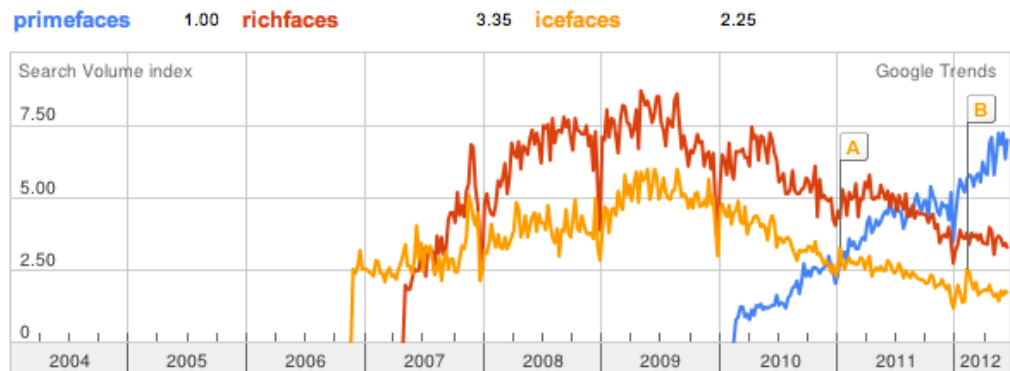
Dentro de los frameworks para JSF, podemos encontrar algunas alternativas como por ejemplo: Primefaces, RichFaces y ICEfaces. Cada una de estas proporciona elementos o componentes que facilitan el desarrollo de aplicaciones:

- Primefaces que es una librería de componentes visuales open source con soporte ajax y es desarrollada y mantenida por Prime Technology.
- RichFaces es una biblioteca de código abierto basada en Java que permite crear aplicaciones web con ajax, mantenida y desarrollada por JBoss.
- ICEFaces es un Framework basado en ajax que permite desarrollar Aplicaciones RIA de forma más fácil y rápida, sigue una serie de estándares que permiten trabajar en un ambiente normal de desarrollo con Java.

2.5.1. COMPARACION DE FRAMEWORKS

En <http://ingmmurillo.blogspot.com/2012/07/popularidad-de-primefaces.html> se muestran la popularidad de los framework JSF:

- Scale is based on the average worldwide traffic of **primefaces** in all years. [Learn more](#)
- An improvement to our geographical assignment was applied retroactively from 1/1/2011. [Learn more](#)



- Scale is based on the average worldwide traffic of **primefaces** in 2012. [Learn more](#)
- An improvement to our geographical assignment was applied retroactively from 1/1/2011. [Learn more](#)

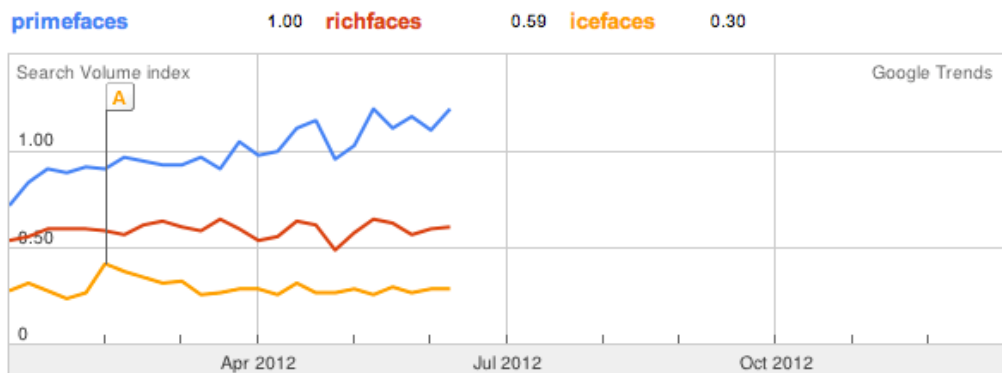


Imagen 12: Popularidad entre frameworks JSF⁹

En <http://devrates.com/stats/index> se muestra el ranking de los frameworks para web más populares:

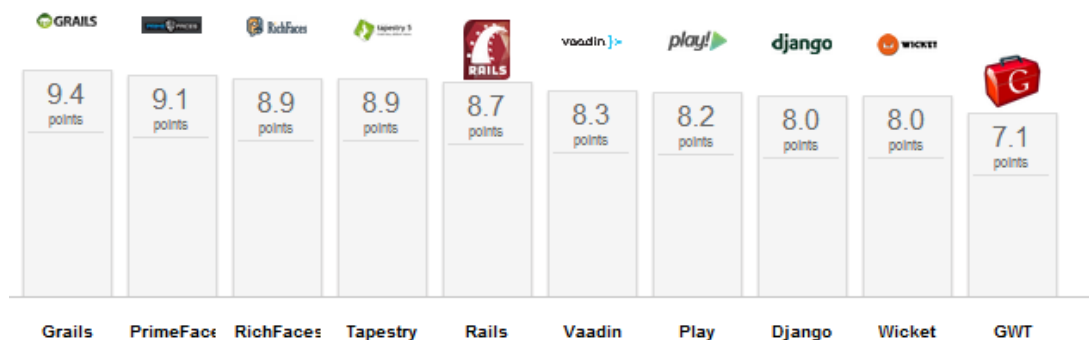


Imagen 13: Ranking Web Frameworks¹⁰

⁹ Tomado de la pagina web: <http://ingmmurillo.blogspot.com/2012/07/popularidad-de-primefaces.html>

¹⁰ Tomado de la pagina web: <http://devrates.com/stats/index>

En <http://www.clearminds-it.com/index.php?/JSF/primefaces-vs-richfaces-vs-icefaces.html> se hace una comparación entre los frameworks antes mencionados y nos muestra los resultados obtenidos luego de realizar pruebas con ellos:




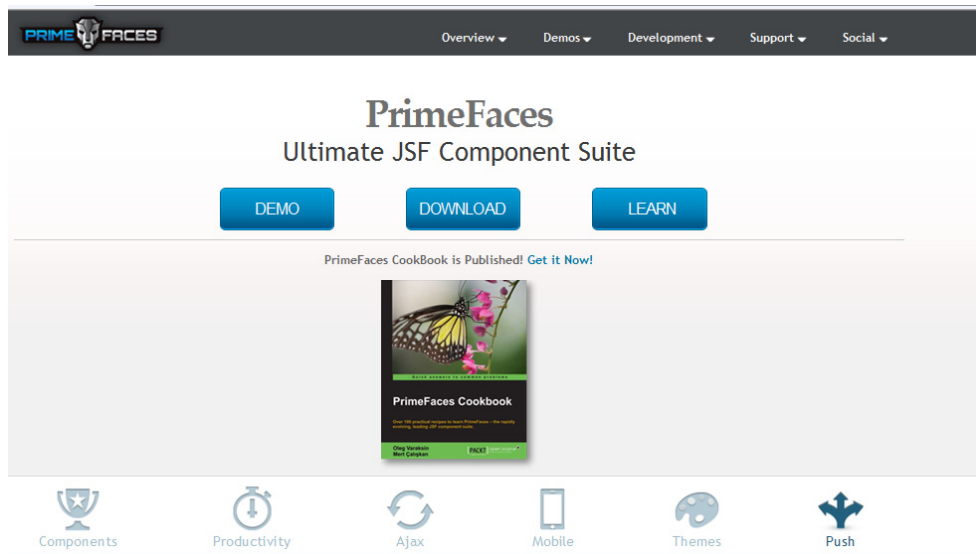
PROBLEMA:	Para probar el rendimiento, se utiliza el componente JSF, un dataTable compuesto de 5 columnas y 100 filas. Cada fila contiene un único String randómico el cual emula (en términos de tiempo) la recuperación de datos desde una Base.		
REQUERIMIENTOS:	Una laptop con un procesador Intel core 5i, dos procesadores, 8GB de ram y Windows 7. Herramienta Apache ab stress tool, cuando se dispararon 5000 request (10 concurrentes)		
			
DOCUMENT LENGTH	47442 bytes	76072 bytes	53904 bytes
REQUESTS PER SECOND:	28.59 [#/sec] (mean)	11.12 [#/sec] (mean)	33.46 [#/sec] (mean)
TIME PER REQUEST:	174.869 [ms] (mean)	899.281 [ms] (mean)	149.422 [ms] (mean)
TIME PER REQUEST:	34.974 [ms] (mean, across all concurrent requests)	89.928 [ms] (mean, across all concurrent requests)	29.884 [ms] (mean, across all concurrent requests)
TRANSFER RATE:	1329.59 [Kbytes/sec] received	829.41 [Kbytes/sec] received	1767.06 [Kbytes/sec] received
CONNECTION TIMES	min mean[+/-sd] median max	min mean[+/-sd] median max	min mean[+/-sd] median max
CONNECT:	78	266	63
PROCESSING:	766	12953	1094
WAITING:	578	12938	594
TOTAL:	766	12969	1094
CONCLUSIÓN:	El mayor resultado lo produjo Primefaces, seguido de Richfaces. Icefaces fue el último, debido mayormente a el hecho de que produce una página HTML más pesada (sobre 70KB)		

Tabla 2.9: Prueba de rendimiento entre Primefaces, Richfaces y Icefaces¹¹

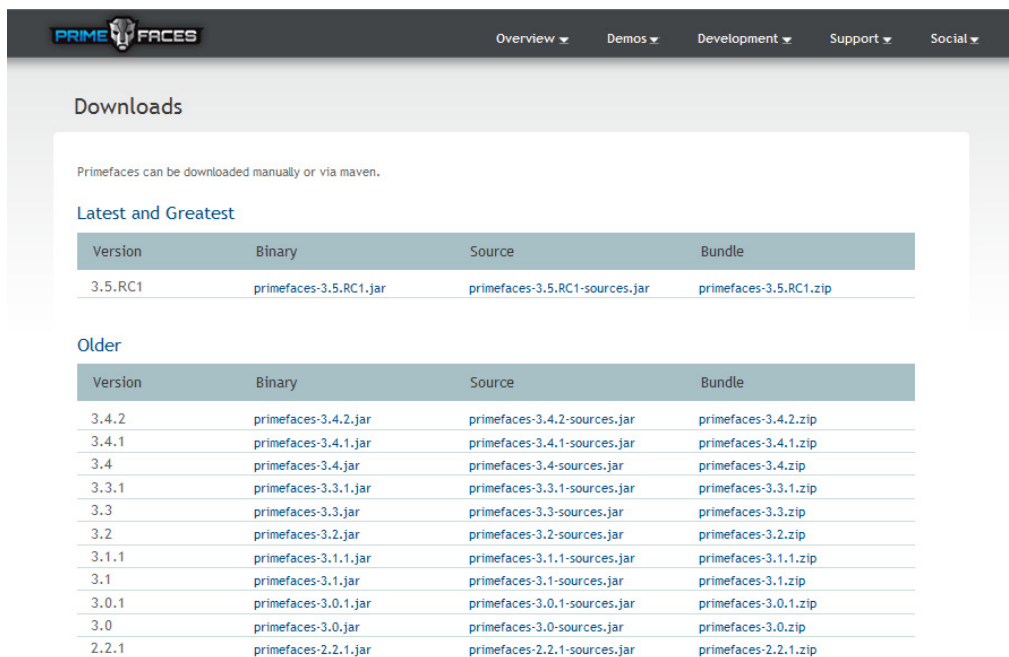
¹¹ Tomado de la pagina web: <http://www.clearminds-it.com/index.php?/JSF/primefaces-vs-richfaces-vs-icefaces.html>

2.5.2. INSTALACIÓN DE PRIMEFACES

Para utilizar los componentes de Primefaces, lo primero que debemos hacer es descargarnos la librería desde la página oficial: <http://primefaces.org/>

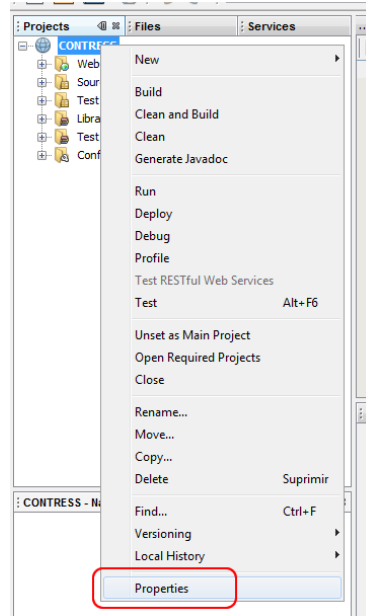


Hacemos clic en  :

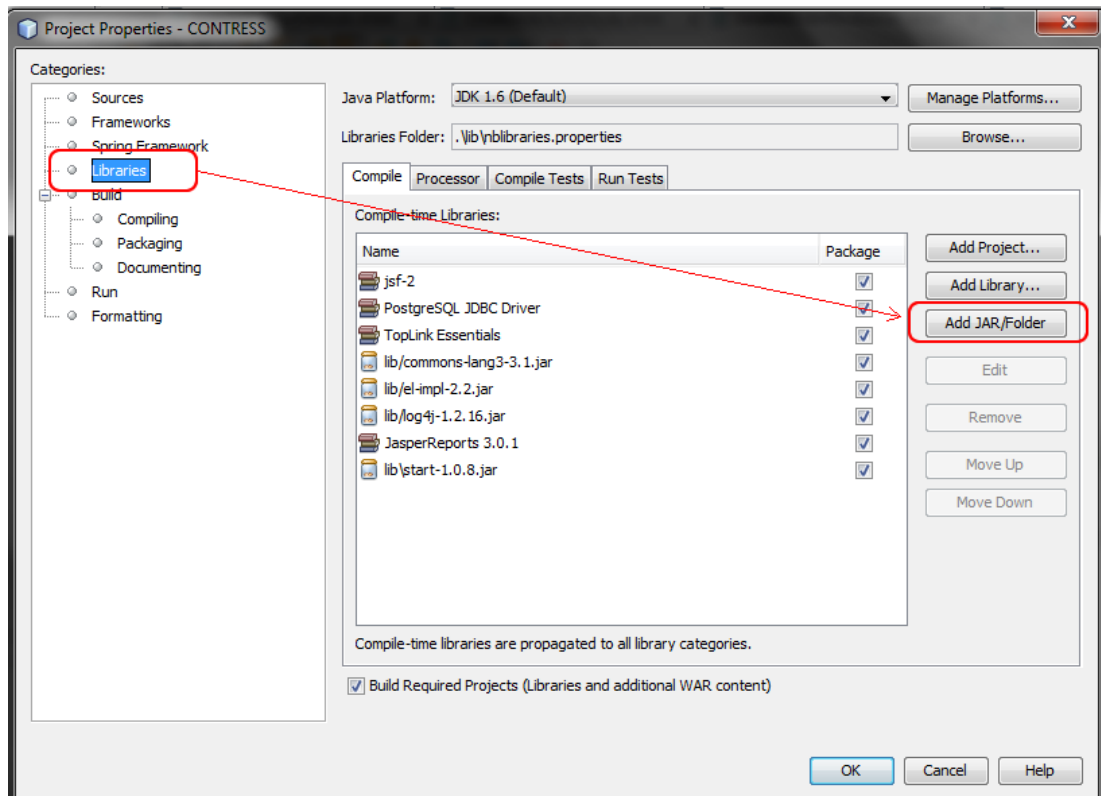


Y seleccionamos la última versión estable, en nuestro caso elegiremos la versión 3.4.2 de tipo jar.

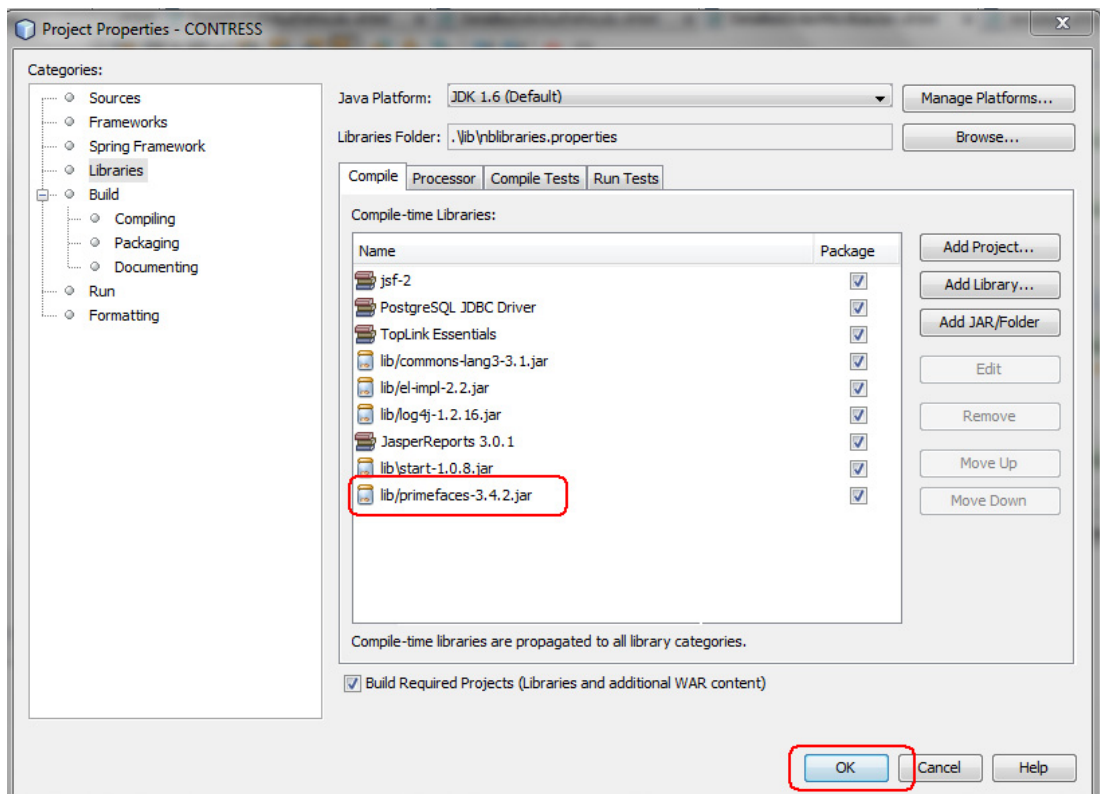
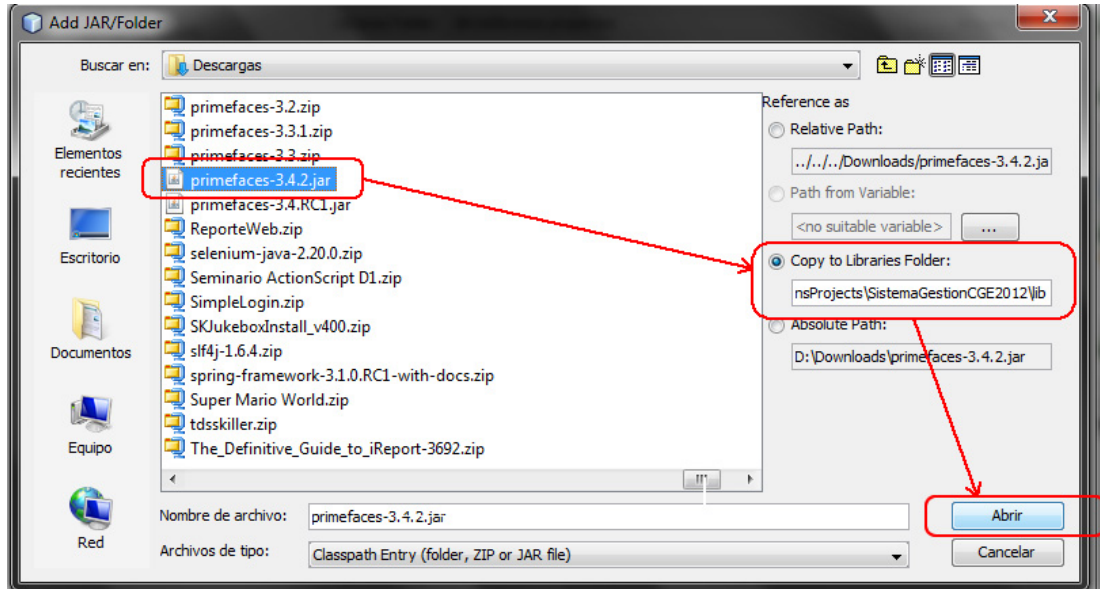
Una vez descargado el archivo, cargaremos la archivo a las librerías de nuestro proyecto, para ello hacemos clic derecho en el proyecto y elegimos la opción **PROPERTIES** :



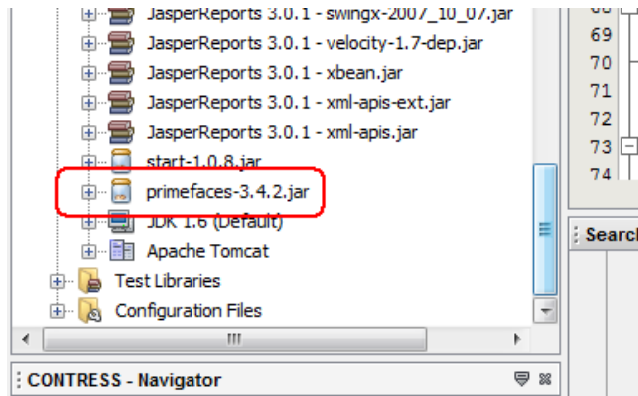
Aparecerá la ventana de propiedades del proyecto, aquí elegiremos la opción **LIBRARIES** y luego la opción **ADD JAR/FOLDER**:



Se abrirá la ventana de carga de archivo, nos ubicaremos en el path en donde descargamos el archivo, seleccionamos el archivo, escogemos la opción **COPY TO LIBRARIES FOLDER** y hacemos clic en abrir:



Listo, tenemos cargado la librería de Primefaces en nuestro proyecto y estamos listos para usarlo:



Ahora, para utilizar los componentes de Primefaces dentro de las páginas debemos definir el namespace de Primefaces (`xmlns:p="http://primefaces.org/ui"`) dentro de la etiqueta `html`, nos quedaría de la siguiente manera:

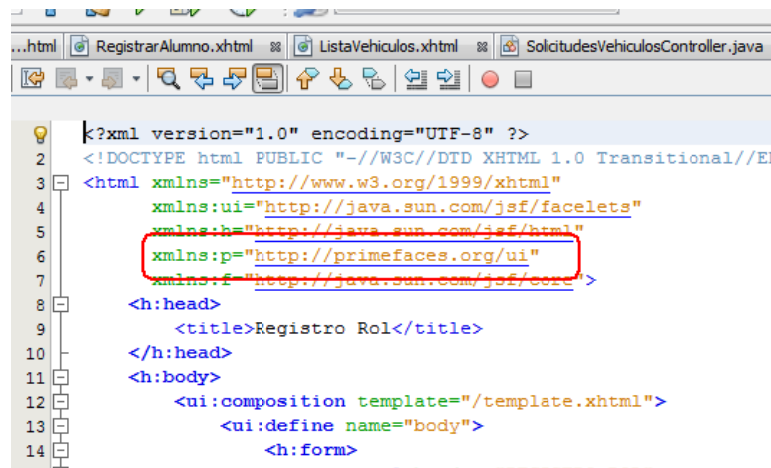


Imagen 14: Definición del namespace de Primefaces en una página web.

Un namespace es un medio que nos permiten organizar clases dentro de un entorno.

Ahora ya podemos utilizar los componentes de Primefaces en nuestras páginas. A continuación un pequeño ejemplo:

```

<h:form>
<h:outputLabel for="firstname" value="Nombre: " />
<p:spacer width="2"/>
<p:inputText id="firstname" value=" " required="true"/>
<p:spacer width="2"/>

```



```
<p:commandButton actionListener="#" icon="ui-icon-check"
value="Submit"/>
</h:form>
```



Ejemplo 44 : Implementación de componentes Primefaces en una pagina web.

2.6. SERVIDORES WEB

2.6.1. ¿QUÉ ES UN SERVIDOR WEB?

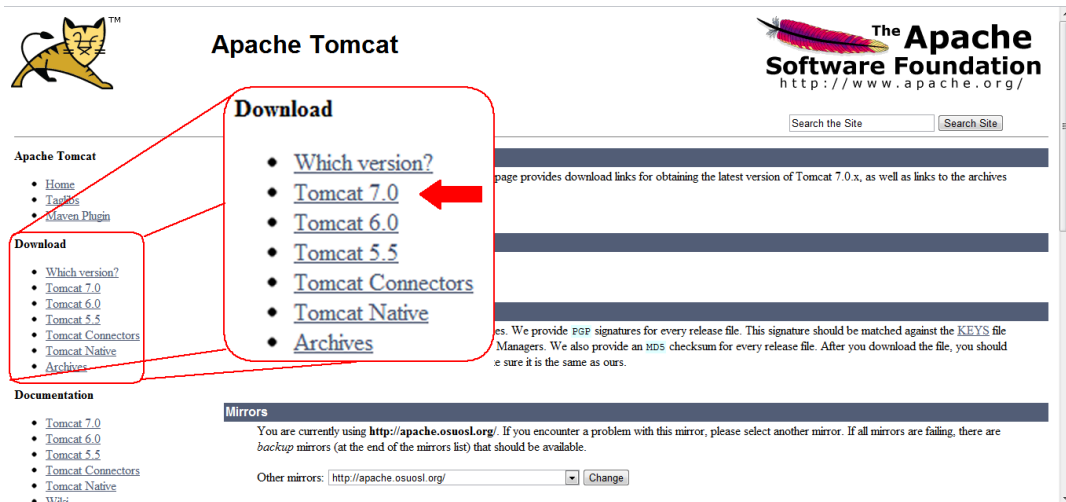
Un servidor web o servidor HTTP es un programa informático que se ejecuta continuamente y que espera y procesa peticiones de ejecución por parte de un cliente o usuario y devolviendo una página web o cualquier tipo de información como resultado.

Entre los servidores web más conocidos y utilizados está Apache Tomcat el cual fue desarrollado por miembros de la Apache Software Foundation, su última versión estable es la 7.0.27, funciona como un contenedor de servlets y JSP, al estar escrito en java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

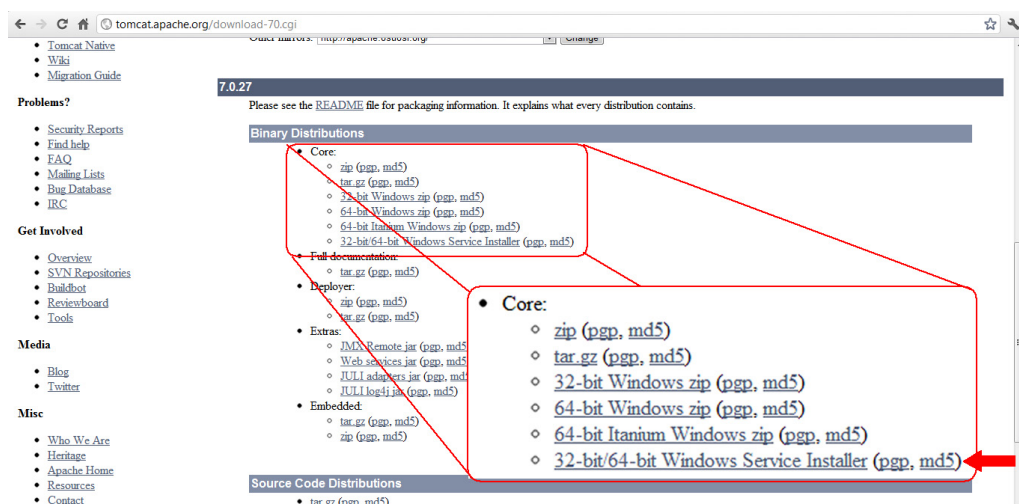
2.6.2. INSTALACION APACHE TOMCAT SERVER

Descargamos el instalador desde la página oficial de Apache Tomcat (tomcat.apache.org).

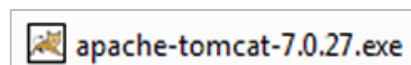
Una vez en la página, nos ubicamos en la sección de descargas y elegimos la versión 7.0:



Al hacer clic nos redirigiremos a las opciones de descarga, como nuestro servidor será instalado en un sistema operativo Windows, necesitaremos un ejecutable, por tanto elegiremos la opción *32-bit/64-bit Windows Service Installer* y esperamos a que se descargue el instalador:

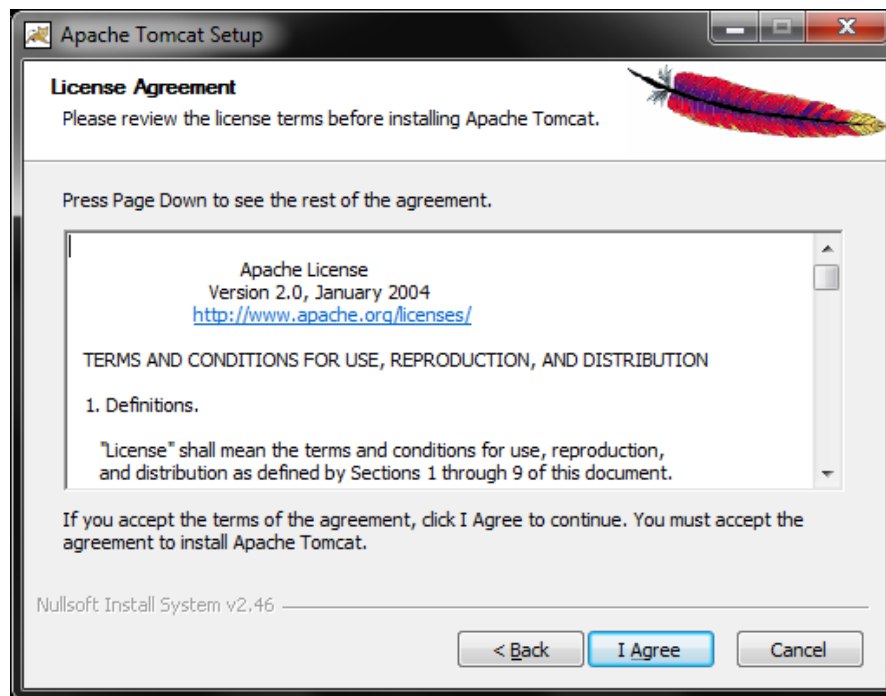


Una vez descargado, haremos doble clic en el archivo de instalación, iniciará la instalación y luego daremos un clic en el botón *next*:

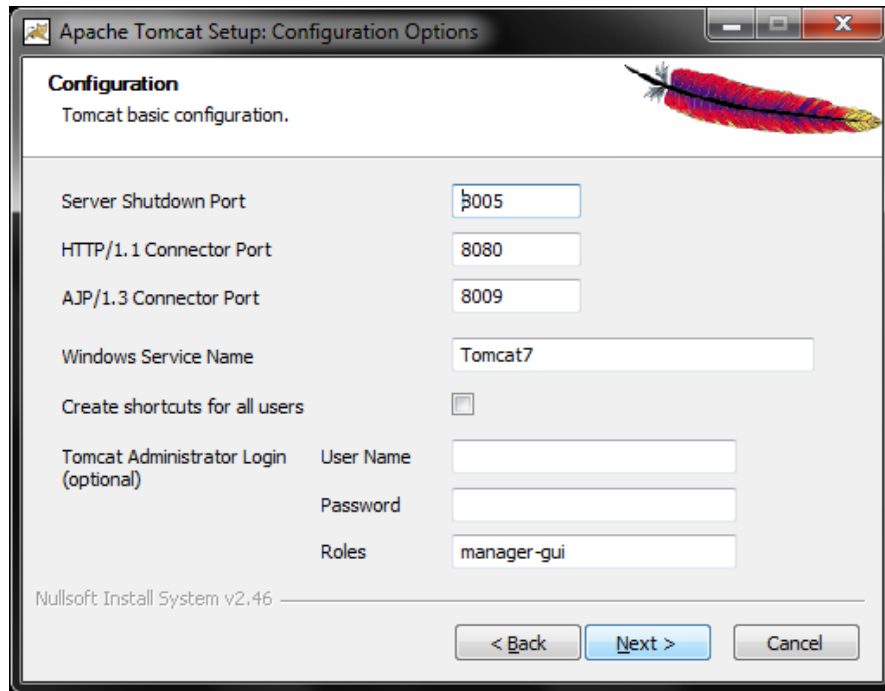




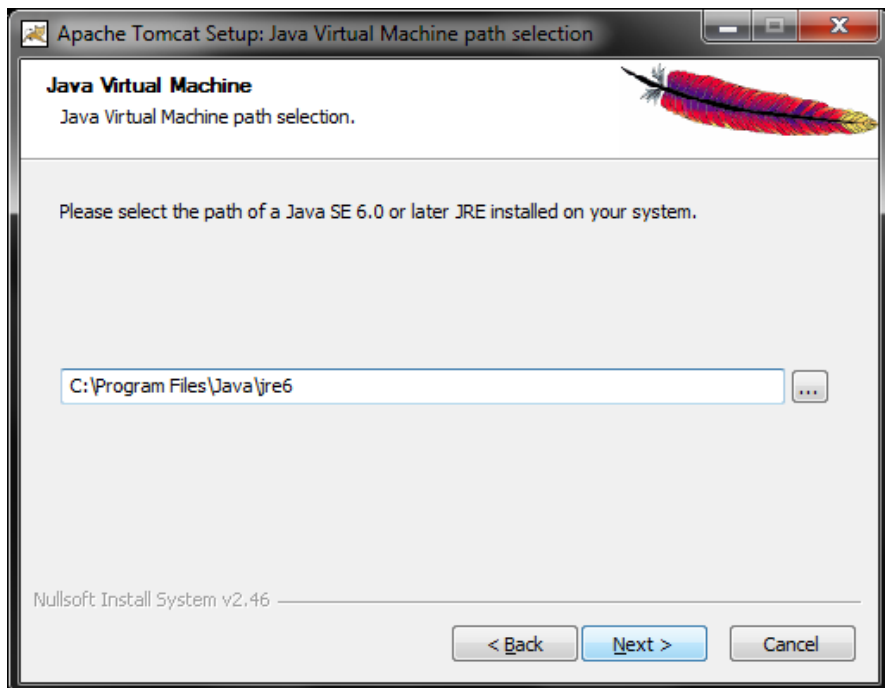
Aceptamos los términos y condiciones de uso de apache-tomcat:



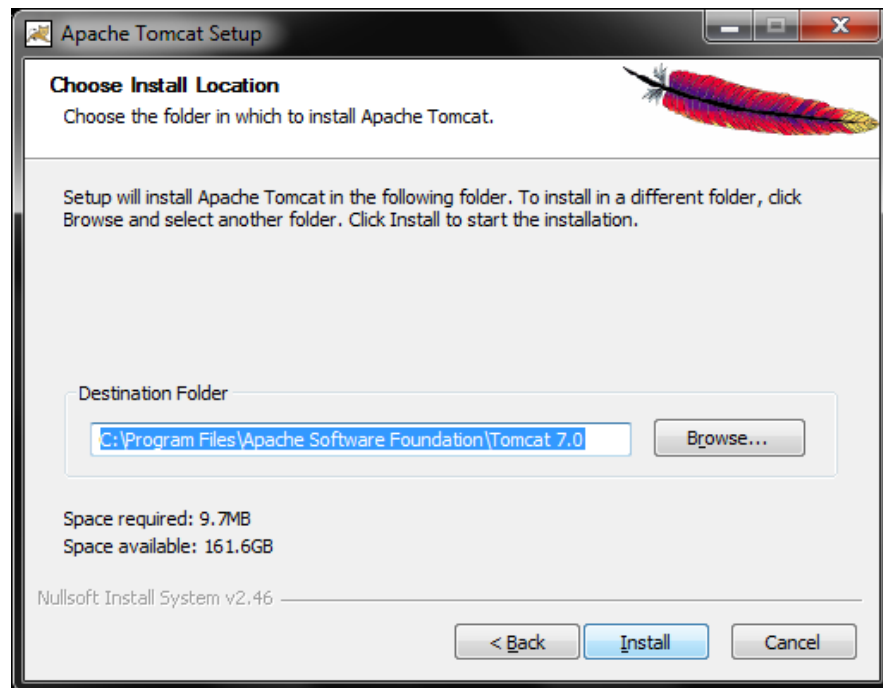
Luego aparecerá una ventana de configuración desde donde configuraremos el puerto de conexión HTTP, nombre del servicio de windows, etc, luego presionamos el botón *next*:



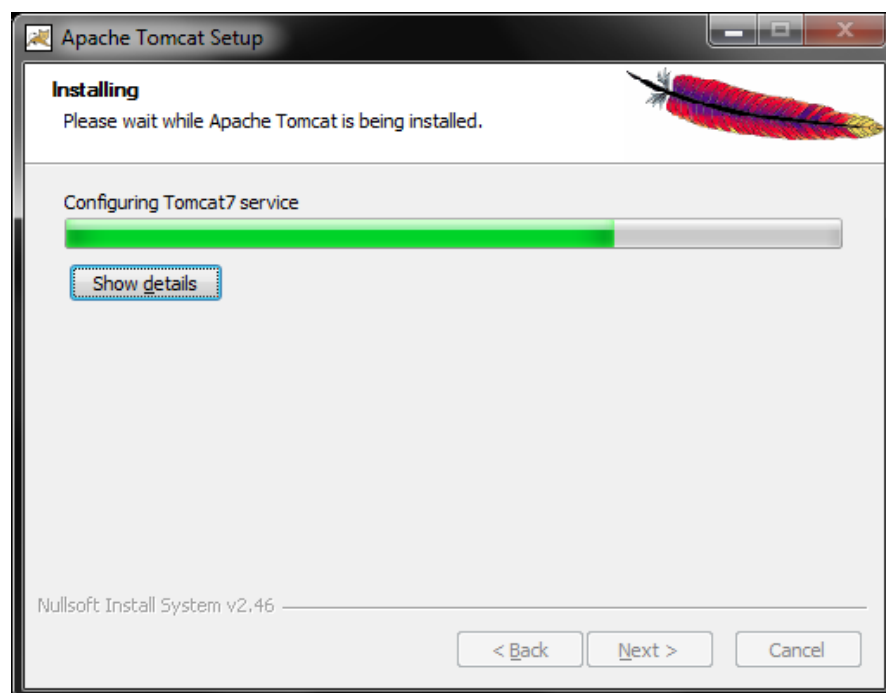
Especificamos el path donde está ubicado el JRE de Java:

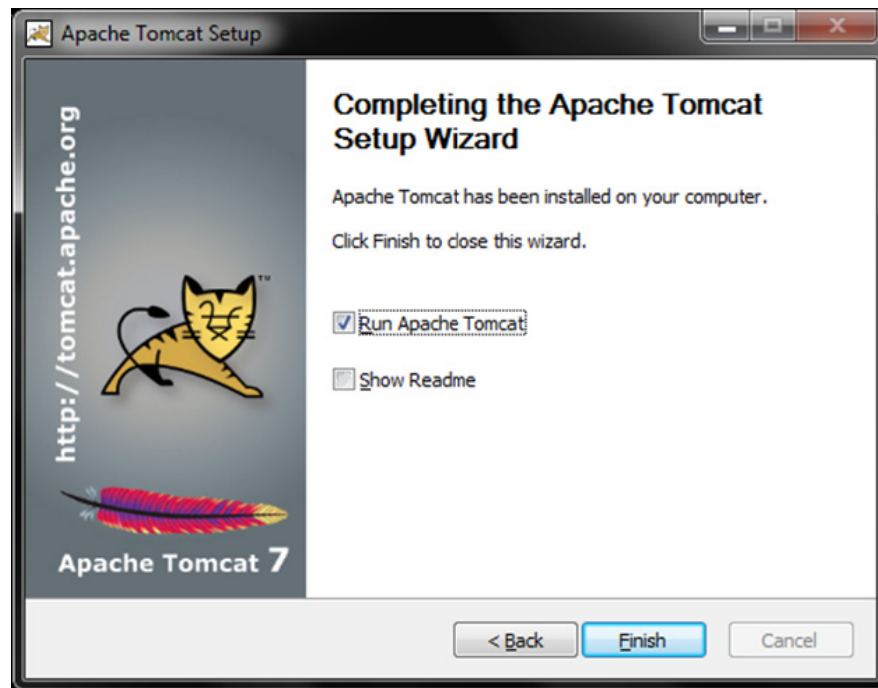


Especificamos el path donde se instalará apache-tomcat:

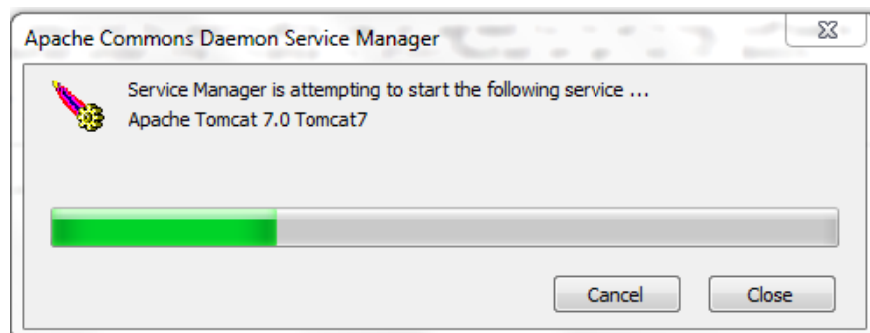



Esperamos a que el proceso de instalación finalice:



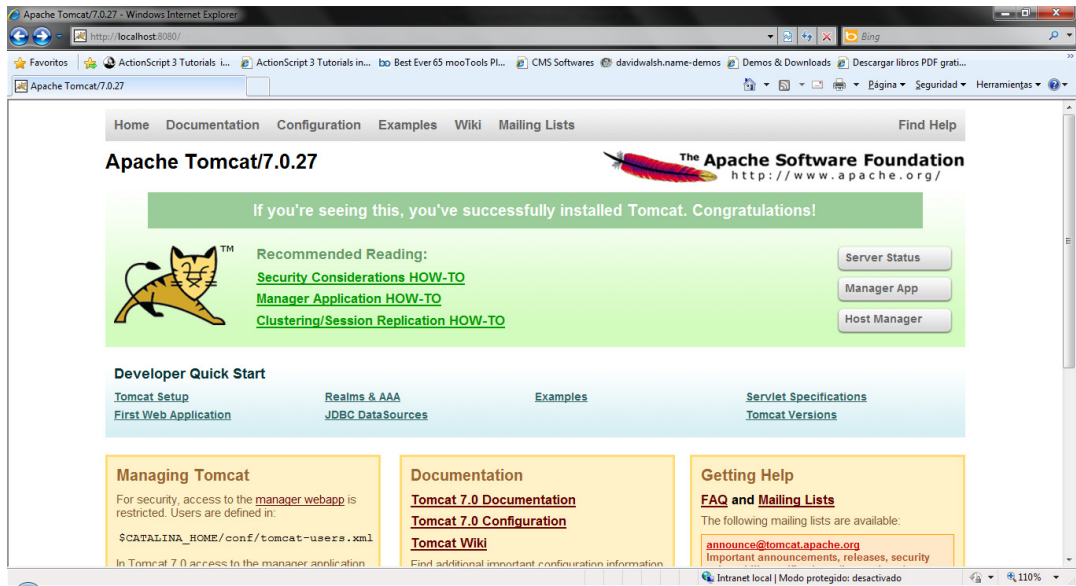


Una vez finalizada la instalación arrancará el servicio automáticamente:



Cuando haya finalizado el arranque del servicio aparecerá en la barra de tareas este ícono , el cual nos indicará el estado del servicio.

Para comprobar si nuestro servidor apache tomcat se está ejecutando correctamente, abriremos una ventana del navegador y en la barra de dirección escribiremos *http://localhost:8080* y nos deberá aparecer la siguiente pantalla:



Listo nuestro servidor web está instalado y funcionando.

2.7. BASES DE DATOS

2.7.1. ¿QUE ES UNA BASE DE DATOS?

Una base de datos o banco de datos es un conjunto de datos relacionados entre sí. Actualmente, y debido al desarrollo tecnológico las bases de datos ofrecen un amplio rango de soluciones al momento de almacenar datos.

Existen programas llamados Sistemas Gestores de Bases de Datos (SGBD) que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Los SGBD se componen de un lenguaje de definición de datos, un lenguaje de manipulación de datos y un lenguaje de consulta.

Existen diferentes tipos de base de datos, entre las más conocidas podemos encontrar las siguientes:

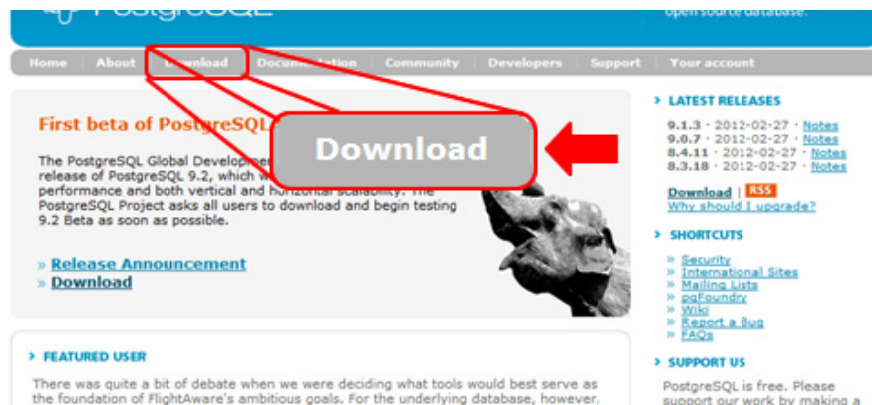
- MySQL: base de datos con licencia GPL basada en un servidor. Se caracteriza por su rapidez. No es recomendable usar para grandes volúmenes de datos.
- PostgreSQL y Oracle: Son bases de datos poderosas. Administran muy bien grandes cantidades de datos, y suelen ser utilizadas en intranets y sistemas de gran tamaño.
- Access: Es una base de datos desarrollada por Microsoft. Está orientada a ser utilizada en un entorno personal o en pequeñas organizaciones. Es muy limitada.
- Microsoft SQL Server: base de datos desarrollada por Microsoft. Se utiliza para manejar grandes volúmenes de informaciones.

Las aplicaciones en las cuales son utilizadas las bases de datos son generalmente para la gestión de datos de empresas (medianas y grandes) e instituciones públicas y privadas, de igual forma son ampliamente utilizadas en entornos científicos con el objetivo de almacenar la información experimental.

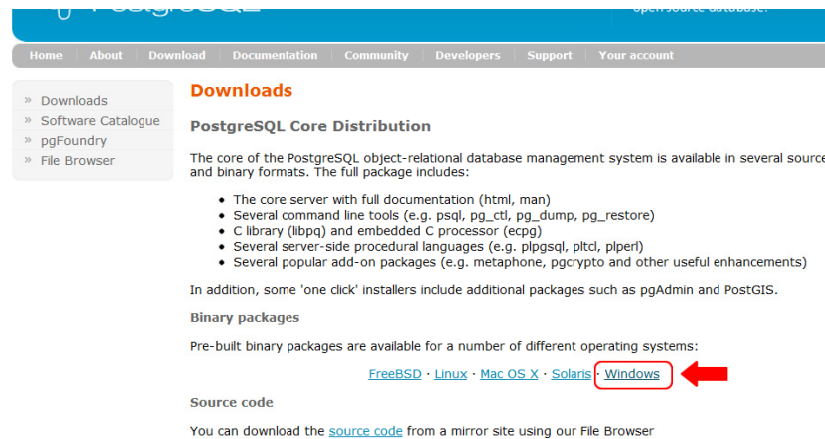
2.7.2. INSTALACIÓN DE POSTGRESQL

Ingresamos a la página oficial de PostgreSQL (www.postgresql.org).

Una vez en la página damos clic en el enlace *DOWNLOAD*:



Dicho enlace nos redirigirá a una nueva página donde seleccionaremos el tipo de sistema operativo para el cual necesitamos el instalador:



Luego aparecerá una nueva página, donde seleccionaremos *DOWNLOAD*:

PostgreSQL open source database.

Home About Download Documentation Community Developers Support Your account

» Downloads
» Software Catalogue
» pgFoundry
» File Browser

Windows packages

The Windows installer for PostgreSQL includes the PostgreSQL server, [pgAdmin III](#); a graphical tool for managing and developing your databases, and StackBuilder; a package manager that can be used to download and install additional PostgreSQL applications and drivers.

One click installer

The one click installer is designed to be as straightforward as possible and the fastest way to get up and running with PostgreSQL on Windows.

- [Download](#) the one click installer

Advanced users can also download a [zip archive](#) of the binaries, without the installer.

The one click installer is maintained by Dave Page at [EnterpriseDB](#).

pgInstaller

pgInstaller is a Windows Installer package that offers advanced configuration options. pgInstaller packages are only being maintained for PostgreSQL 8.3.

- [Download](#) pgInstaller from the win32 subdirectory of the version you require from our file browser.

pgInstaller is maintained by Dave Page and Magnus Hagander.

Después en la siguiente página, seleccionaremos la versión de PostgreSQL y el sistema operativo correspondiente, en nuestro caso Windows x86-32:

PostgreSQL Company

Product, Services & Training Solutions Success Stories Resources & Community Partner Programs About

Download PostgreSQL

Please Note: Cookies should be enabled for the download process to function correctly

Installer version **Version 9.1.3**

Linux x86-32 Linux x86-64 **Win x86-32** Win x86-64 Mac OS X

Installer version **Version 9.0.7**

Linux x86-32 Linux x86-64 Win x86-32 Win x86-64 Mac OS X

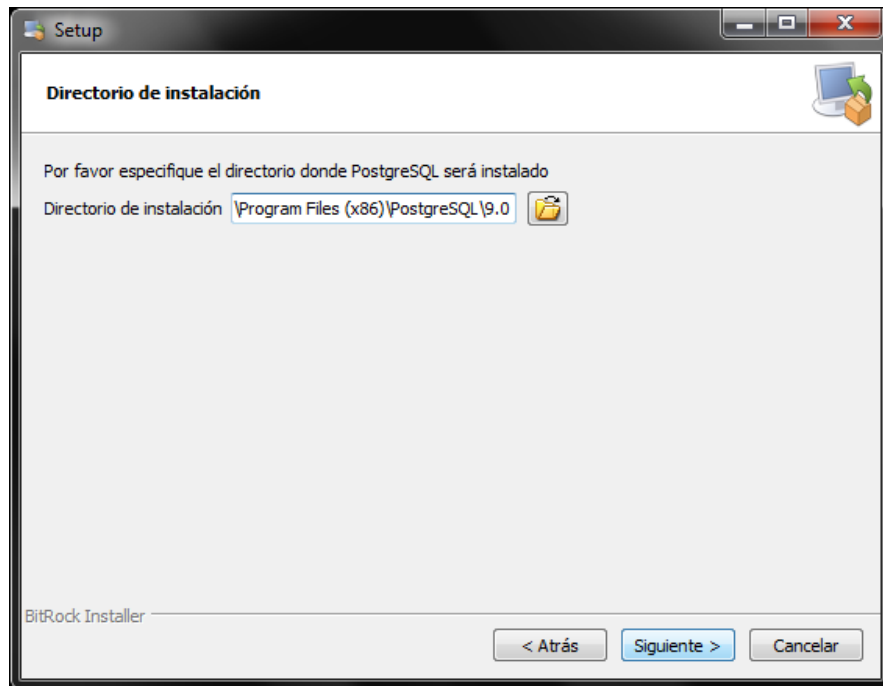
Installer version **Version 8.4.11**

Linux x86-32 Linux x86-64 WINDOWS Mac OS X

Una vez descargado el archivo, le damos doble clic e iniciamos la instalación:



Seleccionamos el path de destino de la instalación:



Luego seleccionamos el destino del directorio de datos:



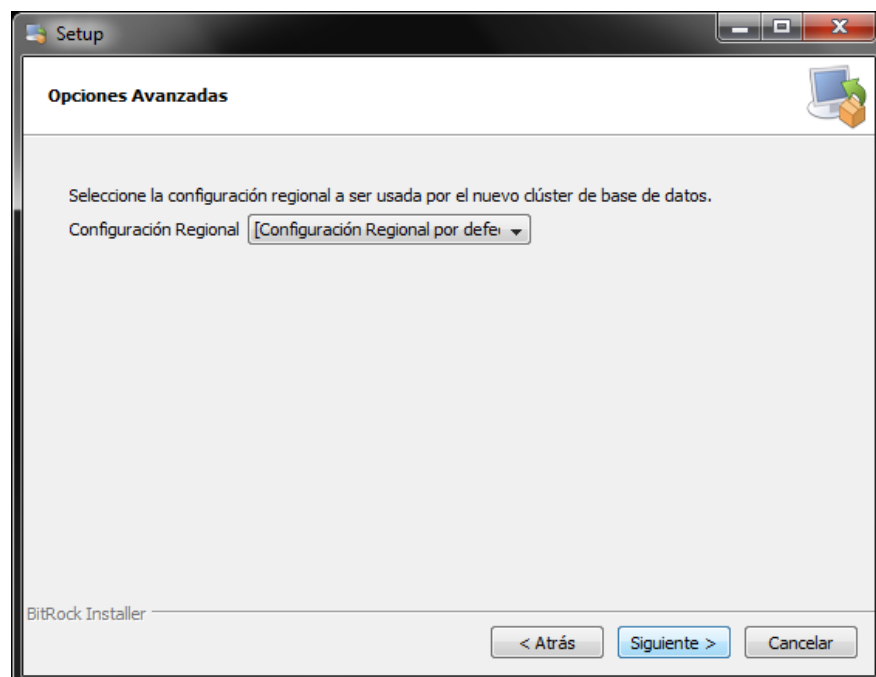
Asignamos una contraseña para el administrador *postgres*:



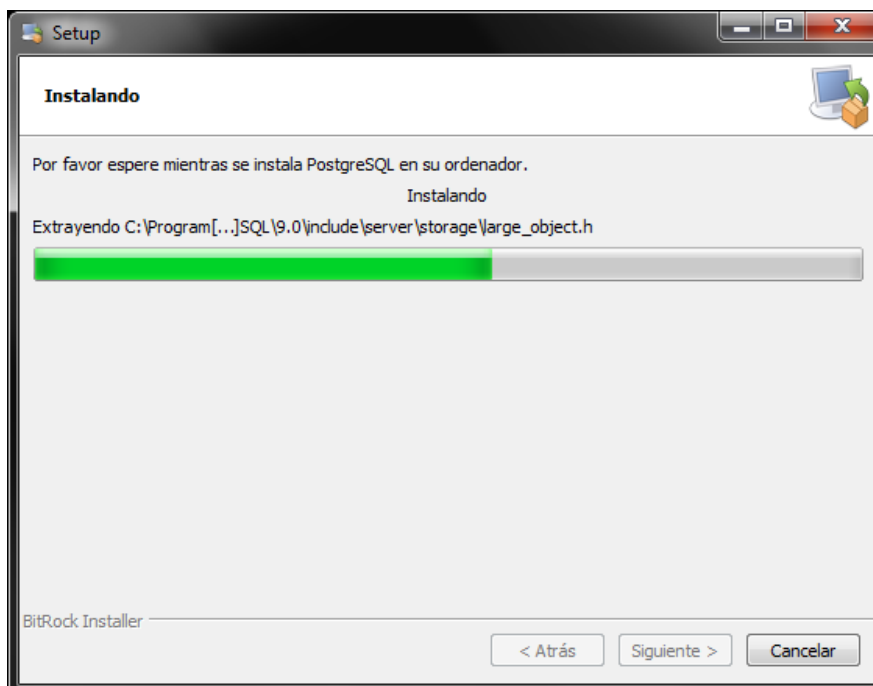
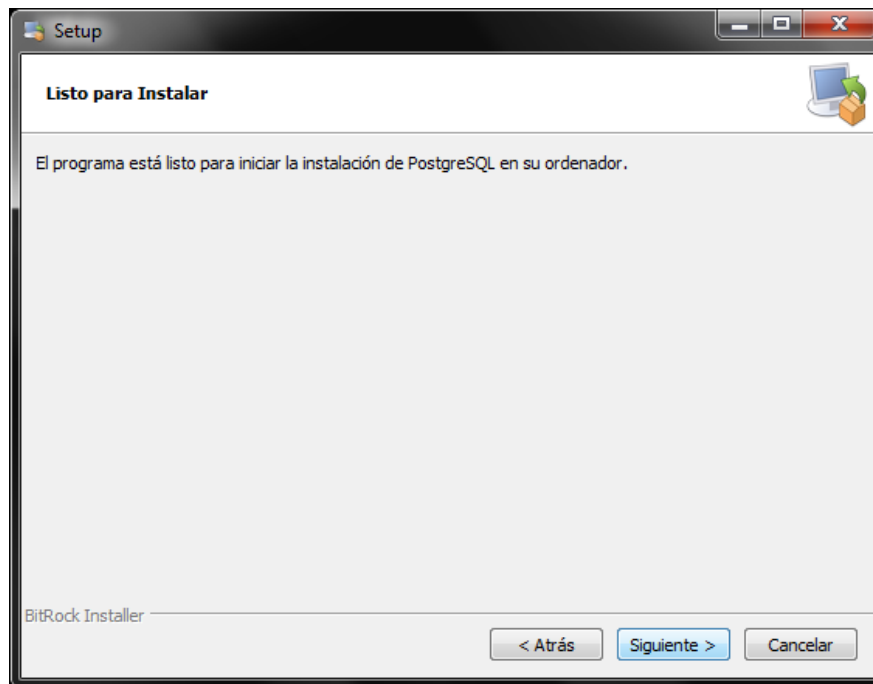
Asignamos el puerto por el que el servidor atenderá las llamadas:



Seleccionamos la configuración regional:



Iniciamos con la instalación:





Listo se ha culminado la instalación de PostgreSQL.

2.8. IREPORT

2.8.1. ¿QUE ES IREPORT?

iReport es un poderosa herramienta OpenSource y basado completamente en Java que facilita la construcción y el diseño de informes de manera visual; es capaz de utilizar datos procedentes de distintas fuentes, permite generar graficas (histogramas, grafico de burbujas, gráficos lineales, etc.) gracias a que integra JFreeChart, además es posible imprimir o exportar los reportes en varios formatos (HTML, PDF, Excel y Word).

2.8.2. CARACTERÍSTICAS DE IREPORT

- Escrito en JAVA
- OPENSOURCE
- Soporta JDBC.
- Soporta JavaBeans como orígenes de datos
- Incluye asistente para generar reportes y subreportes
- Facilidad de instalación.

2.8.3. INSTALACION IREPORT PLUGIN PARA NETBEANS

Requisitos:







- Netbeans 7.0.1
- iReport Plugin NetBeans v4.5.1

Para comenzar, ingresamos a la dirección:

<http://sourceforge.net/projects/ireport/files/iReport%20Plugin%20for%20NetBeans/iReport-4.5.1/> y elegimos *iReport-4.5.1*:

Looking for the latest version? [Download iReport-4.7.0-windows-installer.exe \(99.0 MB\)](#)



Home / iReport Plugin for NetBeans

Name	Modified	Size	Downloads
↑ Parent folder			
📁 iReport-4.7.0	2012-07-30		
📁 iReport-nb-4.6.0	2012-05-21		
📁 iReport-4.5.1	2012-03-06		
📁 iReport-4.5.0	2011-12-09		
📁 iReport-4.1.3	2011-10-30		
📁 iReport-4.1.2	2011-09-16		

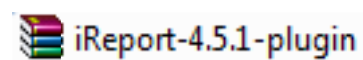
Luego damos clic en el archivo iReport-4.5.1-plugin.zip para su descarga:

Looking for the latest version? [Download iReport-4.7.0-windows-installer.exe \(99.0 MB\)](#)

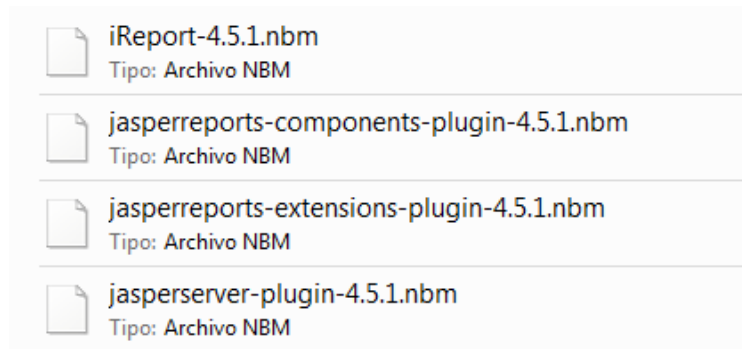
Home / iReport Plugin for NetBeans / iReport-4.5.1

Name	Modified	Size	Downloads
↑ Parent folder			
iReport-4.5.1-plugin.zip	2012-03-06	66.6 MB	21  
Totals: 1 Item		66.6 MB	21

Una vez descargado el archivo lo descomprimos:



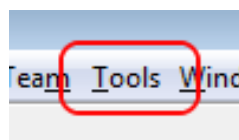
Y tendremos los siguientes archivos:



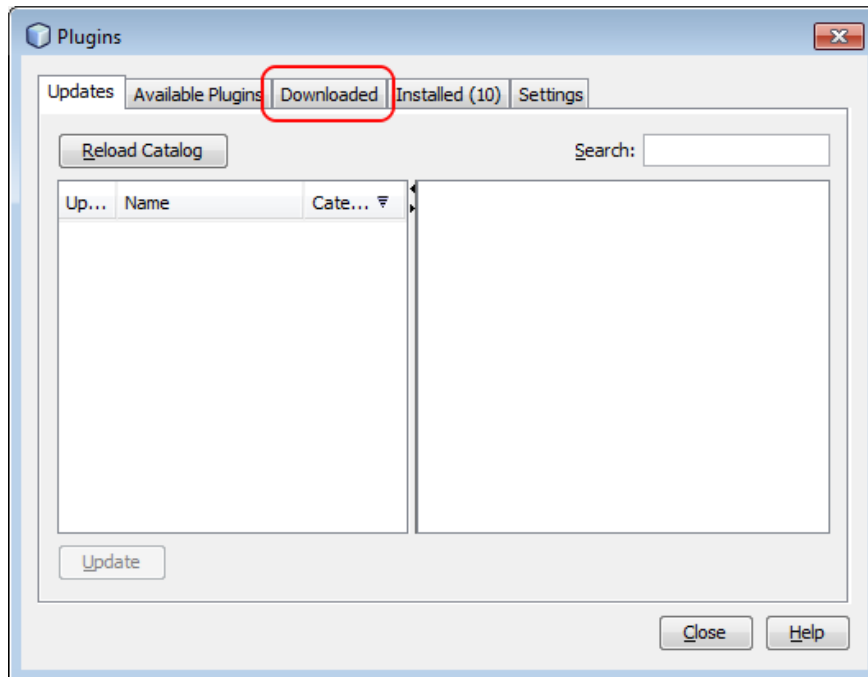
Ahora iniciaremos el Netbeans para iniciar con la instalación del plugin:



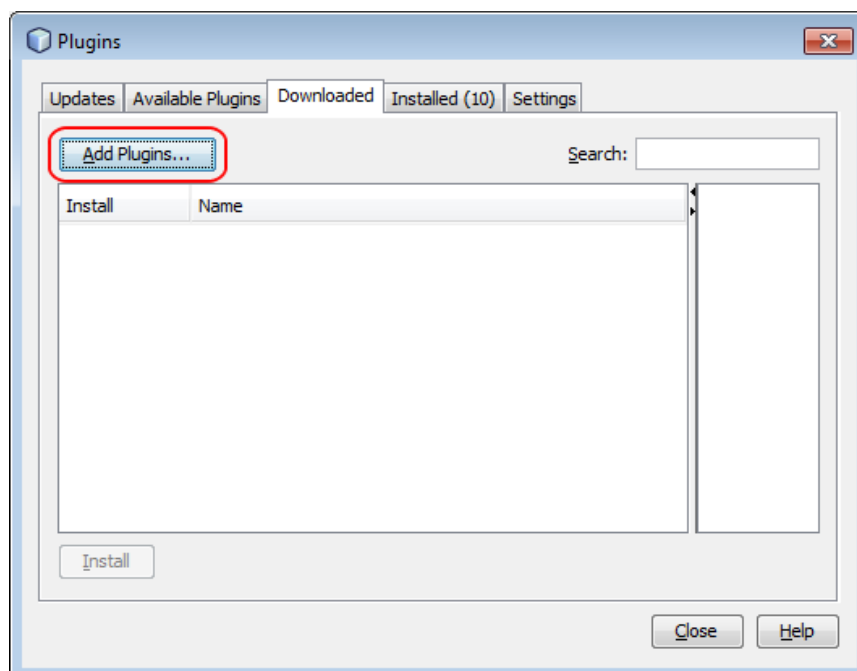
Una vez iniciado el Netbeans, nos dirigiremos hacia la barra de herramientas y seleccionamos la opción *Tools*:



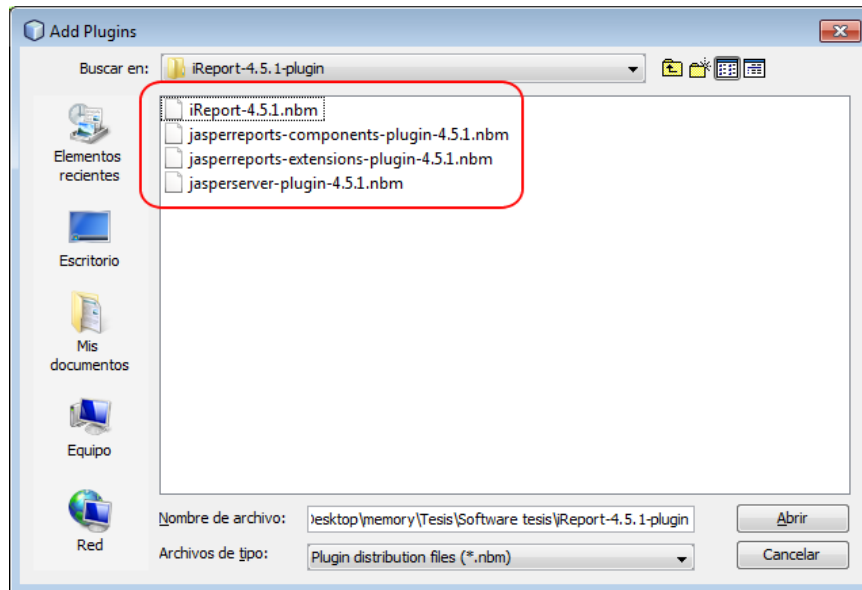
Se nos abrirá una nueva ventana:



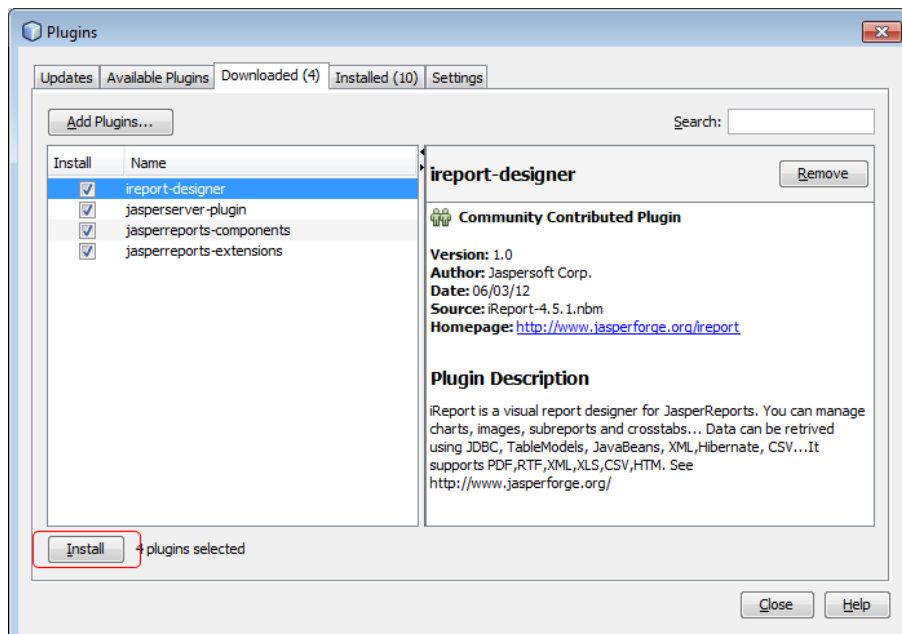
Luego seleccionamos la opción *Add Plugins...*:



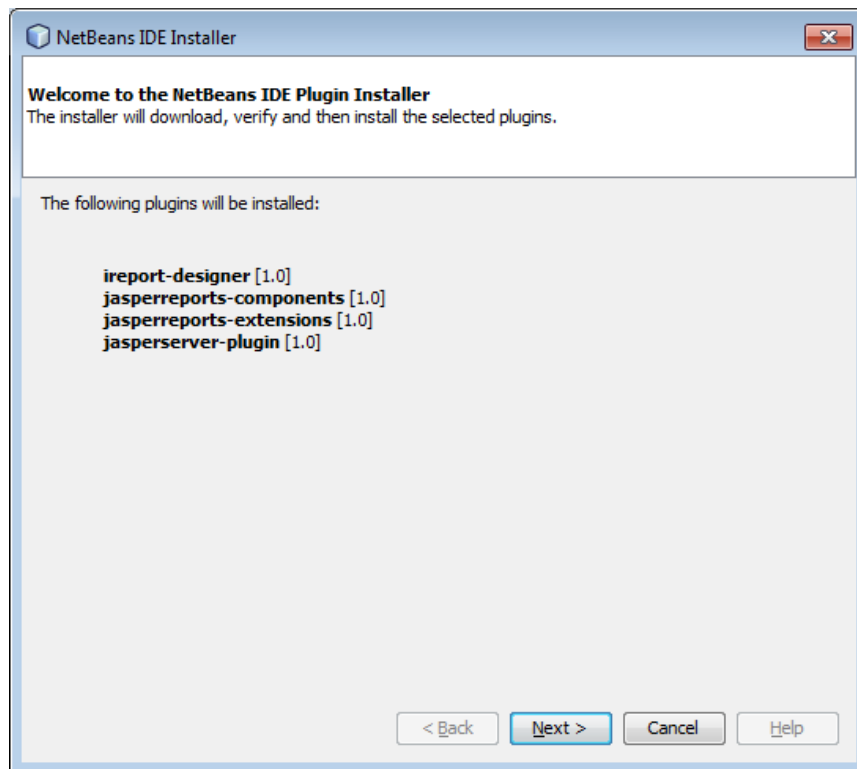
Se nos abrirá un cuadro de dialogo, nos ubicaremos dentro de la carpeta que contiene todos los archivos del plugin:



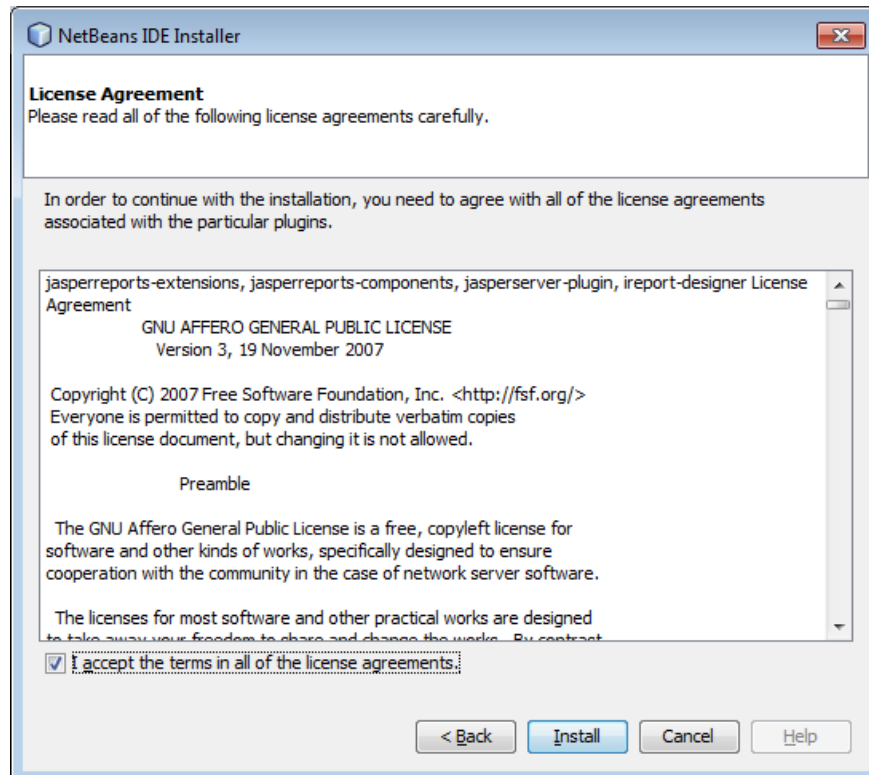
Seleccionamos todos los archivos y damos un clic en *Abrir*:

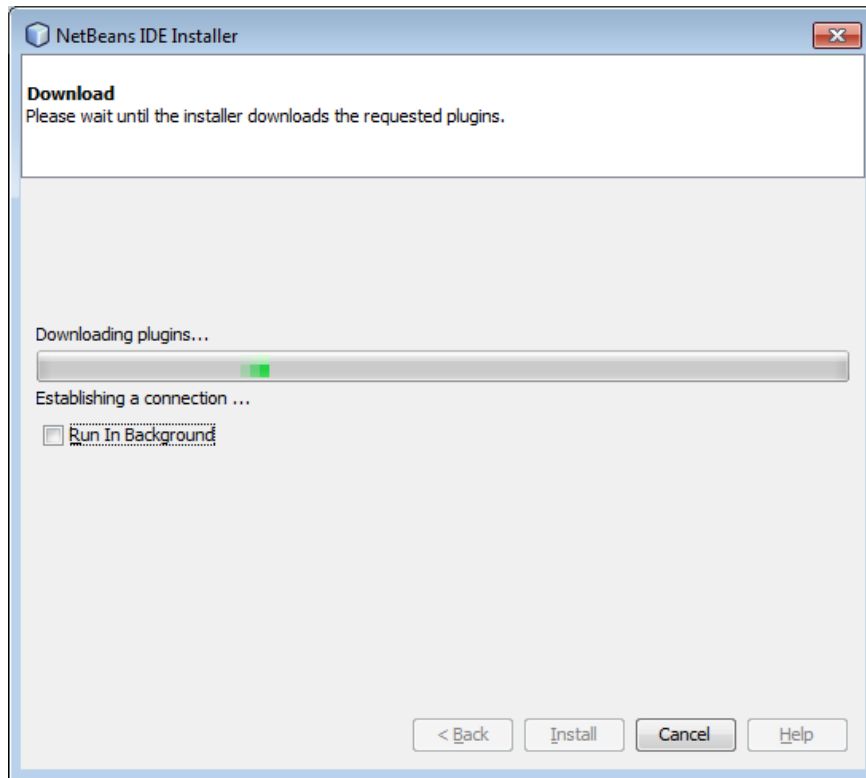


Damos un clic en *Next*:

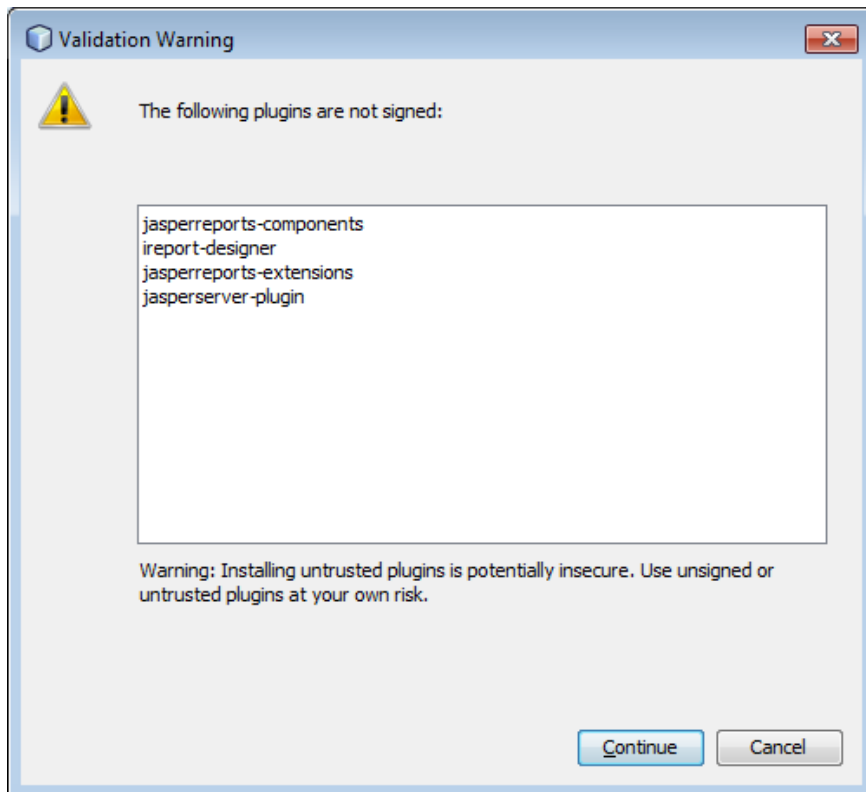


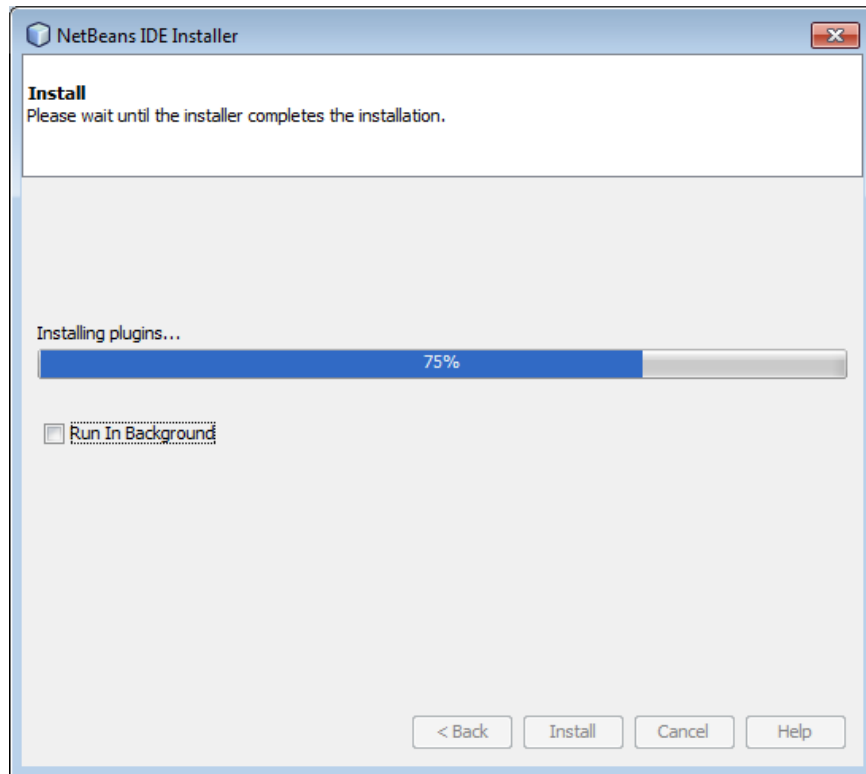
Acceptamos los términos y damos clic en *Install*:



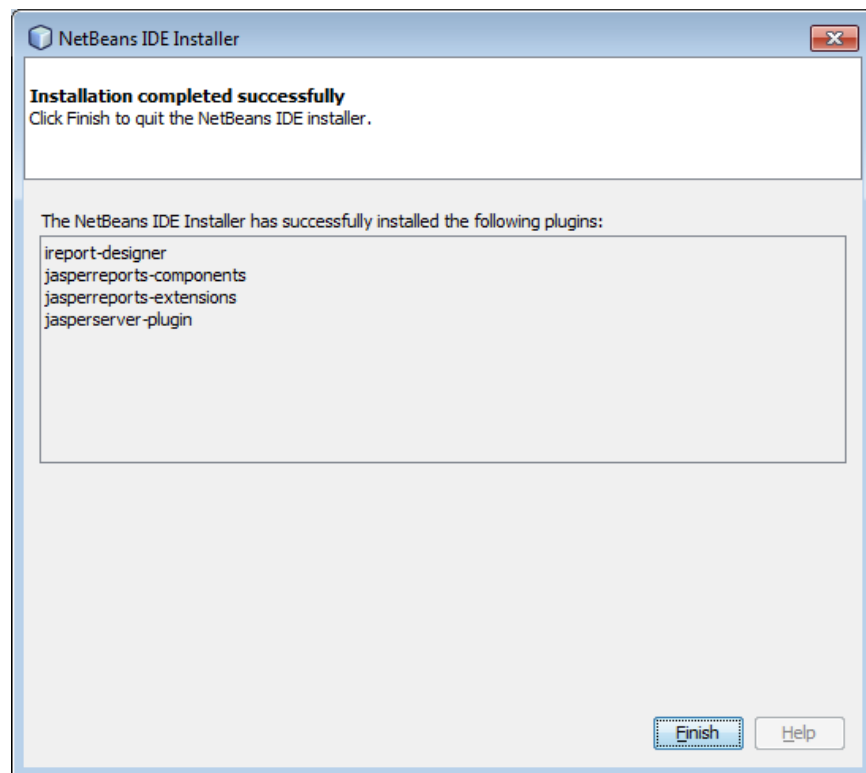


Damos un clic en *Continue*:

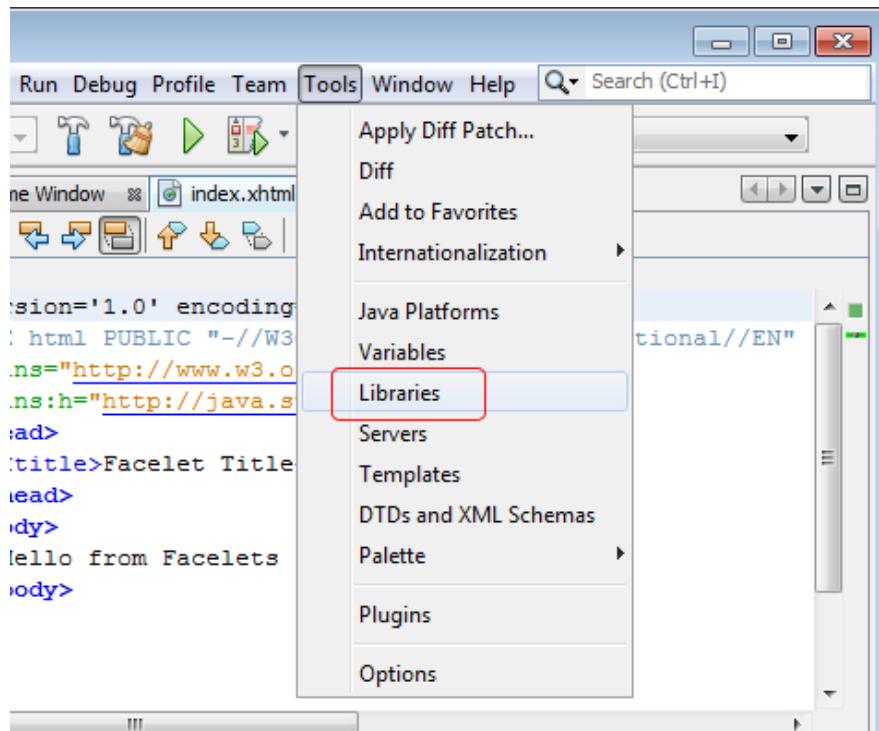




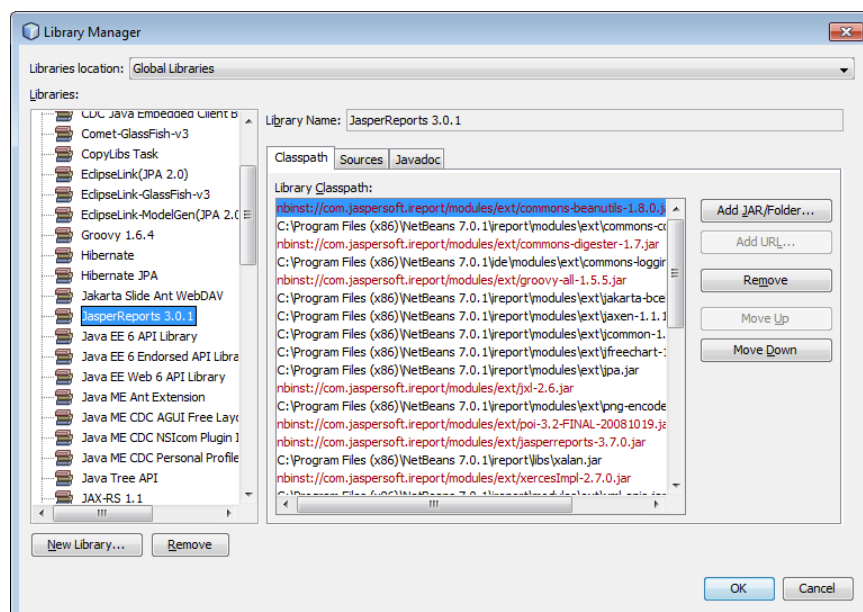
Esperamos hasta que la aplicación se instale y damos clic en *Finish*:



Ahora nos toca hacer algunas modificaciones en la librería de iReport instalada, para ello nos dirigiremos al menú *Tools* y luego escogemos *Libraries*:



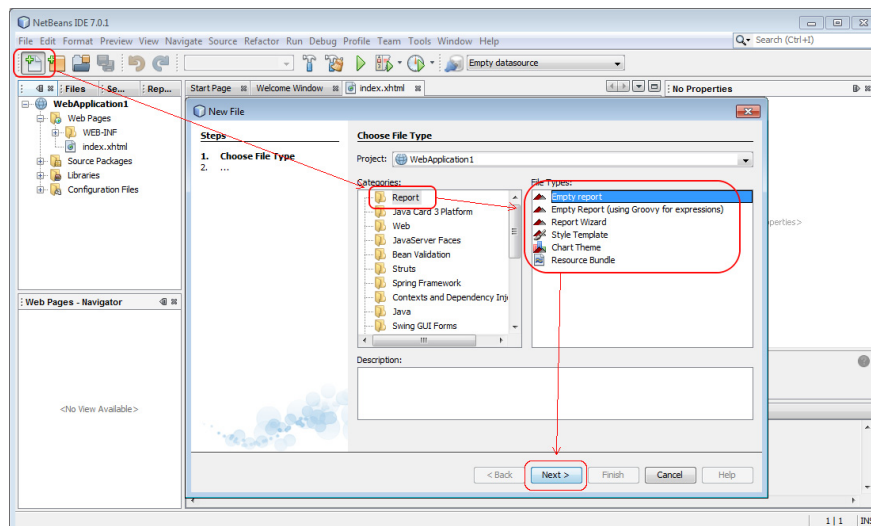
Se nos abrirá una ventana en donde buscaremos y seleccionaremos en *Libraries* la librería *JasperReports 3.0.1*:



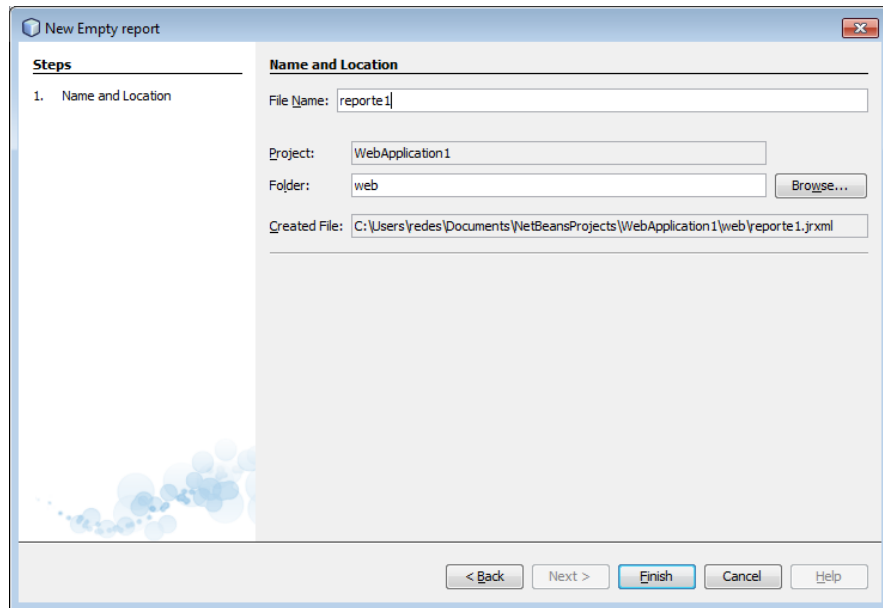
Como se puede observar en la imagen anterior, existen archivos en rojo, esto quiere decir que la ubicación es errónea o los archivos no están disponibles, para resolver este problema de referencia, debemos eliminar los archivos que están en rojo y volverlos a cargar.

El path donde están ubicados los archivos sería: *C:\Program Files\NetBeans 7.0.1\ireport\modules\ext*.

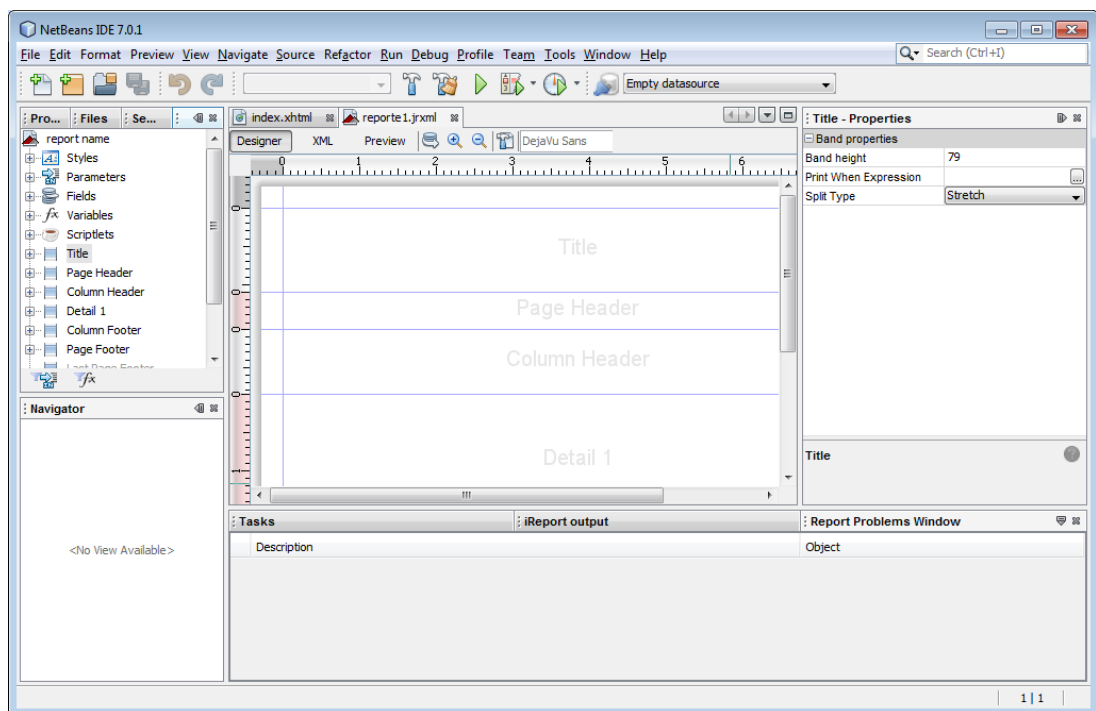
Listo ahora solo nos quedaría agregar a nuestro proyecto un nuevo archivo de reporte, para ellos haremos *New File*, seleccionamos la opción *Report* y luego *Empty report* y damos clic en *Next*:



Pondremos el nombre de nuestro reporte y la ubicación o carpeta en donde se ubicara el mismo y damos clic en *Finish*:



Listo, ahora podemos generar nuestros reportes desde Netbeans.



3. CAPÍTULO III – ALCANCE DEL PROYECTO Y FASE DE ANALISIS DE REQUERIMIENTOS

3.1. ALCANCE DEL PROYECTO

El sistema será implementado para la Dirección Regional 2 de la Contraloría General del Estado y contará con los siguientes módulos:

- Módulo de usuarios
- Módulo de empleados
- Módulo de vehículos
- Módulo de productos
- Módulo control médico
- Módulo de cursos de capacitación
- Auditoría del sistema

A continuación se realiza un detalle de cada uno de los módulos.

MÓDULO DE USUARIOS

Este módulo se encargará de la administración del sistema y de los permisos de acceso a las secciones del sistema, en esta parte será posible habilitar o deshabilitar las opciones del sistema, tendrá acceso completo sistema y a todas las opciones de los módulos.

Como parte de la gestión de usuarios es necesario contar con un módulo que sirva para el registro y modificación de usuarios que utilizarán el sistema, que permita realizar cambios y recuperación de contraseñas, la activación y bloqueos de usuarios según sea el caso y para el control de acceso al sistema mediante autenticación.

Este módulo se encargará de verificar si un usuario está habilitado para utilizar el sistema y a que secciones (páginas) del sistema puede ingresar y que opciones del menú tiene acceso.

Para el control de usuarios se manejarán roles y permisos de acceso a las secciones del sistema.

Los roles a manejar son:

- Administrador: usuario sin ninguna restricción.
- Usuario registrado: usuarios con restricciones, tendrá acceso únicamente a las secciones y opciones de menú asignados por el administrador.

Para el registro de usuario será necesaria la siguiente información:

- Nro. de cédula
- Nombre y apellido completos del usuario
- Nombre de usuario
- Contraseña
- Rol de usuario
- Fecha de registro
- Estado del usuario
- Dependencia donde labora

Se deberá crear un script que permita la creación del usuario administrador, y deberá correrá al momento de ingresar los datos principales dentro de la base de datos.

Es necesario que el usuario se encuentre en la nómina de empleados.

El sistema deberá contar con un solo administrador.

Para el nombre de usuario se deberá seguir los siguientes pasos:

1. Deberá contener la primera letra del nombre más el apellido:

Pedro Rodríguez => p + rodriguez = prodriguez

2. Si hubiese otro usuario con el mismo nombre de usuario del paso 1 se agregará la primera letra del segundo apellido al final:

Pedro Rodríguez Ortiz => p + rodriguez + o = prodriguez

3. Si en caso existiera un usuario igual al del paso 2 se agregará la primera letra del segundo nombre al final de la primera letra del primer nombre:

Pedro Oswaldo Rodríguez Ortiz => p + o + rodriguez + o = porodriguez

4. Un caso extraordinario: si en caso existiera un usuario igual al del paso 3, se procederá de igual manera que el paso 3 pero se agregará al final un valor numérico:

Pedro Oswaldo Rodríguez Ortiz => p + o + rodriguez + o + n = porodriguez1
Donde n = número.

La contraseña del nuevo usuario será una que se le asignara el momento de su creación, esta será cambiada por el usuario al momento de ingresar al sistema por primera vez.

La contraseña deberá tener un tamaño mínimo de 8 caracteres, a su vez se obligará a que contenga caracteres especiales y alfanuméricos, deberá estar encriptado mediante MD5 o SHA1 y estar almacenados en la base de datos. Al momento de un cambio de contraseña no se podrá volver a reutilizar contraseñas anteriores sino hasta pasado 3 meses.

Una vez se haya registrado un nuevo usuario se deberá:

- Asignar los módulos a los cuales tendrá acceso dicho usuario.

- Habilitar los permisos sobre los módulos (crear, editar, borrar, listar, sacar reportes).
- Habilitar al usuario.
- Solicitar cambio de contraseña.

MÓDULO DE EMPLEADOS

Es necesario llevar un registro de los empleados que laboran en la institución, para ello se debe contar con los siguientes datos del empleado:

- Código empleado
- Nro. de cédula
- Nombres completos
- Apellidos completos
- Fecha de nacimiento
- Estado civil
- Dirección domiciliaria
- Teléfono (fijo y móvil)
- Dirección de correo electrónico (personal y laboral)
- Dependencia del empleado
- Fecha de ingreso a la institución pública
- Estado del empleado (ACTIVO, ELIMINADO)

Se desea también registrar sobre las cargas familiares de cada empleado, para ello es necesario:

- Tipo de carga familiar
- Nombre de la carga

A su vez se desea tener el historial de los cargos que un empleado ha tenido durante su vida laboral, por ellos será necesaria la siguiente información:

- Fecha de asignación
- Departamento o área de trabajo
- Nombre del cargo
- Dependencia

Los datos que son necesarios validar son:

- Que no existan campos nulos en los campos obligatorios.
- Nro. de cédula correcto
- Fecha de nacimiento no mayor a la fecha actual.
- Fecha de ingreso a institución no mayor a la fecha actual.
- Fecha de asignación de cargo no mayor a fecha de ingreso a la institución.
- Formato de números de teléfono.
- Correos electrónicos.

MÓDULO DE VEHÍCULOS

Para la movilización de ejecutivos, dirigentes y demás empleados ésta entidad tiene a su cargo unidades vehiculares de uso oficial.

Es necesario contar con la siguiente información del automóvil:

- Nro. de placa
- Marca
- Modelo
- Año del vehículo
- Nro. de chasis
- Km actual

- Ubicación del automotor

Como requerimiento para el uso de un vehículo, es obligatorio llenar una solicitud en la cual deberá constar:

- Lugar de solicitud
- Nro. documento de solicitud
- Empleado que solicita
- Estado de la solicitud

A más de los datos mencionados anteriormente, es necesario que el solicitante especifique los lugares de destino, los motivos de mismo y si fuera el caso el grupo de trabajo.

Al momento de crear la orden de movilización se deberán ingresar los siguientes datos para la orden de movilización:

- Nro. de orden
- Lugar de salida
- Lugar de arribo
- Fecha de salida
- Hora
- Kilometraje actual
- Motivo
- Estado de la orden

Al momento de que una movilización culmina y el vehículo reingresa a la instalación se deberá registrar:

- Fecha de llegada
- Hora de llegada

- Kilometraje actual
- Estado de la orden

Es necesario también llevar un registro de los mantenimientos preventivos y/o correctivos de cada uno de los vehículos.

Para ello es necesario registrar:

- Número de factura o referencia.
- Fecha de realización del mantenimiento.
- Kilometraje de mantenimiento del automotor.
- Costo total.

Otros datos primordiales son:

- Tipo de mantenimiento
- Código del producto
- Observación

Datos que necesitan validación:

- Nro. de placa del vehículo

Adicionalmente se deberá emitir alertas al usuario con indicadores que le digan cuando efectuar el siguiente cambio de parte o mantenimiento del vehículo según los datos almacenados.

MÓDULO DE PRODUCTOS

La entidad al disponer de recursos de oficina desea implementar un sistema de inventario para el control de los mismos, esto es necesario para lograr un control de pedidos, faltantes de productos o cantidad disponible.

Para esto es necesario disponer de la siguiente información del producto:

- Categoría
- Producto
- Descripción del producto
- Unidad de medida
- Estado del producto

Para el inventario es necesario:

- Fecha del movimiento.
- Tipo de movimiento (ingreso/egreso).
- Cantidad de ingreso.
- Cantidad de salida.
- Cantidad total.

Los empleados para disponer de un recurso de oficina deberán llenar una solicitud, que deberá constar de los siguientes datos:

- Código de solicitud
- Fecha y hora
- Datos del empleado que solicita

Para el detalle de la solicitud es necesario:

- Nombre del producto.
- Cantidad.
- Unidad de medida.

Al realizar una transacción o movimiento de producto, el sistema deberá calcular todas operaciones de ingreso o egreso, a más de eso deberá poder notificar en caso de que un producto este sobre el nivel bajo de existencia.

Es importante llevar un registro de las bodegas o almacenes según la ubicación de los productos.

Adicionalmente se deberá emitir alertas al usuario con indicadores que le digan cuando efectuar reposiciones de suministros cuando haya un stock mínimo.

MÓDULO DE CONTROL MÉDICO

La Contraloría al contar con su propio departamento médico, ve necesario registrar el historial clínico de sus empleados y de las consultas médicas que en éste se llevan a cabo.

Para el registro del historial médico del empleado es necesaria la siguiente información:

- Tipo de sangre
- Tipos de alergias
- Tipos de enfermedades
- Genero del paciente

En cuanto a las consultas médicas la información a registrar es:

- Nombre del paciente
- Hora de consulta
- Fecha de consulta
- Motivo de la consulta
- Diagnóstico
- Recetas

Cada receta deberá tener el detalle de medicamentos suscritos por el médico,

- Nombre del medicamento
- Cantidad
- Dosis

Los medicamentos también se verán afectados por el inventario de productos, pues el médico podrá revisar si dicho medicamento se encuentra disponible en el stock de la entidad.

Adicionalmente se deberá emitir alertas al usuario con indicadores que le digan cuando efectuar reposiciones de medicamentos cuando haya un stock mínimo.

MÓDULO CURSOS DE CAPACITACION

La Contraloría desea registrar los cursos de capacitación que oferta para llevar un histórico de los mismos y a su vez registrar los cursos que un empleado a tomado o no. Para ello es necesario guardar la siguiente información:

- Donde se ha dictado el curso
- Modalidad del curso (presenciales u online)

A más de ellos será necesario lo siguiente:

- Nombre del curso ofertado
- Tiempo de duración
- Temas que se trataron
- Costo del curso
- Tipo aprobación
- Calificación
- Número de alumnos participantes

- Número de alumnos aprobados y reprobados
- Número de inscritos
- Estado del curso

También es importante registrar los temas que se dictaran en el curso con su correspondiente facilitador:

- Datos del facilitador:
 - Nombre completo.
 - Especialidad.
 - Calificación
- Datos de los temas
 - Nombre del tema.

Se desea llevar también el registro de la calificación general que los alumnos han otorgado al curso y al facilitador.

Los datos para el registro de alumnos son los siguientes:

- Nro. de cédula
- Nombres completos del alumno
- Lugar de residencia
- Dirección domiciliaria
- Teléfono (fijo y móvil)
- Dirección de correo electrónico
- Lugar de trabajo (Nombre de la institución)
- Título profesional
- Tipo de alumno (Si pertenece dentro o fuera de la institución)
- Estado del alumno (REGISTRADO, INSCRITO, APROBADO, REPROBADO)

AUDITORIA DEL SISTEMA

Este módulo permitirá llevar un historial sobre las actividades que se han realizado en cada uno de los módulos; para ello será necesario contar con:

- Fecha de transacción
- Hora de transacción
- Nombre de la tabla
- Nombre del usuario
- Tipo de transacción
- Campo anterior
- Campo nuevo
- IP del equipo donde se realizo la transacción.

Por cuestiones de auditoría y permanencia de información NO se permitirá eliminar cierta información delicada o importante de la base de datos tan solo se cambiará el estado de dicha información y para ello se manejará dos estados: activos e inactivos y se podrá ver y tener acceso únicamente a aquellos datos en el cual su estado esté activo.

El sistema deberá funcionar para las entidades correspondientes a las provincias del Azuay, Morona Santiago y Cañar, el sistema deberá estar en la capacidad de controlar la ubicación de donde se ha tenido acceso, ubicar a los usuarios por su lugar de trabajo, determinar las ubicaciones de todos los bienes.

El sistema lo podrán acceder únicamente los empleados de la institución que se encuentren registrados.

No es un sistema que permite calificar o gestionar la calificación de los cursos y facilitador por alumno, tan solo proporciona información general del mismo.

3.2. ROLES DE USUARIO

ROL	FUNCIÓN
Administrador	<ul style="list-style-type: none"> - Administración roles y privilegios del sistema - Administración de usuarios - Habilitar cambios de contraseñas - Acceso a todos los módulos - Imprimir reportes
Usuario registrado	<ul style="list-style-type: none"> - Acceso a módulos específicos - Imprimir reportes

3.3. FASE DE ANÁLISIS DE REQUERIMIENTOS

3.3.1. ANÁLISIS DE REQUERIMIENTOS DEL USUARIO

3.3.1.1. MÓDULO USUARIOS DEL SISTEMA

Id. Requerimiento	1.1 Crear rol
Descripción	Registro de roles de usuarios
Entradas	<ul style="list-style-type: none"> - Nombre del rol - Descripción
Salidas	Confirmación de rol registrado
Proceso	El sistema despliega el formulario para registro de nuevos roles, se ingresan los datos de entrada y se registra el rol.
Precondiciones	
Postcondiciones	El rol queda registrado y se puede asignar posteriormente a un usuario.
Efectos Colaterales	Se ingresa un nuevo registro a la base de datos con los datos del nuevo rol.
Prioridad	Alta
Rol que lo ejecuta	Administrador

Id. Requerimiento	1.2 Editar rol
Descripción	Editar un rol de usuario
Entradas	- Nombre del rol - Descripción
Salidas	Confirmación de rol actualizado
Proceso	El sistema despliega el formulario para modificar el rol, se ingresan los datos de entrada y se guardan los cambios del rol.
Precondiciones	
Postcondiciones	
Efectos Colaterales	Se actualiza el registro.
Prioridad	Alta
Rol que lo ejecuta	Administrador

Id. Requerimiento	1.3 Eliminar rol
Descripción	Elimina un rol de usuario
Entradas	Nombre del rol
Salidas	Confirmación de rol eliminado
Proceso	El sistema despliega la lista de roles, se selecciona el rol a eliminar, se elige la opción eliminar, se confirma la eliminación del registro.
Precondiciones	El rol no debe estar asignado a ningún usuario.
Postcondiciones	El rol es eliminado de la base de datos.
Efectos Colaterales	Se elimina el registro.
Prioridad	Alta
Rol que lo ejecuta	Administrador

Id. Requerimiento	1.4 Detalles de rol
Descripción	Visualización detalles de roles de usuarios
Entradas	Nombre del rol
Salidas	Detalles del rol
Proceso	El sistema despliega la lista de roles, se selecciona el rol para ver los detalles, se elige la opción ver detalles, se despliega

	detalles del rol.
Precondiciones	
Postcondiciones	
Efectos Colaterales	
Prioridad	Alta
Rol que lo ejecuta	Administrador

Id. Requerimiento	1.5 Asignar permisos a rol
Descripción	Se asigna permisos y tipo de acceso a un rol.
Entradas	Nombre del rol
Salidas	Confirmación de permiso y tipo de acceso registrado
Proceso	El sistema despliega la lista de roles, se elige el rol al cual se le van a asignar los permiso y el tipo de acceso, se elige la opción asignar permisos, se despliega la lista de módulos, se asignan los permisos, se guardan los cambios.
Precondiciones	
Postcondiciones	Se asignan los permisos y el tipo de acceso a los módulos al rol.
Efectos Colaterales	Se actualiza el rol.
Prioridad	Alta
Rol que lo ejecuta	Administrador

Id. Requerimiento	1.6 Quitar permisos a rol
Descripción	Se quita permisos y tipo de acceso a un rol.
Entradas	Nombre del rol
Salidas	Confirmación de permiso y tipo de acceso registrado
Proceso	El sistema despliega la lista de rol, se elije el rol al que se le revocaran los permisos, se elige la opción asignar permisos, se despliega la lista de módulos, se quitan los permiso al módulo el tipo de acceso, se guardan los cambios.
Precondiciones	
Postcondiciones	Se quitan los permisos y el tipo de acceso a los módulo al rol.

Efectos Colaterales	Se actualiza el rol.
Prioridad	Alta
Rol que lo ejecuta	Administrador

Id. Requerimiento	1.7 Crear usuarios
Descripción	Registrar usuarios para que puedan utilizar el sistema
Entradas	<ul style="list-style-type: none"> - nro. de cedula - Primer nombre - Segundo nombre - Apellido paterno - Apellido materno - nombre de usuario - contraseña - fecha de registro - estado de usuario - rol de usuario - dependencia
Salidas	Confirmación de usuario registrado, mostrar los datos de usuario
Proceso	El sistema despliega el formulario para registro de nuevos usuarios, se ingresan los datos de entrada, se genera el nombre de usuario, se le asigna una contraseña temporal , un rol de usuario y la dependencia (lugar de residencia laboral) y se registra el usuario en el sistema
Precondiciones	
Postcondiciones	El usuario queda registrado y puede ingresar posteriormente al sistema
Efectos Colaterales	<ul style="list-style-type: none"> - Se codifica la contraseña del usuario y se establece como temporal. - Se ingresa un nuevo registro a la base de datos con los datos del nuevo usuario.
Prioridad	Alta

Rol que lo ejecuta	Administrador
---------------------------	---------------

Id. Requerimiento	1.8 Bloquear un usuario
Descripción	El administrador podrá bloquear un usuario para impedir el acceso al sistema
Entradas	Login del usuario administrador
Salidas	Confirmación de usuario bloqueado
Proceso	El administrador inicia una sesión al sistema, busca dentro de la lista de usuarios al usuario que se le desea bloquear, se cambia el estado del usuario a BLOQUEADO, el sistema bloquea al usuario.
Precondiciones	El usuario debe estar registrado en el sistema
Postcondiciones	El usuario no puede ingresar al sistema
Efectos Colaterales	Se modifica el registro de la base de datos con el cambio de estado del usuario.
Prioridad	Alta
Rol que lo ejecuta	Administrador

Id. Requerimiento	1.9 Eliminar un usuario
Descripción	El administrador podrá eliminar un usuario para impedir el acceso al sistema
Entradas	Login del usuario administrador
Salidas	Confirmación de usuario eliminado
Proceso	El administrador inicia una sesión al sistema, busca dentro de la lista de usuarios al usuario que desea eliminar, se cambia el estado del usuario ELIMINADO, el sistema bloquea al usuario.
Precondiciones	El usuario debe estar registrado en el sistema
Postcondiciones	El usuario no puede ingresar al sistema
Efectos Colaterales	Se modifica el registro de la base de datos con el cambio de estado del usuario.
Prioridad	Alta

Rol que lo ejecuta	Administrador
---------------------------	---------------

Id. Requerimiento	1.10 Desbloquear un usuario
Descripción	El administrador podrá desbloquear un usuario que se encuentre bloqueado.
Entradas	Login del usuario administrador
Salidas	Confirmación de usuario desbloqueado
Proceso	El administrador inicia una sesión al sistema, busca dentro de la lista de usuarios al usuario que se desea desbloquear, se cambia el estado del usuario, el sistema desbloquea al usuario.
Precondiciones	El usuario debe estar registrado en el sistema El usuario debe estar bloqueado El usuario no puede estar eliminado
Postcondiciones	El usuario puede ingresar al sistema
Efectos Colaterales	Se modifica el registro de la base de datos con el cambio de estado del usuario.
Prioridad	Alta
Rol que lo ejecuta	Administrador

Id. Requerimiento	1.11 Habilitar cambio de contraseña a un usuario
Descripción	El administrador podrá habilitar la opción de cambio de contraseña a un usuario que solicite cambio de contraseña.
Entradas	- Login del usuario administrador - Contraseña temporal
Salidas	Confirmación de cambio de contraseña habilitado.
Proceso	El administrador inicia sesión en el sistema, busca dentro de la lista de usuarios al usuario que se desea habilitar el cambio de contraseña, se le asigna una contraseña temporal, sistema cambia el estado el estado de la contraseña al usuario.
Precondiciones	El usuario debe estar registrado en el sistema El usuario debe estar activo

Postcondiciones	El usuario puede ingresar al sistema
Efectos Colaterales	Se modifica el estado de la contraseña antigua a CADUCADA.
Prioridad	Alta
Rol que lo ejecuta	Administrador

Id. Requerimiento	1.12 Autenticación Usuario
Descripción	El usuario accede al sistema mediante el ingreso de un nombre de usuario y contraseña, luego de una autenticación positiva el sistema mostrar la interfaz de usuario correspondiente a su rol.
Entradas	<ul style="list-style-type: none"> - Nombre de usuario - Contraseña
Salidas	Interfaz de usuario según el rol asignado
Proceso	El usuario ingresa al formulario de login del sistema, ingresa su nombre de usuario y contraseña, el sistema validará si el usuario existe y si su contraseña es correcta, si todo es correcto se mostrara la interfaz de usuario correspondiente a su rol en el sistema.
Precondiciones	El usuario debe estar creado en el sistema
Postcondiciones	El usuario ingresa al sistema
Efectos Colaterales	<p>El sistema generara una sesión y almacenara la sesión creada junto con los datos del usuario en la BD.</p> <p>Si la contraseña del usuario es incorrecta, se volverá a mostrar el formulario de login.</p> <p>Luego de tres intentos de inicio de sesión, se bloqueara al usuario.</p>
Prioridad	Alta
Rol que lo ejecuta	Administrador y usuario registrado

3.3.1.2. MÓDULO EMPLEADOS

Id. Requerimiento	2.1 Registrar nuevo empleado
Descripción	Un usuario registra los datos personales de un empleado.
Entradas	<ul style="list-style-type: none"> - Código de empleado - Nro. de cédula - Primer nombre - Segundo nombre - Apellido paterno - Apellido materno - Fecha de nacimiento - Estado civil - Dirección domiciliaria - Teléfono fijo - Teléfono móvil - Email personal - Email trabajo - Dependencia (residencia laboral) - Fecha de ingreso a la entidad pública
Salidas	Almacenar datos del empleado, mostrar los datos del empleado
Proceso	El usuario ingresa a la opción REGISTRAR EMPLEADO, llena el formulario con los datos de entrada, el sistema valida los campos y registra en la base de datos.
Precondiciones	El usuario debe haber iniciado sesión en el sistema Debe tener privilegios de creación empleado sobre el módulo de empleados
Postcondiciones	
Efectos Colaterales	Si el nro. de cédula no es válida, se muestra mensaje de error. Si uno o más campos obligatorios son nulos, se muestra mensaje de error.
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	2.2 Registrar Carga familiar
Descripción	El usuario registra las cargas familiares de un empleado.
Entradas	<ul style="list-style-type: none"> - Tipo de carga - Nombre y apellido de la carga
Salidas	Confirmación de carga familiar registrada, mostrar los datos de empleado
Proceso	El usuario busca o selecciona el empleado por numero de cédula, selecciona la opción AGREGAR CARGA FAMILIAR, llena el formulario de carga familiar con los datos de entrada, el sistema valida los campos y registra en la base de datos.
Precondiciones	El usuario debe haber iniciado sesión en el sistema Debe tener privilegios crear carga familiar sobre el módulo de empleados
Postcondiciones	
Efectos Colaterales	Si uno o más campos obligatorios son nulos, se muestra mensaje de error.
Prioridad	Media
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	2.3 Registrar Cargo de empleado
Descripción	El usuario registra los cargos de un empleado.
Entradas	<ul style="list-style-type: none"> - Tipo de cargo - Departamento - Fecha de asignación
Salidas	Confirmación de cargo de empleado registrado, mostrar los datos de empleado
Proceso	El usuario busca o selecciona el empleado por numero de cédula, selecciona la opción AGREGAR CARGO EMPLEADO, llena el formulario de cargos empleado con los datos de entrada, el sistema valida los campos y registra en la

	base de datos.
Precondiciones	El usuario debe haber iniciado sesión en el sistema Debe tener privilegios crear cargo sobre el módulo de empleados
Postcondiciones	
Efectos Colaterales	Si uno o más campos obligatorios son nulos, se muestra mensaje de error.
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	2.4 Modificar datos empleado
Descripción	El usuario modifica datos de un empleado.
Entradas	<ul style="list-style-type: none"> - Código de empleado - Nro. de cédula - Primer nombre - Segundo nombre - Apellido paterno - Apellido materno - Fecha de nacimiento - Estado civil - Dirección domiciliaria - Teléfono fijo - Teléfono móvil - Email personal - Email trabajo - Dependencia (residencia laboral) - Fecha de ingreso a la entidad pública
Salidas	Almacenar datos del empleado modificado, mostrar los datos de empleado
Proceso	El usuario busca o selecciona el empleado por número de cédula, selecciona la opción EDITAR EMPLEADO, modifica el formulario de datos de empleado con los datos de entrada, el sistema valida los campos y registra en la base de datos.

Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios de modificación de empleado sobre el módulo de empleados.
Postcondiciones	
Efectos Colaterales	Si uno o más campos obligatorios son nulos, se muestra mensaje de error.
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	2.5 Modificar carga familiar empleado
Descripción	El usuario modifica datos de carga familiar de un empleado.
Entradas	<ul style="list-style-type: none"> - Tipo de carga - Nombre y apellido de la carga
Salidas	Confirmación de carga familiar modificada de empleado, mostrar los datos de empleado.
Proceso	El usuario busca o selecciona el empleado, selecciona la opción EDITAR CARGA FAMILIAR, modifica el formulario de carga familiar con los datos de entrada, el sistema valida los campos y registra en la base de datos.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios de modificación de carga familiar sobre el módulo de empleados.
Postcondiciones	
Efectos Colaterales	Si uno o más campos obligatorios son nulos, se muestra mensaje de error.
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	2.6 Modificar cargo empleado
Descripción	El usuario modifica el cargo de un empleado.
Entradas	<ul style="list-style-type: none"> - Tipo de cargo - Departamento

	- Fecha de asignación
Salidas	Confirmación de cargo modificado de empleado, mostrar los datos de empleado.
Proceso	El usuario busca o selecciona el empleado, selecciona la opción EDITAR CARGO, modifica los datos del formulario de cargo con los datos de entrada, el sistema valida los campos y registra en la base de datos.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios de modificación de cargos sobre el módulo de empleados.
Postcondiciones	
Efectos Colaterales	Si uno o más campos obligatorios son nulos, se muestra mensaje de error.
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	2.7 Dar de baja empleado
Descripción	El usuario elimina registro de un empleado.
Entradas	- Nro. de cédula
Salidas	Confirmación de empleado eliminado, mostrar página de listado de empleado.
Proceso	El usuario busca o selecciona el empleado por su número de cédula, selecciona la opción ELIMINAR, el sistema actualiza estado de empleado a eliminado.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios de eliminación de empleados sobre el módulo de empleados.
Postcondiciones	
Efectos Colaterales	
Prioridad	Baja
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	2.8 Restablecer empleado
Descripción	Se restablece un empleado que ha sido eliminado.
Entradas	Empleado
Salidas	Confirmación de empleado actualizado
Proceso	El usuario busca o selecciona el empleado por su número de cédula, selecciona la opción RESTABLECER, el sistema actualiza estado de empleado a activo.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios para restablecer un empleado eliminado
Postcondiciones	
Efectos Colaterales	
Prioridad	Alta
Rol que lo ejecuta	Administrador

Id. Requerimiento	2.9 Eliminar carga familiar de empleado
Descripción	El usuario elimina la carga familiar de un empleado.
Entradas	- Nro. de cédula
Salidas	Confirmación de eliminación de carga familiar de empleado, mostrar los datos de empleado.
Proceso	El usuario busca o selecciona el empleado por su número de cédula, selecciona la opción ELIMINAR CARGA FAMILIAR, el sistema actualiza estado de la carga familiar de empleado a eliminado.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios de eliminación de carga familiar sobre el módulo de empleados.
Postcondiciones	

Efectos Colaterales	
Prioridad	Baja
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	2.10 Eliminar cargo empleado
Descripción	El usuario elimina el cargo de un empleado.
Entradas	- Nro. de cédula
Salidas	Confirmación de eliminación de cargo, mostrar los datos de empleado.
Proceso	El usuario busca o selecciona el empleado por su número de cédula, selecciona la opción ELIMINAR CARGO, el sistema actualiza estado de carga a eliminado.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios de eliminación de cargos sobre el módulo de empleados.
Postcondiciones	
Efectos Colaterales	Si uno o más campos obligatorios son nulos, se muestra mensaje de error.
Prioridad	Baja
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	2.11 Imprimir reportes empleado
Descripción	El usuario imprime reportes de empleado.
Entradas	
Salidas	Impresión de reporte.
Proceso	El usuario selecciona la opción REPORTES, selecciona reporte a imprimir, el sistema imprime reporte.
Precondiciones	El usuario debe haber iniciado sesión en el sistema.
Postcondiciones	
Efectos Colaterales	
Prioridad	Alta

Rol que lo ejecuta	Usuario registrado
---------------------------	--------------------

3.3.1.3. MÓDULO VEHÍCULOS

Id. Requerimiento	3.1 Registro de vehículos
Descripción	Registro de vehículos de la empresa
Entradas	<ul style="list-style-type: none"> - Marca - Clase - Tipo - Modelo - Color - Placa del vehículo - Año del vehículo - Nro. de chasis del vehículo - Dependencia a la que pertenece el vehículo
Salidas	Confirmación de registro de vehículo, mostrar datos vehículo.
Proceso	El usuario ingresa a la opción de INGRESO VEHICULO, llena el formulario de registro de vehículo, el sistema valida los datos y almacena en la base de datos.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios de registrar vehículo sobre el módulo de vehículos.
Postcondiciones	
Efectos Colaterales	Si la placa del vehículo no es válida, se muestra mensaje de error. Si uno o más campos obligatorios son nulos, se muestra mensaje de error.
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	3.2 Modificar datos de vehículos
Descripción	Modificación de datos de vehículos de la empresa

Entradas	<ul style="list-style-type: none"> - Marca - Clase - Tipo - Modelo - Color - Placa del vehículo - Año del vehículo - Nro. de chasis del vehículo - Ciudad a la que pertenece el vehículo
Salidas	Confirmación de modificación de vehículo, mostrar datos vehículo.
Proceso	El usuario selecciona o busca por número de placa el vehículo, elige la opción de EDITAR VEHICULO, modifica los datos del formulario de registro de vehículo, el sistema valida los datos y almacena en la base de datos.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios de modificar datos vehículo sobre el módulo de vehículos.
Postcondiciones	
Efectos Colaterales	Si la placa del vehículo no es válida, se muestra mensaje de error. Si uno o más campos obligatorios son nulos, se muestra mensaje de error.
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	3.3 Dar de baja a vehículos
Descripción	Eliminación de datos de vehículos de la empresa
Entradas	<ul style="list-style-type: none"> - Placa del vehículo
Salidas	Confirmación de eliminación de vehículo, mostrar lista de vehículo.
Proceso	El usuario selecciona o busca por número de placa el

	vehículo, elige la opción de ELIMINAR VEHICULO, el sistema actualiza el estado del vehículo a eliminado.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios de eliminación de datos vehículo sobre el módulo de vehículos.
Postcondiciones	
Efectos Colaterales	
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	3.4 Registro solicitud de vehículo
Descripción	Se registra solicitud de un vehículo para movilización de empleados o personal
Entradas	<ul style="list-style-type: none"> - Fecha de solicitud - Hora de solicitud - Datos empleado solicitante - Motivo de solicitud - Destino - Fecha de movilización - Hora de movilización - Equipo de trabajo (opcional) <ul style="list-style-type: none"> - Nro. de cédula - Nombre y apellido
Salidas	Confirmación de solicitud de vehículo, mostrar lista de solicitudes.
Proceso	El usuario accede al formulario de registro de solicitudes de vehículo, llena los datos del formulario (si es necesario registra el equipo de trabajo), el sistema valida los datos y almacena en la base de datos.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios de registro de solicitud de vehículo sobre el módulo de vehículos.
Postcondiciones	

Efectos Colaterales	Si uno o más campos obligatorios son nulos, se muestra mensaje de error.
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	3.5 Registro orden de movilización
Descripción	Se registra la orden de movilización de un vehículo para que se lleve a cabo la movilización del empleado o personal
Entradas	<ul style="list-style-type: none"> - Lugar de salida - Fecha de salida - Hora de salida - Km de salida del vehículo - Vehículo asignado
Salidas	Confirmación de orden de movilización de vehículo, mostrar lista de ordenes de movilización.
Proceso	El usuario accede al formulario de registro de orden de movilización de vehículo, llena los datos del formulario, el sistema valida los datos y almacena en la base de datos.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios de registro de orden de movilización de vehículo sobre el módulo de vehículos.
Postcondiciones	
Efectos Colaterales	Si uno o más campos obligatorios son nulos, se muestra mensaje de error.
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	3.6 Actualización orden de movilización
Descripción	Se actualiza la orden de movilización de un vehículo una vez cumplida la labor.
Entradas	<ul style="list-style-type: none"> - Fecha de llegada - Hora de llegada

	- Km de llegada del vehículo
Salidas	Confirmación de actualización de orden de movilización de vehículo, mostrar lista de ordenes de movilización.
Proceso	El usuario accede a la lista de ordenes de movilizaciones, selecciona o busca la orden de movilización por el numero de orden, elige la opción TERMINAR ORDEN, llena los datos del formulario, el sistema valida los datos y almacena en la base de datos.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios de actualización de orden de movilización de vehículo sobre el módulo de vehículos.
Postcondiciones	
Efectos Colaterales	Si uno o más campos obligatorios son nulos, se muestra mensaje de error.
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	3.7 Registro mantenimiento vehículo
Descripción	El usuario registra los mantenimientos que se ha realizado a los vehículos.
Entradas	<ul style="list-style-type: none"> - Vehículo - Fecha de mantenimiento - Km de mantenimiento - Detalle de mantenimiento: <ul style="list-style-type: none"> - Tipo de mantenimiento - Costo de mantenimiento - Cantidad - Código de producto o repuesto - Observación - Referencia - Km próximo mantenimiento
Salidas	Confirmación de registro de mantenimiento de vehículo, mostrar mantenimientos de vehículo.

Proceso	El usuario ingresa al formulario de registro de mantenimiento, llena el formulario con los datos de entrada, el sistema valida los datos ingresados y los almacena en la base de datos.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios de registro de mantenimiento vehículo sobre el módulo de vehículos.
Postcondiciones	
Efectos Colaterales	Si uno o más campos obligatorios son nulos, se muestra mensaje de error.
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	3.8 Actualización de mantenimiento vehículo
Descripción	El usuario actualiza los datos de los registros de mantenimiento de vehículos.
Entradas	<ul style="list-style-type: none"> - Vehículo - Fecha de mantenimiento - Motivo de mantenimiento - Km de mantenimiento - Detalle de mantenimiento: <ul style="list-style-type: none"> - Tipo de mantenimiento - Costo de mantenimiento - Cantidad - Observación - Referencia - Km próximo mantenimiento
Salidas	Confirmación de registro de mantenimiento de vehículo, mostrar mantenimientos de vehículo.
Proceso	El usuario selecciona o busca el registro por medio de la placa del vehículo, elige la opción de EDITAR, ingresa al formulario de actualización de mantenimiento, llena el formulario con los datos de entrada, el sistema valida los datos ingresados y almacena en la base de datos.

Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios de actualización de registro de mantenimiento vehículo sobre el módulo de vehículos.
Postcondiciones	
Efectos Colaterales	Si uno o más campos obligatorios son nulos, se muestra mensaje de error.
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	3.9 Reporte de solicitud vehículos
Descripción	El usuario imprime un reporte sobre las solicitudes de vehículos.
Entradas	
Salidas	Confirmación de impresión de reporte de vehículo.
Proceso	El usuario elige la opción IMPRIMIR REPORTE, selecciona IMPRIMIR REPORTE SOLICITUDES, el sistema imprime reporte.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios de imprimir reporte sobre el módulo de vehículos.
Postcondiciones	
Efectos Colaterales	
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	3.10 Reporte de movilización vehículos
Descripción	El usuario imprime un reporte sobre las movilizaciones de vehículos.
Entradas	
Salidas	Confirmación de impresión de reporte de vehículo.
Proceso	El usuario elige la opción IMPRIMIR REPORTE, selecciona IMPRIMIR REPORTE MOVILIZACIONES, el sistema

	imprime reporte.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios de imprimir reporte sobre el módulo de vehículos.
Postcondiciones	
Efectos Colaterales	
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	3.11 Reporte de mantenimiento vehículos
Descripción	El usuario imprime un reporte sobre los mantenimientos de vehículos.
Entradas	
Salidas	Confirmación de impresión de reporte de vehículo.
Proceso	El usuario elige la opción IMPRIMIR REPORTE, selecciona IMPRIMIR REPORTE MANTENIMIENTO, el sistema imprime reporte.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios de imprimir reporte sobre el módulo de vehículos.
Postcondiciones	
Efectos Colaterales	
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

3.3.1.4. MÓDULO PRODUCTOS

Id. Requerimiento	4.1 Registro de productos
Descripción	El usuario registra nuevos productos.
Entradas	<ul style="list-style-type: none"> - Categoría de producto - Unidad de medida - Código de producto

	<ul style="list-style-type: none"> - Nombre del producto - Descripción (opcional)
Salidas	Confirmación de registro de producto, mostrar lista de productos.
Proceso	El usuario ingresa al formulario de registro de producto, llena el formulario con los datos de entrada, el sistema valida los datos y almacena en la base de datos.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios de registrar productos sobre el módulo de productos.
Postcondiciones	
Efectos Colaterales	Si uno o más campos obligatorios son nulos, se muestra mensaje de error.
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	4.2 Actualizar datos de productos
Descripción	El usuario actualiza los datos de un producto.
Entradas	<ul style="list-style-type: none"> - Categoría de producto - Unidad de medida - Código de producto - Nombre del producto - Descripción (opcional)
Salidas	Confirmación de registro de producto, mostrar lista de productos.
Proceso	El usuario selecciona o busca el producto, elige la opción EDITAR, modifica los datos del formulario de registro de producto, llena el formulario con los datos de entrada, el sistema valida los datos y almacena en la base de datos.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios de actualizar productos sobre el módulo de productos.
Postcondiciones	

Efectos Colaterales	Si uno o más campos obligatorios son nulos, se muestra mensaje de error.
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	4.3 Registrar producto en bodega
Descripción	El usuario registra un producto en una bodega.
Entradas	<ul style="list-style-type: none"> - Nombre del producto - Ciudad de ubicación de bodega - Cantidad máxima - Cantidad mínima. - Cantidad disponible
Salidas	Confirmación de registro de producto en bodega.
Proceso	El usuario ingresa al formulario de ingreso de producto en bodega, selecciona o busca el producto, elige la opción AGREGAR PRODUCTO, llena el formulario con los datos de entrada, el sistema valida los datos y almacena en la base de datos.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios de agregar productos bodega sobre el módulo de productos.
Postcondiciones	
Efectos Colaterales	Si uno o más campos obligatorios son nulos, se muestra mensaje de error. Si uno o más campos numéricos son menores a 0, mostrar mensaje de error.
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado
Id. Requerimiento	4.4 Registrar movimientos en bodega
Descripción	El usuario registra movimientos de productos en una bodega.
Entradas	<ul style="list-style-type: none"> - Fecha - Concepto - Referencia

	<ul style="list-style-type: none"> - Cantidad - Costo unitario - Tipo de movimiento
Salidas	Confirmación de registro de movimiento de producto en bodega, mostrar lista de movimientos de producto de bodega.
Proceso	El usuario ingresa al formulario de movimiento de producto en bodega, selecciona o busca el producto, elige la opción INGRESO O EGRESO, llena el formulario con los datos de entrada, el sistema valida los datos y almacena en la base de datos.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios de agregar movimiento de producto bodega sobre el módulo de productos.
Postcondiciones	El sistema deberá calcular y actualizar los campos: <ul style="list-style-type: none"> - Valor total - Saldo - Costo unitario de saldo - Valor total del saldo El sistema actualiza el total del producto disponible.
Efectos Colaterales	Si uno o más campos obligatorios son nulos, se muestra mensaje de error. Si uno o más campos numéricos son menores a 0, mostrar mensaje de error.
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	4.5 Registrar de solicitud de productos.
Descripción	El usuario registra solicitudes de productos a bodega.
Entradas	<ul style="list-style-type: none"> - Fecha de solicitud - Hora de solicitud - Total de items - Empleado solicitante - Detalle de items

	- Cantidad
Salidas	Confirmación de solicitud de producto, mostrar lista de solicitudes de producto a bodega.
Proceso	El usuario ingresa al formulario de solicitud de producto, llena el formulario con los datos de entrada, el sistema valida los datos y almacena en la base de datos.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios de agregar solicitudes de producto bodega sobre el módulo de productos.
Postcondiciones	
Efectos Colaterales	Si uno o más campos obligatorios son nulos, se muestra mensaje de error. Si uno o más campos numéricos son menores a 0, mostrar mensaje de error.
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	4.6 Imprimir reportes de productos.
Descripción	El usuario imprime reportes de productos.
Entradas	
Salidas	Confirmación de impresión de reporte de producto
Proceso	El usuario elige la opción reporte de producto.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios de imprimir reporte de producto sobre el módulo de productos.
Postcondiciones	
Efectos Colaterales	
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

3.3.1.5. MÓDULO GESTION MÉDICA

Id. Requerimiento	5.1 Registro historial médico.
Descripción	El usuario registra el historial médico de los empleados.
Entradas	<ul style="list-style-type: none"> - Datos del empleado - Detalle de enfermedad - Detalle de alergias - Tipo de sangre
Salidas	Confirmación de registro de historial médico del empleado
Proceso	El usuario ingresa al formulario de registro de historial, llena los campos con los datos de entrada, el sistema valida los datos y almacena en la base de datos.
Precondiciones	<p>El usuario debe haber iniciado sesión en el sistema.</p> <p>Debe tener privilegios de registrar historial médico sobre el módulo medico.</p> <p>El usuario debe tener el rol de medico.</p>
Postcondiciones	
Efectos Colaterales	
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	5.2 Actualizar historial médico.
Descripción	El usuario modifica el historial médico de los empleados.
Entradas	<ul style="list-style-type: none"> - Datos del empleado - Detalle de enfermedad - Detalle de alergias - Tipo de sangre
Salidas	Confirmación de actualización de historial médico del empleado, mostrar menú principal.
Proceso	El usuario ingresa al formulario de registro de historial, selecciona o busca el historial médico, actualiza los campos con los datos de entrada, el sistema valida los datos y almacena en la base de datos.
Precondiciones	<p>El usuario debe haber iniciado sesión en el sistema.</p> <p>Debe tener privilegios de actualización de historial médico</p>

	sobre el módulo médico. El usuario debe tener el rol de médico.
Postcondiciones	
Efectos Colaterales	
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	5.3 Registrar consulta médico.
Descripción	El usuario registra la consulta médica de un empleado.
Entradas	<ul style="list-style-type: none"> - Datos del empleado - Fecha de consulta - Hora de consulta - Motivo de la consulta - Detalle de la consulta - Prescripción - Diagnóstico - Síntomas
Salidas	Confirmación de registro de consulta médica, mostrar menú principal.
Proceso	El usuario ingresa al formulario de registro de consulta, ingresa los campos con los datos de entrada, el sistema valida los datos y almacena en la base de datos.
Precondiciones	<p>El usuario debe haber iniciado sesión en el sistema.</p> <p>Debe tener privilegios de registrar consulta médica sobre el módulo médico.</p> <p>El usuario debe tener el rol de médico.</p>
Postcondiciones	
Efectos Colaterales	
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	5.4 Imprimir reporte historial médico.
Descripción	El usuario imprime reporte de historial médico.
Entradas	
Salidas	Confirmación de impresión de reporte de historial médico, mostrar menú principal.
Proceso	El usuario selecciona la opción IMPRIMIR REPORTE HISTORIAL MEDICO, el sistema imprime el reporte.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios sobre el módulo medico. El usuario debe tener el rol de medico.
Postcondiciones	
Efectos Colaterales	
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	5.5 Imprimir reporte consulta médico.
Descripción	El usuario imprime reporte de consultas médica.
Entradas	
Salidas	Confirmación de impresión de reporte de consultas médicas, mostrar menú principal.
Proceso	El usuario selecciona la opción IMPRIMIR REPORTE CONSULTA MEDICA, el sistema imprime el reporte.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios sobre el módulo medico. El usuario debe tener el rol de medico.
Postcondiciones	
Efectos Colaterales	
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

3.3.1.6. MÓDULO CURSOS DE CAPACITACION

Id. Requerimiento	6.1 Registro de cursos
Descripción	El usuario registra nuevos cursos.
Entradas	<ul style="list-style-type: none"> - Nombre del curso - Fecha de inicio - Fecha de fin - Número de horas - Lugar donde se dicto el curso - Nombre del coordinador - Modalidad del curso - Costo del curso - Temas del curso - Nombre del facilitadores - Especialidad del facilitador
Salidas	Confirmación de registro de cursos, mostrar lista de cursos.
Proceso	El usuario ingresa al formulario de registro de cursos, llena el formulario con los datos de entrada, el sistema valida los datos y almacena en la base de datos.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios de registrar cursos.
Postcondiciones	
Efectos Colaterales	Si uno o más campos obligatorios son nulos, se muestra mensaje de error.
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	6.2 Modificar cursos
Descripción	El usuario modifica datos de un curso registrado.
Entradas	<ul style="list-style-type: none"> - Nombre del curso - Fecha de inicio - Fecha de fin - Número de horas - Lugar donde se dicto el curso

	<ul style="list-style-type: none"> - Nombre del coordinador - Modalidad del curso - Costo del curso - Temas del curso - Nombre del facilitadores - Especialidad del facilitador
Salidas	Confirmación de actualización de curso, mostrar datos de curso.
Proceso	El usuario ingresa al formulario de actualización de cursos, llena el formulario con los datos de entrada, el sistema valida los datos y almacena en la base de datos.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios sobre el módulo cursos.
Postcondiciones	
Efectos Colaterales	Si uno o más campos obligatorios son nulos, se muestra mensaje de error.
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	6.3 Eliminación cursos
Descripción	El usuario elimina un curso registrado.
Entradas	<ul style="list-style-type: none"> - Nombre del curso
Salidas	Confirmación de eliminación de curso, mostrar lista de cursos.
Proceso	El usuario ingresa a la lista de cursos, selecciona el curso a eliminar, confirma eliminación y se actualiza la base de datos.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios sobre el módulo cursos.
Postcondiciones	
Efectos Colaterales	
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	6.4 Reporte de cursos
Descripción	El usuario imprime un reporte de los curso.
Entradas	
Salidas	Confirmación de impresión de reporte de consultas médicas, mostrar menú principal.
Proceso	El usuario selecciona la opción IMPRIMIR REPORTE CURSOS, el sistema imprime el reporte.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios sobre el módulo cursos.
Postcondiciones	
Efectos Colaterales	
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	6.5 Registro de alumnos
Descripción	El usuario registra alumnos en un curso.
Entradas	<ul style="list-style-type: none"> - Numero de cedula - Nombre del alumno - Lugar de residencia - Dirección - Teléfono fijo - Teléfono móvil - Email - Nombre de la institución a la que pertenece - Especialidad - Tipo de alumno - Estado del alumno
Salidas	Confirmación de registro de alumnos, mostrar lista de alumnos registrados.
Proceso	El usuario ingresa a la lista de cursos, selecciona el curso e ingresa al formulario de registro de alumnos, llena el formulario con los datos de entrada, el sistema valida los

	datos y almacena en la base de datos.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios sobre el módulo cursos.
Postcondiciones	
Efectos Colaterales	Si uno o más campos obligatorios son nulos, se muestra mensaje de error.
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	6.6 Modificar alumnos
Descripción	El usuario modifica datos de un alumno registrado.
Entradas	<ul style="list-style-type: none"> - Numero de cedula - Nombre del alumno - Lugar de residencia - Dirección - Teléfono fijo - Teléfono móvil - Email - Nombre de la institución a la que pertenece - Especialidad - Tipo de alumno - Estado del alumno
Salidas	Confirmación de actualización de alumno, mostrar datos de curso.
Proceso	El usuario ingresa a la lista de cursos, selecciona el curso, ingresa a la lista de alumnos, selecciona el alumno e ingresa al formulario de modificación de alumnos, llena el formulario con los datos de entrada, el sistema valida los datos y almacena en la base de datos.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios sobre el módulo cursos.
Postcondiciones	
Efectos Colaterales	Si uno o más campos obligatorios son nulos, se muestra

	mensaje de error.
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	6.7 Eliminar alumnos
Descripción	El usuario elimina un alumno registrado.
Entradas	- Nombre del alumno
Salidas	Confirmación de eliminación de alumno, mostrar lista de alumnos.
Proceso	El usuario ingresa a la lista de cursos, selecciona el curso, ingresa a la lista de alumnos, selecciona el alumno, confirma eliminación y se actualiza la base de datos.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios sobre el módulo cursos.
Postcondiciones	
Efectos Colaterales	
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

Id. Requerimiento	6.8 Registro de calificación curso
Descripción	El usuario registra la calificación del curso
Entradas	
Salidas	Confirmación de registro de calificación de curso.
Proceso	El usuario ingresa al formulario de registro de calificación de curso, llena el formulario con los datos de entrada, el sistema valida los datos y almacena en la base de datos.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios sobre el módulo de cursos.
Postcondiciones	
Efectos Colaterales	Si uno o más campos obligatorios son nulos, se muestra mensaje de error.
Prioridad	Alta

Rol que lo ejecuta	Usuario registrado
---------------------------	--------------------

Id. Requerimiento	6.9 Reporte de calificación cursos
Descripción	El usuario imprime un reporte de las calificaciones de cursos.
Entradas	
Salidas	Confirmación de impresión de reporte de consultas médicas, mostrar menú principal.
Proceso	El usuario selecciona la opción IMPRIMIR REPORTE CALIFICACION CURSOS, el sistema imprime el reporte.
Precondiciones	El usuario debe haber iniciado sesión en el sistema. Debe tener privilegios sobre el módulo de cursos.
Postcondiciones	
Efectos Colaterales	
Prioridad	Alta
Rol que lo ejecuta	Usuario registrado

3.3.2. ANÁLISIS DE REQUERIMIENTOS NO FUNCIONALES

Disponibilidad:

- El sistema tendrá una funcionalidad de 24/7.

Escalabilidad:

- El sistema debe permitir la reutilización de componentes.
- El sistema debe permitir el desarrollo de nuevas funcionalidades, modificar o eliminar funcionalidades.

Mantenimiento:

- El sistema deberá contar con un manual de funcionamiento.

Plataforma:

- El sistema debe ser escrito en lenguaje JAVA.

Calidad:

- La interfaz de usuario deberá ser lo más clara posible y de fácil uso.

Seguridad:

- El acceso al sistema debe estar restringido por el uso de claves asignadas a cada uno de los usuarios.
- Los usuarios serán clasificados por tipos de usuarios (o roles) con acceso a las opciones de trabajo definidas para cada rol.
- El control de acceso implementado debe permitir asignar los perfiles para cada uno de los roles identificados.
- El sistema deberá filtrar los datos de acuerdo a cada ciudad desde las cuales se va a tener acceso.
- El sistema deberá contar con mecanismos que permitan el registro de actividades con identificación de los cambios realizados.

Compatibilidad:

- El sistema debe ser compatible con la mayoría de los navegadores web.

3.4. REQUERIMIENTOS DE HARDWARE

- Memoria RAM min: 4GB.
- Procesador: Pentium IV o superior.
- Espacio mínimo en disco duro: 10GB

3.5. REQUERIMIENTOS DE SOFTWARE

3.5.1. SERVIDOR DE APLICACIÓN

SOFTWARE	VERSIÓN	VENDEDOR
Sistema Operativo	Windows Server 2003 /2008	Microsoft
Apache Tomcat	7.0.2x	Apache Software Foundation
JRE	1.6.0	SUN

3.5.2.SERVIDOR DE BASES DE DATOS

SOFTWARE	VERSIÓN	VENDEDOR
Motor Base de Datos	PostgreSQL	Versión Libre

3.5.3.ENTORNO DE DESARROLLO

SOFTWARE	VERSIÓN	VENDEDOR
NetBeans	7.0.1	Sun Microsystems
PrimeFaces	3.0 o mayor	
JDK	1.6.0	SUN

4. CAPÍTULO IV – ANALISIS DE RIESGOS

4.1. ANÁLISIS DE RIESGOS

El objetivo del análisis de riesgos es la identificación y control de factores o fuentes que pueden afectar el cumplimiento de los objetivos en el desarrollo del proyecto.

Los riesgos son agentes que se pueden o no presentar y pueden amenazar el plan del proyecto, si estos se logran hacer realidad podrían provocar retrasos y generar pérdidas o aumento de costos tanto de desarrollo como económicos.

PROBABILIDAD	3	3	6	9
	2	2	4	6
	1	1	2	3
		1	2	3
		IMPACTO		

	Aceptable
	Moderado
	Crítico

Tabla 9: Matriz de análisis de probabilidad e impacto.

Todas las entidades tienen su propia forma de gestionar los riesgos e impactos en sus diferentes problemas, en esta ocasión se ha realizado los cálculos mediante el método cardinal que utiliza una calificación numérica, para la elaboración del análisis se siguen los siguientes pasos:

1. Evaluación de la probabilidad del riesgo.
2. Puntuar cada riesgo.
3. Multiplicar la probabilidad por el impacto y obtener la matriz de puntuación.

4.1.1. RIESGOS DEL PRODUCTO

RIESGO	PROBABILIDAD			IMPACTO			RESULTADO		ESTRATEGIA DE PREVENCIÓN
	Bajo	Medio	Alto	Bajo	Medio	Alto			
	1	2	3	1	2	3			
Error en planificación			X			X	9	Critico	Elaborar un cronograma de actividades
Personal insatisfecho		X				X	6	Critico	Realizar reuniones con personal que utilizaran el sistema.
Falta de recursos		X			X		4	Moderado	Conseguir las herramientas necesarias para el desarrollo del sistema
Insatisfacción del usuario final		X			X		4	Moderado	Realizar reuniones con los clientes para obtener requerimientos.
Cambio de requerimientos			X		X		6	Critico	Realizar reuniones con los involucrados en el desarrollo del sistema y establecer requerimientos.
Aumento de funcionalidades		X				X	6	Critico	Especificación correcta del alcance del proyecto.
Complejidad de la aplicación		X			X		4	Moderado	Desarrollar prototipos del sistema.
Inconsistencia de datos		X				X	6	Critico	Genera respaldos

4.1.2. RIESGOS DE NEGOCIO

RIESGO	PROBABILIDAD			IMPACTO			RESULTADO		ESTRATEGIA DE PREVENCIÓN
	Bajo	Medio	Alto	Bajo	Medio	Alto			
	1	2	3	1	2	3			
No cumplir con requerimientos reales			X			X	9	Critico	Determinación de características funcionales y no funcionales que deberá cumplir el sistema
Poca comunicación personal del proyecto			X			X	9	Critico	Buscar una buena comunicación con el personal
Cierre de negocio		X		X			2	Critico	Establecer un contrato donde se establezcan clausulas para el cumplimiento del mismo.
Resistencia de usuarios finales			X		X		6	Critico	Comunicar a los usuarios sobre los objetivos del sistema
Cambio de tecnología		X				X	6	Critico	

4.1.3. RIESGOS DEL SISTEMA

RIESGO	PROBABILIDAD			IMPACTO			RESULTADO		ESTRATEGIA DE PREVENCIÓN
	Bajo	Medio	Alto	Bajo	Medio	Alto			
	1	2	3	1	2	3			
Mala elaboración del diseño		X				X	6	Critico	Analizar previamente la aplicación del software
Implementación(recursos insuficientes en software y hardware)	X				X		2	Aceptable	Establecer los requerimientos mínimos del sistema y comunicar a la empresa
Interfaz (Usuario final)		X				X	6	Critico	Crear prototipos de interfaz de usuarios
Dificultad de Mantenimiento		X				X	6	Critico	Realizar mantenimientos del sistema mensualmente
La tecnología no satisface las expectativas	X				X		2	Aceptable	Analizar varias tecnologías disponibles
Falta de entrenamiento acerca de las herramientas		X			X		4	Moderado	Realizar capacitación de los empleados
Falla e incompatibilidad de componentes		X				X	6	Critico	Realizar estudios y pruebas previas a su implementación.
Cambio de personal involucrado en el desarrollo del proyecto			X			X	9	Critico	Mayor numero de personal involucrado en el desarrollo de proyecto

Reutilización de código		X			X		4	Moderado	Crear repositorio de código y documentación generada
Perdida de documentación y código fuente	X					X	2	Aceptable	Genera respaldos

5. CAPÍTULO V - FASE DE DISEÑO

5.1. IDENTIFICACIÓN DE ACTORES

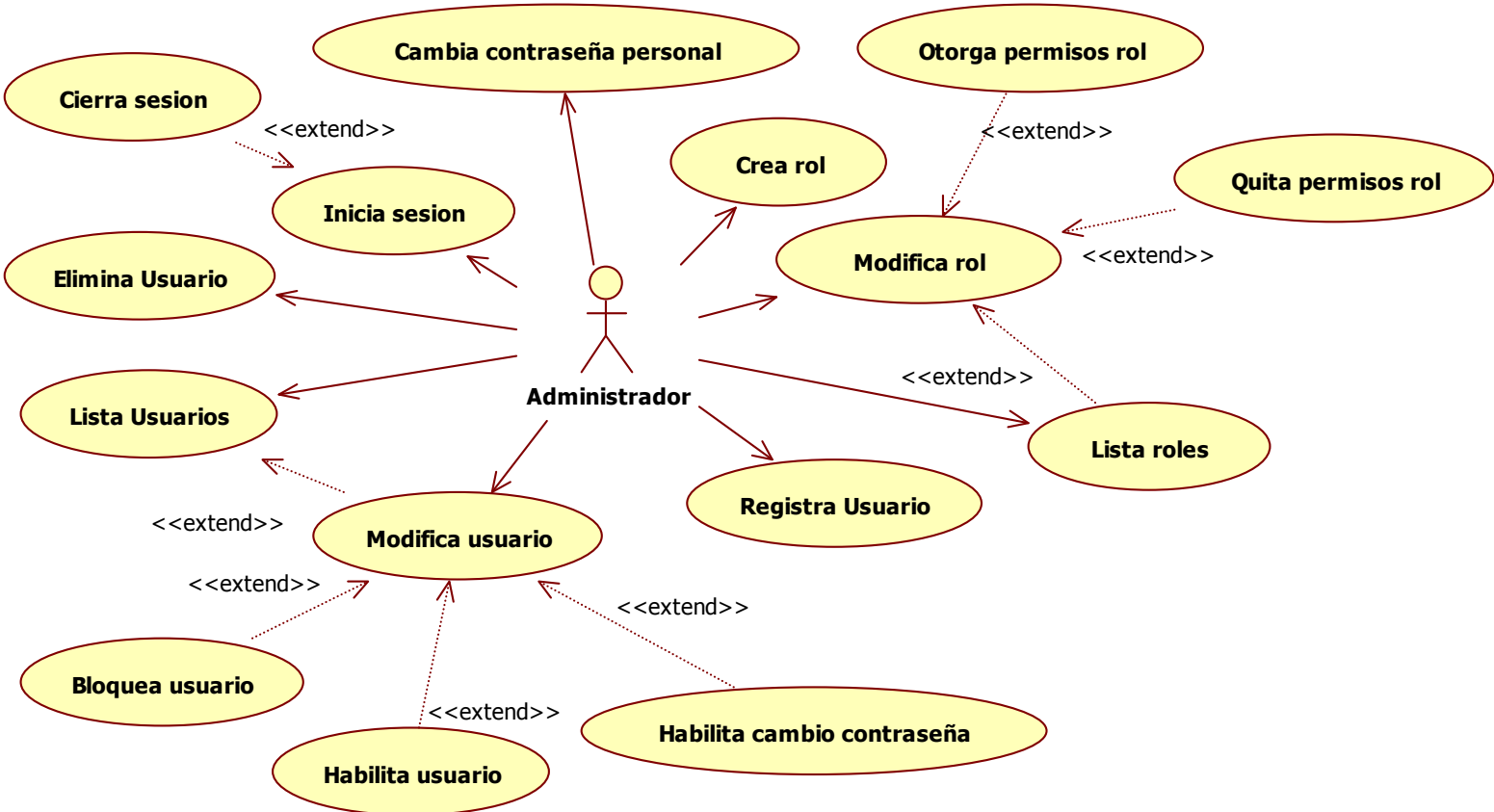
ACTOR	DESCRIPCIÓN
Sistema	Proporciona las interfaces para que el usuario, interactúe con los datos de la base de datos.
Base de datos	Almacena los datos del sistema.
Usuario: Administrador	Se encarga de la administración de usuarios roles y permisos
Usuario: Empleados	Se encarga del registro y mantenimiento de datos e empelados, cargas familiares y el historial de cargos
Usuario: Productos	Se encarga del registro y mantenimientos de los productos del almacén, así como también de las solicitudes y movimientos del producto.
Usuario: Vehículo	Control del registro de vehículos y mantenimientos, solicitudes de vehículos y ordenes de movilización.
Usuario: Medico	Control del registro de historiales médicos y consultas, además del control de medicamentos.
Usuario: Cursos	Control de registro de nuevos cursos, registro de alumnos y calificaciones de alumnos

5.2. IDENTIFICACIÓN DE CASOS DE USO

5.2.1. MÓDULO USUARIOS

ACTOR NRO.	NOMBRE	FUNCIÓN	
		NRO.	DESCRIPCIÓN
1	Usuario: Administrador	CU1	Inicia sesión
		CU2	Cierra sesión
		CU3	Cambia contraseña personal
		CU4	Crea rol
		CU5	Modifica rol
		CU6	Lista rol
		CU7	Agregar permisos rol
		CU8	Quita permisos rol
		CU9	Registra usuario
		CU10	Modifica usuario
		CU11	Elimina usuario
		CU12	Habilita usuario
		CU13	Bloquea usuario
		CU14	Habilita cambio contraseña
		CU15	Lista usuario

5.2.1.1. DIAGRAMA CASOS DE USO MÓDULO USUARIOS



Caso de uso 1: Módulo Usuarios

5.2.1.2. DETALLE CASOS DE USO MÓDULO USUARIOS

Nro. Caso de uso:	CU1	
Nombre:	Iniciar sesión	
Descripción:	El usuario provee su nombre de usuario y contraseña para iniciar una sesión en el sistema	
Actores:	<ul style="list-style-type: none"> - Administrador / Usuario - Sistema 	
Precondiciones:	<ul style="list-style-type: none"> - No haber iniciado una sesión. - Estar registrado en el sistema. - Estar habilitado - Ingresar nombre de usuario correcto - Ingresar contraseña correcta 	
Flujo Normal:		
	Administrador / Usuario	Sistema
1	Inicia el sistema	
2	Ingresar nombre de usuario y contraseña	
3		Valida los datos ingresados
4		Verifica intentos de login
5		Recuperar datos de usuario en la base de datos
6		Autentica al usuario
7		Verifica estado de usuario
8		Registra nueva sesión
9		Re direcciona a la página principal de la aplicación.
Extensiones:		
<p>3.- Si uno o todos los campos están vacíos, regresar a la página de logeo y mostrará mensaje de error.</p> <p>4.- Verificar intentos:</p> <p>4.1.- Si se han realizado 3 intentos de inicio de sesión el sistema se bloquea la cuenta de usuario, se mostrará mensaje de error y mensaje de "Cuenta de usuario</p>		

bloqueada”.

5.- Se recuperan nombre de usuario, contraseña, estado de usuario, rol y privilegios.

6.- Autenticar usuario:

6.1.- Si uno de los campos es incorrecto incrementar intentos y regresar a la página de logeo y mostrará mensaje: “Nombre de usuario o contraseña incorrectos”.

7.- Si el estado del usuario está bloqueado mostrará mensaje de: “*Usuario bloqueado*”.

9.- El sistema almacena en la base de datos una nueva sesión.

Nro. Caso de uso:	CU2	
Nombre:	Cerrar sesión	
Descripción:	El usuario cierra su sesión para salir del sistema	
Actores:	<ul style="list-style-type: none"> • Administrador / Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Haber iniciado sesión. 	
Flujo Normal:		
	Administrador / Usuario	Sistema
1	Elegir opción de cierre de sesión.	
2		Busca sesión del usuario en la base de datos
3		Modifica estado de sesión
4		Elimina sesión del servidor
5		Redirigir a la página de logeo
Extensiones:		
2.- Se busca la última sesión activa del usuario en la base de datos.		
3.- Se modifica su estado y se llenan automáticamente los campos requeridos en la tabla de sesiones de la base de datos.		

Nro. Caso de uso:	CU3
Nombre:	Cambiar contraseña personal

Descripción:	Se desea realizar cambio de contraseña para el acceso al sistema.	
Actores:	<ul style="list-style-type: none"> • Administrador / Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Estar registrado en el sistema • Iniciar sesión. 	
Flujo Normal:		
	Administrador / Usuario	Sistema
1	Elige opción de cambio de contraseña	
2		Despliega formulario de cambio de contraseña.
3	Ingresa la contraseña temporal.	
4		Valida los datos
5		Verifica contraseña
6		Encripta la contraseña
7		Modifica estado contraseña anterior
8		Almacena nueva contraseña.
9		Despliega mensaje de cambio satisfactorio.
Extensiones:		
<p>4.- Valida datos</p> <p>4.1.- El usuario al iniciar sesión deberá cambiar la contraseña.</p> <p>4.2.- Si no cumple con las especificaciones requeridas, despliega mensaje de contraseña no válida.</p> <p>5.- Busca coincidencias con contraseñas anteriores, si existe una igual, mostrar mensaje: <i>“Contraseña ya utilizada, ingrese una nueva”</i>.</p>		

Nro. Caso de uso:	CU4
Nombre:	Crea rol
Descripción:	Se desea crear un nuevo rol de sistema
Actores:	<ul style="list-style-type: none"> • Administrador

	<ul style="list-style-type: none"> • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • No debe existir rol en la base de datos 	
Flujo Normal:		
	Administrador	Sistema
1	Elige la opción crear rol	
2		Muestra formulario de creación de rol
3	Ingresa datos en formulario	
4		Valida datos
5		Almacena datos
6		Muestra mensaje de rol creado exitosamente
Extensiones:		
4.- Valida datos		
4.1.- Si no cumple con las especificaciones requeridas, despliega mensaje de error.		

Nro. Caso de uso:	CU5	
Nombre:	Modificar rol	
Descripción:	Se desea modificar un rol	
Actores:	<ul style="list-style-type: none"> • Administrador • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • Rol debe existir en la base de datos 	
Flujo Normal:		
	Administrador	Sistema
1	Busca o selecciona rol	
2	Elige la opción modificar rol	
3		Muestra formulario de modificar rol
4	Actualiza datos de formulario	
5		Valida datos
6		Almacena datos

7	Muestra mensaje de rol modificado exitosamente
Extensiones:	
4.- Valida datos	
4.1.- Si no cumple con las especificaciones requeridas, despliega mensaje de error.	

Nro. Caso de uso:	CU6	
Nombre:	Lista roles	
Descripción:	Se lista los roles registrados	
Actores:	<ul style="list-style-type: none"> • Administrador • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		
	Administrador	Sistema
1	Elige la opción de listar roles	
2		Consulta base de datos
3		Lista resultados
Extensiones:		
3.- Lista resultados		
3.1.- Si no existen datos, se muestra mensaje de información de registro vacío registrados		

Nro. Caso de uso:	CU7	
Nombre:	Otorga permisos rol	
Descripción:	Se asigna permiso sobre las opciones de un módulo	
Actores:	<ul style="list-style-type: none"> • Administrador • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • Debe existir el rol de usuario 	
Flujo Normal:		
	Administrador	Sistema
1	Busca o selecciona el rol	
2	Elige opción agregar permisos	
3		Muestra formulario de asignar permisos
4	Selecciona los permisos	
5		Valida datos
6		Almacena los datos en la base de datos
7		Muestra mensaje de asignación exitosa
Extensiones:		
5.- Valida datos		
5.1.- Si no cumple con las especificaciones requeridas, despliega mensaje de error.		

Nro. Caso de uso:	CU8	
Nombre:	Quita permisos rol	
Descripción:	Retira permisos a un rol de usuario	
Actores:	<ul style="list-style-type: none"> • Administrador • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • Debe existir el rol de usuario 	
Flujo Normal:		
	Administrador	Sistema
1	Busca o selecciona el rol	
2	Elige opción modificar permisos	
3		Muestra formulario de modificación de permisos
4	Modifica los permisos	
5		Valida datos
6		Almacena los datos en la base de datos
7		Muestra mensaje de modificación exitosa
Extensiones:		
5.- Valida datos		
5.1.- Si no cumple con las especificaciones requeridas, despliega mensaje de error.		

Nro. Caso de uso:	CU9	
Nombre:	Registrar usuario	
Descripción:	Se desea registrar un nuevo usuario para que utilice el sistema	
Actores:	<ul style="list-style-type: none"> • Administrador • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • El usuario no debe estar registrado 	
Flujo Normal:		
	Administrador	Sistema
1	Elige la opción registrar usuario	
2		Muestra formulario de registro de usuario
3	Ingresa datos del usuario	
4		Valida datos ingresados
5		Almacena datos en la base de datos
6		Muestra mensaje de usuario creado satisfactoriamente
Extensiones:		
4.- Valida datos ingresados		
4.1.- Si uno o más campos obligatorios son NULOS, el sistema mostrará un mensaje de error en campos obligatorios.		

4.2.- Si uno o más campos no cumplen con las especificaciones establecidas, se mostrará un mensaje de error en formato de campo.
 5. Almacena los datos en la base de datos
 5.1.- Se deberá almacenar los datos de la contraseña con estado cambio de contraseña o con contraseña caducada, para cuando el usuario ingrese por primera vez al sistema, este se vea obligado a realizar el cambio de contraseña.

Nro. Caso de uso:	CU10	
Nombre:	Modifica usuario	
Descripción:	Se desea modificar los datos de un usuario	
Actores:	<ul style="list-style-type: none"> • Administrador • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • El usuario debe estar registrado en la base de datos 	
Flujo Normal:		
	Administrador	Sistema
1	Busca o selecciona el usuario	
2	Elige la opción modificar usuario	
3		Muestra formulario de modificar usuario
4	Modifica datos del usuario	
5		Valida datos ingresados
6		Almacena datos en la base de datos
7		Muestra mensaje de usuario modificado satisfactoriamente
Extensiones:		
4.- Modifica datos del usuario: Se podrán modificar únicamente los siguientes datos del usuario: 4.1.- El estado del usuario 4.2.- El rol del usuario 4.3.- Habilitar cambio de contraseña 5.- Valida datos 5.1.- Si no cumple con las especificaciones requeridas, despliega mensaje de error.		

Nro. Caso de uso:	CU11	
Nombre:	Elimina usuario	
Descripción:	El administrador elimina un usuario del sistema	
Actores:	<ul style="list-style-type: none"> • Administrador • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • El usuario debe estar registrado en la base de datos 	
Flujo Normal:		

	Administrador	Sistema
1	Busca o selecciona el usuario	
2	Elige la opción eliminar usuario	
3		Muestra mensaje de confirmación
4	Confirma eliminación	
5		Valida datos ingresados
6		Modifica estado de usuario en la base de datos
7		Muestra mensaje de usuario eliminado satisfactoriamente
Extensiones:		
5.- Valida datos		
5.1.- Si no cumple con las especificaciones requeridas, despliega mensaje de error.		
6.- Modifica estado de usuario en la base de datos		
6.1.- El estado del usuario cambia a ELIMINADO		

Nro. Caso de uso:	CU12	
Nombre:	Habilita usuario	
Descripción:	Habilitación de un usuario bloqueado	
Actores:	<ul style="list-style-type: none"> • Administrador • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • El usuario debe estar en estado bloqueado 	
Flujo Normal:		
	Administrador	Sistema
1	Busca o selecciona el usuario bloqueado	
2		Muestra formulario de modificación de usuario
3	Modifica estado de usuario	
4		Valida los datos
5		Actualiza estado de usuario a HABILITADO en la base de datos
6		Muestra mensaje de usuario modificado
Extensiones:		
4.- Valida datos		
4.1.- Si no cumple con las especificaciones requeridas, despliega mensaje de error.		

Nro. Caso de uso:	CU13	
Nombre:	Bloquea usuario	
Descripción:	Bloqueo de un usuario	
Actores:	<ul style="list-style-type: none"> • Administrador • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • El usuario debe estar en estado habilitado 	

Flujo Normal:		
	Administrador	Sistema
1	Busca o selecciona el usuario bloqueado	
2		Muestra formulario de modificación de usuario
3	Modifica estado de usuario	
4		Valida los datos
5		Actualiza estado de usuario a BLOQUEADO en la base de datos
6		Muestra mensaje de usuario modificado
Extensiones:		
4.- Valida datos		
4.1.- Si no cumple con las especificaciones requeridas, despliega mensaje de error.		

Nro. Caso de uso:	CU14	
Nombre:	Habilita cambio contraseña	
Descripción:	Se habilita el cambio de contraseña al iniciar sesión	
Actores:	<ul style="list-style-type: none"> • Administrador • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • El usuario solicita cambio de contraseña 	
Flujo Normal:		
	Administrador	Sistema
1	Busca o selecciona el usuario	
2		Muestra formulario de modificación de usuario
3	Selecciona la opción modificar contraseña	
4		Valida los datos
5		Actualiza estado de usuario a MODIFICAR CONTRASEÑA
6		Muestra mensaje de usuario modificado
Extensiones:		
4.- Valida datos		
4.1.- Si no cumple con las especificaciones requeridas, despliega mensaje de error.		

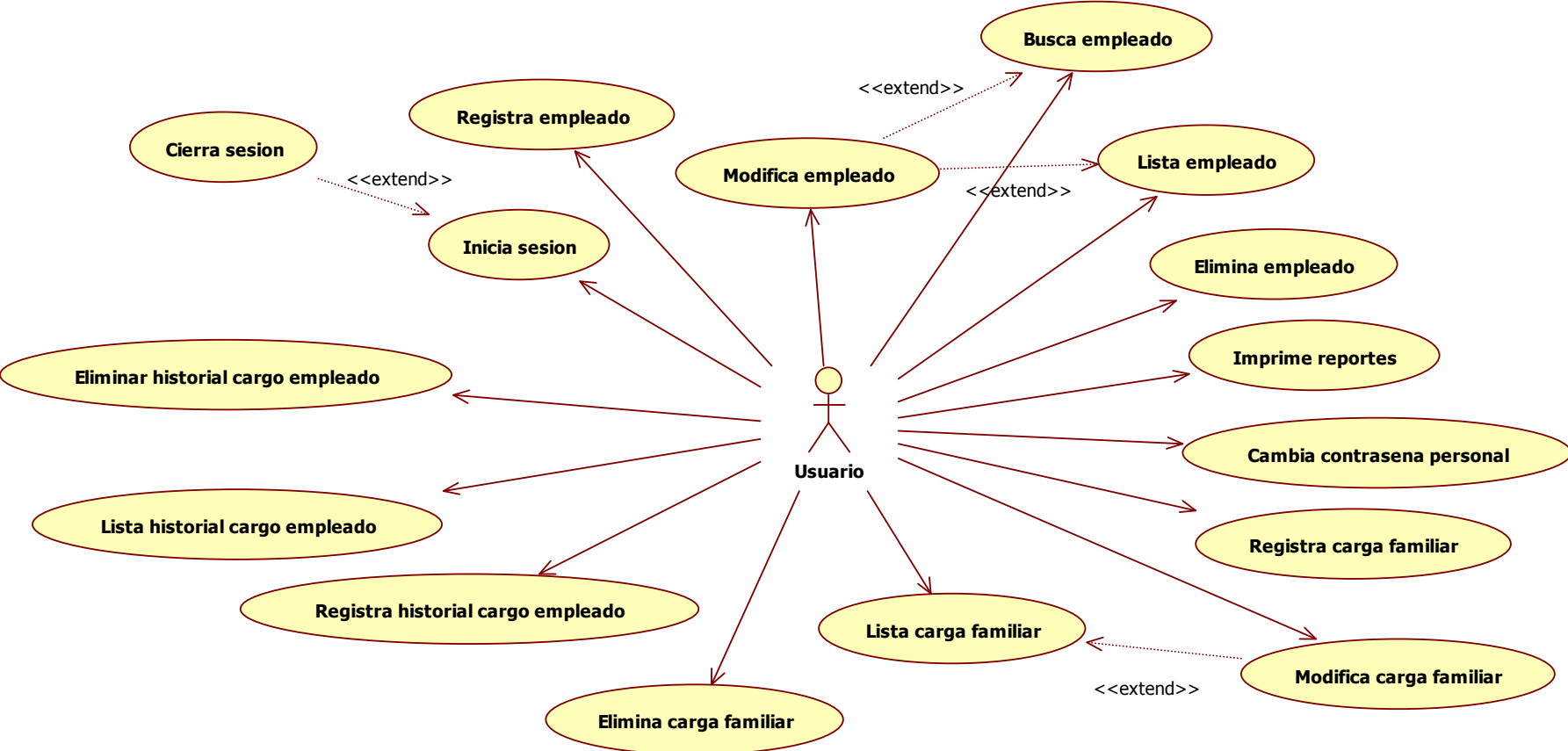
Nro. Caso de uso:	CU15	
Nombre:	Lista usuarios	
Descripción:	Se lista los usuarios registrados	
Actores:	<ul style="list-style-type: none"> • Administrador • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		

	Administrador	Sistema
1	Elige la opción de listar usuarios	
2		Consulta base de datos
3		Lista resultados
Extensiones:		
3.- Lista resultados		
3.1.- Si no existen datos, se muestra mensaje de información de registro vacío.		

5.2.2. MÓDULO EMPLEADOS

ACTOR NRO.	NOMBRE	FUNCIÓN	
		NRO.	DESCRIPCIÓN
2	Usuario: Empleados	CU1	Inicia sesión
		CU16	Registra empleado
		CU17	Modifica empleado
		CU18	Busca empleado
		CU19	Lista empleado
		CU20	Elimina empleado
		CU21	Imprime reporte
		CU22	Registra carga familiar
		CU23	Modifica carga familiar
		CU24	Lista carga familiar
		CU25	Elimina carga familiar
		CU26	Registra historial cargo empleado
		CU27	Elimina historial cargo empleado
CU28	Lista historial cargo empleado		

5.2.2.1. DIAGRAMA CASOS DE USO MÓDULO EMPLEADOS



Caso de Uso 2: Módulo Empleados

5.2.2.2. DETALLES CASOS DE USO MÓDULO EMPLEADOS

Nro. Caso de uso:	CU16	
Nombre:	Registra empleado	
Descripción:	Se registra un nuevo empleado	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • El empleado no debe estar registrado 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción registrar empleado	
2		Muestra formulario de registro de empleado
3	Llena el formulario de registro de empleado y cargo	
4		Valida datos ingresados
5		Almacena datos en la base de datos
6		Muestra mensaje de empleado registrado satisfactoriamente
Extensiones:		
4.- Valida datos ingresados		
4.1.- Si uno o más campos obligatorios son NULOS, el sistema mostrará un mensaje de error en campos obligatorios.		
4.2.- Si uno o más campos no cumplen con las especificaciones establecidas, se mostrará un mensaje de error en formato de campo.		

Nro. Caso de uso:	CU17	
Nombre:	Modifica empleado	
Descripción:	Se modifica los datos de empleado	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • El empleado debe estar registrado 	
Flujo Normal:		
	Usuario	Sistema
1	Busca o selecciona el empleado	
2	Elige la opción modificar empleado	
3		Muestra formulario de modificar empleado
4	Modifica datos del formulario	
5		Valida datos ingresados
6		Actualiza datos en la base de datos
7		Muestra mensaje de empleado modificado satisfactoriamente

Extensiones:
4.- Valida datos ingresados
4.1.- Si uno o más campos obligatorios son NULOS, el sistema mostrará un mensaje de error en campos obligatorios.
4.2.- Si uno o más campos no cumplen con las especificaciones establecidas, se mostrará un mensaje de error en formato de campo.

Nro. Caso de uso:	CU18	
Nombre:	Busca empleado	
Descripción:	Se busca un empleado registrado	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción de listar empleados	
2		Muestra lista de empelados
3	Ingresa criterio de búsqueda	
4		Consulta la base de datos
5		Lista resultado
Extensiones:		
3.- Lista resultado		
3.1.- Si no existen datos, se muestra mensaje de información de registro vacio registrados		

Nro. Caso de uso:	CU19	
Nombre:	Lista empleado	
Descripción:	Se lista los empleado registrados	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción de listar empleados	
2		Consulta la base de datos
3		Lista resultado
Extensiones:		
3.- Lista resultado		
3.1.- Si no existen datos, se muestra mensaje de información de registro vacio registrados		

Nro. Caso de uso:	CU20
Nombre:	Elimina empleado
Descripción:	El usuario elimina un empleado del sistema
Actores:	<ul style="list-style-type: none"> • Usuario

	<ul style="list-style-type: none"> • Sistema 																								
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • El empleado debe estar registrado en la base de datos 																								
Flujo Normal:																									
	<table border="1"> <thead> <tr> <th></th> <th>Usuario</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Busca o selecciona el empleado</td> <td></td> </tr> <tr> <td>2</td> <td>Elige la opción eliminar empleado</td> <td></td> </tr> <tr> <td>3</td> <td></td> <td>Muestra mensaje de confirmación</td> </tr> <tr> <td>4</td> <td>Confirma eliminación</td> <td></td> </tr> <tr> <td>5</td> <td></td> <td>Valida datos ingresados</td> </tr> <tr> <td>6</td> <td></td> <td>Modifica estado de empleado en la base de datos</td> </tr> <tr> <td>7</td> <td></td> <td>Muestra mensaje de empleado eliminado satisfactoriamente</td> </tr> </tbody> </table>		Usuario	Sistema	1	Busca o selecciona el empleado		2	Elige la opción eliminar empleado		3		Muestra mensaje de confirmación	4	Confirma eliminación		5		Valida datos ingresados	6		Modifica estado de empleado en la base de datos	7		Muestra mensaje de empleado eliminado satisfactoriamente
	Usuario	Sistema																							
1	Busca o selecciona el empleado																								
2	Elige la opción eliminar empleado																								
3		Muestra mensaje de confirmación																							
4	Confirma eliminación																								
5		Valida datos ingresados																							
6		Modifica estado de empleado en la base de datos																							
7		Muestra mensaje de empleado eliminado satisfactoriamente																							
Extensiones:																									
5.- Valida datos																									
5.1.- Si no cumple con las especificaciones requeridas, despliega mensaje de error.																									
7.- Modifica estado de empleado en la base de datos																									
7.1.- Se actualiza el registro a un estado de eliminado																									

Nro. Caso de uso:	CU21									
Nombre:	Imprime reporte									
Descripción:	Se imprime reporte de empleados									
Actores:	<ul style="list-style-type: none"> - Usuario - Sistema 									
Precondiciones:	- Iniciar sesión									
Flujo Normal:										
	<table border="1"> <thead> <tr> <th></th> <th>Usuario</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Elige la opción de imprimir reporte</td> <td></td> </tr> <tr> <td>2</td> <td></td> <td>Imprime reporte</td> </tr> </tbody> </table>		Usuario	Sistema	1	Elige la opción de imprimir reporte		2		Imprime reporte
	Usuario	Sistema								
1	Elige la opción de imprimir reporte									
2		Imprime reporte								
Extensiones:										

Nro. Caso de uso:	CU22
Nombre:	Registra carga familiar
Descripción:	Se registra la carga familiar de un empleado
Actores:	<ul style="list-style-type: none"> - Usuario - Sistema
Precondiciones:	<ul style="list-style-type: none"> - Iniciar sesión - Se selecciona un empleado - El empleado debe estar registrado en el sistema - Carga familiar no debe estar registrada
Flujo Normal:	

	Usuario	Sistema
1	Selecciona opción “Registro Carga Familiar”	
2	Busca o selecciona el empleado	
3	Elige la opción “Agregar Carga Familiar”	
4		Muestra formulario de registro de carga familiar
5	Llena formulario de registro de carga familiar	
6		Valida datos ingresados
7		Registra datos en la base de datos
8		Muestra mensaje de carga familiar registrado satisfactoriamente
Extensiones:		
6.- Valida datos		
6.1.- Si no cumple con las especificaciones requeridas, despliega mensaje de error.		

Nro. Caso de uso:	CU23	
Nombre:	Modifica carga familiar	
Descripción:	Se modifica la carga familiar de un empleado	
Actores:	<ul style="list-style-type: none"> - Usuario - Sistema 	
Precondiciones:	<ul style="list-style-type: none"> - Iniciar sesión - El empleado debe estar registrado en el sistema - Carga familiar debe estar registrada 	
Flujo Normal:		
	Usuario	Sistema
1	Selecciona opción “Registro Carga Familiar”	
2	Busca o selecciona el empleado	
3	Elige la opción “Ver Detalles”	
4		Muestra detalles de empleado y cargas familiares
5	Elige la opción modificar carga familiar	
6		Muestra formulario de modificación de carga familiar
7	Actualiza formulario de modificación de carga familiar	
8		Valida datos ingresados
9		Actualiza los datos en la base de datos
10		Muestra mensaje de carga familiar modificada satisfactoriamente
Extensiones:		
8.- Valida datos		
8.1.- Si no cumple con las especificaciones requeridas, despliega mensaje de error.		

Nro. Caso de uso:	CU24	
Nombre:	Lista carga familiar	
Descripción:	Se listan las cargas familiares de un empleado	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • El empleado debe estar registrado en el sistema • Tener registrado carga familiar 	
Flujo Normal:		
	Usuario	Sistema
1	Selecciona opción "Registro Carga Familiar"	
2	Busca o selecciona el empleado	
3	Elige la opción "Ver Detalles"	
4		Muestra detalles de empleado y cargas familiares
Extensiones:		

Nro. Caso de uso:	CU25	
Nombre:	Elimina carga familiar	
Descripción:	Se elimina una carga familiar de un empleado	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • El empleado debe estar registrado en el sistema • Tener registrado carga familiar 	
Flujo Normal:		
	Usuario	Sistema
1	Selecciona opción "Registro Carga Familiar"	
2	Busca o selecciona el empleado	
3	Elige la opción ver detalles	
4		Muestra detalles de empleado y cargas familiares
5	Elige la opción "Eliminar Carga Familiar"	
6		Muestra mensaje de confirmación
7	Confirma eliminación	
8		Valida datos ingresados
9		Actualiza los datos en la base de datos
10		Muestra mensaje de carga familiar eliminación exitosa
Extensiones:		
8.- Valida datos		

8.1.- Si no cumple con las especificaciones requeridas, despliega mensaje de error.

Nro. Caso de uso:	CU26	
Nombre:	Registra historial cargo empleado	
Descripción:	Se registra el historial cargo del empleado	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • Se debe seleccionar un empleado • El empleado debe estar registrado en el sistema 	
Flujo Normal:		
	Usuario	Sistema
1	Selecciona opción “Registro Historial Cargo Empleado”	
2	Busca o selecciona el empleado	
3	Elige la opción “Ver Detalles”	
4		Muestra detalles de empleado y cargos
5	Elige la opción “Agregar Cargo”	
6		Muestra formulario de registro de cargo
7	Llena formulario de registro de cargo	
8		Valida datos ingresados
9		Registra datos en la base de datos
10		Muestra mensaje de cargo registrado satisfactoriamente
Extensiones:		
7.- Valida datos ingresados		
7.1.- Si uno o más campos obligatorios son NULOS, el sistema mostrará un mensaje de error en campos obligatorios.		
7.2.- Si uno o más campos no cumplen con las especificaciones establecidas, se mostrará un mensaje de error en formato de campo.		
8.- Valida datos		
8.1.- Si no cumple con las especificaciones requeridas, despliega mensaje de error.		

Nro. Caso de uso:	CU27	
Nombre:	Elimina historial cargo empleado	
Descripción:	Se elimina un registro del historial cargo empleado	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • Se debe seleccionar un empleado • El empleado debe estar registrado en el sistema 	
Flujo Normal:		
	Usuario	Sistema
1	Selecciona opción “Elimina	

	historial cargo empleado”	
2	Busca o selecciona el empleado	
3	Elige la opción “Ver Detalles”	
4		Muestra detalles de empleado y cargos
5	Elige la opción “Eliminar Cargo”	
6		Valida datos ingresados
7		Elimina el registra de la base de datos
8		Muestra mensaje de cargo eliminado satisfactoriamente
Extensiones:		
6.- Valida datos		
6.1.- Si no cumple con las especificaciones requeridas, despliega mensaje de error.		

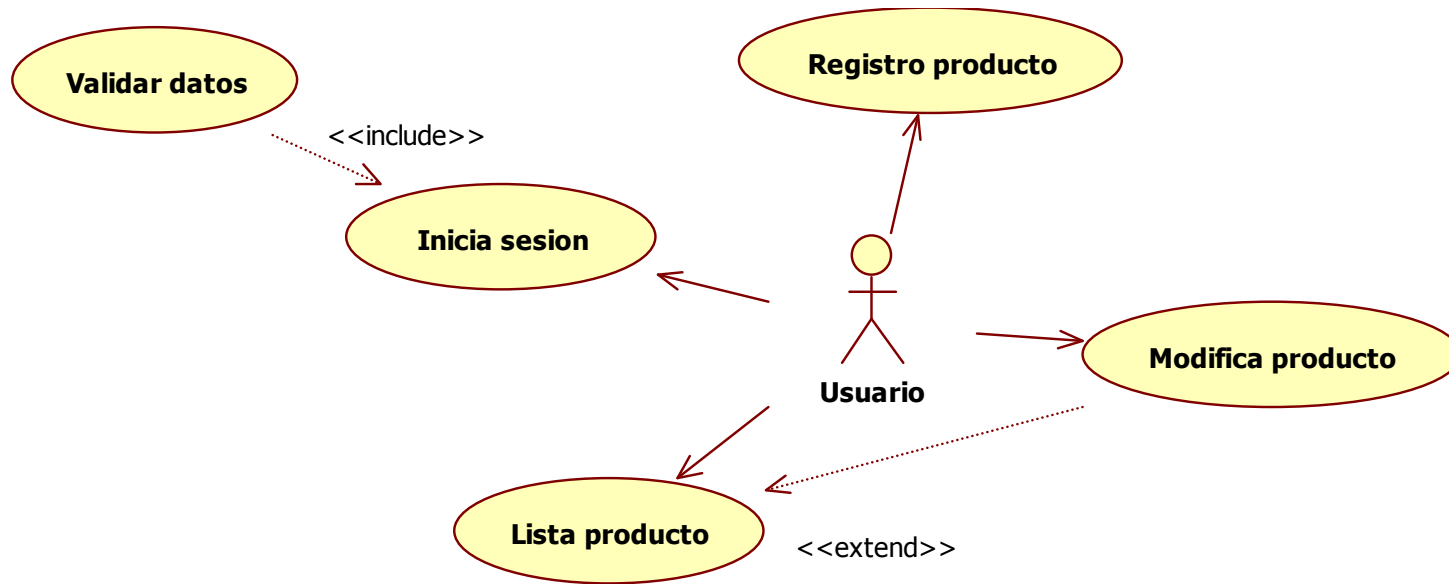
Nro. Caso de uso:	CU28	
Nombre:	Lista historial cargo empleado	
Descripción:	Se lista el historial de cargos de un empleado	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • El empleado debe estar registrado en el sistema • Tener registrado cargos 	
Flujo Normal:		
	Usuario	Sistema
1	Selecciona opción “Consulta Historial Cargo Empleado”	
2	Busca o selecciona el empleado	
3	Elige la opción “Ver Detalles”	
4		Muestra detalles de empleado y el historial de cargos
Extensiones:		

5.2.3. MÓDULO PRODUCTOS

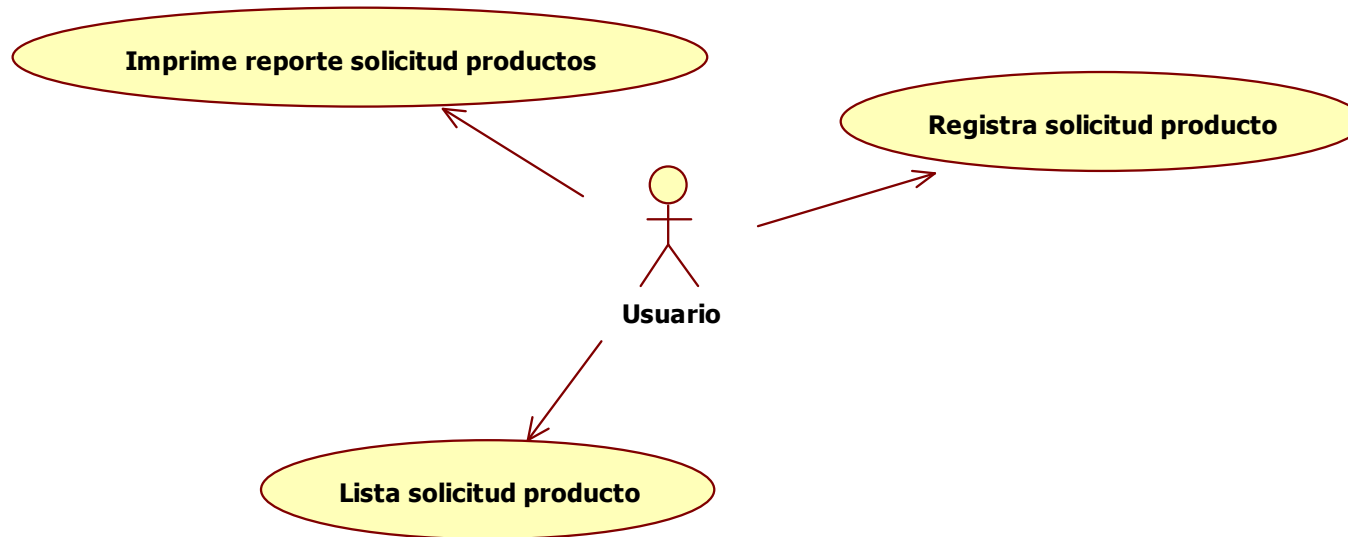
ACTOR NRO.	NOMBRE	FUNCIÓN	
		NRO.	DESCRIPCIÓN
3	Usuario: Productos	CU1	Inicia sesión
		CU29	Registra producto
		CU30	Modifica producto
		CU31	Lista producto

		CU32	Registra producto en bodega
		CU33	Modifica producto de bodega
		CU34	Lista producto de bodega
		CU35	Registra movimiento de producto en bodega
		CU36	Lista movimientos de productos
		CU37	Imprime reporte productos
		CU38	Registra solicitud producto
		CU39	Lista solicitud productos
		CU40	Imprime reporte solicitud productos

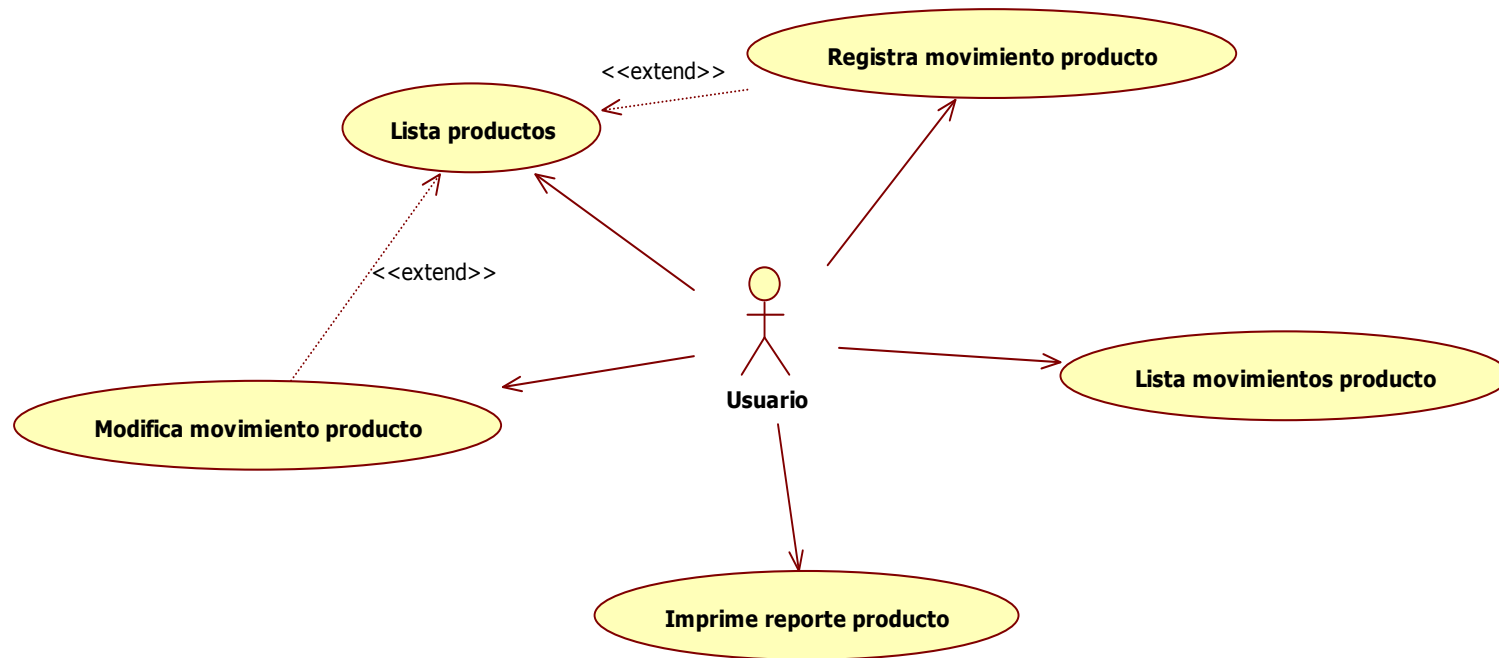
5.2.3.1. DIAGRAMA CASOS DE USO MÓDULO PRODUCTOS



Caso de Uso 3: Módulo Productos



Caso de Uso 4: Módulo Solicitudes Productos



Caso de Uso 5: Módulo Productos Almacén

5.2.3.2. DETALLES CASOS DE USO MÓDULO PRODUCTOS

Nro. Caso de uso:	CU29	
Nombre:	Registra producto	
Descripción:	Se registra un nuevo producto	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • El producto no debe estar registrado 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción “Registrar Producto”	
2	Elige la opción “Agregar Producto”	
3		Muestra formulario de registro de producto
4	Llena el formulario de registro de producto	
5		Valida datos ingresados
6		Almacena datos en la base de datos
7		Muestra mensaje de producto registrado satisfactoriamente
Extensiones:		
<p>4.- Valida datos ingresados</p> <p>4.1.- Si uno o más campos obligatorios son NULOS, el sistema mostrará un mensaje de error en campos obligatorios.</p> <p>4.2.- Si uno o más campos no cumplen con las especificaciones establecidas, se mostrará un mensaje de error en formato de campo.</p>		

Nro. Caso de uso:	CU30	
Nombre:	Modifica producto	
Descripción:	Se modifica los datos del producto	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • El producto debe estar registrado 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción “Registrar Producto”	
2	Busca o selecciona el producto	
3	Elige la opción “Modificar Producto”	
4		Muestra formulario de modificar producto
5	Modifica datos del formulario	

6		Valida datos ingresados
7		Actualiza datos en la base de datos
8		Muestra mensaje de producto modificado satisfactoriamente

Extensiones:

- 4.- Valida datos ingresados
4.1.- Si uno o más campos obligatorios son NULOS, el sistema mostrará un mensaje de error en campos obligatorios.
4.2.- Si uno o más campos no cumplen con las especificaciones establecidas, se mostrará un mensaje de error en formato de campo.

Nro. Caso de uso:	CU31	
Nombre:	Lista producto	
Descripción:	Lista los productos registrados	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción "Registrar Producto"	
2		Consulta la base de datos
3		Muestra resultado
Extensiones:		

Nro. Caso de uso:	CU32	
Nombre:	Registra producto en bodega	
Descripción:	Registro de un producto a una determinada bodega	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • El producto no debe estar registrado en la bodega 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción "Bodega"	
2	Elige la opción "Agregar Producto Bodega"	
3		Muestra formulario de registro de producto bodega
4	Llena formulario de registro de producto bodega	
5		Valida datos ingresados
6		Registra datos en la base de datos
7		Muestra mensaje de producto de bodega registrado satisfactoriamente
Extensiones:		

4.- Valida datos ingresados
4.1.- Si uno o más campos obligatorios son NULOS, el sistema mostrará un mensaje de error en campos obligatorios.
4.2.- Si uno o más campos no cumplen con las especificaciones establecidas, se mostrará un mensaje de error en formato de campo.
5.- Registra datos en la base de datos
5.1.- Se deberá registrar el producto de acuerdo a la ubicación de la bodega

Nro. Caso de uso:	CU33	
Nombre:	Modifica producto de bodega	
Descripción:	Se modifica un producto de una bodega	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • El producto debe estar registrado en la bodega 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción “Bodega”	
2	Busca o selecciona el producto de la bodega	
3	Elige la opción “Modificar Producto Bodega”	
4		Muestra formulario de modificar producto bodega
5	Modifica datos del formulario	
6		Valida datos ingresados
7		Actualiza datos en la base de datos
8		Muestra mensaje de producto de bodega modificado satisfactoriamente
Extensiones:		
5.- Valida datos ingresados		
5.1.- Si uno o más campos obligatorios son NULOS, el sistema mostrará un mensaje de error en campos obligatorios.		
5.2.- Si uno o más campos no cumplen con las especificaciones establecidas, se mostrará un mensaje de error en formato de campo.		

Nro. Caso de uso:	CU34	
Nombre:	Lista producto de bodega	
Descripción:	Lista los productos registrados en una bodega	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción “Bodega”	
2		Muestra resultado

Extensiones:	
-	Si un producto de bodega es igual o está por debajo del de la cantidad mínima, se deberá mostrar una alerta de información.
-	Si un producto de bodega está agotado, se deberá mostrar una alerta de emergencia.

Nro. Caso de uso:	CU35
Nombre:	Registra movimiento de producto en bodega
Descripción:	Registra movimientos de ingreso o egreso de los productos de una bodega
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión

Flujo Normal:		
	Usuario	Sistema
1	Busca o selecciona el producto de la bodega	
2	Elige la opción registrar movimiento producto bodega	
3		Muestra formulario de registrar movimiento producto bodega
4	Llena datos del formulario	
5		Valida datos ingresados
6		Actualiza datos en la base de datos
7		Muestra mensaje de movimiento producto de bodega registrado satisfactoriamente

Extensiones:	
5.-	Valida datos ingresados
5.1.-	Si uno o más campos obligatorios son NULOS, el sistema mostrará un mensaje de error en campos obligatorios.
5.2.-	Si uno o más campos no cumplen con las especificaciones establecidas, se mostrará un mensaje de error en formato de campo.

Nro. Caso de uso:	CU36	
Nombre:	Lista movimientos de productos	
Descripción:	Lista los productos registrados en una bodega	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		
	Usuario	Sistema
1	Busca o selecciona el producto de la bodega	
2	Elige la opción listar movimientos producto bodega	
3		Consulta la base de datos

4	Muestra resultado
Extensiones:	

Nro. Caso de uso:	CU37	
Nombre:	Imprime reporte producto	
Descripción:	Se imprime reporte de productos	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción de imprimir reporte	
2		Imprime reporte
Extensiones:		

Nro. Caso de uso:	CU38	
Nombre:	Registra solicitud producto	
Descripción:	Se registra una solicitud de un producto	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción ingresar solicitud producto	
2		Muestra formulario de registro de solicitud producto
3	Llena formulario	
4		Valida datos ingresados
5		Registra datos en la base de datos
6		Muestra mensaje de solicitud de producto registrado satisfactoriamente
Extensiones:		
<p>4.- Valida datos ingresados</p> <p>4.1.- Si uno o más campos obligatorios son NULOS, el sistema mostrará un mensaje de error en campos obligatorios.</p> <p>4.2.- Si uno o más campos no cumplen con las especificaciones establecidas, se mostrará un mensaje de error en formato de campo.</p> <p>5.- Registra datos en la base de datos</p> <p>5.1.- Se deberá registrar el producto de acuerdo a la ubicación de la solicitud</p>		

Nro. Caso de uso:	CU39	
Nombre:	Lista solicitud productos	
Descripción:	Lista las solicitudes de productos registradas	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción listar solicitud de productos	
2		Consulta la base de datos
3		Muestra resultado
Extensiones:		

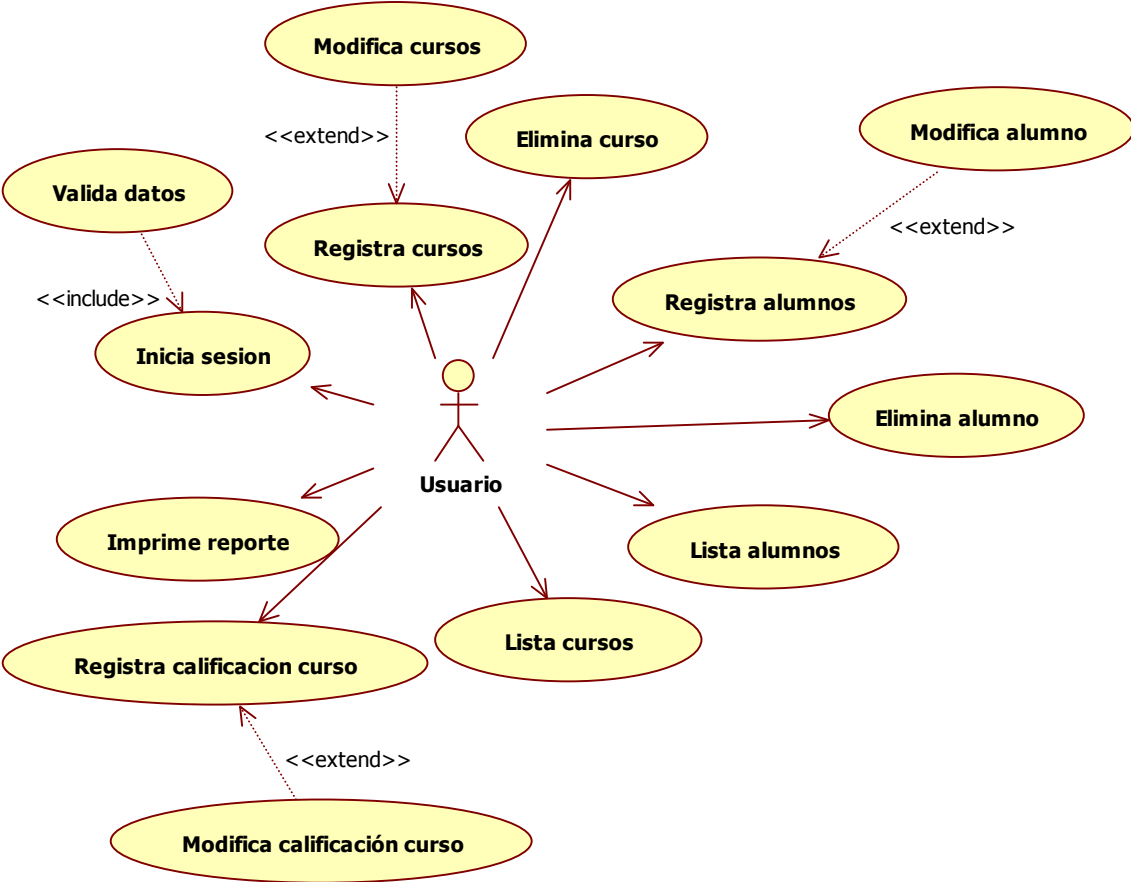
Nro. Caso de uso:	CU40	
Nombre:	Imprime reporte solicitud producto	
Descripción:	Se imprime reporte de las solicitudes de productos realizadas por los empleados	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción de imprimir reporte	
2		Imprime reporte
Extensiones:		

5.2.4. MÓDULO CURSOS

ACTOR NRO.	NOMBRE	FUNCIÓN	
		NRO.	DESCRIPCIÓN
4	Usuario: Cursos	CU1	Inicia sesión
		CU41	Registra cursos
		CU42	Modifica cursos
		CU43	Elimina curso
		CU44	Lista cursos
		CU45	Registra alumnos
		CU46	Modifica alumnos

		CU47	Elimina alumno
		CU48	Lista alumnos
		CU49	Registra calificación curso
		CU50	Modifica calificación curso
		CU51	Imprime reporte

5.2.4.1. DIAGRAMA CASOS DE USO MÓDULO CURSOS



Caso de Uso 6: Módulo Cursos

5.2.4.2. DETALLES CASOS DE USO MÓDULO CURSOS

Nro. Caso de uso:	CU41	
Nombre:	Registra cursos	
Descripción:	Se registra un nuevo curso	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • El curso no debe estar registrado 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción registrar cursos	
2		Muestra formulario de registro de cursos
3	Llena formulario de registro de curso	
4		Valida datos ingresados
5		Registra datos en la base de datos
6		Muestra mensaje de curso registrado satisfactoriamente
Extensiones:		
4.- Valida datos ingresados		
4.1.- Si uno o más campos obligatorios son NULOS, el sistema mostrará un mensaje de error en campos obligatorios.		
4.2.- Si uno o más campos no cumplen con las especificaciones establecidas, se mostrará un mensaje de error en formato de campo.		
5.- Registra datos en la base de datos		

Nro. Caso de uso:	CU42	
Nombre:	Modificar curso	
Descripción:	Se modifican los datos de un curso.	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • El curso debe estar registrado 	
Flujo Normal:		

	Usuario	Sistema
1	Elige la opción listar cursos	
2		Muestra lista de cursos registrados
3	Selecciona curso a modificar	
4		Muestra formulario de modificación de curso
5	Llena formulario de modificación de curso	
6		Valida datos ingresados
7		Registra datos en la base de datos
8		Muestra mensaje de curso actualizado satisfactoriamente

Extensiones:

- 6.- Valida datos ingresados.
6.1.- Si uno o más campos obligatorios son NULOS, el sistema mostrará un mensaje de error en campos obligatorios.
6.2.- Si uno o más campos no cumplen con las especificaciones establecidas, se mostrará un mensaje de error en formato de campo.
7.- Actualiza la base de datos.

Nro. Caso de uso:	CU43
Nombre:	Elimina curso
Descripción:	Se elimina un curso de la base de datos
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • El curso debe estar registrado

Flujo Normal:

	Usuario	Sistema
1	Elige la opción listar cursos	
2		Muestra lista de cursos registrados
3	Busca o selecciona el curso	
4	Elige la opción eliminar curso	
5		Muestra mensaje de confirmación
6	Confirma eliminación	
7		Valida datos ingresados
8		Modifica estado del curso en la base de datos
9		Muestra mensaje de curso eliminado satisfactoriamente

Extensiones:

- 4.- Valida datos ingresados
4.1.- Si uno o más campos obligatorios son NULOS, el sistema mostrará un mensaje de error en campos obligatorios.
4.2.- Si uno o más campos no cumplen con las especificaciones establecidas, se mostrará un mensaje de error en formato de campo.
5.- Actualiza la base de datos

Nro. Caso de uso:	CU44	
Nombre:	Lista cursos	
Descripción:	Se lista los cursos registrados	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción listar cursos	
2		Muestra lista de cursos registrados
Extensiones:		

Nro. Caso de uso:	CU45	
Nombre:	Registra alumno	
Descripción:	Se registra un alumno	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • El alumno no debe estar registrado en el curso 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción listar cursos	
2		Muestra lista de cursos registrados
3	Busca o selecciona el curso	
4	Elige la opción registrar alumnos	
5		Muestra formulario de registro de alumnos
6	Llena formulario de registro de alumnos	
7		Valida datos ingresados
8		Muestra mensaje de alumno registrado satisfactoriamente
Extensiones:		
7.- Valida datos ingresados		
7.1.- Si uno o más campos obligatorios son NULOS, el sistema mostrará un mensaje de error en campos obligatorios.		
7.2.- Si uno o más campos no cumplen con las especificaciones establecidas, se mostrará un mensaje de error en formato de campo.		
8.- Registra datos en la base de datos		

Nro. Caso de uso:	CU46	
Nombre:	Modificación alumnos	
Descripción:	Se modifica los datos de un alumno registrado	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	

Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • El alumno debe estar registrado en el curso 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción listar cursos	
2		Muestra lista de cursos registrados
3	Busca o selecciona el curso	
4	Elige la opción listar alumnos curso	
5		Muestra lista de alumnos registrados
6	Busca o selecciona el alumno	
7	Elige la opción modificar alumnos	
8		Muestra formulario de modificación de alumnos
9	Llena formulario de modificación de alumnos	
10		Valida datos ingresados
11		Muestra mensaje de alumno actualizado satisfactoriamente
Extensiones:		
<p>10.- Valida datos ingresados</p> <p>10.1.- Si uno o más campos obligatorios son NULOS, el sistema mostrará un mensaje de error en campos obligatorios.</p> <p>10.2.- Si uno o más campos no cumplen con las especificaciones establecidas, se mostrará un mensaje de error en formato de campo.</p> <p>11.- Actualiza la base de datos</p>		

Nro. Caso de uso:	CU47	
Nombre:	Elimina alumnos	
Descripción:	Se elimina un alumno registrado	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • El alumno debe estar registrado en el curso 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción listar cursos	
2		Muestra lista de cursos registrados
3	Busca o selecciona el curso	
4	Elige la opción listar alumnos curso	
5		Muestra lista de alumnos registrados
6	Busca o selecciona el alumno	
7	Elige la opción eliminar alumno	
8		Muestra mensaje de confirmación
9	Confirma eliminación	
10		Valida datos ingresados
11		Elimina registro de la base de datos

Extensiones:
10.- Valida datos ingresados
10.1.- Si uno o más campos obligatorios son NULOS, el sistema mostrará un mensaje de error en campos obligatorios.
10.2.- Si uno o más campos no cumplen con las especificaciones establecidas, se mostrará un mensaje de error en formato de campo.
11.- Elimina registro de la base de datos

Nro. Caso de uso:	CU48	
Nombre:	Lista alumnos	
Descripción:	Se lista los alumnos de un curso	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción listar cursos	
2		Muestra lista de cursos registrados
3	Busca o selecciona el curso	
4	Elige la opción listar alumnos curso	
5		Muestra lista de alumnos registrados
Extensiones:		

Nro. Caso de uso:	CU49	
Nombre:	Registra calificación curso	
Descripción:	Se registra la calificación de un curso	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • El curso debe estar registrado • El alumno debe estar registrado • Debe estar habilitado calificación del curso 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción listar cursos	
2		Muestra lista de cursos registrados
3	Busca o selecciona el curso	
4	Elige la opción listar alumnos curso	
5		Muestra lista de alumnos registrados
6	Busca o selecciona el alumno	
7	Elige la opción calificación curso	
8		Muestra formulario de calificación de curso
9	Llena formulario de calificación de	

	curso	
10		Valida datos ingresados
11		Muestra mensaje de calificación registrada satisfactoriamente

Extensiones:

- 4.- Valida datos ingresados
4.1.- Si uno o más campos obligatorios son NULOS, el sistema mostrará un mensaje de error en campos obligatorios.
4.2.- Si uno o más campos no cumplen con las especificaciones establecidas, se mostrará un mensaje de error en formato de campo.
5.- Registra datos en la base de datos
5.1.- Se deberá registrar el vehículo de acuerdo a la ubicación o ciudad a la que pertenece.

Nro. Caso de uso:	CU50
Nombre:	Modifica calificación curso
Descripción:	Se modifica la calificación de un curso
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • El curso debe estar registrado • El alumno debe estar registrado • Debe estar habilitado calificación del curso

Flujo Normal:

	Usuario	Sistema
1	Elige la opción listar cursos	
2		Muestra lista de cursos registrados
3	Busca o selecciona el curso	
4	Elige la opción listar alumnos curso	
5		Muestra lista de alumnos registrados
6	Busca o selecciona el alumno	
7	Elige la opción calificación curso	
8		Muestra formulario de modificación de calificación de curso
9	Llena formulario de calificación de curso	
10		Valida datos ingresados
11		Muestra mensaje de calificación actualizada satisfactoriamente

Extensiones:

- 4.- Valida datos ingresados
4.1.- Si uno o más campos obligatorios son NULOS, el sistema mostrará un mensaje de error en campos obligatorios.
4.2.- Si uno o más campos no cumplen con las especificaciones establecidas, se mostrará un mensaje de error en formato de campo.
5.- Registra datos en la base de datos
5.1.- Se deberá registrar el vehículo de acuerdo a la ubicación o ciudad a la que pertenece.

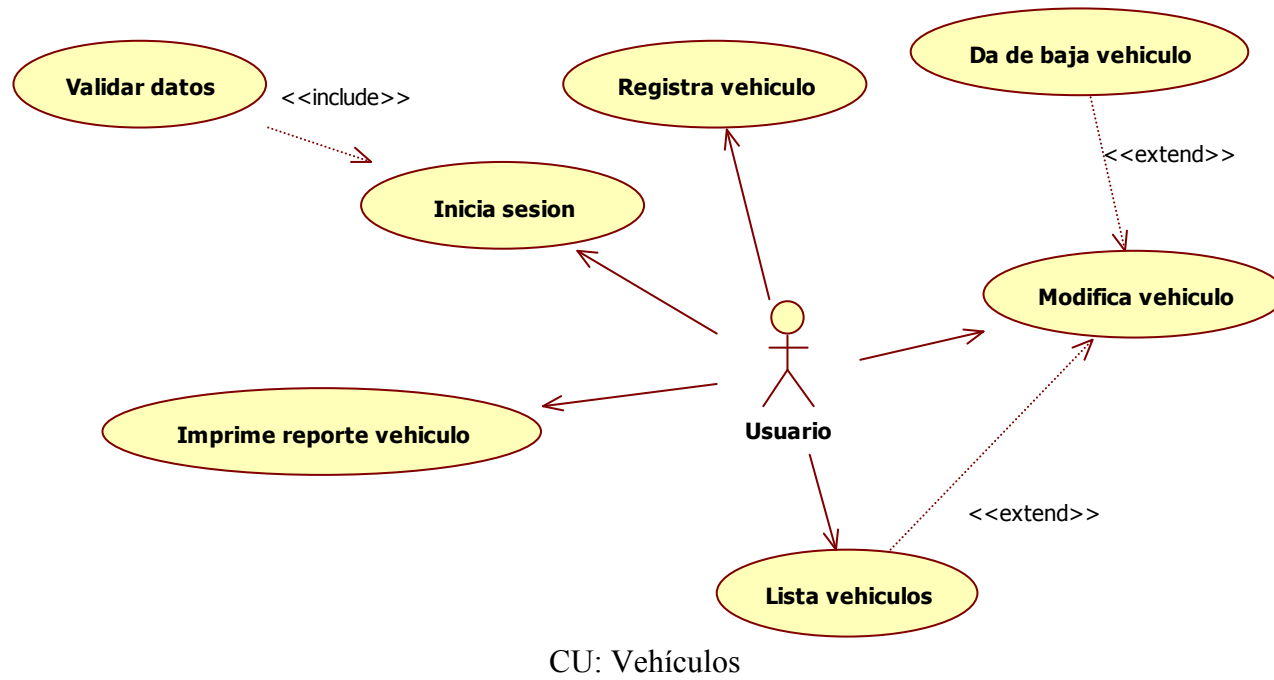
pertenece.

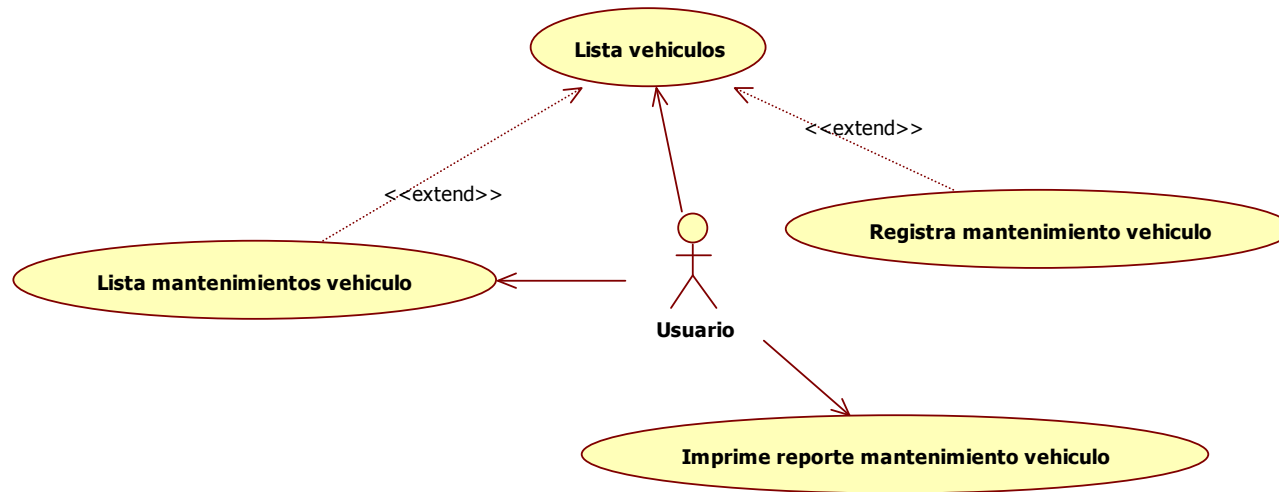
Nro. Caso de uso:	CU51	
Nombre:	Imprime reporte	
Descripción:	Se imprime reporte	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción de imprimir reporte	
2		Imprime reporte
Extensiones:		

5.2.5. MÓDULO VEHÍCULOS

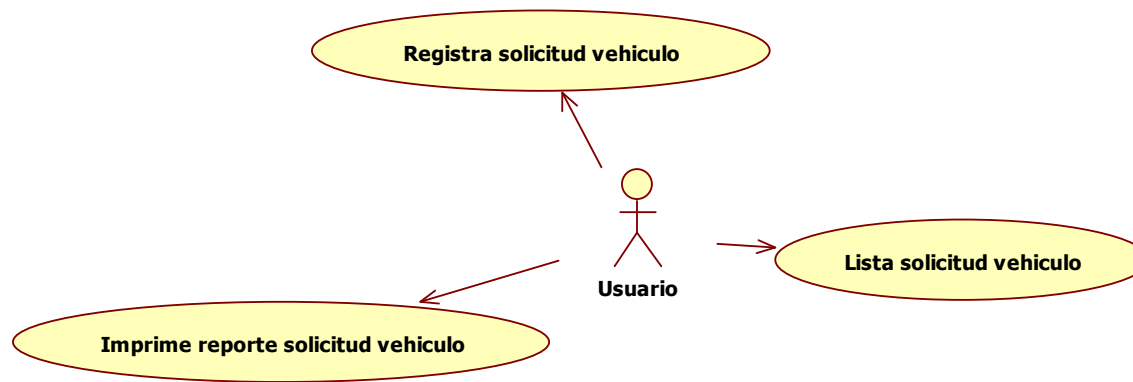
ACTOR NRO.	NOMBRE	FUNCIÓN	
		NRO.	DESCRIPCIÓN
4	Usuario: Vehículos	CU1	Inicia sesión
		CU52	Registra vehículo
		CU53	Modifica vehículo
		CU54	Da de baja vehículo
		CU55	Lista vehículos
		CU56	Imprime reporte vehículo
		CU57	Registra mantenimiento vehículo
		CU58	Lista mantenimientos vehículo
		CU59	Imprime reporte mantenimiento vehículo
		CU60	Registra solicitud vehículo
		CU61	Lista solicitud vehículos
		CU62	Imprime reporte solicitud vehículos
		CU63	Registra orden de movilización
		CU64	Modifica orden de movilización
		CU65	Busca orden movilización
CU66	Lista orden de movilización		
CU67	Imprime reporte orden de movilización		

5.2.5.1. DIAGRAMA CASOS DE USO MÓDULO VEHÍCULOS

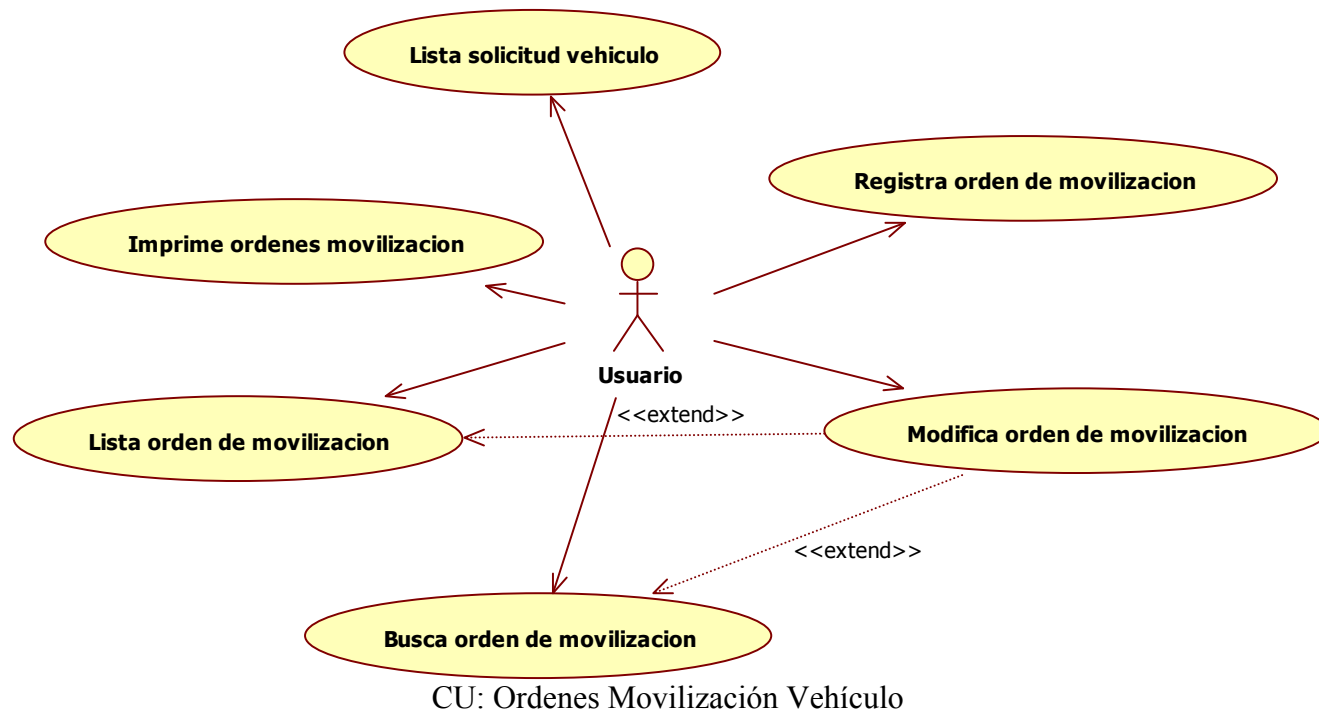




CU: Mantenimientos Vehículo



CU: Solicitudes Vehículos



5.2.5.2. DETALLES CASOS DE USO MÓDULO VEHICULOS

Nro. Caso de uso:	CU52	
Nombre:	Registra vehículo	
Descripción:	Se registra un vehículo	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • El vehículo no debe estar registrado 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción ingresar vehículo	
2		Muestra formulario de registro de vehículo
3	Llena formulario de registro de vehículo	
4		Valida datos ingresados
5		Registra datos en la base de datos
6		Muestra mensaje de vehículo registrado satisfactoriamente
Extensiones:		
<p>4.- Valida datos ingresados</p> <p>4.1.- Si uno o más campos obligatorios son NULOS, el sistema mostrará un mensaje de error en campos obligatorios.</p> <p>4.2.- Si uno o más campos no cumplen con las especificaciones establecidas, se mostrará un mensaje de error en formato de campo.</p> <p>5.- Registra datos en la base de datos</p> <p>5.1.- Se deberá registrar el vehículo de acuerdo a la ubicación o ciudad a la que pertenece.</p>		

Nro. Caso de uso:	CU53	
Nombre:	Modifica vehículo	
Descripción:	Se modifica un vehículo registrado	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • El vehículo debe estar registrado 	
Flujo Normal:		
	Usuario	Sistema
1	Busca o selecciona el vehículo	
2	Elige la opción modificar vehículo	
3		Muestra formulario de modificar vehículo
4	Modifica datos del formulario	
5		Valida datos ingresados
6		Actualiza datos en la base de datos
7		Muestra mensaje de vehículo

	modificado satisfactoriamente
Extensiones:	
5.- Valida datos ingresados 5.1.- Si uno o más campos obligatorios son NULOS, el sistema mostrará un mensaje de error en campos obligatorios. 5.2.- Si uno o más campos no cumplen con las especificaciones establecidas, se mostrará un mensaje de error en formato de campo.	

Nro. Caso de uso:	CU54	
Nombre:	Da de baja vehículo	
Descripción:	Se da de baja a un vehículo	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • El vehículo debe estar registrado 	
Flujo Normal:		
	Usuario	Sistema
1	Busca o selecciona el vehículo	
2	Elige la opción dar de baja vehículo	
3		Muestra mensaje de confirmación
4	Confirma baja de vehículo	
5		Valida datos ingresados
6		Modifica estado de vehículo en la base de datos
7		Muestra mensaje de vehículo eliminado satisfactoriamente
Extensiones:		
5.- Valida datos ingresados 5.1.- Si uno o más campos obligatorios son NULOS, el sistema mostrará un mensaje de error en campos obligatorios. 5.2.- Si uno o más campos no cumplen con las especificaciones establecidas, se mostrará un mensaje de error en formato de campo.		

Nro. Caso de uso:	CU55	
Nombre:	Lista vehículos	
Descripción:	Lista los vehículos registrados	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción listar vehículos	
2		Consulta la base de datos
3		Muestra resultado
Extensiones:		

--

Nro. Caso de uso:	CU56									
Nombre:	Imprime reporte vehículo									
Descripción:	Se imprime reporte de los vehículos registrados									
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 									
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 									
Flujo Normal:										
	<table border="1"> <thead> <tr> <th></th> <th>Usuario</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Elige la opción de imprimir reporte vehículos</td> <td></td> </tr> <tr> <td>2</td> <td></td> <td>Imprime reporte</td> </tr> </tbody> </table>		Usuario	Sistema	1	Elige la opción de imprimir reporte vehículos		2		Imprime reporte
	Usuario	Sistema								
1	Elige la opción de imprimir reporte vehículos									
2		Imprime reporte								
Extensiones:										

Nro. Caso de uso:	CU57																					
Nombre:	Registra mantenimiento vehículo																					
Descripción:	Se registra los mantenimientos de un vehículo																					
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 																					
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 																					
Flujo Normal:																						
	<table border="1"> <thead> <tr> <th></th> <th>Usuario</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Elige la opción ingresar mantenimiento de vehículo</td> <td></td> </tr> <tr> <td>2</td> <td></td> <td>Muestra formulario de registro de mantenimiento de vehículo</td> </tr> <tr> <td>3</td> <td>Llena formulario de registro de mantenimiento de vehículo</td> <td></td> </tr> <tr> <td>4</td> <td></td> <td>Valida datos ingresados</td> </tr> <tr> <td>5</td> <td></td> <td>Registra datos en la base de datos</td> </tr> <tr> <td>6</td> <td></td> <td>Muestra mensaje de mantenimiento de vehículo registrado satisfactoriamente</td> </tr> </tbody> </table>		Usuario	Sistema	1	Elige la opción ingresar mantenimiento de vehículo		2		Muestra formulario de registro de mantenimiento de vehículo	3	Llena formulario de registro de mantenimiento de vehículo		4		Valida datos ingresados	5		Registra datos en la base de datos	6		Muestra mensaje de mantenimiento de vehículo registrado satisfactoriamente
	Usuario	Sistema																				
1	Elige la opción ingresar mantenimiento de vehículo																					
2		Muestra formulario de registro de mantenimiento de vehículo																				
3	Llena formulario de registro de mantenimiento de vehículo																					
4		Valida datos ingresados																				
5		Registra datos en la base de datos																				
6		Muestra mensaje de mantenimiento de vehículo registrado satisfactoriamente																				
Extensiones:																						
<p>4.- Valida datos ingresados</p> <p>4.1.- Si uno o más campos obligatorios son NULOS, el sistema mostrará un mensaje de error en campos obligatorios.</p> <p>4.2.- Si uno o más campos no cumplen con las especificaciones establecidas, se mostrará un mensaje de error en formato de campo.</p>																						

Nro. Caso de uso:	CU58
Nombre:	Lista mantenimientos vehículos
Descripción:	Lista los mantenimientos de un vehículo registrados
Actores:	<ul style="list-style-type: none"> • Usuario

	<ul style="list-style-type: none"> • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción listar mantenimientos vehículo	
2		Consulta la base de datos
3		Muestra resultado
Extensiones:		

Nro. Caso de uso:	CU59	
Nombre:	Imprime reporte mantenimiento vehículo	
Descripción:	Se imprime reporte de los mantenimientos de vehículo registrados	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción de imprimir reporte mantenimiento de vehículos	
2		Imprime reporte
Extensiones:		

Nro. Caso de uso:	CU60	
Nombre:	Registra solicitud vehículo	
Descripción:	Se registra las solicitudes de vehículos	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción ingresar solicitud de vehículo	
2		Muestra formulario de registro de solicitud de vehículo
3	Llena formulario de registro de solicitud de vehículo	
4		Valida datos ingresados
5		Registra datos en la base de datos
6		Muestra mensaje de solicitud de vehículo registrado satisfactoriamente
Extensiones:		

4.- Valida datos ingresados
4.1.- Si uno o más campos obligatorios son NULOS, el sistema mostrará un mensaje de error en campos obligatorios.
4.2.- Si uno o más campos no cumplen con las especificaciones establecidas, se mostrará un mensaje de error en formato de campo.

Nro. Caso de uso:	CU61	
Nombre:	Lista solicitud vehículos	
Descripción:	Lista las solicitudes de un vehículo registrados	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción listar solicitud de vehículo	
2		Consulta la base de datos
3		Muestra resultado
Extensiones:		

Nro. Caso de uso:	CU62	
Nombre:	Imprime reporte solicitud vehículos	
Descripción:	Se imprime reporte de las solicitudes de vehículo registrados	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción de imprimir reporte solicitud de vehículos	
2		Imprime reporte
Extensiones:		

Nro. Caso de uso:	CU63	
Nombre:	Registra orden de movilización	
Descripción:	Se registra las órdenes de movilización de los vehículos	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción ingresar orden de movilización de vehículo	
2		Muestra formulario de registro de

		orden de movilización de vehículo
3	Llena formulario de registro de orden de movilización de vehículo	
4		Valida datos ingresados
5		Registra datos en la base de datos
6		Muestra mensaje de orden de movilización de vehículo registrado satisfactoriamente

Extensiones:

- 4.- Valida datos ingresados
4.1.- Si uno o más campos obligatorios son NULOS, el sistema mostrará un mensaje de error en campos obligatorios.
4.2.- Si uno o más campos no cumplen con las especificaciones establecidas, se mostrará un mensaje de error en formato de campo.

Nro. Caso de uso:	CU64
Nombre:	Modifica orden movilización
Descripción:	Se modifica una orden de movilización de vehículo registrado
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • La orden de movilización de vehículo debe estar registrada

Flujo Normal:

	Usuario	Sistema
1	Busca o selecciona la orden de movilización de vehículo	
2	Elige la opción modificar orden movilización	
3		Muestra formulario de modificar orden movilización vehículo
4	Modifica datos del formulario	
5		Valida datos ingresados
6		Actualiza datos en la base de datos
7		Muestra mensaje de orden movilización de vehículo modificado satisfactoriamente

Extensiones:

- 5.- Valida datos ingresados
5.1.- Si uno o más campos obligatorios son NULOS, el sistema mostrará un mensaje de error en campos obligatorios.
5.2.- Si uno o más campos no cumplen con las especificaciones establecidas, se mostrará un mensaje de error en formato de campo.

Nro. Caso de uso:	CU65
Nombre:	Busca orden movilización

Descripción:	Se realiza la búsqueda de una orden de movilización de vehículo	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción de buscar orden movilización de vehículo	
2		Muestra formulario de búsqueda de orden de movilización de vehículo
3	Ingresar dato a buscar	
4		Valida dato
5		Consulta la base de datos
6		Muestra resultado
Extensiones:		
4.- Valida datos ingresados		
4.1.- Si uno o más campos obligatorios son NULOS, el sistema mostrará un mensaje de error en campos obligatorios.		
4.2.- Si uno o más campos no cumplen con las especificaciones establecidas, se mostrará un mensaje de error en formato de campo.		

Nro. Caso de uso:	CU66	
Nombre:	Lista orden movilización	
Descripción:	Lista las ordenes de movilización de vehículo registrados	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción listar solicitud de vehículo	
2		Consulta la base de datos
3		Muestra resultado
Extensiones:		

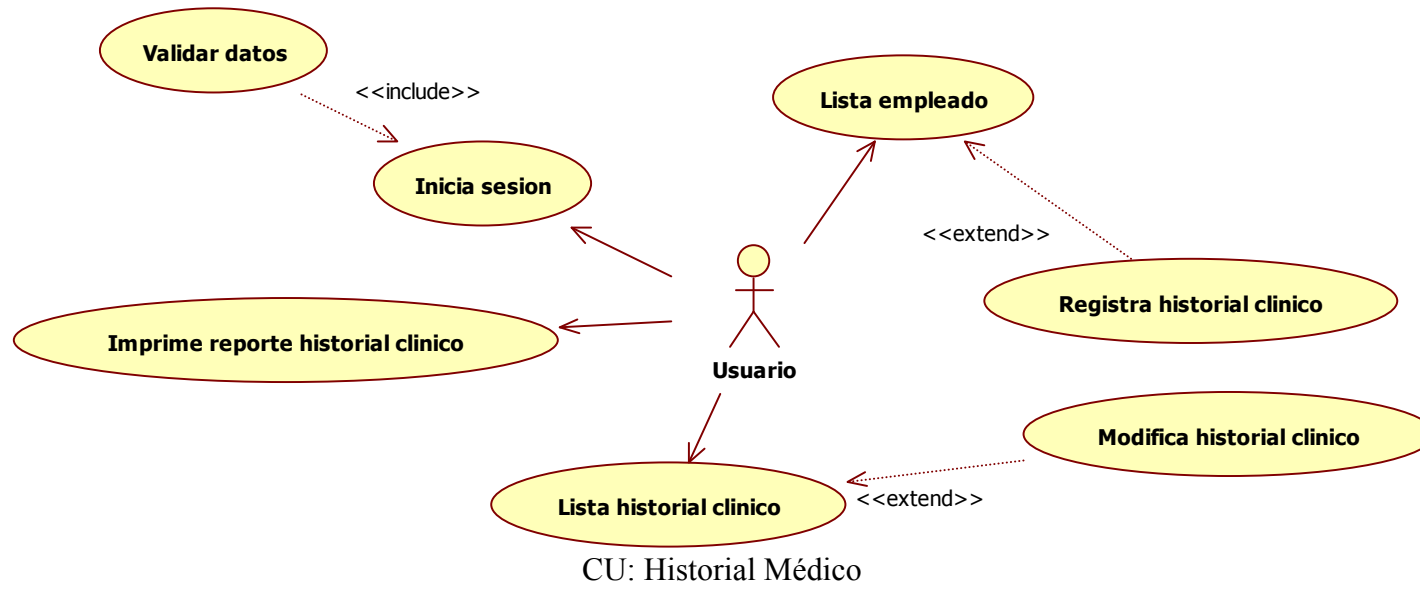
Nro. Caso de uso:	CU67	
Nombre:	Imprime reporte orden de movilización	
Descripción:	Se imprime reporte de las orden movilización registrados	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		

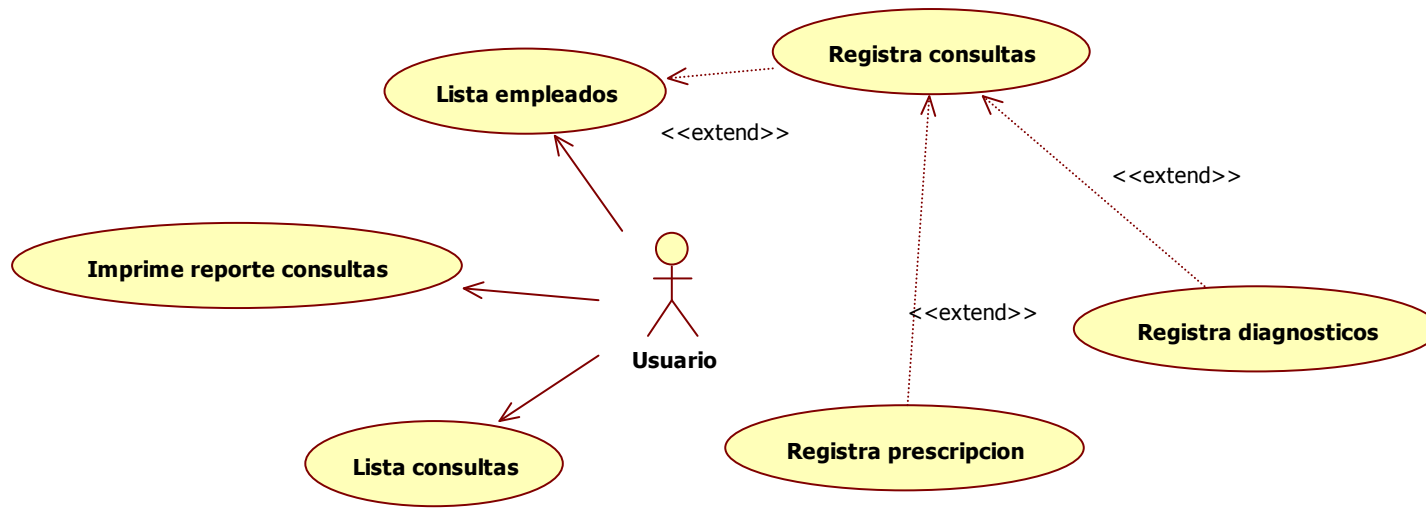
	Usuario	Sistema
1	Elige la opción de imprimir reporte orden movilización	
2		Imprime reporte
Extensiones:		

5.2.6. MÓDULO CONTROL MEDICO

ACTOR NRO.	NOMBRE	FUNCIÓN	
		NRO.	DESCRIPCIÓN
5	Usuario: Médico	CU1	Inicia sesión
		CU21	Lista empleado
		CU68	Registra historial médico
		CU69	Modifica historial médico
		CU70	Lista historial médico
		CU71	Imprime reporte historial médico
		CU72	Registra consulta
		CU73	Registra diagnostico
		CU74	Registra prescripción
		CU75	Lista consultas
		CU76	Imprime reporte consultas

5.2.6.1. DIAGRAMA CASOS DE USO MÓDULO CONTROL MEDICO





CU: Consultas Médicas

5.2.6.2. DETALLES CASOS DE USO MÓDULO CONTROL MEDICO

Nro. Caso de uso:	CU68	
Nombre:	Registra historial médico	
Descripción:	Se registra el historial médico de un empleado	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción ingresar historial médico	
2		Muestra formulario de registro de historial médico
3	Llena formulario de registro de historial médico	
4		Valida datos ingresados
5		Registra datos en la base de datos
6		Muestra mensaje de historial médico registrado satisfactoriamente
Extensiones:		
4.- Valida datos ingresados		
4.1.- Si uno o más campos obligatorios son NULOS o no cumplen con las especificaciones establecidas, el sistema mostrará un mensaje de error en campos obligatorios.		

Nro. Caso de uso:	CU69	
Nombre:	Modifica historial médico	
Descripción:	Se modifica un historial médico	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión • El historial médico debe estar registrada 	
Flujo Normal:		
	Usuario	Sistema
1	Busca o selecciona el empleado	
2	Elige la opción modificar historial médico	
3		Muestra formulario de modificar historial médico
4	Modifica datos del formulario	
5		Valida datos ingresados
6		Actualiza datos en la base de datos
7		Muestra mensaje de historial médico modificado satisfactoriamente
Extensiones:		
5.- Valida datos ingresados		

5.1.- Si uno o más campos obligatorios son NULOS o no cumplen con las especificaciones establecidas, el sistema mostrará un mensaje de error en campos obligatorios.

Nro. Caso de uso:	CU70	
Nombre:	Lista historial médico	
Descripción:	Lista los historiales médicos registrados	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción listar historial médico	
2		Consulta la base de datos
3		Muestra resultado
Extensiones:		

Nro. Caso de uso:	CU71	
Nombre:	Imprime reporte historial médico	
Descripción:	Se imprime reporte de historiales médicos	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción de imprimir reporte historial médico	
2		Imprime reporte
Extensiones:		

Nro. Caso de uso:	CU72	
Nombre:	Registra consulta	
Descripción:	Se registran las consultas médicas de los empelados	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción ingresar consulta médica	
2		Muestra formulario de registro de consulta médica
3	Llena formulario de registro de	

	consulta médica	
4		Valida datos ingresados
5		Registra datos en la base de datos
6		Muestra mensaje de consulta médica registrada satisfactoriamente
Extensiones:		
4.- Valida datos ingresados		
4.1.- Si uno o más campos obligatorios son NULOS o no cumplen con las especificaciones establecidas, el sistema mostrará un mensaje de error en campos obligatorios.		

Nro. Caso de uso:	CU73	
Nombre:	Registra diagnostico	
Descripción:	Se registran los diagnósticos de las consultas médicas	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		
	Usuario	Sistema
1	Busca o selecciona consulta medica	
2	Elige la opción agregar diagnóstico	
3		Muestra formulario de agregar diagnóstico
4	Registra datos del formulario	
5		Valida datos ingresados
6		Registra datos en la base de datos
7		Muestra mensaje de diagnóstico registrado satisfactoriamente
Extensiones:		
4.- Valida datos ingresados		
4.1.- Si uno o más campos obligatorios son NULOS o no cumplen con las especificaciones establecidas, el sistema mostrará un mensaje de error en campos obligatorios.		

Nro. Caso de uso:	CU74	
Nombre:	Registra prescripción	
Descripción:	Se registran las prescripción médicas	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		
	Usuario	Sistema
1	Busca o selecciona consulta médica	
2	Elige la opción agregar prescripción	

3		Muestra formulario de agregar prescripción
4	Registra datos del formulario	
5		Valida datos ingresados
6		Registra datos en la base de datos
7		Muestra mensaje de prescripción registrado satisfactoriamente
Extensiones:		
4.- Valida datos ingresados 4.1.- Si uno o más campos obligatorios son NULOS o no cumplen con las especificaciones establecidas, el sistema mostrará un mensaje de error en campos obligatorios.		

Nro. Caso de uso:	CU75	
Nombre:	Lista consultas	
Descripción:	Lista las consultas médicas registrados	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción listar consultas médicas	
2		Consulta la base de datos
3		Muestra resultado
Extensiones:		

Nro. Caso de uso:	CU76	
Nombre:	Imprime reporte consultas	
Descripción:	Se imprime reporte de consultas médicas	
Actores:	<ul style="list-style-type: none"> • Usuario • Sistema 	
Precondiciones:	<ul style="list-style-type: none"> • Iniciar sesión 	
Flujo Normal:		
	Usuario	Sistema
1	Elige la opción de imprimir reporte consultas médicas	
2		Imprime reporte
Extensiones:		

5.3. DIAGRAMAS DE SECUENCIA

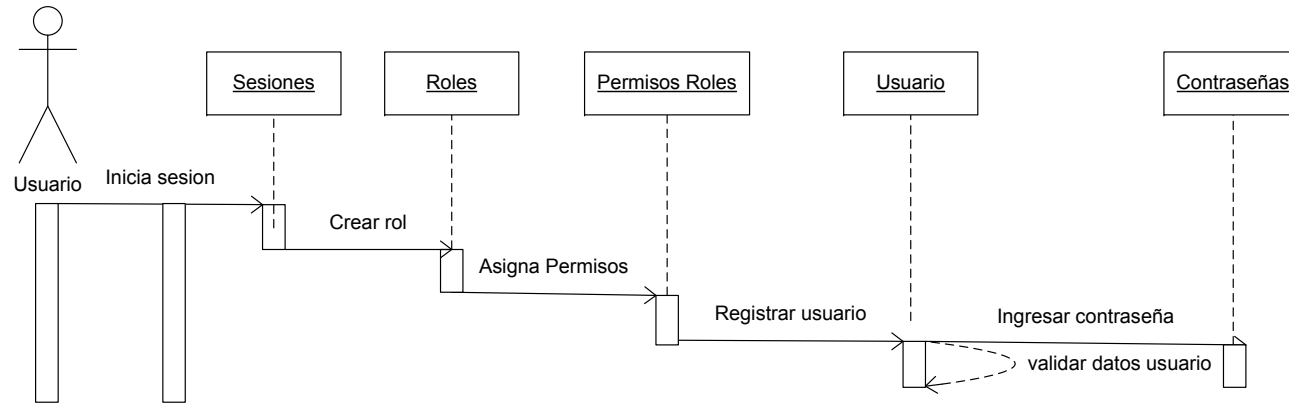


Diagrama de Secuencia 1: Usuario

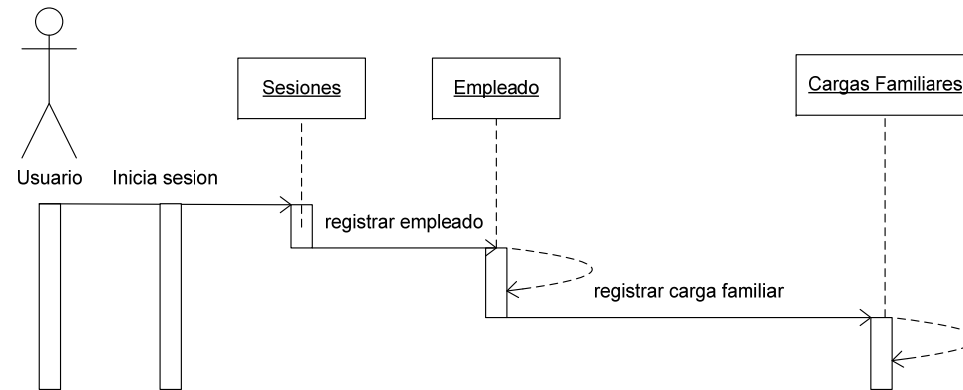


Diagrama de Secuencia 2: Empleados

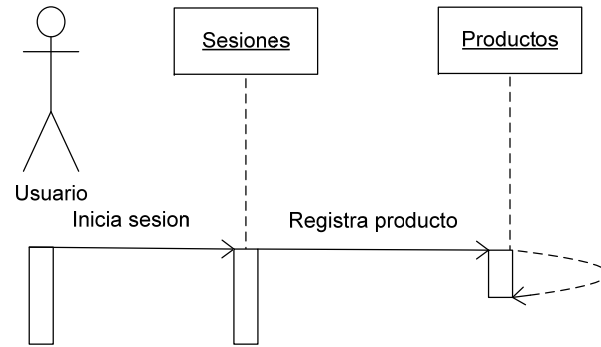


Diagrama de Secuencia 3: Productos

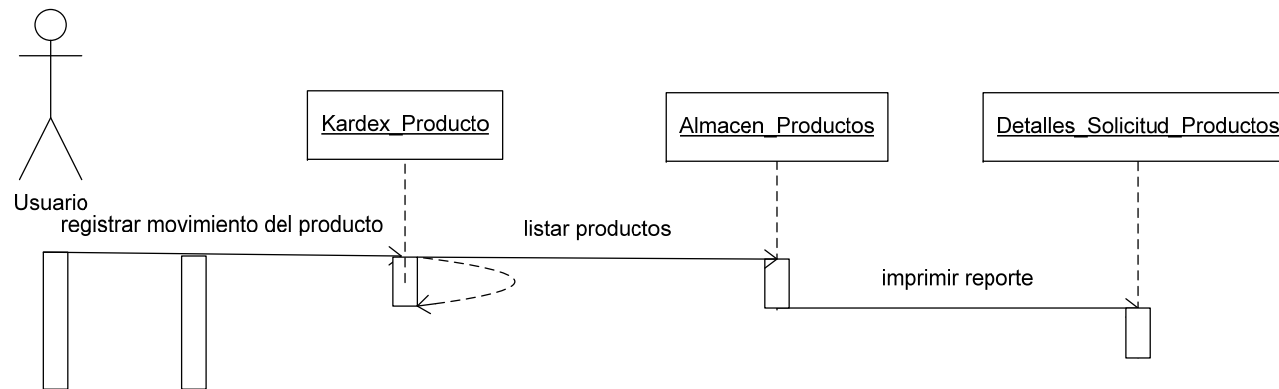


Diagrama de Secuencia 4: Almacén Productos

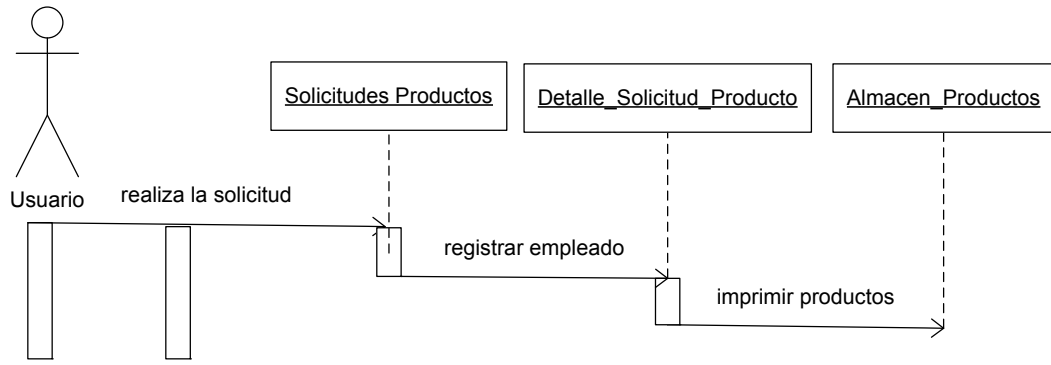


Diagrama de Secuencia 5: Solicitud Productos

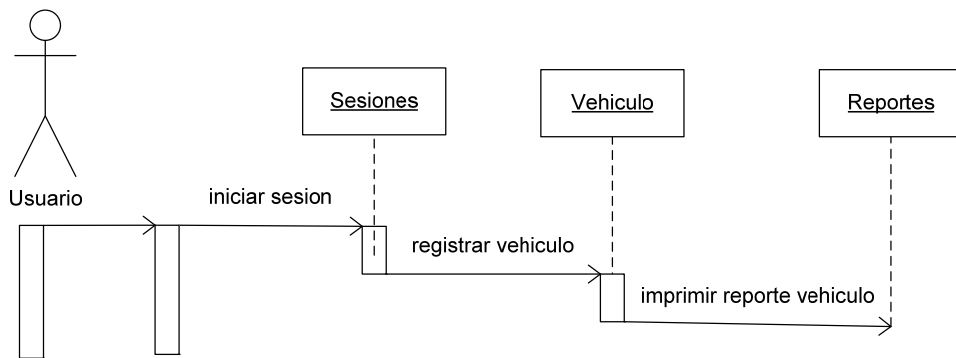


Diagrama de Secuencia 6: Vehículos

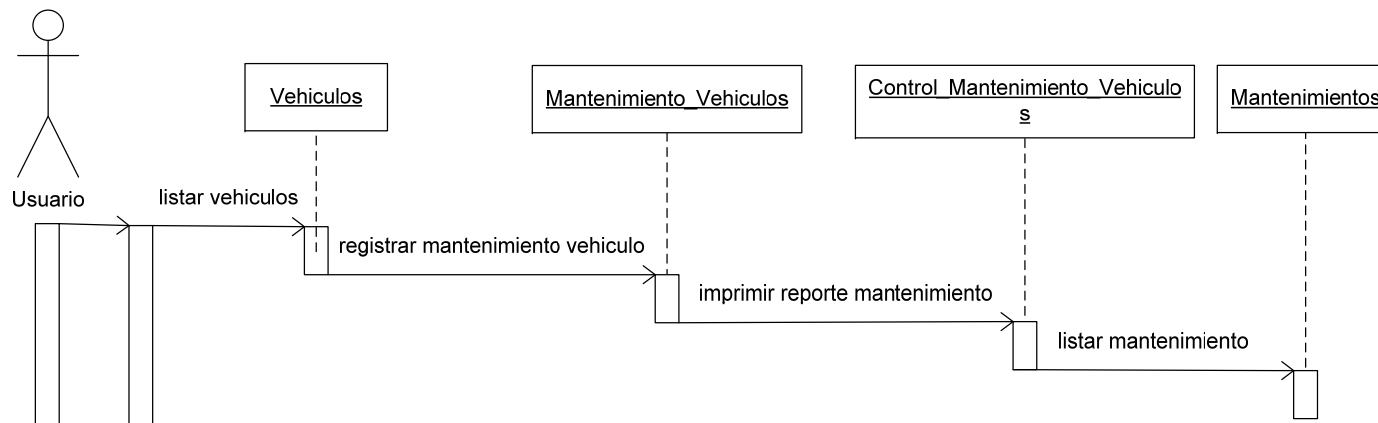


Diagrama de Secuencia 7: Mantenimientos Vehículo.

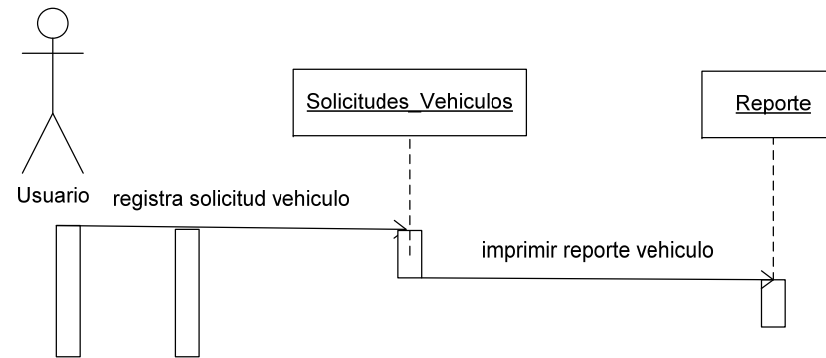


Diagrama de Secuencia 8: Solicitud Vehículos.

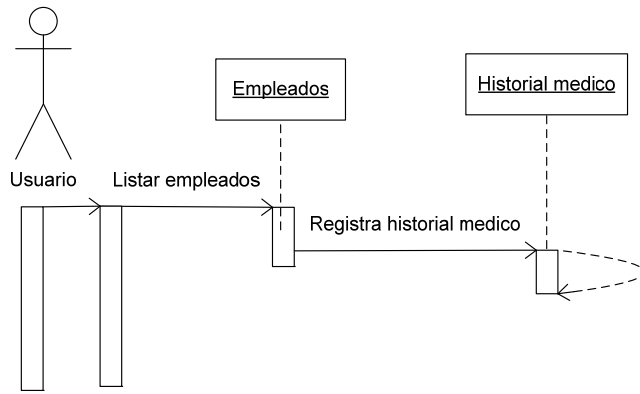


Diagrama de Secuencia 9: Historial Médico.

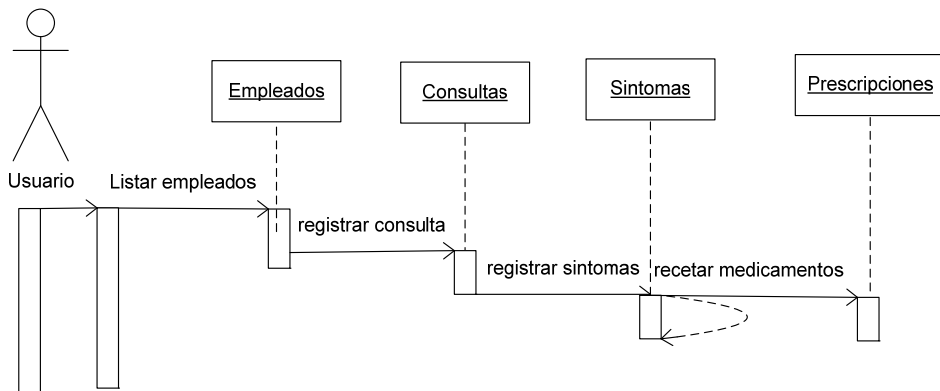


Diagrama de Secuencia 10: Consultas Médicas.

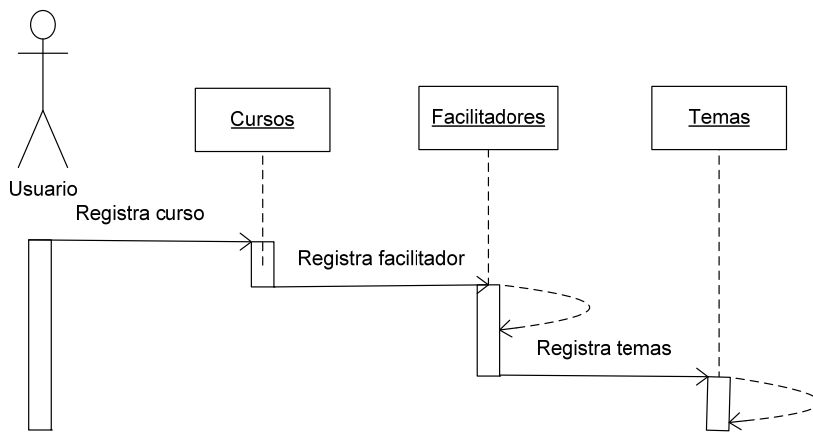


Diagrama de Secuencia 11: Cursos Capacitación.

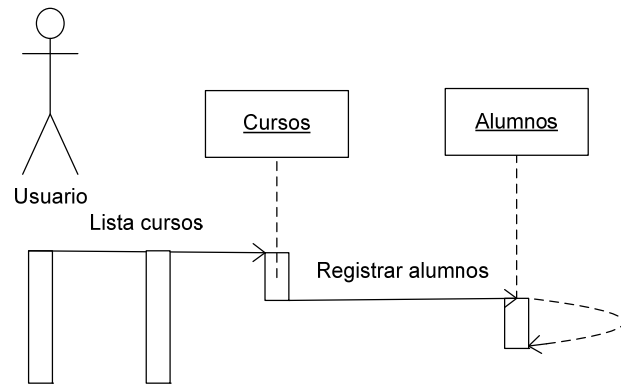


Diagrama de Secuencia 12: Alumnos.

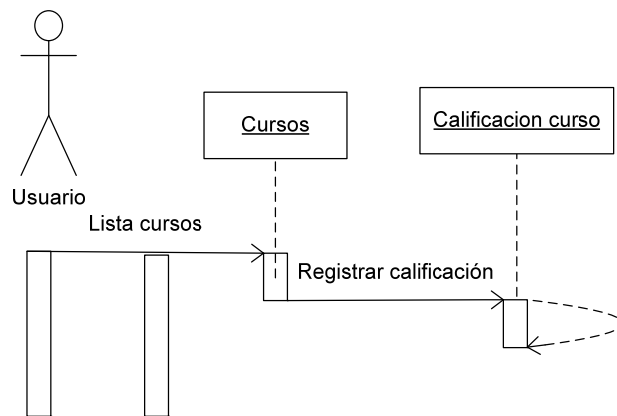


Diagrama de Secuencia 13: Calificación curso.

5.4. DIAGRAMAS DE PROCESOS

5.4.1. MÓDULO USUARIOS

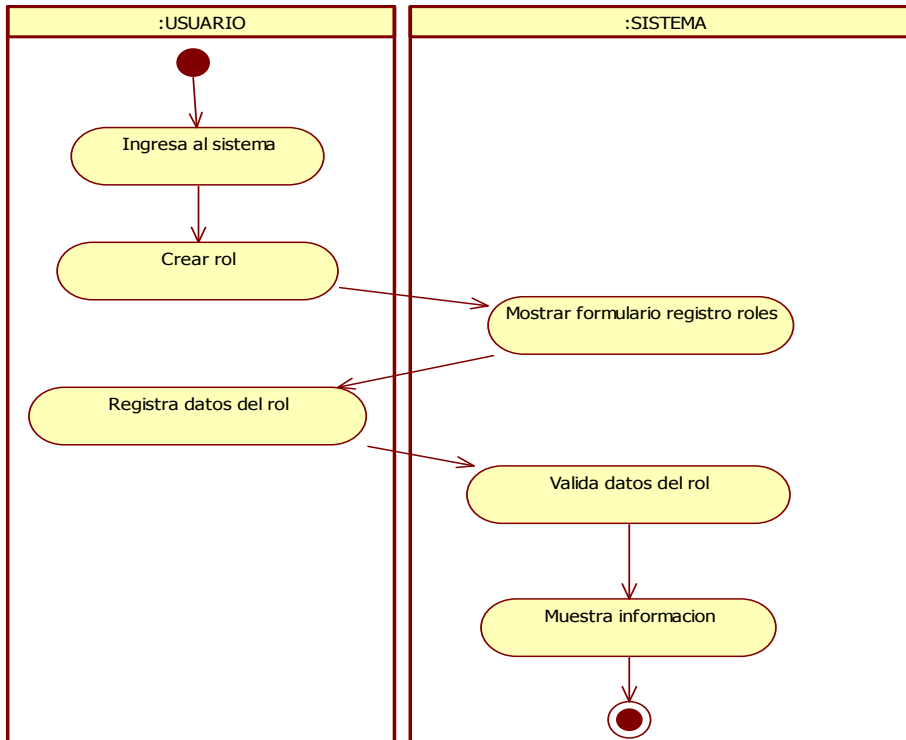


Diagrama de Procesos 1: Registro de roles.

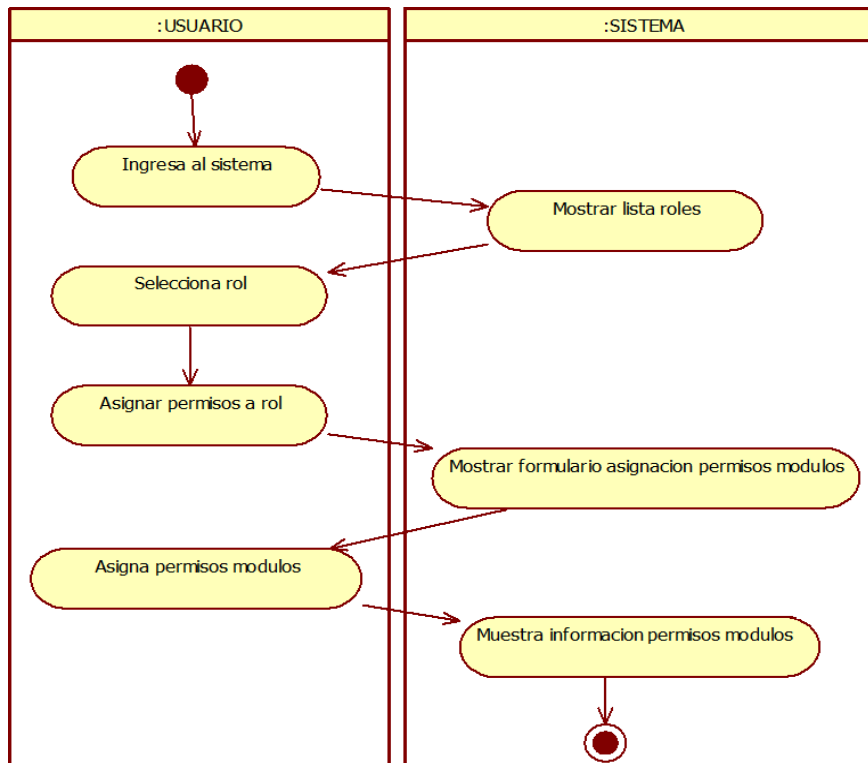


Diagrama de Procesos 2: Asignación de permisos a módulos.

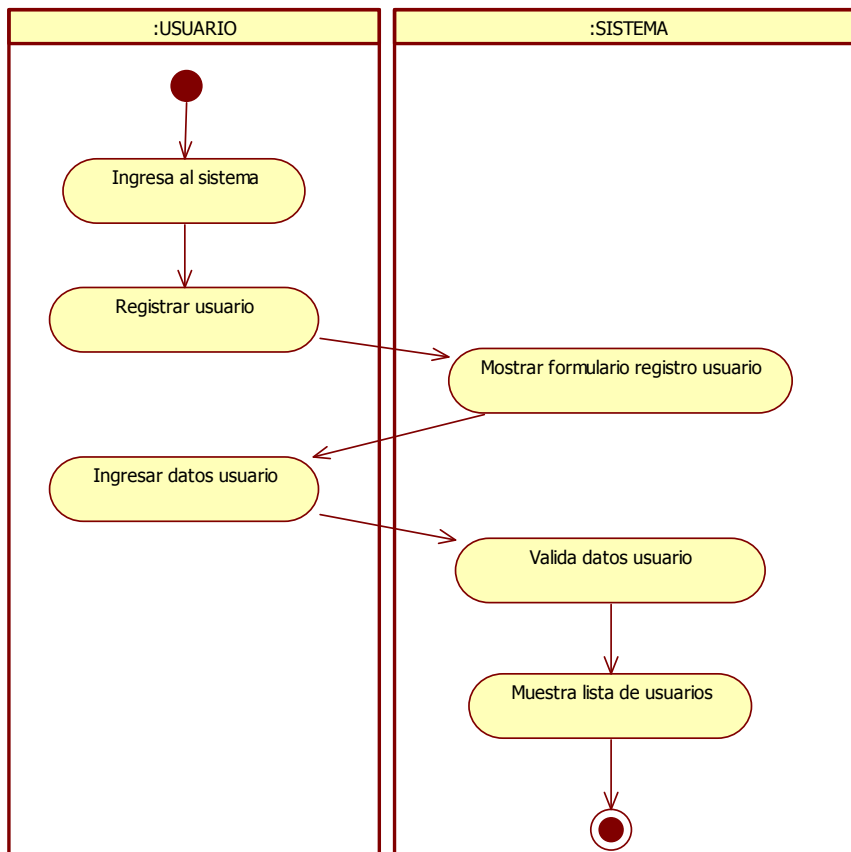


Diagrama de Procesos 3: Creación de usuario.

5.4.2. MÓDULO EMPLEADOS

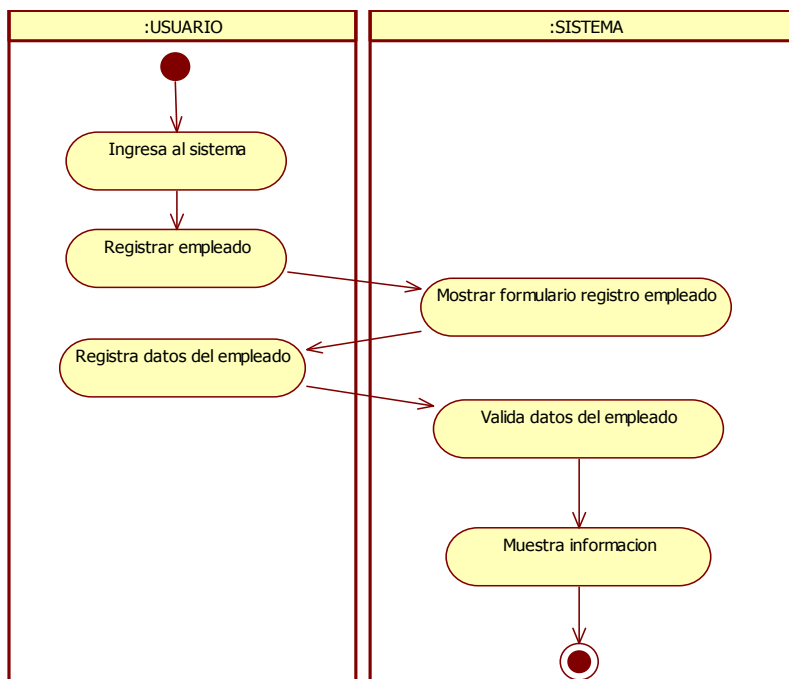


Diagrama de Procesos 4: Gestión Empleado.

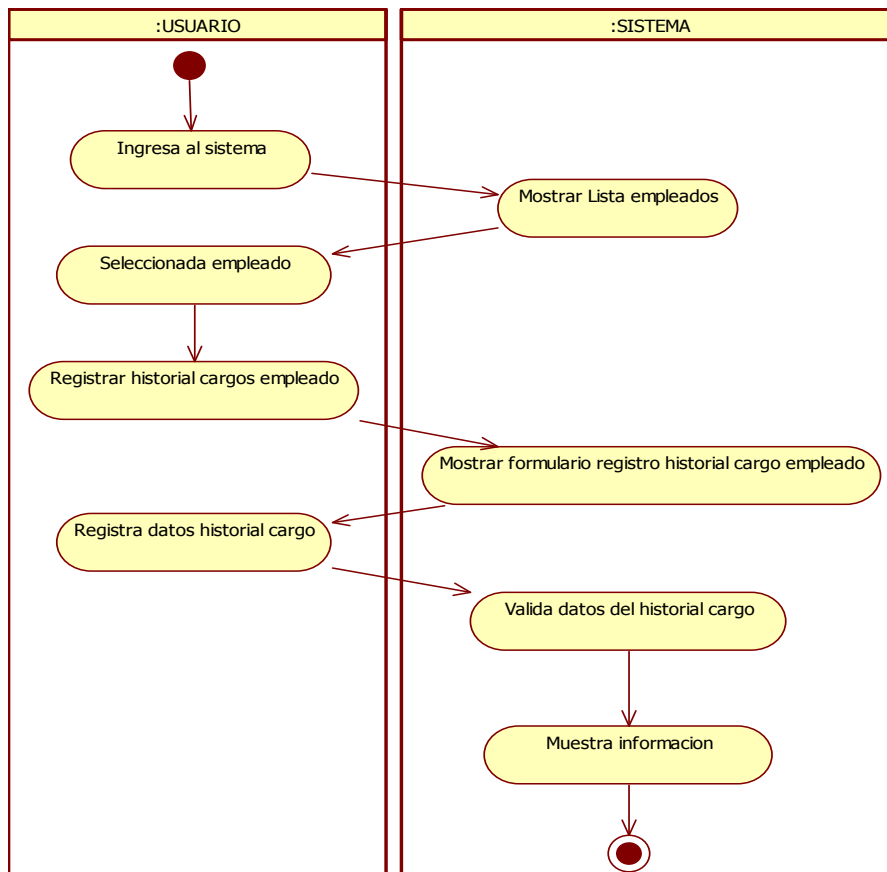


Diagrama de Procesos 5: Gestión Historial Cargos

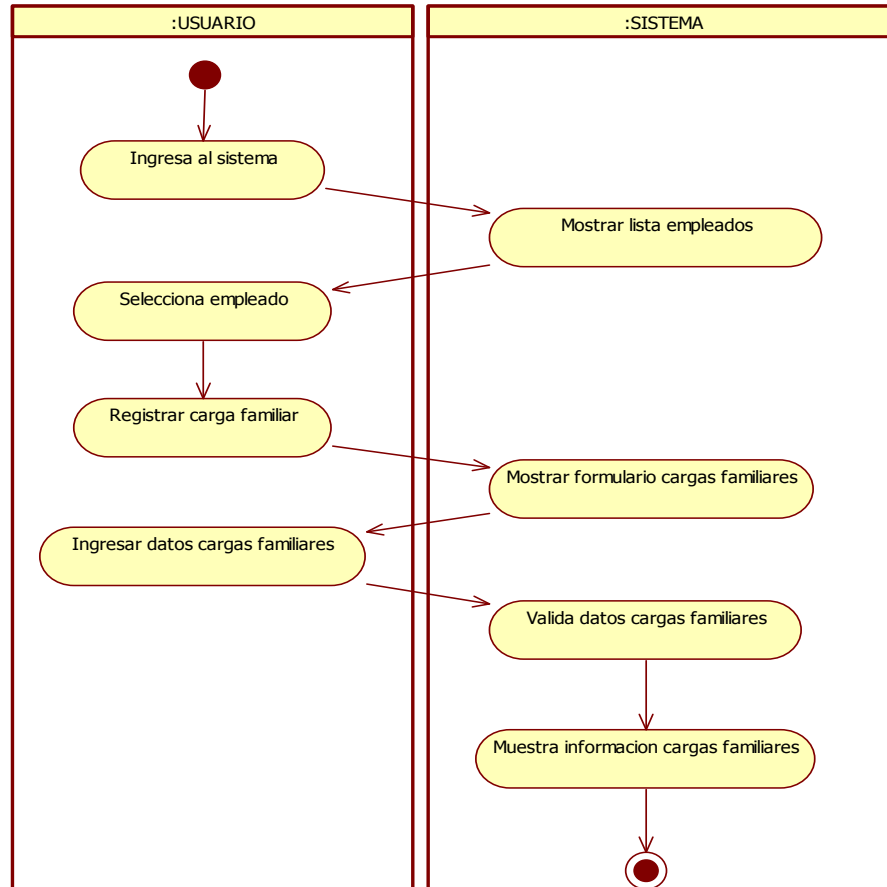


Diagrama de Procesos 6: Gestión Cargas Familiares

5.4.3. MÓDULO PRODUCTOS

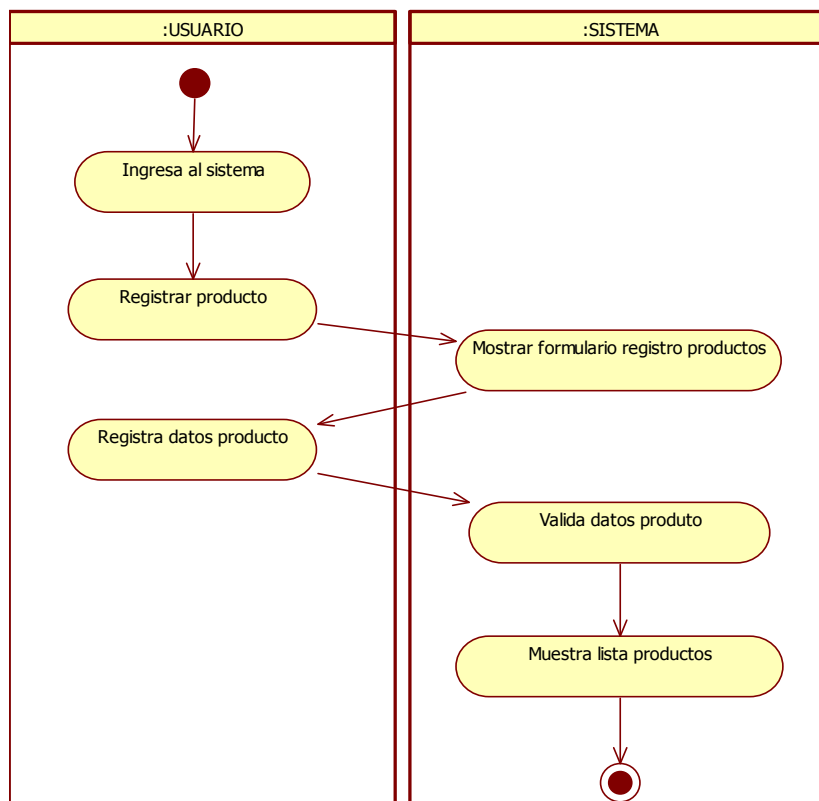


Diagrama de Procesos 7: Registro producto.

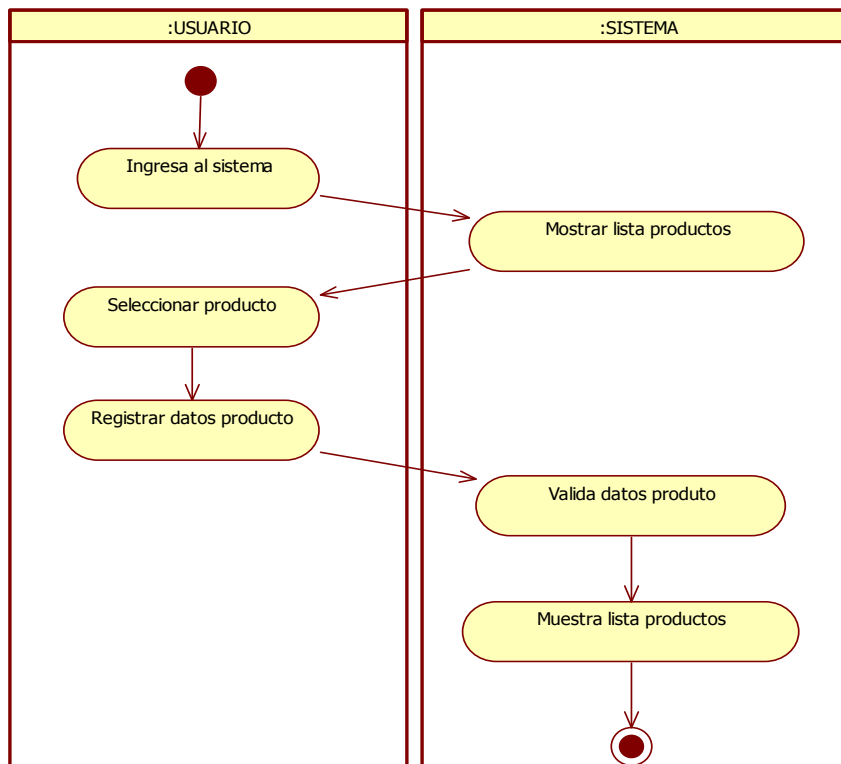


Diagrama de Procesos 8: Registro Producto Almacén.

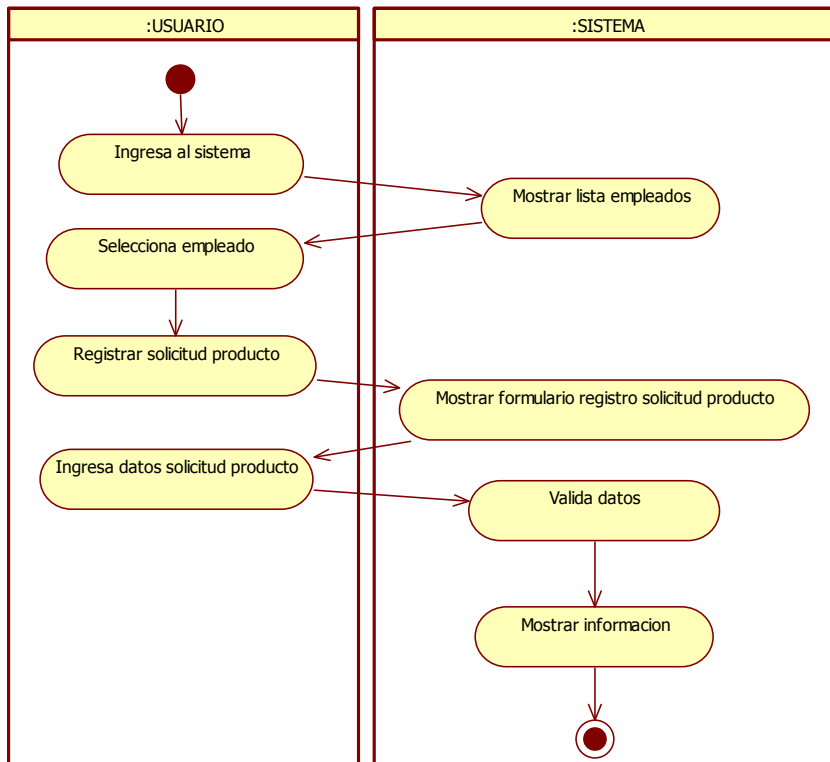


Diagrama de Procesos 9: Solicitud Producto.

5.4.4. MÓDULO CURSOS

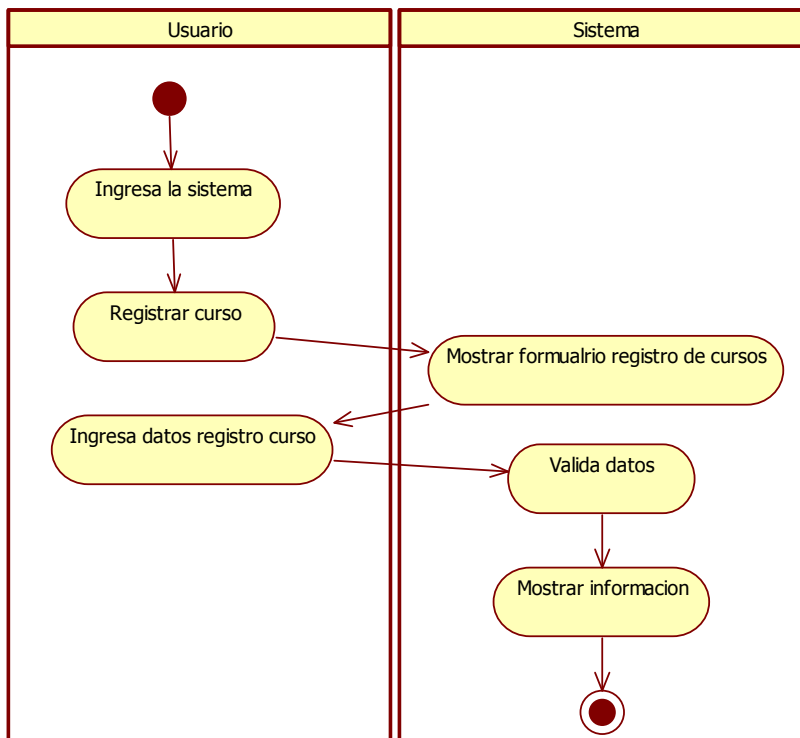


Diagrama de Procesos 10: Registro Curso.

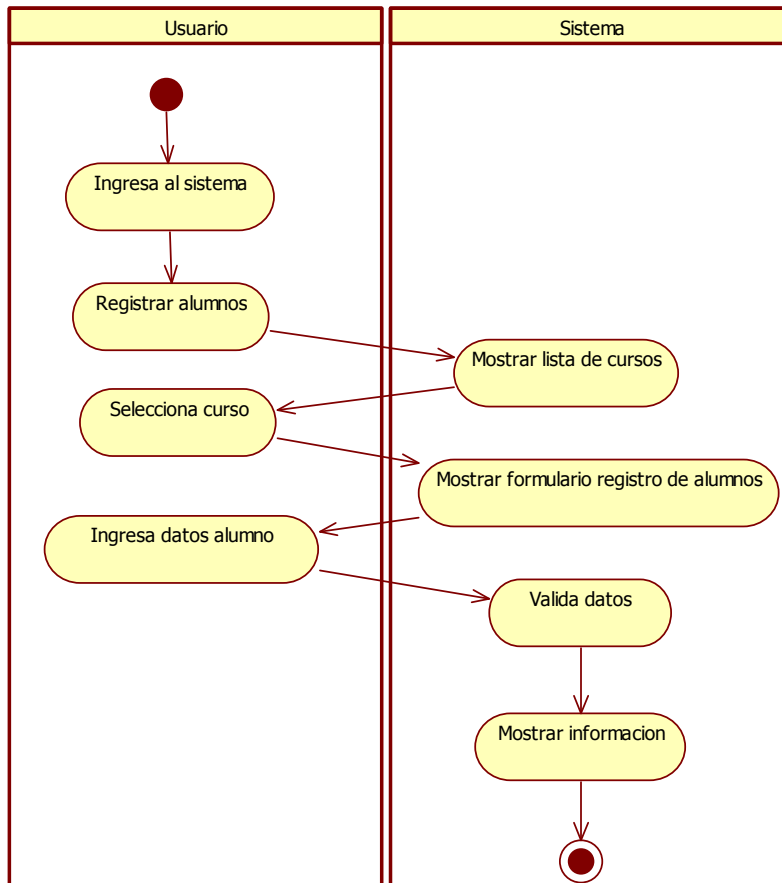


Diagrama de Procesos 11: Registro alumnos.

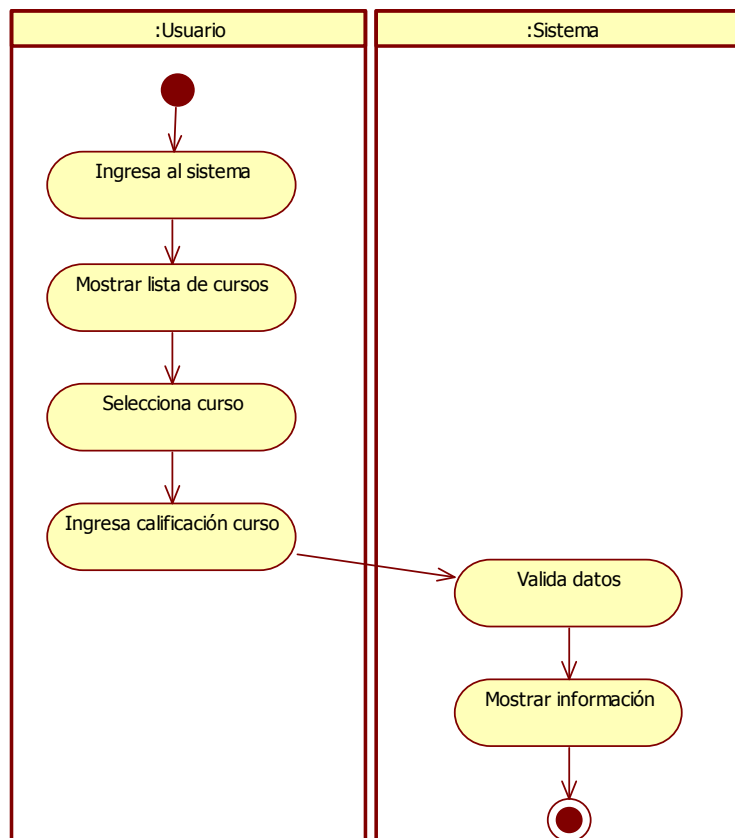


Diagrama de Procesos 12: Registro Calificación de cursos.

5.4.5. MÓDULO VEHÍCULOS

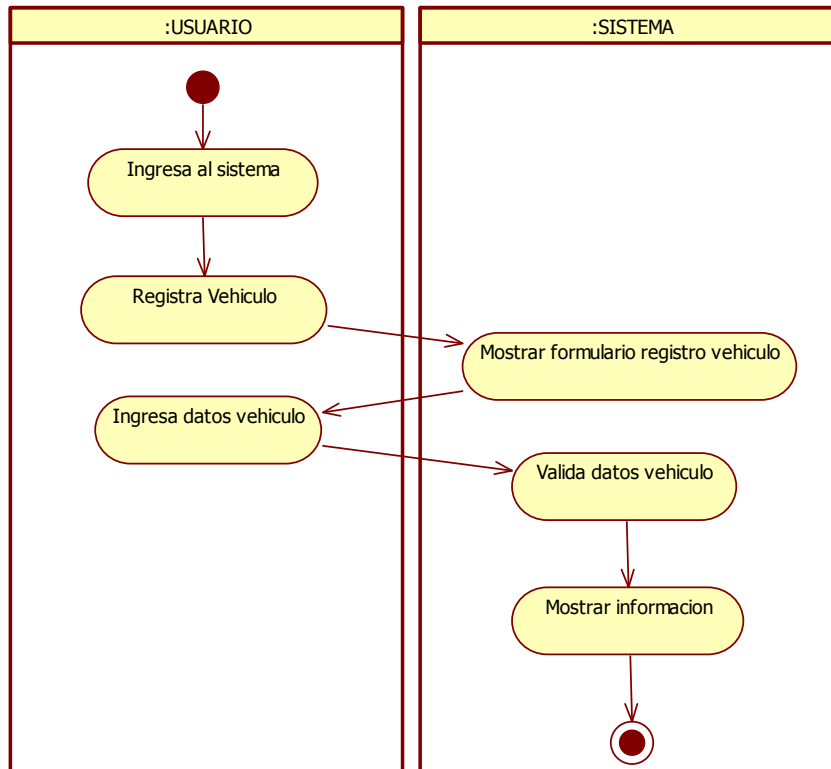


Diagrama de Procesos 13: Registro vehículo.

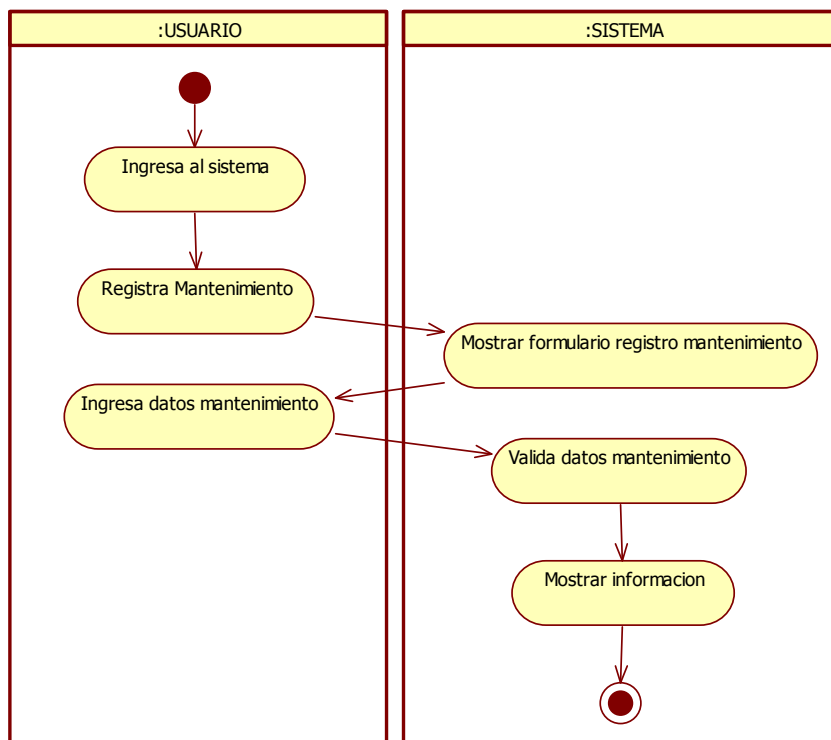


Diagrama de Procesos 14: Registro mantenimientos vehículo.

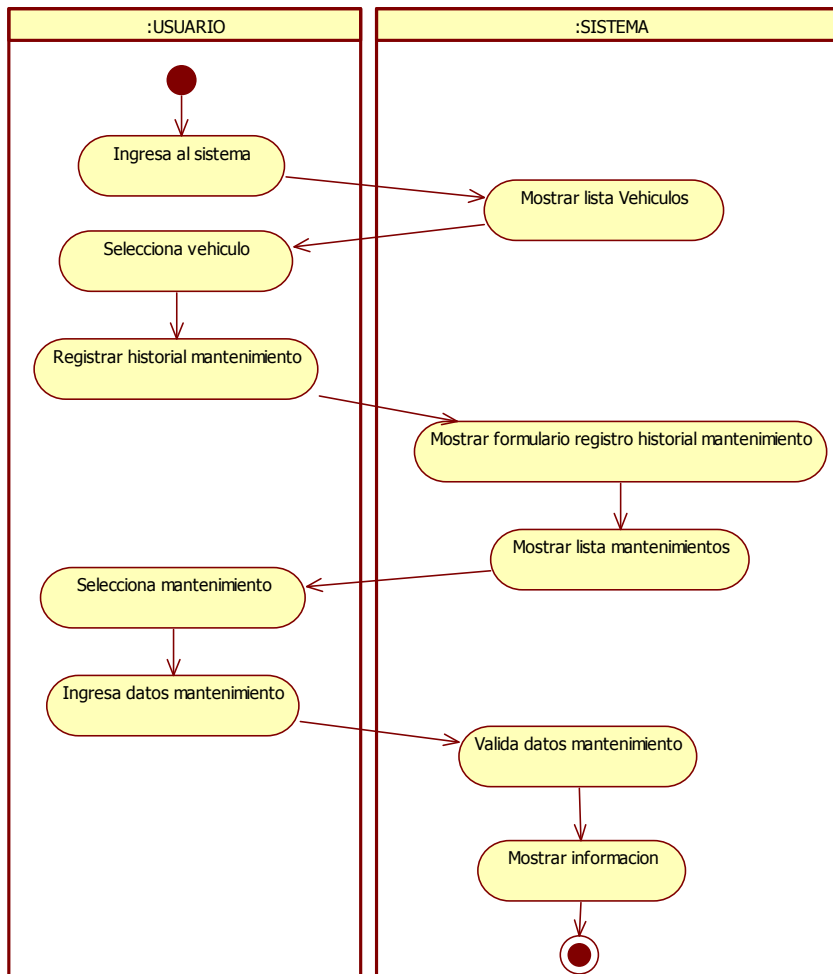


Diagrama de Procesos 15: Registró historial mantenimientos vehículo.

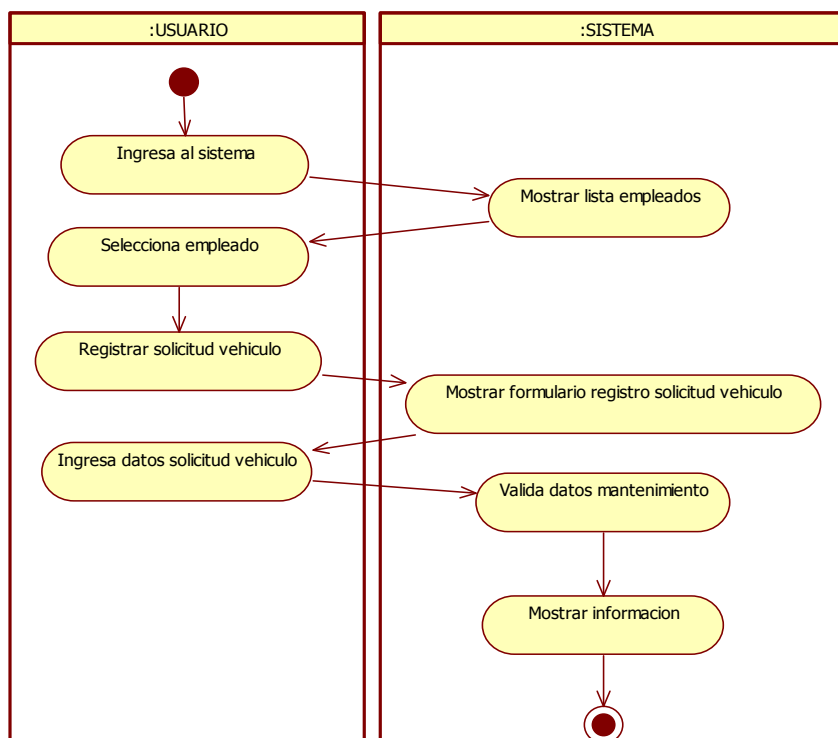


Diagrama de Procesos 16: Solicitud vehículo.

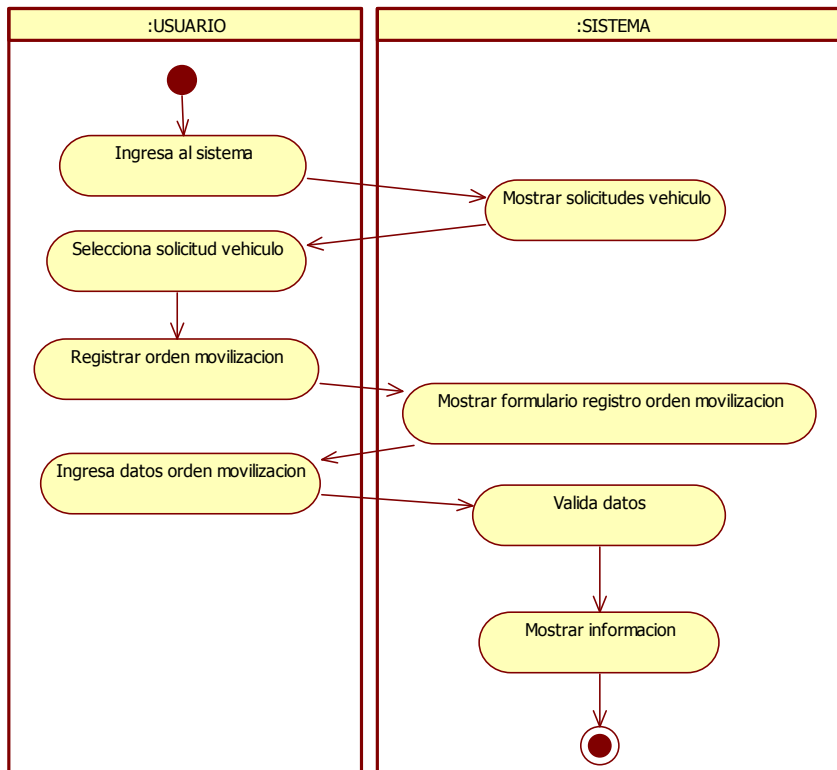


Diagrama de Procesos 17: Orden movilización vehículo.

5.4.6. MÓDULO CONTROL MEDICO

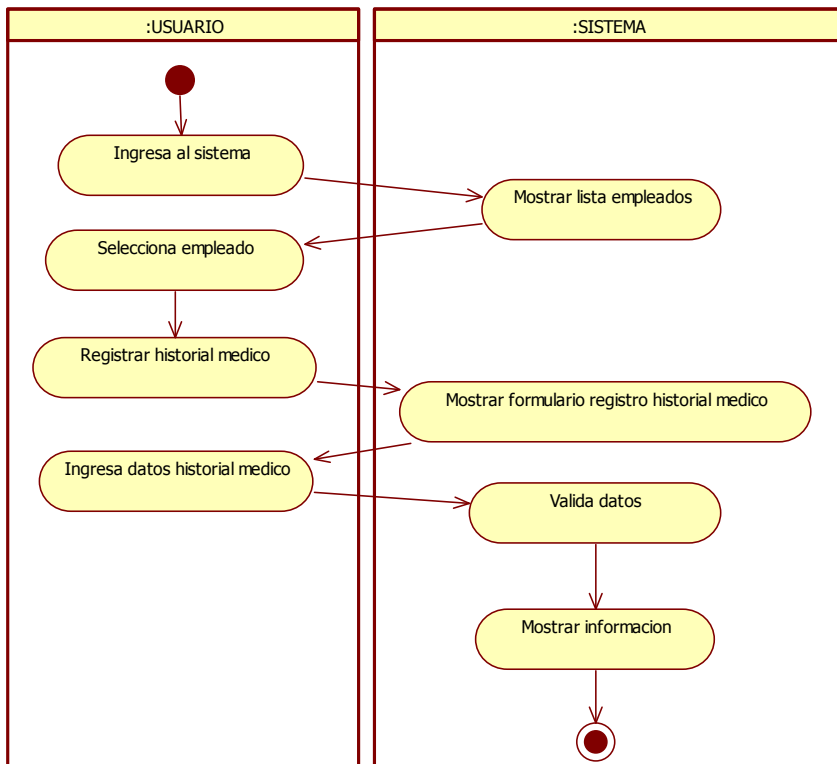


Diagrama de Procesos 18: Historial médico.

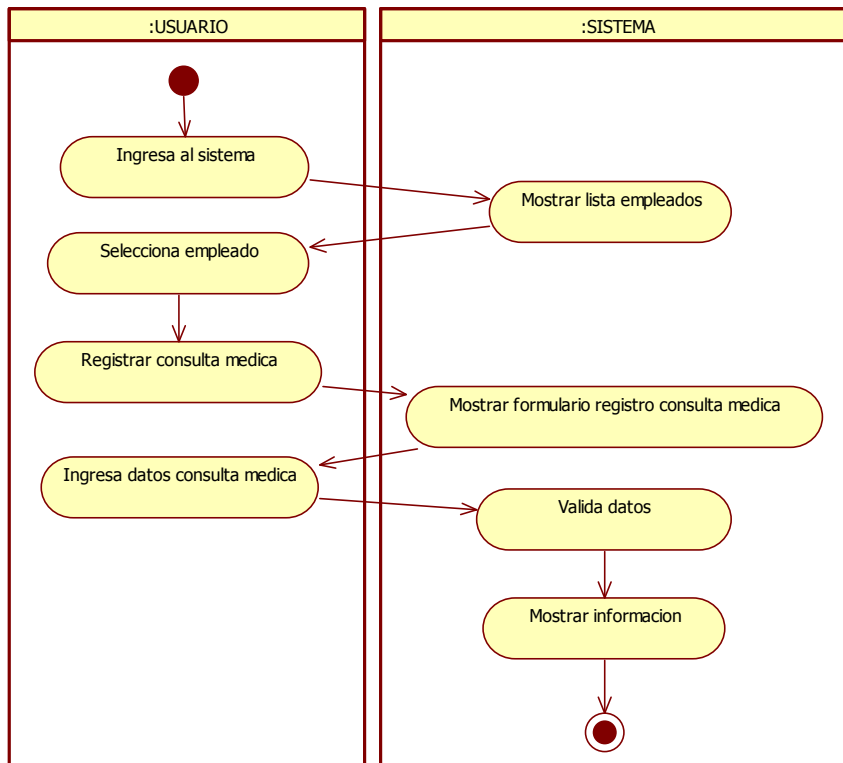
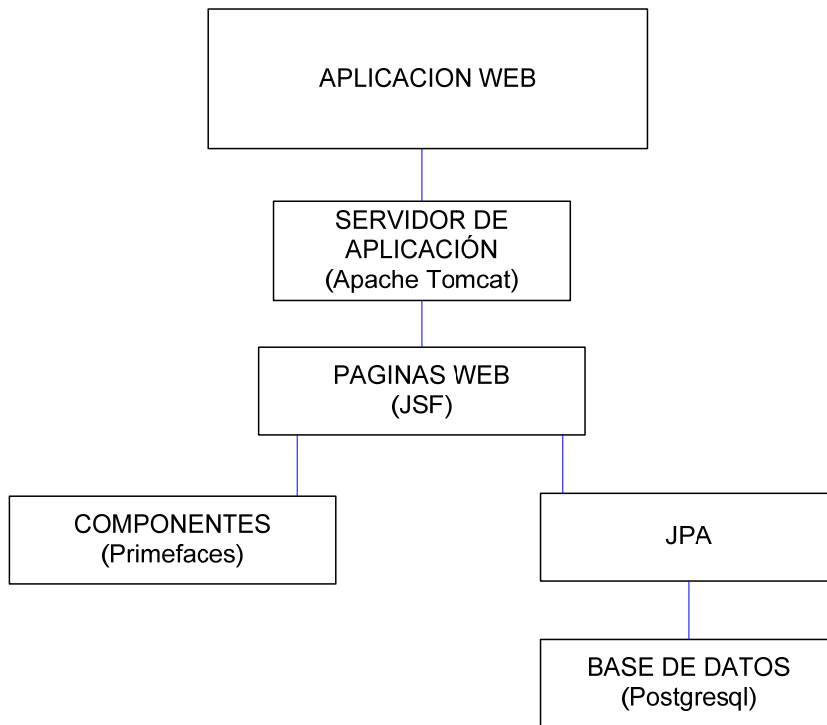


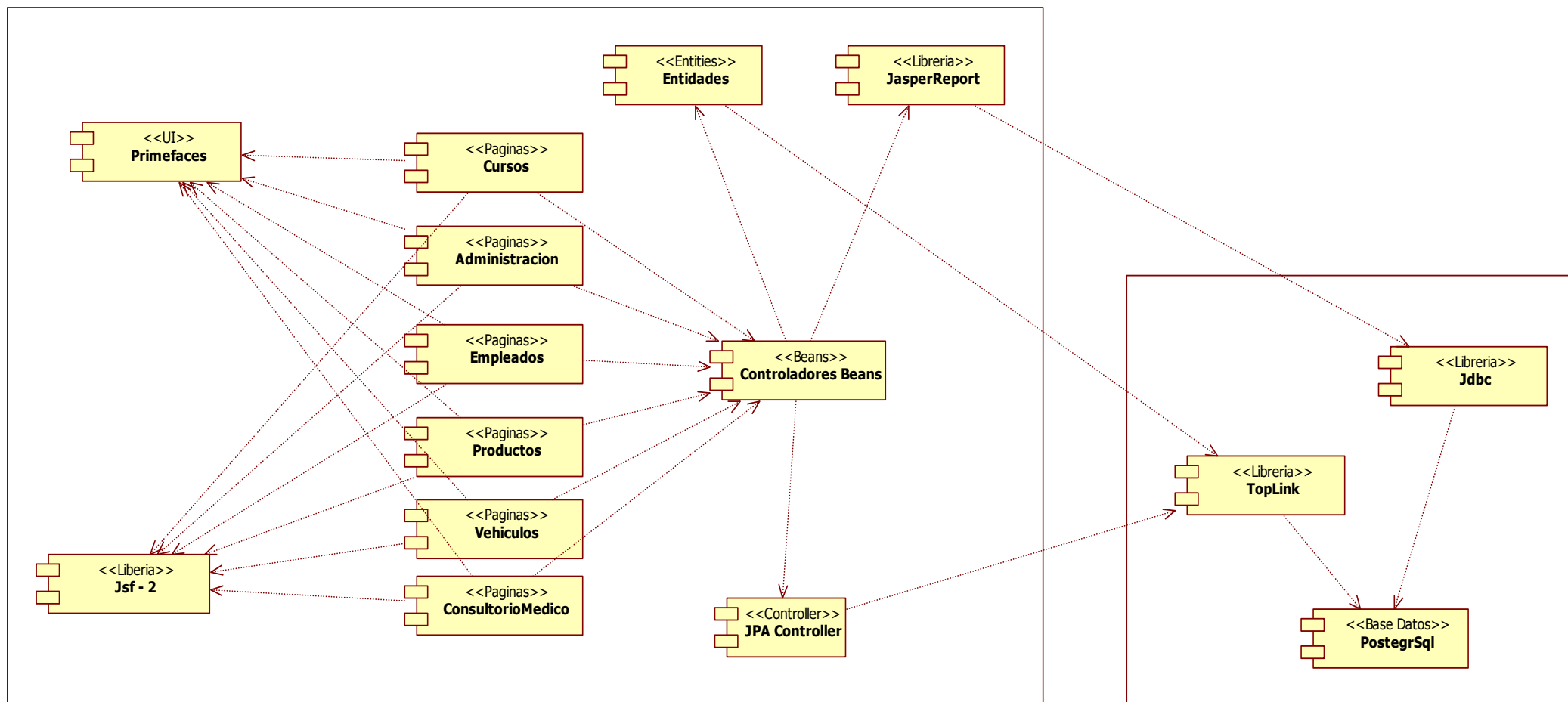
Diagrama de Procesos 19: Consulta médica.

5.5. MODELO CONCEPTUAL



5.6. DIAGRAMA DE CLASES

5.7. MODELO DE COMPONENTES



5.8. DIAGRAMA DE NAVEGACIÓN

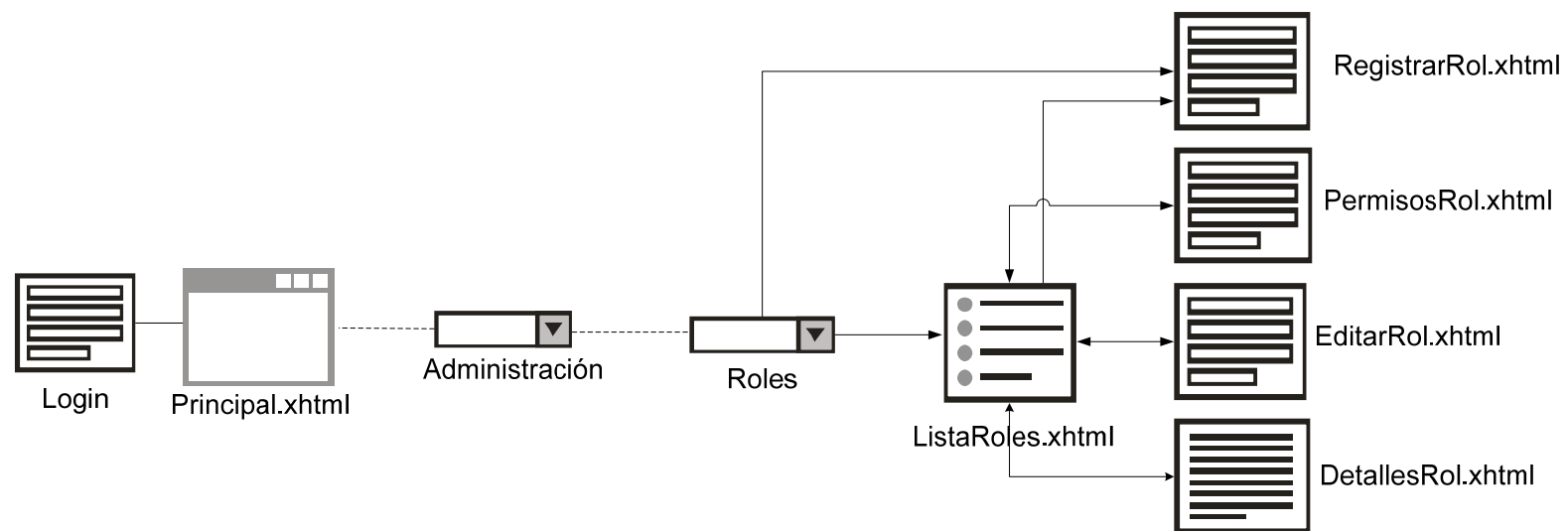


Diagrama Navegación 1: Gestión roles

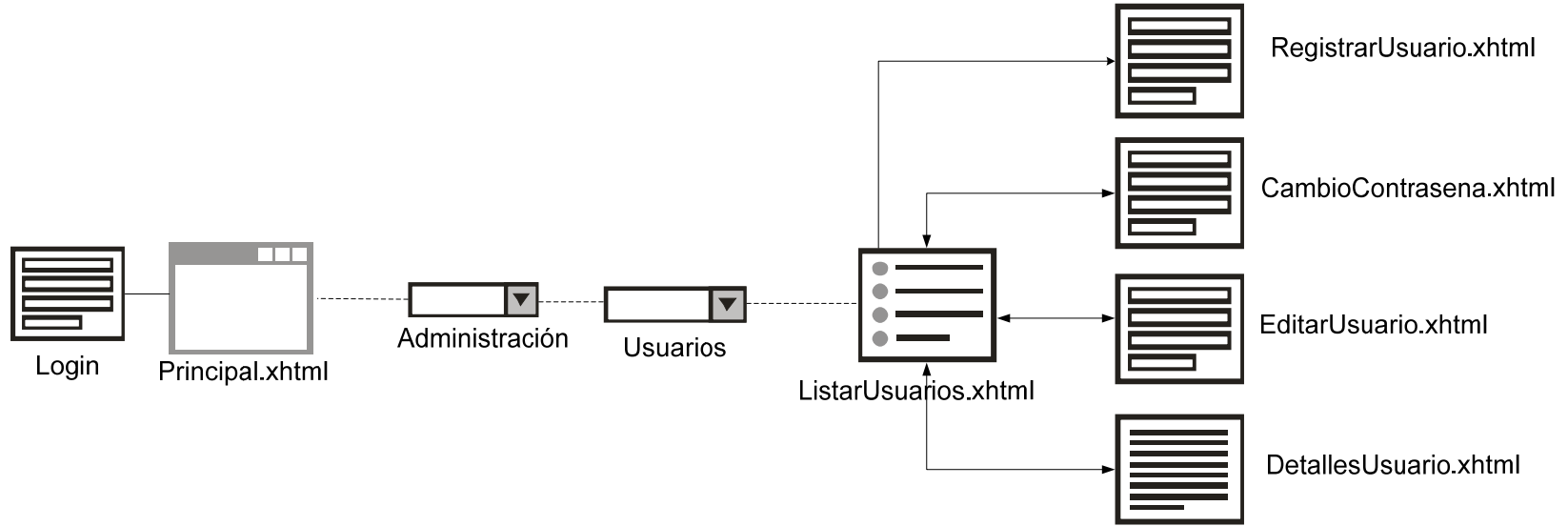


Diagrama Navegación 2: Gestión usuarios.

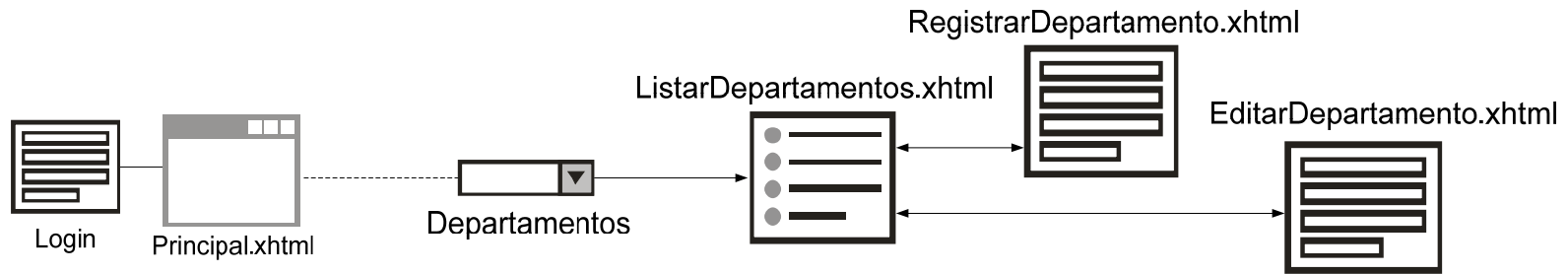


Diagrama Navegación 3: Gestión departamentos.

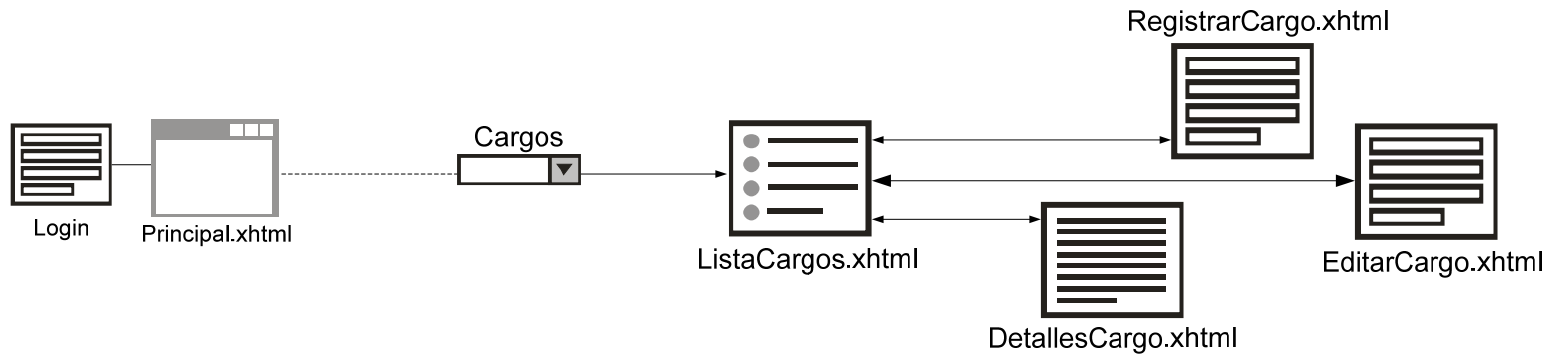


Diagrama Navegación 4: Gestión cargos.

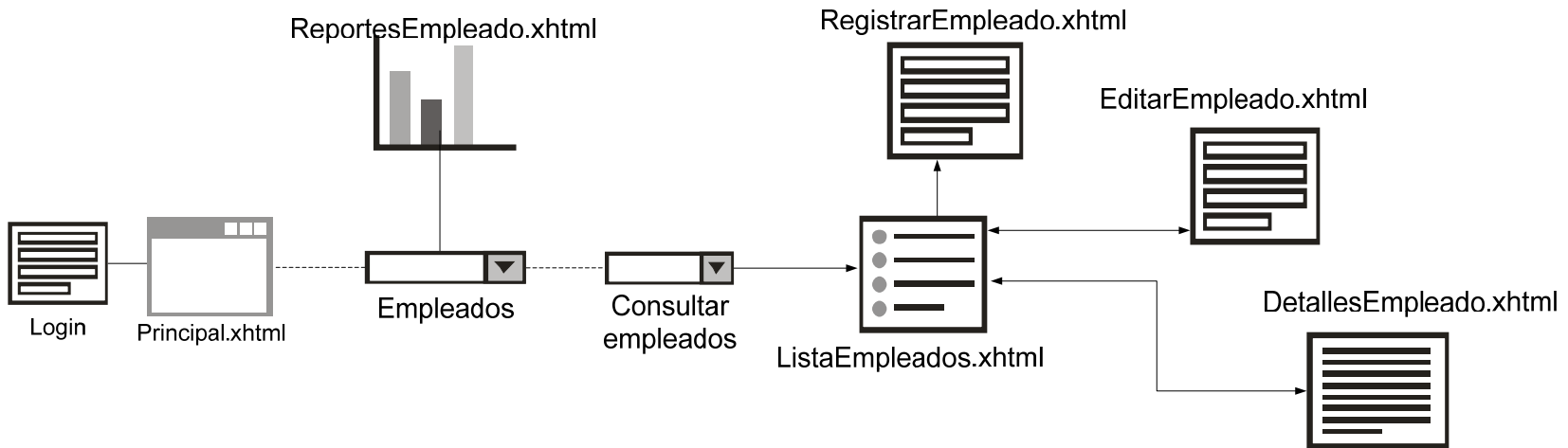


Diagrama Navegación 5: Gestión empleados.

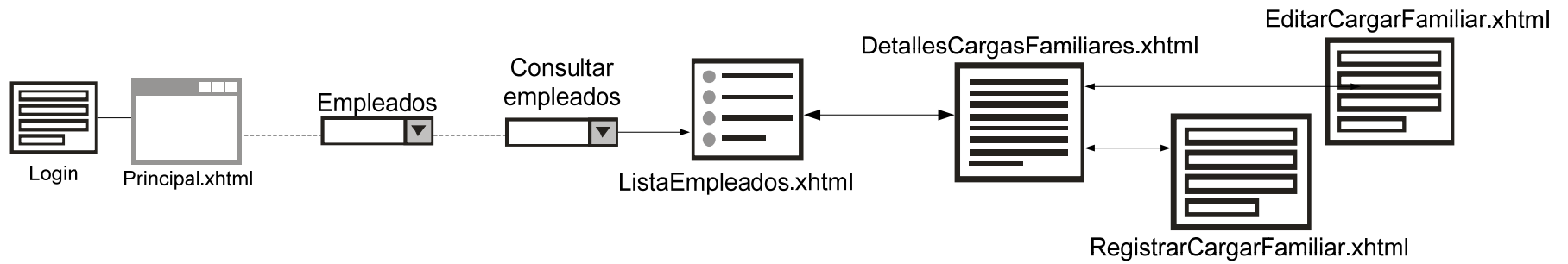


Diagrama Navegación 6: Gestión cargas familiares.

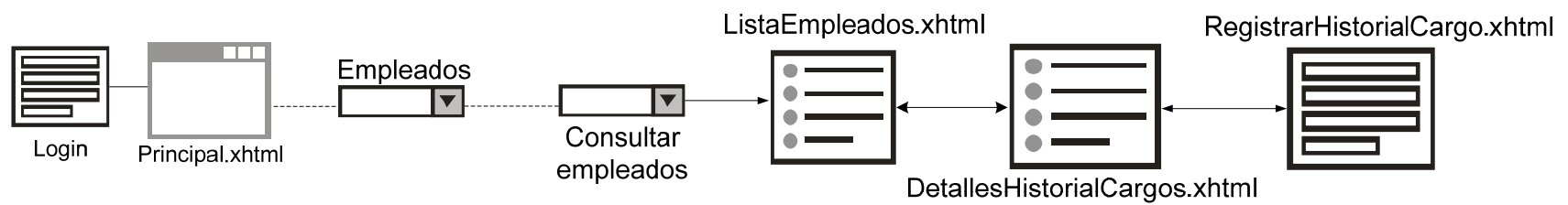


Diagrama Navegación 7: Gestión historial cargos.

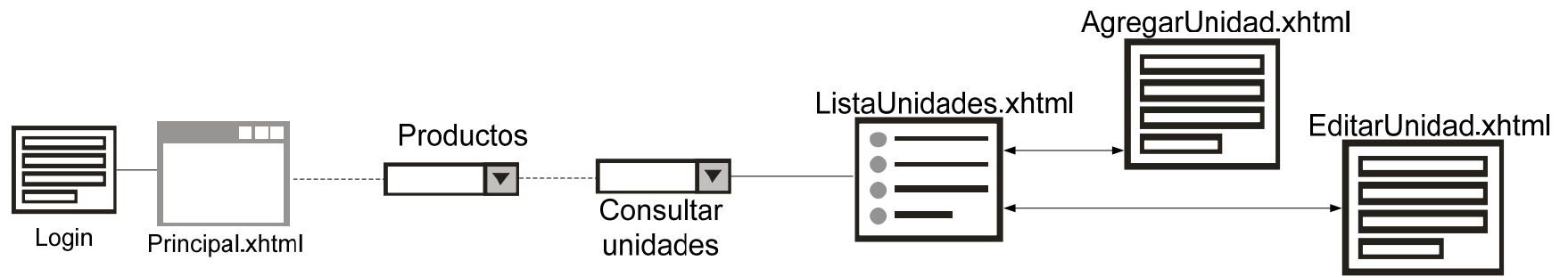


Diagrama Navegación 8: Gestión unidades.

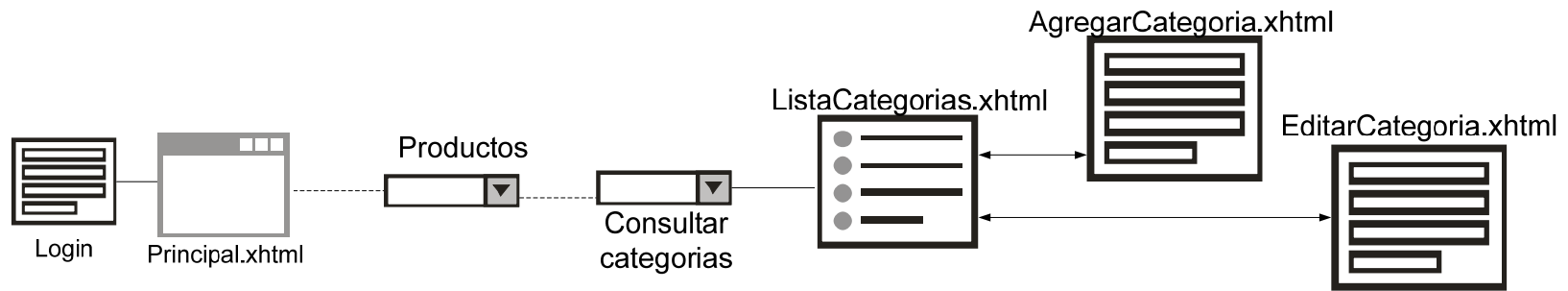


Diagrama Navegación 9: Gestión categorías.

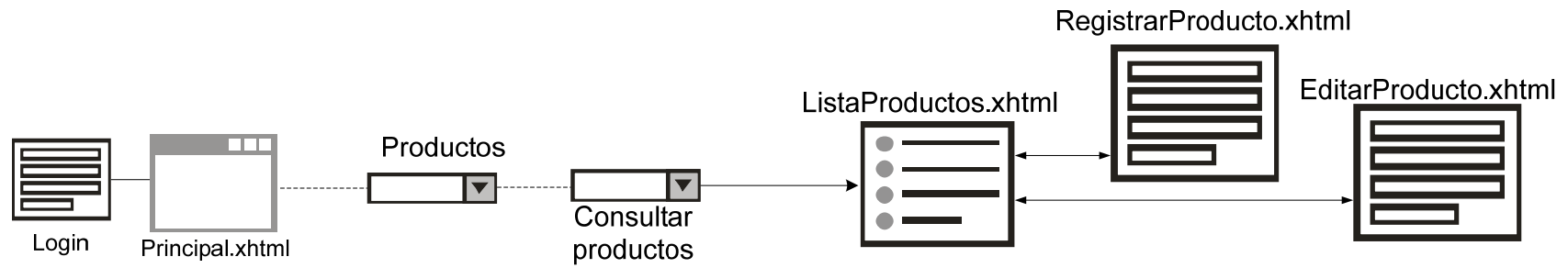


Diagrama Navegación 10: Gestión Productos

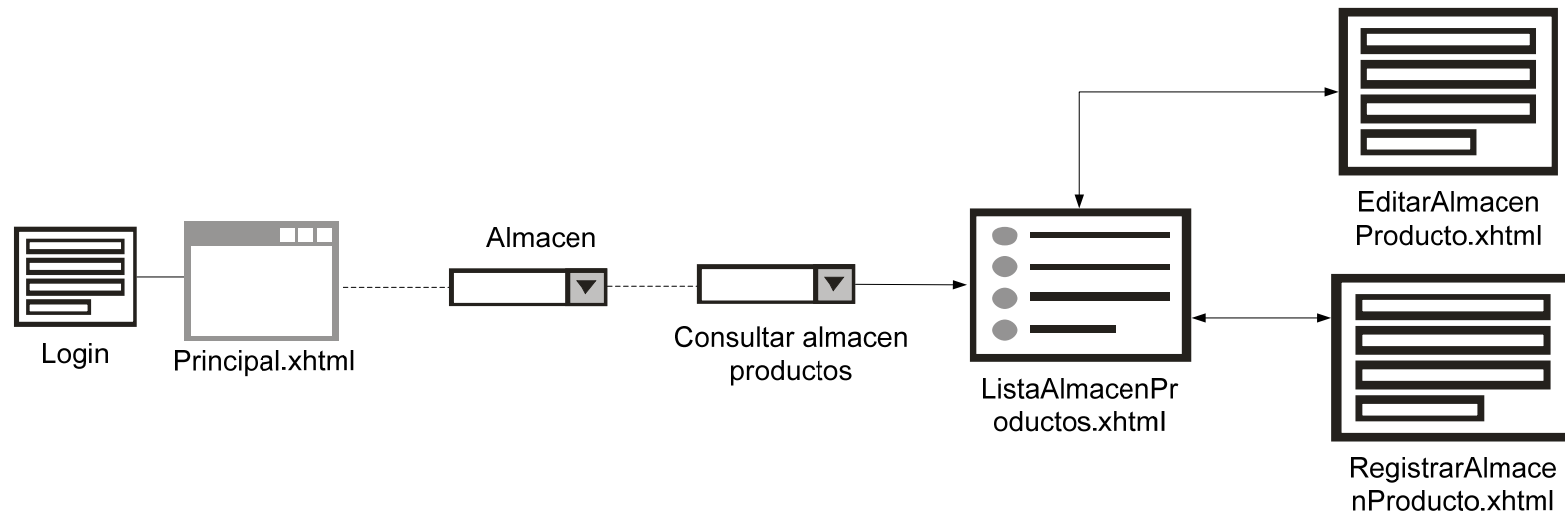


Diagrama Navegación 11: Gestión almacén productos.

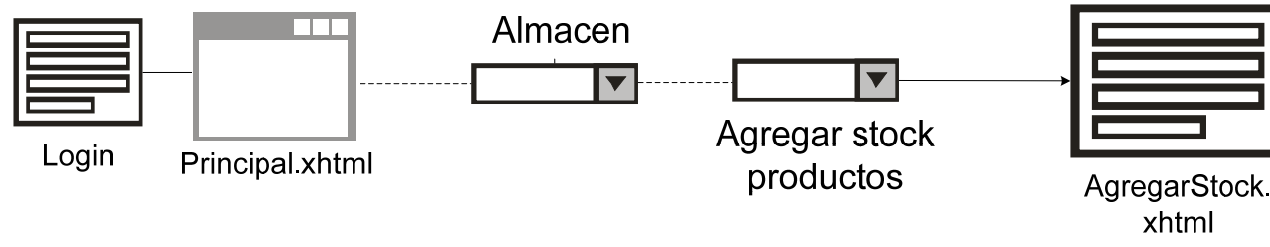


Diagrama Navegación 12: Gestión stock almacén producto.

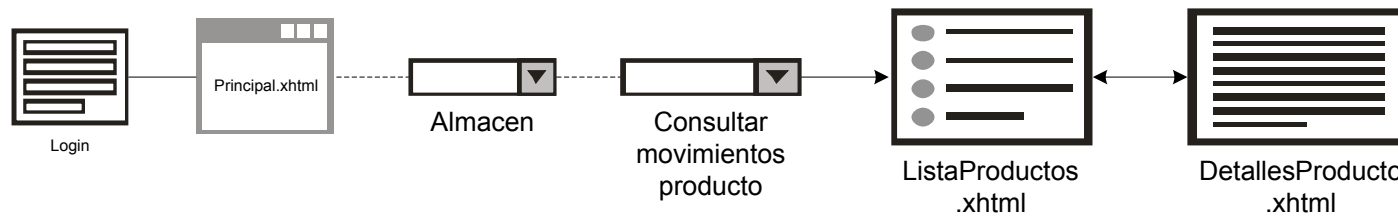


Diagrama Navegación 13: Detalles movimientos producto.

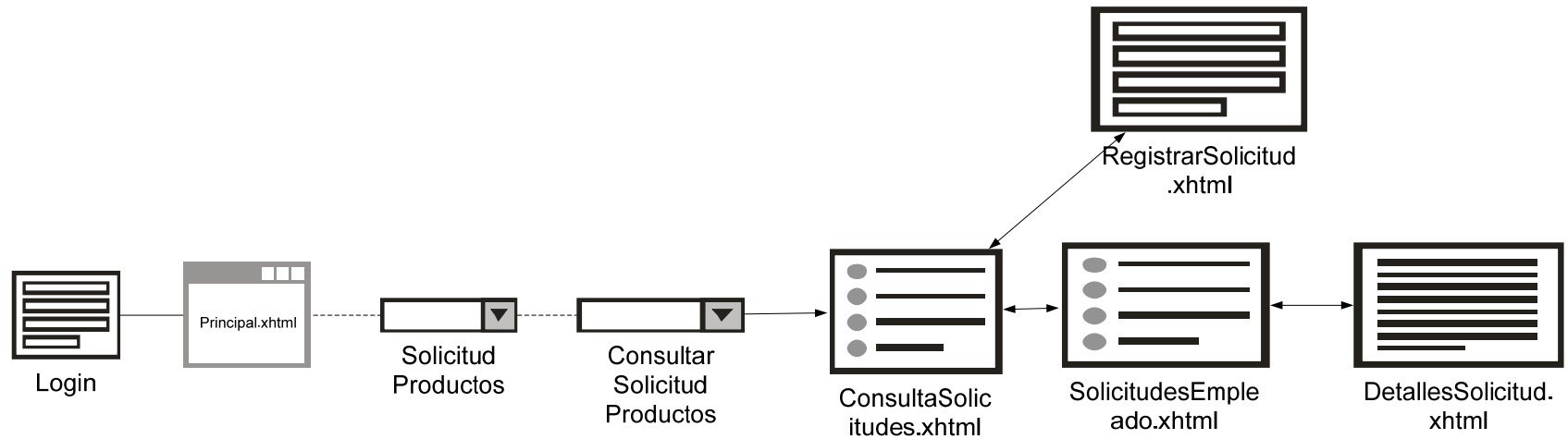


Diagrama Navegación 14: Gestión solicitudes productos.

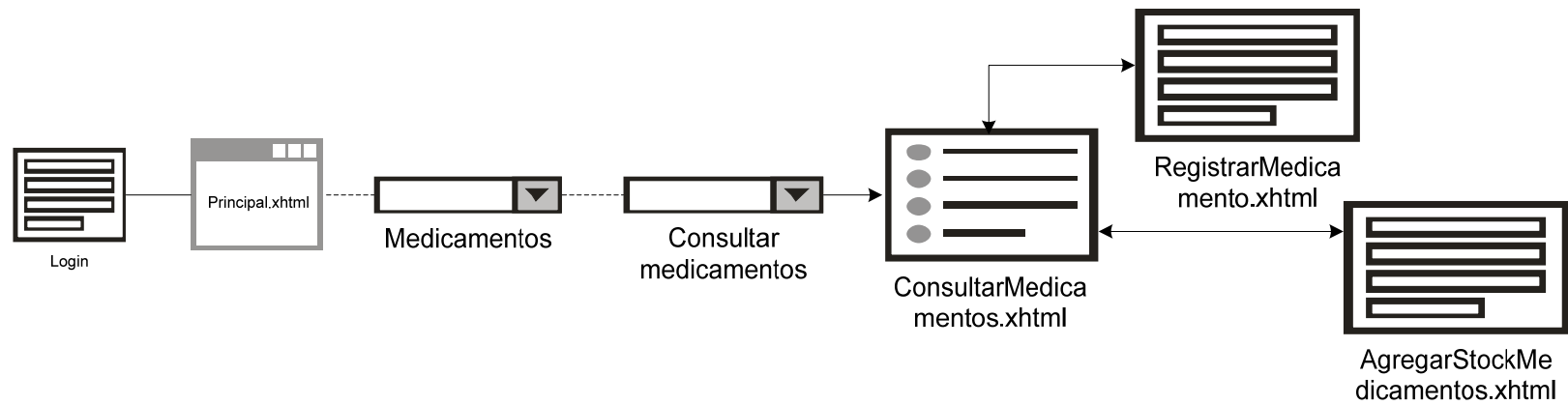


Diagrama Navegación 15: Gestión medicamentos.

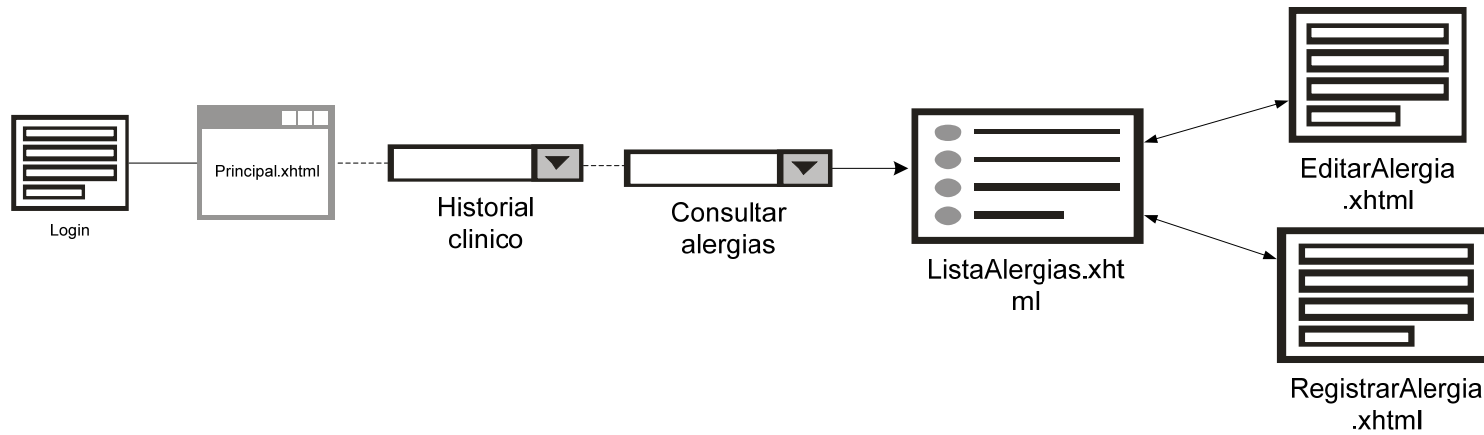


Diagrama Navegación 16: Gestión tipos de alergias.

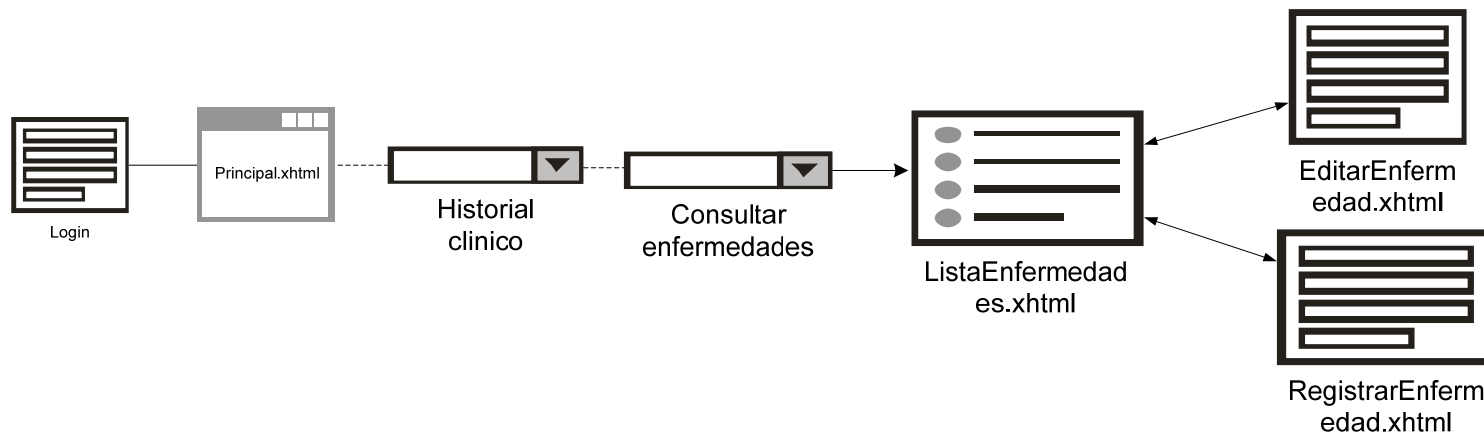


Diagrama Navegación 17: Gestión tipos enfermedades.

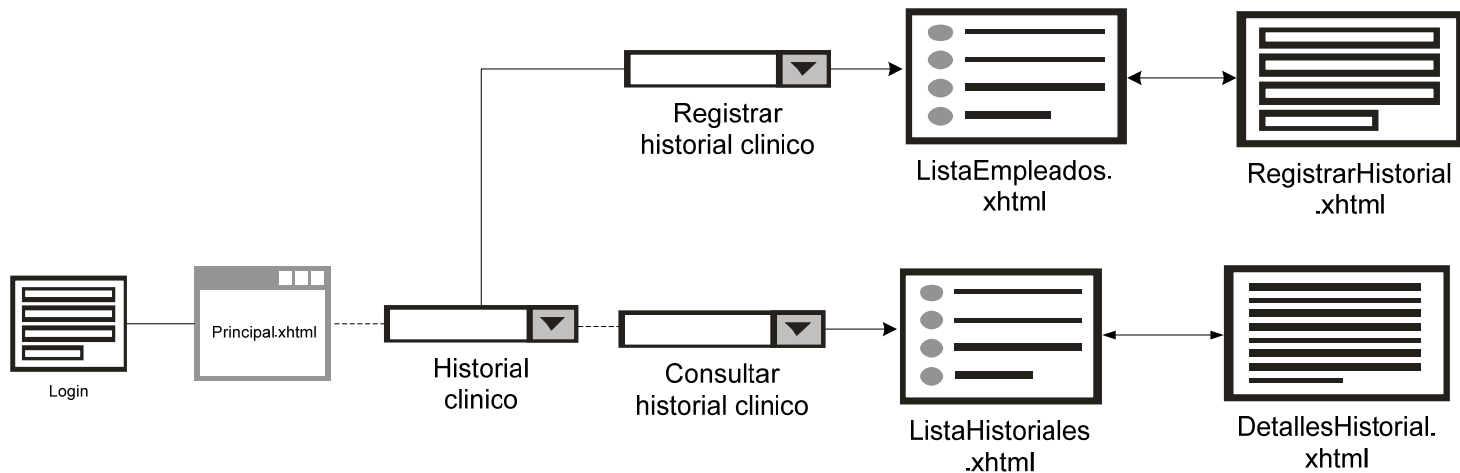


Diagrama Navegación 18: Gestión historiales clínicos.

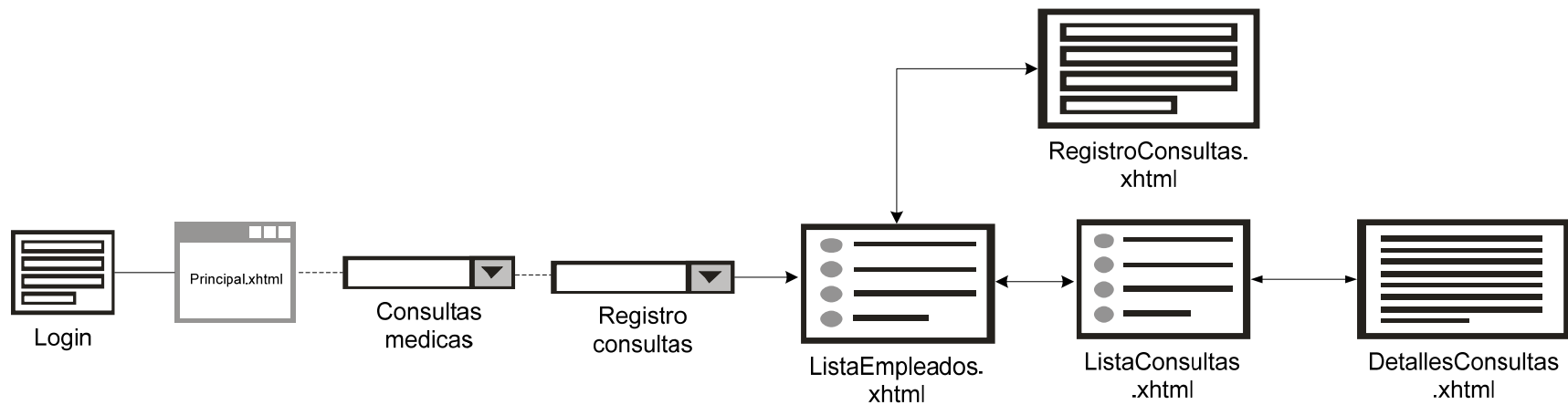


Diagrama Navegación 19: Gestión consultas médicas.

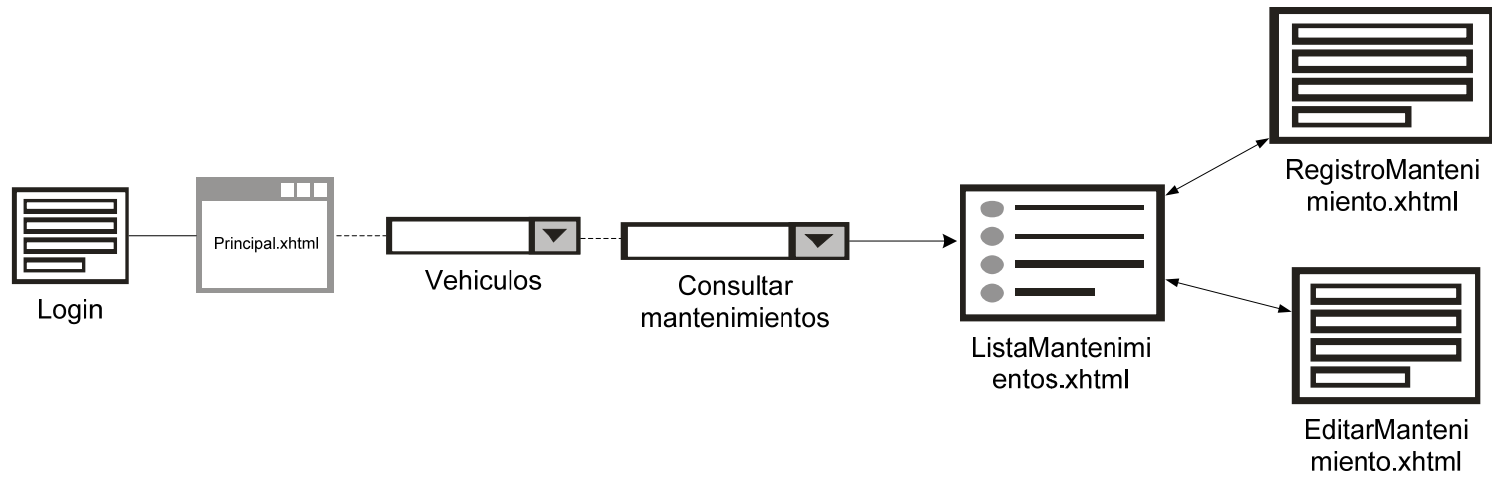


Diagrama Navegación 20: Gestión mantenimientos vehículos.

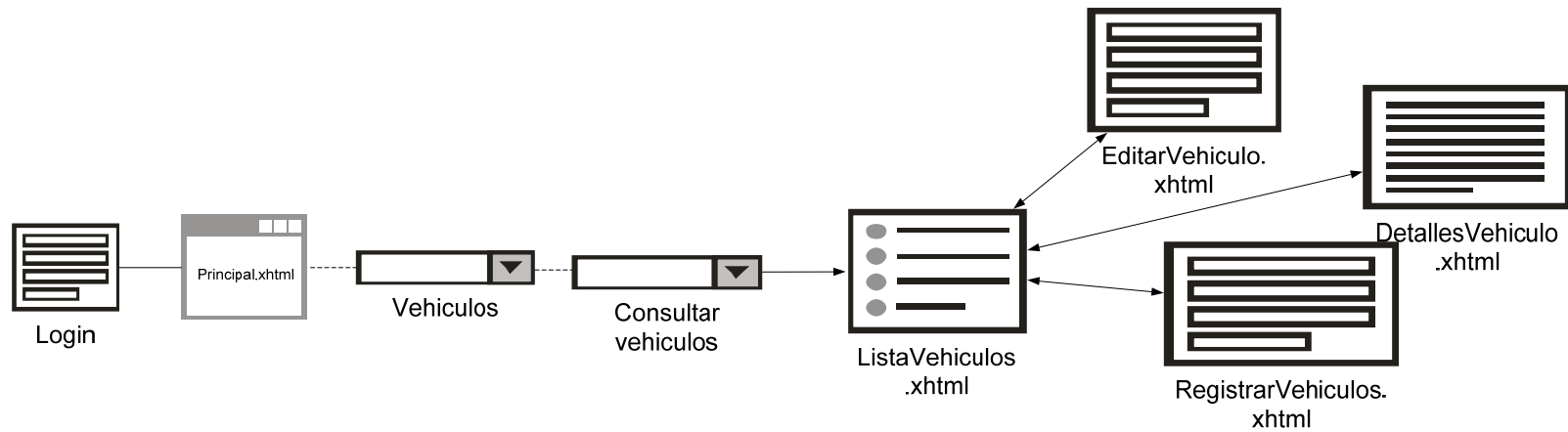


Diagrama Navegación 21: Gestión vehículos.

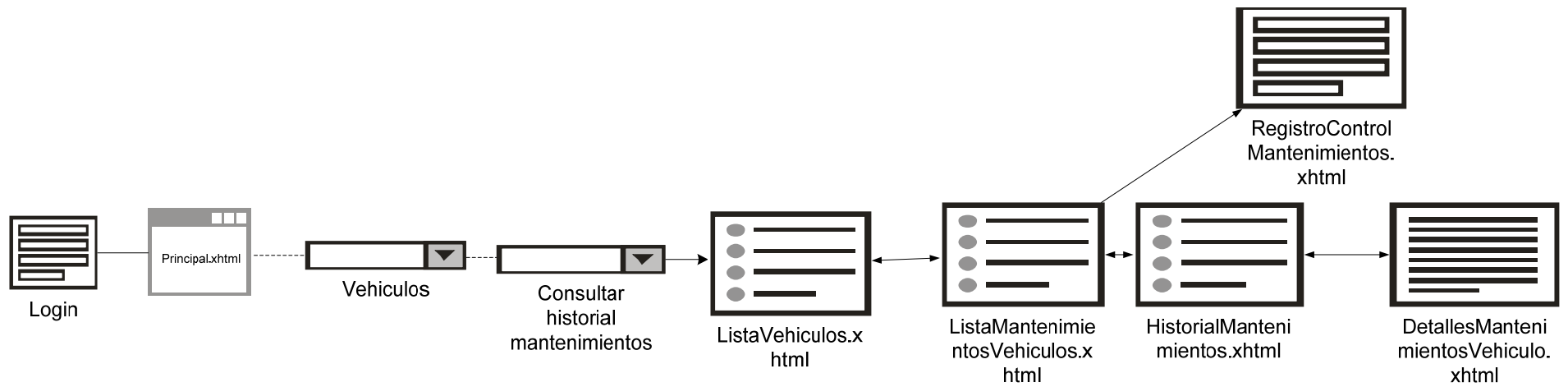


Diagrama Navegación 22: Gestión historial mantenimientos.

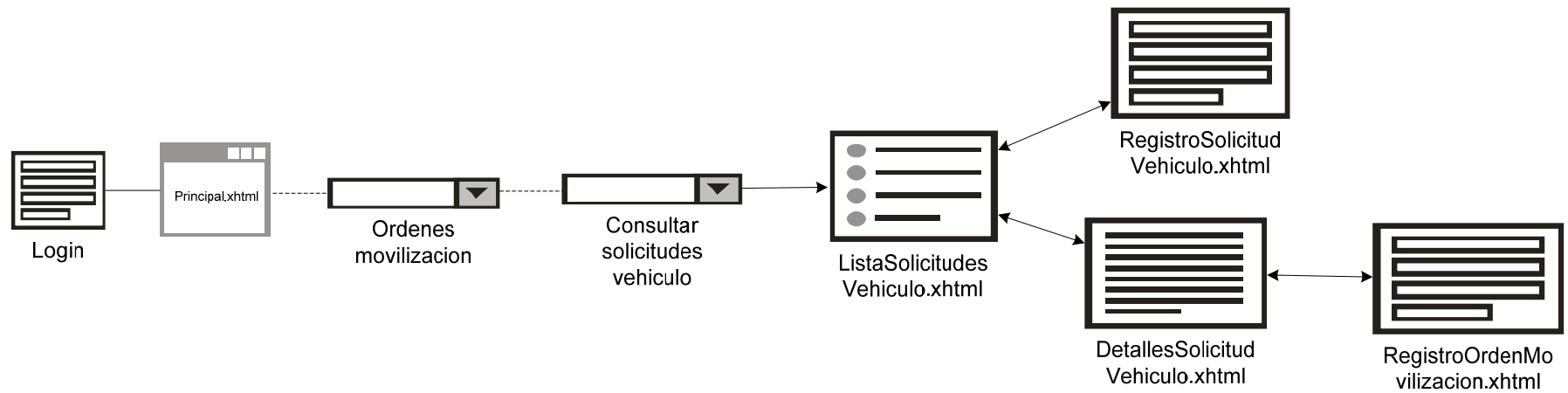


Diagrama Navegación 23: Gestión de solicitudes vehículo y registro de ordenes de movilización.



Diagrama Navegación 24: Gestión ordenes movilización.

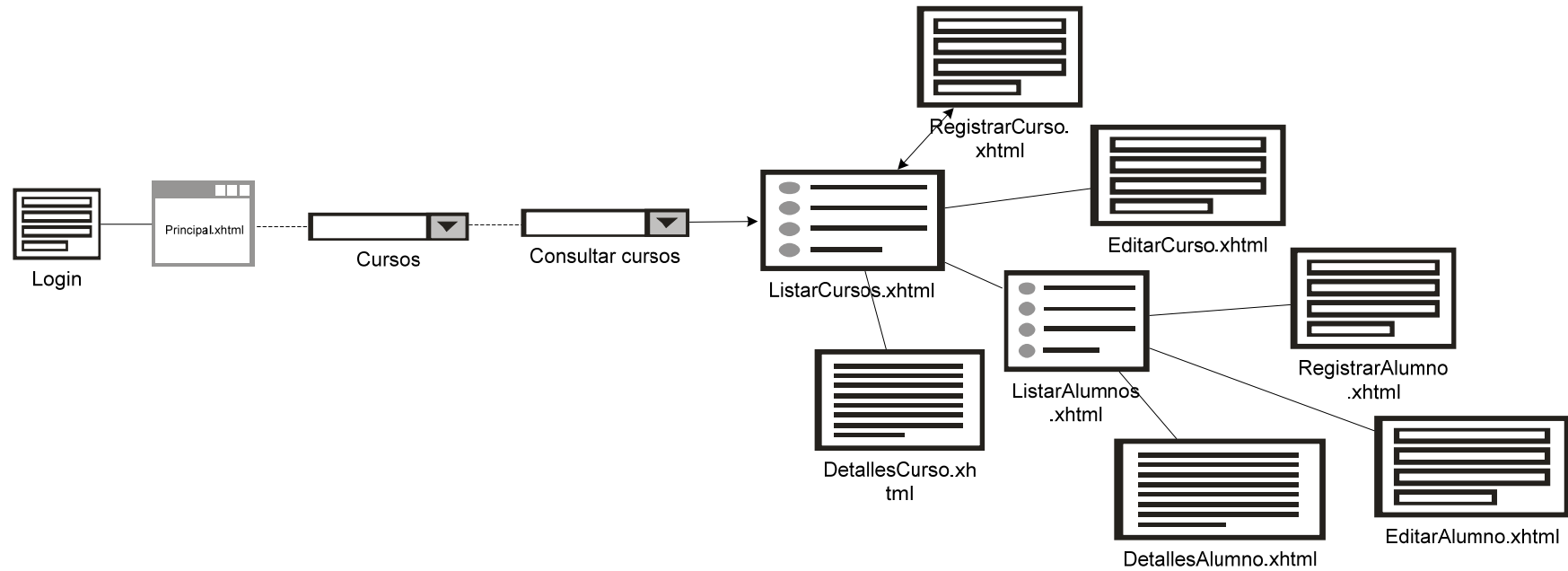


Diagrama Navegación 25: Gestión cursos.

5.9. DICCIONARIO DE DATOS

5.9.1. MÓDULO USUARIOS

NOMBRE DE TABLA:		Roles		
TABLA RELACIONADA:				
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_rol	PK		SERIAL	
rol		200	VARCHAR	
descripcion		500	VARCHAR	

NOMBRE DE TABLA:		Permisos Rol		
TABLA RELACIONADA:		Roles		
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_permiso	PK		SERIAL	
modulo		500	VARCHAR	
cod_modulo			INTEGER	
permiso			INTEGER	
estado			BOOLEAN	
id_Rol	FK		INTEGER	

NOMBRE DE TABLA:		Usuarios		
TABLA RELACIONADA:		Dependencias, Rol		
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_usuario	PK		SERIAL	
nro_cedula		10	VARCHAR	
nombre_usuario		50	VARCHAR	
primer_nombre		50	VARCHAR	
segundo_nombre		50	VARCHAR	
apellido_paterno		50	VARCHAR	
apellido_materno		50	VARCHAR	
fecha_registro			TIMESTAMP	
estado_usuario		1	VARCHAR	
id_rol	FK		INTEGER	
id_dependencia	FK		INTEGER	

NOMBRE DE TABLA:		Contrasenas		
TABLA RELACIONADA:		Usuarios		
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_contrasena	PK		BIGSERIAL	
contrasena		100	VARCHAR	
fecha_registro			TIMESTAMP	
fecha_mantenimiento			TIMESTAMP	
estado_contrasena		1	VARCHAR	
id_usuario	FK		INTEGER	

NOMBRE DE TABLA:		Sesiones		
TABLA RELACIONADA:		Usuarios		
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_sesion	PK		BIGSERIAL	
ip		40	VARCHAR	
fecha_entrada			TIMESTAMP	
hora_ingreso			TIME	
id_usuario	FK		INTEGER	

5.9.2. MÓDULO EMPLEADOS

NOMBRE DE TABLA:		Dependencias		
TABLA RELACIONADA:				
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_dependencia	PK		SERIAL	
dependencia		200	VARCHAR	

NOMBRE DE TABLA:		Cargos		
TABLA RELACIONADA:				
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_cargo	PK		SERIAL	
nombre_cargo		100	VARCHAR	

NOMBRE DE TABLA:	Departamentos			
TABLA RELACIONADA:				
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_departamento	PK		SERIAL	
nombre_departamento		100	VARCHAR	

NOMBRE DE TABLA:	Empleados			
TABLA RELACIONADA:	Sesiones, Dependencias			
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_empleado	PK		BIGSERIAL	
codigo_empleado		100	VARCHAR	
nro_cedula		10	VARCHAR	
primer_nombre		100	VARCHAR	
segundo_nombre		100	VARCHAR	
apellido_paterno		100	VARCHAR	
apellido_materno		100	VARCHAR	
fecha_nacimiento			TIMESTAMP	
estado_civil		10	VARCHAR	
dirección_domicilio		100	VARCHAR	
teléfono_fijo		9	VARCHAR	
teléfono_movil		10	VARCHAR	
email_personal		100	VARCHAR	
email_trabajo		100	VARCHAR	
fecha_ingreso			TIMESTAMP	
estado_empleado		1	VARCHAR	
id_dependencia	FK		INTEGER	
id_sesion	FK		BIGINT	

NOMBRE DE TABLA:	Cargas_Familiares			
TABLA RELACIONADA:	Empleados, Sesiones			
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_carga_familiar	PK		BIGSERIAL	
nombre_carga		200	VARCHAR	
tipo_carga		10	VARCHAR	
id_empleado	FK		BIGINT	
id_sesion	FK		BIGINT	

NOMBRE DE TABLA:	Historial Cargos			
TABLA RELACIONADA:	Cargos, Departamentos, Empleados, Sesiones, Dependencias			
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_historial_cargo	PK		BIGSERIAL	
fecha_asignacion			TIMESTAMP	
estado_historial_cargo		1	VARCHAR	
id_cargo	FK		INTEGER	
id_departamento	FK		INTEGER	
id_empleado	FK		BIGINT	
id_dependencia	FK		INTEGER	
id_sesion	FK		BIGINT	

5.9.3. MÓDULO PRODUCTOS

NOMBRE DE TABLA:	Categoria Productos			
TABLA RELACIONADA:				
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_categoria_producto	PK		SERIAL	
codigo_cat_prod		10	VARCHAR	
categoria_producto		200	VARCHAR	
descripcion		500	VARCHAR	

NOMBRE DE TABLA:	Unidades			
TABLA RELACIONADA:				
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_unidad	PK		SERIAL	
unidad		100	VARCHAR	

NOMBRE DE TABLA:	Productos			
TABLA RELACIONADA:	Unidades, Sesiones, Categorías Productos			
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_producto	PK		SERIAL	
producto		200	VARCHAR	
descripcion		500	VARCHAR	
id_categoria_producto	FK		INTEGER	
id_unidad	FK		INTEGER	

id_sesion	FK		BIGINT	
NOMBRE DE TABLA:	Almacen Productos			
TABLA RELACIONADA:	Productos, Dependencias, Sesiones			
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_almacen_producto	PK		BIGSERIAL	
cantidad_max		(10,2)	NUMERIC	
cantidad_min		(10,2)	NUMERIC	
total_producto		(10,2)	NUMERIC	
estado_producto		1	VARCHAR	
id_sesion	FK		BIGINT	
id_producto	FK		INTEGER	
id_dependencia	FK		INTEGER	

NOMBRE DE TABLA:	Kardex Producto			
TABLA RELACIONADA:	Almacenes Productos, Sesiones			
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_kardex	PK		BIGSERIAL	
movimiento		1	VARCHAR	
fecha			TIMESTAMP	
concepto		300	VARCHAR	
referencia		300	VARCHAR	
cantidad_ingreso		(10,2)	NUMERIC	
costo_unit_ingreso		(10,2)	NUMERIC	
valor_total_ingreso		(10,2)	NUMERIC	
cantidad_salida		(10,2)	NUMERIC	
costo_unit_salida		(10,2)	NUMERIC	
valor_total_salida		(10,2)	NUMERIC	
saldo		(10,2)	NUMERIC	
costo_unit_saldo		(10,2)	NUMERIC	
valor_total_saldo		(10,2)	NUMERIC	
id_almacen_producto	FK		BIGINT	
id_sesion	FK		BIGINT	

NOMBRE DE TABLA:	Solicitudes Productos			
TABLA RELACIONADA:	Dependencias, Sesiones, Empleados			
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_solicitud	PK		BIGSERIAL	
fecha_solicitud			TIMESTAMP	
hora_solicitud			TIME	
total		10,2	NUMERIC	
estado_solicitud		1	VARCHAR	

id_dependencia	FK		INTEGER	
id_sesion	FK		BIGINT	
id_empleado	FK		BIGINT	

NOMBRE DE TABLA:		Detalles Solicitud Producto		
TABLA RELACIONADA:				
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_detalle_solicitud	PK		BIGSERIAL	
cantidad		10,2	NUMERIC	
id_solicitud	FK		BIGINT	
id_almacen_producto	FK		BIGINT	
id_sesion	FK		BIGINT	

5.9.4. MÓDULO CURSOS

NOMBRE DE TABLA:		Facilitadores		
TABLA RELACIONADA:				
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_facilitador	PK		SERIAL	
nombre		200	VARCHAR	
especialidad		100	VARCHAR	
calificacion		(3,2)	NUMERIC	

NOMBRE DE TABLA:		Temas Curso Facilitador		
TABLA RELACIONADA:		Cursos, Facilitadores		
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_tema_curso_facilitador	PK		SERIAL	
nombre_tema		200	VARCHAR	
id_curso	FK		INTEGER	
id_facilitador	FK		INTEGER	

NOMBRE DE TABLA:		CURSOS		
TABLA RELACIONADA:		Dependencias, Sesiones		
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_curso	PK		SERIAL	
nombre_curso		500	VARCHAR	
fecha_inicio			TIMESTAMP	

fecha_fin			TIMESTAMP	
nro_horas			INTEGER	
lugar_curso		500	VARCHAR	
nombre_coordinador		200	VARCHAR	
calificacion_curso		2,2	NUMERIC	
modalidad_curso		1	VARCHAR	
costo_curso		4,2	NUMERIC	
nro_alumnos_aprobados			INTEGER	
nro_alumnos_reprobados			INTEGER	
nro_inscritos			INTEGER	
nro_participantes			INTEGER	
tipo_aprobacion		1	VARCHAR	
calificación_curso		(3,2)	NUMBER	
estado_curso		1	VARCHAR	
id_sesion	FK		BIGINT	
id_dependencia	FK		INTEGER	

NOMBRE DE TABLA:	Alumnos			
TABLA RELACIONADA:	Cursos, Sesiones			
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_alumno	PK		SERIAL	
nro_cedula		10	VARCHAR	
nombre_alumno		100	VARCHAR	
lugar_residencia		200	VARCHAR	
direccion		500	VARCHAR	
telefono_movil		10	VARCHAR	
telefono_fijo		9	VARCHAR	
email		100	VARCHAR	
nombre_institucion		300	VARCHAR	
cargo		100	VARCHAR	
calificación_alumno		2,2	NUMERIC	
tipo_alumno		1	VARCHAR	
estado_alumno		1	VARCHAR	
id_curso	FK		INTEGER	
id_sesion	FK		BIGINT	

NOMBRE DE TABLA:	Alumnos_Curso			
TABLA RELACIONADA:	Cursos, Alumnos, Sesiones			
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_alumno_curso	PK		SERIAL	
tipo_alumno_curso			INTEGER	
calificacion_alumno	FK		INTERGER	
estado_alumno_curso	FK		INTEGER	

id_sesion	FK		BIGINTEGER	
id_curso	FK		INTEGER	
id_alumno	FK		INTEGER	

5.9.5. MÓDULO VEHICULOS

NOMBRE DE TABLA:		Vehiculos		
TABLA RELACIONADA:		Dependencias, Sesiones		
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_vehiculo	PK		SERIAL	
marca		200	VARCHAR	
modelo		200	VARCHAR	
anio_vehiculo			INTEGER	
nro_chasis		100	VARCHAR	
placa_vehiculo		7	VARCHAR	
color		100	VARCHAR	
km_actual			INTEGER	
estado_vehiculo		1	VARCHAR	
id_dependencia	FK		INTEGER	
id_sesion	FK		BIGINT	

NOMBRE DE TABLA:		Solicitudes vehiculos		
TABLA RELACIONADA:		Empleados, Dependencias, Sesiones		
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_solicitud	PK		SERIAL	
nro_solicitud			INTEGER	
fecha_solicitud			TIMESTAMP	
hora_solicitud			TIME	
motivo		500	VARCHAR	
destino		100	VARCHAR	
fecha_movilizacion			TIMESTAMP	
hora_movilizacion			TIME	
estado_solicitud		1	VARCHAR	
id_dependencia	FK		INTEGER	
id_empleado	FK		BIGINT	
id_sesion	FK		BIGINT	

NOMBRE DE TABLA:	Equipos Trabajo			
TABLA RELACIONADA:	Solicitudes_vehiculos			
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_equipo_trabajo	PK		SERIAL	
equipo_trabajo_nombre		200	VARCHAR	
id_solicitud	FK		INTEGER	

NOMBRE DE TABLA:	Ordenes Movilización			
TABLA RELACIONADA:	Solicitudes_Vehiculos, Empleados, Sesiones, Dependencias			
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_orden_movilizacion	PK		SERIAL	
nro_orden			INTEGER	
fecha_salida			TIMESTAMP	
hora_salida			TIME	
lugar_salida		100	VARCHAR	
km_salida			INTEGER	
fecha_llegada			TIMESTAMP	
hora_llegada			TIME	
km_llegada			INTEGER	
estado_orden		1	VARCHAR	
chofer_asignado	FK		BIGINT	Clave foránea de la tabla Empleados
id_solicitud	FK		INTEGER	
id_vehiculo	FK		INTEGER	
id_dependencia	FK		INTEGER	
id_sesion	FK		BIGINT	

NOMBRE DE TABLA:	Mantenimientos			
TABLA RELACIONADA:				
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_mantenimiento	PK		SERIAL	
mantenimiento		200	VARCHAR	
km_mantenimiento			INTEGER	
codigo_tipo_mantenimiento			INTEGER	
tipo_mantenimiento		100	VARCHAR	

NOMBRE DE TABLA:	Mantenimiento_Vehiculos			
TABLA RELACIONADA:	Vehículos, Mantenimiento			
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_mantenimiento_vehiculo	PK		SERIAL	
km_mantenimiento			INTEGER	
fecha_ultima_revision			TIMESTAMP	
estado_mantenimiento		1	VARCHAR	
id_vehiculo	FK		INTEGER	
id_mantenimiento	FK		INTEGER	

NOMBRE DE TABLA:	Control_Mantenimientos_Vehiculos			
TABLA RELACIONADA:	Mantenimientos_Vehiculos, Sesiones			
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_cmv	PK		SERIAL	
fecha_control			TIMESTAMP	
km_control			INTEGER	
marca_producto		200	VARCHAR	
codigo_producto		500	VARCHAR	
referencia		100	VARCHAR	
costo_mantenimiento		6,2	NUMERIC	
observacion		500	VARCHAR	
id_mantenimiento_vehiculo	FK		INTEGER	
id_sesion	FK		BIGINT	

5.9.6. MÓDULO CONTROL MÉDICO

NOMBRE DE TABLA:	Historial_Clinico			
TABLA RELACIONADA:	Empleados. Sesiones			
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_historial_clinico	PK		SERIAL	
tipo_sangre		5	VARCHAR	
sexo		10	VACHAR	
id_empleado	FK		BIGINT	
id_sesion	FK		BIGINT	

NOMBRE DE TABLA:	Tipos_enfermedad			
TABLA RELACIONADA:				
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_tipos_enfermedad	PK		SERIAL	
tipo_enfermedad		20	VARCHAR	

NOMBRE DE TABLA:	Detalles_enfermedades			
TABLA RELACIONADA:	Tipos_Enfermedad, Historial_Clinico, Sesiones			
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_detalle_enfermedad	PK		SERIAL	
descripcion		200	VARCHAR	
id_tipos_enfermedad	FK		INTEGER	
id_historial_clinico	FK		INTEGER	
id_sesion	FK		BIGINT	

NOMBRE DE TABLA:	Tipos_alergia			
TABLA RELACIONADA:				
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_tipo_alergia	PK		SERIAL	
tipo_alergia		50	VARCHAR	

NOMBRE DE TABLA:	Detalles_alergias			
TABLA RELACIONADA:	Tipos_Alergias, Historial_Clinicos, Tipos_alergia			
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_detalle_alergias	PK		SERIAL	
descripcion		200	VARCHAR	
id_tipo_alergia	FK		INTEGER	
id_historial_clinico	FK		INTEGER	
id_sesion	FK		BIGINT	

NOMBRE DE TABLA:	Consultas			
TABLA RELACIONADA:	Empleados, Dependencias, Sesiones			
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_consulta	PK		SERIAL	
hora_consulta			TIME	

fecha_consulta			TIMESTAMP	
motivo_consulta		500	VARCHAR	
sintomas		500	VARCHAR	
diagnostico		500	VARCHAR	
id_dependencia	FK		INTEGER	
id_empleado	FK		BIGINT	
id_sesion	FK		BIGINT	

NOMBRE DE TABLA:		Prescripciones		
TABLA RELACIONADA:		Consultas		
DESCRIPCIÓN:				
CAMPO	CAMPOS CLAVE	TAMAÑO	TIPO DATO	DESCRIPCIÓN
id_prescripcion	PK		SERIAL	
prescripción		500	VACHAR	
cantidad			INTEGER	
dosis		300	VARCHAR	
id_consulta	FK		INTEGER	

5.10. IDENTIFICACIÓN DE ROLES Y PRIVILEGIOS

MÓDULOS	USUARIO										EMPLEADOS														
SUBMÓDULOS	ROLES					USUARIO					EMPLEADOS					CARGAS FAMILIARES					CARGOS				
ACCIONES	C	M	E	I	L	C	M	E	I	L	C	M	E	I	L	C	M	E	I	L	C	M	E	I	L
ROLES																									
Administrador	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
RRHH											X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Medico															X										
Vehículo															X										
Bodega															X										

MÓDULO	PRODUCTOS																			
SUBMÓDULOS	PRODUCTOS					PRODUCTOS BODEGA					MOVIMIENTOS PRODUCTOS BODEGA					SOLICITUD PRODUCTOS				
ACCIONES	C	M	E	I	L	C	M	E	I	L	C	M	E	I	L	C	M	E	I	L
ROLES																				
Administrador	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
RRHH																				
Medico																				
Mantenimiento																				
Bodega	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

MÓDULO	VEHÍCULOS																			
SUBMÓDULOS	VEHÍCULOS					SOLICITUD VEHÍCULOS					ORDENES MOVILIZACIÓN					MANTENIMIENTO VEHÍCULOS				
ACCIONES	C	M	E	I	L	C	M	E	I	L	C	M	E	I	L	C	M	E	I	L
ROLES																				
Administrador	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
RRHH																				
Medico																				
Mantenimiento																				
Bodega	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

MÓDULO	CONTROL MÉDICO										CURSOS														
SUBMÓDULOS	HISTORIAL MÉDICO					CONSULTAS MÉDICAS					CURSOS					ALUMNOS					CALIFICACION CURSO				
ACCIONES	C	M	E	I	L	C	M	E	I	L	C	M	E	I	L	C	M	E	I	L	C	M	E	I	L
ROLES																									
Administrador	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
RRHH											X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Medico	X	X	X	X	X	X	X	X	X	X															
Mantenimiento																									
Bodega																									

C = Crear, M = Modificar, E = Eliminar, I = Imprimir reporte, L = Listar

5.11. INTERFACES DE USUARIO

5.11.1. AUTENTICACIÓN DEL SISTEMA



INICIAR SESION

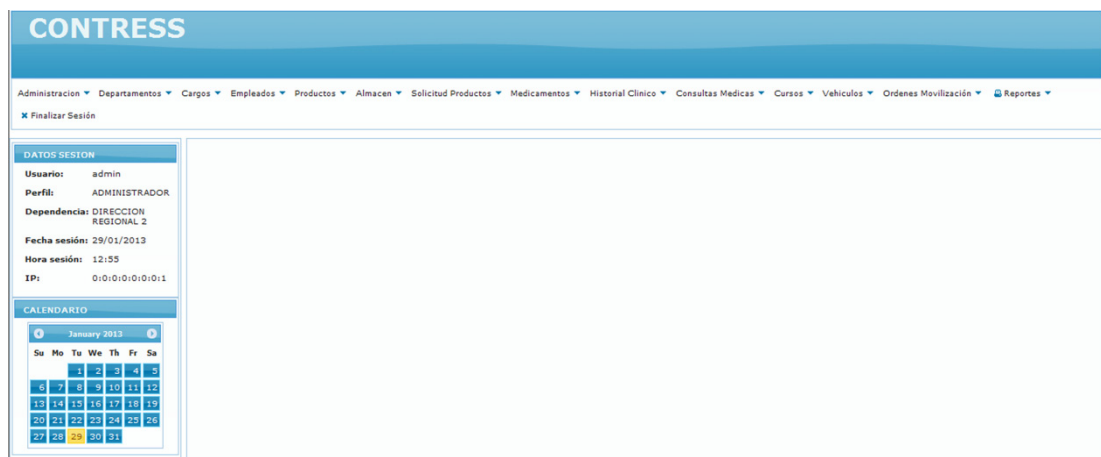
Nombre de usuario:

Contraseña:

Ingresar Cancelar

Imagen 15: Interfaz de autenticación

5.11.2. PANTALLA PRINCIPAL



CONTRESS

Administración ▾ Departamentos ▾ Cargos ▾ Empleados ▾ Productos ▾ Almacen ▾ Solicitud Productos ▾ Medicamentos ▾ Historial Clinico ▾ Consultas Medicas ▾ Cursos ▾ Vehiculos ▾ Ordenes Movilización ▾ Reportes ▾

✕ Finalizar Sesión

DATOS SESION

Usuario: admin
Perfil: ADMINISTRADOR
Dependencia: DIRECCION REGIONAL 2
Fecha sesión: 29/01/2013
Hora sesión: 12:55
IP: 0101010101011

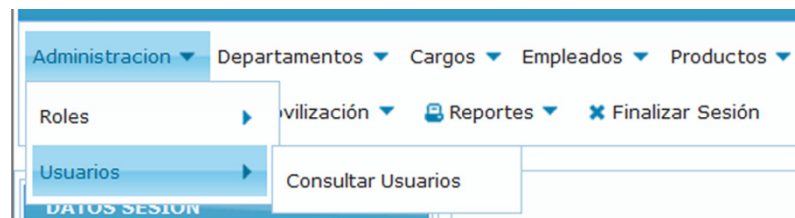
CALENDARIO

January 2013

Su	Mo	Tu	We	Th	Fr	Sa
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Imagen 16: Pantalla principal del sistema.

5.11.3. MENUS DEL SISTEMA



Administración ▾ Departamentos ▾ Cargos ▾ Empleados ▾ Productos ▾

Roles ▸ Movilización ▾ Reportes ▾ Finalizar Sesión

Usuarios ▸ Consultar Usuarios

DATOS SESION



Imagen 17: Menú del sistema.

5.11.4. GESTIÓN DE ROLES



Imagen 18: Interfaz para la gestión de roles.

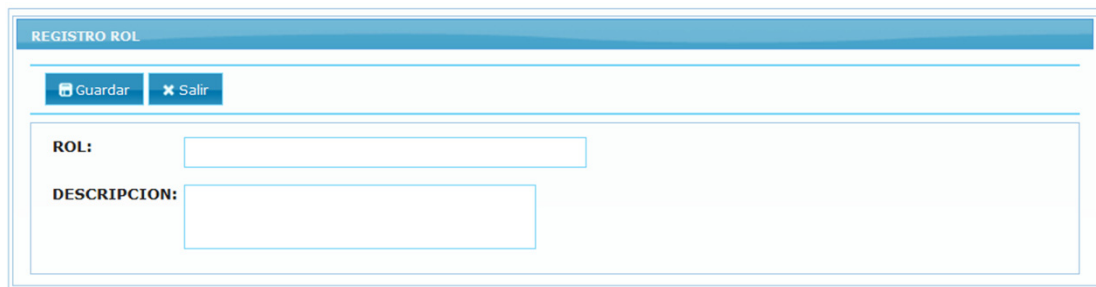


Imagen 19: Formulario para el registro de roles.

PERMISOS ROLES

ASIGNACION PERMISOS - ROL COMPRAS

Guardar Salir

MODULO	ACCESO MODULO	PERMISOS	
MODULO CURSOS	DENEGADO	SIN PERMISO	
MODULO CONTROL MEDICO	<input type="checkbox"/>	SIN PERMISO SIN PERMISO LECTURA, ESCRITURA LECTURA	
MODULO PRODUCTOS	DENEGADO		
MODULO EMPLEADOS	DENEGADO	SIN PERMISO	
MODULO VEHICULOS	DENEGADO	SIN PERMISO	
MODULO ADMINISTRACION	DENEGADO	SIN PERMISO	

Imagen 20: Formulario para asignar acceso y permisos a módulos.

5.11.5. GESTIÓN DE USUARIOS

LISTA USUARIOS

Registrar Editar Bloquear Desbloquear Restablecer contraseña Detalles Restaurar Eliminar Salir

Buscar:

(1 of 1) 1 10

NOMBRE USUARIO	ROL USUARIO	Dependencia	ESTADO
admin	ADMINISTRADOR	DIRECCION REGIONAL 2	ACTIVO
ocastro	COMPRAS	DELEGACION PROV. CANAR	ACTIVO
zpeña	LOGISTICA	DELEGACION PROV. MORONA SANTIAGO	ACTIVO

Total de registros 3

Imagen 21: Interfaz para la gestión de usuarios.

REGISTRO USUARIO

Guardar Salir

DATOS USUARIO

NRO. CEDULA:

NOMBRE COMPLETO:

DEPENDENCIA:

NOMBRE USUARIO:

CONTRASEÑA TEMPORAL:

ROL USUARIO:

- ADMINISTRADOR
- COMPRAS
- LOGISTICA
- CONTABILIDAD

Imagen 22: Formulario para el registro de usuarios.

5.11.6. GESTIÓN DE EMPLEADOS

LISTA EMPLEADOS

Registrar empleado Detalles empleado Editar empleado Restaurar empleado Eliminar empleado Historial Cargos Cargas Familiares Salir

Buscar:

(1 of 1) 1

CODIGO	NRO. CEDULA	APELLIDO	NOMBRE	DEPENDENCIA	ESTADO
96	1310019144	Alvarado Jimenes	Diego Oswaldo	DIRECCION REGIONAL 2	ACTIVO
AUD15	0104809504	Castro Arteaga	Oscar Steve	DELEGACION PROV. CANAR	ACTIVO
123	1717997355	Jimenes Alvarado	Omar Pepe	DELEGACION PROV. CANAR	ACTIVO
451	1709221293	Maldonado Cardenas	Anibal Ernesto	DIRECCION REGIONAL 2	ACTIVO
ADM03	0101874675	Peña Cordero	Zoila Justina	DELEGACION PROV. MORONA SANTIAGO	ACTIVO
ADM01	0103032405	Ulloa Peña	Maria Angeles	DIRECCION REGIONAL 2	ELIMINADO

Total de registros 6

Imagen 23: Interfaz para la gestión de empleados.

REGISTRO EMPLEADOS

Guardar Salir

DATOS PERSONALES

Código Empleado:

Nro Cédula:

Primer Nombre:

Segundo Nombre:

Apellido Paterno:

Apellido Materno:

Fecha Nacimiento:

Estado Civil:

Dirección Domicilio:

Teléfono Fijo:

Teléfono Móvil:

Email Personal:

Email Trabajo:

Dependencia perteneciente:

Fecha ingreso institución:

DATOS CARGO ACTUAL

Departamento actual:

Cargo actual:

Fecha asignación cargo actual:

Imagen 24: Formulario para el registro de empleados.

5.11.7. GESTIÓN DE PRODUCTOS

LISTA PRODUCTOS

Registrar producto Editar producto Restaurar producto Eliminar producto Salir

Buscar:

(1 of 1) 1

PRODUCTO	DESCRIPCION	UNIDAD	CATEGORIA	ESTADO
ACEITE	ACEITE DE MOTOR	LITROS	MATERIALES DE MANTENIMIENTO	DISPONIBLE
IBUPROFENO	IBUPROFENO 600GR	UNIDAD	MEDICAMENTOS GENERICOS	DISPONIBLE
LUBRICANTES	LIQUIDO DE FREÑOS	LITROS	MATERIALES DE MANTENIMIENTO	DISPONIBLE
PAPEL	PAPEL DE 80GR	RESMA DE 500 HOJAS	SUMINISTROS OFICINA	DISPONIBLE
TONER	TONER PARA HP	UNIDAD	SUMINISTROS OFICINA	DISPONIBLE

Total de registros 7

Imagen 25: Interfaz para la gestión de productos.

REGISTRO PRODUCTO

Guardar Salir

DATOS PRODUCTO

Nombre producto:

Descripción:

Unidad producto: UNIDAD

Categoría producto: SUMINISTROS OFICINA

-
- SUMINISTROS OFICINA
- MATERIALES DE MANTENIMIENTO
- MEDICAMENTOS GENERICOS

Imagen 26: Formulario para el registro de productos.

5.11.8. GESTIÓN DE ALMACÉN PRODUCTOS

PRODUCTOS ALMACEN

Registrar producto almacen Editar producto almacen Salir

INFORMACION PRODUCTOS

✔ Stock disponible(4 productos)
 ! Stock bajo(0 productos)
 ✘ Stock agotado(2 productos)

LISTA PRODUCTOS ALMACEN

Buscar:

(1 of 1)

Producto	Categoría	Cant. Mínima	Cant. Máxima	Total disponible	Almacen
ACEITE	MATERIALES DE MANTENIMIENTO	100.00	9999.00	220.00	DELEGACION PROV. CANAR
ACEITE	MATERIALES DE MANTENIMIENTO	0.00	9999.00	0.00	DIRECCION REGIONAL 2
ACETAMINOFEN	MEDICAMENTOS GENERICOS	10.00	9999.00	45.00	DIRECCION REGIONAL 2
IBUPROFENO	MEDICAMENTOS GENERICOS	0.00	9999.00	0.00	DIRECCION REGIONAL 2
PAPEL	SUMINISTROS OFICINA	100.00	500.00	585.00	DELEGACION PROV. MORONA SANTIAGO

Total registros 6

Imagen 27: Interfaz para la gestión almacén productos.

REGISTRO PRODUCTOS ALMACEN

Guardar Salir

PRODUCTOS DISPONIBLES

Buscar:

(1 of 1)

PRODUCTO	UNIDAD	CATEGORIA	ESTADO
LUBRICANTES	LITROS	MATERIALES DE MANTENIMIENTO	DISPONIBLE
PAPEL	RESMA DE 500 HOJAS	SUMINISTROS OFICINA	DISPONIBLE
TONER	UNIDAD	SUMINISTROS OFICINA	DISPONIBLE

Características del producto

Cantidad mínima:

Cantidad máxima:

Localización producto: DELEGACION PROV. CANAR

-
- DIRECCION REGIONAL 2
- DELEGACION PROV. CANAR
- DELEGACION PROV. MORONA SANTIAGO

Imagen 28: Formulario para el registro de un producto en un almacén específico.

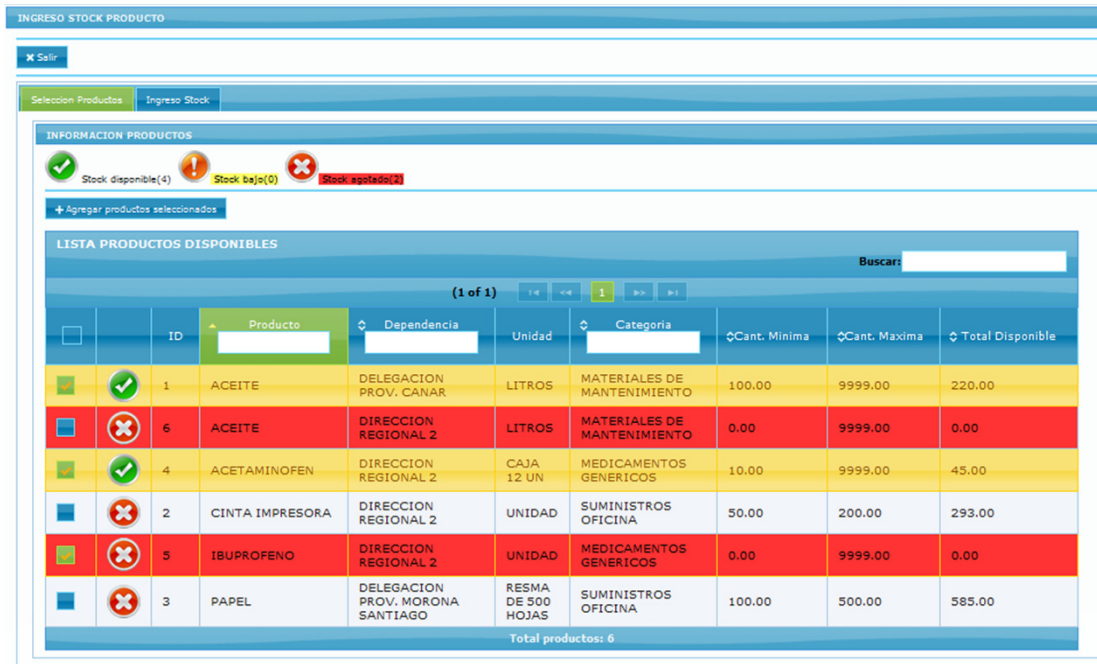


Imagen 29: Interfaz para el ingreso de stock de productos.



Imagen 30: Formulario para el registro de stock de productos.

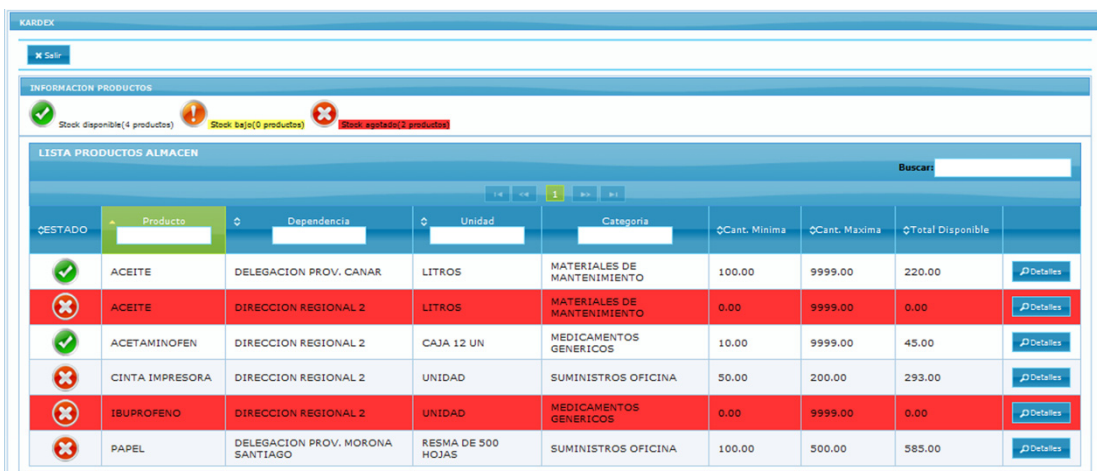


Imagen 31: Interfaz para la consulta de kardex.

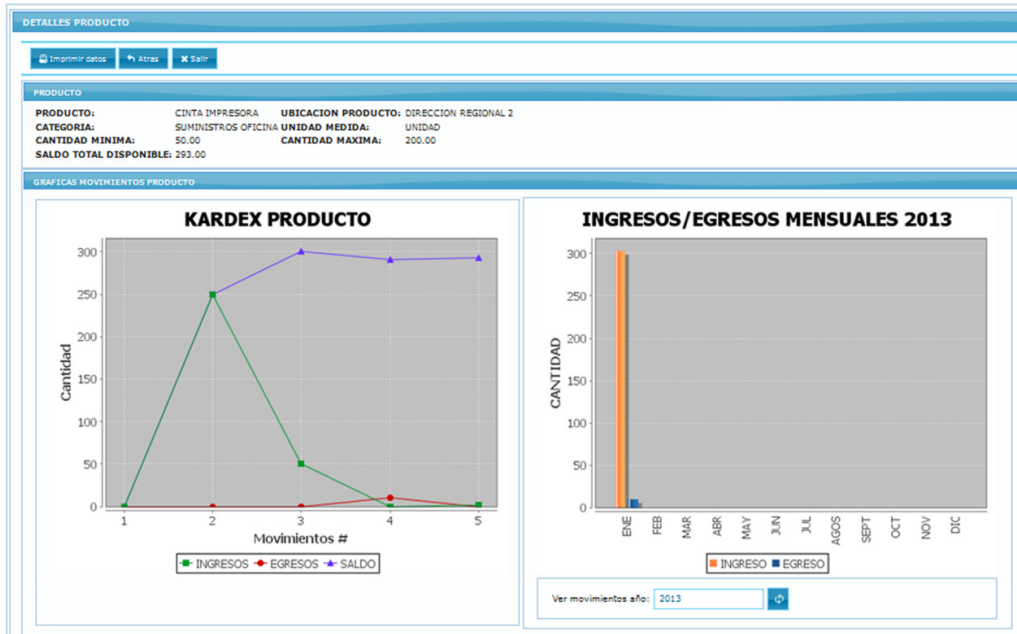


Imagen 32: Graficas de los movimientos del producto.

MOVIMIENTOS PRODUCTOS

FILTRAR POR: Actual | Todos | Fecha | Rango fechas

CRITERIO: [v]

(1 of 1)

FECHA	CONCEPTO	REFERENCIA	INGRESO			EGRESO			SALDO		
			CANT.	COST UNIT.	TOTAL	CANT.	COST UNIT.	TOTAL	CANT.	COST UNIT.	TOTAL
26/01/2013	SALDO INICIAL		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
26/01/2013	COMPRA	fc56	250.00	3.00	750.00	0.00	0.00	0.00	250.00	3.00	750.00
26/01/2013	TRANSFERENCIA	tc3	51.00	3.00	153.00	0.00	0.00	0.00	301.00	3.00	903.00
26/01/2013	ORDEN SALIDA MERCADERIA	RQ23	0.00	0.00	0.00	10.00	3.00	30.00	291.00	3.00	873.00
27/01/2013	COMPRA	fc000123	2.00	54.25	108.50	0.00	0.00	0.00	293.00	3.35	981.50

Imagen 33: Tabla de movimientos del producto.

5.11.9. CONTROL MÉDICO

Registro Historial Médico

Guardar | Salir

Datos Paciente

Nro cédula: 1310019144 Nombre: Alvarado Jimenes Diego Oswaldo
 Dependencia: DIRECCION REGIONAL 2
 Dirección domicilio: CDLA CHAULLABAMBA
 Teléfono Fijo: 072814781 Teléfono Móvil: 0999222035
 Genero: [v] Tipo de Sangre: [v]

Alergias

Lista Alergias

MANI

Alergias Paciente

→
←
←
←
←

Enfermedades

Lista Enfermedades

ASMA

Enfermedades Paciente

→
←
←
←
←

Imagen 34: Formulario para el registro del historial clínico.

REGISTRO CONSULTA MEDICA

✖ Salir

DATOS PACIENTE | DATOS CONSULTA | PRESCRIPCION MEDICA | DATOS A GUARDAR

PACIENTE	
Código Empleado:	ADM03
Nro Cédula:	0101874675
Nombre:	Peña Cordero Zoila Justina
Fecha Nacimiento:	09/05/1962
Estado Civil:	DIVORCIADO
Dirección Domicilio:	ANTISUYO
Teléfono Fijo:	072866281
Teléfono Movil:	098153747
Dependencia:	DELEGACION PROV. MORONIA SANTIAGO
Tipo de sangre:	A+
Género:	MASCULINO

HISTORIAL CLINICO	
ALERGIAS PACIENTE	
MANI	
ENFERMEADES PACIENTE	
ASMA	

→ Next

Imagen 35: Formulario para registro de consultas médicas (parte 1).

REGISTRO CONSULTA MEDICA

✖ Salir

DATOS PACIENTE | **DATOS CONSULTA** | PRESCRIPCION MEDICA | DATOS A GUARDAR

Motivo consulta:

Sintomas:

Diagnostico:

← Back

→ Next

Imagen 36: Formulario para registro de consultas médicas (parte 2).

REGISTRO CONSULTA MEDICA

✖ Salir

DATOS PACIENTE | DATOS CONSULTA | **PRESCRIPCION MEDICA** | DATOS A GUARDAR

Prescripción | MEDICAMENTOS DISPONIBLES

Nombre medicamento:

Cantidad:

Indicación:

➕ Agregar Medicamento | LIMPIAR

MEDICAMENTO	CANTIDAD	INDICACION	OPCIONES
ACETAMINOFEN	12	1 cada 12 horas	✓ ✕
IBUPROFENO	0		✎

Quitar Medicamento

← Back

→ Next

Imagen 37: Formulario para registro de consultas médicas (parte 3.1).

REGISTRO CONSULTA MEDICA

Salir

DATOS PACIENTE | DATOS CONSULTA | **PRESCRIPCION MEDICA** | DATOS A GUARDAR

Prescripción **MEDICAMENTOS DISPONIBLES**

INFORMACION MEDICAMENTOS

Stock normal(1) Stock bajo(0) Stock agotado(1)

Buscar:

	MEDICAMENTO	DEPENDENCIA	UNIDAD MED.	CATEGORIA	TOTAL DISPONIBLE
<input checked="" type="checkbox"/>	ACETAMINOFEN	DIRECCION REGIONAL 2	CAJA 12 UN	MEDICAMENTOS GENERICOS	45.00
<input checked="" type="checkbox"/>	IBUPROFENO	DIRECCION REGIONAL 2	UNIDAD	MEDICAMENTOS GENERICOS	0.00

+ AGREGAR SELECCION

Back Next

Imagen 38: Formulario para registro de consultas médicas (parte 3.2).

REGISTRO CONSULTA MEDICA

Salir

DATOS PACIENTE | DATOS CONSULTA | PRESCRIPCION MEDICA | **DATOS A GUARDAR**

Guardar consulta | Imprimir prescripción

CONSULTA

Fecha consulta: 01/29/2013

Hora consulta: 16:47:28

MOTIVO CONSULTA: CONSULTA GENERAL

SINTOMAS: DOLOR DE CABEZA

DIAGNOSTICO: estres

PRESCRIPCION

ACETAMINOFEN	12	1 CADA 12 HORAS
IBUPROFENO	5	1 CADA 8 HORAS

Back

Imagen 39: Formulario para registro de consultas médicas (parte 4).

5.11.10. GESTIÓN DE CURSO

LISTA CURSOS

Registrar curso | Editar curso | Detalles curso | Imprimir datos | Lista alumnos | Eliminar curso | Salir

Buscar:

(1 of 1)

NOMBRE CURSO	DEPENDENCIA	COORDINADOR	MODALIDAD	FECHA INICIO	FECHA FINALIZACION	TOTAL HORAS	CALIFICACION	ESTADO
JAVA	DIRECCION REGIONAL 2	OSCAR	PRESENCIAL	14/01/2013	18/01/2013	120	5.00	INACTIVO

Total de registros 1

Imagen 40: Interfaz para la gestión de cursos.

REGISTRO CURSOS

Guardar Salir

DATOS CURSOS

Nombre Curso: Nombre Coordinador:

Ciudad del evento: Lugar del evento:

Fecha inicio curso: Fecha finalización curso:

Nro. de horas: Costo del curso:

Modalidad del curso: Tipo aprobación curso:

Calificación otorgada: Estado del curso:

DATOS FACILITADOR - TEMAS

Nombre facilitador:

Especialidad:

Calificación otorgada:

+ Agregar facilitador

FACILITADORES		
NOMBRE FACILITADOR	ESPECIALIDAD	CALIFICACIÓN FACILITADOR
(SIN FACILITADORES REGISTRADOS)		

+ Nuevo tema + Borrar facilitador

TEMAS FACILITADOR	
TEMA	FACILITADOR
(SIN TEMAS REGISTRADOS)	

+ Borrar tema

Imagen 41: Formulario para el registro de cursos.

LISTA ALUMNOS JAVA

Registrar alumno Editar alumno Detalles alumno Eliminar alumno Imprimir lista alumnos Atras Salir

OPCIONES LISTADO

Todos Aprobados Reprobados

(1 of 1)

NRO CEDULA	NOMBRE ALUMNO	ESPECIALIDAD	TIPO	CALIFICACION	ESTADO
0104809504	CASTRO ARTEAGA OSCAR STEVE	INFORMATICA	INTERNO	70.00	APROBADO
1310019144	ALVARADO JIMENES DIEGO OSWALDO	MECCINA	INTERNO	50.00	REPROBADO
1717997355	JIMENES ALVARADO OMAR PEPE	CONTADOR	INTERNO	60.00	REPROBADO

Total de registros: 3

Imagen 42: Interfaz para la gestión de alumnos.

REGISTRO ALUMNOS

Guardar Salir

DATOS ALUMNO

ID curso:

Nombre de curso:

Nro. cédula:

Nombre alumno:

Lugar residencia:

Direccion domicilio:

Teléfono fijo:

Teléfono móvil:

Email:

Institución perteneciente:

Especialidad:

Tipo alumno:

Calificación alumno:

Imagen 43: Formulario para el registro de alumnos.

5.11.11. GESTIÓN MANTENIMIENTOS VEHÍCULO

The screenshot shows a web interface titled "MANTENIMIENTOS VEHICULO". At the top, there are navigation buttons: "Registrar mantenimiento", "Editar mantenimiento", "Eliminar mantenimiento", and "Salir". Below this is a section titled "LISTA MANTENIMIENTOS" with a search bar labeled "Buscar:". The main content is a table with the following data:

MANTENIMIENTO	KM MANTENIMIENTO	TIPO MANTENIMIENTO
CAMBIO ACEITE	5000	PREVENTIVO
CAMBIO NEUMATICOS	100000	CORRECTIVO
CAMBIO BUJIAS	10000	CORRECTIVO
CAMBIO FILTRO ACEITE	50000	PREVENTIVO

At the bottom of the table, there is a pagination control showing "(1 of 1)", navigation arrows, and a page number "10". Below the table, it says "Total de registros 4".

Imagen 44: Interfaz para la gestión de mantenimientos de vehículo.

The screenshot shows a form titled "REGISTRO MANTENIMIENTOS" with "Guardar" and "Salir" buttons. The form contains the following fields:

- Mantenimiento: CAMBIO FILTRO AIRE
- Km mantenimiento: 10000
- Tipo mantenimiento: A dropdown menu with "CORRECTIVO" and "PREVENTIVO" options.

Imagen 45: Formulario para el registro de mantenimientos de vehículo.

5.11.12. GESTIÓN VEHÍCULOS

The screenshot shows a web interface titled "VEHICULOS". At the top, there are navigation buttons: "Registrar Vehículo", "Editar Vehículo", "Detalles Vehículo", "Restaurar vehículo", and "Eliminar vehículo". Below this is a section titled "LISTA VEHICULOS" with a search bar labeled "Buscar:". The main content is a table with the following data:

PLACA VEHICULO	DEPENDENCIA PERTENECIENTE	KM ACTUAL	ESTADO
CBK0331	DIRECCION REGIONAL 2	120010	DISPONIBLE

At the bottom of the table, there is a pagination control showing "(1 of 1)", navigation arrows, and a page number "10". Below the table, it says "Total de registros 1".

Imagen 46: Interfaz para la gestión de vehículos.

The screenshot shows a form titled "REGISTRO VEHICULOS" with "Guardar" and "Salir" buttons. The form contains the following fields:

- Marca: [Empty text box]
- Número placa: AAA0123
- Modelo: [Empty text box]
- Color: [Empty text box]
- Año fabricación: 0
- Km actual: 0
- Nro chasis: [Empty text box]
- Dependencia perteneciente: [Dropdown menu with "----" selected]

Imagen 47: Formulario para el registro de vehículos.

5.11.13. GESTIÓN HISTORIAL MANTENIMIENTOS VEHÍCULO

LISTA MANTENIMIENTOS VEHICULOS - CBK0331

Registrar historial mantenimiento Ver historial mantenimiento Atras Salir

Buscar:

(1 of 1) 1 10

INFO	MANTENIMIENTO	TIPO MANTENIMIENTO	KM ACTUAL	KM PROXIMO MANTENIMIENTO	ULTIMA REVISION	ESTADO
✓	CAMBIO ACEITE	PREVENTIVO	120010	125010	27/01/2013	OK
✓	CAMBIO NEUMATICOS	CORRECTIVO	120010	220010		OK
✓	CAMBIO BUJIAS	CORRECTIVO	120010	130010		OK
✓	CAMBIO FILTRO ACEITE	PREVENTIVO	120010	170010		OK

Total de registros 4

Imagen 48: Interfaz para la gestión de historia de mantenimientos de vehículo.

INGRESO CONTROL MANTENIMIENTO VEHICULO - CBK0331

Salir

SELECCION MANTENIMIENTOS INGRESO DATOS

Buscar:

(1 of 1) 1 10

<input type="checkbox"/>	INFO	MANTENIMIENTO	TIPO MANTENIMIENTO	KM ACTUAL	KM PROXIMO MANTENIMIENTO	ULTIMA REVISION	ESTADO MANTENIMIENTO
<input checked="" type="checkbox"/>	✓	CAMBIO ACEITE	PREVENTIVO	120010	125010	27/01/2013	OK
<input type="checkbox"/>	✓	CAMBIO NEUMATICOS	CORRECTIVO	120010	220010		OK
<input type="checkbox"/>	✓	CAMBIO BUJIAS	CORRECTIVO	120010	130010		OK
<input checked="" type="checkbox"/>	✓	CAMBIO FILTRO ACEITE	PREVENTIVO	120010	170010		OK

+ Agregar mantenimiento seleccionado

Imagen 49: Formulario para el registro de mantenimientos de vehículo (parte 1).

INGRESO CONTROL MANTENIMIENTO VEHICULO - CBK0331

Salir

SELECCION MANTENIMIENTOS INGRESO DATOS

Guardar

Km actual: 120010
Nuevo Km:

#	MANTENIMIENTO	KM MANTENIMIENTO	FECHA DE CONTROL	KM DE CONTROL	MARCA PRODUCTO	CODIGO PRODUCTO
✓ X 1	CAMBIO FILTRO ACEITE	170010	<input type="text" value="dia/mes/año"/>	<input type="text" value="120010"/>	<input type="text"/>	<input type="text"/>
2	CAMBIO ACEITE	125010		120010		

Imagen 50: Formulario para el registro de mantenimientos de vehículo (parte 2).

5.11.14. GESTIÓN SOLICITUD VEHÍCULO

LISTA SOLICITUDES VEHICULO

Registrar solicitud vehiculo Ver detalles solicitud vehiculo Salir

Buscar por fecha:

Listar solicitudes: Hoy Todas Pendientes

Buscar:

NRO DOCUMENTO SOLICITUD	NOMBRE SOLICITANTE	DESTINO	MOTIVO	FECHA MOVILIZACION	HORA MOVILIZACION	ESTADO SOLICITUD
85412	Castro Arteaga Oscar Steve	AMBATO	REUNION AMBATO	31/01/2013	10:00	PENDIENTE
45216	Castro Arteaga Oscar Steve	QUITO	RUENION QUITO	31/01/2013	12:30	CANCELADA

Imagen 51: Interfaz para la gestión de solicitudes de vehículos.

REGISTRO SOLICITUD VEHICULOS

Guardar Salir

DATOS SOLICITUD

Buscar empleado (nro. cédula):

Fecha solicitud: 29/01/2013
 Hora solicitud: 18:09
 Nombre solicitante:
 Nro documento solicitud: 0

Motivo solicitud:
 Destino:
 Fecha de movilización: dia/mes/año
 Hora de movilización: hora:minutos

LISTA EQUIPO TRABAJO

#	NOMBRE COMPLETO	OPCIONES
+ Agregar campo Quitar		

Imagen 52: Formulario para el registro de solicitudes vehículo.

5.11.15. GESTIÓN ORDENES ED MOVILIZACIÓN

DETALLES SOLICITUD VEHICULOS

Generar Orden Movilizacion Anular solicitud vehiculo Salir

DATOS SOLICITUD

Nro solicitud: 2
 Nro documento solicitud: 85412
 Nombre solicitante: Castro Arteaga Oscar Steve
 Fecha solicitud: 29/01/2013
 Hora solicitud: 17:44

Motivo solicitud: REUNION AMBATO
 Destino: AMBATO
 Fecha de movilización: 31/01/2013
 Hora de movilización: 10:00
 Estado solicitud: PENDIENTE

LISTA EQUIPO TRABAJO

--

Imagen 53: Interfaz para la gestión de órdenes de movilización.

REGISTRO ORDEN MOVILIZACION

Guardar Salir

DATOS SOLICITUD		DATOS ORDEN MOVILIZACION	
Nro Solicitud:	1	Fecha de salida:	29/01/2013
Nro documento solicitud:	85412	Hora de salida:	18:12
Nombre solicitante:	Castro Arteaga Oscar Steve	Vehículo asignado:	---
Motivo solicitud:	REUNION AMBATO	Km actual:	0
Destino:	AMBATO	Chofer asignado:	---
# Equipo trabajo:	0		

Imagen 54: Formulario para el registro de orden de movilización.

LISTA ORDENES MOVILIZACION

Detalles orden movilizacion Salir

BUSCAR POR: Todos

Buscar:

NRO ORDEN	NOMBRE SOLICITANTE	FECHA SALIDA	DESTINO	MOTIVO	HORA SALIDA	ESTADO
3	Castro Arteaga Oscar Steve	29/01/2013	AMBATO	REUNION AMBATO	18:17	EJECUCION

Total registros: 1

Imagen 55: Interfaz para la gestión de las órdenes de movilización.

DETALLES ORDEN MOVILIZACION

Finalizar Cancelar Salir

NRO ORDEN	3	DOC SOLICITUD	85412
NOMBRE SOLICITANTE	Castro Arteaga Oscar Steve	FECHA SALIDA	29/01/2013
HORA SALIDA	18:17	DESTINO	AMBATO
MOTIVO	REUNION AMBATO	CHOFER ASIGNADO	Maldonado Cardenas Anibal Ernesto
VEHICULO ASIGNADO	CBK0331	KM SALIDA	120010
FECHA LLEGADA	29/01/2013	HORA LLEGADA	18:24
KM LLEGADA		ESTADO ORDEN:	E

Imagen 56: Formulario para terminación o cancelación de orden de movilización.

6. CAPÍTULO VI – FASE DE DESARROLLO

En este capítulo se describirán los principales archivos de configuración de la aplicación para una mejor comprensión de los mismos.

Aquí veremos cómo está estructurado el archivo de persistencia.xml, faces-config.xml, web.xml y demás archivos de configuración importantes.

Los archivos de configuración son archivos XML que contienen las estructuras de datos lógicas de configuración utilizados para establecer o cambiar valores en la aplicación sin la necesidad de compilarla nuevamente, estos archivos utilizan etiquetas para marcar el inicio y el final de un elemento de configuración.

6.1. ARCHIVOS DE CONFIGURACION

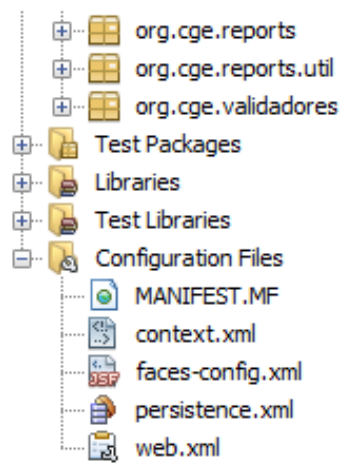


Imagen 57: Ubicación de archivos de configuración en Netbeans.

6.1.1. FACES-CONFIG.XML

Es el archivo de configuración principal de JavaServer Faces, aquí es donde se especifican: eventos, acciones, beans de control, reglas de navegación entre otros.

6.1.1.1. ETIQUETAS PRINCIPALES DE FACES-CONFIG.XML

- application: definir las distintas clases de implementación para una aplicación web, por medio de la etiqueta el-resolver se puede personalizar las variables y el comportamiento de las propiedades de expresiones EL (Expression Lenguaje).

```
<application>
  <resource-bundle>
    <base-name>/Bundle</base-name>
    <var>bundle</var>
  </resource-bundle>
  <el-resolver>
    org.cge.entities.beans.controllers.util.JsfCrudELResolver
  </el-resolver>
</application>
```

- lifecycle: listener del ciclo de vida, permite realizar una acción o un seguimiento en los diferentes ciclos de vida de la página.

```
<lifecycle>
  <phase-listener>
    org.cge.filtros.FiltroAutenticacion
  </phase-listener>
</lifecycle>
```

- validator: permite especificar una clase de validación específica. Se debe especificar un identificador y la clase del validador.

```
<validator>
  <validator-id>validarCedulaRUC</validator-id>
  <validator-class>org.cge.jsf.validadores.ValidarCedulaRUC
</validator-class>
</validator>
```


- converter: permite especificar una clase de conversión específica. Se debe especificar un identificador y la clase del convertidor.

```
<converter>
  <converter-id>lowercaseConverter</converter-id>
  <converter-class>
    org.cge.jsf.converter.LowercaseConverter</converter-class>
</converter>
```

- navigation-rule: representa una regla de navegación que será utilizado por la aplicación para decidir sobre qué vista debe aparecer, se base en obtener una cadena lógica por medio de una acción para luego resolver la regla de navegación si esta cumple con la condición establecido. S debe establecer el caso, la cadena lógica y la acción a efectuar.

```
<navigation-rule>
  <navigation-case>
    <from-outcome>welcome</from-outcome>
    <to-view-id>/index.xhtml</to-view-id>
    <redirect/>
  </navigation-case>
</navigation-rule>
```

6.1.2. PERSISTENCE.XML

Persistence.xml contendrá las unidades de persistencia o persistence unit, aquí es donde especificaremos con que implementación JPA vamos a trabajar, el grupo de entidades, las propiedades de conexión a la base de datos, entre otros parámetros más.

6.1.2.1. ETIQUETAS PRINCIPALES DE PERSISTENCE.XML

- persistence-unit: define la unidad de persistencia:

```
< persistence-unit  
name="CONTRESSPU"  
transaction-type="RESOURCE_LOCAL">
```

- provider: Especifica la implementación JPA que utilizaremos:

```
<provider>oracle.toplink.essentials.PersistenceProvider</provider>
```

- class: especificamos el grupos de entidades (una por cada clase):

```
<class>org.cge.entities.AlmacenProductos</class>  
<class>...</class>
```

- properties: establecemos las propiedades de conexión a la Base de datos:

```
<properties>  
<property name="toplink.jdbc.user" value="root"/>  
<property name="toplink.jdbc.password" value="123"/>  
<property name="toplink.jdbc.url" value="jdbc:postgresql://localhost:5432/db"/>  
<property name="toplink.jdbc.driver" value="org.postgresql.Driver"/>  
</properties>
```

6.1.3. WEB.XML

El archivo web.xml de configuración o descriptor de despliegue (DD) que establece la forma en que debe iniciar una aplicación web, proporciona las clases, los recursos las configuraciones que utilizará la aplicación web y él como deberá manejarlos el servidor web al responder las peticiones del usuario.

6.1.3.1. ETIQUETAS PRINCIPALES DE WEB.XML

- servlet: permite declarar la configuración de los servlets.

```
<servlet>
  <servlet-name>Resource Servlet</servlet-name>
  <servlet-class>
    org.primefaces.resource.ResourceServlet
  </servlet-class>
</servlet>
```

- context-param: permite la configurar la inicialización del contexto de una servlet.

```
<context-param>
  <param-name>
    javax.faces.FACELETS_REFRESH_PERIOD
  </param-name>
  <param-value>1</param-value>
</context-param>
```

- servlet-mapping: relaciona los servlets para las solicitudes cuyas URL coinciden con cierto patrón.

```
<servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>*.xhtml</url-pattern>
</servlet-mapping>
```

- session-config: permite establecer parámetros para configurar la sesión.

```
<session-config>
  <session-timeout>30</session-timeout>
</session-config>
```

- welcome-file-list: especifica los archivos de bienvenida.

```
<welcome-file-list>
    <welcome-file>index.xhtml</welcome-file>
</welcome-file-list>
```

- filter: permite especificar filtros que nos ayudaran a procesar solicitudes.

```
<filter>
    <filter-name>DisableUrlSessionFilter</filter-name>
    <filter-class>org.cge.filtros.DisableUrlSessionFilter</filter-
class>
</filter>
```

- filter-mapping: asigna un filtro a múltiples URL.

```
<filter-mapping>
    <filter-name>CustomCharacterEncodingFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

- error-page: permite asociar errores y excepciones en HTML.

```
<error-page>
    <error-code>404</error-code>
    <location>/errorPages/error400.html</location>
</error-page>
<error-page>
    <exception-type>
        javax.faces.application.ViewExpiredException
    </exception-type>
    <location>/index.xhtml</location>
</error-page>
```

7. CAPÍTULO VII – PRUEBAS DEL SISTEMA

7.1. PRUEBAS DEL SISTEMA

Las pruebas de sistemas son procesos en el cual se ejecuta un programa o una parte del mismo con la intención de descubrir errores, pues es muy habitual cometer algún tipo de equivocación al momento de generar un procedimiento, módulo, etc.

El objetivo entonces es mostrar un error no descubierto mediante el diseño de casos de prueba que revelen errores que puedan existir dentro del sistema.

Existen dos formas en las que se pueden realizar las pruebas del sistema:

- Pruebas de caja blanca: se analiza desde la lógica del diseño procedimental.
- Pruebas de caja negra: se llevan a cabo sobre la interfaz del software sin considerar la lógica interno o la estructura del programa.

7.1.1. PRUEBAS MÓDULO USUARIO

7.1.1.1. PRUEBA CAJA BLANCA

CREACION NOMBRE USUARIO

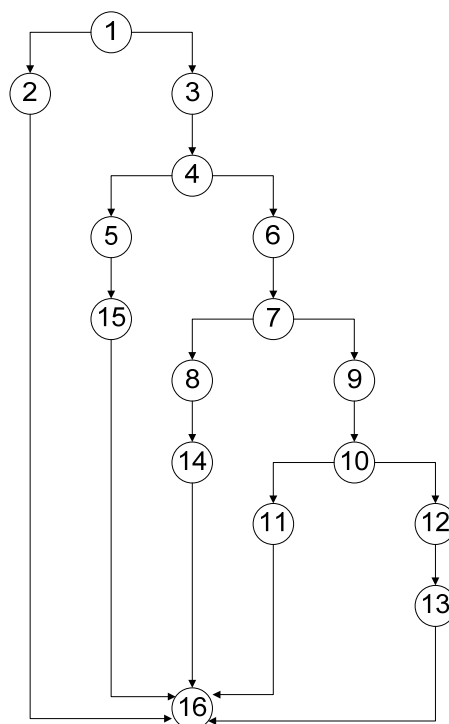
```
public void crearNombreUsuario(){
    crearNombreUsuario(true, false, false, false);
    1- if(getUsuariosJpaController().findUsuariosEntities(getNombreUsuario())==null){
    usuario.setNombreUsuario(getNombreUsuario());
    2- }
    3- else{
    crearNombreUsuario(true, true, false, false);
    4- if(getUsuariosJpaController().findUsuariosEntities(getNombreUsuario())==null){
    usuario.setNombreUsuario(getNombreUsuario());
    5- }
    6- else{
```

```

crearNombreUsuario(true, true, true, false);
7-if(getUsuariosJpaController().findUsuariosEntities(getNombreUsuario())==null){
usuario.setNombreUsuario(getNombreUsuario());
8- }
9- else{
crearNombreUsuario(true, true, true, true);
10-if(getUsuariosJpaController().findUsuariosEntities(getNombreUsuario())==null){
usuario.setNombreUsuario(getNombreUsuario());
auxUsuario = 1;
11- }
12- else{
crearNombreUsuario();
13- }
14- }
15- }
16- }
}

```

FLUJOGRAMA



COMPLEJIDAD CICLOMÁTICA

- $V(G) = \text{Aristas} - \text{Nodos} + 2$
- $V(G) = 19 - 16 + 2$
- $V(G) = 5$

CAMINOS INDEPENDIENTES

- 1, 2, 16
- 1, 3, 4, 5, 15, 16
- 1, 3, 4, 6, 7, 8, 14, 16
- 1, 3, 4, 6, 7, 9, 10, 11, 16
- 1, 3, 4, 6, 7, 9, 10, 12, 13, 16

RESULTADOS DE LA PRUEBA

CAMINO	VALOR	RESULTADO
1	Crespo Albarracin Juan Carlos	jcespo
2	Crespo Albarracin Juan Carlos	jcespoa
3	Crespo Albarracin Juan Carlos	jccrespoa
4	Crespo Albarracin Juan Carlos	jccrespoa1
5	Crespo Albarracin Juan Carlos	jccrespoa2...

Tabla 10: Resultados caja blanca creación nombre usuario.

7.1.1.2. PRUEBA CAJA NEGRA

ESCENARIOS

- Registro de rol
- Registro de permisos rol
- Registro de usuario

DESCRIPCIÓN

En el caso de prueba mencionado se indican los valores correctos que debe ingresar el usuario, para realizar una solicitud de administración de parámetros.

CONDICIONES DE EJECUCIÓN

- Se debe iniciar correctamente la sesión en el sistema.
- Únicamente el usuario administrador podrá crear nuevos usuarios.

ENTRADA

- Seleccionar la opción Agregar Rol
- Ingresa nombre del rol
- Seleccionar la opción Agregar Permisos Rol
- Selecciona los permisos del rol
- Seleccionar la opción Registrar Usuario
- Ingresa número de cédula
- Ingresa contraseña temporal
- Selecciona el rol de usuario

RESULTADO ESPERADO

Al registrar un nuevo rol se pretende obtener los siguientes resultados:

- Campo nombre de rol obligatorio.
- Si ya está registrado el rol, nos muestra un mensaje de rol ya registrado.

Al registrar los permisos del rol se pretende obtener los siguientes resultados:

- El usuario accede a los roles que se le asigna.

Al registrar un usuario se pretende obtener los siguientes resultados:

- Campo número de cédula acepta solo caracteres numéricos.
- Campo numero de cédula obligatorio
- Campo contraseña temporal es obligatorio.
- Campo rol de usuario obligatorio.

Si existe el usuario muestra mensaje de usuario ya registrado.

RESULTADO OBTENIDO

Al registrar los datos del usuario el sistema realiza las siguientes validaciones:

DATOS DE ENTRADA		DATOS DE SALIDA
CEDULA	abcd	No acepta alfabéticos
	01048	No acepta número de caracteres menores a 10
	1709221293	Cédula correcta
	1728820201	Cédula incorrecta

Tabla 11: Resultados pruebas caja negra registro usuario.

7.1.2. PRUEBAS MÓDULO EMPLEADOS

7.1.2.1. PRUEBA CAJA BLANCA

VALIDACION DE CEDULA

```

1- public static boolean esCedulaValida(String cedula) {
2- if (!(cedula.length() == 10) && cedula.matches("[0-9]{10}$")) {
return false;
3- }
int prov = Integer.parseInt(cedula.substring(0, 2));
4- if (!(prov > 0) && (prov <= NUMERO_DE_PROVINCIAS)) {
return false;
5- }
int[] d = new int[10];
for (int i = 0; i < d.length; i++) {
d[i] = Integer.parseInt(cedula.charAt(i) + "");

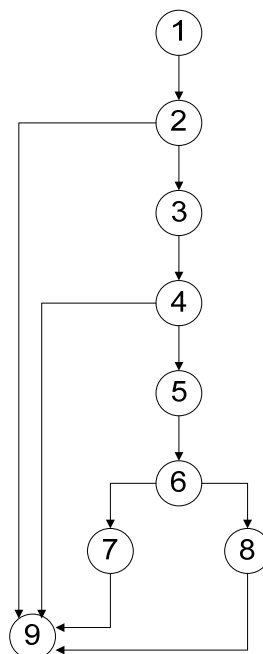
```

```

}
int imp = 0;
int par = 0;
for (int i = 0; i < d.length; i += 2) {
d[i] = ((d[i] * 2) > 9) ? ((d[i] * 2) - 9) : (d[i] * 2);
imp += d[i];
}
for (int i = 1; i < (d.length - 1); i += 2) {
par += d[i];
}
int suma = imp + par;
int d10 = Integer.parseInt(String.valueOf(suma + 10).substring(0, 1) + "0") - suma;
6- d10 = (d10 == 10) ? 0 : d10;
return d10 == d[9]; (7-) (8-)
9- }

```

FLUJOGRAMA



COMPLEJIDAD CICLOMÁTICA

- $V(G) = \text{Aristas} - \text{Nodos} + 2$
- $V(G) = 11 - 9 + 2$
- $V(G) = 4$

CAMINOS INDEPENDIENTES

- 1, 2, 9
- 1, 2, 3, 4, 9
- 1, 2, 3, 4, 5, 6, 7, 9
- 1, 2, 3, 4, 5, 6, 8, 9

RESULTADOS DE LA PRUEBA

CAMINO	VALOR	RESULTADO
1	01048as	“No valida”
2	8801245213	“No valida”
3	0200832251	“Valida”
4	1702619302	“No valida”

Tabla 12: Resultados prueba caja blanca validación cédula.

7.1.2.2. PRUEBA CAJA NEGRA

ESCENARIOS

- Registro de empleados.

DESCRIPCIÓN

- Se registra los datos de un nuevo empleado y su cargo actual.

CONDICIONES DE EJECUCIÓN

- Se debe iniciar correctamente la sesión en el sistema.

ENTRADA

- Seleccionar menú Empleados
- Seleccionar la opción Registrar Empleado
- Ingresar los datos del empleado:
 - Código de empleado
 - Nro. de cédula
 - Primer nombre
 - Segundo nombre
 - Apellido paterno
 - Apellido Materno
 - Fecha de nacimiento
 - Estado civil
 - Dirección domicilio
 - Teléfono fijo
 - Teléfono móvil
 - Email personal
 - Email trabajo
 - Dependencia perteneciente
 - Fecha ingreso institución
- Ingresar datos del cargo actual
 - Departamento actual
 - Cargo actual
 - Fecha asignación cargo actual

RESULTADO ESPERADO

Al registrar un nuevo empleado se pretende obtener los siguientes resultados:

- Código de empleado obligatorio
- Código de empleado debe contener únicamente datos numéricos
- Nro. de cédula obligatorio
- Nro. de cédula acepta solo campos numéricos

- Primer nombre obligatorio
- Segundo nombre obligatorio
- Apellido paterno obligatorio
- Apellido Materno obligatorio
- Fecha de nacimiento obligatorio
- Estado civil obligatorio
- Dirección domicilio obligatorio
- Teléfono fijo solo acepta caracteres numéricos
- Teléfono fijo no es obligatorio
- Teléfono móvil solo acepta caracteres numéricos
- Teléfono móvil no es obligatorio
- Email personal no es obligatorio
- Email trabajo no es obligatorio
- Dependencia perteneciente obligatorio
- Fecha ingreso institución obligatorio

Ingresar datos del cargo actual:

- Departamento actual obligatorio
- Cargo actual obligatorio
- Fecha asignación cargo actual obligatorio

RESULTADO OBTENIDO

Tras el registro el sistema realiza las siguientes validaciones:

DATOS DE ENTRADA		DATOS DE SALIDA
Cedula	abcd	No acepta alfabéticos
	01048	No acepta número de caracteres menores a 10
	1709221293	Cédula correcta
	1728820201	Cédula incorrecta
Emails	jfernandez	Correo no valido
	jfernandez.com	Correo no valido
	jfernandez@.com	Correo no valido

	jfernandez@cge.com	Correo valido
	jfernandez@hotmail.com	Correo valido
Fecha ingreso institución	-Fecha nacimiento: 20/01/1980 05/10/1970	“FECHA INGRESO INSTITUCIÓN NO PUEDE SER MENOR A FECHA NACIMIENTO”
Fecha asignación cargo	-Fecha ingreso institución: 10/10/2001 04/10/2000	“FECHA ASIGNACIÓN CARGO NO PUEDE SER MENOR A FECHA DE INGRESO INSTITUCION”

Tabla 13: Resultados caja negra registro empleado.

7.1.3. PRUEBAS MÓDULO PRODUCTOS

7.1.3.1. PRUEBA CAJA BLANCA

TRIGGER KARDEX

```

1- BEGIN
2- IF NEW.cantidad_ingreso IS NULL
3- THEN
RAISE EXCEPTION 'CANTIDAD DE INGRESO NO PUEDE SER NULA';
4- END IF;
5- IF NEW.costo_unit_ingreso IS NULL
6- THEN
RAISE EXCEPTION 'COSTO UNITARIO DE INGRESO NO PUEDE SER
NULO', NEW.costo_unitario;
7- END IF;
8- IF NEW.cantidad_ingreso < 0
9- THEN
RAISE EXCEPTION 'CANTIDAD NO PUEDE SER NEGATIVO', NEW.cantidad;
10- END IF;
11- IF NEW.costo_unit_ingreso < 0
12- THEN

```

```

RAISE EXCEPTION 'COSTO UNITARIO NO PUEDE SER NEGATIVO',
NEW.costo_unitario;
13- END IF;
SELECT saldo,costo_unit_saldo,valor_total_saldo
INTO var_cantidad,var_costoUnit,var_total
FROM kardex_producto
WHERE id_kardex = (SELECT max(id_kardex) FROM kardex_producto WHERE
id_almacen_producto = NEW.id_almacen_producto);
14- IF upper(NEW.movimiento) = upper('N')
15- THEN
NEW.saldo := NEW.cantidad_ingreso;
NEW.costo_unit_saldo := NEW.costo_unit_ingreso;
NEW.valor_total_saldo :=NEW.saldo*NEW.costo_unit_ingreso;
NEW.fecha:=to_char(now(),'dd/mm/YYYY');
UPDATE almacen_productos SET total_producto = New.saldo where
id_almacen_producto = NEW.id_almacen_producto;
16- ELSE
17- IF NEW.movimiento = 'I'
18- THEN
NEW.valor_total_ingreso := NEW.cantidad_ingreso*NEW.costo_unit_ingreso;
NEW.fecha:=to_char(now(),'dd/mm/YYYY');
NEW.saldo:=(var_cantidad+NEW.cantidad_ingreso-NEW.cantidad_salida);
19- IF NEW.saldo = 0
20- THEN
NEW.valor_total_saldo:=NEW.saldo;
NEW.costo_unit_saldo:=NEW.saldo;
21- ELSE
NEW.valor_total_saldo:=(var_total+NEW.valor_total_ingreso);
NEW.costo_unit_saldo:=NEW.valor_total_saldo/NEW.saldo;
22- END IF;
UPDATE almacen_productos SET total_producto = New.saldo where
id_almacen_producto = NEW.id_almacen_producto;
23- ELSE
24- IF NEW.movimiento = 'S'

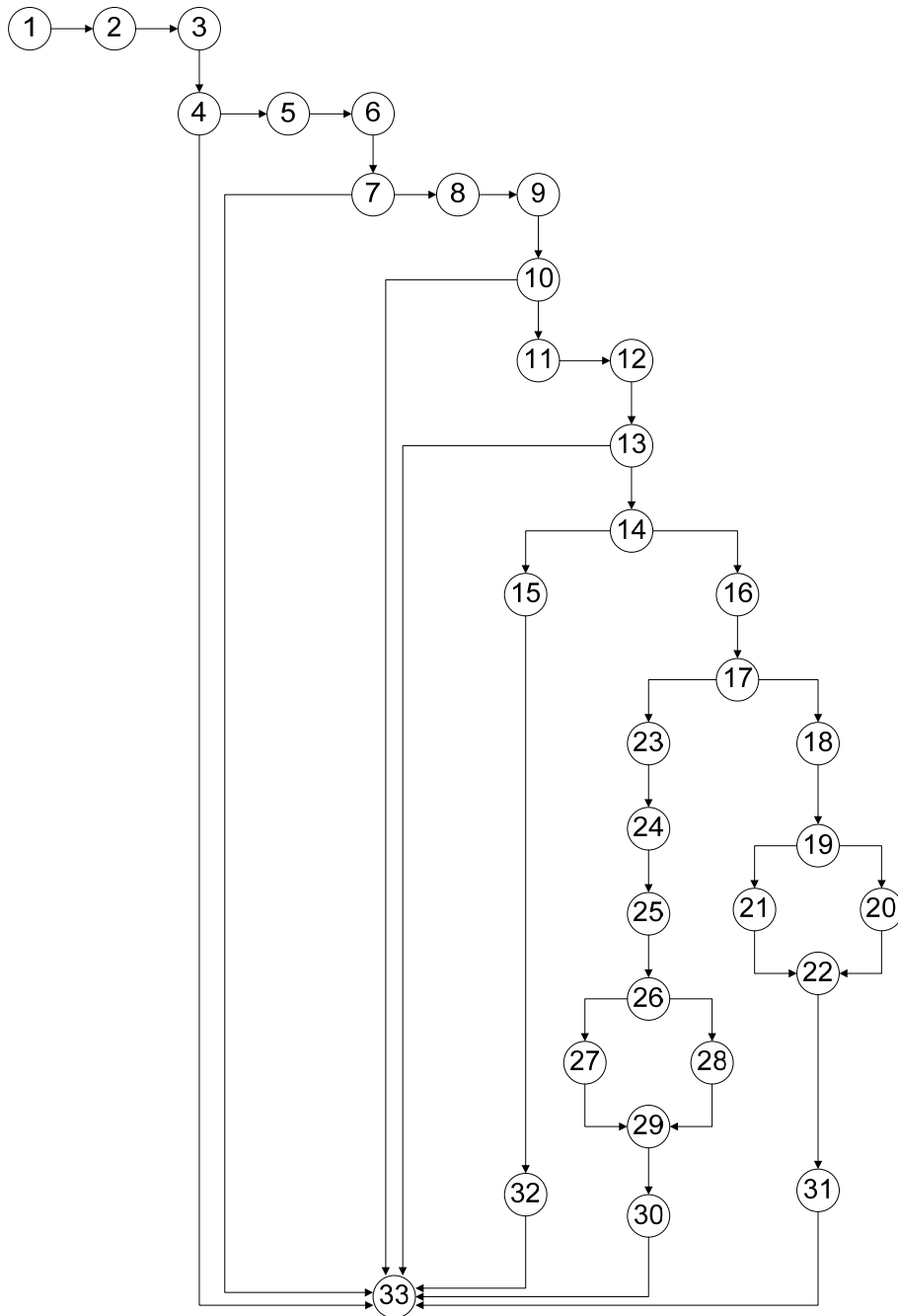
```

```

25- THEN
NEW.valor_total_salida := NEW.cantidad_salida*var_costoUnit;
NEW.costo_unit_salida:=var_costoUnit;
NEW.fecha:=to_char(now(),'dd/mm/YYYY');
NEW.saldo:=(var_cantidad+NEW.cantidad_ingreso-NEW.cantidad_salida);
26- IF NEW.saldo = 0
27- THEN
        NEW.valor_total_saldo:=NEW.saldo;
        NEW.costo_unit_saldo:=NEW.saldo;
28- ELSE
        NEW.valor_total_saldo:=(var_total-NEW.valor_total_salida);
        NEW.costo_unit_saldo:=NEW.valor_total_saldo/NEW.saldo;
29- END IF;
UPDATE almacen_productos SET total_producto = New.saldo where
id_almacen_producto = NEW.id_almacen_producto;
30- END IF;
31- END IF;
32- END IF;
RETURN NEW;
33- END;

```


FLUJOGRAMA



COMPLEJIDAD CICLOMÁTICA

- $V(G) = \text{Aristas} - \text{Nodos} + 2$
- $V(G) = 40 - 33 + 2$
- $V(G) = 9$

CAMINOS INDEPENDIENTES

- 1,2,3,4,33
- 1,2,3,4,5,6,7,33
- 1,2,3,4,5,6,7,8,9,10,33
- 1,2,3,4,5,6,7,8,9,10,11,12,13,33
- 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,32,33
- 1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,20,22,31,33
- 1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,21,22,31,33
- 1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,23,24,25,26,27,29,30,33
- 1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,23,24,25,26,28,29,30,33

RESULTADOS DE LA PRUEBA

CAMINO	VALOR	RESULTADO
1	movimiento=N cantidad_ingreso=null costo_unit_ingreso=null valor_total_ingreso=0	'CANTIDAD DE INGRESO NO PUEDE SER NULA'
2	movimiento=N cantidad_ingreso=null costo_unit_ingreso=null valor_total_ingreso=0	'COSTO UNITARIO DE INGRESO NO PUEDE SER NULO'
3	movimiento=N cantidad_ingreso=-25 costo_unit_ingreso=0 valor_total_ingreso=0	'CANTIDAD NO PUEDE SER NEGATIVO'
4	movimiento=N cantidad_ingreso=0 costo_unit_ingreso=-25 valor_total_ingreso=0	'COSTO UNITARIO NO PUEDE SER NEGATIVO'
5	movimiento=N cantidad_ingreso=0	Registra saldo inicial

	costo_unit_ingreso=0 valor_total_ingreso=0	
6	movimiento=I cantidad_ingreso=10 costo_unit_ingreso=2.5	Registra ingreso y actualiza saldo del producto.
7	movimiento=I cantidad_ingreso=5 costo_unit_ingreso=3.5	Registra ingreso y actualiza saldo del producto.
8	movimiento=S cantidad_salida=5	Registra salida y actualiza saldo del producto.
9	movimiento=S cantidad_salida=5	Registra salida y actualiza saldo del producto.

Tabla 14: Resultados prueba caja blanca kardex producto.

7.1.3.2. PRUEBA CAJA NEGRA

ESCENARIOS

- Registro de stock de producto.

DESCRIPCIÓN

- Se ingresa un nuevo stock de productos.

CONDICIONES DE EJECUCIÓN

- Se debe iniciar correctamente la sesión en el sistema.

ENTRADA

- Seleccionar menú Almacén
- Seleccionar la opción Agregar Stock Productos
- Seleccionar o buscar producto(s).
- Ingresar el nuevo stock del producto:
 - Concepto

- Referencia
- Cantidad ingresa
- Costo unitario

RESULTADO ESPERADO

Al registrar un nuevo stock de producto se obtuvieron los siguientes resultados:

- Seleccionar el o los productos de los cuales se desea ingresar el nuevo stock.
- Campo cant. ingreso acepta solo caracteres numéricos.
- Campo costo unit. acepta solo caracteres numéricos.
- Campo cant. ingreso acepta solo números mayores a 0.
- Campo costo unit. acepta solo números mayores a 0.

RESULTADO OBTENIDO

Tras el registro el sistema realiza las siguientes validaciones:

DATOS DE ENTRADA		DATOS DE SALIDA
cant ingreso	abc	No acepta datos.
costo unit	abc	No acepta datos.
cant ingreso	-5	CANTIDAD INGRESO no valido en producto
costo unit	-5	CANTIDAD COSTO UNIT no valido en producto
cant ingreso	0	CANTIDAD INGRESO no valido en producto
costo unit	0	CANTIDAD COSTO UNIT no valido en producto
cant ingreso	15	Ingreso Stock - Registrado satisfactoriamente
costo unit	0.20	

Tabla 15: Resultados caja negra registró stock producto.

7.1.4. PRUEBAS MÓDULO CURSOS

7.1.4.1. PRUEBA CAJA BLANCA

- No se efectúan pruebas de caja blanca.

7.1.4.2. PRUEBA CAJA NEGRA

ESCENARIOS

- Registro cursos.

DESCRIPCIÓN

- Se registra los datos de un curso.

CONDICIONES DE EJECUCIÓN

- Se debe iniciar correctamente la sesión en el sistema.

ENTRADA

- Seleccionar menú Cursos
- Seleccionar la opción Registrar Curso
- Ingresar los datos del curso:
 - Nombre curso
 - Nombre coordinador
 - Ciudad del evento
 - Lugar del evento
 - Fecha inicio curso
 - Fecha finalización curso
 - Nro. de horas.
 - Costo del curso
 - Modalidad del curso

- Tipo aprobación curso
- Calificación curso
- Estado del curso
- Ingresar datos facilitadores:
 - Nombre facilitador
 - Especialidad
 - Calificación facilitador
- Ingresamos temas del curso:
 - Nombre de tema

RESULTADO ESPERADO

Al registrar un curso se pretende obtener los siguientes resultados:

- Ingresar los datos del curso:
 - Nombre curso obligatorio
 - Nombre coordinador obligatorio
 - Ciudad del evento obligatorio
 - Lugar del evento obligatorio
 - Fecha inicio curso obligatorio
 - Fecha finalización curso obligatorio
 - Nro. de horas:
 - acepta solo dígitos numéricos
 - acepta solo números mayores a 0.
 - Costo del curso
 - Obligatorio
 - Acepta solo dígitos numéricos
 - Acepta solo números mayores a 0.
 - Modalidad del curso obligatorio
 - Tipo aprobación curso obligatorio
 - Calificación obligatorio
 - Calificación debe estar en el rango de 0-5
 - Estado del curso obligatorio
- Ingresar datos facilitadores:

- Nombre facilitador
 - Obligatorio
 - No se acepta duplicado
- Especialidad obligatorio
- Calificación obligatorio
 - Calificación debe estar en el rango de 0-5
- Ingresamos temas del curso:
 - Nombre de tema
 - Obligatorio
 - No se acepta duplicado

RESULTADO OBTENIDO

Tras el registro el sistema realiza las siguientes validaciones:

DATOS DE ENTRADA		DATOS DE SALIDA
Nro. de horas	5	Valor valida
	-5	Valor no valida
	ABC	Nro. horas debe ser un número entre - 2147483648 y 2147483647.
Costo curso	5	Valor valida
	-5	Valor no valida
	ABC	Nro. horas debe ser un número entre - 2147483648 y 2147483647.

Tabla 16: Resultados pruebas caja negra registro curso.

7.1.5. PRUEBAS MÓDULO VEHÍCULOS

7.1.5.1. PRUEBA CAJA BLANCA

VALIDACION PLACAS VEHICULOS

```
private static final String PLACA_REGEXP = "[A-Za-z]{3}[0-9]{4}";
```

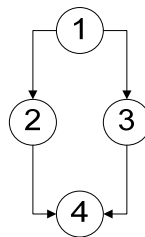
```
1- public static boolean validarPlacaVehiculo(String placa)
```

```

{
    Pattern mascara = null;
    mascara = Pattern.compile(PLACA_REGEX);
    Matcher coincidencia = mascara.matcher(placa);
    return coincidencia.matches(); (2) (3)
4- }

```

FLUJOGRAMA



COMPLEJIDAD CICLOMÁTICA

- $V(G) = \text{Aristas} - \text{Nodos} + 2$
- $V(G) = 4 - 4 + 2$
- $V(G) = 2$

CAMINOS INDEPENDIENTES

- 1,2,4
- 1,3,4

RESULTADOS DE LA PRUEBA

CAMINO	VALOR	RESULTADO
1	AS0125	No valida
2	ABC0123	Valida

Tabla 17: Resultados pruebas caja blanca validación placa vehículos.

7.1.5.2. PRUEBA CAJA NEGRA

ESCENARIOS

- Registro de vehículos.

DESCRIPCIÓN

- Se registra los datos de un vehículo.

CONDICIONES DE EJECUCIÓN

- Se debe iniciar correctamente la sesión en el sistema.

ENTRADA

- Seleccionar menú Vehículos
- Seleccionar la opción Registro Vehículo
- Ingresar los datos del vehículo:
 - Marca
 - Modelo
 - Año fabricación
 - Nro. chasis
 - Número de placa
 - Color
 - Km actual
 - Dependencia perteneciente

RESULTADO ESPERADO

Al registrar un vehículo se pretende obtener los siguientes resultados:

- Marca obligatorio
- Modelo obligatorio

- Año fabricación obligatorio
- Nro. chasis obligatorio
- Número de placa obligatorio
- Color obligatorio
- Km actual obligatorio
- Km actual acepta solo caracteres numéricos
- Dependencia perteneciente obligatorio

RESULTADO OBTENIDO

Tras el registro el sistema realiza las siguientes validaciones:

DATOS DE ENTRADA		DATOS DE SALIDA
PLACA VEHÍCULO	AS0125	Placa de vehículo no valida
	ABC0123	Placa de vehículo valida

Tabla 18: Resultados pruebas caja negra registro vehículos.

7.1.6. PRUEBAS MÓDULO CONTROL MEDICO

7.1.6.1. PRUEBA CAJA BLANCA

- No se efectúan pruebas de caja blanca.

7.1.6.2. PRUEBA CAJA NEGRA

ESCENARIOS

- Registro de consulta médica.

DESCRIPCIÓN

- Se registra los datos de una consulta médica de un empleado.

CONDICIONES DE EJECUCIÓN

- Se debe iniciar correctamente la sesión en el sistema.

ENTRADA

- Seleccionar menú Consultas Medicas
- Seleccionar la opción Registro Consultas
- Seleccionar empleado
- Ingresar datos de la consulta:
 - Motivo consulta
 - Síntomas
 - Diagnostico
- Ingresar datos prescripción médica:
 - Nombre medicamento
 - Cantidad
 - Indicación

RESULTADO ESPERADO

Al registrar una consulta médica se pretende obtener los siguientes resultados:

- Datos de la consulta:
 - Motivo consulta obligatorio
 - Síntomas
 - Diagnostico obligatorio
- Datos prescripción médica:
 - Nombre medicamento obligatorio
 - Cantidad:
 - Campo obligatorio
 - No menor o igual a 0 (cero)
 - Acepta solo dígitos numéricos.
 - Indicación obligatorio

RESULTADO OBTENIDO

Tras el registro el sistema realiza las siguientes validaciones:

DATOS DE ENTRADA		DATOS DE SALIDA
CANTIDAD	0	CANTIDAD no puede tener valores menores o iguales a 0 (cero).
	ABC	Cantidad debe ser un número entre -2147483648 y 2147483647.
	-5	CANTIDAD no puede tener valores menores o iguales a 0 (cero).
	5	CANTIDAD valida.

Tabla 19: Resultados pruebas caja negra registro vehículos.

CONCLUSIONES

Dentro de todo proyecto se hace necesario el análisis de varios factores que determinan el cumplimiento pleno de los objetivos, es así que se ha llegado al final logrando establecer varios factores que han sido de gran valor en el proceso de culminación.

Entre unos de los factores más importantes en el desarrollo de toda implementación es la colaboración por parte de la entidad beneficiaria y el investigador: el mismo que no se ha dado de manera óptima, lo cual ha dificultado llegar a culminar este proyecto en un tiempo de respuesta mayormente eficaz y eficiente, impidiendo revisiones periódicas y de gran valor en el proceso de implementación del sistema.

Lo expuesto anteriormente conlleva a que el investigador, tome por sí todas las decisiones en el transcurso del desarrollo, considerando únicamente su criterio, y tomando así el mayor riesgo en sus manos que es que el diseño y el análisis del proyecto sea únicamente desde su punto de vista y se crea correcto, pero al momento de usarse se considere como un fracaso: sin embargo mediante pruebas de funcionamiento por terceras personas se ha logrado moderar transiciones del sistema para que sea mayormente amigables con el usuario y cumpla con su finalidad.

Otro de los factores importantes, dentro de este proceso ha sido la investigación bibliográfica, que se ha realizado tanto en libros, tesis desarrolladas anteriormente con objetivos similares, pero mayormente dentro del mundo virtual, mediante descarga de tutoriales, ejemplos desarrollados, entre otros.

De esta manera se llegó a la conclusión de utilizar JSF y los componentes de Primefaces, que han permitido un manejo adecuado en cuanto a validaciones, introducción de datos, ahorro de código fuente, manejo de menús, pero sobre todo la interfaz que permite al usuario ser parte del sistema, así como también el uso de librerías como JFreechart, HSSFWorkbook e IReport, que han sido de mucha utilidad al momento de generar reportes y graficas.

RECOMENDACIONES

En el transcurso del desarrollo del presente proyecto de tesis, es necesario recomendar varias cosas que espero sean de valor para quienes decidan realizar proyectos similares.

- En cuanto al desarrollo de una aplicación web, estar siempre alerta a las actualizaciones de la misma, para cubrir errores e inseguridades.
- Realizar un análisis completo sobre las necesidades de la entidad beneficiaria y llegar a establecer un cronograma de revisión de la aplicación, que sea respetado por las dos partes, con el fin de evitar inconvenientes futuros.
- Adquirir uniformidad en el sistema, no solo en cuanto a la presentación del mismo sino también al momento de su implementación.
- Analizar varias opciones de software, para poder elegir, el que mejor se acople a las necesidades del sistema y sobre todo que este dentro de los alcances de la empresa.

ANEXOS

BIBLIOGRAFIA

Evolución De La Web (16 abril 2008), recuperado 19 enero 2012 desde <http://moduloweb.wikidot.com/evolucion-de-la-web>

Las generaciones de la Web (20 agosto 2011), recuperado 19 enero 2012 desde <http://www.juansinmiedo.com/las-generaciones-de-la-web/>

Psicología del color (5 junio 2012), recuperado 20 junio 2012 desde http://es.wikipedia.org/wiki/Psicolog%C3%ADa_del_color

Aplicación web (19 enero 2012), recuperado 19 enero 2012 desde http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web

Programación por capas (17 enero 2012), recuperado 19 enero 2012 desde http://es.wikipedia.org/wiki/Programaci%C3%B3n_por_capas

Lenguaje de programación interpretado (28 agosto 2011), recuperado 19 enero 2012 desde http://es.wikipedia.org/wiki/Lenguaje_interpretado

Interfaz de entrada común (5 noviembre 2011), recuperado 19 enero 2012 desde http://es.wikipedia.org/wiki/Interfaz_de_entrada_com%C3%BAn

Multipurpose Internet Mail Extensions (5 enero 2012), recuperado 20 enero 2012 desde <http://es.wikipedia.org/wiki/MIME>

Rich Internet Applications (14 diciembre 2011), recuperado 20 enero 2012 desde http://es.wikipedia.org/wiki/Rich_Internet_Applications

Diferencias Entre Aplicaciones Web Y Aplicaciones Desktop (o de Escritorio), recuperado 20 enero 2012 desde <http://www.buyto.es/general-diseno-web/diferencias-entre-aplicaciones-web-y-aplicaciones-desktop>

Dirección de Proyectos Planificación de los riesgos del proyecto (18 noviembre 2010), recuperado 20 enero 2012 desde http://es.wikibooks.org/wiki/Direcci%C3%B3n_de_Proyectos/Planificaci%C3%B3n_de_los_riesgos_del_proyecto#4.3.1_Identificaci.C3.B3n_de_riesgos

¿Qué son las bases de datos? (26 octubre 2007), recuperado 20 enero 2012 <http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/>

Servidor Web (19 enero 2012), recuperado 20 enero 2012 desde http://es.wikipedia.org/wiki/Servidor_web

Tomcat (15 enero 2012), recuperado 20 enero 2012 desde <http://es.wikipedia.org/wiki/Tomcat>

Web Applications, recuperado 20 enero 2012 desde http://webtrends.about.com/od/webapplications/a/web_application.htm

Java Persistence/ Print versión, recuperado 22 enero 2012 desde http://upload.wikimedia.org/wikipedia/commons/8/81/Java_Persistence.pdf

Annotations (2011), recuperado 22 enero 2012 desde <http://docs.oracle.com/javase/6/docs/technotes/guides/language/annotations.html>

Introducción a JPA(24 Enero 2010), recuperado 23 enero 2012 desde <http://www.davidmarco.es/blog/entrada.php?id=144>

Introduccion a JPA(01 mayo 2011), recuperado 28 enero 2012 desde <http://es.scribd.com/doc/46308101/Introduccion-a-JPA>

JPA Tutorial 1 - Getting Started , recuperado 28 enero 2012 desde <http://schuchert.wikispaces.com/JPA+Tutorial+1+-+Getting+Started>

Validar la cédula y el RUC de Ecuador en Java (27 febrero 2011) recuperado 10 marzo 2012 desde http://mygnet.net/codigos/java/varios/validar_la_cedula_y_el_ruc_de_ecuador_en_java.3335

Set Data Format in Excel Using POI 3.0 (14 marzo 2008), recuperado 20 noviembre 2012 desde <http://www.roseindia.net/java/poi/setDataFormat.shtml>

Java Persistence Query Language (JPQL), recuperado 15 marzo 2012 desde <http://www.aprendiendojava.com.ar/index.php?topic=59.0>

JSF Validators/Converters(11 octubre 2007), recuperado 16 marzo 2012 desde <http://johnderinger.wordpress.com/2007/10/11/validators-and-converters/>

Tema 4: lenguaje de consultas jpql for Desarrollo de Aplicaciones Web II, recuperado 20 marzo 2012 <http://es.scribd.com/doc/63832893/13/TEMA-4-LENGUAJE-DE-CONSULTAS-JPQL>

Persistencia EJB3 con JPA(), recuperado 20 marzo 2012 desde <http://es.scribd.com/doc/87160197/48/Expresiones-condicionales-y-Operadores>

Primefaces showcase, recuperado 15 abril 2012 desde <http://www.primefaces.org/showcase-labs/ui/home.jsf>

Faces Config Structure, recuperado 18 abril 2012 desde

<http://www.jetbrains.com/idea/webhelp/faces-config-structure.html>

web.xml Deployment Descriptor Elements, recuperado 18 abril 2012 desde
http://docs.oracle.com/cd/E13222_01/wls/docs81/webapp/web_xml.html

Save POI Excel file in JSF, recuperado 20 abril 2012 desde
<http://www.coderanch.com/t/416520/java/java/Save-POI-Excel-file-JSF>

S/A, *The Java EE 5 Tutorial For Sun Java System Application Server 9*, 2010.
<http://docs.oracle.com/javaee/5/tutorial/doc/javaetutorial5.pdf>

SCHMULLER, Joseph, *Aprendiendo UML en 24 horas*, Editorial Prentice Hall.

BURNS, Ed, y otros, *JavaServer Faces 2.0: The Complete Reference*, Editorial McGraw Hill, 2010.

GINESTÀ, Marc Gibert, *Bases de datos en PostgreSQL*.

FRIEDL, Jeffrey E. F., *Mastering Regular Expressions* 2da Ed, Editorial O'Reilly, Julio 2002.

DANCIU, Teodor, y otros, *The Definitive Guide to JasperReports*, Editorial Apress, 2007.

HEFFELFINGER, David R., *JasperReports for Java Developers*, Editorial Packt Publishing, 2006.

GÜLCÜ, Ceki, *The Complete log4j Manual*, Editorial draft, 2002.

ÇIVICI, Çağatay, *Primefaces user's guide 3.4*.
http://primefaces.googlecode.com/files/primefaces_users_guide_3_4.pdf

GILBERT, David, *The JFreeChart Class Library* Version 1.0.4, 2007.