

Universidad Central “Marta Abreu” de Las Villas
Facultad de Matemática, Física y Computación
Licenciatura en Ciencia de la Computación



Trabajo de Diploma

Título:

Sistema para el control de servicios de alquiler
de autos estatales.

Autor: Osniel Torres Garrido

Tutor: Lic. Frank Reyes García

Santa Clara, 2016

DICTAMEN

Osniel Torres Garrido, hago constar que el trabajo titulado “Sistema para el control de servicios de alquiler de autos estatales” fue realizado en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de los estudios de la especialidad de Licenciatura en Ciencia de la Computación, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

Firma del autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del tutor

Firma del jefe del Laboratorio

Fecha

PENSAMIENTO

“Nuestra recompensa se encuentra en el esfuerzo y no en el resultado. Un esfuerzo total es una victoria completa.”

Mahatma Gandhi

DEDICATORIA

A mi abuela, impulsora incansable y por quererme tanto.

A mi novia Claudia por apoyarme en cada momento.

A mis suegros, y a todas mis amistades por su apoyo.

AGRADECIMIENTOS

Agradezco sinceramente:

Al Dr. Daniel Gálvez Lio por su ayuda durante estos cinco años.

A mi tutor Lic. Frank Reyes García por hacerse cargo de mí y ser ejemplo profesional a seguir.

A toda mi familia por su ánimo y confianza durante mi formación profesional.

A mis compañeros de estudio y mis hermanos de universidad en especial: Leonardo, Rafael, Radner, Luis Daniel, Armando y Yanir.

A los profesores que se empeñaron en hacer de mí y de mis compañeros buenos profesionales.

RESUMEN

RESUMEN

En la actualidad en la Empresa de CubaTaxi de la provincia de Villa Clara los departamentos de Puesto de Mando y Operaciones realizan el almacenamiento de la información referente a los servicios prestados y a su personal de forma manual, lo cual trae consigo inconsistencia, pérdida y errores humanos en los datos que en ella se manejan. En aras de solucionar la situación planteada el presente trabajo de diploma tiene como objetivo el desarrollo de un sistema de información para la gestión de estas actividades en ambos departamentos. Para ello se incluyen las etapas de análisis de requerimientos, diseño, implementación y prueba del sistema informático.

Como resultado se obtiene un sistema de información que permite manejar y recuperar la información referente a ambos departamentos, a partir del diseño de una base de datos orientada a los requerimientos, ofreciendo información oportuna y la confección de informes libre de errores humanos.

ABSTRACT

Today at the headquarters of the company CubaTaxi province of Villa Clara departments Command Post Operations and perform the storage of information relating to the services provided and their staff manually, which brings inconsistency, loss and human errors in the data handled therein. In order to solve the situation in question this dissertation describes the development of an information system for managing these activities in both departments. For this stage of requirements analysis, design, implementation and testing of the system it is included.

As a result, an information system that allows managing and retrieving information concerning both departments, from the design of a database -oriented requirements, providing timely information and making free human error reports is obtained.

ÍNDICE

Tabla de Contenido

INTRODUCCIÓN	1
CAPÍTULO 1. TECNOLOGÍAS Y HERRAMIENTAS DE DESARROLLO.	6
1.1 Análisis de la problemática	6
1.2 Principales conceptos y tipos de Sistemas de Información	9
1.2.1 Sistemas de información operacionales	11
1.2.2 Sistemas de apoyo a la toma de decisiones	12
1.2.3 Almacén de datos y mercado de datos	13
1.2.4 Ciclo de vida de los SI	14
1.3 Lenguaje de Programación Java.....	14
1.3.1 Principales características del lenguaje.....	15
1.3.2 Programación orientada a objetos.....	17
1.3.3 Facilidad de uso y trabajo en red	18
1.4 NetBeans	19
1.4.1 Java.sql. *	20
1.4.2 Javax.swing.*	21
1.4.3 PostgreSQL-8.3-603.jdbc3.jar.....	22
1.5 Sistema Gestor de Base de Datos PostgreSQL	22
1.5.1 Características, limitaciones y ventajas.....	22
1.5.2 Características Generales	22
1.6 JasperReports	23
1.6.1 Funcionamiento de JasperReports.....	23
1.6.2 IReport.....	24
1.7 Conclusiones Parciales	24
CAPÍTULO 2. ANÁLISIS Y DISEÑO DEL SISTEMA “CAAE” CONTROL DE AUTOS DE ALQUILER ESTATALES.....	26

2.1	Descripción del Sistema	26
2.2	Gestión de Requerimientos	26
2.2.1	Requerimientos no funcionales	26
2.2.2	Requerimientos funcionales.....	27
2.3	Análisis y diseño	27
2.3.1	Diagrama de Caso de Uso.....	28
2.3.2	Esquema conceptual de la Base de Datos.....	35
2.3.3	Diagrama de actividad.....	40
2.3.4	Diagrama de componentes	42
2.3.5	Diagrama de Clases	43
2.3.6	Diagrama de Despliegue.....	45
2.3.7	Patrón de arquitectónico Modelo vista Controlador (MVC)	46
2.3.8	Herramientas para el desarrollo	47
2.4	Conclusiones parciales	48
CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA DEL SISTEMA “CAAE”.		50
3.1	Sistema de Pruebas.....	50
3.2	Resultados de las Pruebas	51
3.3	Requerimientos de Instalación	56
3.3.1	Instalación de la BD	57
3.3.2	Requerimientos e Instalación del “CAAE.jar”	57
3.4	Descripción del sistema.....	57
3.4.1	Pasos iniciales.....	57
3.4.2	Descripción del comando pg_dump.....	63
3.5	Conclusiones Parciales.....	66
CONCLUSIONES		67
RECOMENDACIONES		68
REFERENCIAS BIBLIOGRÁFICAS		69

Lista de Figuras

Figura 1.1: Flujo de Información.....	6
Figura 1.2: Relación de los SI con los niveles organizacionales de una empresa.....	11
Figura 1.3: Panorama de un sistema de apoyo a la toma de decisiones.	12
Figura 1.4 Proceso de desarrollo de sistemas.....	14
Figura 1.5: Proceso de Interpretación.....	16
Figura 1.6: Máquina Virtual de Java	17
Figura 2.7: Diagrama de Caso de Uso de (“usuario PM” y “Técnico PM”).....	28
Figura 2.8 : Diagrama de Caso de Uso "Administrador"	29
Figura 2.9: Diagrama de Caso de Uso “Subdirector”.....	30
Figura 2.10:Diagrama Entidad-Relación.....	36
Figura 2.11:Diagrama de Actividad para el Caso de Uso “Agregar Chofer” (extensión del Case de Uso “Gestionar Chofer”).....	41
Figura 2.12: Diagrama de Actividad para el Caso de Uso "Autenticación".....	42
Figura 2.13: Diagrama de Componentes.	43
Figura 2.14: Diagrama de Clases.....	44
Figura 2.15: Diagrama de Despliegue de la aplicación.	46
Figura 2.16: Modelo Vista Controlador.	47
Figura 17: Amigable.....	52
Figura 18:Legible.	52
Figura 19: Eficaz.	53
Figura 20: Eficiente.	53
Figura 21: Satisfacción.....	54
Figura 22: Reversible.	54
Figura 23: Autonomía.....	55
Figura 24: Interfaz Gráfica.	55
Figura 25: Mensaje de Error.....	56

Figura 26: Ventana de Autenticación.....	58
Figura 27: Vista del “Administrador”.	58
Figura 28: Vista de Registro de Usuarios.....	59
Figura 29: Vista del “Subdirector”.	60
Figura 30: Vista de Creación de Reportes.	61
Figura 31: Vista de Creación de Resguardo.	62
Figura 32: Método Resguardo.	62
Figura 33: Vista del “Técnico de Puesto de Mando ”.	64
Figura 34: Vista del "Usuario de puesto de Mando".	65
Figura 35: Resumen del Turno.	65

Lista de Tablas

Tabla 1.1: Framerwok, tecnología, lenguajes de programación y plugin integrados a Netbeans.	19
Tabla 2.2: Caso de Uso “Gestionar Vehículo”.	32
Tabla 2.3: Caso de Uso “Gestionar Reportes”.	35
Tabla 2.4: Herramientas para el desarrollo.....	48

INTRODUCCIÓN

INTRODUCCIÓN

La Computación es una de las ramas de la ciencia con mayor velocidad de desarrollo, y cada vez se introduce más en todas las esferas de la vida del hombre, desde los procesos científicos, tecnológicos e industriales hasta las tareas domésticas. Con el transcurso del tiempo, la gestión de datos se convierte en un asunto medular. El hombre, con el nacimiento de la ciencia computacional, como actor y beneficiario principal de este proceso, desempeña un rol activo en el perfeccionamiento de métodos y herramientas que le permiten automatizar la manera de procesar la información de forma transparente y lograr su almacenamiento de forma segura.

Los sistemas de información (SI) se han convertido en el soporte que garantiza el correcto funcionamiento de la gestión corporativa. Permite que las organizaciones se inserten en la competencia, mejoren su productividad y tomen decisiones acertadas (Prieto & Martínez 2004).

En la actualidad, un gran número de empresas invierten grandes sumas de dinero en hardware, software y equipos de telecomunicaciones para los SI, además de consultoría y servicios de negocios y administrativos. Entre 1980 y 2004 las inversiones de las empresas privadas en tecnología de información crecieron de 34 a 50 por ciento del capital total invertido (Laudon 2004).

La utilización de esta tecnología aún resulta incipiente, especialmente en los países subdesarrollados y en vías de desarrollo. Aunque Cuba se incluye en este último grupo de naciones, ya se inserta en el empleo de herramientas de informatización.

A pesar de ello, todavía existen en la nación instituciones y organismos que se mantienen fieles a métodos tradicionales de almacenamiento de información. Incluso, algunos que manejan grandes cantidades de datos y documentos aún no se afilian a los mecanismos tecnológicos.

CubaTaxi es la empresa encargada de dar servicio de transportación a diferentes usuarios. En la provincia de Villa Clara existe una representación de dicha entidad y en cada uno de los municipios siendo la entidad principal la que radica en el municipio de Santa Clara

tanto por el número de vehículos y de choferes como por el número de servicios que brinda.

El subdirector de operaciones es el responsable del control de todas las actividades que realiza la entidad e interactúa para ello con el puesto de mando, el departamento económico, el departamento de facturación, el de recursos humanos y el departamento de mantenimiento. Esta interacción debe garantizar la integridad de la información manejada en esta entidad que incluye elementos como el consumo de combustible, el estado técnico de los vehículos, la salud, estado de licencia y disponibilidad de los choferes, los servicios que presta un determinado chofer ya sea piquera, contratos o recogidas además de las recaudaciones y liquidaciones de los mismos.

La empresa cuenta con 15 modelos para el manejo y control de la información los cuales se denominan CAAE – XX donde XX representa el número que lo identifica según su funcionalidad. Cada modelo tiene establecido en un documento la metodología de uso y llenado de los mismos, según las normas y procedimientos establecidos por la empresa. El modelo primario de información es la Tarjeta de Circulación (CAAE-3) que refleja todas las incidencias del auto en operaciones la cual es llenada por los choferes y registrada por el operativo en el puesto de mando.

Toda esta información incluyendo el llenado de los modelos y los cálculos que ellos requieren es realizada de forma manual por diferentes especialistas en los departamentos correspondientes. Dando lugar a que exista inconsistencia en la información entre los diferentes departamentos ya sea por errores humanos, extravío o llegada tardía de documentos, etc. Estas inconsistencias deben ser resueltas a partir de una conciliación periódica entre los diferentes departamentos que garantice una correcta información.

Por todo lo anterior se formula el siguiente **problema de investigación**: La representación de entidad de CubaTaxi en la provincia de Villa Clara no cuenta con un sistema computacional que permita el llenado de los modelos y los cálculos que en estos se realizan. Además, no se garantiza el control, la integridad y el almacenamiento de toda la información de la empresa de forma histórica, para poder emitir informes sobre el trabajo de la empresa en distintos años.

Objetivo General

Desarrollar un sistema computacional para el control de alquiler de autos estatales en la entidad CubaTaxi del municipio de Santa Clara que garantice el flujo, almacenamiento e integridad de la información manejada por la entidad.

Para lograr dicho objetivo general, se proponen los **objetivos específicos**:

- 1- Obtener el modelo de negocio de la entidad para comprender su funcionamiento y el flujo de información entre los distintos departamentos.
- 2- Diseñar la base de datos que garantice el manejo y almacenamiento de la información en la entidad.
- 3- Implementar un software que automatice el modelo de negocio de la entidad para los departamentos de Operaciones y Puesto de Mando.
- 4- Implantar el software en la entidad y verificar la satisfacción del usuario del sistema.

Justificación

La entidad no cuenta con un sistema capaz de brindar servicio a los diferentes departamentos de forma rápida, concisa y segura. Para que los mismos puedan gestionar las diferentes informaciones que en ella se almacena sin temor a errores humanos o inconsistencia entre los departamentos. Además, la entidad cuenta con los medios básicos necesarios para llevar a cabo este proyecto.

Esta tesis se encuentra estructurada en tres capítulos:

En el capítulo 1: Se resumen los principales elementos de carácter teórico-práctico relacionados con el estudio de tecnologías y herramientas que permiten dar solución efectiva al problema planteado. También se realiza una descripción del problema caracterizando el flujo de información entre los departamentos.

En el capítulo 2: Se tratan elementos esenciales sobre el análisis y diseño del sistema, se analizan los requerimientos detectados y a partir de estos se definen los casos de uso vinculados a los mismos. En la etapa de programación se diseñan los UML necesarios, además se propone una solución al resguardo de los datos almacenados.

En el capítulo 3: Se expone aspectos fundamentales relacionados con la implementación del sistema y se explican algunas de las características de la interfaz visual para facilitar el entendimiento por el usuario. También se abordan los requisitos necesarios para la instalación del sistema. Además, se explican que son los sistemas de pruebas y explicamos las pruebas aplicadas al sistema “CAAE”

Todos los capítulos terminan con un epígrafe de consideraciones finales en el que se resumen los aspectos más importantes que se trataron. Finalmente se enuncian las conclusiones, recomendaciones, y se relaciona la bibliografía.

*TECNOLOGÍAS Y
HERRAMIENTAS DE
DESARROLLO*

CAPÍTULO 1. TECNOLOGÍAS Y HERRAMIENTAS DE DESARROLLO.

En el presente capítulo se analizan los datos necesarios para el desarrollo de un sistema de información para la empresa de CubaTaxi de la provincia de Vila Clara. Además, se muestran los conceptos fundamentales sobre Sistemas de Información; así como los elementos indispensables para garantizar el funcionamiento como instrumento de una entidad determinada. Además, se analizan las características de las tecnologías a utilizar como el entorno de desarrollo integrado NetBeans y el gestor de base de datos PostgreSQL.

1.1 Análisis de la problemática

La empresa de CubaTaxi es una de las empresas encargadas de prestar servicios de transportación a diferentes tipos de clientes. Entre los servicios que se brindan en esta entidad destacan algunos como son: la recogida de pasajeros, el trabajo en las distintas piqueras existentes en la provincia y los contratos realizados con otras empresas. La Figura 1.1: Flujo de Información. describe de forma muy breve el flujo de información entre los departamentos de Operaciones y el de Puesto de Mando. Dicho flujo de información se puede separar en tres procesos.



Figura 1.1: Flujo de Información.

Capítulo 1 Tecnologías y herramientas de desarrollo

Como se puede observar en la figura anterior la información llega al usuario de puesto de mando a través de los distintos choferes los mismos proporcionan toda la información necesaria mediante la tarjeta de circulación más conocida como CAAE-3 donde quedan plasmados datos como:

- Folio
- Número del Vehículo
- Chapa
- Marca
- Km Disponibles
- Nombre del Chofer
- Licencia
- Fecha de Salida
- Fecha de Entrada
- Odómetro Inicial
- Odómetro Final
- KM Totales
- KM Pasajeros
- Cantidad de Combustible
- Número de la Tarjeta de Combustible
- Cantidad de Viajes
- Total, de Pasajeros
- Hora de Salida
- Hora de Entrada
- Tipo de Servicio
- Cuanto Recaudo

El (usuario PM) almacena toda la información durante su turno de trabajo, y una vez terminado su turno procede a la realización de un resumen del mismo el cual es entregado al departamento de Economía con los siguientes datos:

- Folio
- Número del Vehículo
- Nombre del Chofer

Capítulo 1 Tecnologías y herramientas de desarrollo

- Recaudado
- Subtotal
- Total, General

También el (usuario PM) deberá almacenar las incidencias ocurridas durante su turno de trabajo. El Técnico es el encargado de modificar algún error cometido a la hora de almacenar alguna información y de todos los datos de los (usuario PM). Mientras que en el departamento de Operaciones se almacena toda la información y es donde se realizan todas las operaciones necesarias para los balances de forma periódica. Se almacenan y modifican además todos los datos de los vehículos tale como:

- Número del vehículo
- Chapa
- Categoría
- Cantidad de Asientos
- FICAV
- LOT
- KM Disponibles
- Norma de Consumo
- Tipo de Combustible
- Estado del Vehículo

También en el departamento se almacenan y modifican los datos de los choferes dentro de los cuales se encuentran:

- Nombre y Apellidos
- Carnet de Identidad
- Licencia
- Fecha de vencimiento de la licencia
- Categorías de la licencia
- Psicológico
- Chequeo medico
- Recalificación
- Estado del Chofer

Como se muestra anteriormente queda almacenada parte de la información en el departamento de operaciones haciendo énfasis en los datos más relevantes. Dentro de los controles periódicos que se realizan en la empresa se conforman una serie de reportes que al igual que la información anterior queda registrada en el departamento.

1.2 Principales conceptos y tipos de Sistemas de Información

La información es un conjunto organizado de datos procesados, que constituyen un mensaje que cambia el estado de conocimiento del sujeto o sistema que recibe dicho mensaje. Según lo expresado en (Trasobares 2003) los datos representan realidades concretas sobre hechos ocurridos en las organizaciones o en el entorno físico, aisladamente los datos no poseen significado alguno. Solo se puede apreciar la información contenida en dichos datos cuando un conjunto de ellos se estructura, organiza o examinan conjuntamente a la luz de un enfoque, hipótesis o teoría.

Los SI constituyen una herramienta computacional para almacenar y gestionar datos que satisfacen una necesidad de información. Contienen un grupo de elementos que mantienen relación entre sí (Laudon 2004). Los mismos, “recolectan (o recuperan), procesan, almacenan y distribuyen información”, de acuerdo con el propio autor.

Los SI son de gran utilidad pueden ayudar a los administradores y el personal a elevar el nivel de conocimiento, analizar problemas, visualizar asuntos complejos y crear nuevos productos y viabilizar el apoyo de la toma de decisiones y el desarrollo de acciones; mejoran la efectividad gerencial y profesional, formando parte de la estrategia de un negocio. Además, deben aportar a la directiva la claridad necesaria para ver una realidad compleja: el hombre, la empresa y el entorno. Permitiendo el aumento de la productividad y le facilita a la directiva a fijar estrategias para incrementar la eficiencia, la efectividad, la rentabilidad y el aprovechamiento de sus recursos, los que conlleva a un crecimiento sostenido de la productividad en todos los niveles (Prieto & Martínez 2004)

El hardware, la computadora y el recurso humano como personal operativo para la explotación de las funcionalidades del mismo resucita el recurso material como parte integrante del SI.

Los SI se clasifican de acuerdo a su estructura y funcionamiento. Existen la presencia de dos tipos de clasificación de SI, según(Montilva 1999):

Capítulo 1 Tecnologías y herramientas de desarrollo

Formal: “en un conjunto de normas, estándares y procedimientos que permiten que la información se genere y llegue a quién la necesite en el momento deseado”.

Informal: “en la comunicación no formalizada, ni predefinida entre las personas de la organización”.

Cuando el hombre, auxiliado por cierto equipo (máquinas de escribir, calculadoras, archivos, etc.), realiza las principales funciones de recopilación, registro, almacenamiento, cálculo y generación de información, se entienden como manuales, en correspondencia con el elemento principal de proceso de la información conforme al criterio de (Peña Ayala 2006) estos SI se ubican en un segundo nivel de clasificación. Cuando las principales funciones del procesamiento la ejecutan una maquinaria, contrariamente, se les llama mecanizados.

En procesos sencillos, que manejan pequeños volúmenes de datos, sin efectuar cálculos complejos y en los que mantener actualizada la información no resulta problemática, los sistemas manuales son adecuados. Los que requieren manipular altos volúmenes de datos en un tiempo corto de respuesta son los llamados mecanizados.

Se concibe diferentes tipos SI ya que un solo tipo SI no puede proveer a la organización de toda la información que necesita debido a la existencia de diferentes intereses, especialidades y niveles de organización.

De acuerdo con (Karen & Lares 2000), para unificar las funciones comunes los SI se distribuyen en tres niveles de jerarquía organizativa: el transaccional u operacional, el de apoyo a las decisiones y el estratégico

El transaccional u operacional: se dirige a los procesos operacionales para automatizarlos como actividad de un negocio. Comúnmente se seleccionan como los primeros a implantar en una organización, pues se conforman como la base de la información productiva utilizada posteriormente.

Sistema de apoyo a las decisiones: constituyen la plataforma de información e intentan integrar los sistemas transaccionales más relevantes de la empresa para ser utilizada por los mandos intermedios y la alta administración (toma de decisiones).

Los sistemas estratégicos se emplean con el propósito de buscar superioridad competitiva, o bien para reducir la ventaja de los competidores. Según (Karen & Lares 2000), los desarrolla el personal de la organización.

Al clasificar SI en nivel operativo, nivel gerencial y nivel estratégico en la Figura 1.2: Relación de los SI con los niveles organizacionales de una empresa se muestra la tipología expuesta por (Laudon et al. 1996) similar a la planteada por (Karen & Lares 2000)



Figura 1.2: Relación de los SI con los niveles organizacionales de una empresa.

1.2.1 Sistemas de información operacionales

Son recolectores de gran cantidad de información para realizar sus operaciones, que generan grandes sumas de datos, ahorran en gran medida el trabajo manual, y sus cálculos y procesos suelen ser simples y poco sofisticados. Se conocen También como Sistema de Procesamiento de Transacciones (TPS, por sus siglas en inglés), De acuerdo con (Laudon 2004). La condición de una empresa requiere un sistema computarizado que ejecuta y registra las transacciones ordinarias, cotidianas de la misma.

Responder las preguntas rutinarias y dar seguimiento al flujo de transacciones en la organización es el objetivo primordial de estos sistemas. Necesitan los gerentes de estos sistemas para supervisar el estado de las operaciones internas y las relaciones de la empresa con el entorno externo además de ser productores importantes de información para los demás tipos de sistemas.

1.2.2 Sistemas de apoyo a la toma de decisiones

Para decidir qué rumbo tomar. Los sistemas de apoyo a la toma de decisiones o (DSS por sus siglas en inglés) son los que permiten realizar análisis de diferentes cuestiones de una organización. Para apoyar a la gerencia intermedia en la toma de decisiones estos se enfocan en extraer grandes cantidades de datos.

Suelen establecerse a partir de sistemas operacionales previamente establecidos en la empresa debido a que estos últimos constituyen las fuentes de datos. Además de utilizar otras fuentes externas. Son interactivos y amigables gracias a su completo diseño gráfico y visual, utilizando diversos modelos para el análisis de datos.

Ofrecen acceso a bases de datos y herramientas de análisis permitiendo la creación de reportes, pronósticos, gráficos y pueden apoyar el intercambio de información dentro de la organización (Laudon 2004).

Algunos DSS son orientados a datos y utilizan procesamiento analítico en línea (OLAP) y minería de datos para analizar tendencias y patrones de comportamiento en grandes concentraciones de datos.

Sus componentes son las bases de datos utilizadas para consultas y análisis, un sistema de software con modelos, minería de datos y otras herramientas analíticas, así como una interfaz de usuario, como se observa en la Figura 1.3: Panorama de un sistema de apoyo a la toma de decisiones.



Figura 1.3: Panorama de un sistema de apoyo a la toma de decisiones.

Según (Laudon 2004) la base de datos de un DSS es un conjunto de datos históricos y actuales de varias aplicaciones que pueden combinarse con datos externos. La base de datos del DSS podría ser un gran almacén de datos que continuamente se actualice por los sistemas fuentes operacionales que son los encargados de capturar las transacciones. Debido al gran avance en las diferentes tecnologías en la actualidad existen DSS que cuentan con una interfaz Web así aprovechando el despliegue de gráfico, la interactividad y la facilidad de su uso.

1.2.3 Almacén de datos y mercado de datos

La tecnología de los almacenes de datos (DW por sus siglas en inglés) basa sus conceptos y diferencias entre los dos tipos de SI mencionados anteriormente: los sistemas técnico-operacionales y los sistemas de soporte de decisiones.

Por lo cual un DW es una colección de datos transformados y separados físicamente de la fuente operacional que se usa para el soporte del proceso de toma de decisiones. Su objetivo principal de esta tecnología es proporcionar una plataforma sólida, a partir de los datos que se guardan en un amplio período de tiempo, para hacer análisis respecto a determinados indicadores de una empresa.

De acuerdo con (Inmon 2005) un DW se define como un repositorio de datos con las siguientes propiedades:

- Orientado a temas: Los datos en la base de datos están organizados de manera que todos los elementos de datos relativos al mismo evento u objeto del mundo real queden unidos entre sí.
- Variante en el tiempo: Los cambios producidos en los datos a lo largo del tiempo quedan registrados para que los informes que se puedan generar reflejen esas variaciones.
- No volátil: La información no se modifica ni se elimina, una vez almacenado un dato, éste se convierte en información de sólo lectura y se mantiene para futuras consultas.
- Integrado: La base de datos contiene los datos de todos los sistemas operacionales de la organización y dichos datos deben ser consistentes.

El DW es un sistema que extrae, limpia, ajusta, y proporciona datos de un origen a un almacén de datos dimensional y luego apoya e implementa la consulta y análisis con el propósito de tomar decisiones (Ross & Kimball 2013). Se puede segmentar en varios mercados de datos que son subconjuntos del DW para cierto grupo de usuarios o funciones del negocio, se construye para consultas más rápidas y menos usuarios.

El diseño de un DW se estructura mediante un modelo multidimensional, representado a través de un esquema relacional que se basa en los conceptos de dimensión y medida.

1.2.4 Ciclo de vida de los SI

Los SI requieren para su desarrollo la integración de varios factores, a la vez que se construyen sobre una secuencia de pasos que fortalecen su diseño. Cuando la organización decide desarrollar un sistema reviste gran importancia la concepción inicial del mismo, por ello es preciso consultar al capital humano que posee el conocimiento de sus actividades, de los recursos disponibles y las necesidades de información.

Las opiniones de (Laudon et al. 1996) y (Peña Ayala 2006) coinciden en cuanto al proceso de desarrollo de los SI como muestra la (Figura 1.4 Proceso de desarrollo de sistemas.



Figura 1.4 Proceso de desarrollo de sistemas.

1.3 Lenguaje de Programación Java

Java es un lenguaje de desarrollo de propósito general, y como tal es válido para realizar todo tipo de aplicaciones profesionales desarrollado por Sun Microsystems el cual fue presentado en la segunda mitad del año 1995 y desde entonces se ha convertido en un lenguaje de programación muy popular.

Este lenguaje se creó con cinco objetivos principales:

1. Debería usar el paradigma de la programación orientada a objetos.
2. Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.
3. Debería incluir soporte para trabajo en red de forma predeterminada.
4. Debería diseñarse para ejecutar código en sistemas remotos de forma segura.
5. Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos.

1.3.1 Principales características del lenguaje.

Entre las características que enriquecen y forman el lenguaje se encuentran algunas como:
Es intrínsecamente orientado a objetos. Funciona perfectamente en red, aprovecha

Capítulo 1 Tecnologías y herramientas de desarrollo

características de los lenguajes modernos evitando sus inconvenientes; en particular los del C++. Tiene una gran funcionalidad debido a sus librerías (clases), no tiene punteros manejables por el programador, aunque los maneja de forma interna. El manejo de la memoria la gestiona el propio lenguaje y no el programador. Incorpora Multi-Threading (para permitir la ejecución de tareas concurrentes dentro de un mismo programa).

Además, una de las características más importantes de los programas “ejecutables”, creados por el compilador de Java, es que son independientes de la arquitectura. Se ejecutan indistintamente en una gran variedad de equipos con diferentes microprocesadores y sistemas operativos. (Groussard 2012). El lenguaje Java es interpretado, necesita un proceso de compilación y luego de terminado este proceso se crea un fichero que almacena lo que es denominado bytecodes (pseudocódigo prácticamente al nivel de código máquina). Para la ejecución del programa se necesita de un intérprete la JVM (Java Virtual Machine) o máquina virtual Java; quien es el encargado de leer los bytecodes y traducirlos a instrucciones ejecutables directamente en un microprocesador, de una forma eficiente.

Que el programa deba ser “interpretado” no hace que sea poco eficiente en cuanto a velocidad, ya que la interpretación se hace prácticamente al nivel de código máquina. Debido a compiladores como Just-In-Time (JIT) que son capaces de transformar los bytecode de un programa en código nativo de la plataforma donde se ejecute. (Gosling & Arnold 2001)

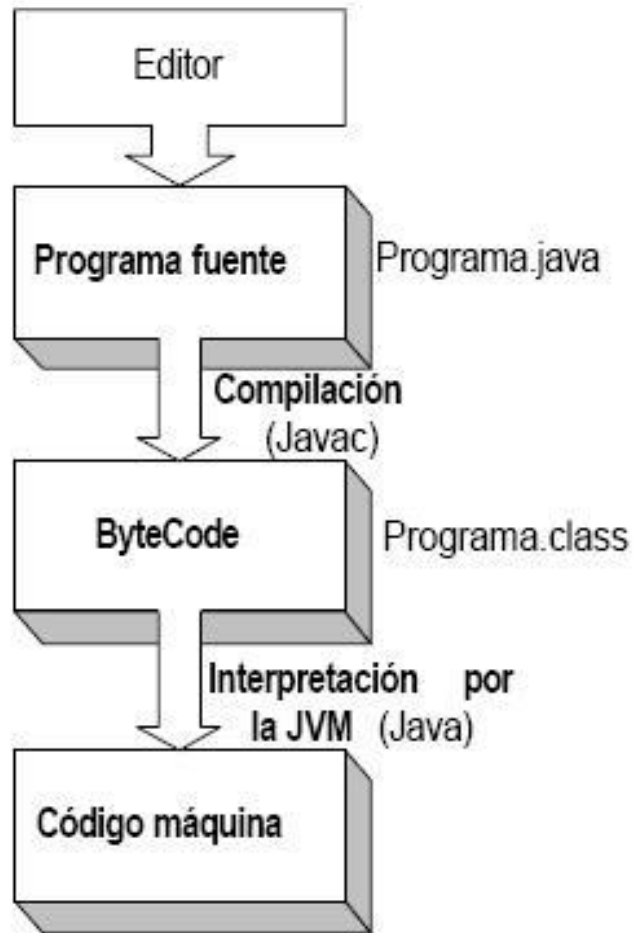


Figura 1.5: Proceso de Interpretación

La máquina virtual es la entidad que proporciona la independencia de la plataforma de cualquier programa en java que sea compilado en bytecode.

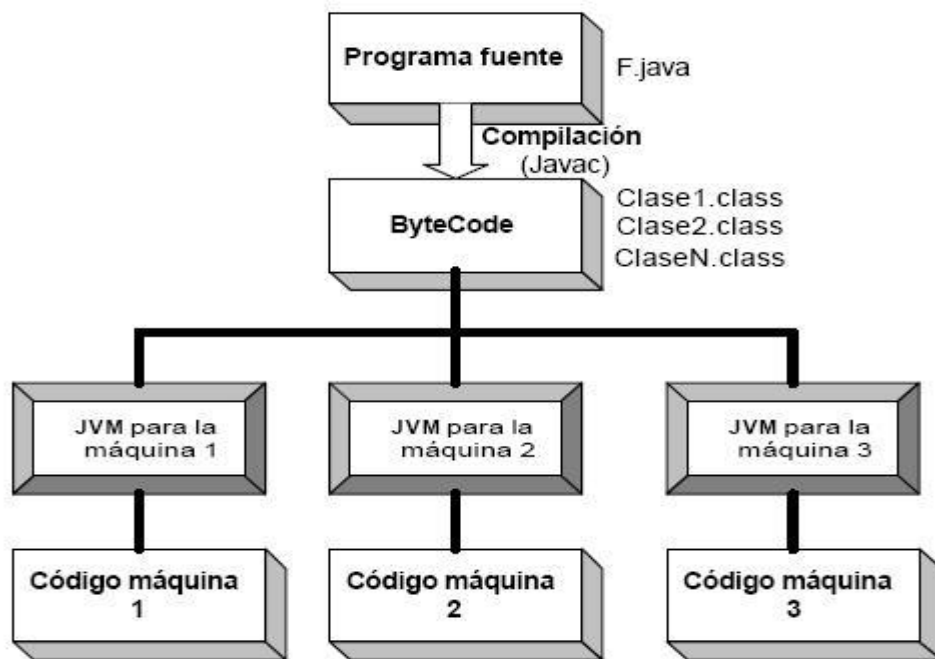


Figura 1.6: Máquina Virtual de Java

Genera el mismo fichero bytecode un programa que sea compilado en distintas plataforma o sistemas operativos. Lo que significa que existe un traductor de ese bytecode a código nativo de la máquina sobre la que se ejecuta. De dicha tarea se encarga la JVM de la cual existen diferentes versiones para cada plataforma.

1.3.2 Programación orientada a objetos

En la programación orientada a objetos, se definen objetos que conforman una aplicación. Estos objetos están formados por una serie de características y operaciones que se pueden realizar sobre los mismos. Estos objetos no están aislados en la aplicación, sino que se comunican entre ellos.

Cada objeto puede definirse en función de multitud de características y se pueden realizar innumerables operaciones sobre él. Ya en términos de programación, será misión del programador determinar qué características y que operaciones interesa mantener sobre un objeto. En terminología de programación orientada a objetos (POO), a las características del objeto se les denomina ATRIBUTOS y a las operaciones MÉTODOS. Cada uno de estos métodos es un procedimiento o una función perteneciente a un objeto. Un objeto está formado por una serie de características o datos (atributos) y una serie de operaciones (métodos). No puede concebirse únicamente en función de los datos o de las operaciones sino en su conjunto. (Narváez Coello 2014)

1.3.3 Facilidad de uso y trabajo en red

El diseño, su fácil portabilidad y su robustez han hecho de Java un lenguaje con mayor crecimiento y amplitud en la industria Informática. En la actualidad existen muchas aplicaciones gráficas de usuario basadas en Java. El entorno de ejecución Java (JRE) se ha convertido en un componente habitual en las computadoras de usuario, de los diferentes sistemas operativos más usados en el mundo; muchas aplicaciones Java lo incluyen dentro del propio paquete de la aplicación de modo que se ejecuten en cualquier computadora (Deitel 2004).

En interfaces de programación de aplicaciones (API) de desarrollo gráfico (AWT) existían limitaciones en las en las primeras versiones de la plataforma. Desde la aparición de la biblioteca Swing, la situación mejoró substancialmente y posteriormente con las bibliotecas como SWT hacen que el desarrollo de aplicaciones de escritorio complejas y con gran dinamismo, usabilidad, etc. sea relativamente sencillo.

Seguridad (Sánchez 2003)

En java debido a que el código es interpretado por la JVM se puede delimitar las operaciones peligrosas, por lo cual la seguridad es fácil mente controlable. También se eliminan los punteros que dan la posibilidad de crear programas para atacar sistemas y las instrucciones dependientes de la máquina.

La primera línea de seguridad de java es un verificador de bytecode que permite comprobar que el comportamiento del código es correcto y que sigue las reglas de java. Por lo generar los compiladores de java no pueden generar código que se salte las reglas de java, aunque un programador malicioso podría generar bytecode que las viole el verificador tratara de eliminarlas.

Además, existe un segundo paso que verifica la seguridad del código que es el verificador de clases que no es más que el programa que proporciona las clases necesarias del código. Este verificador de clases se encarga de que las clases que se cargan son realmente las del sistema original y no artificiales.

También se cuenta con un programa configurables que permite indicar a los usuarios niveles de seguridad a sus programas en java el mismo se conoce como administrador de seguridad.

1.4 NetBeans

En (Mendoza González 2008) se describe al IDE como una herramienta que se utiliza para desarrollar aplicaciones Web, Móvil y de Escritorio para diferentes lenguajes de programación como son Java, C++, Ruby y PHP entre otros. Es de código abierto, es multiplataforma, multilenguaje, contiene servidores web y es fácil de instalarlo e utilizarlo.

Algunos de los framerwok, tecnología, lenguajes de programación y plugin que están integrados en la herramienta de Netbeans (Mendoza González 2008) son los siguientes:

Lenguaje de Programación	Tecnología	Framework
PHP	PHP 5.4	Zend Framework Symfony 1 y 2 Framework Nette Framework 2
JAVA	JAVA SE JAVA ME 2 JAVA EE 7,6, 6 Y 1.4 JAVA FX 2.2	Swing JSF, Struts 1.3 EJB, Spring 3.2 JPA, Hibernate 3.6 Web RESTFull(JAX-RS), Maven
C/C++	C/C++	
HTML 5	HTML 5	

Tabla 1.1: Framerwok, tecnología, lenguajes de programación y plugin integrados a Netbeans.

Además, incluye Servidores de aplicaciones:

- GlassFish Server Open Source 4.x

- Apache Tomcat 7.x
- JBoss AS 7.x
- WebLogic 11g

También incluye plugin como:

- Inreport
- Sublime Theme
- Ruby on Rails
- JavaScript

Esta plataforma ofrece diferentes servicios para aplicaciones de escritorio entre las que se encuentran:

1. Gestionar las interfaces de usuario con menús y barras de herramientas.
2. Gestionar las configuraciones del usuario.
3. Administración del almacenamiento.
4. Administración de ventanas.

Entre las principales bibliotecas para el trabajo basadas en Java útiles para dar solución al problema planteado se encuentran:

1.4.1 Java.sql. *

Es una biblioteca importada por Java para hacer consultas en BD, si se usa únicamente esta biblioteca, entonces podemos realizar una conexión básica y simple, sirve para acceder a bases de datos SQL mediante una llamada a un método de la biblioteca Java.sql. Esta biblioteca contiene su propia estructura y hace llamado de sus propias clases fue implementada por la API (Interfaz de Programación de Aplicaciones) de JDBC.

Breve descripción de algunas de sus clases (Sánchez 2003)

Driver: Permite conectarse a una Base de Datos, cada gestor de Base de Datos requiere un Driver distinto.

- DriverManager: Permite gestionar todos los Drivers instalados en el sistema.
- DriverPropertyInfo: Proporciona diversa información acerca de un Driver.

- Connection: Representa una conexión con una Base de Datos. Una aplicación puede tener más de una conexión a más de una Base de Datos.
- DatabaseMetadata: Proporciona información acerca de una Base de Datos, como las tablas que contiene, etc.
- Statement: Permite ejecutar sentencias SQL sin parámetros.
- PreparedStatement: Permite ejecutar sentencias SQL con parámetros de entrada.
- CallableStatement: Permite ejecutar sentencias SQL con parámetros de entrada y salida.
- ResultSet: Contiene las filas o registros obtenidos al ejecutar un SELECT.
- ResultSetMetadata: Permite obtener información sobre un ResultSet, como el número de columnas, sus nombres, etc.

1.4.2 Javax.swing.*

Swing es un conjunto de clases desarrolladas por primera vez para Java 1.2 (el llamado Java2), para reemplazar al anterior paquete que implementaba clases para fabricar interfaces de usuario, el llamado AWT (*Abstract Window Tools*) que en la actualidad todavía se encuentra en uso frecuente.

Tanto Swing como AWT forman parte de una colección de clases llamada JFC (*Java Foundation Classes*) que incluyen paquetes dedicados a la programación de interfaces gráficas. Sin embargo, AWT tenía varios problemas y por ello aparece Swing en la versión 1.2 como parte del JFC que es el kit de clases más importante de Java para las producciones gráficas.

Según (Sánchez 2003) los problemas de AWT son:

- AWT tenía problemas de compatibilidad en varios sistemas.
- A AWT le faltaban algunos componentes avanzados (árboles, tablas).
- Consumía excesivos recursos del sistema.

Swing aporta muchas más clases, consume menos recursos y construye mejor la apariencia de los programas.

Los componentes son los elementos básicos de la programación con Swing. Todo lo que se ve en un GUI de Java es un componente. Los componentes se colocan en otros elementos llamados contenedores que sirven para agrupar componentes.

La clase `javax.swing.JComponent` es la clase padre de todos los componentes. A su vez, `JComponent` descende de `java.awt.container` y ésta de `java.awt.component`. De esto se deduce que Swing es una extensión de AWT, de hecho, su estructura es análoga.

1.4.3 PostgreSQL-8.3-603.jdbc3.jar

Esta biblioteca cuenta con los elementos necesarios, conocidos como manipuladores de dispositivos (drivers), para realizar las conexiones de acuerdo al SGBD al cual se desea conectar. En Cuba se estimula el uso de PostgreSQL, para el cual es necesario contar con la biblioteca PostgreSQL-8.3-603.jdbc3.jar.

1.5 Sistema Gestor de Base de Datos PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarles a otras bases de datos comerciales. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.(Martínez 2010)

1.5.1 Características, limitaciones y ventajas

La última serie de producción es la 9.1. Sus características técnicas la hacen una de las bases de datos más potentes y robustas del mercado. Su desarrollo comenzó hace más de 16 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez a l sistema.

1.5.2 Características Generales

Entre las múltiples características generales que presenta el PostgreSQL se destacan algunas como el soporte a distintos tipos de datos, que además de soportar los base también soporta datos de tipo fecha, monetarios, elementos gráficos, sobre redes (MAC, IP), cadenas de bits, etc. También permite la creación de tipos propios. Incluye herencia

entre tablas, por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales. Completa documentación. Además de todas las antes mencionadas se encuentra disponible para Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows 32/64bit.(Martínez 2010)

Mediante un sistema denominado MVCC (*Acceso concurrente multiversión*, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo commit.

Ventajas:

Algunas de las ventajas que se pueden obtener de la utilización de PostgreSQL son:

Estabilidad y confiabilidad legendarias:

En contraste a muchos sistemas de bases de datos comerciales, es extremadamente común que compañías reporten que PostgreSQL nunca ha presentado caídas en varios años de operación de alta actividad, ni una sola vez, simplemente funciona.

Diseñado para ambientes de alto volumen:

PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC para conseguir una mejor respuesta en ambientes de grandes volúmenes de datos. Los principales proveedores de sistemas de bases de datos comerciales usan también esta tecnología, por las mismas razones.(Martínez 2010).

1.6 JasperReports

JasperReports es la mejor herramienta de código libre en Java para generar reportes. Entrega excelentes presentaciones o diseños en pantalla, para la impresora o para archivos en diferentes formatos como: PDF, HTML, RTF, XLS, CSV y XML. Está completamente escrita en Java y se puede ser utilizada en gran variedad de aplicaciones de Java, incluyendo J2EE o aplicaciones Web, y así poder generar contenido dinámico.(Siddiqui 2010).

1.6.1 Funcionamiento de JasperReports

JasperReports trabaja de forma similar a un compilador y a un intérprete, el usuario diseña el reporte codificándolo en lenguaje XML y definiéndolo en un archivo llamado

jasperreports.dtd. El archivo fuente una vez compilado obtiene la versión compilada del mismo en un archivo que será nombrado “archivo jasper”. Luego de los procesos anteriores se obtiene un “archivo print”, y el mismo puede exportarse en diferentes formatos como PDF, HTML, RTF, XML, XLS, CVS, etc.(Siddiqui 2010).

1.6.2 IReport

iReport es un diseñador visual de código libre para JasperReports escrito en Java. Es un programa que ayuda a los usuarios y desarrolladores que usan la biblioteca JasperReports para diseñar reportes visualmente. A través de una interfaz rica y simple de usar, iReport provee las funciones más importantes para crear reportes de calidad en poco tiempo. El mismo provee a los usuarios de una interfaz visual para construir y generar reportes ayudando a aquellos que no conocen la sintaxis del lenguaje XML. iReport nace como una Herramienta de desarrollo, pero se puede usar como una de oficina para acceder a datos almacenados en una base de datos sin depender de otra herramienta. (Siddiqui 2010)

1.7 Conclusiones Parciales

Durante este capítulo se analizó la problemática existente en la empresa CubaTaxi en la provincia de Villa Clara identificando sus antecedentes y la manera que actualmente es utilizada para el manejo, almacenamiento y obtención de la información. Se describieron las características de los SI. También se detallaron las herramientas a utilizar durante el diseño, implementación y prueba de un sistema de información.

Análisis y Diseño
del sistema “CAAE”

CAPÍTULO 2. ANÁLISIS Y DISEÑO DEL SISTEMA “CAAE” CONTROL DE AUTOS DE ALQUILER ESTATALES.

En este capítulo se hará referencia al análisis y diseño del sistema. Se especificarán los diagramas que se utilizaron en las fases de análisis y diseño mediante UML. Se describen los requisitos tanto funcionales como no funcionales del sistema para tener un amplio conocimiento del caso de estudio. Otro aspecto que se trata en el capítulo es el modo de resguardo de la información de la base de datos.

2.1 Descripción del Sistema

En el departamento de Operaciones se cuenta con toda la información referente a la gestión de uso y alquiler de autos estatales en la provincia de Villa Clara ya sea para prestar servicios de contrato, recogida o piquera. Además, se encuentra toda la información de los autos, ya sea los que se encuentran en activo o no y la de todo el personal que trabaja en función del alquiler de autos.

2.2 Gestión de Requerimientos

La gestión de requerimiento es el punto de partida fundamental para el proceso de creación de cualquier sistema computacional, las entrevistas con el cliente interesado en el desarrollo del sistema es la fuente de información principal para la obtención de requisitos. En esta tarea es el analista el encargado de obtener del cliente toda la información necesaria para que se complete el proyecto de forma exitosa.(Chaves 2011)

2.2.1 Requerimientos no funcionales

Los requerimientos no funcionales son parte importante de la especificación. Forman parte significativa en donde los clientes y los usuarios pueden apreciar y valorar características no funcionales del producto como cuán usable, seguro, conveniente y agradable, pueden marcar grandes diferencias entre un producto bien aceptado y de gran agrado a uno con muy poca aceptación.

A continuación, se muestran algunos de los requisitos no funcionales del sistema propuesto:

1. Memoria RAM de la máquina hospedera debe ser 512 megabytes o superior.
2. Espacio en disco para el código fuente debe ser mayor o igual que 120 megabytes.
3. Sistema Operativo hospedero debe ser Windows XP o superior.

4. Los clientes ligeros que interactúan con la maquina hospedera deben mantener la fecha y hora actual.

2.2.2 Requerimientos funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe ser capaz de cumplir. Estos requerimientos se mantienen invariables sin importar con que propiedades o cualidades se relacionan, en este proceso se obtienen las actividades que serán objeto de automatización. Estas actividades no son exactamente los requerimientos funcionales, pero son el punto de partida para identificar qué es lo que debe realizar el sistema. Como resultado se determinan cuáles son los requisitos con los que debe cumplir este trabajo:

1. Almacenar toda la información referente a los servicios prestados por la empresa.
2. Almacenar el llenado de las tarjetas de circulación que se realiza forma diaria.
3. Almacenar todos los datos de los trabajadores vinculados a esta área de trabajo.
4. Modificar cualquier dato almacenado por el sistema.
5. Obtener reportes tanto de los choferes como de los vehículos relacionados con temas como son la explotación del vehículo o las recaudaciones por chofer en un día de trabajo entre otras.
6. La seguridad de los datos almacenados.
7. Autenticarse.

Al referirse a la gestión de los datos, el sistema se encarga de la inserción, modificación y muestra de los mismos, dando así al usuario facilidades de manipulación y almacenamiento de estos en la Base de Datos.

2.3 Análisis y diseño

Para el diseño del sistema se utilizó el Lenguaje Unificado de Modelado (UML) el cual es un lenguaje de modelado visual que se utiliza para especificar, visualizar, construir y documentar artefactos de un sistema. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios (Jacobson et al. 2000).

De todos los posibles diagramas UML, se realizaron los siguientes:

1. Diagrama de Caso de Uso.

2. Diagrama de Actividad.
3. Diagrama de Clase.
4. Diagrama de Artefacto.

Para el modelado de los diagramas correspondientes a las fases de análisis y diseño se utilizó la herramienta Visual_Paradigm_for_UML_Windows_8.

2.3.1 Diagrama de Caso de Uso

Según Larman (Larman 1999), los casos de uso son eventos de un actor en donde se describen secuencias que conforman un documento narrativo que es utilizado por un sistema para completar un proceso. Estos modelos permiten de forma intuitiva la captura de los requisitos funcionales, basándose en los requerimientos del usuario o actor. Los diagramas de Caso de Uso constituyen el punto de partida para llevar a cabo la mayoría de las actividades: análisis, diseño y prueba del software (Jacobson et al. 2000).

En el siguiente modelo se realiza identificando cada actor del sistema como los posibles usuarios para los cuales está realizado el mismo y las distintas operaciones que podrá realizar cada usuario dentro del sistema. El sistema CAAE cuenta con cuatro tipos de actores los cuales se definen por los diferentes privilegios y acciones que pueden realizar sobre la base de datos.

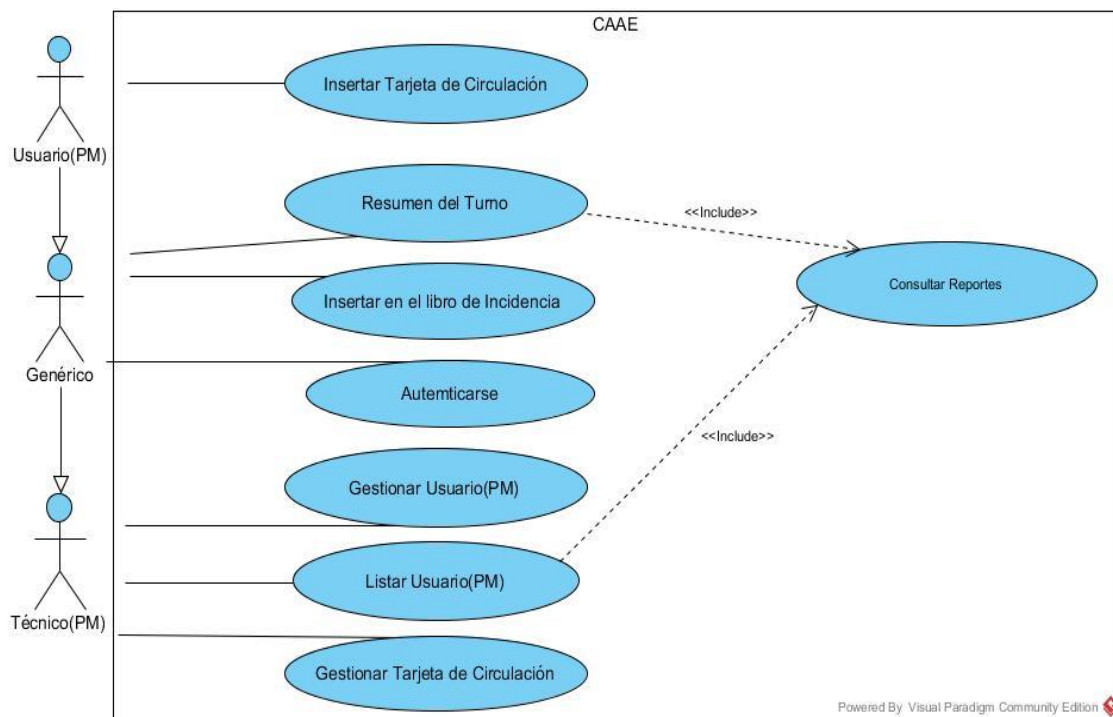


Figura 2.7: Diagrama de Caso de Uso de (“usuario PM” y “Técnico PM”)

Los usuarios de Puesto de Mando (PM) solo pueden insertar o pedir datos a la base de datos. Mediante el caso de uso **"Resumen del Turno"** puede realizar un resumen de su turno completo y de esta forma imprimir dos copias del mismo para poder en el departamento de Economía dejar una junto a las recaudaciones realizadas. A través del caso de uso que se nombra **"Escribir en el Libro de Incidencias"** dejara plasmado en el libro de incidencias si durante su turno ocurrió alguna, también quedara registrado el o los choferes involucrados, vehículos y una observación de cómo ocurrieron los hechos. Además, de los casos de uso antes mencionado el (usuario PM) tiene que hacer uso del caso de uso **"Autenticarse"** para de esta forma quedar registrado en el sistema y mediante el caso de uso **"Insertar tarjeta de Circulación"** se insertará en las tablas correspondientes los datos pertinentes a la tarjeta de circulación.

Los (técnicos PM) podrán realizar las mismas funciones que los (usuarios PM) incrementando el alcance del caso de uso **"Gestionar Tarjeta de Circulación"** en el cual además de poder insertar, podrán modificar los datos almacenados en las tablas correspondientes. También constaran con dos nuevos casos, el primero se nombra **"Gestionar Usuario"** el cual les permite inserciones y actualizaciones de (usuarios PM) y (Técnicos PM). El segundo caso es **"Listar usuarios PM"** mediante el cual se obtiene un reporte con toda la información almacenada referente a estos usuarios.

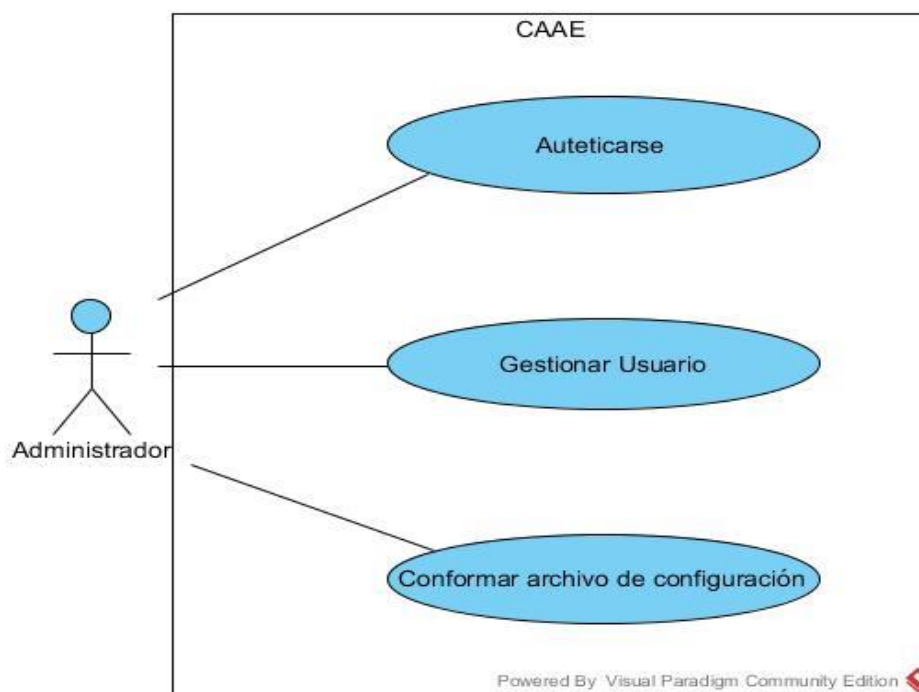


Figura 2.8 : Diagrama de Caso de Uso "Administrador"

En la Figura 2.8 : Diagrama de Caso de Uso "Administrador" se describe el actor “*Administrador*”. Este actor es el encargado de “*Conformar el archivo de configuración*” del sistema, además a de escoger la ruta y el nombre donde se creará el resguardo de los datos almacenados. También como administrador del sistema puede gestionar los usuarios que se encuentran en el mismo, insertando o actualizando los datos de los mismos.

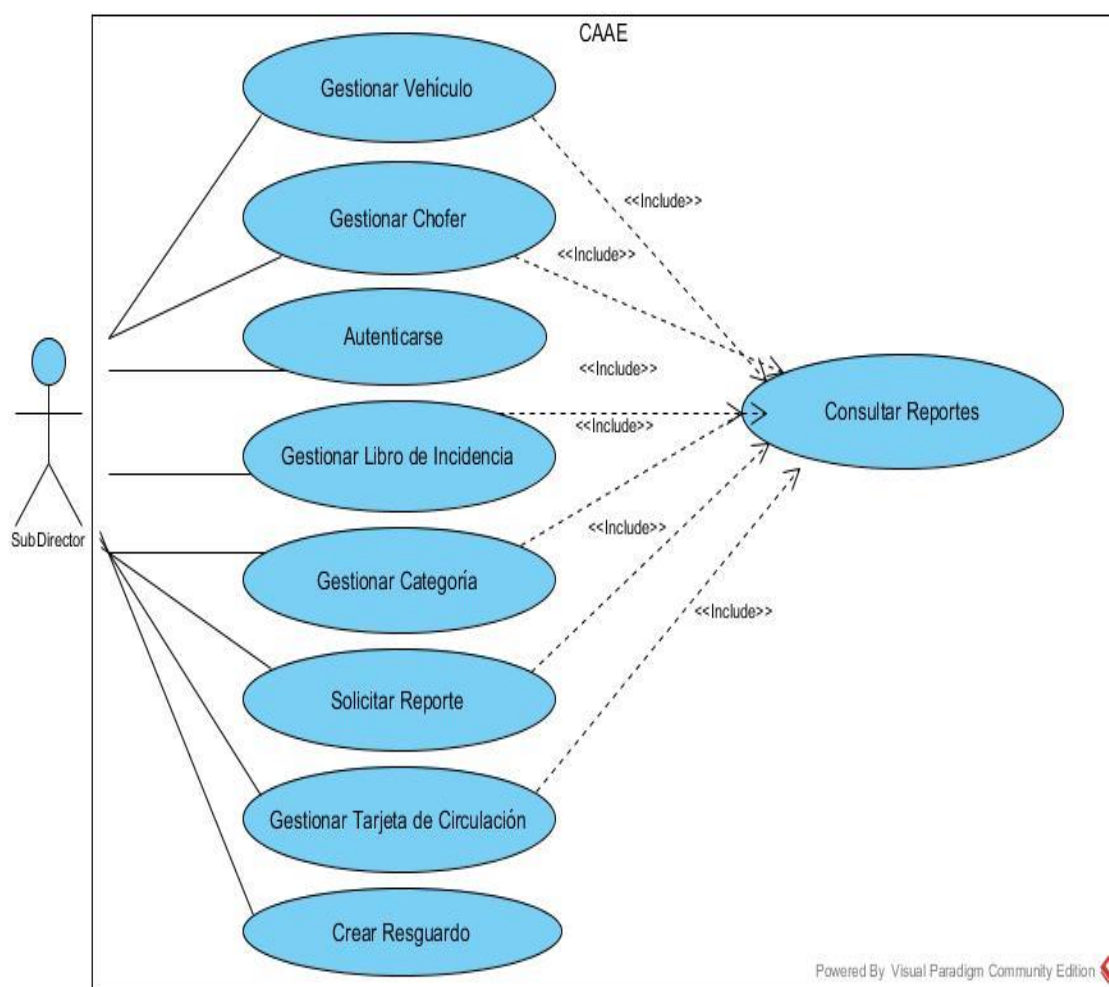


Figura 2.9: Diagrama de Caso de Uso “Subdirector”.

El actor nombrado “*Subdirector de Operaciones*” es el más importante del sistema el cual se explicará de una forma más detallada algunos de sus casos.

Caso de Uso: “Gestionar Vehículo”

Propósito: Gestionar los datos referentes al trabajo con los vehículos.

- Flujo normal de eventos

Acción del actor	Respuesta del sistema
1-Es usuario selecciona la opción de Vehículos	2-Se muestra dos opciones a-) Gestionar b-) Listar
3-El usuario elige la opción “a”	4-Se muestra una ventana con dos submenús a-) Adicionar un vehículo que trae consigo todos los datos referentes al mismo. b-) Modificar el cual permite buscar un Vehículo por su número y modificar sus datos.
5-El usuario escoge la opción ”a” y llena todos los campos pertenecientes al vehículo y selecciona el botón aceptar.	6-Se comprueba que todos los campos estén llenos y que cumplan con las restricciones de los mismos. Además, se comprueba que no exista una fila en la BD con el mismo identificador de Vehículo.
	7-Almacena los datos introducidos y se muestra un mensaje de aceptación de los mismos.
8-El usuario escoge el botón cancelar	9-Se serra la ventana abierta en el paso 4
5-El usuario escoge la opción “b” y manda a buscar el Vehículo por su número	10-Sebusca la fila con ese número en la base de datos. Se muestran todos los datos pertenecientes al vehículo.
11- El usuario modifica los campos que desee y presiona el botón aceptar	6- Se comprueba que todos los campos estén llenos y que cumplan con las restricciones de los mismos.

	7-Almacena los datos introducidos y se muestra un mensaje de aceptación de los mismos.
8-El usuario escoge el botón cancelar	9-Se cerrara la ventana abierta en el paso 4
3-El usuario elige la opción “b”	12-Aparecera la ventana del Reporte
13-El usuario escoge la opción de guardar	14- Muestra la ventana guardar.
15-Selecciona la carpeta donde desea guardar los datos así como el formato de documento en que los desea almacenar y el nombre del mismo	16-Comprueba que todos los campos estén llenos.
	17-Almacena los datos en un documento con el nombre y formato seleccionado y los guarda en la carpeta especificada.

Tabla 2.2: Caso de Uso “Gestionar Vehículo”.

Caso de Uso: “Gestionar Reportes”

Propósito: Gestionar los datos referentes a la solicitud de Reportes.

- Flujo normal de eventos

Acción del actor	Respuesta del sistema
1- Es usuario selecciona la opción de Gestionar Reportes.	2-Se muestra tres opciones a-) Operaciones b-) Técnica c-) Economía
3-El usuario elige la opción “a”.	4- Se muestra una ventana en la que el usuario podrá elegir dos opciones:

	<p>A1-) Realizar los reportes por Choferes.</p> <p>B1-) Realizar los reportes por Vehículo.</p>
5-El usuario escoge la opción “A1” y llena los campos correspondientes.	<p>6-Se desbloquean dos opciones a elegir:</p> <p>A2-) Recaudación</p> <p>B2-) Resultados</p>
7-El usuario escoge la opción “A2” y presiona el botón aceptar.	8-Se muestra un reporte con todos los campos relacionados con la recaudación del chofer entre las fechas especificadas.
9-El usuarios escoge la opción guardar.	10-Muestra la ventana guardar.
11-Selecciona la carpeta donde desea guardar los datos así como el formato de documento en que los desea almacenar y el nombre del mismo	12-Comprueba que todos los campos estén llenos.
	13-Almacena los datos en un documento con el nombre y formato seleccionado y los guarda en la carpeta especificada.
14-Selecciona la opción de Imprimir el reporte.	15-Muestra la ventana Imprimir.
16-Selecciona el botón Aceptar	17- Imprime el reporte.
3-El usuario elige la opción “B2”.	8-Se muestra un reporte con todos los campos relacionados
9-El usuarios escoge la opción guardar.	10-Muestra la ventana guardar.
11-Selecciona la carpeta donde desea guardar los datos así como el formato de documento en que los desea almacenar y el nombre del mismo	12-Comprueba que todos los campos estén llenos.

	13-Almacena los datos en un documento con el nombre y formato seleccionado y los guarda en la carpeta especificada.
3-El usuario elige la opción “B1”.	6-Se desbloquean dos opciones a elegir: c-Recaudación d-Informe de Explotación e-Informe de Explotación Resumen f-Tarjeta de Circulación Recibidas g-Comportamiento de Consumo
3-El usuario elige la opción “c”.	8-Se muestra un reporte con todos los campos relacionados
9-El usuarios escoge la opción guardar.	10-Muestra la ventana guardar.
11-Selecciona la carpeta donde desea guardar los datos así como el formato de documento en que los desea almacenar y el nombre del mismo	12-Comprueba que todos los campos estén llenos.
	13-Almacena los datos en un documento con el nombre y formato seleccionado y los guarda en la carpeta especificada.
14-Selecciona la opción de Imprimir el reporte.	15-Muestra la ventana Imprimir.
16-Selecciona el botón Aceptar	17- Imprime el reporte.
3-El usuario elige la opción “d”.	8-Se muestra un reporte con todos los campos relacionados
9-El usuarios escoge la opción guardar.	10-Muestra la ventana guardar.

11-Selecciona la carpeta donde desea guardar los datos así como el formato de documento en que los desea almacenar y el nombre del mismo	12-Comprueba que todos los campos estén llenos.
	13-Almacena los datos en un documento con el nombre y formato seleccionado y los guarda en la carpeta especificada.
14-Selecciona la opción de Imprimir el reporte.	15-Muestra la ventana Imprimir.
16-Selecciona el botón Aceptar	17- Imprime el reporte.
Se repiten los pasos 3 y del 8 al 17 para las opciones en que el usuario elige “e”, ”f”, ”g”.	

Tabla 2.3: Caso de Uso “Gestionar Reportes”.

2.3.2 Esquema conceptual de la Base de Datos

La base de datos (BD) que sostiene un sistema de información es el motor principal del mismo, partiendo de la arquitectura ANSI/SPARC de tres niveles. Mediante el modelo Entidad-Relación como referencia (Chen 1994) en el esquema conceptual de la BD se representan interrelaciones y entidades que son las encargadas de soportar los procesos que intervienen en el negocio. Una interrelación puede expresar restricciones de dominio, asociaciones, especificaciones, generalizaciones entre otras, mientras que las entidades pueden ser los actores o la información que se puede obtener de aplicar un proceso.

La BD es el medio que nos permite mediante consultas SQL acceder a toda la información almacenada dentro del sistema. En la Figura 2.10:Diagrama Entidad-Relación. se muestra el diagrama de Entidad-Relación (DER n.d.) para el sistema propuesto. Puesto que los requerimientos de un sistema pueden cambiar fácilmente con la aparición de nuevas características, se pretende lograr un diseño de la BD el cual le imprima un carácter flexible a la información. La base de datos contiene tipos de entidades centrales para establecer un orden lógico en el soporte de los procesos, en concordancia con los casos de uso propuestos con anterioridad.

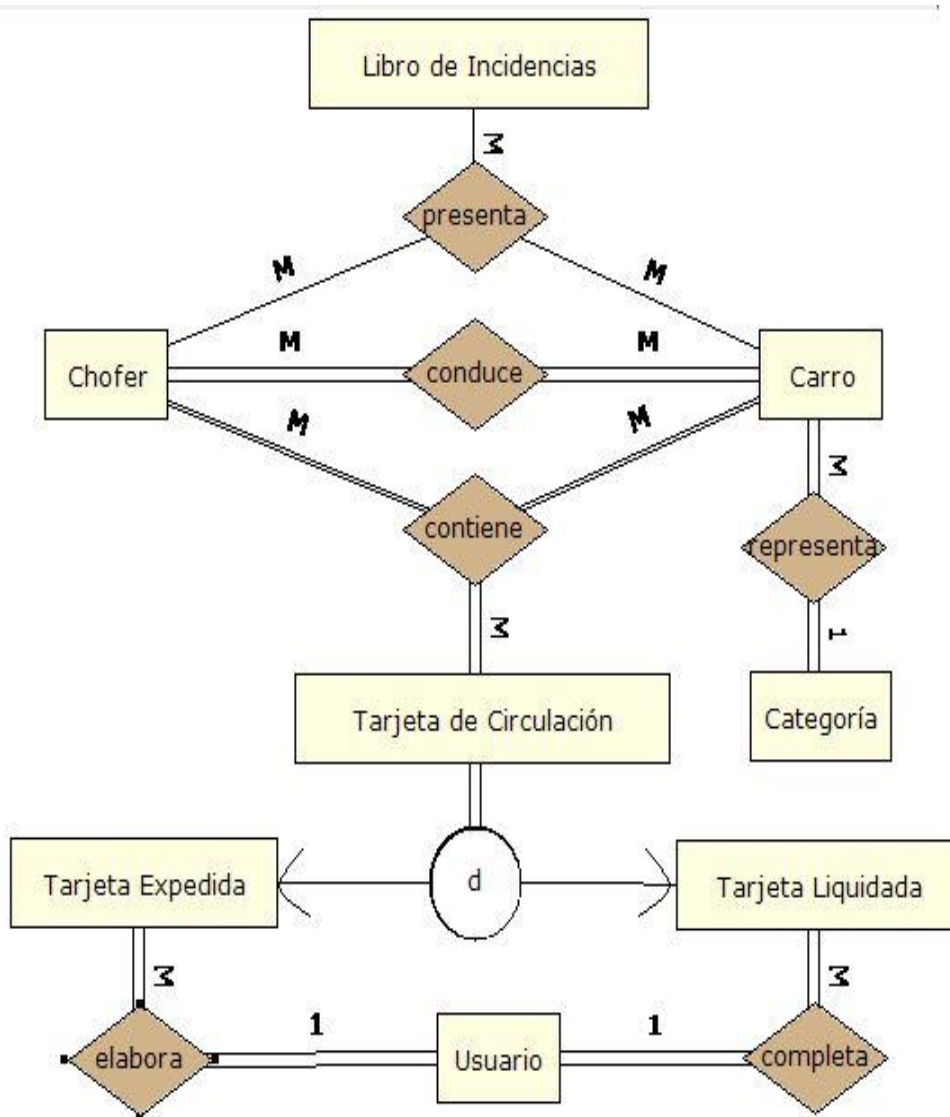


Figura 2.10:Diagrame Entidad-Relación.

El tipo de entidad Chofer contiene todos los datos referentes al mismo como son: ci, nombre, primer apellido, segundo apellido y un conjunto de fechas en las que vencen sus chequeos médicos y permisos de transportación. Las distintas interrelaciones ternarias que se modelan en este diagrama generan algunas tablas como son el caso de “contiene” que relaciona las entidades Chofer, Carro y Tarjeta de Circulación en la cual se muestran los identificadores de las mismas en los distintos días de trabajo. Además, las interrelaciones contribuyen a la implementación de características automáticas en el sistema al manejar la información.

Uno de los aspectos más importante que se tiene en cuenta en este modelo es que en la tabla Tarjeta de Circulación y por consecuencia en las que heredan de ella como son tarjeta Expedida y Tarjeta Liquidada no se puede insertar si el Chofer seleccionado no tiene todos sus chequeos pertinentes en orden y su estado es Activo, al igual que con el Vehículo seleccionado el cual debe tener todos sus requerimientos funcionales en orden, que la cantidad de kilómetros disponibles sea mayor que 50 y que su estado sea Activo. Todas estas restricciones se logran cumplir mediante un disparador el cual se muestra a continuación:

```
CREATE OR REPLACE FUNCTION asignacion()

    RETURNS trigger AS

$BODY$

    declare ficav date;

    declare lot date;

    declare km_disp integer;

    declare estado character varying;

    declare venc_lic date;

    declare fsicof date;

    declare cheq_medico date;

    declare recalificacion date;

    declare estadochof character varying;

    declare carr character varying;

    declare chof character varying;

    begin

    select "tarjeta_circulacion"."id_carro",

           "tarjeta_circulacion"."ci_chofer" into carr,chof

    from "tarjeta_circulacion"

    where "tarjeta_circulacion"."folio"=new."tarjeta_circulacion_folio";

    select "carro"."ficav","carro"."lot","carro"."km_disponible",

           "carro"."estado" into ficav,lot,km_disp,estado

    from "carro"
```

```
Where "carro"."id_carro"="carr";

select "chofer"."fecha_vence_licencia","chofer"."fsicofsiologico",
       "chofer"."chequeo_medico","chofer"."recalificacion",
       "chofer"."estado" into venc_lic,fsicof,cheq_medico,
                               recalificacion,estadochof

from "chofer"

where "chofer"."ci"="chof";

if "ficav"<new."fecha_salida" then

delete

from "tarjeta_circulacion"

where tarjeta_circulacion"."folio"=new."tarjeta_circulacion_folio";

raise exception 'Asignación no válida se ha vencido el ficav del
                Vehículo';

return null;

elsif "lot"<new."fecha_salida" then

delete

from "tarjeta_circulacion"

where "tarjeta_circulacion"."folio"=new."tarjeta_circulacion_folio";

raise exception 'Asignación no válida se ha vencido el lot del
                Vehículo';

return null;

elsif "km_disp" <= 50 then

delete

from "tarjeta_circulacion"

where "tarjeta_circulacion"."folio"=new."tarjeta_circulacion_folio";

raise exception 'Asignación no válida los kilmetros disponibles del
                Vehículo son menores o iguales a 50';

return null;

elsif "estado"!='Activo'then

delete
```

```
from "tarjeta_circulacion"

where "tarjeta_circulacion"."folio"= new."tarjeta_circulacion_folio";

raise exception 'Asignación no válida el estado del Vehículo no

permite prestar servicio con el mismo';

return null;

elsif "venc_lic"<new."fecha_salida" then

delete

from "tarjeta_circulacion"

where "tarjeta_circulacion"."folio"= new."tarjeta_circulacion_folio";

raise exception 'Asignación no válida la licencia del

chofer ha vencido';

return null;

elsif "fsicof" < new."fecha_salida" then

delete

from "tarjeta_circulacion"

where "tarjeta_circulacion"."folio"= new."tarjeta_circulacion_folio";

raise exception 'Asignación no válida el fsicofsiologico del chofer a

vencido';

return null;

elsif "cheq_medico" < new."fecha_salida" then

delete

from "tarjeta_circulacion"

where "tarjeta_circulacion"."folio"= new."tarjeta_circulacion_folio";

raise exception 'Asignación no válida el chequeo médico del chofer a

vencido';

return null;

elsif "recalificacion" <new."fecha_salida" then

delete

from "tarjeta_circulacion"

where "tarjeta_circulacion"."folio"= new."tarjeta_circulacion_folio";
```

```
raise exception 'Asignación no válida la recalificación del chofer a
vencido';

return null;

elsif "estadochof"!='Activo'THEN

delete

from "tarjeta_circulacion"

where "tarjeta_circulacion"."folio"= new."tarjeta_circulacion_folio";

raise exception 'Asignación no válida el estado del chofer no es
activo por lo que no puede prestar servicios';

return null;

else

return new;

end if;

end;

$BODY$

LANGUAGE plpgsql VOLATILE

COST 100;

ALTER FUNCTION asignación ()

OWNER TO Osniel;
```

2.3.3 Diagrama de actividad

El diagrama de actividad se utiliza para el modelado de los aspectos dinámicos de los sistemas y fundamentalmente en un diagrama que lo que haga es mostrar el flujo de control entre las actividades. El diagrama de actividad para el caso de uso de la **¡Error!** **No se encuentra el origen de la referencia.** que se muestra describe el proceso que se debe seguir para agregar los datos pertenecientes a la inserción de un nuevo chofer en el sistema. En el mismo se muestran los pasos a transitar para la validación de los datos introducidos (Canchala 2004).

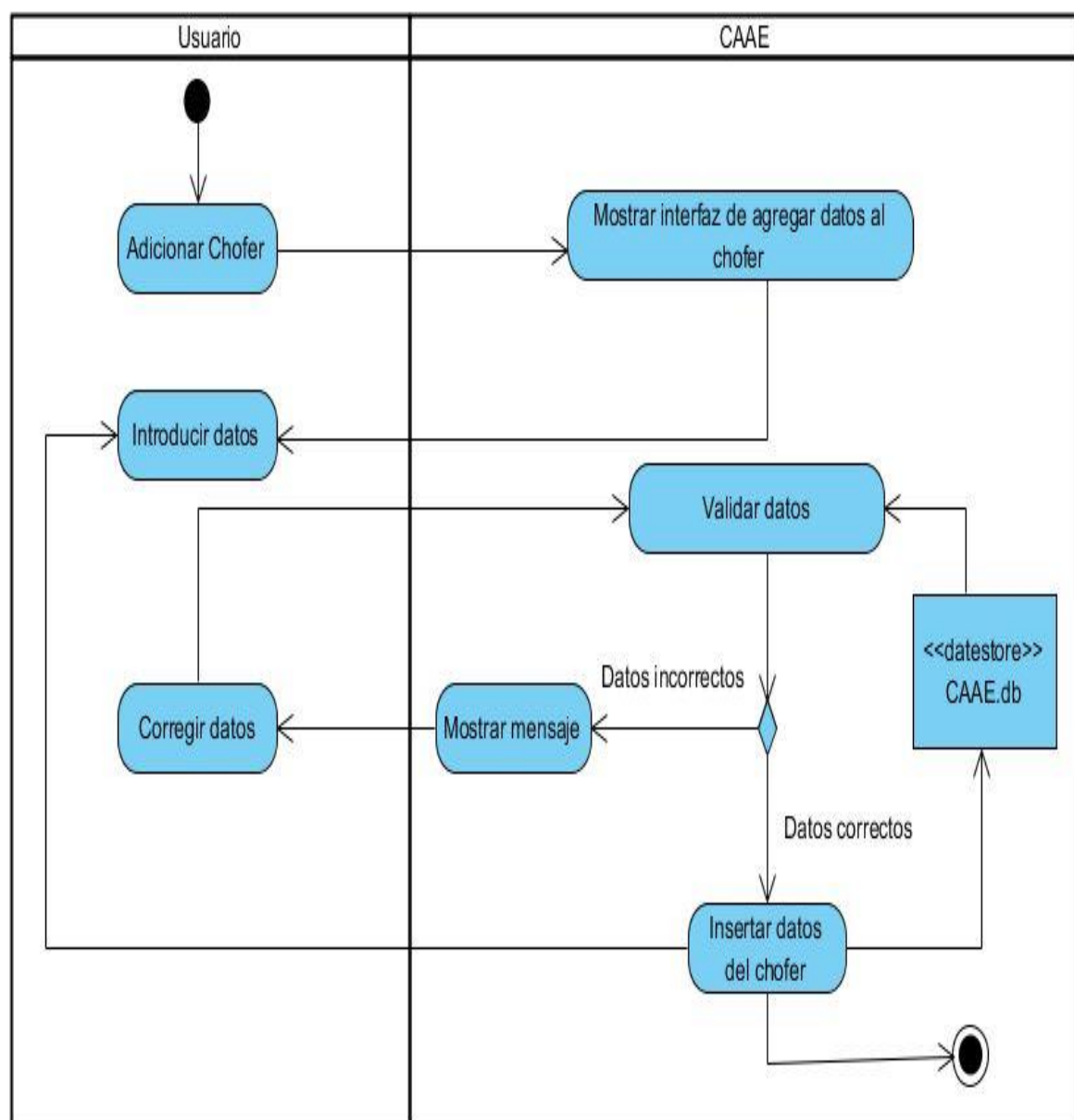


Figura 2.11:Diagrama de Actividad para el Caso de Uso “Agregar Chofer” (extensión del Case de Uso “Gestionar Chofer”).

Para fortalecer la seguridad del sistema es de forma obligatoria la autenticación como se muestra en la; **Error! No se encuentra el origen de la referencia..** Con la acción de “Validar usuario y contraseña” se verifica si el usuario existe en la BD o no, mostrándose un mensaje en caso de que este usuario no exista.

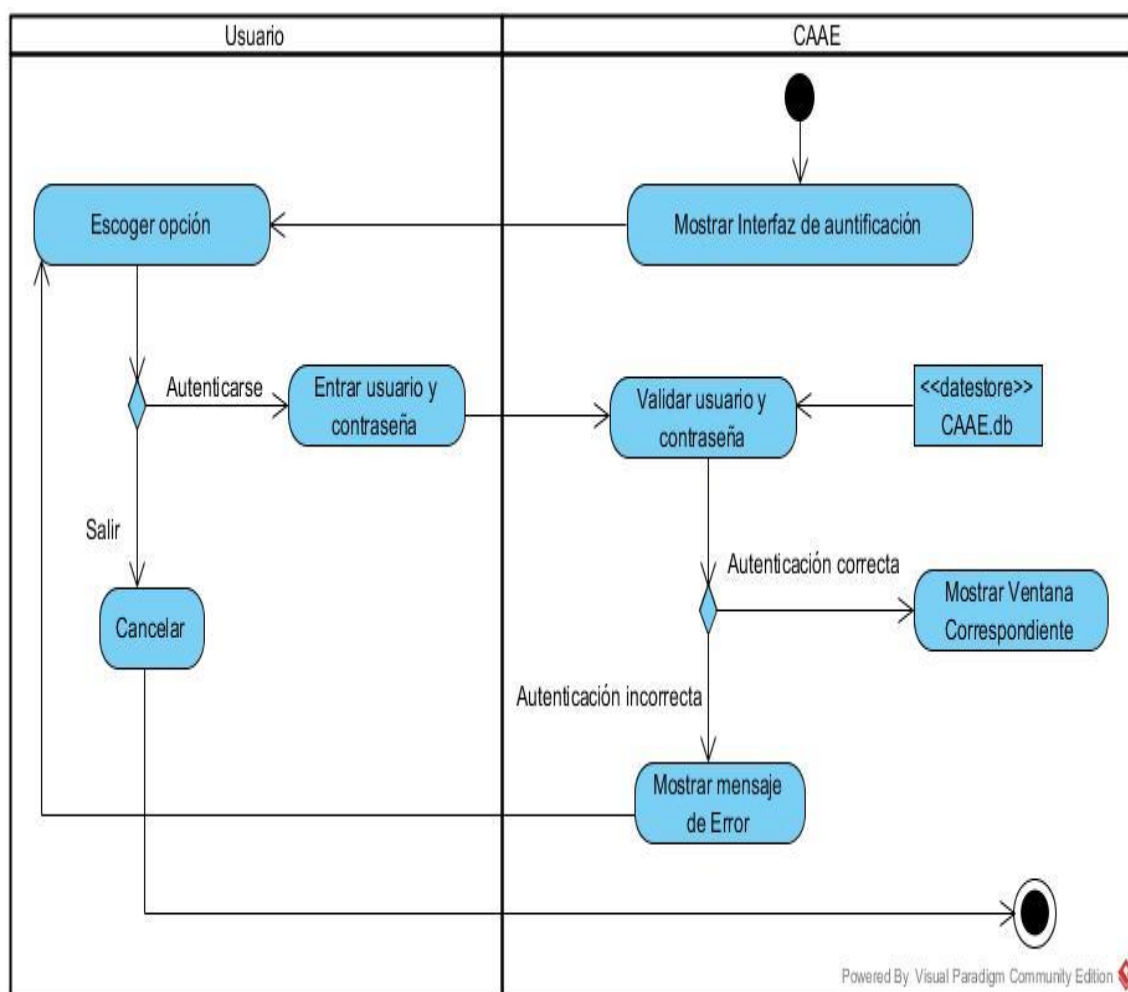


Figura 2.12: Diagrama de Actividad para el Caso de Uso "Autenticación".

2.3.4 Diagrama de componentes

Este tipo de diagrama es uno de los que aparecen cuando se modelan aspectos físicos de los sistemas orientados a objetos. Muestra la organización y dependencias entre un conjunto de componentes y se utiliza para comunicar un aspecto de la vista de implementación estática de un sistema (Torres 2004).

Los componentes se utilizan para modelar elementos físicos tales como: ejecutables, bibliotecas, tablas, archivos y documentos.

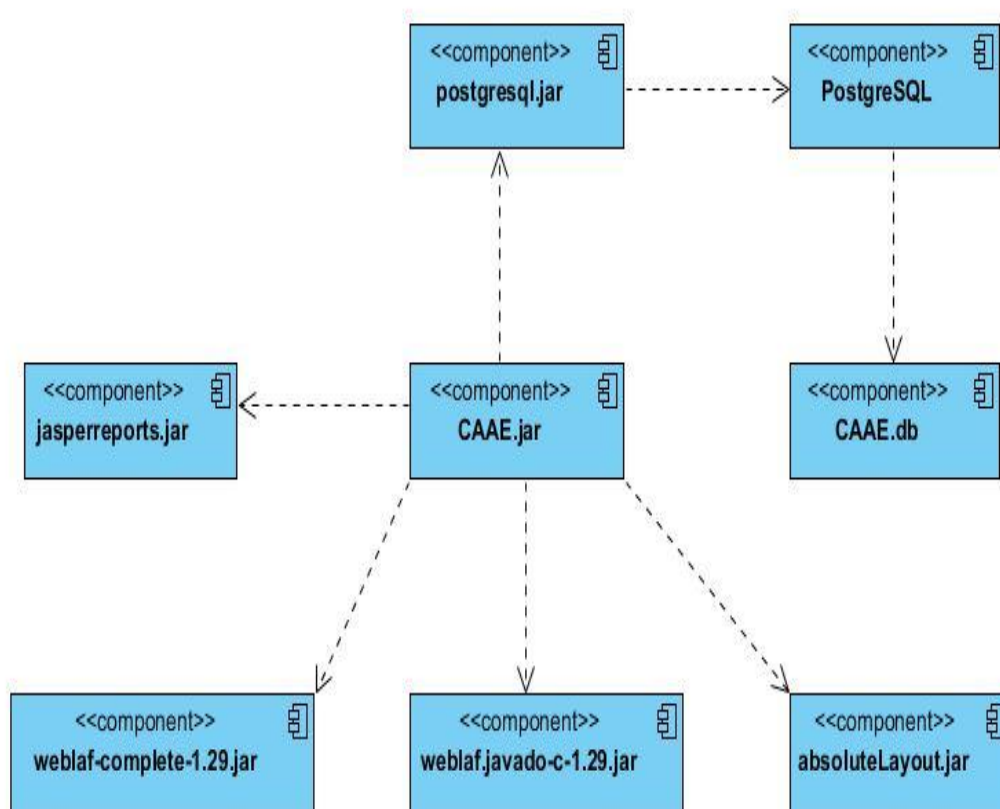


Figura 2.13: Diagrama de Componentes.

2.3.5 Diagrama de Clases

El diagrama de clases describe los tipos de objetos que hay en un sistema y las diversas clases de relaciones estáticas (asociaciones, subtipos) que existen entre ellos. En la Figura 2.14: Diagrama de Clases. se muestra el diagrama de clases referente a nuestro sistema, dividido por los diferentes paquetes en los cuales se encuentran todas las clases necesarias para lograr satisfacer los requerimientos funcionales (Zapata et al. 2006).

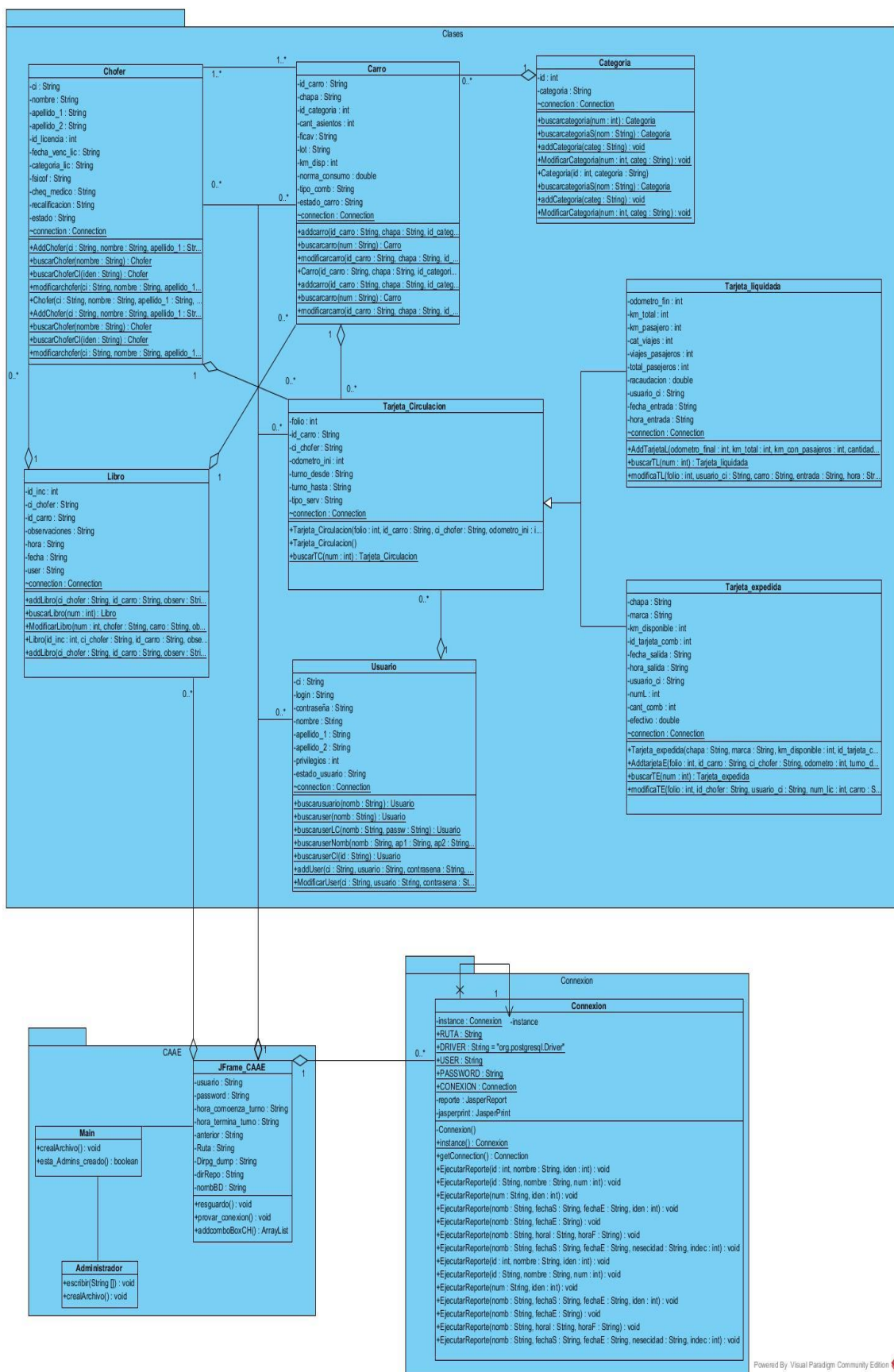


Figura 2.14: Diagrama de Clases.

2.3.6 Diagrama de Despliegue

Según (Booch et al. 1999), los diagramas de despliegue son un tipo de diagrama del UML que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. Algunos de los usos que se les da a los diagramas de despliegue son para modelar:

Sistemas empotrados: Un sistema empotrado es una colección de hardware con una gran cantidad de software que interactúa con el mundo físico.

Sistemas cliente-servidor: Los sistemas cliente-servidor son un extremo del espectro de los sistemas distribuidos y requieren tomar decisiones sobre la conectividad de red de los clientes a los servidores y sobre la distribución física de los componentes software del sistema a través de nodos.

Sistemas completamente distribuidos: En el otro extremo se encuentran aquellos sistemas que son amplia o totalmente distribuidos, y que normalmente incluyen varios niveles de servidores. Tales sistemas contienen, a menudo, varias versiones de componentes software, alguno de los cuales pueden incluso migrar de un nodo a otro. El diseño de tales sistemas requiere tomar decisiones que permitan un cambio continuo de la topología del sistema.

El diagrama de despliegue muestra la disposición física de los distintos nodos que entran en la composición de un sistema y el reparto de los programas ejecutables sobre estos nodos. En nuestro caso se usa una arquitectura cliente-servidor. En el servidor debe estar instalado un gestor de Base de Datos PostgreSQL versión 9.0.1 y en los clientes el Java Runtime Environment o JRE 7, lo cual va a permitir la ejecución de programas Java y el sistema CAAE.jar.

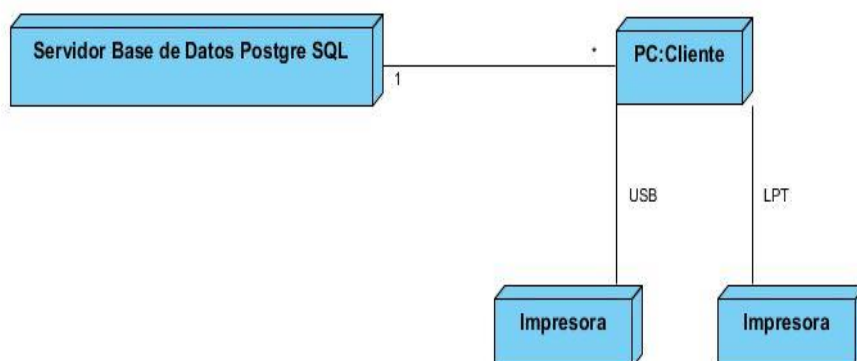


Figura 2.15: Diagrama de Despliegue de la aplicación.

2.3.7 Patrón de arquitectónico Modelo vista Controlador (MVC)

El MVC (González & Romero 2012) es un patrón de arquitectura de las aplicaciones software que nos permite mantener individualizadas, las “responsabilidades” dentro de un sistema, permitiéndonos diferenciar y aislar el diseño del sistema (modelo), de la interfaz gráfica del usuario (GUI) y su correspondiente lógica de negocios (vistas), utilizando como “conector intermediario” un objeto controlador, incrementando la reutilización y flexibilidad del código. A continuación, se ofrece una breve descripción del patrón:

Modelo: Es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos.

Vista: Presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

Controlador: Responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

Aunque se pueden encontrar diferentes implementaciones del patrón Modelo-Vista-Controlador, el flujo que sigue el control generalmente es el siguiente:

- El usuario interactúa con la interfaz (por ejemplo, pulsa un botón de enlace).

- El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
- El controlador accede al modelo actualizándolo de acuerdo a la solicitud del usuario. Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
- El controlador delega a los objetos de la vista, la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar una interfaz apropiada para el usuario donde refleja los cambios en el modelo.
- La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

En la Figura 2.16: Modelo Vista Controlador. se muestra de forma sencilla Modelo-Vista-Controlador

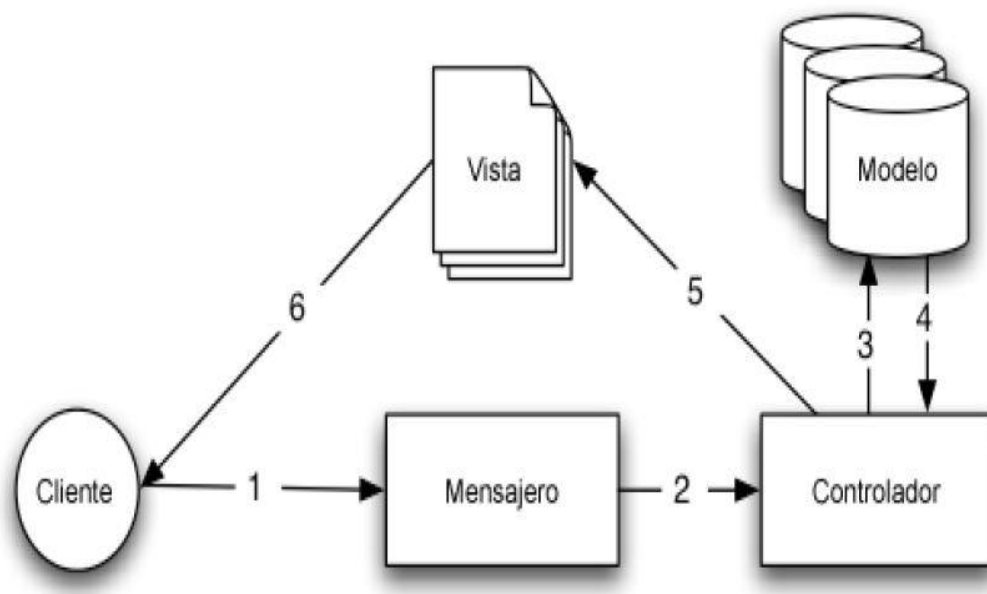


Figura 2.16: Modelo Vista Controlador.

2.3.8 Herramientas para el desarrollo

Para la creación de sistema CAAE se utilizó como lenguaje de programación Java, el cual se conecta con el servidor de BD PostgreSQL versión 9.4.1. El entorno de desarrollo utilizado es la versión 8.0.2 del IDE NetBaens y a él se le debe incorporar la versión 5.5.0

del plugin iReport. Para continuar extendiendo el sistema se debe agregar al proyecto un conjunto de bibliotecas que fueron utilizadas en su desarrollo:

Bibliotecas	Uso de la Biblioteca
AbsoluteLayout.jar	Para el trabajo con imágenes.
Weblaf.jar	Para el trabajo con los campos de tipo fecha.
nimrodlf-1.2.jar	Modifica la parte visual del proyecto
postgresql-9.2-1002.jdbc3.jar	Conexión con el gestor de Base de Datos

Tabla 2.4: Herramientas para el desarrollo.

2.4 Conclusiones parciales

En este capítulo se realizó un análisis de los requerimientos funcionales y no funcionales del sistema y un estudio de los casos de uso que se encargan de delimitar el sistema definiendo las funciones que debe cumplir para un usuario específico (los actores). Se obtuvo un diseño mediante el modelado de las características principales del sistema utilizando la notación UML para una mejor comprensión de la estructura. También se diseñó la base de datos y la disposición de las clases utilizando la arquitectura Modelo-Vista-Controlador para enfrentar la solución del problema siguiendo la estrategia de divide y vencerás.

*Implementación y
prueba del
sistema “CAAE”*

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA DEL SISTEMA “CAAE”.

En el siguiente capítulo se exponen aspectos relacionados con la implementación del sistema y se explican algunas de las características del software para facilitar su entendimiento por el usuario. También se abarcarán los sistemas de pruebas y explicaremos las pruebas aplicadas al sistema “CAAE”.

3.1 Sistema de Pruebas

Un sistema de pruebas implica la aplicación del mismo a través de condiciones controladas y la consiguiente evaluación del mismo. Dentro de las condiciones controladas se deben incluir situaciones tanto normales como anormales. El objetivo del sistema de pruebas es encontrar errores en nuestra aplicación. El sistema de pruebas está conformado por un conjunto pruebas que se enuncian a continuación:

- **Pruebas de caja negra:** El sistema de pruebas de caja negra no están basadas en el conocimiento del diseño interno del programa. Estas pruebas están enfocadas en los requerimientos establecidos y la funcionabilidad del sistema (López et al. 2005).
- **Pruebas de caja blanca:** Las pruebas de caja blanca (también conocidas como pruebas de caja de cristal o pruebas estructurales) se centran en los detalles procedimentales del software, por lo que su diseño está fuertemente ligado al código fuente. Se escogen distintos valores de entrada para examinar cada uno de los posibles flujos de ejecución del programa y cerciorarse de que se devuelven los valores de salida adecuados. Al estar basadas en una implementación concreta, si ésta se modifica, por regla general las pruebas también deberán rediseñarse (Yagüe & Garbajosa 2009).
- **Pruebas de Integración:** Las pruebas de integración buscan probar la combinación de las distintas partes de la aplicación para determinar si funcionan correctamente en conjunto (Figuerola 2009).
- **Pruebas de sistema:** Son similares a las pruebas de caja negra lo que estas buscan probar el sistema como un todo. Están basadas en los requerimientos generales y abarcan todas las partes combinadas del sistema (Gutiérrez et al. 2006).

Haciendo uso de las pruebas explicada anteriormente se definieron tres rubros en base a los cuales se van a definir los puntos a evaluar:

- **Pruebas de Contenido:** Estas pruebas buscan verificar que el contenido del sistema sea coherente y consistente a la vez. También se debe verificar que las palabras usadas para transmitir una idea al usuario sean la adecuadas. Gran parte de las pruebas de contenido realizadas se enfocaron en los modelos “CAAE-3” y “CAAE-4” ya que son los de mayor interacción con el usuario.
- **Pruebas de Funcionalidad:** Este tipo de pruebas examina si el sistema cubre sus necesidades de funcionamiento, acorde a las especificaciones de diseño. En ellas se verifica si el sistema lleva a cabo correctamente todas las funciones requeridas, se validan los datos y se deben realizar pruebas de comportamiento en diferentes escenarios. Estas pruebas se deben enfocar a tareas, límites del sistema, a condiciones planeadas de error. Para estas pruebas usamos los esquemas de caja negra, puesto que nos interesa saber si funciona o no, independiente de la forma en que lo haga.
- **Pruebas de Usabilidad:** En estas pruebas se verifica que tan fácil de usar es nuestro sistema. Con ella se deben verificar aprendizaje (qué tan fácil es para los usuarios realizar tareas básicas la primera vez que interactúan con el sistema), eficiencia (una vez que los usuarios han aprendido algo del sistema, que tan rápido pueden llevar a cabo las tareas), manejo de errores (cuantos errores comete 1 usuario, la gravedad de los mismos y que tan fácil se recupera de ellos) y el grado de satisfacción (que tan satisfactorio es usar el sistema). Según (Nielsen & Loranger 2006) para identificar los problemas más importantes de usabilidad de un sistema es suficiente con cinco usuarios. Además, para obtener resultados realistas es importante dejar que los usuarios resuelvan los problemas por sí mismo, sin ayuda nuestra de lo contrario estaríamos contaminando la prueba.

3.2 Resultados de las Pruebas

Las pruebas del sistema, se llevaron a cabo en el transcurso de una semana con los usuarios encargados de trabajar con el sistema en la empresa de CubaTaxi. Durante el transcurso de la semana se recopilaron los datos necesarios para la conformación de los gráficos que se muestran a continuación.

Cabe destacar que todas las evaluaciones se realizaron en una escala de evaluación de 1 a 5 siendo 1 “Muy bajo”, 2 “Bajo”, 3 “Media”, 4 “Alto” y 5 “Muy alto”. En estas se cuantificaron los siguientes atributos del sistema.

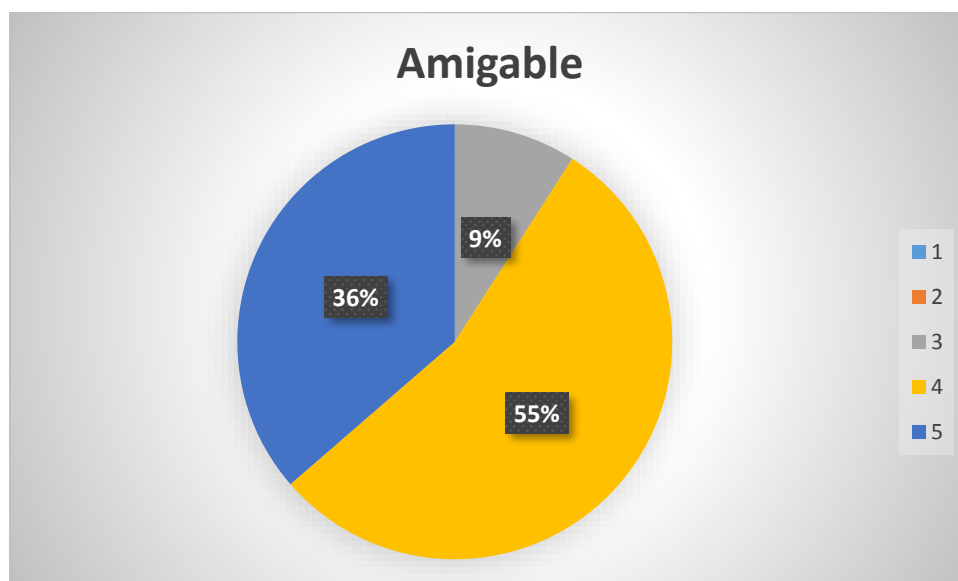


Figura 17: Amigable.

Amigable: Se refiere a la facilidad de interacción del sistema con el usuario sin tener que consultar la ayuda.

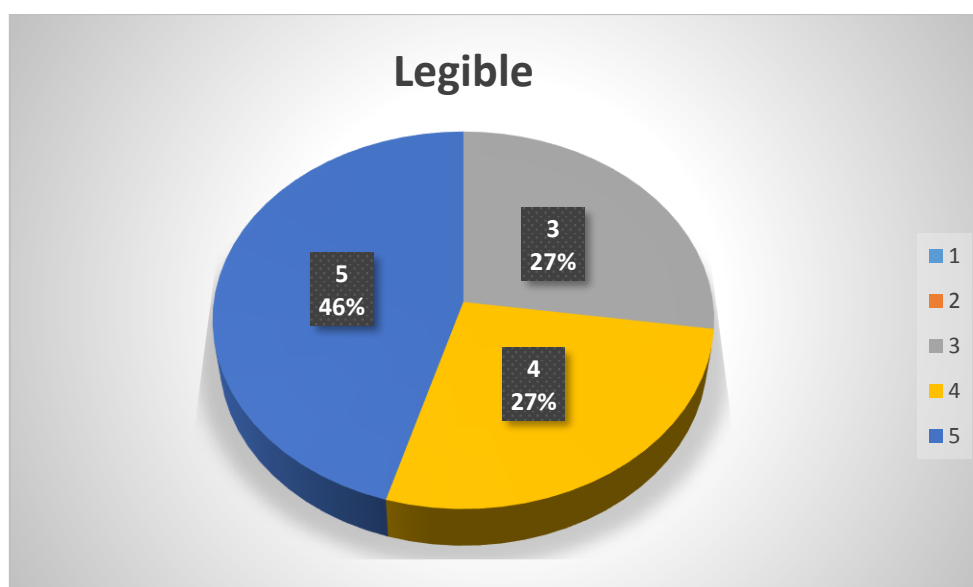


Figura 18: Legible.

Legibilidad: En esta prueba se evalúa el color de los textos el contraste de los mismos con el fondo y el tamaño de la letra que debe ser adecuado para todos los usuarios. En

otras palabras, es cuan comprensible es la interfaz del software para los distintos tipos de usuarios que interactúan con él.

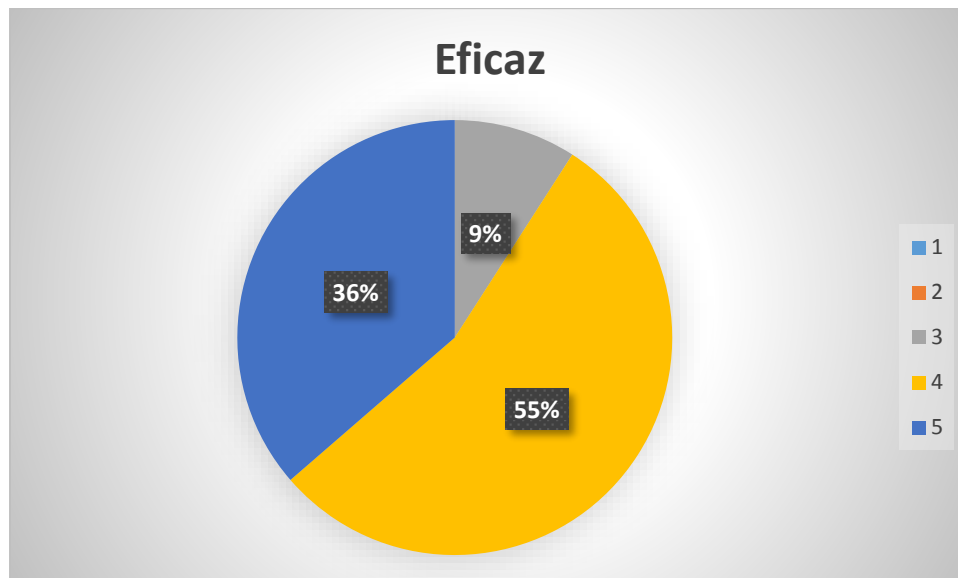


Figura 19: Eficaz.

Efectividad: Es el cumplimiento de una tarea sin complicaciones.

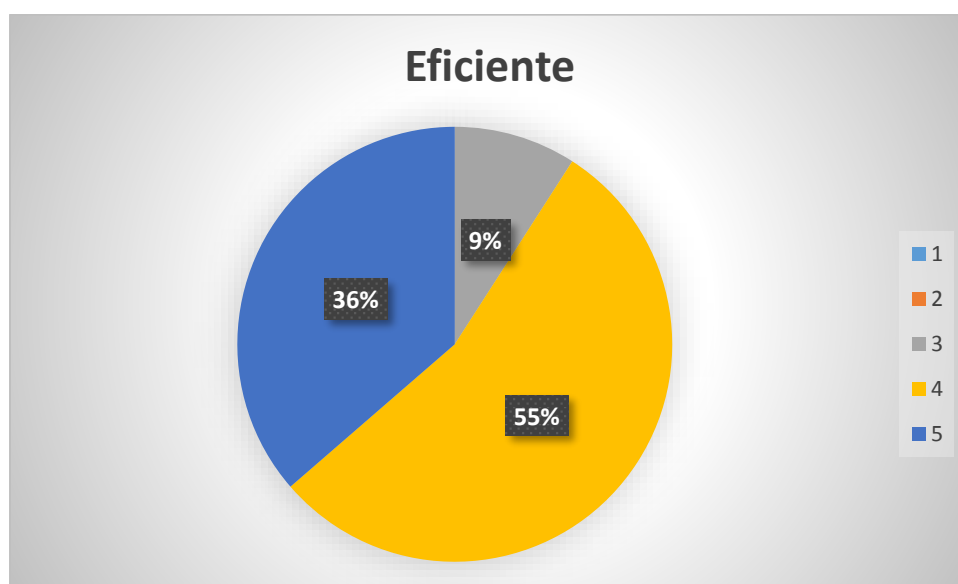


Figura 20: Eficiente.

Eficiencia: Cuando las tareas que se llevan a cabo, pueden ser analizadas de forma fácil y rápida.



Figura 21: Satisfacción.

Satisfacción: Es que tan a gusto quedaron los usuarios con las tareas realizadas por el sistema.

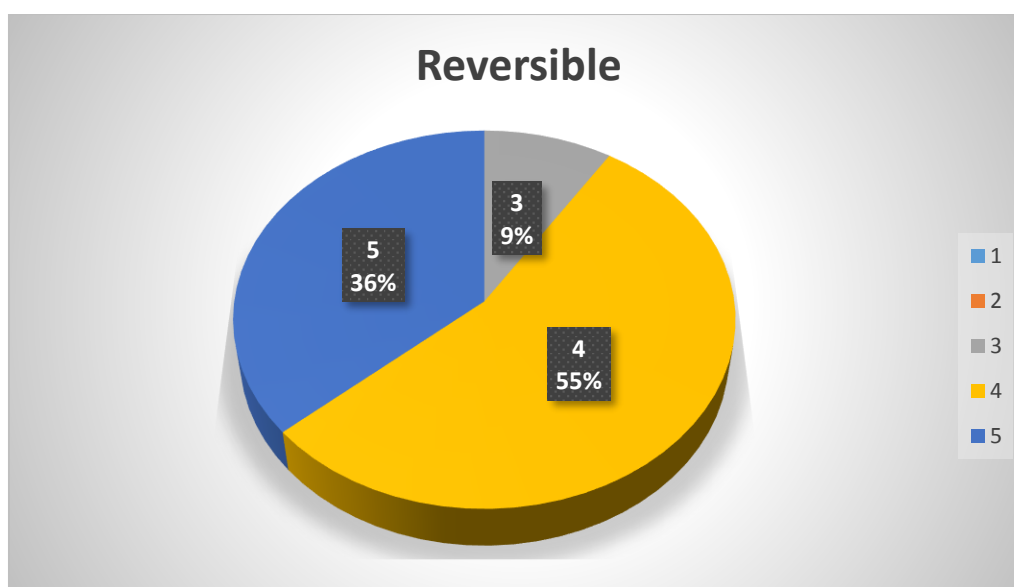


Figura 22: Reversible.

Reversibilidad: La capacidad del sistema para permitir deshacer las acciones realizadas.

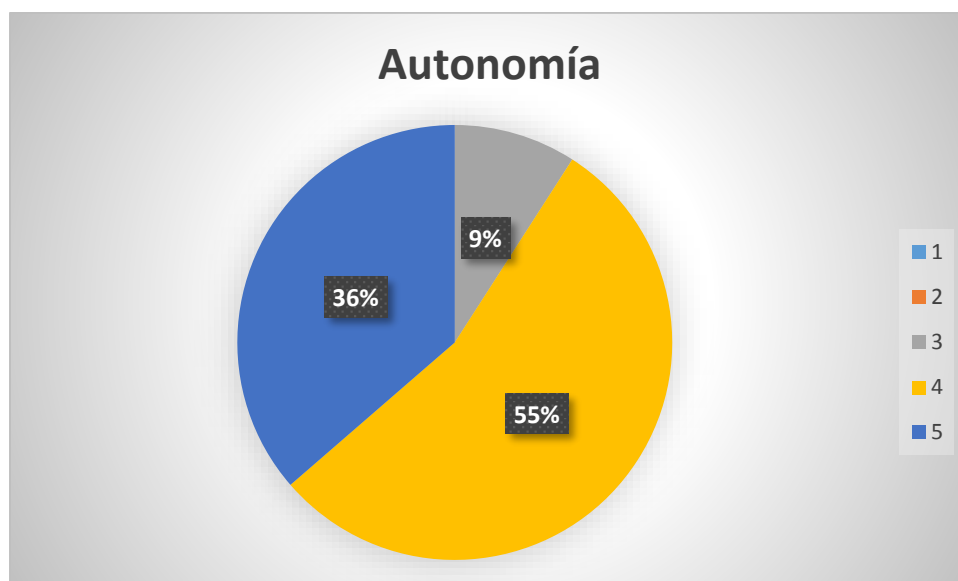


Figura 23: Autonomía.

Autonomía: Se refiere a que los usuarios tienen el control sobre el sistema todo el tiempo.

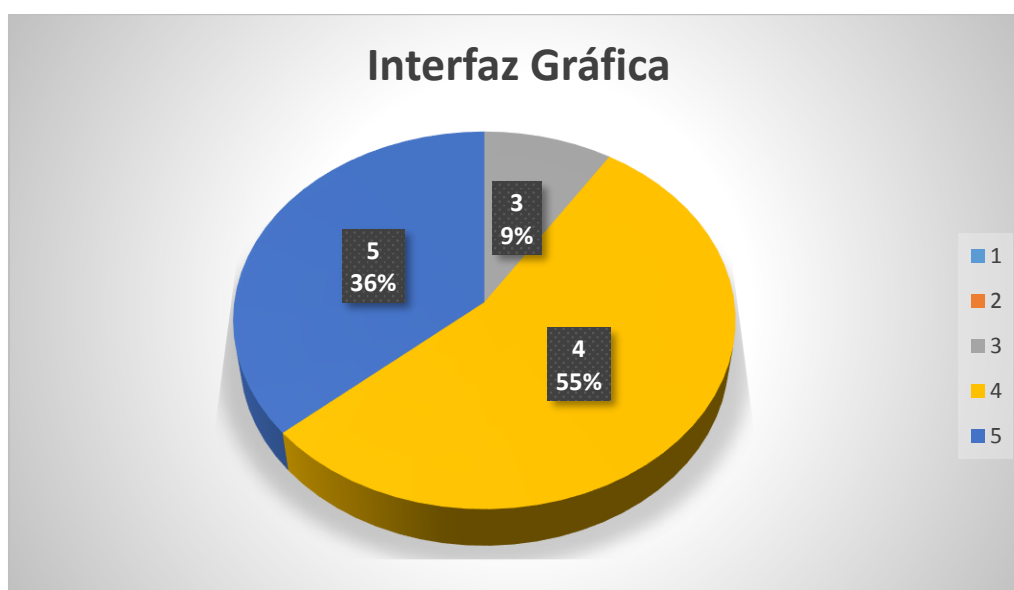


Figura 24: Interfaz Gráfica.

Interfaz Gráfica: Significa que tan a gusto se sienten los usuarios trabajando con el sistema. Esto incluye las imágenes, colores y la posición de los elementos que la conforman.

Dentro de las distintas pruebas de funcionalidad aplicadas al sistema para comprobar que cumpliera todas las necesidades se muestra en la Figura 25: Mensaje de Error una de las

condiciones planeadas de error. En este caso es que el chofer designado no puede salir a prestar servicio porque el chequeo médico a vencido.

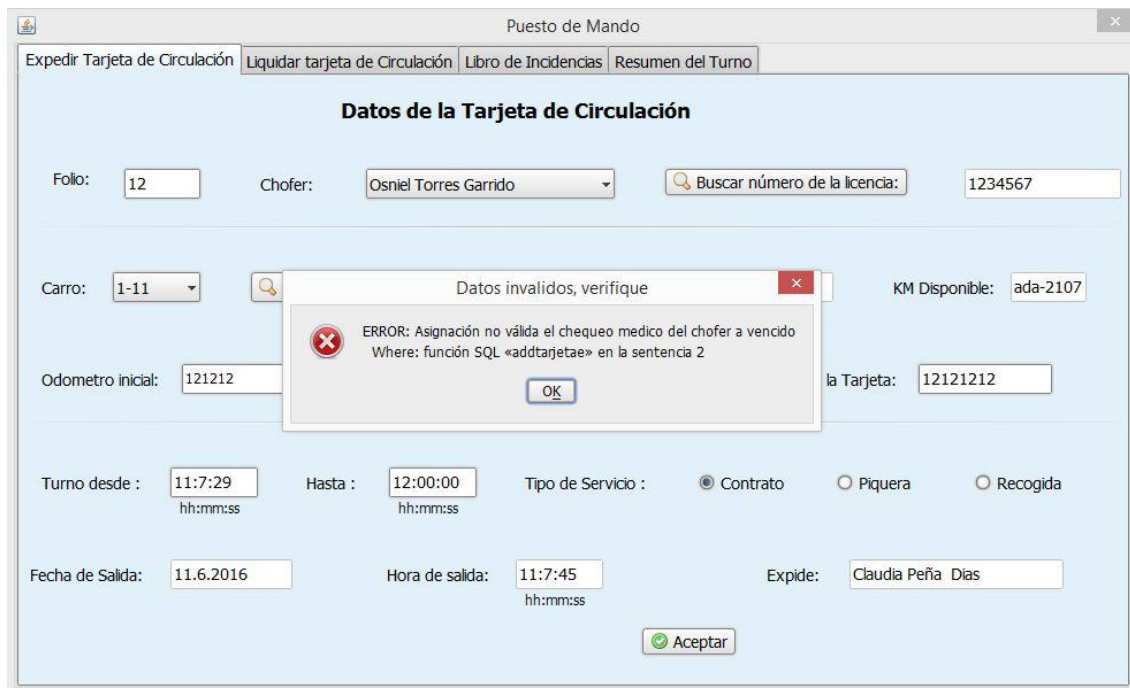


Figura 25: Mensaje de Error

Existen otras condiciones de error como que el vehículo ya ha recorrido los 6000 km designados y es de estricto cumplimiento el periodo de mantenimiento. También que a la hora de almacenar los datos relacionados con los choferes y lo usuarios el carnet de identidad solo contenga 11 dígitos y todos números.

En las pruebas de contenido se pudo comprobar que los cálculos realizados por el sistema son correctos a la hora de mostrar los reportes necesarios a cada departamento. Además, durante las pruebas de usabilidad los usuarios que trabajaron con el sistema comprendieron las ideas que se trataron de transmitir mediante las palabras usadas en los distintos tipos de mensajes ya fuesen de error o de información.

3.3 Requerimientos de Instalación

Para instalar cada uno de las componentes del sistema se debe tener en cuenta cada uno de los requisitos específicos de cada uno de ellos.

3.3.1 Instalación de la BD

Para crear la BD y lograr un correcto funcionamiento de la misma se deben seguir los siguientes pasos:

1. Tener un servidor de BD PostgreSQL 9.2.
2. Crear una nueva BD en el servidor.
3. Restaurar en la BD creada en el paso 2 usando la opción de Restore con un archivo con extensión (backup).

3.3.2 Requerimientos e Instalación del “CAAE.jar”

Antes de pasar a la ejecución del CAAE.exe, se deben revisar los siguientes requisitos:

1. Procesador Pentium o superior.
2. Al menos 256 Megabytes de memoria RAM.
3. Microsoft Windows XP o superior.
4. Máquina Virtual de java jre8.
5. PostgreSQL 9.2.

Si todos los requerimientos anteriores están verificados se puede pasar a la ejecución del “CAAE.exe”.

3.4 Descripción del sistema

El sistema cuenta con cuatro actores los cuales son: *“Administrador”*, *“Técnico de Puesto de Mando”*, *“Usuario de Puesto de Mando”* y el *“Subdirector”*, donde cada uno de ellos tiene privilegios y funciones diferentes en función de su usuario y contraseña.

3.4.1 Pasos iniciales

En la fase de despliegue del sistema el mismo por si solo crea un archivo de configuración en donde guarda un usuario y una contraseña para el *“Administrador”* que por defecto es *root* y ok. Después al *“Administrador”* autenticarse con esos parámetros como se muestra en la Figura 26: Ventana de Autenticación. aparece la vista de administrador desde la cual podrá completar el archivo de configuración y además definir los usuarios y permisos sobre la base de datos. Figura 27: Vista del “Administrador”.



Figura 26: Ventana de Autentificación.

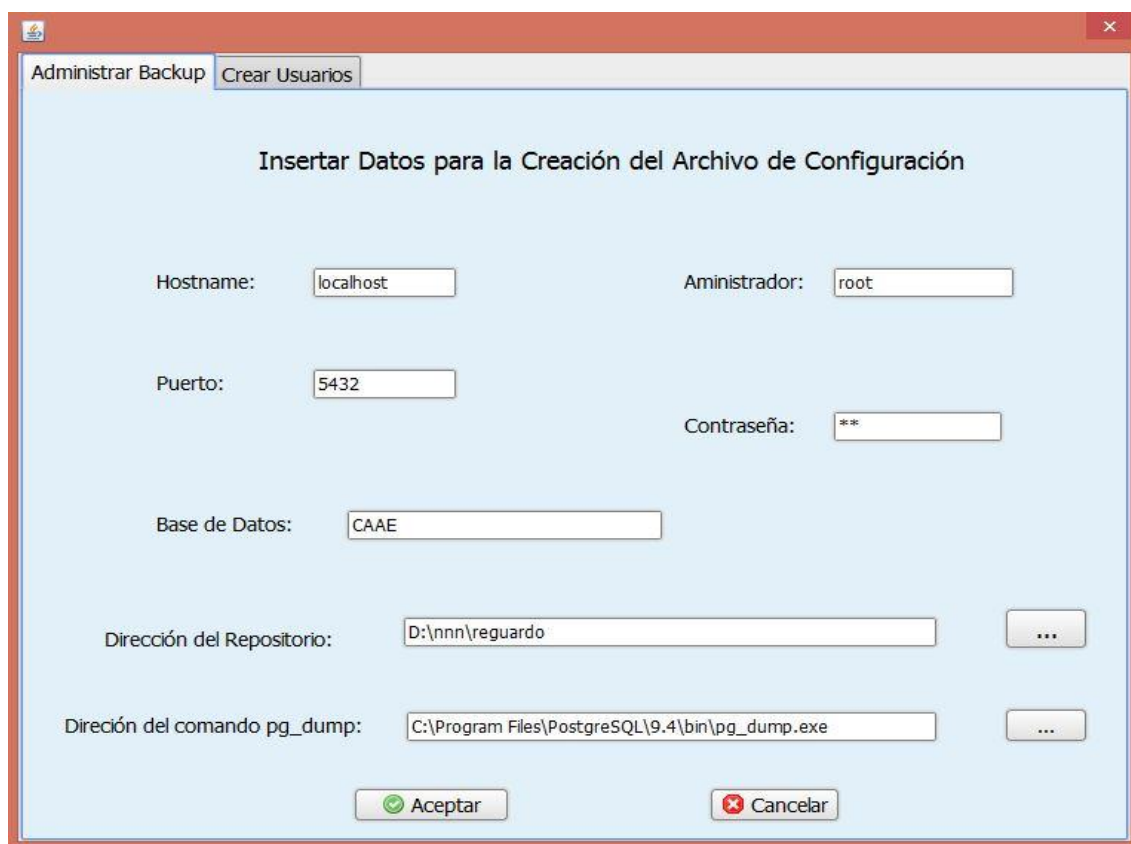


Figura 27: Vista del “Administrador”.

En el archivo de configuración el “**Administrador**” guardara datos como:

Nombre da la base de datos con que se trabajara.

Dirección del comando pg_dump.

Dirección donde se guardarán los repositorios.

En la Figura 28: Vista de Registro de Usuarios. se muestran los datos relativos que se deben almacenar de cualquier usuario que se le otorguen permisos en el sistema.

Figura 28: Vista de Registro de Usuarios.

La vista del actor “**Subdirector**” es la de mayor importancia, en ella se encuentran la mayor cantidad de funciones y métodos del sistema. Entre ellos resaltan la realización de reportes, la modificación de las Tarjetas de Circulación y la creación de los resguardos de las bases de datos para poder tener los datos históricos almacenados sin que estos afecten el rendimiento el sistema.

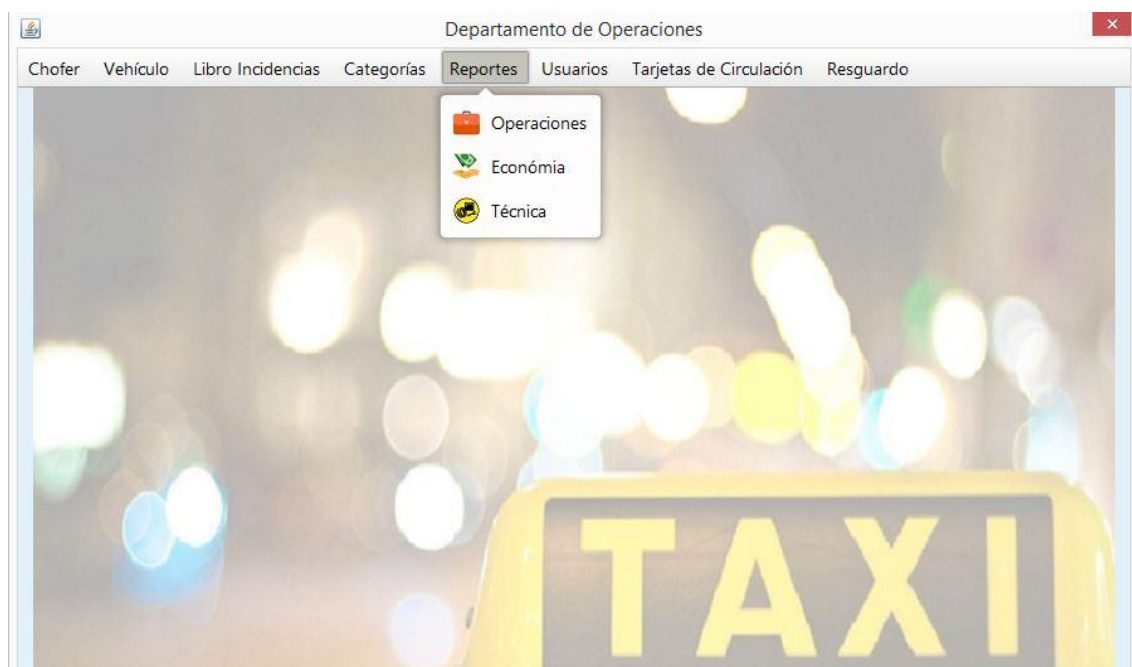


Figura 29: Vista del “Subdirector”.

Como se puede apreciar en la figura anterior los reportes se encuentran divididos por departamentos y dentro de ellos los distintos tipos de reportes pertenecientes al mismo. Por ejemplos en el departamento de operaciones encontramos seis tipos de reportes los cuales son:

- Recaudación.
- Resultados
- Tarjetas de Circulación Recibidas.
- Informe de Explotación.
- Informe de Explotación Resumen.
- Comportamiento del Consumo.

Para la realización de los mismos se necesita la fecha inicial y final del periodo en que se desea realizar y también si es de un chofer o de un vehículo.

Gestionar Reportes

Escojer fechas de los Reportes

Inicial: Final:

Reportes por

☐ Chofer

☐ Vehículo


Tipos de Reportes

☐ Recaudación ☐ Tarjetas de Circulación Recibidas ☐ Resultados

☐ Informe de Explotación ☐ Informe Explotación Resumen ☐ Comportamiento de Consumo

Figura 30: Vista de Creación de Reportes.

Otro de los métodos fundamentales es el de la creación de resguardos ya que dentro de los requisitos funcionales planteados con anterioridad existe uno referente a la seguridad de los datos almacenados en el sistema. Para cumplir con lo planteado se crea un método de resguardo el cual será ejecutado por el actor (Subdirector de Operaciones) cada vez que este lo estime conveniente. Este método creara un resguardo de la base de datos en la dirección especificada en el archivo de configuración por el “Administrador”.



The image shows a Windows-style dialog box titled "Datos del Resguardo". It has a light blue background. At the top, the title "Datos del Resguardo" is centered in a large black font. Below the title, there are two input fields. The first is labeled "Nombre:" and is empty. The second is labeled "Dirección:" and contains the text "D:\nnn". Below these fields is a section titled "Creación de Resguardo" which contains a progress bar. The progress bar is currently at 0%. At the bottom of the dialog, there are two buttons: "Crear" and "Cancelar".

Figura 31: Vista de Creación de Resguardo.

Como se muestra en la Figura 32: Método Resguardo. el método resguardo dentro del paquete “caae” utiliza variables globales que fueron llenadas al iniciar la sección por el método leer, el cual tomo los datos del archivo de configuración. Estos datos fueron el hostname, puerto, nombre de la base de datos, la dirección donde se desea almacenar la copia y el camino en de la aplicación PostgreSQL *pg_dump.exe*, encargado de realizar el resguardo.

Para hacer esto se crea un “Thread” el cual llama al método *resguardo* () iniciándose y este es el encargado de ejecutar el proceso *pg_dump*.

```

75 private void resguardo(String nom) {
76     try {
77         barEns1.setValue(0);
78         barEns1.setString("Creando Resguardo...");
79         barEns1.setIndeterminate(true);
80         String path = dirRepo + "/" + nom + "." + "backup";
81         Runtime r = Runtime.getRuntime();
82         String user = "postgres";
83         ProcessBuilder pb;
84         Process p;
85         pb = new ProcessBuilder(Dirpg_dump, "-i", "-h",
86             "localhost", "-p", "5432", "-U", user, "-F", "c", "-b", "-v", "-f",
87             path, nombBD);
88         pb.environment().put("PGPASSWORD", "root");
89         pb.redirectErrorStream(true);
90         p = pb.start();
91         barEns1.setString("Terminado");
92         barEns1.setIndeterminate(false);
93         JOptionPane.showMessageDialog(null, "El resguardo se ha creado correctamente",
94             "Información", JOptionPane.INFORMATION_MESSAGE);
95         jDResguardo.setVisible(false);
96         jDOperaciones.setVisible(true);
97     } catch (Exception ex) {
98         JOptionPane.showMessageDialog(null, ex);
99     }
100 }

```

Figura 32: Método Resguardo.

3.4.2 Descripción del comando pg_dump

Es una aplicación cliente que es capaz de generar archivos de texto con comandos SQL representando un snapshot de la base de datos. Es una funcionalidad que nos brinda el sistema gesto de bases de datos PostgreSQL mediante la cual podemos obtener copias de las bases de datos y reconstruirlas posteriormente.

Sintaxis utilizada para la creación de resguardos:

C:\Archivos de programa\PostgreSQL\8.3\bin\pg_dump.exe -i -h localhost -p 5432 -U postgres -F c -b -v -f "C:\mibase.backup" mibase

Donde:

- *C:\Archivos de programa\PostgreSQL\8.3\bin\pg_dump.exe* dirección del comando.
- *-i* le indica que ignore la versión (entre el comando y la base de datos).
- *-h localhost* especifica el host del motor de base de datos (localhost en este caso).
- *-p 5432* es la indicación del puerto donde corre el servicio.
- *-U postgres* es para indicarle que usaremos el usuario Postgres.
- *-F* indica el formato de salida.
- *c* es la elección del formato (comprime en este caso).
- *-b* indica que vuelque los LO (large objects).
- *-v* verbose.

- *-f* es el archivo de salida.
- *"C:\mibase.backup"* es el nombre para el archivo de salida (con path).
- *mibase* es el nombre de la base a backupear.

La vista del “**Técnico de Puesto de Mando**” desplegada después de verificado los permisos del mismo en la BD como se puede apreciar en la Figura 33: Vista del “Técnico de Puesto de Mando”. este actor puede gestionar Tarjetas de Circulación despachadas por el departamento de Puesto de Mando o trabajar como un usuario más de ese departamento. Además, el mismo posee permiso para modificar o agregar usuarios de Puesto de Mando o como Técnico del mismo departamento.

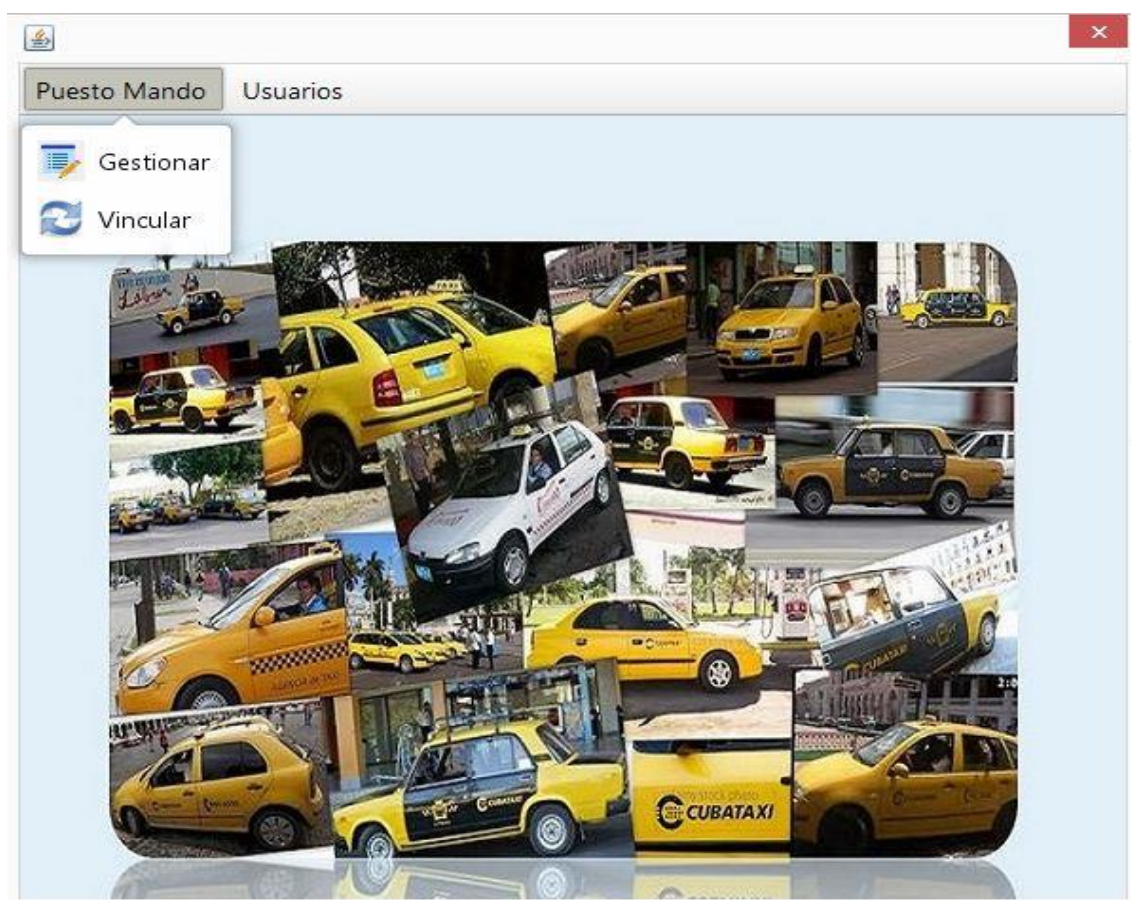


Figura 33: Vista del “Técnico de Puesto de Mando”.

El actor “**Usuario de Puesto de Mando**” solo puede insertar datos en las tablas de Tarjeta de Circulación y de Libro de Incidencias. Además, al concluir su turno de trabajo debe realizar un resumen del mismo mediante un reporte que brinda el sistema. En la Figura 34: Vista del “Usuario de puesto de Mando”. se pudo apreciar la vista de este actor y en la Figura 35: Resumen del Turno. como quedaría un resumen de un turno de trabajo.

Puesto de Mando

Expedir Tarjeta de Circulación Liquidar tarjeta de Circulación Libro de Incidencias Resumen del Turno

Datos de la Tarjeta de Circulación

Folio: Chofer:

Carro: Chapa: Marca: KM Disponible:

Odometro inicial: Cantidad combustible: Número de la Tarjeta:

Turno desde : Hasta : Tipo de Servicio : ☒ Contrato ☐ Piquera ☐ Recogida
hh:mm:ss hh:mm:ss

Fecha de Salida: Hora de salida: Expide:
hh:mm:ss

Figura 34: Vista del "Usuario de puesto de Mando".

Resumen del Turno

CubaTaxi

Resumen del Turno

Empresa provincial de transporte de Villa Clara Departamento de Operaciones

Folio	Nombre Chofer	Carro	Recaudado
2	Claudia Peña Dias	1-11	1232.0
3	Osniel Torres Garrido	1-21	231.0

Total General: 1463.0

Solicitado por: Claudia Peña Dias

Figura 35: Resumen del Turno.

3.5 Conclusiones Parciales

En el transcurso de este capítulo se aplicaron pruebas de integración del sistema con la base de datos, además también se aplicaron pruebas de usabilidad, funcionalidad y de contenido obteniendo resultados satisfactorios en el cumplimiento de los objetivos con los que se inició el proyecto. Se determinaron los requerimientos para la instalación y uso del sistema. Se diseñó una interfaz para la manipulación de la base de datos creada lográndose así un sistema de información que permite llevar a cabo el manejo y almacenamiento de los departamentos de Operaciones y Puesto de Mando de la empresa de CubaTaxi.

CONCLUSIONES

Al concluir el trabajo de diploma Sistema para el control de servicios de alquiler de autos estatales:

1. Se analizaron las principales características de los SI, haciendo énfasis en los operacionales y los de apoyo de toma de decisiones.
2. Se diseñó una base de datos relacional orientada a los requerimientos de los usuarios identificados.
3. Se concibió una interfaz para la manipulación de la base de datos creada permitiendo llevar a cabo el trabajo de almacenamiento y control de las actividades de los servicios de alquiler de autos estatales.
4. Se confeccionaron reportes con vista a la toma de decisiones por parte del subdirector.
5. Se diseñaron e implantaron una serie de pruebas que permitieron la validación del sistema de información propuesto.

.

RECOMENDACIONES

En la continuación del software presentado es importante se consideren las siguientes recomendaciones:

1. Extender el sistema a los demás departamentos de la empresa de CubaTaxi.
2. Implementación de las nuevas funcionalidades requeridas por los nuevos departamentos.
3. Implementar un método de restauración de backup para que el subdirector pueda trabajar con BD anteriores sin afectar a los demás departamentos.

REFERENCIAS BIBLIOGRÁFICAS

- Booch, G. et al., 1999. *El lenguaje unificado de modelado*, Addison-Wesley.
- Canchala, A., 2004. UML, ejemplo sencillo sobre Modelado de un Proyecto.
- Chaves, M.A., 2011. La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. *InterSedes*, 6(10). Available at:
<http://revistas.ucr.ac.cr/index.php/intersedes/article/view/790>.
- Chen, P.P.-S., 1994. The entity-relationship model—toward a unified view of data. In *Readings in database systems (2nd ed.)*. Morgan Kaufmann Publishers Inc., pp. 741–754.
- Deitel, P.J., 2004. *Como programar en Java*, Pearson Educación.
- DER, D.E.-R., Diseño conceptual de la base de datos.
- Figuerola, M.C.M.A.A., 2009. MeISE: Metodología de ingeniería de software educativo. *Revista Internacional Internacional Internacional de Educación en Ingeniería Educación en Ingeniería ISSN*, 1940, p.1116.
- González, Y.D. & Romero, Y.F., 2012. Patrón Modelo-Vista-Controlador. *Revista Telem@tica*, 11(1), pp.47–57.
- Gosling, J. & Arnold, K., 2001. El lenguaje de programación Java. *Estados Unidos*, Editorial Addison Wesley.
- Groussard, T., 2012. *JAVA 7: Los fundamentos del lenguaje Java*, Ediciones Eni.
- Gutiérrez, J.J. et al., 2006. Modelos de pruebas para pruebas del sistema. *Taller de Desarrollo de Software Dirigido por Modelos. XIII Jornadas sobre Ingeniería del Software y Bases de Datos JISBD*.
- Inmon, W., 2005. Building the data warehouse. *John wiley & sons*.
- Jacobson, I., Booch, G. & Rumbaugh, J., 2000. *El proceso unificado de desarrollo de software*, Addison Wesley Reading.
- Karen, C. & Lares, E.D., 2000. Sistemas de Información para los Negocios. Un

- enfoque de toma de decisiones. *McGraw-Hill. México.*
- Larman, C., 1999. UML y Patrones. Introducción al análisis y diseño orientado a objetos. Ed.
- Laudon, J.P., 2004. *Sistemas de información gerencial: administración de la empresa digital*, Pearson Educación.
- Laudon, K.C., Laudon, J.P. & Rodríguez, J.R., 1996. *Administración de los Sistemas de Información: Organización y tecnología*, Prentice Hall Hispanoamericana.
- López, C., Marticorena, R. & Martín, D.H., 2005. Pruebas de caja negra: una experiencia real en laboratorio. *Actas de las XI Jornadas de Enseñanza Universitaria de Informática, Jenui 2005*, pp.189–196.
- Martínez, R., 2010. Sobre PostgreSQL. *PostgreSQL-es*. Available at: http://www.postgresql.org.es/sobre_postgresql.
- Mendoza González, G., 2008. Herramienta de Desarrollo Netbeans.
- Montilva, J., 1999. Desarrollo de sistemas de información. *Universidad de Los Andes*.
- Narváez Coello, J.I., 2014. Guía de las mejores prácticas administrativas, seguridad y alta disponibilidad, caso de estudio: PostgreSQL.
- Nielsen, J. & Loranger, H., 2006. *Usabilidad. Prioridad en el diseño web*,
- Peña Ayala, A., 2006. Ingeniería de Software: una guía para crear Sistemas de Información. *México DF (México): Instituto Politécnico Nacional*.
- Prieto, A. & Martínez, M., 2004. Sistemas de información en las organizaciones: una alternativa para mejorar la productividad gerencial en las pequeñas y medianas empresas. *Revista de Ciencias Sociales*, 10, pp.322–337.
- Ross, M. & Kimball, R., 2013. *The data warehouse toolkit: the complete guide to dimensional modeling*, Wiley.
- Sánchez, J., 2003. Java 2 incluye Swing, Threads, programación en red, JDBC y JSP / Servlets.

- Siddiqui, B., 2010. *JasperReports 3.6 Development Cookbook*,
- Torres, P.L., 2004. Desarrollo de Software Orientado a Objeto usando UML.
Universidad Politecnica de Valencia (UPV)–España.
- Trasobares, A.H., 2003. Los sistemas de información: evolución y desarrollo. *Proyecto social: Revista de relaciones laborales*, (10), pp.149–165.
- Yagiüe, A. & Garbajosa, J., 2009. Comparativa práctica de las pruebas en entornos tradicionales y ágiles. *REICIS. Revista Española de Innovación, Calidad e Ingeniería del Software*, 5(4), pp.19–32.
- Zapata, C.M., Gelbukh, A. & Arango, F., 2006. UN-Lencep: Obtención automática de diagramas UML a partir de un lenguaje controlado. *Memorias del VII Encuentro Nacional de Computación ENC'06*, pp.254–259.