



UNIVERSIDAD DEL AZUAY

**FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN
ESCUELA DE INGENIERÍA DE SISTEMAS**

**“Modelo para gestión de entregas por rutas óptimas,
mediante GPS y conexión celular.”**

Tesis previa a la obtención del título
de Ingeniero de Sistemas

AUTOR:

Nadia Verónica López Villalba

DIRECTOR:

Ing. Marcos Orellana

CUENCA, ECUADOR

2011

DEDICATORIA

Dedico este trabajo de Tesis a mi familia de la cual aprendí que las mejores cosas llegan con esfuerzo, que para surgir se necesita valor y constancia y sobre todo, que con amor se puede disfrutar plenamente la vida.

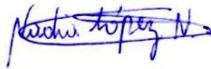
Con amor para mi mamá Elisita, mi papá Fernando, mi sobrinito Daniel y mi hermana Fernanda.

AGRADECIMIENTO

Agradezco a todas aquellas personas que de una u otra manera apoyaron a la consecución de este trabajo y de manera especial a mi tutor Ing. Marcos Orellana quien aportó con su conocimiento para que este trabajo de tesis cubriera todas las expectativas.

A su vez, extiendo un agradecimiento al personal y dueño de la compañía Madeco, quienes siempre fueron atentos y serviciales para con mi persona.

Las ideas y opiniones vertidas
en esta tesis están bajo la
responsabilidad de su autora.



Nadia Verónica López Villalba

ÍNDICE DE CONTENIDOS

DEDICATORIA	II
AGRADECIMIENTO	III
DOCUMENTO DE RESPONSABILIDAD	IV
ÍNDICE DE CONTENIDOS	V
ÍNDICE DE ILUSTRACIONES.....	XI
ÍNDICE DE TABLAS	XIV
RESUMEN.....	XV
INTRODUCCIÓN	1

CAPITULO I

RECOLECCIÓN, EVALUACIÓN, DEPURACIÓN Y LEVANTAMIENTO DE INFORMACIÓN REQUERIDA PARA EL DESARROLLO DEL MODELO PARA GESTIÓN DE ENTREGAS POR RUTAS ÓPTIMAS, MEDIANTE GPS Y CONEXIÓN CELULAR.	2
1.1. INTRODUCCIÓN	2
1.2. MADECO	2
1.3. GESTIÓN DE ENTREGAS	3
1.3.1. Definición.....	3
1.3.2. Los problemas de rutas de vehículos (VRP).....	4
1.3.3. La red de transporte.....	4
1.3.4. Los clientes y su servicio	5
1.3.5. Los almacenes o depósitos	5
1.3.6. La flota de vehículos	6
1.3.7. Las rutas	6
1.4. LA INTEGRACIÓN LOGÍSTICA.....	7
1.4.1.El Sistema de Información Geográfico.....	8
1.4.2. Base de datos Postgres	8
1.4.3. MapServer	9
1.4.4. GPS	9
1.4.5. Prolog	9
1.4.6. General Packet Radio Service (GPRS)	9
1.5. GLOBAL POSITIONING SYSTEM (GPS)	9

1.5.1. Historia.....	10
1.5.2. Funcionamiento.....	11
1.5.2.1. Los satélites.....	12
1.5.2.2. Las estaciones de seguimiento.....	13
1.5.2.3. Señales GPS.....	13
1.6. ERRORES EN SEÑAL GPS.....	14
1.6.1. Error inducido por la atmósfera.....	14
1.6.2. Error multitrayectoria.....	14
1.7. RASTREO Y LOCALIZACIÓN.....	15
1.7.1. Global System for Mobile Communication (GSM).....	15
1.7.2. General Packet Radio Service (GPRS).....	15
1.7.3. GPS Vehicle Tracker.....	15
1.8. INTELIGENCIA ARTIFICIAL.....	16
1.8.1. Historia I.A.....	16
1.8.2. Problema de planificación.....	19
1.8.3. Planificación heurística.....	20
1.8.4. Redes de tareas jerárquicas.....	21
1.8.5. Planificación de rutas.....	23

CAPITULO II

ESTUDIO Y REVISIÓN DE LAS HERRAMIENTAS PARA EL DESARROLLO..... 24

2.1. INTRODUCCIÓN.....	24
2.2. SERVICIO WEB MAP SERVICE (WMS).....	24
2.2.1. Funcionamiento.....	24
2.2.2. Operaciones.....	26
2.3. MAPSERVER.....	27
2.3.1. Estructura.....	27
2.3.2. El archivo de configuración “.map”.....	28
2.4. OPENLAYERS.....	29
2.5. LENGUAJE DE PROGRAMACIÓN PROLOG.....	29
2.5.1. Estructura.....	30
2.5.2. Base de Conocimiento.....	30
2.5.3. Inferencia.....	30

2.5.4. JPL	31
2.6. GPS VEHICLE TRACKER (TRAX S6).....	32
2.6.1. Características	32
2.6.2. Funcionamiento.....	34
2.6.3. Protocolo de Comunicación GAP	35
2.6.3.1. Descripción del protocolo GAP	36
2.6.3.2. Tipos de mensajes	36
2.6.3.3. Identificador de dispositivo.....	36
2.6.3.4. Identificador o número de mensaje	36
2.6.3.5. Destino o redirección de respuesta.....	37
2.6.4. Reportes	39
Reporte de Posición extendida, estados varios, fecha y hora del Equipo	39
2.6.5. Aplicaciones	41
2.7. COMUNICACIONES MÓVILES.....	42
2.7.1. GSM.....	42
2.7.2. GPRS.....	43
2.7.3. El sistema de mensajes cortos (SMS)	44
2.7.3.1. Estándar de comunicación AT para módems GSM.....	45
2.7.3.2. El formato de envío de datos PDU.....	47
2.7.4. SAMSUNG SGH-E256	48
2.8. JPCAP	49

CAPITULO III

ANÁLISIS Y DISEÑO DE LOS PROCEDIMIENTOS QUE COMPRENDERÁN EL DESARROLLO DEL MODELO PARA GESTIÓN DE ENTREGAS POR RUTAS ÓPTIMAS, MEDIANTE GPS Y CONEXIÓN CELULAR, ASÍ COMO LA RECOLECCIÓN, LEVANTAMIENTO Y DEPURACIÓN DE INFORMACIÓN GEOGRÁFICA PARA CREACIÓN DE RUTAS	51
3.1. INTRODUCCIÓN	51
3.2. DEFINICIÓN Y ANÁLISIS DEL SISTEMA.....	51
3.2.1. Identificación de los usuarios participantes y finales.....	51
3.2.2. Establecimiento de requisitos:	52
Requisitos funcionales	52

Requisitos no funcionales	53
3.2.3. Herramientas a utilizar	54
3.2.4. Especificación de casos de uso:	56
3.2.4.1. Paquetes principales de casos de uso	57
3.2.4.2. Paquetes de casos de uso.....	58
3.2.4.3. Listado de Casos de Uso	59
Caso de uso - Configuración de Procesos	61
Caso de uso - Definir Empresa.....	63
Caso de uso - Definir Usuario.....	66
Caso de uso - Definir Marca	71
Caso de uso - Definir Modelo	74
Caso de uso - Definir Artículo	77
Caso de uso - Definir Cliente.....	80
Caso de uso - Definir Empleado	84
Caso de uso - Definir GPS	87
Caso de uso - Definir Calles.....	91
Caso de uso - Generación Cartográfica.....	94
Caso de uso - Definir Vehículo.....	96
Caso de uso - Definir Bodega	99
Caso de uso - Realizar Expedición.....	103
Caso de uso - Definir Proveedor	104
Caso de uso - Definir Orden.....	108
Caso de uso - Definir Catálogo de Artículos	109
Caso de uso - Definir Ruta	112
Caso de uso - Definir Catálogo Consultas Ruteo.....	113
Caso de uso - Realizar Consulta.....	115
Caso de uso - Realizar Proceso	116
3.3. ELABORACIÓN DEL MODELO DE DATOS	117
3.4. DEFINICIÓN DE LAS INTERFAZ DEL SISTEMA.....	122
3.4.1. Especificación de principios generales de la interfaz	122
3.5. DEFINICIÓN DE LA ARQUITECTURA DEL SISTEMA	124
3.6.1. Cartografía base	126
3.6.2. Depuración de datos.....	133

CAPITULO IV

PROGRAMACIÓN DEL MODELO PARA GESTIÓN DE ENTREGAS POR RUTAS ÓPTIMAS, MEDIANTE GPS Y CONEXIÓN CELULAR 142

4.1. INTRODUCCIÓN 142

4.2. CREACIÓN DE LA BASE DE DATOS 142

4.3. CONFIGURACIÓN DEL GPS 143

1. Contratar el servicio de operadora para GPRS 143

2. Configuración del GPS 144

4.4 APLICACIÓN PARA LA CAPTURA DE PAQUETES DEL REPORTE ENVIADO POR GPS 147

1. Obtener la lista de las interfaces de red en la máquina 147

2. Abrir una interfaz de red 148

3. Captura de paquetes de la interfaz de red..... 149

4. Grabar en base de datos..... 150

4.5 APLICACIÓN PARA LA RECEPCIÓN DE ESTADO DE ORDEN ENTREGA – RECEPCIÓN AL SERVIDOR A TRAVÉS DE SMS 153

1. Agregar librería javax comm 154

2. Configurar el CNMI del celular Samsung 154

3. Establecer formato de mensaje preferido (+ CMGF)..... 154

4. Recepción de mensajes 155

4.6 ALGORITMO DE CÁLCULO DE RUTAS ÓPTIMAS 164

1. Creación de la base de conocimientos a utilizar 165

2. Creación del algoritmo..... 166

4.7 APLICACIÓN DE MONITOREO EN MAPSERVER..... 168

1. Elaboración del archivo .map..... 168

2. Definición de la configuración inicial página del servidor de mapas 168

3. Creación de la interfaz 170

4. Búsqueda..... 171

5. Monitoreo..... 171

4.8 MODELO DE GESTIÓN DE ENTREGAS POR RUTAS ÓPTIMAS 171

1. Creación de la interfaz de usuario..... 172

2. Definición de los botones de interacción con el usuario.....	172
3. Creación de los formularios	172
4. Geolocalización.....	172
5. Integración de Prolog con java.....	174
6. Ordenes de Entrega-Recepción	175
7. Gestión de entregas	175
8. Rastreo de vehículos	176
9. Actualización estado de la orden.....	176
10. Definición de Consulta.....	176
11. Ayuda	176
CONCLUSIONES	177
RECOMENDACIONES	179
GLOSARIO	180
BIBLIOGRAFÍA	184

ÍNDICE DE ILUSTRACIONES

Gráfico 1. Ejemplo de la solución a un problema básico VRP	7
Gráfico 2. Estructura y principales componentes del modelo.....	8
Gráfico 3. Representación del Sistema GPS	10
Gráfico 4. Ejemplo de Triangulación.....	13
Gráfico 5. Posibles errores en la señal del satélite	14
Gráfico 6. Rastreo de Vehículo con GPS.....	16
Gráfico 7. Esquema del Planificación.....	20
Gráfico 8. Funcionamiento del servicio WMS	25
Gráfico 9. Esquema de la estructura de una documento XML	26
Gráfico 10. Estructura principal de Mapserver	28
Gráfico 11. Estructura de un archivo .map	29
Gráfico 12. Envío de un mensaje SMS	44
Gráfico 13. Compresión de 7 bits a 8 bits.....	47
Gráfico 14. Samsung SGH-E256	48
Gráfico 15. Simbología para casos de uso	56
Gráfico 16. Simbología de Paquetes Principales de Casos de Uso.....	57
Gráfico 17. Casos de Uso.....	58
Gráfico 18. Configuración de Procesos	61
Gráfico 19. Definir empresa.....	63
Gráfico 20. Definir usuario	66
Gráfico 21. Definir marca	71
Gráfico 22. Definir modelo	74
Gráfico 23. Definir artículo.....	77
Gráfico 24. Definir cliente	80
Gráfico 25. Definir empleado	84
Gráfico 26. Definir GPS.....	87
Gráfico 27. Definir calles	91
Gráfico 28. Generación cartográfica.....	94
Gráfico 29. Definir vehículo	96
Gráfico 30. Definir bodega	99
Gráfico 31. Realizar expedición.....	103
Gráfico 32. Definir proveedor.....	104

Gráfico 33. Definir orden.....	108
Gráfico 34. Definir catálogo de artículos.....	109
Gráfico 35. Definir ruta.....	112
Gráfico 36. Definir catálogo consultas ruteo	113
Gráfico 37. Realizar consulta.....	115
Gráfico 38. Realizar proceso.....	116
Gráfico 39. Modelo de datos para el sistema de gestión de transporte y logística ..	117
Gráfico 40. Modelo de Gestión de Entregas	120
Gráfico 41. Diseño estándar de las pantallas del sistema.....	123
Gráfico 42. Diseño estándar del Servidor de Mapas.....	124
Gráfico 43. Mapa de la Ciudad de Cuenca, Datum Horizontal PSAD56 Z17S, Escala 1:1 000.....	128
Gráfico 44. Tabla de atributos de la capa Vías	129
Gráfico 45. Error detectado en la parroquia Gil Ramírez Dávalos	130
Gráfico 46. Error detectado en la parroquia San Sebastián	130
Gráfico 47. Error detectado en la parroquia San Blas.....	130
Gráfico 48. Mapa de la Ciudad de Cuenca, Datum Horizontal WGS84 Z17S, Escala 1:1 000.....	132
Gráfico 49. Tabla de Atributos de la capa temática Calles_postgres.....	134
Gráfico 50. Tabla de atributos de la capa temática Vías.....	135
Gráfico 51. Información gráfica.....	136
Gráfico 52. Tabla de atributos de la capa temática POINT_Vias	136
Gráfico 53. Archivo original	137
Gráfico 54. Archivo procesado	137
Gráfico 55. Generación de líneas en la nueva capa a partir de los nodos	138
Gráfico 56. Tabla de atributos de la capa TT_BLOCK_10k_CGSWGS84_Pos.....	138
Gráfico 57. Tabla de atributos de la capa temática tt_block_10k_CGS_WGS84_Postgres	139
Gráfico 58. Representación de la tabla TT_BLOCK, TT_POINT y áreas verdes...	139
Gráfico 59. Mapa del centro de la Ciudad de Cuenca.....	141
Gráfico 60. Estructura de Base creada	142
Gráfico 61. CommLog – Programa Terminal Serie y UDP para equipos TRAX145	
Gráfico 62. Pantalla captura de paquetes del reporte enviado por GPS.....	152
Gráfico 63. Resultado Network Interface	153

Gráfico 64. Interfaz de aplicación de recepción de mensajes escritos a través del celular SAMSUNG SGH-E256.....	161
Gráfico 65. Listado de Puertos COM.....	162
Gráfico 66. Opciones de Modem SAMSUNG SGH-E256.....	162
Gráfico 67. Formato de envío de mensajes	163
Gráfico 68. Lectura de mensaje en memoria	164
Gráfico 69. Visualización de Mapserver.....	170
Gráfico 70. Ejemplo de localización.....	173
Gráfico 71. Pantalla principal de Gestión de Entregas	175

ÍNDICE DE TABLAS

Tabla 1. Ejemplos de situaciones que disparan eventos y las acciones a tomar	35
Tabla 2. Tipos de mensajes	36
Tabla 3. Identificadores de mensajes específicos	37
Tabla 4. Destino o redirección de respuestas	38
Tabla 5. Listado de comandos PH (Q - R)	40
Tabla 6. Listado de comandos AT para envío y recepción de mensajes SMS	46
Tabla 7. Listado de Requisitos Funcionales.....	53
Tabla 8. Listado de Requisitos No Funcionales.....	54
Tabla 9. Listado de casos de Uso	60
Tabla 10. Parámetros de Configuración para guardar eventos LOG	146
Tabla 11. Parámetros del método JpcapCaptor.openDevice()	149
Tabla 12. Partes que componen un mensaje	156
Tabla 13. Formatos de Texto	156

RESUMEN

En un intento de vencer permanentemente al espacio con el factor tiempo en un mundo donde la movilidad plantea cuestiones cada vez más difíciles, y que requieren de una inexorable gestión de circuitos por medio de la logística. Planteamos una solución para llevar a cabo un manejo adecuado de la gestión de entregas con una mejor relación de calidad, precio y plazos óptimos.

Dentro de este contexto el transporte debe contribuir a afrontar los nuevos retos del mercado, incorporando nuevas y mejores tecnologías de forma tal que se entreguen mejores soportes a los procesos de análisis y a la gestión logística.

El presente proyecto de Tesis desarrolla un modelo para la gestión de entregas a través de Rastreo Vehicular y la Generación de Rutas Óptimas en tiempo real. Planteando, acciones que conlleven a una mejor gestión y a un ahorro de recursos.

INTRODUCCIÓN

La gestión del servicio de entrega de productos mediante vehículos y logística a menudo representan problemas que generan retrasos a las empresas, implicando un perjuicio a nivel de costos, tiempos y optimización de recursos.

La empresa ecuatoriana Madeco Cia. Ltda. es una de las empresas que se ha visto afectada debido a esta problemática. Madeco es una compañía dedicada a la venta al por mayor y menor de materiales de construcción; inicio sus actividades en el año de 1974 teniendo hasta la fecha una larga trayectoria. Actualmente el crecimiento de la empresa ha permitido que el servicio de entrega de su mercadería a los clientes se efectúe en cualquier sitio, por lo cual el modelo de gestión que aquí se propone pretende mejorar las rutas óptimas de entregas así como el manejo adecuado de los recursos.

Mediante la utilización de un dispositivo móvil GPS que cuenta con servicio de comunicación y recepción de información, se desarrollará un sistema que permita la inferencia para la obtención de rutas óptimas en tiempo real.

En este contexto, el presente trabajo de tesis propone un modelo demo de dicha gestión de entregas, se tomará como área de estudio el centro de la Ciudad de Cuenca, un equipo inteligente que integra la tecnología GPS y comunicación inalámbrica. De esta manera, la empresa Madeco podrá brindar un mejor servicio al momento de entregar su mercadería, además de que podrá rastrear el trayecto sus vehículos en tiempo real.

CAPÍTULO I

RECOLECCIÓN, EVALUACIÓN, DEPURACIÓN Y LEVANTAMIENTO DE INFORMACIÓN REQUERIDA PARA EL DESARROLLO DEL MODELO PARA GESTIÓN DE ENTREGAS POR RUTAS ÓPTIMAS, MEDIANTE GPS Y CONEXIÓN CELULAR.

1.1. INTRODUCCIÓN

La recolección, evaluación, depuración y levantamiento de información es sumamente importante, ya que, forma la base del desarrollo del modelo para gestión de entregas por rutas óptimas. En este capítulo se determinarán los componentes necesarios para realizar este proyecto para la Empresa Madeco.

1.2. MADECO

Madeco Cia. Ltda. es una empresa ecuatoriana dedicada a la venta al por mayor y menor de materiales de construcción. Inició sus actividades en Agosto de 1974, con su primer local en la calle Gran Colombia 2-32 y Manuel Vega de la ciudad de Cuenca.

Gracias a la confianza de clientes y proveedores fue cimentándose esta institución, de tal manera que en el año 1978 construye su primer local propio en la calle González Suárez 5-101 de esta misma ciudad.

En el año 1985, se construye un segundo local, con el nombre de Alherta y está ubicado en la calle Guapondelig y González Suárez.

En el año 1990 se construye y habilita una bodega, ubicada en la Av. González Suárez 5-125. Paralelamente a esto, se decide cambiar la razón social de los negocios, los mismos que de ser unifamiliar

El 21 de Julio del año 2000 abrieron el local en la Av. Los Andes y González Suárez. Y recientemente, su último local se encuentra en el sector Chilcapamaba, el valle (Cuenca-Ecuador). (Madeco Materiales de Construcción)

Hoy en día el crecimiento de la Empresa genera una mayor entrega de mercancía a sus clientes en diferentes ubicaciones. La transportación de cualquier tipo de mercancía es una de las actividades de más importe tanto a nivel social como económico.

Una de las formas de analizar esta importancia es a través de los datos económicos, para ello tenemos como ejemplo el coste asociado al transporte de mercancías en EEUU supone 47 billones de dólares al año y en Europa 168, según King and Mast (1997).

1.3. GESTIÓN DE ENTREGAS

1.3.1. Definición

La gestión de entregas (transportación) consiste en el intento de vencer permanentemente al espacio con el factor tiempo. Requiere una organización rigurosa de la gestión de los circuitos por medio de la logística.

La logística tiene como finalidad la obtención del producto deseado en el lugar adecuado con la mejor relación calidad/precio y dentro de unos plazos óptimos de entrega.

La logística también implica la asociación de tareas complejas en un mundo donde la movilidad plantea cuestiones cada vez más difíciles, y suscita problemas que superan en muchos sentidos la noción de transporte como tal. Así, el transporte debe contribuir a afrontar los nuevos retos imperativos del mercado. (Sánchez Silva, 2006)

Una serie de adelantos tecnológicos sirven de soporte a los procesos de análisis y gestión logística, casi todos ellos asociados a los avances científicos técnicos de la informática.

En los últimos años se han desarrollado sistemas para la generación, análisis y representación de información georeferenciadas de gran utilidad en los estudios del transporte, la industria, la agricultura, el

comercio, etc. Son los denominados SIG (Sistemas de Información Geográfica) cuya utilización en sistemas de logística es prometedora. El análisis conjunto, derivado de la combinación de información gráfica en forma de mapas (información espacial) y atributos asociados (información no espacial), da a los SIG su particular potencial de aplicación al sector transporte y sus anexos con el resto de la infraestructura logística.

1.3.2. Los problemas de rutas de vehículos (VRP)

Los problemas de rutas de vehículos o de distribución física de mercancías desde almacenes a clientes aparecen en la literatura científica como Vehicle Routing Problems, más comúnmente como VRP. También se puede encontrar, aunque en menor medida, referencias como Vehicle Scheduling Problems. En términos generales, un problema de rutas de vehículos consiste en determinar las rutas de un conjunto (o flota) de vehículos que deben iniciar un recorrido (y finalizarlo) en los almacenes (o depósitos) para atender la demanda de servicio de un conjunto disperso de clientes sobre una red.

Las diferentes características de los clientes, la demanda, los almacenes y los vehículos, así como de las restricciones operativas sobre las rutas, horarios, etc. dan lugar a gran número de variantes del problema. (Rodríguez, Integración de un sig con modelos de cálculo y optimización de rutas de vehículos cvrp y software de gestión de flotas, 2007)

1.3.3. La red de transporte

La red de carreteras o servicio utilizada para el transporte de bienes, se describe generalmente como un grafo donde los arcos representan los segmentos o secciones de las vías, y los vértices corresponden a las uniones o nodos de la red. En algunos casos los clientes o los depósitos pueden estar situados en dichos nodos, mientras que en otros casos pueden estar localizados en un arco del grafo. Los arcos (y por consiguiente el grafo) puede ser dirigidos o no dirigidos, dependiendo de

si pueden ser circulados en un único sentido o en ambos (por ejemplo, calles de una única dirección o de ambos sentidos de circulación).

Cada arco tendrá asociado un coste que puede representar su longitud en distancia, el tiempo de viaje, que tan transitado es, o el coste monetario del mismo.

1.3.4. Los clientes y su servicio

Cada cliente tendrá cierta necesidad de servicio o demanda que deberá ser atendida por algún vehículo. Es común que la demanda sea la necesidad de un conjunto de productos que ocupan volumen y peso en los vehículos, y como la capacidad de transporte del vehículo es limitada, es usual que un mismo vehículo no pueda satisfacer la demanda de todos los clientes. Un mismo vehículo podría (en teoría) visitar a todos los clientes.

En muchas ocasiones se trata de visitar al cliente exactamente una vez, sin embargo, en otros casos puede aceptarse que su demanda pueda ser atendida de manera fragmentada o por vehículos diferentes. Los clientes podrían tener restricciones de horario, en forma de intervalos o ventanas de tiempo dentro de las cuales se debe atender su servicio. También podría tenerse en cuenta no sólo el tiempo de recorrido por la red, sino el tiempo de servicio al cliente (carga y descarga).

También podrían existir restricciones de asociación entre vehículos y clientes, de manera que determinados clientes sólo puedan ser atendidos por determinados vehículos (por ejemplo, vehículos grandes y pesados que no pueden circular por calles estrechas o el centro urbano).

1.3.5. Los almacenes o depósitos

Tanto los productos a transportar como los vehículos, suelen estar localizados en los depósitos. Las rutas dan comienzo y/o finalizan en dichos depósitos.

1.3.6. La flota de vehículos

Los vehículos se definen por un conjunto de atributos, como su capacidad de carga en peso, en volumen, sus costes asociados, etc. En un vehículo se pueden transportar diferentes tipos de productos o uno sólo. En la utilización de un vehículo se incurre en unos costes fijos por uso, y variables en función del tiempo, distancia u otros parámetros.

El número de vehículos disponibles de una flota puede ser un dato conocido o una variable de decisión. Es común que el objetivo sea intentar utilizar la menor cantidad de vehículos y en segundo lugar minimizar la distancia o tiempo empleado de su ruta.

1.3.7. Las rutas

Los problemas de rutas de vehículos tratan por tanto de determinar la ruta o rutas para cada uno de los vehículos de la flota cumpliendo con todo el conjunto de restricciones e intentando alcanzar los objetivos propuestos. Por ejemplo: minimizar los costes fijos, minimizar los costes totales, minimizar el tiempo total de transporte y/o la distancia total recorrida, minimizar las esperas, maximizar la función de utilidad del cliente, o su beneficio y satisfacción. Un vehículo sólo recorrerá una ruta en el período de planificación, pero también se pueden encontrar modelos en los que un mismo vehículo podría participar de más de una ruta.

El Gráfico 1 representa un ejemplo típico de solución a un problema de rutas. En el gráfico se puede observar 4 rutas diferentes con origen y destino final en el depósito central. Los arcos de la ruta solución deben ser necesariamente arcos de la red de transporte. (Rodríguez, Integración de un sig con modelos de cálculo y optimización de rutas de vehículos cvrp y software de gestión de flotas, 2007)

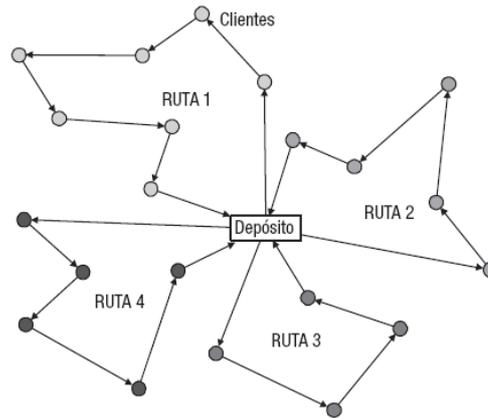


Gráfico 1. Ejemplo de la solución a un problema básico VRP

1.4. LA INTEGRACIÓN LOGÍSTICA

El desarrollo que se presentará facilitará la resolución de problemas reales de flotas de vehículos, el cálculo de rutas y su gestión. En el desarrollo del modelo para gestión de entregas por rutas óptimas, mediante GPS y conexión celular se integra inteligentemente seis elementos: el sistema de información geográfica SIG, la base de datos Postgres con la información relativa a la gestión de entregas, MapServer (Servidor de mapas), la información del sistema logístico a través del algoritmo escrito en PROLOG el cual permite resolver los problemas de rutas para flotas de vehículos en tiempo real, GPS/GPRS y conexión celular.

El éxito en la gestión logística depende de la capacidad de integración (información y sistemas, clientes, recursos y decisiones, etc.). Se ha prestado especial interés a la integración necesaria para la optimización del transporte, la toma de decisiones y la gestión de entrega. En el Gráfico 2 se representa la estructura y principales componentes del modelo de gestión.

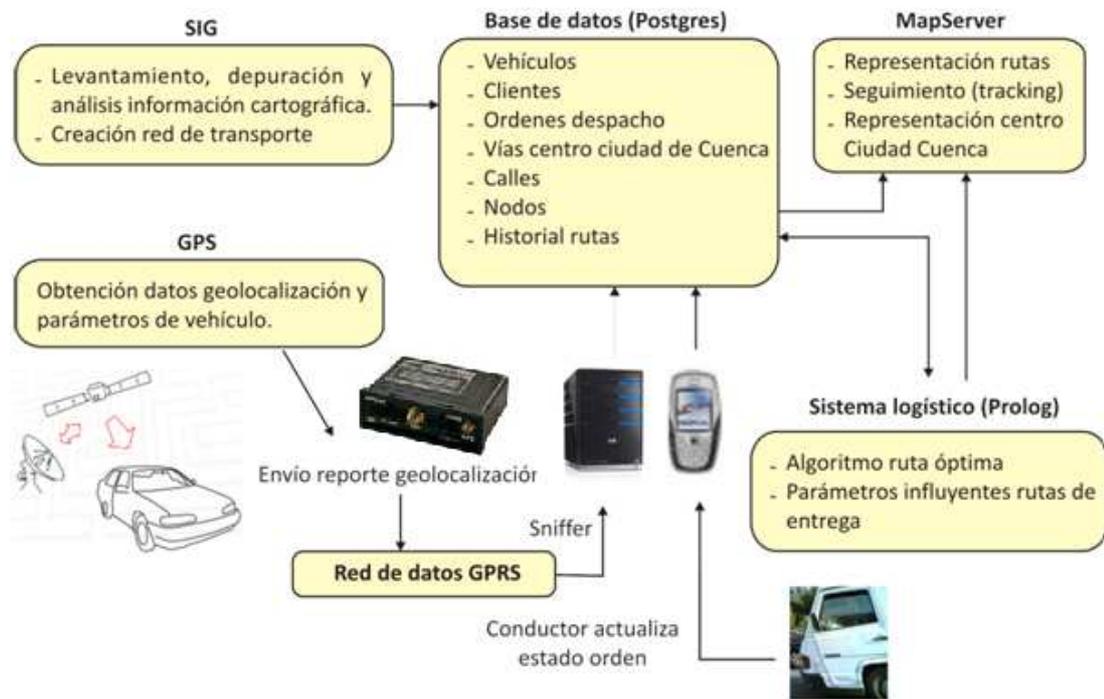


Gráfico 2. Estructura y principales componentes del modelo

A continuación se da una breve descripción de los componentes expuestos:

1.4.1. El Sistema de Información Geográfico (SIG)

El SIG es una colección organizada de datos geográficos que permite analizar y desplegar en todas sus formas la información geográficamente referenciada. SIG se utiliza para el análisis de la red de transporte, red de carreteras, sentido de circulación de las vías, tráfico, etc. Para el modelo de gestión de entregas utilizaremos el SIG para el levantamiento, depuración y análisis de la información. Y así finalmente crear la red de transporte.

1.4.2. Base de datos Postgres

En esta base de datos se almacenará la información necesaria para la representación y gestión de rutas óptimas.

1.4.3. MapServer

En el Servidor de mapas MapServer se representarán gráficamente el Centro de la Ciudad de Cuenca, así como las rutas.

1.4.4. GPS

Es un programa de navegación y posicionamiento basado en satélites. El mismo nos permitirá, además de determinar el posicionamiento exacto de un lugar en la tierra, controlar en tiempo real el desplazamiento de vehículos en las diferentes rutas del transporte.

1.4.5. Prolog

Prolog es un programa diseñado para ayudar a resolver problemas asociados con la inteligencia artificial. En nuestro caso, nos permitirá resolver la ruta más óptima.

1.4.6. General Packet Radio Service (GPRS)

Se utiliza para enviar paquetes de datos sobre la red GPS.

Para el modelado, resolución y análisis de este tipo de problemas, es necesario gestionar una enorme cantidad de información: datos sobre las características de la flota de vehículos, información geográfica, las restricciones, etc.

De esta manera se integra todo un conjunto de funciones que facilita la interacción con otros elementos de la cadena logística como por ejemplo:

- Cálculo y gestión de distancias, tiempos y costes de transporte.
- Seguimiento de vehículos (*GPS tracking*).
- Generación de mapas, mejora de la documentación logística.

1.5 GLOBAL POSITIONING SYSTEM (GPS)

Sistema de posicionamiento Global, consiste en toda una constelación de satélites que orbitan en la Tierra, los cuales son utilizados por los dispositivos de rastreo para obtener su posición. (CosmoTRACS, 2009)



Gráfico 3. Representación del Sistema GPS

1.5.1 Historia

En 1957, la antigua URSS lanzó el primer satélite hecho por el hombre: el Sputnik 1. Los científicos rápidamente se dieron cuenta primero de que, sirviéndose del Doppler Effect, se podía calcular la órbita de un satélite. Además, los satélites podían servir para calcular la posición de un receptor en tierra.

El ejército estadounidense estableció, a principios de los años 60, las bases del GPS moderno. La marina, la fuerza aérea y el ejército concibieron sus propios diseños e ideas y, en 1973 el gobierno norteamericano aprobó un diseño que incorporaba elementos de cada una de las propuestas. Así nació el NAVSTAR.

El primer satélite para el nuevo GPS del NAVSTAR se lanzó en 1974. Entre 1978 y 1985 se lanzaron otros 11 satélites de prueba. La constelación completa de 24, que hoy en día permite a su sistema de navegación disfrutar de cobertura GPS en todo el mundo, se completó en 1993.

Inicialmente el GPS se creó para darle únicamente un uso militar. Pero el presidente estadounidense Reagan propuso que el GPS se pusiera a disposición de los civiles para evitar que errores de navegación provocasen catástrofes.

Tras gastar unos 12.000 millones de dólares para desarrollar el sistema de navegación más preciso del mundo, el gobierno de Estados Unidos introdujo en el NAVSTAR una opción llamada disponibilidad selectiva (DS) que limitaba la precisión ofrecida a los usuarios civiles para asegurarse de que ningún enemigo ni grupo terrorista pudiese utilizar el GPS para fabricar armas de precisión.

Funcionaba introduciendo errores deliberados en los datos que transmitía a cada uno de los satélites. Los militares podían acceder al sistema exacto descifrando una segunda frecuencia protegida que se emitía al mismo tiempo.

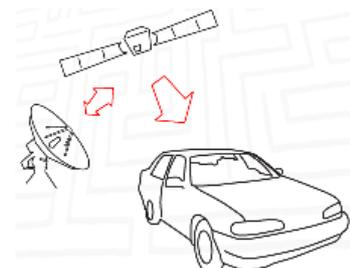
En el año 2000, el presidente Clinton anunció que la DS se deshabilitaría completamente. La razón era que las "evaluaciones de amenazas" llevadas a cabo por el gobierno estadounidense indicaban que eliminar la DS no afectaría prácticamente a la seguridad nacional. Sin embargo, en el mismo discurso, Clinton dijo que Estados Unidos podría “denegar selectivamente” las señales GPS en determinadas zonas cuando la seguridad nacional se viese amenazada.

El GPS ha evolucionado a partir de su objetivo primario hasta convertirse en un recurso insustituible para todo tipo de gente en todas las esferas de la vida. Desde los sectores del transporte y los servicios hasta marinos que surcan el océano. Desde la circulación relajada en vacaciones, localización de vehículos, hasta el viaje de la manera más rápida y eficiente de A hasta B. (TomTom, sistemas autonavegadores con GPS, 2010)

1.5.2 Funcionamiento

Un GPS consta de tres elementos:

- Satélites en el espacio
- Estaciones de seguimiento en tierra
- Receptor GPS



1.5.2.1 Los satélites

El sistema GPS posee 24 satélites orbitando alrededor de la Tierra repartidos en 6 planos orbitales, que envían señales de radio informando sobre su posición en órbita elevada, a 20.300 kilómetros/12.600 millas sobre nuestras cabezas. Al estar tan alto, la señal de cada satélite cubre una amplia zona de la superficie terrestre y además, sus órbitas están localizadas de forma que el receptor GPS que está en tierra recibe en todo momento señal de por lo menos cuatro satélites, la cantidad necesaria para determinar su ubicación.

El cuarto satélite comprueba la medición del tiempo de los otros tres satélites para asegurarse de que la información sobre su posición es lo más precisa posible. Ya que, como las distancias son tan grandes, si la medición del tiempo del viaje de una señal de satélite se equivocara en tan sólo 1/1.000as de segundo, se generaría un error en el cálculo de su posición de casi 300 km.

Por este motivo, nuestro receptor y satélite necesitan relojes que puedan sincronizarse hasta el nanosegundo. Cada uno de los satélites transporta un reloj atómico. Su receptor utiliza un reloj de cuarzo normal, mucho menos preciso, pero calcula el error del reloj de cuarzo mediante la medición realizada por un cuarto satélite.

Esto indica que si el dispositivo de navegación tuviera un reloj perfecto, todos los rangos de satélite se cruzarían en un único punto (su posición). Pero con relojes imperfectos, una cuarta medición desde un cuarto satélite, realizada como cotejo, no se cruza con los tres primeros rangos.

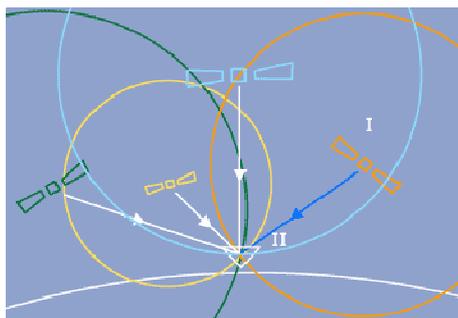


Gráfico 4. Ejemplo de Triangulación

I = cuarto satélite:
comprobación y corrección
de las mediciones de los
satélites

II = punto de intersección de
las señales de los satélites:
su posición

Un dispositivo de navegación calcula su distancia desde un satélite GPS a partir del tiempo que tarda la señal en viajar al receptor desde el satélite.

1.5.2.2 Las estaciones de seguimiento

Hay 5 estaciones de seguimiento: la estación matriz en Colorado Springs, EE.UU., y cuatro estaciones sin personal: una en Hawái y las otras tres en localizaciones remotas tan cerca al Ecuador como sea posible: Isla de Ascensión en el medio del Atlántico, Kwajalein en el Pacífico y el atolón de Diego García en el Océano Índico.

Estas cuatro estaciones sin personal reciben constantemente datos procedentes de los satélites, que ellas a su vez reenvían a la estación matriz, que “corrige” los datos y los devuelve a los satélites GPS.

1.5.2.3 Señales GPS

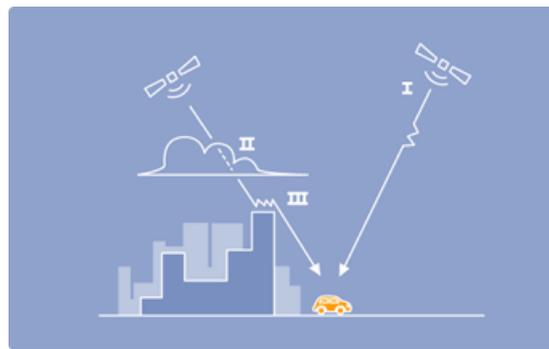
El receptor GPS capta señales de los satélites GPS para determinar dónde se encuentra usted. Cada satélite transmite señales de radio de baja energía a distintas frecuencias distintos usuarios. Las señales viajan a través de la llamada “línea de visión”. Esto significa que pasa a través de nubes, vidrio y plástico, pero normalmente no a través de objetos sólidos, como edificios.

El GPS puede determinar la localización a partir de los datos de cuatro satélites por "trilateración". La idea es básicamente la misma en la triangulación, pero sin utilizar ángulos.

1.6 ERRORES EN SEÑAL GPS

1.6.1 Error inducido por la atmósfera

Una señal GPS real atraviesa por partículas cargadas de la ionosfera y luego vapor de agua de la troposfera; estos elementos ralentizan su velocidad. Por ello, junto con su señal, los satélites también envían al dispositivo de navegación, la información sobre la "densidad" de la ionosfera. A partir de estos datos, el dispositivo de navegación puede corregir el error de la señal.



I = Interferencia por partículas de la ionosfera

II = Interferencia por vapor de agua de la troposfera

III = Interferencia por rebote en edificios y grandes construcciones

Gráfico 5. Posibles errores en la señal del satélite

Resultado = posibles errores en la señal de satélite

1.6.2 Error multitrayectoria

Las señales pueden rebotar en obstrucciones locales, como p.ej. montañas, antes de llegar a su dispositivo de navegación. Por este motivo, la señal tarda más en llegar a su receptor que si viajara en una trayectoria directa y el receptor, por lo tanto, supone que el satélite está más lejos de lo que está en realidad. Puede ocurrir también que una señal rebote en las obstrucciones locales mientras que otra viaje directamente a su dispositivo de navegación, por lo que las señales no estarían en sincronía con el dispositivo de navegación. Esto traería consigo lo que se conoce con el nombre de "error de multitrayectoria". Para fines de

navegación civiles esto no constituye ningún problema. (TomTom, sistemas autonavegadores con GPS, 2010)

1.7 RASTREO Y LOCALIZACIÓN

La precisión que tiene la tecnología de rastreo y localización GPS con GPRS es de aproximadamente 10 a 15 metros respecto a las posiciones reportadas por los dispositivos. (CosmoTRACS, 2009)

El rastreo GPS es la técnica de utilizar el sistema GPS para que un dispositivo instalado en un vehículo pueda obtener los datos de su Geo – localización y mediante una red de datos como GPRS los envíe contantemente a una plataforma AVL para que se pueda realizar el rastreo o seguimiento de los vehículos en tiempo real.

1.7.1 Global System for Mobile Communication (GSM)

Sistema Global para Comunicaciones Móviles, es el canal que las operadoras de telefonía celular utilizan para prestar el servicio de comunicaciones móviles. (CosmoTRACS, 2009)

1.7.2 General Packet Radio Service (GPRS)

Se utiliza para enviar paquetes de datos sobre la red GPS. El envío y la recepción de información es a través de una red de teléfonos móviles que cobra por cantidad de datos enviados, en lugar de por tiempo de conexión. En el caso de los dispositivos de rastreo GPS, estos lo utilizan para enviar sus reportes de posiciones a la plataforma implementada en los servidores. (CosmoTRACS, 2009)

1.7.3 GPS Vehicle Tracker

El dispositivo GPS de rastreo de vehículos combina el sistema de posicionamiento global GPS y el sistema de comunicación GSM/GPRS para informar la posición y estado del vehículo. Se utiliza para una infinidad de aplicaciones tales como manejo de flotas comerciales, logística de camiones de reparto, transporte de pasajeros, etc.



Gráfico 6. Rastreo de Vehículo con GPS

El GPS Vehicle tracker que se utilizará es el **TRAX S6** desarrollados por GTE, este equipo inteligente integra las tecnologías GPS y de comunicación inalámbrica para comunicarse con una base y enviarle reportes sobre su posición y estado en momentos o situaciones predeterminadas por el usuario.

1.8 INTELIGENCIA ARTIFICIAL

En las últimas décadas, investigadores de diversas ramas científicas han enfocado su trabajo al desarrollo e implementación de herramientas matemáticas de uso computacional eficiente para resolver problemas que surgen en áreas de trabajo como planificación, toma de decisiones, logística y predicción, estas herramientas están agrupadas en el campo del conocimiento generalmente conocido como Inteligencia Artificial (IA).

1.8.1 Historia I.A.

La Inteligencia Artificial puede definirse como "el arte de crear maquinas con capacidad de realizar funciones que realizadas por personas requieren inteligencia"[Rich y Knight 1991]. Según [Bellman 1978], la Inteligencia Artificial es "la automatización de actividades que vinculamos con procesos de pensamiento humano, actividades como la toma de decisiones, resolución de problemas, aprendizaje,...".

Esta disciplina comienza formalmente en 1956, cuando se acuñó el término Inteligencia Artificial (I.A.), aunque ya se investigaba en esta

área desde hacía varios años. En los inicios, los científicos perseguían un objetivo muy ambicioso: construir una maquina inteligente. Sin embargo, la I.A. ha resultado ser algo mucho más complejo de lo que muchos imaginaron en un principio. La I.A. se dividió en diversos campos, cada uno de ellos dedicado a distintas áreas del razonamiento aplicado a la resolución de problemas concretos. (Ortega & García, 2006)

Un problema de planificación consiste en encontrar una secuencia de acciones que permita, partiendo del estado actual del entorno, alcanzar un determinado estado objetivo.

El objetivo de un planificador consiste en explorar este estado de la forma más eficiente posible, o bien en reducir este espacio al máximo de manera que la búsqueda sea menos costosa. En cualquier caso, es necesario recurrir a técnicas sofisticadas para resolver el problema.

Las raíces de la planificación en I.A. se encuentran en la resolución de problemas mediante la búsqueda en un espacio de estados y otras técnicas similares (GPS [Newell y Simon 1963], QA3 [Green 1969], etc.) y en las necesidades de la robótica. De hecho, se considera que la planificación como campo específico surgió hace tres décadas con el primer sistema de planificación importante: STRIPS (Stanford Research Institute Problem Solver [Fikes y Nilsson 1971]). STRIPS realizó contribuciones muy importantes, como la definición de un lenguaje de especificación de problemas de planificación que ha servido como base para otros lenguajes considerados ahora como estándar.

A principios de los años 90, la mayoría de planificadores se basan en el modelo de planificación de orden parcial (que define los planes sin necesidad de establecer un orden entre todas sus acciones), y ninguno de ellos puede generar planes de más de 30 acciones [Long y Fox 2002]. Esta situación, sin embargo, cambio radicalmente con la aparición en 1995 del planificador Graphplan [Blum y Furst 1997], que seguía una nueva aproximación basada en grafos. El éxito de Graphplan revitalizo la

investigación en el área de la planificación independiente del dominio, lo que propicio el desarrollo de nuevos sistemas basados en técnicas diferentes de las empleadas tradicionalmente:

- En [Kautz y Selman 1996] se demostró que un algoritmo de satisfactibilidad era capaz de mejorar el rendimiento de Graphplan y de otros algoritmos especialmente diseñados para trabajar con problemas de planificación.
- En [Bonet y Geffner 2001] se mostró que la utilización de técnicas de búsqueda heurística en los algoritmos de planificación permitía obtener resultados muy interesantes.

En los últimos años, el rendimiento de las técnicas independientes del dominio ha experimentado mejoras significativas. Los nuevos planificadores desarrollados, como FF [Hoffman y Nebel 2001], LPG [Gerevini y Serina 2000] o SGPlan [Chen et al. 2004], permiten resolver problemas más complejos, lo que se traduce en importantes avances en el área.

Durante muchos años, los esfuerzos de investigación en el área de planificación se centraron en la búsqueda de soluciones óptimas. Debido a la intratabilidad del problema de planificación, los algoritmos desarrollados sólo eran capaces de resolver problemas muy sencillos. Para facilitar esta labor, se impusieron una serie de restricciones que pocas veces se satisfacen en aplicaciones reales.

El mundo se considera estático si no cambia mientras el planificador está calculando el plan y, durante la ejecución del plan, sólo cambia de acuerdo con el resultado de las acciones del plan. Determinista implica que las acciones tienen un resultado bien definido y conocido por el planificador. El planificador conoce todos los hechos relevantes del entorno.

Probablemente, una de las aproximaciones independientes del dominio más eficientes para la planificación en entornos dinámicos es la integración del proceso de planificación y ejecución [Ambros-Ingerson y Steel 1988]. El planificador genera progresivamente un plan al mismo tiempo que se ejecutan las partes del plan que ya han sido calculadas. De esta forma, el planificador puede incorporar información que sólo está disponible en tiempo de ejecución, y puede reaccionar ante cualquier situación inesperada (sin necesidad de haberla previsto anticipadamente).

Esta aproximación, sin embargo, requiere unos tiempos de respuesta razonables para evitar saltos en la ejecución. Muchos sistemas reales, además, necesitan que las acciones se lleven a cabo dentro de un plazo de tiempo limitado. En estos sistemas, que reciben el nombre de sistemas de tiempo real, el tiempo que se tarda en obtener la respuesta es tan importante o más que la respuesta en sí misma.

Este comportamiento es complejo de obtener, por lo que apenas existen planificadores independientes del dominio basados en esta técnica.

1.8.2 Problema de planificación

Un problema de planificación es un problema de búsqueda que requiere encontrar una secuencia eficiente de acciones que conducen a un sistema desde un estado inicial hasta un estado objetivo. Desde el punto de vista de agentes inteligentes, el campo de la planificación busca construir algoritmos de control que permitan a un agente sintetizar una secuencia de acciones que le lleve a alcanzar sus objetivos.

Más formalmente, un problema de planificación se puede definir mediante una tupla (L, O, I, G) , donde cada uno de estos elementos representa lo siguiente:

- L = Conjunto de fórmulas atómicas, denominadas hechos o literales. Representa el conjunto de hechos que son relevantes en el problema.

- O = Conjunto de acciones definidas en el dominio del problema.
- I = Conjunto inicial de hechos que forman la situación inicial del problema
- G = Conjunto final de hechos que deben formar parte de la situación final del problema.



Gráfico 7. Esquema del Planificación

Las investigaciones en el campo de la planificación en I.A. surgieron para proporcionar técnicas, basadas en el razonamiento inteligente sobre las acciones y sus consecuencias, capaces de afrontar estos problemas.

1.8.3 Planificación heurística

La planificación heurística se basa en la utilización de funciones de estimación para explorar el espacio de búsqueda de una forma más inteligente. Las funciones de estimación (heurísticas) se pueden clasificar en admisibles (A^*) o no admisibles [Nilsson 1994]. Las heurísticas admisibles permiten alcanzar soluciones óptimas, mientras que las no admisibles no ofrecen garantías de que sean óptimas, aunque disminuyen notablemente el tiempo de cómputo.

Tradicionalmente, las funciones heurísticas empleadas eran dependientes del dominio [Penberthy y Weld 1992]. Sin embargo, con la aparición de la planificación basada en grafos, han surgido diversas heurísticas independientes del dominio, suficientemente informadas para guiar el proceso de planificación de forma eficiente.

El principio general para generar funciones heurísticas es formular una versión simplificada (o relajada) del problema. Resolver el problema relajado es, en general, más sencillo que la resolución del problema original. La solución del problema relajado se emplea como heurística para estimar la distancia hasta el objetivo.

Sin embargo, la resolución óptima del problema relajado sigue siendo un problema intratable [Bylander 1994]. Por ello, hay principalmente dos métodos para afrontar este problema:

- Resolver el problema relajado de forma subóptima. Esta es la opción empleada, por ejemplo, en el cálculo de la heurística de FF.
- Considerar que los hechos son independientes, es decir, que no existen interacciones entre los objetivos y que, por lo tanto, se puede calcular un plan independiente para cada uno de ellos. Este es el caso de las heurísticas hadd de Unpop y HSP [Bonet y Geffner 2001c].

1.8.4 Redes de tareas jerárquicas

La planificación jerárquica (HTN) explota un conjunto ordenado de abstracciones para controlar la búsqueda. Esta aproximación ha demostrado ser muy efectiva para reducir la complejidad de los problemas de planificación. Bajo ciertas asunciones, es posible reducir un espacio de búsqueda exponencial a un tamaño lineal [Knoblock 1991].

A pesar de que los planificadores HTN emplean generalmente un motor de planificación independiente del dominio, la definición de las distintas abstracciones para cada dominio se suele realizar manualmente. La flexibilidad y eficiencia de esta aproximación, sin embargo, han dado lugar a numerosos sistemas de planificación como O-Plan [Currie y Tate 1991], las distintas versiones de SHOP [Nau et al. 2003] y SIPE [Wilkins 1999], HyHTN [McCluskey et al. 2003] y UCMP [Erol et al. 1994].

O-Plan es un sistema abierto y flexible orientado a la generación y ejecución de planes. Este sistema combina diversas técnicas como

heurísticas en el control de la búsqueda, restricciones temporales y recursos. El lenguaje que utiliza (Task Formalism, TF) es suficientemente expresivo como para definir las jerarquías existentes entre las tareas de un dominio.

SHOP es otro planificador jerárquico cuya principal característica es que planifica las tareas en el mismo orden en el que se ejecutarán. Esta técnica evita las complejas interacciones entre los objetivos que sufren otros planificadores HTN. Además, le permite conocer el estado completo en cada paso de la planificación, por lo que puede manejar representaciones muy expresivas de los dominios, incluyendo expresiones numéricas complejas, interacción con agentes externos y acceso a fuentes de información externas.

El sistema SIPE, y su sucesor SIPE-2, integra los procesos de planificación y ejecución. HyHTN, por el contrario, es un planificador más clásico.

Implementado en Prolog, su aportación más relevante reside en la combinación de estrategias HTN con una búsqueda progresiva orientada hacia los objetivos. Además, incorpora la herramienta GIPO-II para la generación y validación de los modelos jerárquicos de los dominios. (Ortega & García, 2006)

Prolog es una herramienta práctica para programación lógica que puede ser utilizada en el desarrollo de programas de inteligencia artificial debido a que permite realizar con facilidad consultas y recibir respuestas sujetas a axiomas previamente definidos.

Prolog permite la obtención de respuestas a partir de consultas (interrogantes) a una base de conocimientos conformada por relaciones geográficas existentes entre las calles de la ciudad.

Los lenguajes de objetos actuales permiten interactuar con el usuario, capturar datos, procesarlos y presentar los resultados de manera gráfica y amigable. Tomando en cuenta las características expuestas, surge la necesidad de explotar ambos tipos de lenguajes tomando lo mejor de sus particularidades al momento de desarrollar el modelo de gestión. (Aruquipa, Hurtado, Márquez, & Saravia, 2007)

En el diseño de un proyecto se hace deseable desarrollar interfaces de usuario en plataformas como Java y por otra parte el conocimiento representado por medio de Prolog.

1.8.5 Planificación de rutas

Planificación de rutas consiste en encontrar una ruta óptima capaz de guiar un vehículo desde una posición inicial a una final pasando por una serie de puntos intermedios de forma que se cumplan una serie de requisitos impuestos. (Del Río, 2006) La eficiencia es muy importante, una ruta óptima permite ahorrar tiempo y dinero en combustible.

La inteligencia artificial nos proporciona las herramientas necesarias para resolver el principal problema que se nos presenta para llevar a cabo este modelo: cómo calcular la ruta óptima de entrega dentro del centro de ciudad de Cuenca teniendo en cuenta las restricciones establecidas.

Para los requisitos de la aplicación a desarrollar. Se ha identificado como necesario el uso del lenguaje declarativo Prolog que a través de un proceso de inferencia sobre la Base de Conocimiento y con la información recibida en tiempo real determinará la ruta a seguir.

CAPÍTULO II

ESTUDIO Y REVISIÓN DE LAS HERRAMIENTAS PARA EL DESARROLLO DEL MODELO PARA GESTIÓN DE ENTREGAS POR RUTAS ÓPTIMAS, MEDIANTE GPS Y CONEXIÓN CELULAR.

2.1. INTRODUCCIÓN

Para desarrollar un ambiente amigable, funcional y eficiente del modelo de gestión de entregas es necesario determinar el conjunto de herramientas a utilizarse; la integración de las mismas es sumamente importante, pues de ello dependerá el éxito de la aplicación. En este capítulo se describirá dichas herramientas y como se utilizará a lo largo del desarrollo.

2.2. SERVICIO WEB MAP SERVICE (WMS)

El servicio Web Map Service fue definido por el Open Geospatial Consortium (OGC), este produce mapas de forma dinámica a partir de información geográfica. (Wikipedia Enciclopedia Libre, 2010)

La especificación internacional WMS del OGC define un mapa como una representación de la información geográfica en forma de archivo de imagen digital, adaptado para la visualización en una pantalla de computador. Un mapa no consiste en los propios datos, sino en una imagen de los mismos.

Los Servicios de Mapas en Red (WMS: Web Map Services) permiten visualizar información geográfica georreferenciada a través de la web. La información, que puede ser del tipo vectorial o raster, se presenta en forma de capas de información. Para generar un determinado mapa, las capas son superpuestas según un orden y un valor de transparencia. (Instituto Geográfico Nacional, Ministerio de Fomento, 2007)

2.2.1. Funcionamiento

El servicio de WMS puede invocarse a través de un navegador web (cliente) enviando una petición en forma de URL (Uniform Resource

Locator). Esta petición es recibida y procesada por el servidor WMS y como respuesta, devuelve al cliente una imagen en calidad de pantalla en formato imagen, como son JPEG, GIF, PNG, etc.

Los mapas son generalmente dibujados en un formato de imágenes como PNG, GIF o JPEG, y ocasionalmente como gráficos vectoriales en formato SVG o Web CGM. El uso de formatos de imagen que soportan fondos transparentes (ej: GIF o PNG) permite que las capas subyacentes sean visibles.



Gráfico 8. Funcionamiento del servicio WMS

Los mapas generados por los WMS pueden visualizarse a través de un navegador web o a través de un software que debe instalarse en el PC. Se pueden solicitar capas individuales de diversos servidores, produciendo el solapamiento de capas procedentes de diferentes fuentes. Las capas de información pueden estar almacenadas en distintos servidores

Para que la interoperabilidad de la información y los servicios de mapas sea posible, es necesario que los WMS cumplan con ciertos estándares.

Los WMS tienen la capacidad de leer los datos en sus formatos originales (.dgn, ESRI .shp, geotiff, .ecw, conexiones con bases de datos Postgis, Oracle Spatial, ESRI arcSDE, etc.), y generar como producto de salida

una imagen en formato png, gif, jpg, wmf, etc. Esto evita tener que transformar el formato de almacenamiento de los datos.

2.2.2. Operaciones

La especificación WMS del OGC define tres operaciones, de las cuales las dos primeras son obligatorias:

- GetCapabilities (obligatoria): Devuelve los metadatos del servicio, es decir una descripción del contenido de información del WMS y de los parámetros de petición admisibles.

La respuesta a una operación GetCapabilities es un documento xml con información general sobre el servicio e información específica sobre las capas disponibles en él, es decir los metadatos del servicio. El documento XML es acorde con el DTD de la normativa o con el XML Schema según la versión del WMS. Tanto el DTD como el XML Schema especifican el contenido obligatorio y opcional de la respuesta del GetCapabilities.

En el siguiente esquema se muestra la estructura general del documento XML de capacidades.

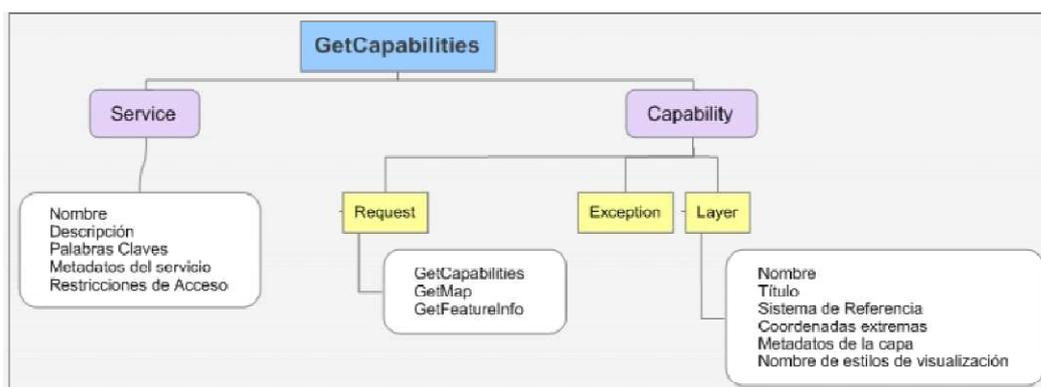


Gráfico 9. Esquema de la estructura de una documento XML

- GetMap (obligatoria): La operación GetMap está diseñada para devolver un mapa, a través de una imagen gráfica o un conjunto de

elementos gráficos. Cuando el cliente envía una solicitud GetMap, el servidor de mapas la interpreta y devuelve un mapa, en caso que no pueda generarlo el servidor debe lanzar una excepción, esto es un error en el mismo formato que se ha solicitado.

- GetFeatureInfo (opcional): La operación GetFeatureInfo está diseñada para mostrar los atributos de los fenómenos (features) que aparecen en el mapa obtenido como resultado de una operación GetMap anterior. Por lo tanto, esta operación proporciona la posibilidad de consultar los atributos del objeto que se encuentra en un determinado que se selecciona en pantalla.

La información que devuelve la operación GetFeatureInfo, depende de la información alfanumérica que el responsable de la cartografía haya deseado hacer pública.

Las operaciones WMS se invocan utilizando un navegador estándar, realizando peticiones en la forma de URLs (Uniform Resource Locators). El contenido de tales URLs depende de la operación solicitada. (Instituto Geográfico Nacional, Ministerio de Fomento, 2007)

2.3. MAPSERVER

Es un software Open Source para la creación de servicios WMS. Es uno de los programas más utilizados para la publicación de mapas a través de Internet. Desarrollado originalmente por la Universidad de Minnesota (UMN) en cooperación con la NASA y el Departamento de Recursos Naturales de Minnesota.

2.3.1. Estructura

El archivo principal de MapServer “mapserv.exe” se almacena en el directorio “cgi-bin”.

Este archivo se ejecuta como una aplicación CGI desde el Servidor HTTP y necesita de los siguientes recursos:

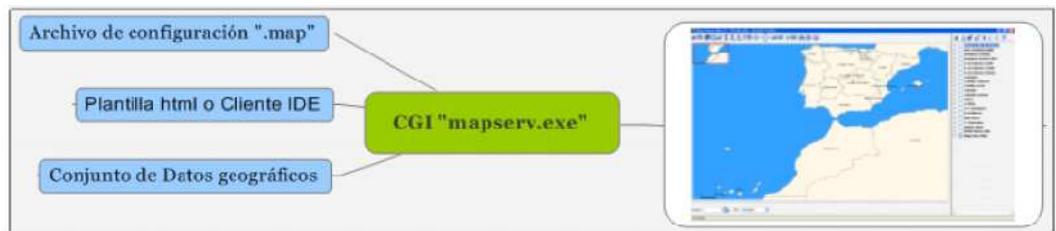


Gráfico 10. Estructura principal de Mapserver

1. Un servidor HTTP como Apache HTTP Server o Internet Information Server.
2. El software MapServer. (ej: mapserv.exe).
3. Un archivo de configuración de extensión “.map”, cuya función es indicar a MapServer que datos se visualizarán y cómo.
4. Una plantilla que controla la interfaz del usuario de la aplicación MapServer en un navegador de Internet (ej: plantillas html) o un Cliente IDE como gvSIG.
5. Un conjunto de datos (ej: raster, vector, bases de datos, etc.).

2.3.2. El archivo de configuración “.map”

El archivo de configuración “.map” es el corazón de MapServer, ya que en él se definirán las capas de información que contendrá el servicio, cómo serán visualizadas y consultadas, el estilo con que se representarán, su simbología, formato en que se generará la imagen, el sistema de referencia, etc. Se trata de un archivo de texto, donde se definen distintos parámetros y sus respectivos valores.

El objeto MAP da inicio al archivo de configuración. Este objeto anida a otros objetos, como se puede observar en el siguiente gráfico. Dentro de cada objeto se definen una serie de parámetros, algunos de los cuales son obligatorios, mientras que otros son opcionales o tienen un valor asignado por defecto.

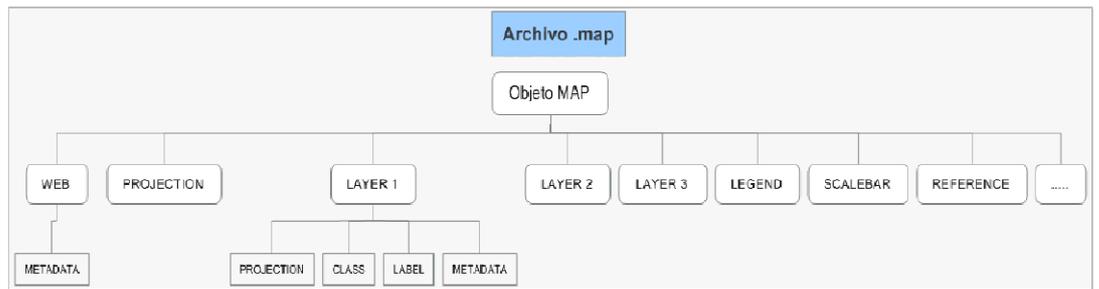


Gráfico 11. Estructura de un archivo .map

2.4. OPENLAYERS

“OpenLayers”, se trata de una librería JavaScript Open Source, que permitirá publicar fácilmente mapas dinámicos en páginas web. OpenLayers se compone de:

- Una librería JavaScript, llamada “OpenLayers.js”.
- Código JavaScript incrustado en un archivo html.

A través de MapServer y OpenLayers se desarrollará el ambiente gráfico de la aplicación, cumpliendo los requerimientos de GetCapabilities, GetMap, y GetInfo. (Instituto Geográfico Nacional, Ministerio de Fomento, 2007)

2.5. LENGUAJE DE PROGRAMACIÓN PROLOG

Prolog es un lenguaje de programación creado para representar y utilizar el conocimiento que se tiene sobre un determinado dominio, es decir, que es usado para resolver problemas que envuelven objetos y las relaciones entre ellos, orientado a la Inteligencia Artificial, usando la programación lógica.

El dominio es un conjunto de objetos y el conocimiento se representa por un conjunto de relaciones que describen las propiedades de los objetos y sus interrelaciones. Un conjunto de reglas que describa estas propiedades y estas relaciones es un programa PROLOG.

Su definición basada en la lógica formal, permite ejecutar estatutos: oraciones de un lenguaje lógico elemental particular de cláusulas. (Tuckler)

2.5.1. Estructura

Un programa Prolog está formado por una secuencia de enunciados (cláusulas): hechos, reglas y variables.

- **Hechos:** Expresan relaciones entre objetos.
- **Reglas:** Las reglas se utilizan en Prolog para significar que un hecho depende de uno o más hechos.
- **Variables:** Representan objetos que el mismo PROLOG determina. Una variable puede estar instanciada o no instanciada. Esta instanciada cuando existe un objeto representado por una variable.

2.5.2. Base de Conocimiento

Una base de Conocimiento es un tipo especial de base de datos para la gestión del conocimiento. Provee los medios para la recolección, organización y recuperación computarizada de conocimiento.

Para el desarrollo de la aplicación se definirá una base de conocimiento acerca de todos los elementos relevantes del área, esto es, información de vías del centro de la ciudad de Cuenca. Facilitando el desarrollo de los mecanismos de inferencia genéricos sobre el conocimiento de cada área concreta para la ruta óptima.

2.5.3. Inferencia

A partir de la base de conocimiento se inferirá la ruta óptima entre los lugares a recorrer, en concreto se utilizará un algoritmo que calcula la ruta óptima entre uno o varios destinos de entrega, usando para ello un algoritmo que englobe las necesidades del sistema e interactúe con las restricciones impuestas para alcanzar la ruta óptima a recorrer.

El algoritmo recorre desde el origen hasta encontrar el destino, explorando siempre la rama del mismo que tenga menor peso total (los pesos en este caso representan las distancias entre nodos y su interacción con el peso de la mercancía).

2.5.4. JPL

JPL es una librería cuyo archivo `jpl.jar` proporcionado por SWI-Prolog que define un conjunto de clases Java y funciones C que proveen una interfaz entre Java y Prolog. JPL usa una interfaz nativa de Java (JNI) para conectarse al motor Prolog Foreign Language Interface (FLI).

JPL está diseñado en dos capas, una interfaz de bajo nivel a la FLI Prolog y Java una interfaz de alto nivel para el programador de Java que no se ocupa de los detalles de la FLI Prolog. (Aruquipa, Hurtado, Márquez, & Saravia, 2007)

Objetivos JPL:

- Habilitar las aplicaciones Prolog para explotar cualquier clase de Java, instancias, métodos, etc.
- Habilitar las aplicaciones Java para manipular cualquier librería estándar de Prolog y sus predicados.
- Habilitar aplicaciones híbridas Prolog+Java para ser implantadas para tomar la mejor ventaja de ambos sistemas de lenguaje, y que sea depurable y mantenible.
- Impacto mínimo en deliberación. El soporte en tiempo de ejecución debe ser dependiente de la posición, árbol de almacenamiento de archivos autosuficiente, sin cambios al registro, librerías de sistema, archivos de configuración, etc.
- Dependencia mínima: Después de la JVM, el soporte en tiempo de ejecución Prolog+Java no debe depender de nada que no se pueda asumir como existente y soportado en instalaciones típicas de S.O.
- Vulnerabilidad mínima en la deliberación: el Prolog+Java debe ser inmune a variaciones legítimas de su entorno de trabajo. (Morales, 2007)

El desarrollo de la aplicación del modelo de gestión para entregas por rutas óptimas tiene como objetivo interactuar entre java y librerías, predicados, etc., que son propias de Prolog, dando lugar a que estas aplicaciones compuestas sean diseñadas e implementadas de tal manera

que se aproveche lo mejor de ambos lenguajes. Se describirán los pasos necesarios a seguir para que ambos lenguajes interactúen. Una vez logrado este objetivo, se demostrará el funcionamiento de lo obtenido mediante un caso de estudio, en el cual la interfaz visual estará desarrollada en Java, mientras que la inferencia estará a cargo de rutinas escritas en Prolog.

La aplicación está orientada a poder obtener respuestas a partir de consultas (interrogantes) a una base de conocimientos Prolog conformada por relaciones geográficas existentes entre las calles del centro histórico de la ciudad de Cuenca.

2.6. GPS VEHICLE TRACKER (TRAX S6)

Como se estableció en el Capítulo I, el dispositivo GPS de rastreo de vehículos que combina el sistema de posicionamiento global GPS y el sistema de comunicación GSM/GPRS para informar la posición y estado del vehículo que se utilizará es el **TRAX S6** desarrollados por GTE.

2.6.1. Características

Este dispositivo posee las siguientes características:

- **Inteligencia integrada:** La capacidad de poder resolver por sí mismo condiciones y/o circunstancias determinadas y realizar las acciones pertinentes para dicha situación.
- **Robustez y confiabilidad:** Diseño de hardware con tecnología de última generación, firmware extremadamente estable, procesos de producción y control rigurosos, prestaciones escalables y el cumplimiento de las más exigentes normas automotrices.
- **Administrador de eventos:** Se puede seleccionar qué tipo de eventos, acciones o combinación de ellos generan reportes, y según la prioridad del tipo de reporte generado si es necesario comunicarlo con urgencia o almacenarlo y esperar a que lo solicite la base.

- **Conectividad:** Posee soporte multiprotocolo, lo que le da la capacidad de conectarse a través de distintos tipos de comunicadores como celulares, radios digitales de doble vía o transceptores satelitales.

Cuando la conexión se realiza por Internet, la capacidad de resolución de nombres por DNS le da un poder exclusivo para aquellos clientes que posean dominios registrados y que por alguna razón puedan cambiar repentinamente de dirección IP.

Cuando el medio de comunicación es la red GSM se tiene una excelente estabilidad y optimización en la gestión de los paquetes enviados y recibidos por GPRS.

- **Control de comunicaciones:** Se puede configurar el tipo y el medio de comunicación entre el equipo y la base, así como también los tiempos de espera, momento, duración, cantidad de intentos, etc.
- **Control de mensajes:** Se encuentra integrado a la unidad un sistema de mensajería capaz de enviar, recibir y administrar mensajes de texto.
- **Código de acceso:** Asegura que la configuración del equipo y la información almacenada en él sólo puedan ser accedidos por usuarios habilitados.
- **Autenticidad de los reportes o mensajes enviados:** El *TRAX S6* está dotado con la posibilidad de poder emitir una firma digital en cada mensaje enviado para establecer la autenticidad de dichos paquetes.

- **Entradas y salidas digitales:** El *TRAX S6* cuenta con entradas digitales para la conexión de diferentes sensores o llaves que pueden ser utilizados para disparar eventos.
- **Entradas analógicas:** Dos entradas pueden ser utilizadas para el procesamiento de señales analógicas, como por ejemplo sensores; pudiendo establecer ventanas de rangos permitidos para analizar sus estados.
- **Capacidad de almacenamiento:** Cuenta con una elevada capacidad de espacio en disco para almacenamiento

2.6.2. Funcionamiento

TRAX S6 genera reportes a partir de eventos o acciones predeterminadas por el usuario. Esto significa que la unidad generará un reporte sólo si ocurre un evento o una combinación de ellos, y lo comunicará a la base o lo almacenará en su memoria según el nivel de prioridad o los requerimientos que hayan sido preestablecidos, optimizando las comunicaciones, ya que la unidad sólo utiliza recursos de comunicación durante algunos instantes cuando ocurran estos eventos.

La configuración de reportes permite activar/desactivar una salida digital, realizar operaciones lógicas complejas o realizar cualquier otra acción deseada. Estas acciones se deciden según la ocurrencia de eventos y/o disparadores predeterminados por el usuario en su configuración. Los eventos pueden ser cualquier situación en la que intervenga un cambio de estado de alguna señal u objeto, como por ejemplo salir de una región predefinida, el incremento de un contador o timer, exceder o estar por debajo del límite de velocidad, no llegar al destino en horario, la apertura de una puerta o cualquier otra situación.

También se pueden realizar relaciones de combinación lógica entre estos eventos, aumentando así la flexibilidad.

En el siguiente cuadro se detallan ejemplos de las posibles situaciones que disparan eventos y las acciones a tomar. Cualquier combinación de las siguientes situaciones es un evento:

Eventos	Acciones
<ul style="list-style-type: none"> - Entrar/Salir de una zona predefinida - Distancia recorrida - Tiempo transcurrido - Velocidad del vehículo - Cambio rumbo del vehículo - Nivel de las RPM del motor - Detención del vehículo - Cambios de estado de encendido/apagado del motor - Cambios de estado de las entradas digitales y/o analógicas - Cambios de estado de las salidas digitales - Estado del GPS - Lógica combinacional entre eventos - Estado de los reportes en los diferentes destino (IP0 , IP1 ,SM0 ,SM1, LOG, SDM) - Etc. 	<ul style="list-style-type: none"> - Enviar un reporte a la base - Guardar un reporte en la unidad en los diferentes destinos (IP0 , IP1 ,SM0 ,SM1, LOG, SDM) - Enviar información y/o reportes a través de los puertos seriales - Encender/apagar las salidas digitales - Activar o reiniciar un timer - Incrementar un registro contador - Modificar el estado de una bandera de estado del sistema (verdadero/falso) - Reconfigurar el sistema para otra situación de trabajo - Etc.

Tabla 1. Ejemplos de situaciones que disparan eventos y las acciones a tomar

El sistema tiene la capacidad de poder cargar un reporte en diferentes destinos al mismo tiempo, como también la posibilidad de ejecutar múltiples acciones en el mismo evento.

2.6.3. Protocolo de Comunicación GAP

Para la configuración el **TRAX** posee un amplio y versátil set de instrucciones y comandos. El protocolo utilizado es GAP (GTE ASCII Protocolo).

2.6.3.1. Descripción del protocolo GAP

Todos los mensajes comienzan con el carácter '>' y terminan con el carácter '<'. El mensaje está compuesto por un encabezado que especifica el tipo de mensaje, una cantidad de datos variable según el tipo, un identificador de dispositivo (opcional), un número o identificador de mensaje (opcional), un destino de respuesta (opcional) y un checksum en formato hexadecimal (opcional en el envío) que se calcula haciendo una or-exclusiva con todos los datos comenzando con '>' y terminando por el carácter '*' (indicador de checksum) inclusive.

2.6.3.2. Tipos de mensajes

Los tipos de mensajes pueden ser 4 y están identificados con las letras 'S', 'Q', 'R' y 'E' seguidos del correspondiente comando, a estas letras son los calificadores de mensajes.

Tipo	Comando
S	Set; es usado para escribir parámetros en el dispositivo.
Q	Query; es usado para preguntar el valor de parámetros al dispositivo.
R	Response; indica la respuesta del dispositivo a un comando S o Q
E	Error; indica que la información requerida de comando S o Q fue errónea.

Tabla 2. Tipos de mensajes

2.6.3.3. Identificador de dispositivo

El identificador de dispositivo contiene el siguiente encabezado ';ID=XXXX', donde XXXX es el identificador propiamente dicho. Este parámetro es opcional y si es enviada la respuesta también viene acompañada de este.

2.6.3.4. Identificador o número de mensaje

Contiene el siguiente encabezado ';#XXXXXXXX', donde XXXXXXXX es el identificador del mensaje propiamente dicho y puede contener cualquier carácter alfanumérico hasta un tamaño de 8

caracteres. Este parámetro es opcional y si es enviada la respuesta también viene acompañada de este.

Existen identificadores de mensajes específicos, estos pueden ser:

Identificador	Comando
; #IP0:xxxx	Para los reportes enviados por IP0
; #IP1:xxxx	Para los reportes enviados por IP1
; #SM0:xxxx	Para los reportes enviados por SM0
; #SM1:xxxx	Para los reportes enviados por SM1
; #LOG:xxxx	Para los reportes enviados por LOG
; #SDM:xxxx	Para los reportes enviados provenientes de la memoria SD

Tabla 3. Identificadores de mensajes específicos

Donde xxxx es el número de secuencia y puede ser un valor en hexadecimal desde 0000 a FFFF ; en el caso de IP0, IP1, SM0 y SM1 si este valor va de 0000 a 7FFF son mensajes de prioridad NORMAL, en cambio de 8000 a FFFF son mensajes de prioridad ALTA. Estos identificadores con sus correspondientes números de secuencia son utilizados para aceptar los mensajes recibidos.

Tanto si el identificador de dispositivo y/o el checksum son inválidos el equipo no retorna respuesta alguna.

2.6.3.5. Destino o redirección de respuesta

Este parámetro tiene como finalidad la redirección de la respuesta por diferentes medios y/o direcciones. El encabezado es el siguiente '**@AAABB..BB'** donde AAA corresponde al tipo de destino y BB..BB a los parámetros adicionales.

Parámetro	Comando
:@IP0	La respuesta es enviada vía UDP a la dirección IP0 y al puerto destino preestablecido.
:@IP1	La respuesta es enviada vía UDP a la dirección IP1 y al puerto destino preestablecido.
:@IP:24.232.65.23	La respuesta es enviada vía UDP a la dirección IP 24.232.65.23 y al puerto destino preestablecido.
:@IP:24.232.65.23:1234	La respuesta es enviada vía UDP a la dirección IP 24.232.65.23 y al puerto destino 1234.
:@SM0	La respuesta es enviada vía SMS al destino preestablecido en SM0.
:@SM1	La respuesta es enviada vía SMS al destino preestablecido en SM1.
:@SM:0111560234345	La respuesta es enviada vía SMS al teléfono de destino con el número 0111560234345.
:@TRM	La respuesta es enviada al Puerto Serial (TRM).

Tabla 4. Destino o redirección de respuestas

Ejemplos:

Se envía por cualquier medio:

>QSG02;@SM:0111560234345<

Se recibe por SMS en el teléfono 0111560234345:

>RSG02;*5B<

Se envía por cualquier medio:

>QSG02;ID=1234;#MSG_NUM1;@IP:24.232.65.23:1234;*61<

Se recibe por UDP en la dirección IP 24.232.65.23 y en el puerto destino 1234:

>RSG02;ID=1234;#MSG_NUM1;*2D<

La recepción de cualquier instrucción o comando genera por defecto una respuesta por el mismo tipo de destino que fue recibido tanto para UDP, SMS y TERMINAL. En el caso de SMS la respuesta será enviada al mismo número destinatario que emitió el comando.

2.6.4. Reportes

El reporte que se utilizará para enviar y/o almacenar información ante la ejecución de un evento es:

Reporte de Posición extendida, estados varios, fecha y hora del Equipo

Existen dos(2) comandos capaces de poder entregarnos esta información, PH y PI, la única diferencia entre ambos es que PI nos entrega solo la última posición en 3D y filtrada, o sea que el campo número diez(10), tipo de posición (0 = 2D GPS, 1 = 3D GPS) será siempre igual a uno(1) y la cantidad de satélites, campo número ocho(8), nunca menor a cuatro(4).

Comando PH (Q-R)

Para consultar: **QPH**

El dispositivo responde:

RPH241207150026-3460180-05847823156025106000010131121000000720F400

Representación	
241207	Fecha actual (ddmmaa)
150026	Hora actual (hhmmss)
-3460180	Latitud (Ej : -34.60180°) de la última posición válida
-05847823	Longitud (Ej : -58.47823°) de la última posición válida
156	Dirección (0..359) de la última posición válida Norte = 0, Este = 90, Sur = 180, Oeste = 270
025	Velocidad en KPH de la última posición válida (0..999)
1	Estado de energía del GPS (0=apagado 1=prendido)
06	Cantidad de satélites detectados en la última posición válida

0000	Edad de señal GPS de la última posición válida en segundos (0000 a 9999)
1	Tipo de posición (0 = 2D GPS, 1 = 3D GPS)
013	HDop *10 en la última posición válida
1	Estado de conexión GPRS con IP local activa (0=inactivo 1=activo)
1	Estado de registro GSM 0 not registered, ME is not currently searching a new operator to register to 1 registered, home network 2 not registered, but ME is currently searching a new operator to register to 3 registration denied 4 unknown 5 registered, roaming
21	Nivel de señal GSM 0 -113 dBm o menor 1 -111 dBm 2...30 -109... -53 dBm 31 -51 dBm o mayor 99 no detectado
0000000720	Odómetro Total en Metros
F4	Estado de las entradas digitales en hexadecimal 0x80 Estado de Contacto o Ignición 0x40 Estado de Alimentación Principal 0x20 Entrada digital 5 0x10 Entrada digital 4 0x08 Entrada digital 3 0x04 Entrada digital 2 0x02 Entrada digital 1 0x01 Entrada digital 0
00	Número de evento que generó el reporte en decimal. Cuando el reporte es generado por una consulta es 00

Tabla 5. Listado de comandos PH (Q - R)

2.6.5. Aplicaciones

- **Salida o entrada a zonas delimitadas:** Es posible delimitar zonas prohibidas o permitidas. Esto es si el vehículo entra a una zona prohibida definida de antemano se reportará inmediatamente a la base; para el caso de que la zona de trabajo del vehículo sea conocida y esté delimitada, se puede reportar una alarma si éste llegara a salir de la misma. La forma de las zonas puede ser circular o una combinación de zonas circulares. La zona se define a partir de un punto (coordenadas del centro) y el radio.
- **Uso no autorizado:** Es posible proteger la flota de vehículos contra robo o uso no autorizado, si se define el horario en el que los vehículos no son utilizados y deberían estar estacionados. Así, en el caso de que se le dé arranque al vehículo durante estas horas, se puede reportar a la base con exactitud la hora y lugar en donde se encuentra éste. Está la posibilidad de disparar un evento si el vehículo está en movimiento pero no se haya encendido el motor, como puede ser el caso de que lo remolquen. Otra alternativa es implementar un sistema de corte de corriente o combustible a través de las salidas del dispositivo.
- **Exceso de velocidad:** Con este sistema se puede monitorear la velocidad de los vehículos para asegurarse de que los conductores manejen prudentemente. A su vez es posible establecer límites de velocidad en áreas predefinidas, haciendo una combinación de la definición de la zona y el límite de velocidad. Esto es útil para un vehículo que circule por áreas urbanizadas, ya que se puede definir un límite más bajo para esa región y otro más alto para las rutas o autopistas.
- **Medición de distancia recorrida:** Con el GPS y los potentes algoritmos de cálculos integrados en el equipo, medir la distancia recorrida acumulada con gran precisión es tarea sencilla, sin la

necesidad de recurrir a medios externos para su realización. Utilizando los registros del sistema, se puede tomar ventaja de esta característica para implementar diferentes odómetros según las distancias parciales o totales que se deseen medir. (GTE, 2009)

2.7. COMUNICACIONES MÓVILES

Para el diseño de este modelo la recepción del estatus de la orden se lo realiza a través de mensajes de textos cortos o SMS, permitiéndonos recibir un mensaje alfanumérico con el reporte de estado correspondiente automáticamente. Además, con el servicio GPRS se obtiene la información del reporte enviado por el GPS.

Los equipos celulares permiten realizar el control y monitoreo del sistema utilizando la red de telefonía celular y SMS. La Red de Telefonía celular está basada en la comunicación inalámbrica móvil de voz y datos. El sistema celular se encuentra constituido básicamente por radio bases que cuentan con antenas y sistemas de control para la administración de un área de cobertura.

2.7.1. GSM

El desarrollo de GSM comenzó con un grupo formado por la CEPT (European Conference of Postal and Telecommunications Administrations), para investigar el desarrollo de un estándar de comunicaciones móviles para ser usado en Europa. Este grupo era conocido como el “Groupe Special Mobile”, o GSM, es de ahí de donde vienen las siglas de este estándar, ahora más conocido como Global System for Mobile Communications.

El deseo de un sistema unificado en Europa había sido una constante desde que muchos de los países que lo componen habían desarrollado diferentes tecnologías que eran incompatibles entre sí.

Paralelamente en Estados Unidos, las tecnologías móviles estaban avanzadas también, pero al ser un solo país las necesidades de *roaming* no eran tan importantes como en Europa.

GSM fue adoptado como un estándar europeo por el ETSI, para operar en tres regiones de frecuencias principales:

- 900 MHz
- 1800 MHz
- 1900 MHz

Actualmente, GSM abarca sobre el 80% del mercado global de los sistemas móviles. Aunque las redes evolucionan a 3G, GSM no debería ser desplazada, ya que es una parte de las redes GPRS y UMTS.

2.7.2. GPRS

GPRS es un servicio de datos que permite el envío y recepción de tráfico de paquetes (normalmente paquetes IPv4 o IPv6) a través de una red móvil. El protocolo punto a punto (PPP) permite el cambio de protocolos de forma transparente como pueden ser Appletalk e IPX. Está diseñado para complementar tráfico mediante conmutación de circuitos y el servicio de mensajes cortos (SMS), además de otros nuevos servicios.

En muchos casos GPRS se ve como un paso en la evolución hacia 3G, considerándolo como una tecnología de 2.5G. Siempre se le ha conocido como *always connected*, ya que tras la conexión inicial, las siguientes que se producen son casi instantáneas y con un retardo mínimo.

Las especificaciones de GPRS fueron dadas, originalmente, por la organización ETSI (a partir de 1994), pero finalmente, en el verano de 2000, su estandarización correspondió al 3GPP (Third Generation Partnership Project).

GPRS trabaja aportando los servicios de una red de conmutación de paquetes sobre la red existente de GSM. Esto permite al usuario continuar utilizando la red GSM para la voz pero si se necesita una transferencia de datos, se puede realizar vía GPRS. De esta manera infraestructura existente se puede reutilizar.

GPRS introduce un núcleo de red basado en la conmutación de paquetes, pero utiliza gran parte de la funcionalidad de GSM. También aporta nuevas características, como la capacidad para transportar diferente tipo de tráfico de forma más eficiente para los recursos de la red, y permite introducir una amplia gama de servicios.

Para GPRS/GSM, el interfaz de radio (el aire) es utilizado de una forma flexible, teniendo de uno a ocho canales multiplexados en el tiempo (TDM) para tráfico GPRS.

Un dispositivo móvil con una tarjeta SIM estándar puede conectar a una red GPRS y usar sus servicios si el operador de red tiene diseñadas e implementadas estas características. (Murillo, 2005)

2.7.3. El sistema de mensajes cortos (SMS)

Uno de los servicios que más ha crecido con la llegada del sistema GSM (sistema global de comunicaciones móviles), son los mensajes SMS. Esto se debe a su bajo costo de tiempo aire y a que no ocupa los canales de voz para su transmisión, ya que viajan sobre los canales de señalización y control de la red.

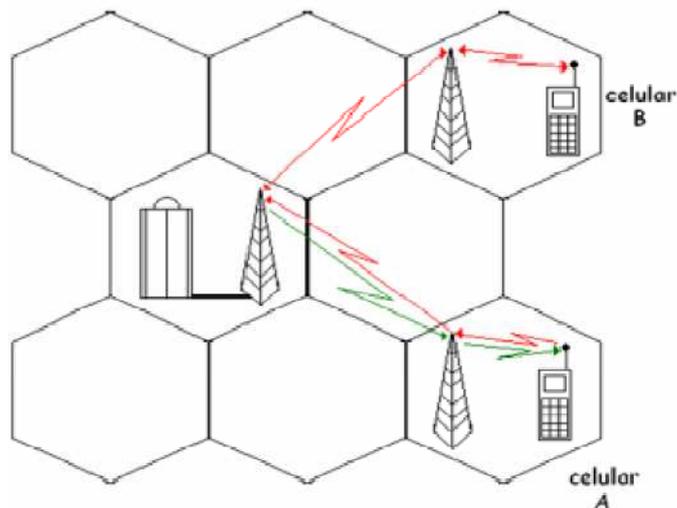


Gráfico 12. Envío de un mensaje SMS

El sistema SMS está soportado por la red de telefonía celular y un centro de mensajes que se encarga de la coordinación de los mensajes. Es importante mencionar que los mensajes no se envían de manera directa de un teléfono a otro, ya que primero viajan a la radio base y de ahí al centro de mensajes donde se re-envían al móvil que se desea que reciba el mensaje. Este proceso se aprecia en el Gráfico 12. Los fabricantes de teléfonos proporcionan un método de acceso al MODEM interconstruido, y a su vez se proporciona un método para controlar el mismo. Este método se basa en los comandos de control de MODEM ‘s denominados comandos AT. (González, 2005)

2.7.3.1. Estándar de comunicación AT para módems GSM

Los comandos AT son instrucciones codificadas que conforman un lenguaje de comunicación entre el hombre y un terminal módem.

Los comandos AT se denominan así por la abreviatura de *attention*. Aunque la finalidad principal de los comandos AT es la comunicación con módems, la telefonía móvil GSM también ha adoptado como estándar este lenguaje para poder comunicarse con sus terminales. De esta forma, todos los teléfonos móviles GSM poseen un juego de comandos AT específico que sirve de interfaz para configurar y proporcionar instrucciones a los terminales. Este juego de instrucciones puede encontrarse en la documentación técnica de los terminales GSM y permite acciones tales como realizar llamadas de datos o de voz, leer y escribir en la agenda de contactos y enviar mensajes SMS, además de muchas otras opciones de configuración del Terminal.

La implementación de los comandos AT corre de cuenta del dispositivo GSM y no depende del canal de comunicación a través del cual estos comandos sean enviados, ya sea cable serial, canal infrarrojo, Bluetooth, etc.

Existen varios tipos de comandos AT de acuerdo a su formato, éstos son:

- AT
- AT&
- AT\
- AT%
- AT+C

En el proyecto actual se requiere del control y manipulación del celular para el envío y recepción de mensajes SMS. En la siguiente tabla se describen los comandos más importantes, todos ellos del tipo AT+C. Una lista de comandos AT para teléfonos móviles se detalla en el Anexo 1.

Comando	Descripción
+CMGF str	Formato del mensaje SMS Str = 0: Establece el formato PDU Str = =?: Retorna los formatos disponibles Str = ?: Retorna el formato actual.
+CMGL	Lista los mensajes SMS almacenados.
+CMGR=n	Lee el mensaje SMS número n
+CMGW=n <PDU string> <Ctrl-Z>	Escribe un mensaje SMS en la memoria con una longitud en octetos de PDU string = n
+CMSS=n	Envía el SMS almacenado n a la red GSM.
+CPMS?	Retorna el almacenamiento de mensajes SMS actual.
+CSCA=str	Dirección del Centro de Servicios SMS Str = ?: Retorna los parámetros actuales Str = =?: Comprueba el comando Str = n: Establece el número del Centro a n.
+CSMS=?	Retorna el servicio de mensajería seleccionado.

Tabla 6. Listado de comandos AT para envío y recepción de mensajes SMS

(Biblioteca de Ingeniería Eléctrica y Electrónica, 2010)

2.7.3.2. El formato de envío de datos PDU

Al acceder al MODEM del teléfono, es posible enviar o recibir mensajes SMS utilizando una terminal, o como en nuestro caso un micro controlador con un puerto serie. Uno de los formatos más utilizados es el formato PDU, este formato contiene una gran cantidad de información adicional al mensaje.

Sin embargo una desventaja de este sistema es que la información del mensaje toma caracteres de 7 bits y los coloca en datos de 8 bits, este procedimiento se realiza tanto en la recepción como en la transmisión, por ejemplo, el mensaje “hola israel” es comprimido como se muestra en el siguiente gráfico.

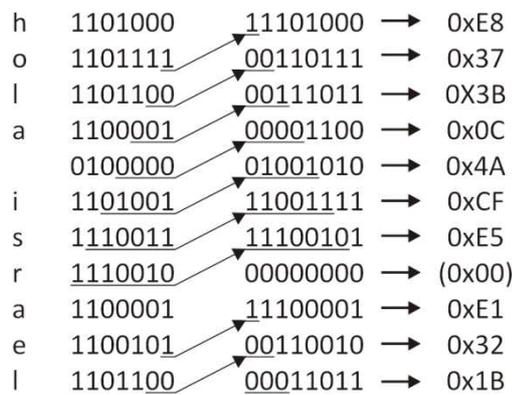


Gráfico 13. Compresión de 7 bits a 8 bits

Este método de compresión permite eliminar 8 bits por cada 8 caracteres que se envían, esto es 1/7 de la información que puede ser considerable debido al tráfico de los mensajes. (González, 2005)

2.7.4. SAMSUNG SGH-E256



Gráfico 14. Samsung SGH-E256

El módem GSM que será utilizado en el presente proyecto de tesis, será el módem incorporado en el teléfono celular de tecnología GSM marca Samsung modelo SGH-E256. Se ha seleccionado este dispositivo por su buen rendimiento y bajo precio. En la siguiente tabla se presentan sus especificaciones técnicas:

Color: Negro

Tapa: No

Tipo de Cámara: VGA 640x480 píxeles

Reproductor MP3: Si

Antena: Interna

Altura x Ancho x Profundidad: 10 x 4.9 x 1.4 cm

Peso: 81 gr

Tecnología EDGE: Si

Tipo de Pantalla: LCD 256.000

Pantalla externa: No

Tiempo de conversación: 5 horas

Tiempo de espera: 250 horas

Conectividad: Bluetooth y USB

Tipos de Ringtones: Mp3 y Polifónicos

Reproducción de Video: Si

Características de la llamada: Multiconferencia

Características de los mensajes: SMS, EMS, MMS

Características Adicionales: Radio Fm

Diseño: Slider

Tecnología: GSM

Memoria externa: microSD

Memoria interna: 10 Mb

2.8. JPCAP

Jpcap es una biblioteca Java de código abierto para la captura y el envío de paquetes de red.

A través de esta biblioteca se puede desarrollar aplicaciones para capturar, visualizar y analizar los paquetes de una interfaz de red en Java. También enviar paquetes arbitrarios a través de una interfaz de red.

Jpcap provee los siguientes procesos:

- Capturar los paquetes de primas en vivo desde el cable.
- Guardar los paquetes capturados en un archivo sin conexión, y lee los paquetes capturados en un archivo sin conexión.
- Identificar automáticamente los tipos de paquetes y genera correspondientes objetos Java (para Ethernet, IPv4, IPv6, ARP / RARP paquetes, TCP, UDP, y ICMPv4).
- Filtrar los paquetes de acuerdo a las normas especificadas por el usuario antes de enviarlos a la aplicación.
- Enviar los paquetes a la red.

Jpcap funciona en Microsoft Windows (98/2000/XP/Vista), Linux (Fedora, Mandriva, Ubuntu), Mac OS X (Darwin), FreeBSD y Solaris; puede capturar Ethernet, IPv4, IPv6, ARP / RARP paquetes, TCP, UDP, y ICMPv4. (Jpcap Java library for capturing and sending network packets)

Para el desarrollo del modelo se determinó que es mejor utilizar el servicio GPRS con transmisión de datos al servidor, ya que, el proceso es más óptimo recibiendo directamente la información sin pasar por algún otro proceso como es el SMS. Con esta biblioteca se puede capturar la señal del GPS a través de un puerto definido, se creará una aplicación con esta biblioteca para que capture los datos continuamente desglosando la información e ingresando a la tabla de la base de datos correspondiente, con esto se tendría el registro del rastreo y ruta que ha seguido el conductor.

CAPÍTULO III

ANÁLISIS Y DISEÑO DE LOS PROCEDIMIENTOS QUE COMPRENDERÁN EL DESARROLLO DEL MODELO PARA GESTIÓN DE ENTREGAS POR RUTAS ÓPTIMAS, MEDIANTE GPS Y CONEXIÓN CELULAR, ASÍ COMO LA RECOLECCIÓN, LEVANTAMIENTO Y DEPURACIÓN DE INFORMACIÓN GEOGRÁFICA PARA CREACIÓN DE RUTAS

3.1. INTRODUCCIÓN

En este capítulo se presenta de una manera organizada las especificaciones detalladas del sistema para la gestión de entregas por rutas óptimas que satisfaga las necesidades de información de la empresa y que sirva de base para un modelo. El análisis y diseño es fundamental pues es la base del sistema, así mismo, la información cartográfica forma el corazón del sistema pues para una correcta inferencia de ruta óptima es preciso contar con una información cartográfica precisa, esta reflejará los nodos que conectan las vías del centro de la ciudad, los parámetros necesarios para la captura de señal enviada del GPS al servidor y los medios para recibir y descifrar la información recibida.

3.2. DEFINICIÓN Y ANÁLISIS DEL SISTEMA

Como se ha mencionado anteriormente, se definió que el área para el caso de estudio es el centro histórico de la Ciudad de Cuenca. A continuación se presenta la definición y análisis de los requisitos del sistema, procedimientos, casos de uso, herramientas y modelo de base de datos a utilizar, los mismos nos servirán para realizar el modelo.

3.2.1. Identificación de los usuarios participantes y finales

Dentro de la empresa existen diferentes tipos de usuarios que interactúan con el sistema, en función de las responsabilidades que poseen. En primer lugar tenemos al administrador que puede gestionar los recursos y funcionamiento de la intranet, le siguen los usuarios que pertenecen a diferentes áreas como son: bodega, ventas, compras, transporte, entre otros; y por último tenemos al usuario invitado que tendrá acceso a las funcionalidades básicas de la intranet.

Para el desarrollo del modelo se toma en cuenta que en la empresa existen trabajadores que desempeñan múltiples funciones y por lo mismo acceden a funcionalidades de diferentes áreas. Se debe aclarar también que la empresa (Madeco) ya cuenta con un sistema con el que realiza todas las funcionalidades pertinentes a facturación, ventas, bodega, inventario, etc. En este modelo demo se extrajo solo las partes necesarias para realizar una gestión de entregas a través de rutas óptimas, por lo que no se hará énfasis en el resto de módulos. Se añadió las especificaciones correspondientes a la gestión óptima de entregas.

3.2.2. Establecimiento de requisitos

El establecimiento de requisitos que seguirá el modelo se lleva a cabo a través de la definición, análisis y validación de los requerimientos a partir de la información obtenida por el usuario, los requisitos del sistema, de las herramientas y de las aplicaciones a utilizar. A continuación se detallan los requisitos obtenidos:

Requisitos funcionales		
Identificador	Tipo de requisito	Descripción
Gestión de usuarios	Funcional	El sistema permitirá la creación, modificación y eliminación de usuarios del sistema. Se realizará los procesos de ingreso básico de los usuarios, así como la asignación del cargo (rol), cambio de contraseña y asignación a empresas.
Gestión de empresas	Funcional	El sistema será multiempresa, por lo cual todas las relaciones respectivas deben estar contempladas en su creación.
Información de la intranet	Funcional	El sistema debe facilitar la información básica sobre el funcionamiento, errores en el sistema, acceso de usuarios y funcionamiento del sistema.
Visualización de consultas	Funcional	El sistema debe permitir a los usuarios acceder a las consultas definidas.

Inferencia de rutas óptimas	Funcional	El sistema debe realizar la inferencia de rutas óptimas en base a las ordenes entrega – recepción que el conductor debe realizar.
Localización de clientes	Funcional	El sistema generará la georeferenciación de los clientes en base a su dirección (calle principal y calle secundaria).
Localización de bodegas	Funcional	El sistema generará la georeferenciación de las bodegas en base a su dirección (calle principal y calle secundaria).
Rastreo de vehículos	Funcional	El sistema receptorá la información de la localización del vehículo a través del reporte generado por el GPS y receptado a través del celular conectado al servidor, ejecutando el proceso de desglose de información y guardado de datos.
Visualización de rutas	Funcional	El sistema permitirá la visualización de las rutas realizadas así como la ruta óptima sugerida.

Tabla 7. Listado de Requisitos Funcionales

Requisitos no funcionales		
Identificador	Tipo de requisito	Descripción
Interfaz gráfica amigable	Apariencia y estilo	El sistema tendrá una interfaz de usuario amigable para facilitar su uso.
Interfaz homogénea	Apariencia y estilo	El sistema contará con una interfaz homogénea para todas sus funcionalidades.
Campos obligatorios	Usabilidad	El sistema tendrá un indicador (*) para los campos que son obligatorios.
Fácil aprendizaje	Usabilidad	El sistema tendrá que ser fácil de aprender, normalmente el usuario a excepción de las funcionalidades complejas podrá interactuar correctamente con la intranet.
Compatible con navegadores actuales	Operacional y de entorno	El sistema debe funcionar correctamente desde los navegadores Web Internet Explorer 8.0 o Mozilla Firefox 2.6 o superiores.
Prevención de datos incorrectos	Seguridad	El sistema bloqueará siempre que sea posible la introducción de datos incorrectos.
Mensajes de alerta y/o error	Seguridad	El sistema mostrará el debido mensaje cuando se tenga alguna alerta o un error determinado.

Accesibilidad a datos privados	Seguridad	Solo las personas que tengan autorización podrán acceder a los datos del sistema. El nivel de acceso está determinado en base a su cargo (rol).
Cálculos optimizados	Rendimiento	El sistema realizará la inferencia de rutas óptimamente.
Robusto	No funcional	El sistema no debe permitir la entrada incorrecta de datos ni la duplicidad de los mismos, garantizando la integridad de los datos del sistema.
CentOS 5, PostgreSQL, PostGIS, MapServer, OpenLayers, Prolog, Netbeans IDE 6.1	Implementación	El sistema se implementará en CentOS 5, en la base de datos PostgreSQL con extensión de PostGIS, el servidor de mapas MapServer con OpenLayers; para la inferencia de rutas óptimas el lenguaje declarativo Prolog y para el desarrollo de las aplicaciones Netbeans IDE 6.1.
Trax S6	Implementación	Para el rastreo de vehículos pesados se utilizará el GPS TRAX S6.
Celular Samsung SGH-E256	Implementación	Dispositivo para recepción de datos referentes al estado de la Orden de Entrega enviada por el conductor.
JPCAP	Implementación	JPCAP para la captura y el envío de paquetes de red.
JPL	Implementación	JPL para la comunicación bidireccional entre Java y SWI-Prolog.
Operadora PORTA	Implementación	Los chips SIMS utilizados por el GPS y celular serán proporcionados por la operadora PORTA.

Tabla 8. Listado de Requisitos No Funcionales

Nota: El detalle de los últimos 6 puntos correspondientes a la implementación se encuentran en el siguiente apartado de herramientas a utilizar.

3.2.3. Herramientas a utilizar

Para realizar el modelo dispondremos de varias herramientas que nos ayudarán a desarrollar el proyecto. En primera instancia, el sistema será

implementado sobre un Sistema Operativo de código abierto, para lo cual se utilizará CentOS 5. Dicho sistema operativo se encuentra en uno de los servidores de la empresa Madeco, en el mismo se procederá a instalar el resto de herramientas necesarias para el desarrollo de la aplicación.

Los lenguajes de programación, base de datos y servidores de mapas a utilizar de código abierto son:

- PostgreSQL y PostGIS
- MapServer
- Lenguaje declarativo Prolog
- Netbeans IDE 6.1 para el desarrollo de programas en Java.

Se pretende realizar este sistema con herramientas de código abierto para ayudar a simplificar los costos asumidos por la empresa Madeco, ya que no se dispone de fondos para la adquisición de licencias para ninguna de las herramientas antes mencionadas.

Además se contará con un dispositivo GPS para vehículos pesados, en este caso se está utilizando el GPS TRAX S6. Este equipo inteligente integra las tecnologías GPS y de comunicación inalámbrica para estar en contacto con la base y enviarle reportes sobre su posición y estado en situaciones predeterminadas por el usuario. Se debe agregar que, dicho dispositivo dispone de un manual el mismo que ayudará para su configuración y programación. Además requiere de una línea telefónica para que pueda ejecutar la comunicación con la base.

Para el desarrollo del modelo se determinó que es mejor utilizar el servicio GPRS con transmisión de datos al servidor, ya que, el proceso es más óptimo recibiendo directamente la información sin pasar por algún otro proceso como es el SMS, por lo cual para la recepción de la información enviada por el GPS es necesario contar con un sniffer el mismo será desarrollado a partir de la librería JPCAP.

Con respecto a las rutas óptimas como se dijo anteriormente se utilizará la librería JPL para realizar la interfaz entre java y Prolog.

Para realizar la actualización de estado de la orden se utilizarán los mensajes escritos, estos serán procesados y extraídos para actualizar las órdenes. En este proyecto se utilizará el celular correspondiente a la marca Samsung modelo SGH-E256. Se utilizará el programa ArcGIS para la depuración de los datos, así como generación de mapas.

En el Capítulo II se detallaron todas estas herramientas, las librerías y su uso.

3.2.4. Especificación de casos de uso

A continuación se presentan los diagramas de casos de usos correspondientes al modelo de gestión; para definir la granularidad del proyecto se agruparon los casos de uso utilizando la siguiente nomenclatura:

Simbología	
Actores	Casos de Uso
 Actor complejo (Interface gráfica de usuario)	 Caso de uso principal
 Actor medio (Interface Protocolo comunicación)	 Caso de uso reutilizado
 Actor sencillo (No dispone de interfaz)	 Caso de uso especializado

Gráfico 15. Simbología para casos de uso

3.2.4.1. Paquetes principales de casos de uso

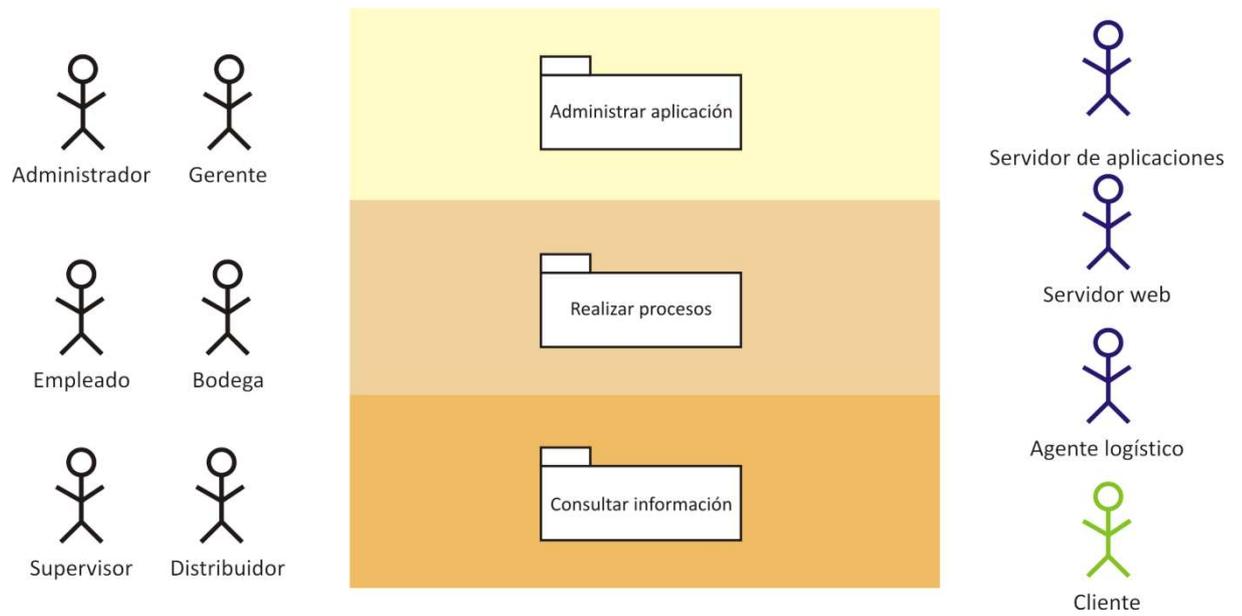


Gráfico 16. Simbología de Paquetes Principales de Casos de Uso

3.2.4.2. Paquetes de casos de uso:

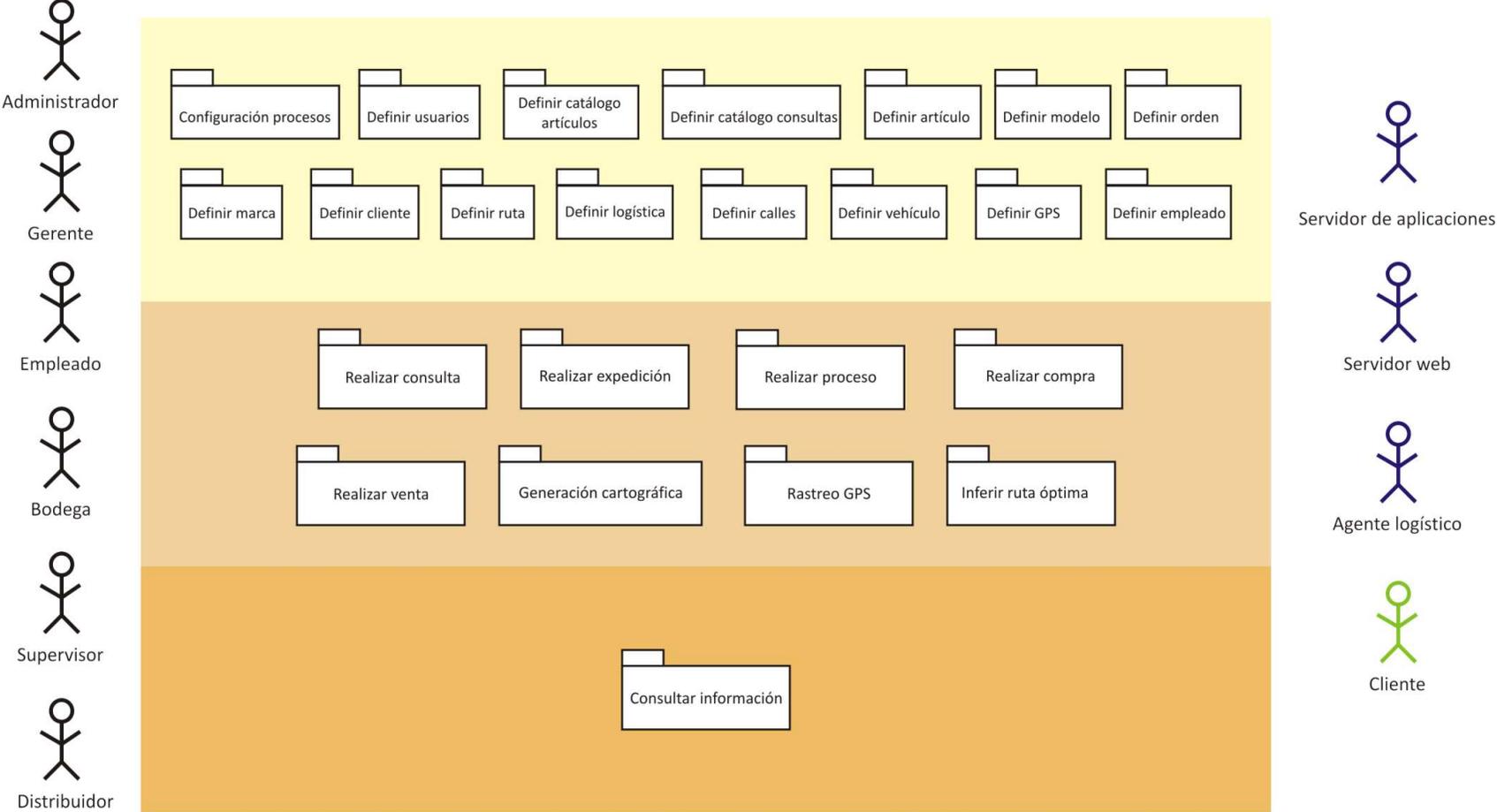


Gráfico 17. Casos de Uso

3.2.4.3. Listado de Casos de Uso

Listado de casos de uso		
Nombre	ID	Objetivo
Configuración de Procesos	UC1.1	Definir la configuración global de la estructura, catálogo de consultas, reglas de ruteo, catálogo de artículos, zona de estudio, reglas de comunicación, y generación de información correspondiente a los nodos y bloques de las calles.
Definir empresa	UC2.1	Definir empresa.
	UC2.2	Modificar los datos de una empresa.
	UC2.3	El administrador quiere eliminar una empresa.
Definir usuario	UC3.1	Definir usuario en la intranet.
	UC3.2	Modificar los datos de un usuario.
	UC3.3	El administrador quiere modificar la contraseña de un usuario.
	UC3.4	El administrador quiere dar de baja usuario del sistema.
	UC3.5	El administrador quiere cambiar de cargo (rol) al usuario.
Definir marca	UC4.1	Definir marca.
	UC4.2	Modificar los datos de una marca.
	UC4.3	El usuario quiere eliminar marca.
Definir modelo	UC5.1	Definir modelo.
	UC5.2	Modificar los datos de un modelo.
	UC5.3	El usuario quiere eliminar modelo.
Definir artículo	UC6.1	Definir artículo.
	UC6.2	Modificar los datos de un artículo.
	UC6.3	El usuario quiere eliminar un artículo.
Definir cliente	UC7.1	Definir cliente.
	UC7.2	Modificar los datos de un cliente.
	UC7.3	El usuario quiere eliminar un cliente.
	UC7.4	El usuario quiere visualizar la localización del cliente en el mapa.
Definir empleado	UC8.1	Definir empleado.
	UC8.2	Modificar los datos de un empleado.
	UC8.3	El usuario quiere eliminar un empleado.
Definir GPS	UC9.1	Definir GPS.
	UC9.2	Modificar los datos de un GPS.

	UC9.3	El usuario quiere eliminar un GPS.
	UC9.4	Rastreo de vehículo.
Definir Calles	UC10.1	Definir calles.
	UC10.2	Modificar los datos de una calle.
	UC10.3	El usuario quiere eliminar una calle.
Generación cartográfica	UC11.1	Generar información cartográfica (bloques y nodos).
	UC11.2	Consultar los datos de un bloque.
	UC11.3	Consultar los datos de un nodo.
Definir vehículo	UC12.1	Definir vehículo.
	UC12.2	Modificar los datos de un vehículo.
	UC12.3	El administrador quiere eliminar un vehículo.
Definir bodega	UC13.1	Definir bodega.
	UC13.2	Modificar los datos de una bodega.
	UC13.3	El administrador quiere eliminar una bodega.
	UC13.4	El administrador quiere visualizar la localización de la bodega en el mapa.
Realizar expedición	UC14.1	Determinar si el artículo existe para su venta realizando expedición.
Definir proveedor	UC16.1	Definir proveedor.
	UC16.2	Modificar los datos de un proveedor.
	UC16.3	El usuario quiere eliminar un proveedor.
	UC16.4	El usuario quiere visualizar la localización del proveedor en el mapa.
Definir orden	UC18.1	Definir orden de Entrega – Recepción (E-R).
Definir catálogo de artículos	UC19.1	Definir catálogo de artículos.
	UC19.2	Modificar los datos de un catálogo de artículos.
	UC19.3	El usuario quiere eliminar un catálogo de artículo.
Definir ruta	UC20.1	Definir ruta de Entrega - Recepción
Definir catálogo consultas ruteo	UC21.1	Definir catálogo de consultas ruteo
Realizar consulta	UC22.1	Realizar consultas
Realizar proceso	UC23.1	Realizar proceso de cálculos.

Tabla 9. Listado de casos de Uso

Caso de uso – Configuración de Procesos

El siguiente caso de uso muestra globalmente los procesos a seguir para la definición de la estructura base del sistema. Cada uno de los procesos está detallado posteriormente.

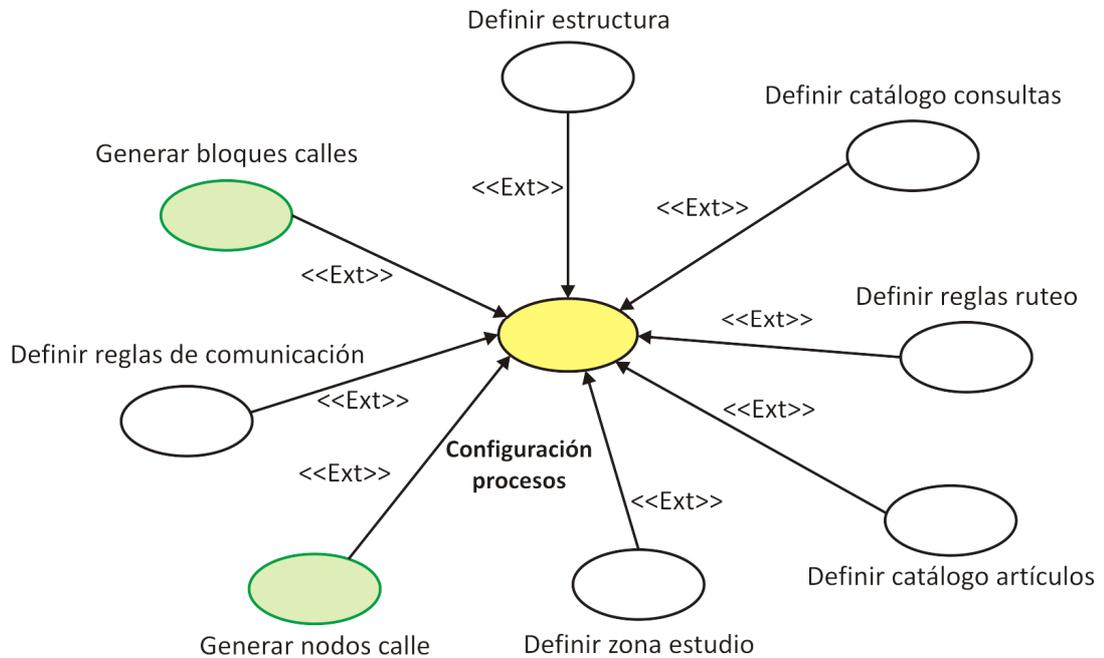


Gráfico 18. Configuración de Procesos

Identificador: UC1.1

Actores: Administrador

Objetivos: Definir la configuración global de la estructura, catálogo de consultas, reglas de ruteo, catálogo de artículos, zona de estudio, reglas de comunicación, y generación de información correspondiente a los nodos y bloques de las calles.

Condiciones previas: Ninguna

Condiciones posteriores: Existe base de datos, catálogos de consultas e información de nodos y bloques de las calles en el sistema.

Frecuencias de uso: Infrecuente. Este caso de uso es crítico, ya que sin él no se establecería la estructura base del sistema con la información necesaria del área de estudio.

Flujo Básico:

1. El caso de uso empieza en la definición de estructura de la base de datos, después de haber definido las herramientas necesarias, se creará la base de datos en PostgreSQL.
2. El administrador definirá el catálogo de consultas en base a la necesidad de información que se establezca que el usuario requiere.
3. El administrador definirá las reglas de ruteo, esto hace referencia a la frecuencia con la que el GPS adquirirá información, su configuración, el disparo de eventos, etc.
4. El administrador definirá la estructura del catálogo de artículos que el usuario utilizará.
5. El administrador definirá la zona de estudio, en este caso se tomará de muestra el Centro histórico de la ciudad de Cuenca.
6. El administrador definirá las reglas de comunicación, como son los protocolos a seguir para recepción y envío de información, configuración del servidor, configuración base de los programas correspondientes a usar.
7. El administrador generará la información correspondiente a los nodos y bloques de las calles ajustados a la zona de estudio en la base de datos.
8. Se crearán las pantallas donde se muestran los campos que se deben llenar para cada tabla definida en la estructura de la base de datos.

Flujo alternativo: Reglas de comunicación incorrecta.

Línea 6: Si los protocolos utilizados son incorrectos en cualquiera de los procesos, se volverán a definir las reglas de comunicación y se vuelve a 6.

Flujo alternativo: Nodos y bloques de las calles ya existen.

Línea 7: Si ya existen los nodos y bloques de las calles el sistema muestra un mensaje de información al administrador y se pasa a 8.

Caso de uso – Definir Empresa

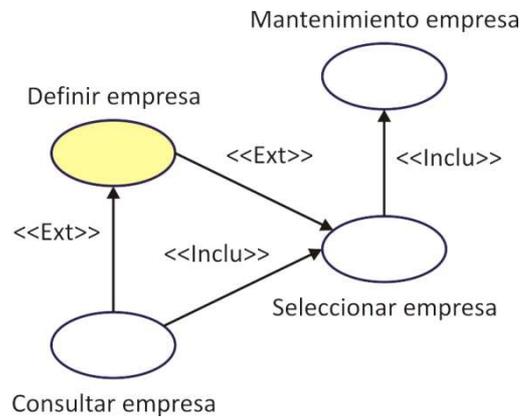


Gráfico 19. Definir empresa

Identificador: UC2.1

Actores: Administrador

Objetivos: Definir empresa

Condiciones previas: Ninguna

Condiciones posteriores: Existe una nueva empresa

Frecuencias de uso: Infrecuente. Este caso de uso es necesario para poder realizar transacciones para la mayoría de procesos correspondientes a una empresa en específico.

Flujo Básico:

1. El caso de uso empieza cuando el administrador quiere crear una nueva empresa.
2. El sistema mostrará una pantalla donde se muestran los campos que se deben llenar para la creación de la empresa.
3. El administrador introduce el nombre y descripción de la empresa.
4. El administrador indica al sistema que ya se ha acabado de introducir los datos aceptándolos.
5. El sistema crea y guarda los datos de la nueva empresa.
6. El sistema muestra al administrador un mensaje indicando que la operación se ha realizado con éxito.

Flujo alternativo: Conflicto de empresa ya existente.

Línea 4: Si la empresa ya existe con el mismo nombre, el sistema muestra un mensaje de información al administrador y se vuelve al paso 3.

Flujo alternativo: Datos obligatorios incorrectos.

Línea 4: Si los datos introducidos se encuentran vacíos o incorrectos, el sistema muestra un mensaje de error al administrador y se vuelve al paso 3.

Flujo alternativo: Error inesperado BD.

Línea 5: La función de guardar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 4.

Identificador: UC2.2

Actores: Administrador

Objetivos: Modificar los datos de una empresa.

Condiciones previas: Existe la empresa que se quiere modificar.

Condiciones posteriores: Los datos de la empresa son modificados con los nuevos datos.

Frecuencias de uso: Infrecuente. Se usa normalmente si se ha equivocado en algún dato.

Flujo Básico:

1. El caso de uso empieza cuando el administrador quiere modificar los datos de la empresa.
2. El sistema mostrará una pantalla con un listado de empresas con su información.
3. El administrador selecciona la empresa y carga la información de la empresa seleccionada.
4. El administrador modifica los datos que quiere actualizar, todos los datos son modificables.
5. El administrador indica al sistema que ya ha acabado de modificar los datos aceptándolos.
6. El sistema hace las modificaciones en los datos de la empresa y los actualiza.
7. El sistema muestra un mensaje al administrador indicando que la operación se ha realizado con éxito.

Flujo alternativo: Los datos de nombre de la empresa modificada coinciden con otro registro.

Línea 6: Si los datos introducidos se encuentran coincidentes con otro registro, el sistema muestra un mensaje de error al administrador y se vuelve al paso 4.

Flujo alternativo: Datos obligatorios incorrectos.

Línea 6: Si los datos introducidos se encuentran vacíos o incorrectos, el sistema muestra un mensaje de error al administrador y se vuelve al paso 4.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de modificar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Identificador: UC2.3

Actores: Administrador

Objetivos: El administrador quiere eliminar una empresa.

Condiciones previas: Existe la empresa que se quiere eliminar.

Condiciones posteriores: Empresa ha sido eliminado del sistema.

Frecuencias de uso: Infrecuente. Normalmente se elimina si hubo error al ingresar empresa.

Flujo Básico:

1. El caso de uso empieza cuando el administrador quiere eliminar una empresa.
2. El sistema mostrará una pantalla con un listado de empresas con su información.
3. El administrador selecciona empresa.
4. El administrador indica al sistema que desea eliminar la empresa aceptando su eliminación.
5. El sistema elimina la empresa del sistema.
6. El sistema muestra un mensaje al administrador indicando que la operación se ha realizado con éxito.

Flujo alternativo: Empresa vinculada a registro de otra tabla.

Línea 5: Si la empresa está vinculada a uno o más registros de otras tablas no podrá ser eliminada. En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de eliminar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Caso de uso – Definir Usuario

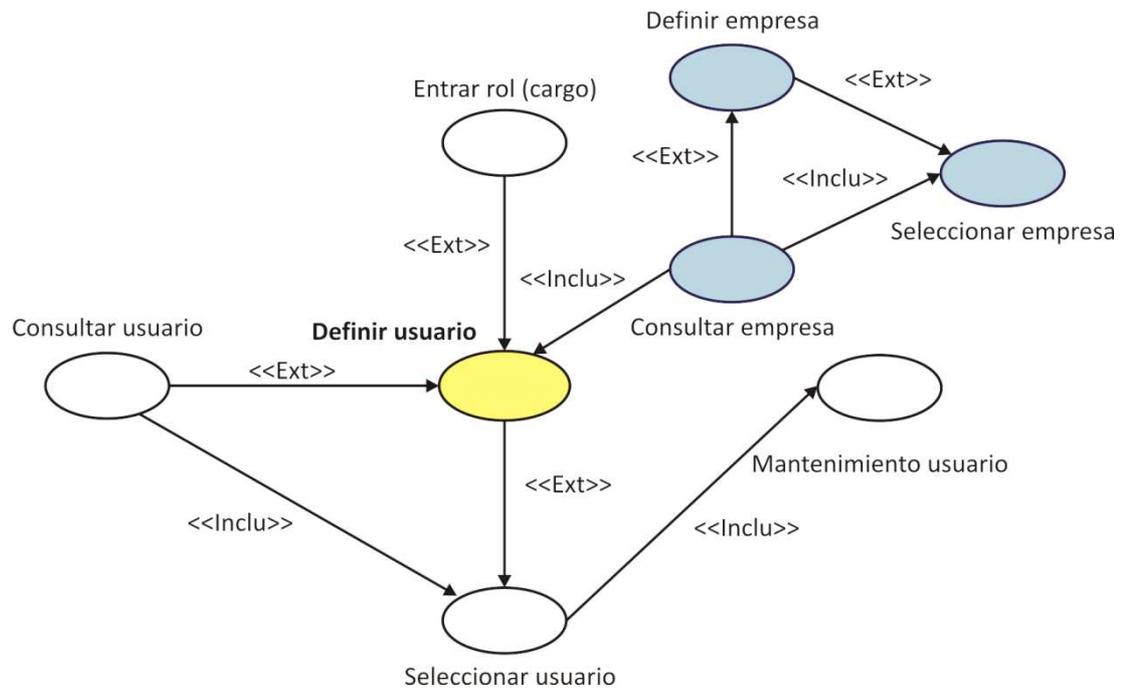


Gráfico 20. Definir usuario

Identificador: UC3.1

Actores: Administrador

Objetivos: Definir usuario en la intranet.

Condiciones previas: Ninguna

Condiciones posteriores: Existe un nuevo usuario en intranet.

Frecuencias de uso: Infrecuente. Este caso de uso es crítico, ya que si no se pueden crear los usuarios, estos no tendrán acceso a las funcionalidades del sistema.

Flujo Básico:

1. El caso de uso empieza cuando el administrador quiere crear un nuevo usuario.
2. El sistema mostrará una pantalla donde se muestran los campos que se deben llenar para la creación del usuario.

3. El administrador introduce la cédula, nombres, apellidos, dirección, teléfono, celular, contraseña, tipo de usuario (Rol) y se asigna las empresas a la que pertenece usuario.
4. El administrador indica al sistema que ya se ha acabado de introducir los datos aceptándolos.
5. El sistema crea y guarda los datos del nuevo usuario.
6. El sistema muestra al administrador un mensaje indicando que la operación se ha realizado con éxito.

Flujo alternativo: Conflicto de usuario ya existente.

Línea 4: Si el usuario ya existe con la misma identificación y empresa, el sistema muestra un mensaje de información al administrador y se vuelve al paso 3.

Flujo alternativo: Datos obligatorios incorrectos.

Línea 4: Si los datos introducidos se encuentran vacíos o incorrectos, el sistema muestra un mensaje de error al administrador y se vuelve al paso 3.

Flujo alternativo: Error inesperado BD.

Línea 5: La función de guardar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 4.

Identificador: UC3.2

Actores: Administrador

Objetivos: Modificar los datos de un usuario.

Condiciones previas: Existe el usuario que se quiere modificar.

Condiciones posteriores: Los datos del usuario son modificados con los nuevos datos.

Frecuencias de uso: Infrecuente. Se usa normalmente si se ha equivocado en algún dato o el usuario actualizó su información.

Flujo Básico:

1. El caso de uso empieza cuando el administrador quiere modificar los datos del usuario.
2. El sistema mostrará una pantalla con un listado de nombres, apellidos de usuario y estado (activo, inactivo) de usuarios.
3. El administrador selecciona el usuario y carga la información del usuario seleccionado.

4. El administrador modifica los datos que quiere actualizar, todos los datos son modificables excepto la cédula.
5. El administrador indica al sistema que ya ha acabado de modificar los datos aceptándolos.
6. El sistema hace las modificaciones en los datos del usuario y los actualiza.
7. El sistema muestra un mensaje al administrador indicando que la operación se ha realizado con éxito.

Flujo alternativo: Datos obligatorios incorrectos.

Línea 6: Si los datos introducidos se encuentran vacíos o incorrectos, el sistema muestra un mensaje de error al administrador y se vuelve al paso 5.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de modificar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Identificador: UC3.3

Actores: Administrador

Objetivos: El administrador quiere modificar la contraseña de un usuario.

Condiciones previas: Existe el usuario que se quiere modificar.

Condiciones posteriores: El usuario tiene una nueva contraseña asignada.

Frecuencias de uso: Infrecuente. Se utiliza si el usuario se ha olvidado de la contraseña y solicita al administrador que se lo modifique.

Flujo Básico:

1. El caso de uso empieza cuando el administrador quiere modificar la contraseña del usuario.
2. El sistema mostrará una pantalla con un listado de nombres, apellidos de usuario y estado (activo, inactivo) de usuarios.
3. El administrador selecciona el usuario y carga la información del usuario seleccionado.
4. El administrador introduce la nueva contraseña del usuario dos veces para validación.
5. El administrador indica al sistema que ya ha acabado de modificar los datos aceptándolos.
6. El sistema modifica la contraseña del usuario.

7. El sistema muestra un mensaje al administrador indicando que la operación se ha realizado con éxito.

Flujo alternativo: Las contraseñas no coinciden.

Línea 4: Si las contraseñas introducidas no coinciden el sistema muestra un mensaje indicando el error y se vuelve al paso 4.

Flujo alternativo: Datos obligatorios incorrectos.

Línea 4: Si los datos introducidos se encuentran vacíos o incorrectos, el sistema muestra un mensaje de error al administrador y se vuelve al paso 3.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de modificar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Identificador: UC3.4

Actores: Administrador

Objetivos: El administrador quiere dar de baja usuario del sistema.

Condiciones previas: Existe el usuario que se quiere eliminar.

Condiciones posteriores: El usuario dado de baja cambia de estado a inactivo.

Frecuencias de uso: Infrecuente. Normalmente se da de baja si el usuario deja de trabajar en la empresa.

Flujo Básico:

1. El caso de uso empieza cuando el administrador quiere dar de baja a un usuario.
2. El sistema mostrará una pantalla con un listado de nombres, apellidos de usuario y estado (activo, inactivo) de usuarios.
3. El administrador selecciona el usuario y carga información del mismo.
4. El sistema muestra los datos de usuario.
5. El administrador indica al sistema que se desea dar de baja a usuario cambiando su estado a inactivo. Actualiza la información.
6. El sistema da de baja al usuario del sistema cambiando su estado a inactivo.
7. El sistema muestra un mensaje al administrador indicando que la operación se ha realizado con éxito.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de dar baja puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Identificador: UC3.5

Actores: Administrador

Objetivos: El administrador quiere cambiar de cargo (rol) al usuario.

Condiciones previas: Existe el usuario que se quiere cambiar rol.

Condiciones posteriores: Se ha cambiado rol de usuario.

Frecuencias de uso: Infrecuente. Normalmente se cambia rol si existe una equivocación cuando se creó usuario.

Flujo Básico:

1. El caso de uso empieza cuando el administrador quiere cambiar de rol al usuario.
2. El sistema mostrará una pantalla con un listado de nombres, apellidos de usuario y estado (activo, inactivo) de usuarios.
3. El administrador selecciona el usuario y carga información del mismo.
4. El sistema muestra los datos de usuario.
5. El administrador cambia de rol a usuario y actualiza la información aceptándolo.
6. El sistema cambiando rol de usuario.
7. El sistema muestra un mensaje al administrador indicando que la operación se ha realizado con éxito.

Flujo alternativo: Datos obligatorios incorrectos.

Línea 5: Si los datos introducidos se encuentran vacíos o incorrectos, el sistema muestra un mensaje de error al administrador y se vuelve al paso 3.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de modificar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Caso de uso – Definir Marca

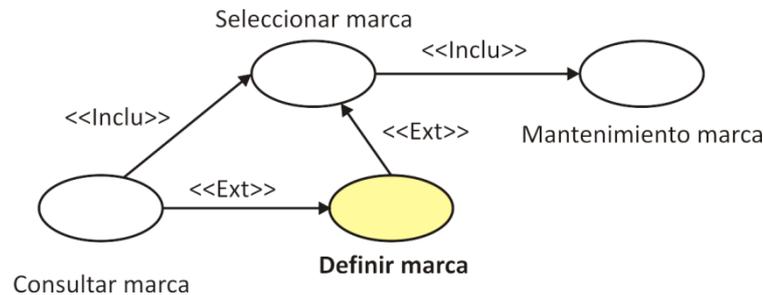


Gráfico 21. Definir marca

Identificador: UC4.1

Actores: Usuario

Objetivos: Definir marca.

Condiciones previas: Ninguna

Condiciones posteriores: Existe una nueva marca.

Frecuencias de uso: Frecuente. Este caso de uso es necesario para poder crear artículos y vehículos.

Flujo Básico:

1. El caso de uso empieza cuando el usuario quiere crear una nueva marca.
2. El sistema mostrará una pantalla donde se muestran los campos que se deben llenar para la creación de la marca.
3. El usuario introduce la descripción y selecciona tipo de marca.
4. El usuario indica al sistema que ya se ha acabado de introducir los datos aceptándolos.
5. El sistema crea y guarda los datos de la nueva marca.
6. El sistema muestra al usuario un mensaje indicando que la operación se ha realizado con éxito.

Flujo alternativo: Conflicto de marca ya existente.

Línea 4: Si la marca ya existe con la misma descripción y tipo, el sistema muestra un mensaje de información al usuario y se vuelve al paso 3.

Flujo alternativo: Datos obligatorios incorrectos.

Línea 4: Si los datos introducidos se encuentran vacíos o incorrectos, el sistema muestra un mensaje de error al usuario y se vuelve al paso 3.

Flujo alternativo: Error inesperado BD.

Línea 5: La función de guardar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 4.

Identificador: UC4.2

Actores: Usuario

Objetivos: Modificar los datos de una marca.

Condiciones previas: Existe la marca que se quiere modificar.

Condiciones posteriores: Los datos de la marca son modificados con los nuevos datos.

Frecuencias de uso: Infrecuente. Se usa normalmente si se ha equivocado en algún dato.

Flujo Básico:

1. El caso de uso empieza cuando el usuario quiere modificar los datos de la marca.
2. El sistema mostrará una pantalla con un listado de marcas con su descripción y tipo.
3. El usuario selecciona la marca y carga la información de la marca seleccionada.
4. El usuario modifica los datos que quiere actualizar, todos los datos son modificables.
5. El usuario indica al sistema que ya ha acabado de modificar los datos aceptándolos.
6. El sistema hace las modificaciones en los datos de la marca y los actualiza.
7. El sistema muestra un mensaje al usuario indicando que la operación se ha realizado con éxito.

Flujo alternativo: Datos obligatorios incorrectos.

Línea 6: Si los datos introducidos se encuentran vacíos o incorrectos, el sistema muestra un mensaje de error al usuario y se vuelve al paso 4.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de modificar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Identificador: UC4.3

Actores: Usuario

Objetivos: El usuario quiere eliminar marca.

Condiciones previas: Existe la marca que se quiere eliminar.

Condiciones posteriores: La marca ha sido eliminada del sistema.

Frecuencias de uso: Infrecuente. Normalmente se elimina si hubo error al ingresar marca.

Flujo Básico:

1. El caso de uso empieza cuando el usuario quiere eliminar marca.
2. El sistema mostrará una pantalla con un listado de marcas con su descripción y tipo.
3. El usuario selecciona la marca.
4. El usuario indica al sistema que desea eliminar la marca aceptando su eliminación.
5. El sistema elimina la marca del sistema.
6. El sistema muestra un mensaje al usuario indicando que la operación se ha realizado con éxito.

Flujo alternativo: Marca vinculada a registro de otra tabla.

Línea 5: Si la marca está vinculada a uno o más registros de otras tablas no podrá ser eliminada. En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de eliminar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Caso de uso – Definir Modelo

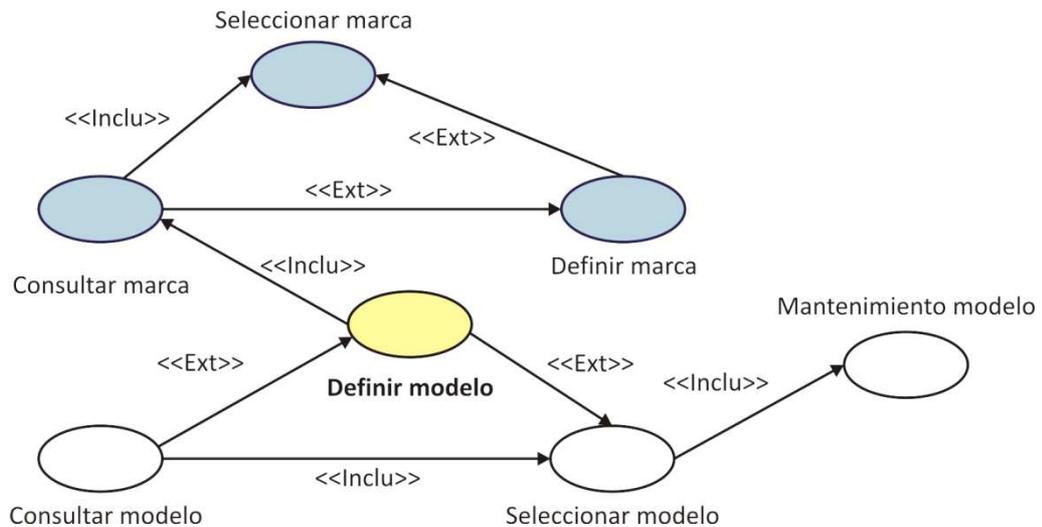


Gráfico 22. Definir modelo

Identificador: UC5.1

Actores: Usuario

Objetivos: Definir modelo.

Condiciones previas: Marca debe existir.

Condiciones posteriores: Existe un nuevo modelo.

Frecuencias de uso: Frecuente. Este caso de uso es necesario para poder crear artículos y vehículos.

Flujo Básico:

1. El caso de uso empieza cuando el usuario quiere crear un nuevo modelo.
2. El sistema mostrará una pantalla donde se muestran los campos que se deben llenar para la creación del modelo.
3. El usuario introduce la descripción del modelo y selecciona la marca a la que pertenece.
4. El usuario indica al sistema que ya se ha acabado de introducir los datos aceptándolos.
5. El sistema crea y guarda los datos del nuevo modelo.
6. El sistema muestra al usuario un mensaje indicando que la operación se ha realizado con éxito.

Flujo alternativo: Conflicto de modelo ya existente.

Línea 4: Si el modelo ya existe con la misma descripción y marca, el sistema muestra un mensaje de información al usuario y se vuelve al paso 3.

Flujo alternativo: Datos obligatorios incorrectos.

Línea 4: Si los datos introducidos se encuentran vacíos o incorrectos, el sistema muestra un mensaje de error al usuario y se vuelve al paso 3.

Flujo alternativo: Error inesperado BD.

Línea 5: La función de guardar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 4.

Identificador: UC5.2

Actores: Usuario

Objetivos: Modificar los datos de un modelo.

Condiciones previas: Existe el modelo que se quiere modificar.

Condiciones posteriores: Los datos del modelo son modificados con los nuevos datos.

Frecuencias de uso: Infrecuente. Se usa normalmente si se ha equivocado en algún dato.

Flujo Básico:

1. El caso de uso empieza cuando el usuario quiere modificar los datos del modelo.
2. El sistema mostrará una pantalla con un listado de modelos con su descripción y marca.
3. El usuario selecciona el modelo y carga la información del modelo seleccionado.
4. El usuario modifica los datos que quiere actualizar, todos los datos son modificables.
5. El usuario indica al sistema que ya ha acabado de modificar los datos aceptándolos.
6. El sistema hace las modificaciones en los datos del modelo y los actualiza.
7. El sistema muestra un mensaje al usuario indicando que la operación se ha realizado con éxito.

Flujo alternativo: Todos los datos del modelo modificado coinciden con otro registro.

Línea 6: Si los datos introducidos se encuentran coincidentes con otro registro, el sistema muestra un mensaje de error al usuario y se vuelve al paso 4.

Flujo alternativo: Datos obligatorios incorrectos.

Línea 6: Si los datos introducidos se encuentran vacíos o incorrectos, el sistema muestra un mensaje de error al usuario y se vuelve al paso 4.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de modificar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Identificador: UC5.3

Actores: Usuario

Objetivos: El usuario quiere eliminar modelo.

Condiciones previas: Existe el modelo que se quiere eliminar.

Condiciones posteriores: El modelo ha sido eliminado del sistema.

Frecuencias de uso: Infrecuente. Normalmente se elimina si hubo error al ingresar modelo.

Flujo Básico:

1. El caso de uso empieza cuando el usuario quiere eliminar modelo.
2. El sistema mostrará una pantalla con un listado de modelos con su descripción y marca.
3. El usuario selecciona el modelo.
4. El usuario indica al sistema que desea eliminar el modelo aceptando su eliminación.
5. El sistema elimina el modelo del sistema.
6. El sistema muestra un mensaje al usuario indicando que la operación se ha realizado con éxito.

Flujo alternativo: Modelo vinculado a registro de otra tabla.

Línea 5: Si el modelo está vinculado a uno o más registros de otras tablas no podrá ser eliminado. En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de eliminar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Caso de uso – Definir Artículo

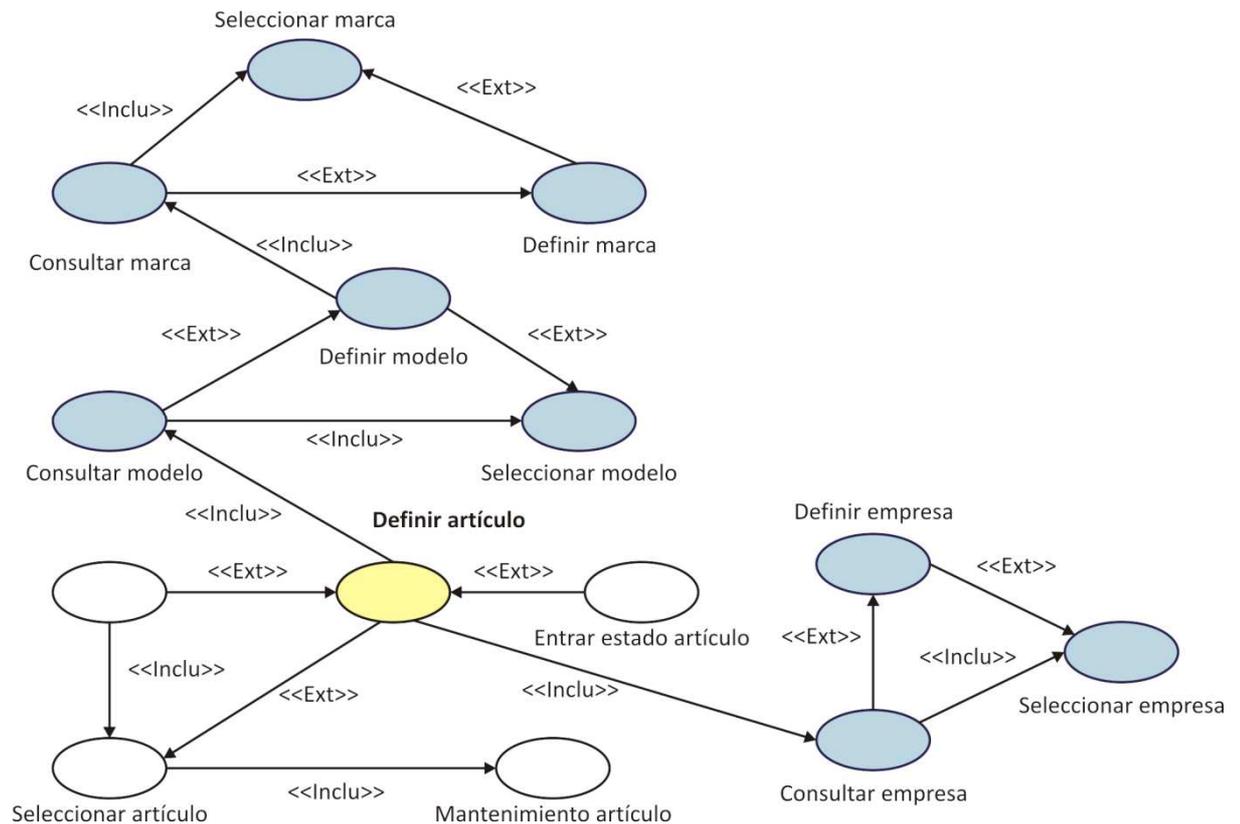


Gráfico 23. Definir artículo

Identificador: UC6.1

Actores: Usuario

Objetivos: Definir artículo.

Condiciones previas: Marca y modelo deben existir.

Condiciones posteriores: Existe un nuevo artículo.

Frecuencias de uso: Frecuente. Este caso de uso es necesario para poder realizar ordenes de entrega - recepción.

Flujo Básico:

1. El caso de uso empieza cuando el usuario quiere crear un nuevo artículo.

2. El sistema mostrará una pantalla donde se muestran los campos que se deben llenar para la creación del artículo.
3. El usuario introduce el nombre, descripción, cantidad, precio y selecciona la marca, modelo y empresa a la que pertenece.
4. El usuario indica al sistema que ya se ha acabado de introducir los datos aceptándolos.
5. El sistema crea y guarda los datos del nuevo artículo.
6. El sistema muestra al usuario un mensaje indicando que la operación se ha realizado con éxito.

Flujo alternativo: Conflicto de artículo ya existente.

Línea 4: Si el artículo ya existe con el mismo nombre y modelo, el sistema muestra un mensaje de información al usuario y se vuelve al paso 3.

Flujo alternativo: Datos obligatorios incorrectos.

Línea 4: Si los datos introducidos se encuentran vacíos o incorrectos, el sistema muestra un mensaje de error al usuario y se vuelve al paso 3.

Flujo alternativo: Error inesperado BD.

Línea 5: La función de guardar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 4.

Identificador: UC6.2

Actores: Usuario

Objetivos: Modificar los datos de un artículo.

Condiciones previas: Existe el artículo que se quiere modificar.

Condiciones posteriores: Los datos del artículo son modificados con los nuevos datos.

Frecuencias de uso: Infrecuente. Se usa normalmente si se ha equivocado en algún dato.

Flujo Básico:

1. El caso de uso empieza cuando el usuario quiere modificar los datos del artículo.
2. El sistema mostrará una pantalla con un listado de artículos con su información.

3. El usuario selecciona el artículo y carga la información del artículo seleccionado.
4. El usuario modifica los datos que quiere actualizar, todos los datos son modificables.
5. El usuario indica al sistema que ya ha acabado de modificar los datos aceptándolos.
6. El sistema hace las modificaciones en los datos del modelo y los actualiza.
7. El sistema muestra un mensaje al usuario indicando que la operación se ha realizado con éxito.

Flujo alternativo: Los datos de nombre y modelo del artículo modificado coinciden con otro registro.

Línea 6: Si los datos introducidos se encuentran coincidentes con otro registro, el sistema muestra un mensaje de error al usuario y se vuelve al paso 4.

Flujo alternativo: Datos obligatorios incorrectos.

Línea 6: Si los datos introducidos se encuentran vacíos o incorrectos, el sistema muestra un mensaje de error al usuario y se vuelve al paso 4.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de modificar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Identificador: UC6.3

Actores: Usuario

Objetivos: El usuario quiere eliminar un artículo.

Condiciones previas: Existe el artículo que se quiere eliminar.

Condiciones posteriores: El artículo ha sido eliminado del sistema.

Frecuencias de uso: Infrecuente. Normalmente se elimina si hubo error al ingresar artículo.

Flujo Básico:

1. El caso de uso empieza cuando el usuario quiere eliminar un artículo.
2. El sistema mostrará una pantalla con un listado de artículos con su información.
3. El usuario selecciona el artículo.

4. El usuario indica al sistema que desea eliminar el artículo aceptando su eliminación.

5. El sistema elimina el artículo del sistema.

6. El sistema muestra un mensaje al usuario indicando que la operación se ha realizado con éxito.

Flujo alternativo: Artículo vinculado a registro de otra tabla.

Línea 5: Si el artículo está vinculado a uno o más registros de otras tablas no podrá ser eliminado. En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de eliminar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Caso de uso – Definir Cliente

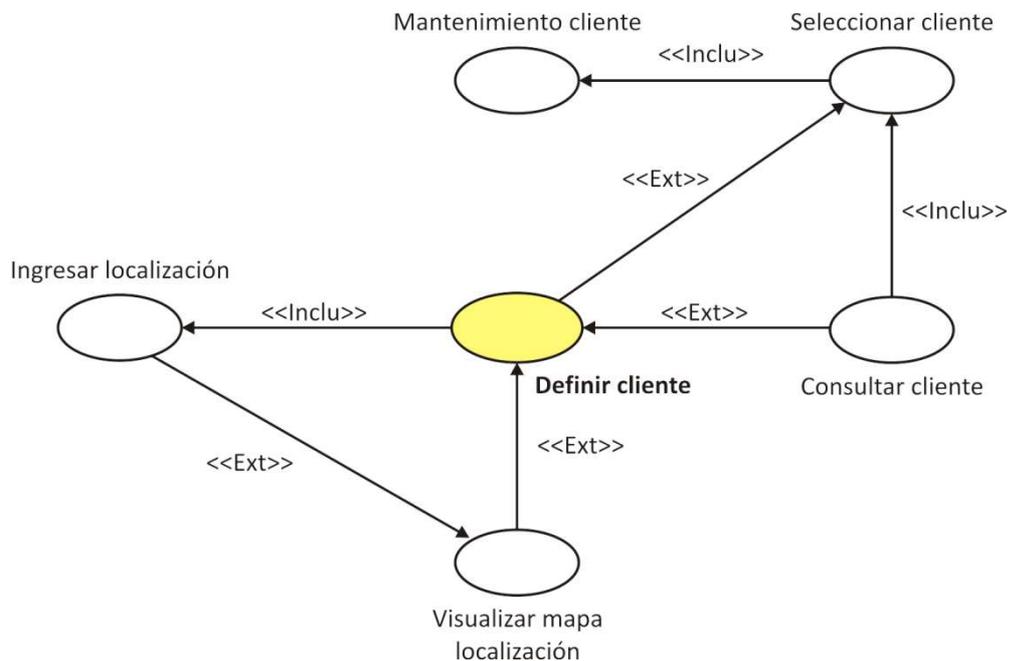


Gráfico 24. Definir cliente

Identificador: UC7.1

Actores: Usuario

Objetivos: Definir cliente.

Condiciones previas: Ninguna

Condiciones posteriores: Existe un nuevo cliente.

Frecuencias de uso: Frecuente. Este caso de uso es necesario para poder realizar ventas, facturas, ordenes de entrega - recepción.

Flujo Básico:

1. El caso de uso empieza cuando el usuario quiere crear un nuevo cliente.
2. El sistema mostrará una pantalla donde se muestran los campos que se deben llenar para la creación del cliente.
3. El usuario introduce la cédula, nombres, apellidos, dirección (calle principal, calle secundaria, nro), teléfono y celular.
4. El usuario indica al sistema que ya se ha acabado de introducir los datos aceptándolos.
5. El sistema crea y guarda los datos del nuevo cliente.
6. La base de datos activa el trigger para la creación geográfica de la localización a través de las calles.
7. El sistema muestra al usuario un mensaje indicando que la operación se ha realizado con éxito.

Flujo alternativo: Conflicto de cliente ya existente.

Línea 4: Si el cliente ya existe con la misma cédula, el sistema muestra un mensaje de información al usuario y se vuelve al paso 3.

Flujo alternativo: Datos obligatorios incorrectos.

Línea 4: Si los datos introducidos se encuentran vacíos o incorrectos, el sistema muestra un mensaje de error al usuario y se vuelve al paso 3.

Flujo alternativo: Error inesperado BD.

Línea 5: La función de guardar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 4.

Identificador: UC7.2

Actores: Usuario

Objetivos: Modificar los datos de un cliente.

Condiciones previas: Existe el cliente que se quiere modificar.

Condiciones posteriores: Los datos del cliente son modificados con los nuevos datos.

Frecuencias de uso: Infrecuente. Se usa normalmente si se ha equivocado en algún dato o si el cliente actualizó sus datos.

Flujo Básico:

1. El caso de uso empieza cuando el usuario quiere modificar los datos del cliente.
2. El sistema mostrará una pantalla con un listado de clientes con su información.
3. El usuario selecciona el cliente y carga la información del cliente seleccionado.
4. El usuario modifica los datos que quiere actualizar, todos los datos son modificables a excepción de la cédula.
5. El usuario indica al sistema que ya ha acabado de modificar los datos aceptándolos.
6. El sistema hace las modificaciones en los datos del cliente y los actualiza.
7. El sistema muestra un mensaje al usuario indicando que la operación se ha realizado con éxito.

Flujo alternativo: Datos obligatorios incorrectos.

Línea 6: Si los datos introducidos se encuentran vacíos o incorrectos, el sistema muestra un mensaje de error al usuario y se vuelve al paso 4.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de modificar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Identificador: UC7.3

Actores: Usuario

Objetivos: El usuario quiere eliminar un cliente.

Condiciones previas: Existe el cliente que se quiere eliminar.

Condiciones posteriores: El cliente ha sido eliminado del sistema.

Frecuencias de uso: Infrecuente. Normalmente se elimina si hubo error al ingresar cliente.

Flujo Básico:

1. El caso de uso empieza cuando el usuario quiere eliminar un cliente.

2. El sistema mostrará una pantalla con un listado de clientes con su información.
3. El usuario selecciona el cliente.
4. El usuario indica al sistema que desea eliminar el cliente aceptando su eliminación.
5. El sistema elimina el cliente del sistema.
6. El sistema muestra un mensaje al usuario indicando que la operación se ha realizado con éxito.

Flujo alternativo: Cliente vinculado a registro de otra tabla.

Línea 5: Si el cliente está vinculado a uno o más registros de otras tablas no podrá ser eliminado. En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de eliminar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Identificador: UC7.4

Actores: Usuario

Objetivos: El usuario quiere visualizar la localización del cliente en el mapa.

Condiciones previas: Existe el cliente que se quiere visualizar localización.

Condiciones posteriores: Visualización de la localización en mapa del cliente.

Frecuencias de uso: Infrecuente. Normalmente se utiliza en casos de consulta.

Flujo Básico:

1. El caso de uso empieza cuando el usuario quiere visualizar localización en mapa.
2. El sistema mostrará una pantalla con un listado de clientes con su información.
3. El usuario selecciona el cliente.
4. El usuario indica al sistema que desea visualizar el cliente.
5. El sistema muestra una pantalla con la visualización de la localización del cliente en un mapa.

Flujo alternativo: Localización errónea.

5. El sistema crea y guarda los datos del nuevo empleado.

6. El sistema muestra al usuario un mensaje indicando que la operación se ha realizado con éxito.

Flujo alternativo: Conflicto de empleado ya existente.

Línea 4: Si el empleado ya existe con la misma cédula, el sistema muestra un mensaje de información al usuario y se vuelve al paso 3.

Flujo alternativo: Datos obligatorios incorrectos.

Línea 4: Si los datos introducidos se encuentran vacíos o incorrectos, el sistema muestra un mensaje de error al usuario y se vuelve al paso 3.

Flujo alternativo: Error inesperado BD.

Línea 5: La función de guardar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 4.

Identificador: UC8.2

Actores: Usuario

Objetivos: Modificar los datos de un empleado.

Condiciones previas: Existe el empleado que se quiere modificar.

Condiciones posteriores: Los datos del empleado son modificados con los nuevos datos.

Frecuencias de uso: Infrecuente. Se usa normalmente si se ha equivocado en algún dato o si el empleado actualizó sus datos.

Flujo Básico:

1. El caso de uso empieza cuando el usuario quiere modificar los datos del empleado.

2. El sistema mostrará una pantalla con un listado de empleados con su información.

3. El usuario selecciona el empleado y carga la información del empleado seleccionado.

4. El usuario modifica los datos que quiere actualizar, todos los datos son modificables a excepción de la cédula.

5. El usuario indica al sistema que ya ha acabado de modificar los datos aceptándolos.

6. El sistema hace las modificaciones en los datos del empleado y los actualiza.

7. El sistema muestra un mensaje al usuario indicando que la operación se ha realizado con éxito.

Flujo alternativo: Datos obligatorios incorrectos.

Línea 6: Si los datos introducidos se encuentran vacíos o incorrectos, el sistema muestra un mensaje de error al usuario y se vuelve al paso 4.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de modificar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Identificador: UC8.3

Actores: Usuario

Objetivos: El usuario quiere eliminar un empleado.

Condiciones previas: Existe el empleado que se quiere eliminar.

Condiciones posteriores: El empleado ha sido eliminado del sistema.

Frecuencias de uso: Infrecuente. Normalmente se elimina si hubo error al ingresar empleado.

Flujo Básico:

1. El caso de uso empieza cuando el usuario quiere eliminar un empleado.
2. El sistema mostrará una pantalla con un listado de empleados con su información.
3. El usuario selecciona el empleado.
4. El usuario indica al sistema que desea eliminar el empleado aceptando su eliminación.
5. El sistema elimina el empleado del sistema.
6. El sistema muestra un mensaje al usuario indicando que la operación se ha realizado con éxito.

Flujo alternativo: Empleado vinculado a registro de otra tabla.

Línea 5: Si el empleado está vinculado a uno o más registros de otras tablas no podrá ser eliminado. En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de eliminar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Caso de uso – Definir GPS

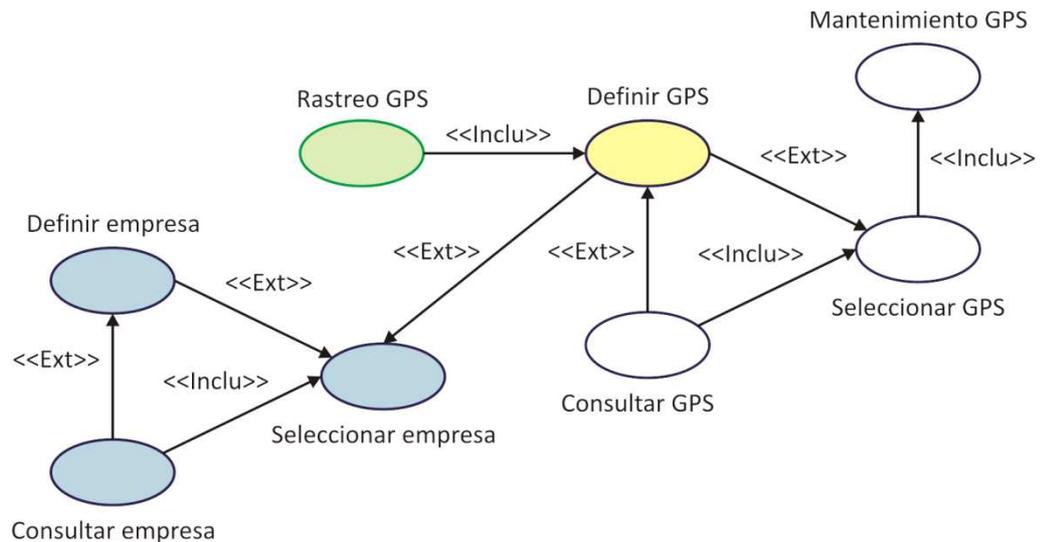


Gráfico 26. Definir GPS

Identificador: UC9.1

Actores: Usuario

Objetivos: Definir GPS.

Condiciones previas: Empresa debe existir.

Condiciones posteriores: Existe un nuevo GPS.

Frecuencias de uso: Frecuente. Este caso de uso es crítico ya que sin él no podría funcionar el rastreo y localización de vehículos.

Flujo Básico:

1. El caso de uso empieza cuando el usuario quiere crear un nuevo GPS.
2. El sistema mostrará una pantalla donde se muestran los campos que se deben llenar para la creación del GPS.
3. El usuario introduce el ID, SIM1, SIM2, IP del servidor asociado, selecciona la empresa a la que pertenece, así como su modelo y marca.
4. El usuario indica al sistema que ya se ha acabado de introducir los datos aceptándolos.

5. El sistema crea y guarda los datos del nuevo GPS.

6. El sistema muestra al usuario un mensaje indicando que la operación se ha realizado con éxito.

Flujo alternativo: Conflicto de GPS ya existente.

Línea 4: Si el GPS ya existe con el mismo ID, el sistema muestra un mensaje de información al usuario y se vuelve al paso 3.

Flujo alternativo: Datos obligatorios incorrectos.

Línea 4: Si los datos introducidos se encuentran vacíos o incorrectos, el sistema muestra un mensaje de error al usuario y se vuelve al paso 3.

Flujo alternativo: Error inesperado BD.

Línea 5: La función de guardar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 4.

Identificador: UC9.2

Actores: Usuario

Objetivos: Modificar los datos de un GPS.

Condiciones previas: Existe el GPS que se quiere modificar.

Condiciones posteriores: Los datos del GPS son modificados con los nuevos datos.

Frecuencias de uso: Infrecuente. Se usa normalmente si se ha equivocado en algún dato o si se actualizó sus datos de los SIMS o IP del GPS.

Flujo Básico:

1. El caso de uso empieza cuando el usuario quiere modificar los datos del GPS.

2. El sistema mostrará una pantalla con un listado de GPS con su información.

3. El usuario selecciona el GPS y carga la información del GPS seleccionado.

4. El usuario modifica los datos que quiere actualizar, todos los datos son modificables a excepción del ID.

5. El usuario indica al sistema que ya ha acabado de modificar los datos aceptándolos.

6. El sistema hace las modificaciones en los datos del GPS y los actualiza.

7. El sistema muestra un mensaje al usuario indicando que la operación se ha realizado con éxito.

Flujo alternativo: Datos obligatorios incorrectos.

Línea 6: Si los datos introducidos se encuentran vacíos o incorrectos, el sistema muestra un mensaje de error al usuario y se vuelve al paso 4.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de modificar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Identificador: UC9.3

Actores: Usuario

Objetivos: El usuario quiere eliminar un GPS.

Condiciones previas: Existe el GPS que se quiere eliminar.

Condiciones posteriores: El GPS ha sido eliminado del sistema.

Frecuencias de uso: Infrecuente. Normalmente se elimina si hubo error al ingresar GPS.

Flujo Básico:

1. El caso de uso empieza cuando el usuario quiere eliminar un GPS.
2. El sistema mostrará una pantalla con un listado de GPS con su información.
3. El usuario selecciona el GPS.
4. El usuario indica al sistema que desea eliminar el GPS aceptando su eliminación.
5. El sistema elimina el GPS del sistema.
6. El sistema muestra un mensaje al usuario indicando que la operación se ha realizado con éxito.

Flujo alternativo: GPS vinculado a registro de otra tabla.

Línea 5: Si el GPS está vinculado a uno o más registros de otras tablas no podrá ser eliminado. En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de eliminar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Identificador: UC9.4

Actores: Actor medio (Protocolo de comunicación)

Objetivos: Rastreo de vehículo.

Condiciones previas: El GPS se encuentra conectado al vehículo y funcionando; la aplicación especializada de celular se encuentra funcionando y el celular se encuentra conectado al servidor.

Condiciones posteriores: El GPS envía información de la posición actual del vehículo.

Frecuencias de uso: Frecuente. Funciona siempre para el rastreo de vehículos y determinación de rutas óptimas.

Flujo Básico:

1. El caso de uso empieza cuando se dispara un evento previamente configurado en el GPS.
2. Un celular conectado al servidor funcionando a través de una aplicación especializada obtiene los datos del reporte y desglosa la información para ser grabada en la base de datos.
3. El sistema funciona continuamente y transparente al usuario.

Flujo alternativo: El GPS no envía información del reporte.

Línea 2: Si el GPS no está enviando información se lo debe reconfigurar o asegurar de que esté conectado al vehículo y encendido. En tal caso, se debe retornar al paso 1.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de recepción información puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 1.

Caso de uso – Definir Calles

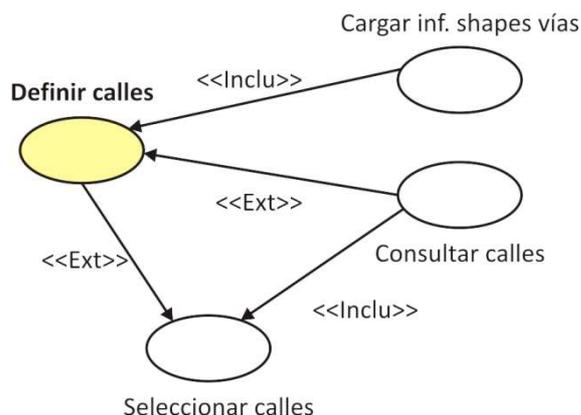


Gráfico 27. Definir calles

Identificador: UC10.1

Actores: Usuario

Objetivos: Definir calles.

Condiciones previas: Tener el shapefile de las vías del centro histórico de Cuenca.

Condiciones posteriores: Existe una nueva calle.

Frecuencias de uso: Infrecuente. Este caso de uso es crítico ya que sin él no podría funcionar el rastreo y localización de vehículos.

Flujo Básico:

1. El caso de uso empieza cuando el usuario quiere crear una nueva calle.
2. El sistema mostrará una pantalla donde se muestran los campos que se deben llenar para la creación de la calle.
3. El usuario introduce el nombre y selecciona el tipo de calle (Autopista, Principal, Secundaria).
4. El usuario indica al sistema que ya se ha acabado de introducir los datos aceptándolos.
5. El sistema crea y guarda los datos de la nueva calle.
6. El sistema muestra al usuario un mensaje indicando que la operación se ha realizado con éxito.

Flujo alternativo: Conflicto de calle ya existente.

Línea 4: Si la calle ya existe, el sistema muestra un mensaje de información al usuario y se vuelve al paso 3.

Flujo alternativo: Datos obligatorios incorrectos.

Línea 4: Si los datos introducidos se encuentran vacíos o incorrectos, el sistema muestra un mensaje de error al usuario y se vuelve al paso 3.

Flujo alternativo: Error inesperado BD.

Línea 5: La función de guardar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 4.

Identificador: UC10.2

Actores: Usuario

Objetivos: Modificar los datos de una calle.

Condiciones previas: Existe la calle que se quiere modificar.

Condiciones posteriores: Los datos de la calle son modificados con los nuevos datos.

Frecuencias de uso: Infrecuente. Se usa normalmente si se ha equivocado en algún dato o si se actualizó los datos de la calle como es su tipo.

Flujo Básico:

1. El caso de uso empieza cuando el usuario quiere modificar los datos de la calle.
2. El sistema mostrará una pantalla con un listado de calles con su información.
3. El usuario selecciona la calle y carga la información de la calle seleccionada.
4. El usuario modifica los datos que quiere actualizar, todos los datos son modificables.
5. El usuario indica al sistema que ya ha acabado de modificar los datos aceptándolos.
6. El sistema hace las modificaciones en los datos de la calle y los actualiza.
7. El sistema muestra un mensaje al usuario indicando que la operación se ha realizado con éxito.

Flujo alternativo: Datos obligatorios incorrectos.

Línea 6: Si los datos introducidos se encuentran vacíos o incorrectos, el sistema muestra un mensaje de error al usuario y se vuelve al paso 4.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de modificar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Identificador: UC10.3

Actores: Usuario

Objetivos: El usuario quiere eliminar una calle.

Condiciones previas: Existe la calle que se quiere eliminar.

Condiciones posteriores: La calle ha sido eliminada del sistema.

Frecuencias de uso: Infrecuente. Normalmente se elimina si hubo error al ingresar calle.

Flujo Básico:

1. El caso de uso empieza cuando el usuario quiere eliminar una calle.
2. El sistema mostrará una pantalla con un listado de calles con su información.
3. El usuario selecciona la calle.
4. El usuario indica al sistema que desea eliminar la calle aceptando su eliminación.
5. El sistema elimina la calle del sistema.
6. El sistema muestra un mensaje al usuario indicando que la operación se ha realizado con éxito.

Flujo alternativo: Calle vinculada a registro de otra tabla.

Línea 5: Si la calle está vinculada a uno o más registros de otras tablas no podrá ser eliminada. En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de eliminar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Caso de uso – Generación Cartográfica

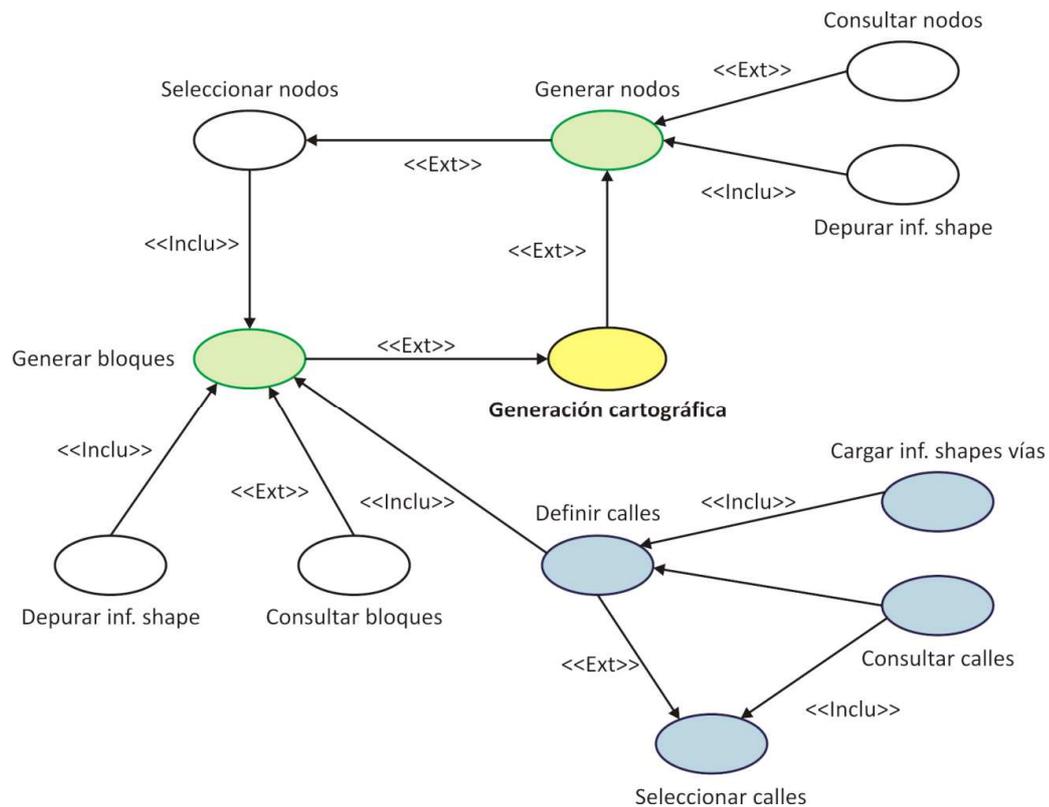


Gráfico 28. Generación cartográfica

Identificador: UC11.1

Actores: Administrador

Objetivos: Generar información cartográfica (bloques y nodos).

Condiciones previas: Tener el shapefile de las vías del centro histórico de Cuenca depurado y las calles deben existir.

Condiciones posteriores: Bloques y nodos de calles generados.

Frecuencias de uso: Infrecuente. Este caso de uso es crítico ya que sin él no podría funcionar el rastreo y localización de vehículos. Se lo realiza una sola vez.

Flujo Básico:

1. El caso de uso empieza cuando el administrador tiene que generar la información necesaria de bloques y nodos.
2. Se generará un archivo .txt o .csv de la información correspondiente a los bloques y nodos depurados en las capas shapefile generadas a partir del shape de vías de la zona de estudio.

3. El administrador importará los datos de los bloques y de los nodos a través del archivo .txt o .csv a la base de datos en PostgreSQL.

4. El sistema muestra al administrador un mensaje indicando que la operación se ha realizado con éxito.

Flujo alternativo: Conflicto de bloque o nodo ya existente.

Línea 4: Si el nodo o bloque ya existe, la base de datos muestra un mensaje de información al administrador y se vuelve al paso 3.

Flujo alternativo: Error inesperado BD.

Línea 4: La función de guardar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 3.

Identificador: UC11.2

Actores: Administrador

Objetivos: Consultar los datos de un bloque.

Condiciones previas: Existe el bloque que se quiere consultar.

Condiciones posteriores: Los datos del bloque son mostrados.

Frecuencias de uso: Infrecuente. Se usa solo para información al administrador.

Flujo Básico:

1. El caso de uso empieza cuando el administrador quiere consultar los datos de un bloque.

2. El sistema mostrará una pantalla con un listado de bloques con su información.

Flujo alternativo: Información de bloques no ha sido cargada a la BD.

Si no existe información de bloques en la BD se deberá realizar el caso de uso UC10.1.

Flujo alternativo: Error inesperado BD.

Línea 2: La función de consulta puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Identificador: UC11.3

Actores: Administrador

Objetivos: Consultar los datos de un nodo.

Condiciones previas: Existe el nodo que se quiere consultar.

Condiciones posteriores: Los datos del nodo son mostrados.

Frecuencias de uso: Infrecuente. Se usa solo para información al administrador.

Flujo Básico:

1. El caso de uso empieza cuando el administrador quiere consultar los datos de un nodo.
2. El sistema mostrará una pantalla con un listado de nodos con su información.

Flujo alternativo: Información de nodos no ha sido cargada a la BD.

Si no existe información de nodos en la BD se deberá realizar el caso de uso UC11.1.

Flujo alternativo: Error inesperado BD.

Línea 2: La función de consulta puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Caso de uso – Definir Vehículo

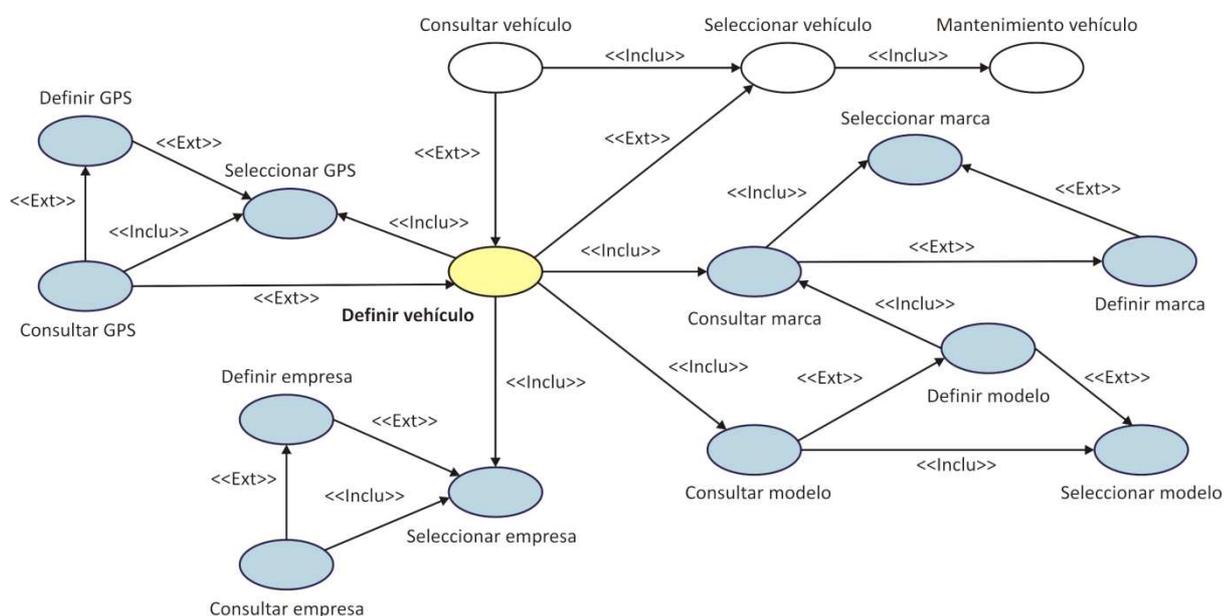


Gráfico 29. Definir vehículo

Identificador: UC12.1

Actores: Administrador

Objetivos: Definir vehículo.

Condiciones previas: Marca, modelo, GPS y empresa deben existir.

Condiciones posteriores: Existe un nuevo vehículo.

Frecuencias de uso: Frecuente. Este caso de uso es necesario para poder realizar rastreo de vehículo e inferencia de rutas óptimas.

Flujo Básico:

1. El caso de uso empieza cuando el administrador quiere crear un nuevo vehículo.
2. El sistema mostrará una pantalla donde se muestran los campos que se deben llenar para la creación del vehículo.
3. El administrador introduce la placa, color, número de motor y selecciona la marca, modelo, GPS y empresa a la que pertenece.
4. El administrador indica al sistema que ya se ha acabado de introducir los datos aceptándolos.
5. El sistema crea y guarda los datos del nuevo vehículo.
6. El sistema muestra al administrador un mensaje indicando que la operación se ha realizado con éxito.

Flujo alternativo: Conflicto de vehículo ya existente.

Línea 4: Si la placa del vehículo ya existe, el sistema muestra un mensaje de información al administrador y se vuelve al paso 3.

Flujo alternativo: Datos obligatorios incorrectos.

Línea 4: Si los datos introducidos se encuentran vacíos o incorrectos, el sistema muestra un mensaje de error al administrador y se vuelve al paso 3.

Flujo alternativo: Error inesperado BD.

Línea 5: La función de guardar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 4.

Identificador: UC12.2

Actores: Administrador

Objetivos: Modificar los datos de un vehículo.

Condiciones previas: Existe el vehículo que se quiere modificar.

Condiciones posteriores: Los datos del vehículo son modificados con los nuevos datos.

Frecuencias de uso: Infrecuente. Se usa normalmente si se ha equivocado en algún dato.

Flujo Básico:

1. El caso de uso empieza cuando el administrador quiere modificar los datos del vehículo.
2. El sistema mostrará una pantalla con un listado de vehículos con su información.
3. El administrador selecciona el vehículo y carga la información del vehículo seleccionado.
4. El administrador modifica los datos que quiere actualizar, todos los datos son modificables a excepción de la placa.
5. El administrador indica al sistema que ya ha acabado de modificar los datos aceptándolos.
6. El sistema hace las modificaciones en los datos del vehículo y los actualiza.
7. El sistema muestra un mensaje al administrador indicando que la operación se ha realizado con éxito.

Flujo alternativo: Datos obligatorios incorrectos.

Línea 6: Si los datos introducidos se encuentran vacíos o incorrectos, el sistema muestra un mensaje de error al administrador y se vuelve al paso 4.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de modificar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Identificador: UC12.3

Actores: Administrador

Objetivos: El administrador quiere eliminar un vehículo.

Condiciones previas: Existe el vehículo que se quiere eliminar.

Condiciones posteriores: El vehículo ha sido eliminado del sistema.

Frecuencias de uso: Infrecuente. Normalmente se elimina si hubo error al ingresar vehículo.

Flujo Básico:

1. El caso de uso empieza cuando el administrador quiere eliminar un vehículo.

2. El sistema mostrará una pantalla con un listado de vehículos con su información.
3. El administrador selecciona el vehículo.
4. El administrador indica al sistema que desea eliminar el vehículo aceptando su eliminación.
5. El sistema elimina el vehículo del sistema.
6. El sistema muestra un mensaje al administrador indicando que la operación se ha realizado con éxito.

Flujo alternativo: Vehículo vinculado a registro de otra tabla.

Línea 5: Si el vehículo está vinculado a uno o más registros de otras tablas no podrá ser eliminado. En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de eliminar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Caso de uso – Definir Bodega

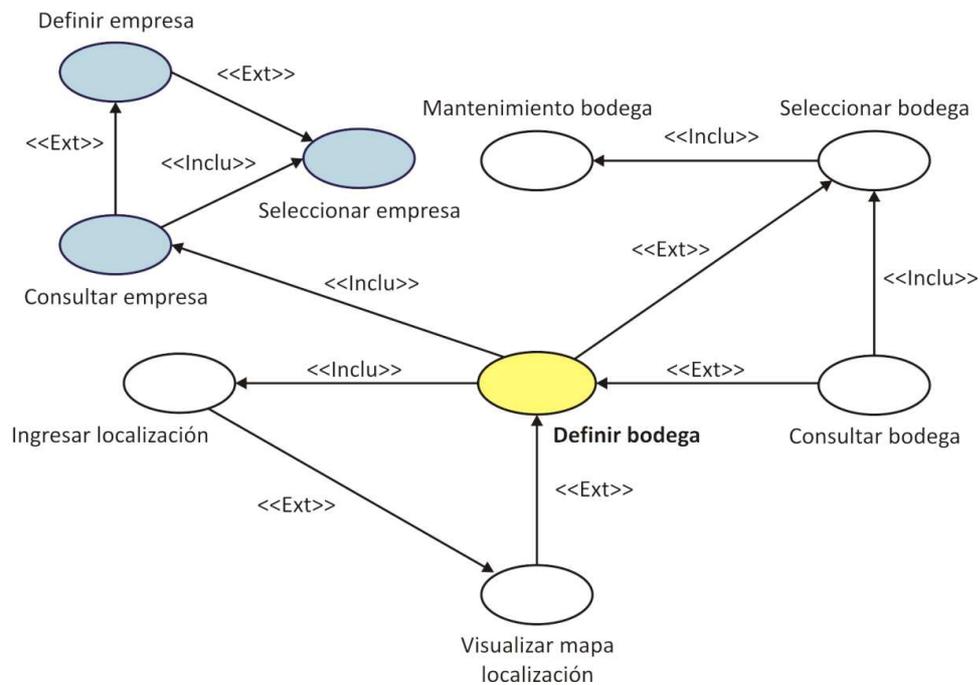


Gráfico 30. Definir bodega

Identificador: UC13.1

Actores: Administrador

Objetivos: Definir bodega.

Condiciones previas: Empresa debe existir.

Condiciones posteriores: Existe una nueva bodega.

Frecuencias de uso: Frecuente. Este caso de uso es necesario para poder realizar ventas.

Flujo Básico:

1. El caso de uso empieza cuando el administrador quiere crear una nueva bodega.
2. El sistema mostrará una pantalla donde se muestran los campos que se deben llenar para la creación de la bodega.
3. El administrador introduce el nombre, sigla, teléfono, selecciona dirección (calle principal, calle secundaria, nro.) y empresa.
4. El administrador indica al sistema que ya se ha acabado de introducir los datos aceptándolos.
5. El sistema crea y guarda los datos de la nueva bodega.
6. La base de datos activa el trigger para la creación geográfica de la localización a través de las calles.
7. El sistema muestra al administrador un mensaje indicando que la operación se ha realizado con éxito.

Flujo alternativo: Conflicto de bodega ya existente.

Línea 4: Si la bodega ya existe con el mismo nombre y empresa, el sistema muestra un mensaje de información al administrador y vuelve al paso 3.

Flujo alternativo: Datos obligatorios incorrectos.

Línea 4: Si los datos introducidos se encuentran vacíos o incorrectos, el sistema muestra un mensaje de error al administrador y se vuelve al paso 3.

Flujo alternativo: Error inesperado BD.

Línea 5: La función de guardar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 4.

Identificador: UC13.2

Actores: Administrador

Objetivos: Modificar los datos de una bodega.

Condiciones previas: Existe la bodega que se quiere modificar.

Condiciones posteriores: Los datos de la bodega son modificados con los nuevos datos.

Frecuencias de uso: Infrecuente. Se usa normalmente si se ha equivocado en algún dato o si la bodega actualizó sus datos.

Flujo Básico:

1. El caso de uso empieza cuando el administrador quiere modificar los datos de la bodega.
2. El sistema mostrará una pantalla con un listado de bodegas con su información.
3. El administrador selecciona la bodega y carga la información de la bodega seleccionada.
4. El administrador modifica los datos que quiere actualizar, todos los datos son modificables.
5. El administrador indica al sistema que ya ha acabado de modificar los datos aceptándolos.
6. El sistema hace las modificaciones en los datos de la bodega y los actualiza.
7. El sistema muestra un mensaje al administrador indicando que la operación se ha realizado con éxito.

Flujo alternativo: Datos obligatorios incorrectos.

Línea 6: Si los datos introducidos se encuentran vacíos o incorrectos, el sistema muestra un mensaje de error al administrador y se vuelve al paso 4.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de modificar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Identificador: UC13.3

Actores: Administrador

Objetivos: El administrador quiere eliminar una bodega.

Condiciones previas: Existe la bodega que se quiere eliminar.

Condiciones posteriores: La bodega ha sido eliminada del sistema.

Frecuencias de uso: Infrecuente. Normalmente se elimina si hubo error al ingresar bodega.

Flujo Básico:

1. El caso de uso empieza cuando el administrador quiere eliminar una bodega.
2. El sistema mostrará una pantalla con un listado de bodegas con su información.
3. El administrador selecciona la bodega.
4. El administrador indica al sistema que desea eliminar la bodega aceptando su eliminación.
5. El sistema elimina la bodega del sistema.
6. El sistema muestra un mensaje al usuario indicando que la operación se ha realizado con éxito.

Flujo alternativo: Bodega vinculada a registro de otra tabla.

Línea 5: Si la bodega está vinculada a uno o más registros de otras tablas no podrá ser eliminada. En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de eliminar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Identificador: UC13.4

Actores: Administrador

Objetivos: El administrador quiere visualizar la localización de la bodega en el mapa.

Condiciones previas: Existe la bodega que se quiere visualizar localización.

Condiciones posteriores: Visualización de la localización en mapa de la bodega.

Frecuencias de uso: Infrecuente. Normalmente se utiliza en casos de consulta.

Flujo Básico:

1. El caso de uso empieza cuando el administrador quiere visualizar localización en mapa.
2. El sistema mostrará una pantalla con un listado de las bodegas con su información.

3. El administrador selecciona la bodega.
4. El administrador indica al sistema que desea visualizar la bodega.
5. El sistema muestra una pantalla con la visualización de la localización de la bodega en un mapa.

Flujo alternativo: Localización errónea.

Línea 5: Si se ha llenado mal los datos de localización de la bodega, la generación de la localización es errónea. En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de eliminar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Caso de uso - Realizar Expedición

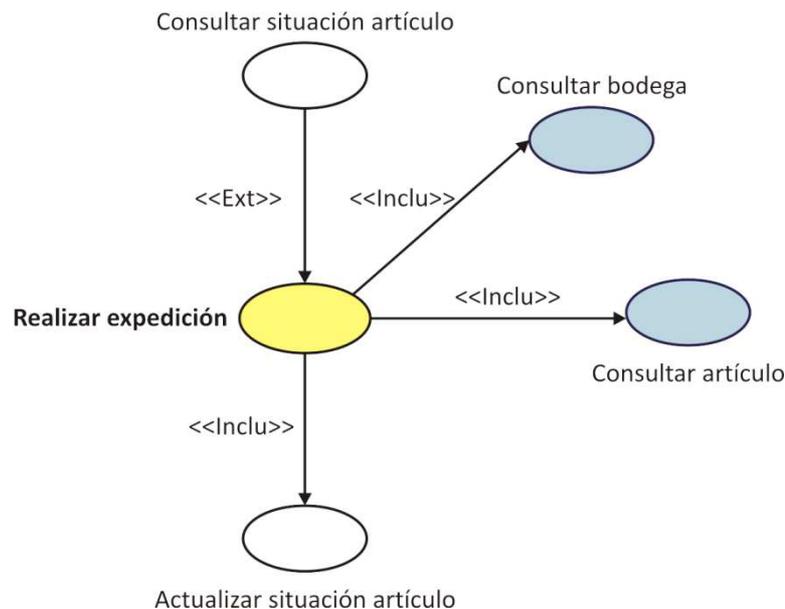


Gráfico 31. Realizar expedición

Identificador: UC14.1

Actores: Usuario

Objetivos: Determinar si el artículo existe para su venta realizando expedición.

Condiciones previas: Bodega y artículo deben existir.

Condiciones posteriores: Si se vende actualizar situación del artículo.

Frecuencias de uso: Frecuente. Este caso de uso es necesario para poder realizar ventas.

Flujo Básico:

1. El caso de uso empieza cuando el usuario necesita saber la situación de un artículo y la bodega en la que se encuentra.
2. El sistema mostrará una pantalla con un listado de los artículos con su información correspondiente.
3. El usuario selecciona el artículo necesario.

Flujo alternativo: Artículo sin inventario.

Línea 3: Si el artículo no posee inventario, el sistema muestra un mensaje de información al usuario y vuelve al paso 2.

Flujo alternativo: Error inesperado BD.

Línea 3: La función de seleccionar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Caso de uso – Definir Proveedor

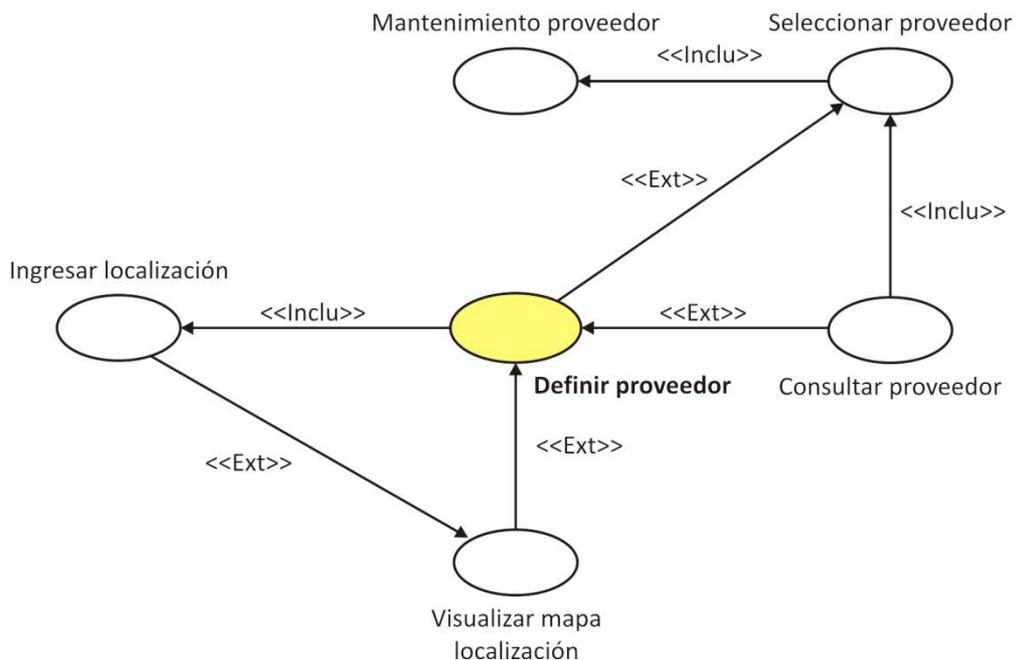


Gráfico 32. Definir proveedor

Identificador: UC16.1

Actores: Usuario

Objetivos: Definir proveedor.

Condiciones previas: Ninguna

Condiciones posteriores: Existe un nuevo proveedor.

Frecuencias de uso: Frecuente. Este caso de uso es necesario para poder realizar compras, ordenes de entrega - recepción.

Flujo Básico:

1. El caso de uso empieza cuando el usuario quiere crear un nuevo proveedor.
2. El sistema mostrará una pantalla donde se muestran los campos que se deben llenar para la creación del proveedor.
3. El usuario introduce el RUC, razón social, descripción, dirección (calle principal, calle secundaria, nro) y teléfono.
4. El usuario indica al sistema que ya se ha acabado de introducir los datos aceptándolos.
5. El sistema crea y guarda los datos del nuevo proveedor.
6. La base de datos activa el trigger para la creación geográfica de la localización a través de las calles.
7. El sistema muestra al proveedor un mensaje indicando que la operación se ha realizado con éxito.

Flujo alternativo: Conflicto de proveedor ya existente.

Línea 4: Si el proveedor ya existe con el mismo RUC, el sistema muestra un mensaje de información al usuario y se vuelve al paso 3.

Flujo alternativo: Datos obligatorios incorrectos.

Línea 4: Si los datos introducidos se encuentran vacíos o incorrectos, el sistema muestra un mensaje de error al usuario y se vuelve al paso 3.

Flujo alternativo: Error inesperado BD.

Línea 5: La función de guardar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 4.

Identificador: UC16.2

Actores: Usuario

Objetivos: Modificar los datos de un proveedor.

Condiciones previas: Existe el proveedor que se quiere modificar.

Condiciones posteriores: Los datos del proveedor son modificados con los nuevos datos.

Frecuencias de uso: Infrecuente. Se usa normalmente si se ha equivocado en algún dato o si el proveedor actualizó sus datos.

Flujo Básico:

1. El caso de uso empieza cuando el usuario quiere modificar los datos del proveedor.
2. El sistema mostrará una pantalla con un listado de proveedores con su información.
3. El usuario selecciona el proveedor y carga la información del proveedor seleccionado.
4. El usuario modifica los datos que quiere actualizar, todos los datos son modificables a excepción del RUC.
5. El usuario indica al sistema que ya ha acabado de modificar los datos aceptándolos.
6. El sistema hace las modificaciones en los datos del proveedor y los actualiza.
7. El sistema muestra un mensaje al usuario indicando que la operación se ha realizado con éxito.

Flujo alternativo: Datos obligatorios incorrectos.

Línea 6: Si los datos introducidos se encuentran vacíos o incorrectos, el sistema muestra un mensaje de error al usuario y se vuelve al paso 4.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de modificar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Identificador: UC16.3

Actores: Usuario

Objetivos: El usuario quiere eliminar un proveedor.

Condiciones previas: Existe el proveedor que se quiere eliminar.

Condiciones posteriores: El proveedor ha sido eliminado del sistema.

Frecuencias de uso: Infrecuente. Normalmente se elimina si hubo error al ingresar proveedor.

Flujo Básico:

1. El caso de uso empieza cuando el usuario quiere eliminar un proveedor.
2. El sistema mostrará una pantalla con un listado de proveedores con su información.
3. El usuario selecciona el proveedor.
4. El usuario indica al sistema que desea eliminar el proveedor aceptando su eliminación.
5. El sistema elimina el proveedor del sistema.
6. El sistema muestra un mensaje al usuario indicando que la operación se ha realizado con éxito.

Flujo alternativo: Proveedor vinculado a registro de otra tabla.

Línea 5: Si el proveedor está vinculado a uno o más registros de otras tablas no podrá ser eliminado. En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de eliminar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Identificador: UC16.4

Actores: Usuario

Objetivos: El usuario quiere visualizar la localización del proveedor en el mapa.

Condiciones previas: Existe el proveedor que se quiere visualizar localización.

Condiciones posteriores: Visualización de la localización en mapa del proveedor.

Frecuencias de uso: Infrecuente. Normalmente se utiliza en casos de consulta.

Flujo Básico:

1. El caso de uso empieza cuando el usuario quiere visualizar localización en mapa.
2. El sistema mostrará una pantalla con un listado de proveedores con su información.
3. El usuario selecciona el proveedor.
4. El usuario indica al sistema que desea visualizar el proveedor.
5. El sistema muestra una pantalla con la visualización de la localización del proveedor en un mapa.

Flujo alternativo: Localización errónea.

Línea 5: Si el proveedor ha dado mal sus datos, la generación de la localización es errónea. En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de eliminar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Caso de uso – Definir Orden

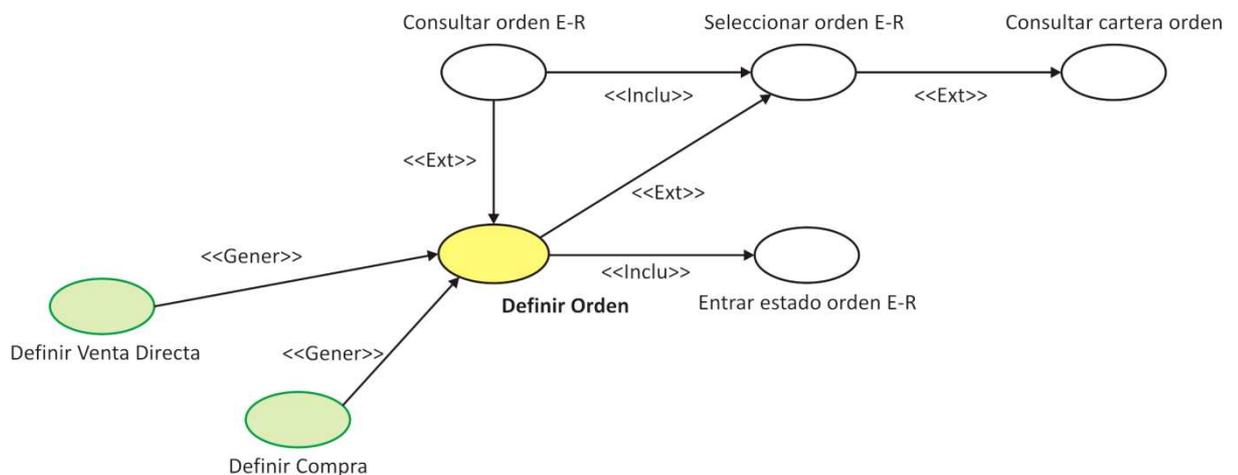


Gráfico 33. Definir orden

Identificador: UC18.1

Actores: Usuario

Objetivos: Definir orden de Entrega – Recepción (E-R).

Condiciones previas: Compra y venta directa deben existir.

Condiciones posteriores: Orden realizada.

Frecuencias de uso: Frecuente. Este caso de uso es necesario para poder entregar o retirar producto.

Flujo Básico:

1. El caso de uso empieza cuando el usuario necesita realizar una orden.
2. El sistema mostrará una pantalla donde seleccionará a través de otra pantalla de un listado las compras y ventas que se tienen que entregar o retirar al cliente o distribuidor correspondientemente.
3. El usuario ingresa el estado de la orden E-R.

4. El sistema procesa la orden correspondiente.

Flujo alternativo: Compra y/o venta no existe.

Línea 2: Si la venta y/o compra no existen, el usuario deberá realizar los casos de uso UC15.1 y UC17.1 correspondientemente, se retorna al paso 2.

Flujo alternativo: Error inesperado BD.

Línea 4: La función de procesar orden puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Caso de uso – Definir Catálogo de Artículos

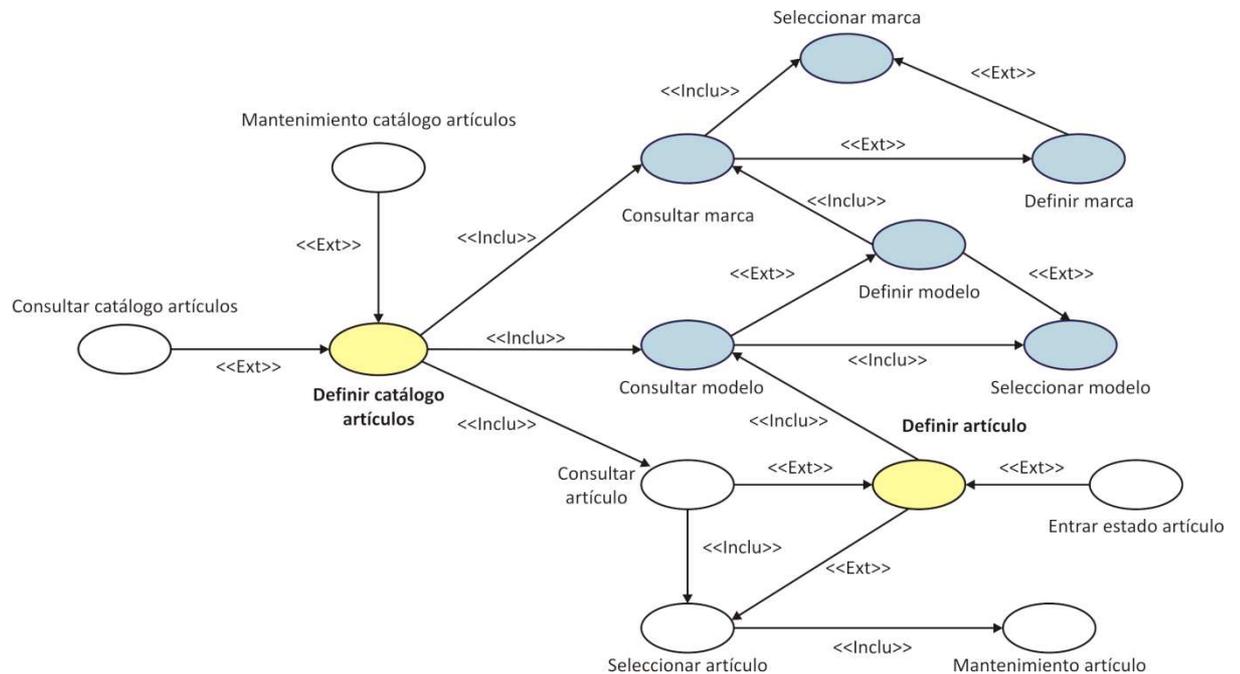


Gráfico 34. Definir catálogo de artículos

Identificador: UC19.1

Actores: Usuario

Objetivos: Definir catálogo de artículos

Condiciones previas: Marca, modelo y artículo deben existir.

Condiciones posteriores: Catálogo de artículos existe.

Frecuencias de uso: Frecuente. Este caso de uso es necesario para poder realizar ventas.

Flujo Básico:

1. El caso de uso empieza cuando el usuario necesita realizar un catálogo de artículo.

2. El sistema mostrará una pantalla donde seleccionará a través de otra pantalla de un listado los artículos que se venderán.
3. El usuario asigna el precio respectivo y condiciones de venta.
4. El sistema procesa el catálogo.

Flujo alternativo: Artículo no existe.

Línea 3: Si el artículo no existe, el sistema muestra un mensaje de error y se retorna al paso 2.

Flujo alternativo: Error inesperado BD.

Línea 4: La función de procesar el catálogo puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Identificador: UC19.2

Actores: Usuario

Objetivos: Modificar los datos de un catálogo de artículos.

Condiciones previas: Existe el catálogo de artículos que se quiere modificar.

Condiciones posteriores: Los datos del catálogo son modificados con los nuevos datos.

Frecuencias de uso: Infrecuente. Se usa normalmente si se ha equivocado en algún dato o si el artículo necesita actualización de sus datos.

Flujo Básico:

1. El caso de uso empieza cuando el usuario quiere modificar los datos del catálogo.
2. El sistema mostrará una pantalla con el catálogo listando los artículos con su información.
3. El usuario selecciona el artículo y carga la información del artículo seleccionado.
4. El usuario modifica los datos que quiere actualizar, solo son modificables los precios de venta del artículo.
5. El usuario indica al sistema que ya ha acabado de modificar los datos aceptándolos.
6. El sistema hace las modificaciones en los datos del proveedor y los actualiza.
7. El sistema muestra un mensaje al usuario indicando que la operación se ha realizado con éxito.

Flujo alternativo: Datos obligatorios incorrectos.

Línea 6: Si los datos introducidos se encuentran vacíos o incorrectos, el sistema muestra un mensaje de error al usuario y se vuelve al paso 4.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de modificar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Identificador: UC19.3

Actores: Usuario

Objetivos: El usuario quiere eliminar un catálogo de artículo.

Condiciones previas: Existe el catálogo que se quiere eliminar.

Condiciones posteriores: El catálogo de artículo ha sido eliminado del sistema.

Frecuencias de uso: Infrecuente. Normalmente se elimina si hubo error al ingresar catálogo.

Flujo Básico:

1. El caso de uso empieza cuando el usuario quiere eliminar un catálogo de artículo.
2. El sistema mostrará una pantalla con el catálogo listado los artículos con su información.
3. El usuario selecciona el artículo del catálogo.
4. El usuario indica al sistema que desea eliminar el artículo aceptando su eliminación.
5. El sistema elimina el artículo del catálogo.
6. El sistema muestra un mensaje al usuario indicando que la operación se ha realizado con éxito.

Flujo alternativo: Artículo de catálogo vinculado a registro de otra tabla.

Línea 5: Si el artículo del catálogo está vinculado a uno o más registros de otras tablas no podrá ser eliminado. En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Flujo alternativo: Error inesperado BD.

Línea 6: La función de eliminar puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Caso de uso – Definir Ruta

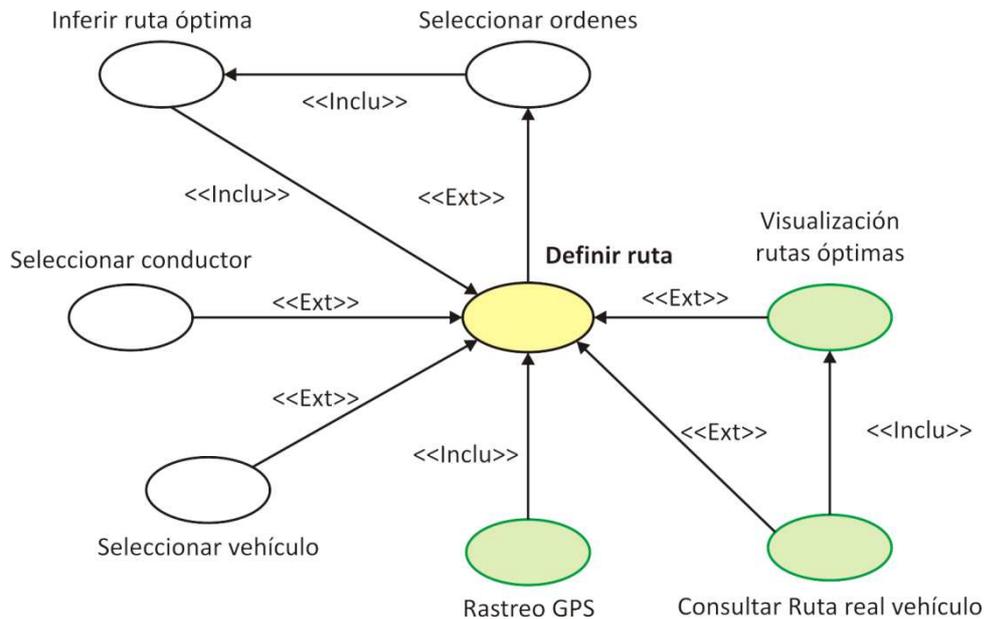


Gráfico 35. Definir ruta

Identificador: UC20.1

Actores: Usuario

Objetivos: Definir ruta de entrega-recepción

Condiciones previas: Rastreo GPS funcionando, vehículo, conductor y ordenes deben existir.

Condiciones posteriores: Inferencia de ruta óptima

Frecuencias de uso: Frecuente. Este caso de uso es necesario para poder realizar la logística de gestión de entregas.

Flujo Básico:

1. El caso de uso empieza cuando el usuario necesita realizar una entrega y/o recepción.
2. El sistema mostrará una pantalla donde seleccionará a través de otra pantalla de un listado las ordenes que se entregarán o recibirán, se selecciona el conductor del vehículo y el vehículo que se utilizará.
3. El sistema realiza la inferencia de la ruta óptima.
4. El sistema muestra la ruta óptima.
5. El conductor realiza la entrega y/o recepción de artículos.

6. El sistema de rastreo envía constantemente la información de localización del vehículo.

7. El sistema en cualquier momento puede indicar la ruta real del vehículo a través del rastreo GPS.

Flujo alternativo: Rastreo GPS no funciona.

Línea 6: Si el sistema no recibe información del GPS, verificar si se encuentra funcionando, puerto recibiendo información y corriendo aplicación especializada. Volver a línea 6.

Flujo alternativo: Error inesperado BD.

Línea 3: La función de inferir ruta óptima puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Caso de uso - Definir Catálogo Consultas Ruteo

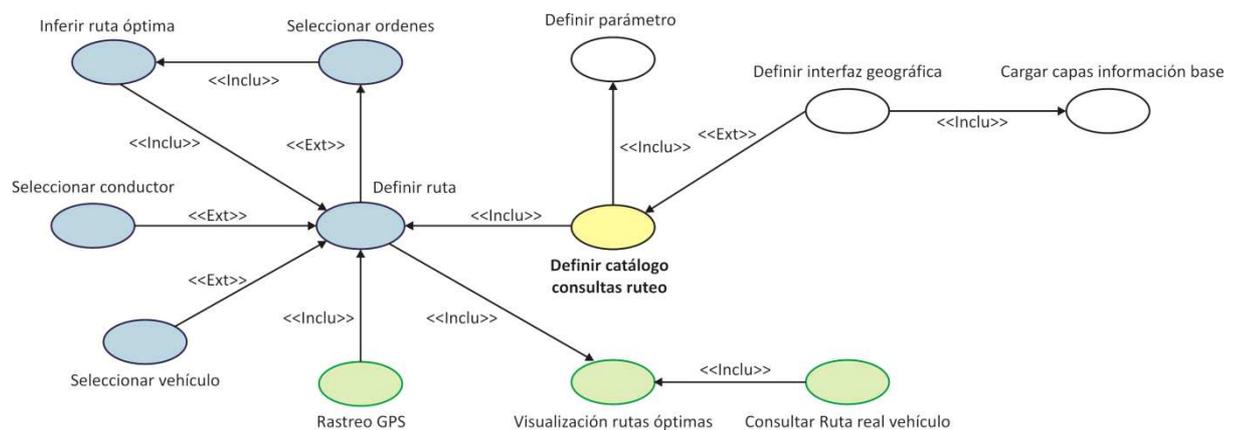


Gráfico 36. Definir catálogo consultas ruteo

Identificador: UC21.1

Actores: Usuario

Objetivos: Definir catálogo de consultas ruteo

Condiciones previas: Definición ruta óptima debe existir, rastreo GPS funcionando, aplicación especializada funcionando.

Condiciones posteriores: Definición del catálogo de consultas de ruteo.

Frecuencias de uso: Frecuente. Este caso de uso es necesario para poder visualizar las rutas que el conductor realizó, así como la información cartográfica de la zona de estudio.

Flujo Básico:

1. El caso de uso empieza cuando el usuario necesita realizar una consulta de ruta.
2. El sistema mostrará una pantalla donde el usuario filtrará la información según parámetros de fecha, vehículo, conductor, GPS y/o orden de entrega.
3. El usuario acepta búsqueda de información.
4. El sistema mostrará una pantalla con la información filtrada.
5. El usuario selecciona el registro que desea visualizar.
6. El sistema cargará la información cartográfica base.
7. El sistema mostrará una pantalla con la visualización de las rutas reales realizadas así como la ruta óptima sugerida.

Flujo alternativo: Información no existe.

Línea 4: Si no existe la información filtrada el sistema mostrará un mensaje de información. Volver a línea 2.

Flujo alternativo: Error inesperado BD.

Línea 3: La función de búsqueda puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Caso de uso - Realizar Consulta

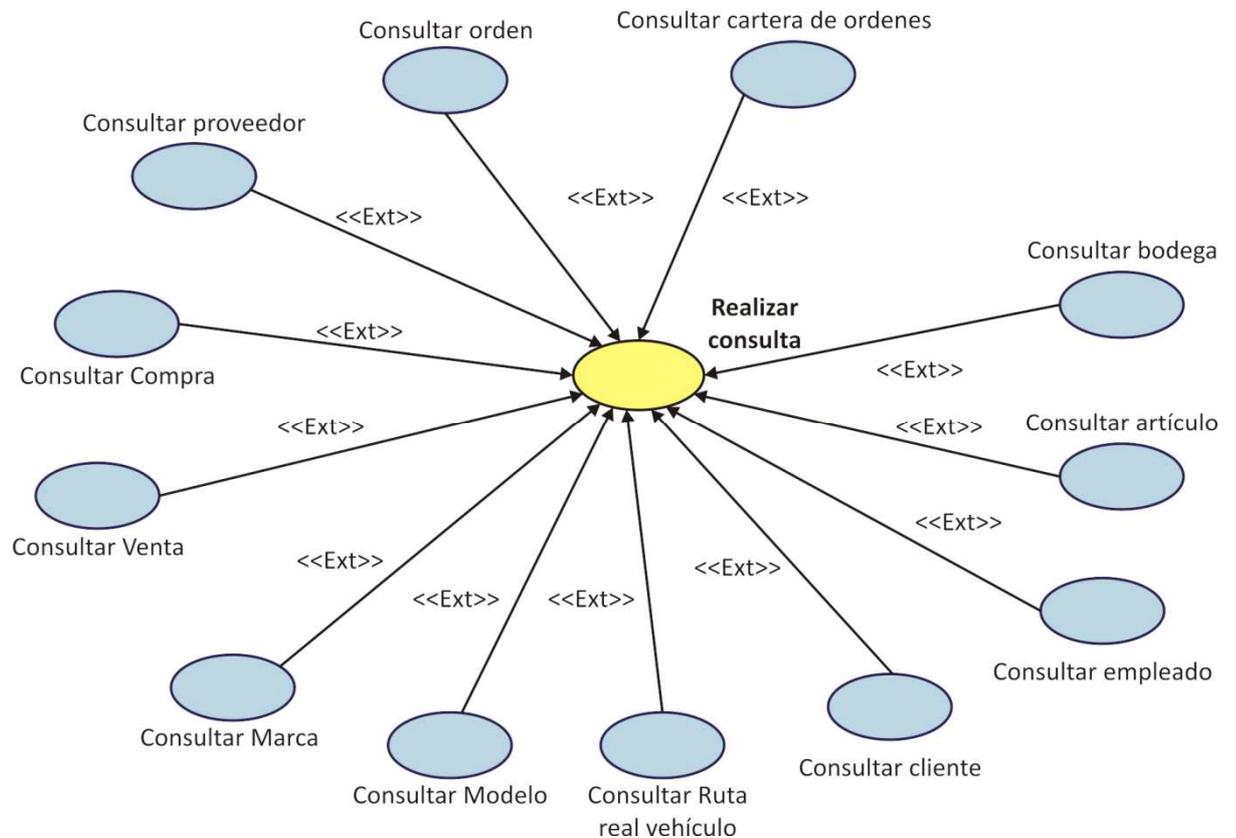


Gráfico 37. Realizar consulta

Identificador: UC22.1

Actores: Usuario

Objetivos: Realizar consultas

Condiciones previas: Ninguna.

Condiciones posteriores: Consulta de modelo, ruta real vehículo, orden, cartera de órdenes, bodega, artículo, empleado, cliente, marca, venta, compra o proveedor.

Frecuencias de uso: Frecuente. Este caso de uso es necesario para poder realizar consultas.

Flujo Básico:

1. El caso de uso empieza cuando el usuario necesita realizar una consulta.
2. El sistema mostrará una pantalla donde el usuario seleccionará que información desea conocer.

3. El sistema mostrará una pantalla con un listado de la información correspondiente.

Flujo alternativo: Información no existe.

Línea 3: Si no existe la información el sistema mostrará un mensaje de información. Volver a línea 2.

Flujo alternativo: Error inesperado BD.

Línea 3: La función de consulta puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

Caso de uso - Realizar Proceso

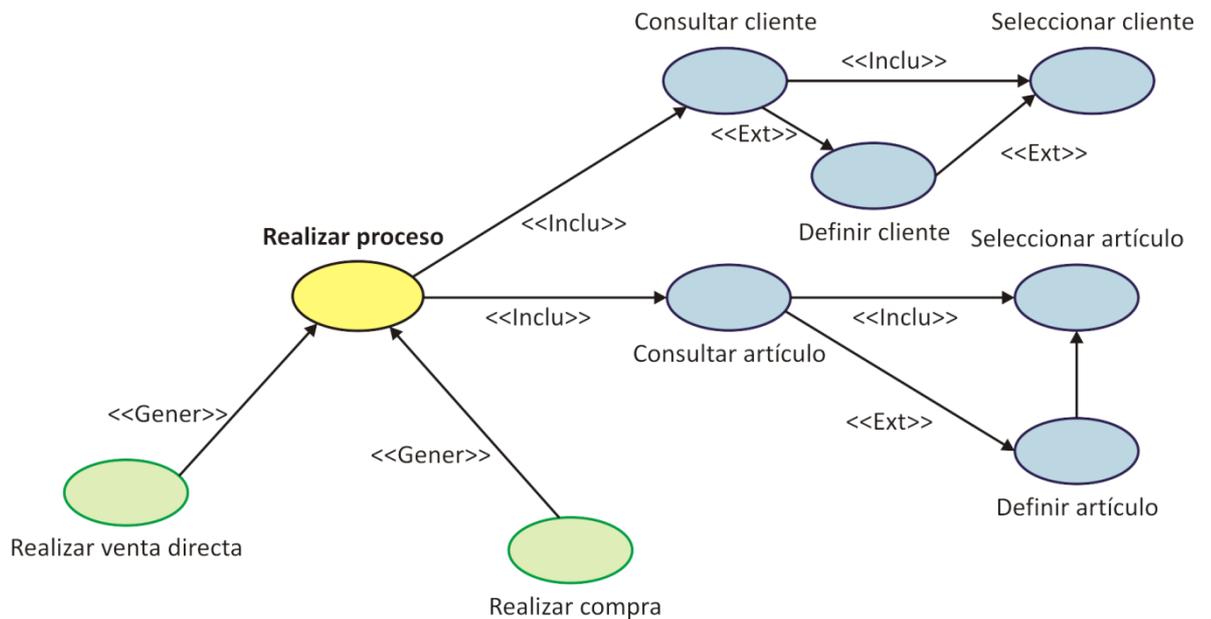


Gráfico 38. Realizar proceso

Identificador: UC23.1

Actores: Usuario medio (Protocolo de comunicación)

Objetivos: Realizar proceso de cálculos.

Condiciones previas: Compra, venta directa por ejecutarse, artículo debe existir.

Condiciones posteriores: Cálculos realizados

Frecuencias de uso: Frecuente. Este caso de uso es necesario para poder realizar proceso de cálculo.

Flujo Básico:

1. El caso de uso empieza cuando el usuario necesita realizar una compra o una venta.
2. El sistema calcula los valores correspondientes accedendo a la tabla de artículo.
3. El sistema retorna a la pantalla correspondiente de compra o venta.
4. Los cálculos son transparentes al usuario.

Flujo alternativo: Error inesperado BD.

Línea 2: La función de proceso puede fallar en cualquier momento inesperado (el gestor de BD deja de funcionar). En tal caso, se mostrará un mensaje de error y se retornará al paso 2.

3.3. ELABORACIÓN DEL MODELO DE DATOS

Una vez definidos los casos de uso, se presenta a continuación dos modelos de base de datos, se los separó por funcionalidad para mejor entendimiento. El primero es el modelo de base de datos que almacenará la información espacial para gestión en transporte y logística:

Sistema para gestión en transporte y logística

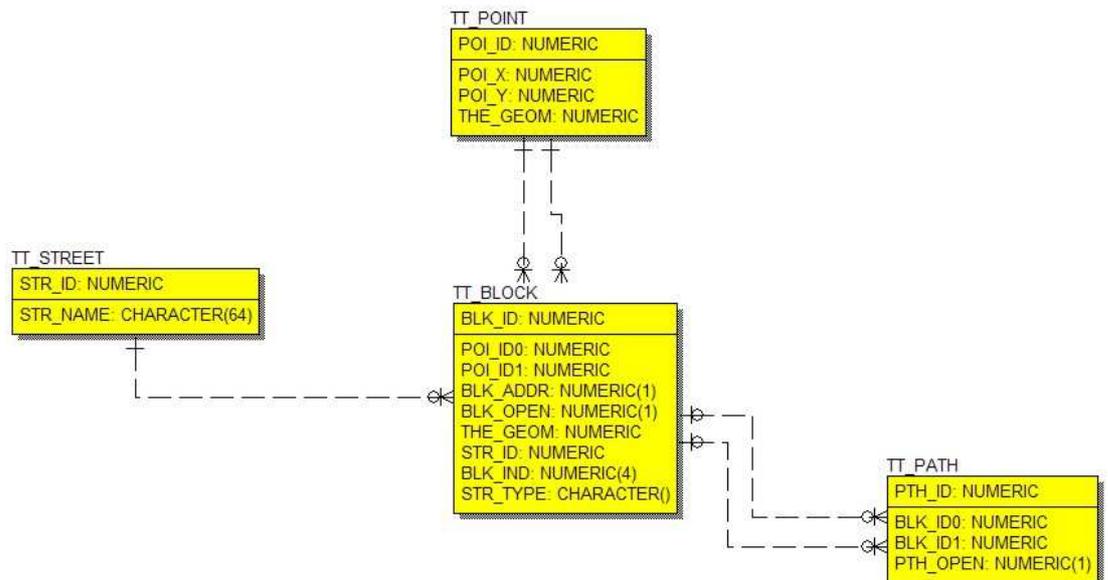


Gráfico 39. Modelo de datos para el sistema de gestión de transporte y logística

A continuación se detalla una descripción de las tablas y atributos a utilizar:

TT_STREET: Definición de las calles, sus atributos son:

- **STR_ID:** Código de la calle.
- **STR_NAME:** Nombre de la calle.

TT_POINT: Definición de intersecciones, sus atributos son los siguientes:

- **POI_ID:** Identificación del punto.
- **POI_X:** Longitud
- **POI_Y:** Latitud
- **THE_GEOM:** Campo que almacena la geometría del punto.

TT_BLOCK: Definición de caminos, como atributos tenemos:

- **BLK_ID:** Identificación del bloque
- **POI_ID0:** ID de las coordenadas correspondientes al punto inicial del bloque.
- **POI_ID1:** ID de las coordenadas correspondientes al punto final del bloque.
- **BLK_ADDR:** Se refiere al sentido del bloque (calle). Sus valores pueden ser 0 ó 1.
 - 0:** Significa que el bloque va en sentido de POI_ID0 a POI_ID1. (→→)
 - 1:** Significa que el bloque va en sentido de POI_ID1 a POI_ID0. (←←)
 - 3:** Significa desconocido.
- **BLK_OPEN:** Hace referencia al estado en que se encuentra el bloque de calles. Sus valores pueden ser 0 ó 1.
 - 0:** Significa que el bloque de calle está cerrado.
 - 1:** Significa que el bloque de calle está abierto.
- **STR_ID:** Es el código de la calle al que está asociado.
- **BLK_D:** Hace referencia a la longitud de la calle en metros.
- **STR_TYPE:** Es el tipo de calle que es, sus valores pueden ser:
 - 1:** Autopista.
 - 2:** Principal.
 - 3:** Secundaria.
- **THE_GEOM:** Campo que almacena la geometría del punto.

TT_PATH: Señal de tránsito aplicado a la unión de dos cuadras:

- **PTH_ID:** Identificación del path.
- **BLK_ID0:** Referencia del bloque con respecto a BLK_ID1, refleja la unión del bloque 0 con el bloque 1.
- **BLK_ID1:** Referencia del bloque con respecto a BLK_ID0.
- **PTH_OPEN:** Se refiere a si puede pasar por dicho camino. Puede adoptar dos valores, 0 y 1.
 - 0:** No puede seguir camino.
 - 1:** Si puede seguir camino.

La información necesaria para llenar dicha base de datos se extraerá de los shapes files depurados, además a través de la observación de la realidad se reflejará de mejor manera los sentidos originales de las calles, así como intersecciones. Las coordenadas en latitud y longitud de la ciudad de Cuenca serán reales, asegurando así un correcto uso del GPS del vehículo.

A continuación se presenta el modelo de gestión de entregas:

Modelo de gestión en entregas

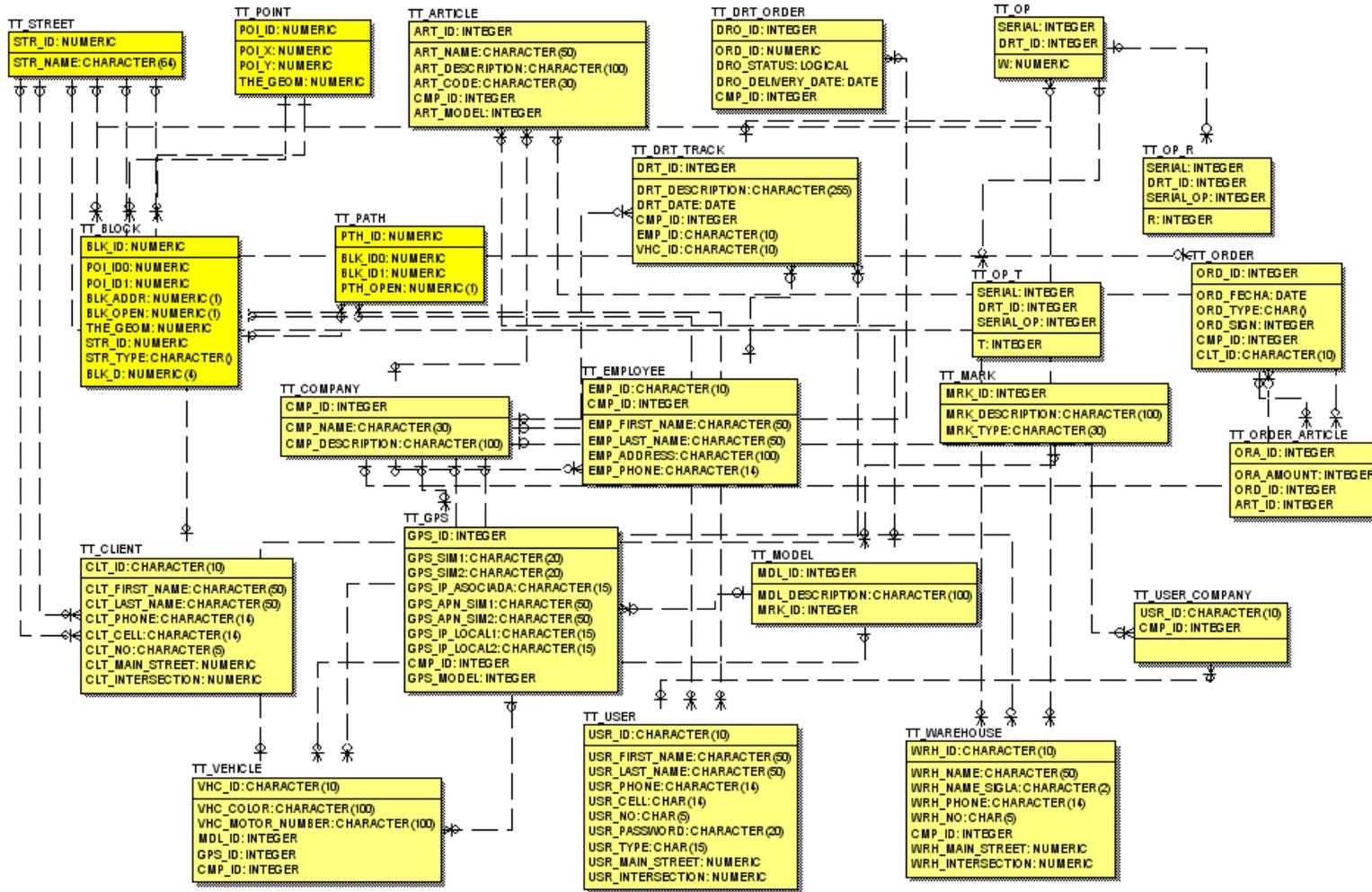


Gráfico 40. Modelo de Gestión de Entregas

El modelo de gestión de entregas presentado es un abstracto del modelo real que tiene la empresa Madeco y consta con las tablas bases para la gestión de entregas, por lo mismo se realizará un demo que simula el comportamiento del sistema.

A continuación se describe el contenido de cada una de las tablas:

TT_COMPANY: Contiene los datos de la empresa, como son el id, el nombre y la descripción.

TT_USER: Contiene los datos del usuario que tiene acceso al sistema.

TT_USER_COMPANY: Esta tabla hace referencia a las empresas que puede acceder el usuario.

TT_CLIENT: Contiene los datos de los clientes, la dirección será generada con geom a través de la dirección principal e intersección que se vincula con la tabla de calles.

TT_EMPLOYEE: Hace referencia a los datos del empleado.

TT_GPS: Tiene los datos propios del GPS, esta tabla es importante porque en ella se tiene los datos básicos para el rastreo como son el Id del GPS, el APN, las IP local y del servidor.

TT_MARK: Tiene la información de las marcas.

TT_MODEL: Contiene la información del modelo vinculado a una marca y asociado al artículo, vehículo y GPS.

TT_VEHICLE: Tiene la información de los vehículos que posee la empresa.

TT_WAREHOUSE: Tiene la información de las bodegas de la empresa, la dirección será generada con geom a través de la dirección principal e intersección que se vincula con la tabla de calles.

TT_OP, TT_OP_R Y TT_OP_T: Tiene la información de las rutas óptimas generadas por Prolog, no son visibles para el usuario.

TT_ORDER: Tiene la información de las cabeceras de las ordenes de entrega – recepción.

TT_ORDER_ARTICLE: Tiene la información del detalle de la orden de entrega – recepción.

TT_DRT_TRACK: Tiene la información de la cabecera de las ordenes que el conductor tiene que entregar en el día.

TT_DRT_ORDER: Tiene el detalle de las ordenes a entregar por el conductor.

3.4. DEFINICIÓN DE LAS INTERFAZ DEL SISTEMA

3.4.1. Especificación de principios generales de la interfaz

El correcto funcionamiento de la aplicación depende de una interfaz amigable que cumpla con todos los requisitos de usabilidad, accesibilidad, rendimiento, etc. Para la construcción de la interfaz se tomará en cuenta que:

- Los usuarios deben tener control del sistema.
- El sistema debe ser intuitivo y fácil de usar.
- La interfaz debe ser consistente y clara.
- Los mensajes de error y de aviso serán explicativos y describirá lo que el usuario debe hacer.
- Se definirán las pantallas a partir de los casos de uso identificados.
- Cada pantalla de caso de uso debe tener los campos necesarios para su ingreso.
- El diseño será homogéneo.

La pantalla principal al acceder a la intranet deberá solicitar el inicio de sesión del usuario. De esta manera el usuario estará autenticado y el sistema únicamente permitirá el acceso a las diferentes funcionalidades según su cargo. El usuario en cualquier momento podrá cerrar sesión y acceder introduciendo su nombre de usuario y contraseña.

Después de que el usuario inicie sesión el sistema mostrará la pantalla principal que está compuesta por un conjunto de links de acceso a las diferentes funcionalidades. El sistema mostrará en todo momento dichos links para poder acceder rápidamente a las funcionalidades que el usuario tenga acceso.

Si el usuario no tiene acceso a alguna funcionalidad el sistema indicará a través de un mensaje la restricción.

Al acceder a una funcionalidad concreta el sistema mostrará la pantalla de esta funcionalidad. Todas las pantallas seguirán el mismo diseño estándar.

A continuación se muestra el diseño de las diferentes aplicaciones:

Diseño estándar de las pantallas del Sistema:

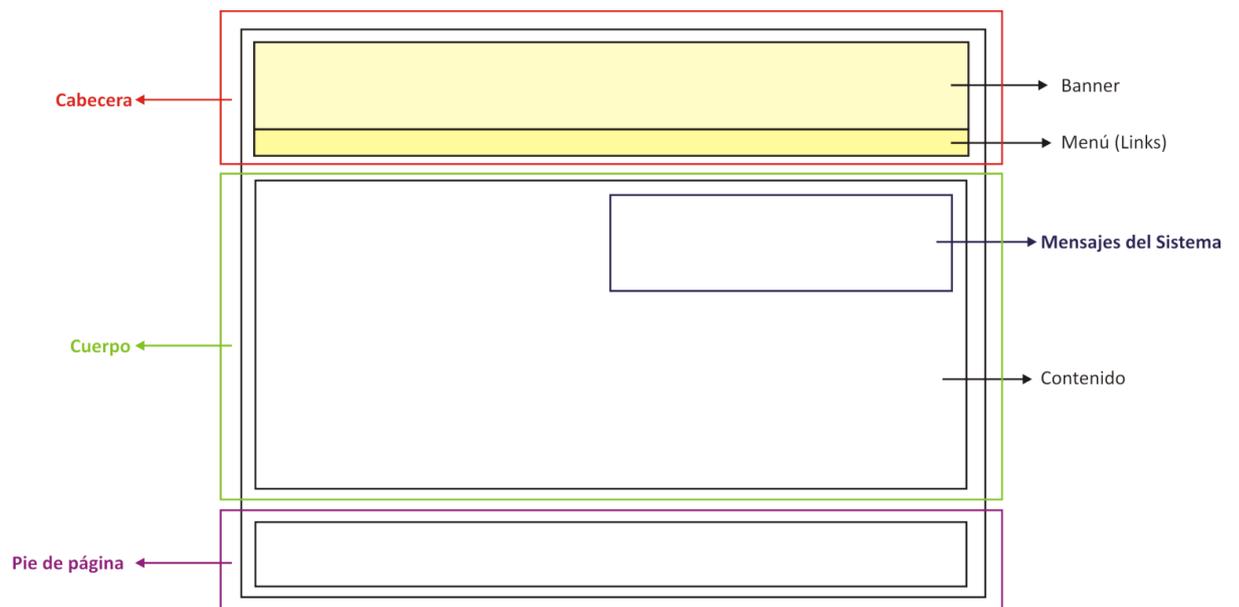


Gráfico 41. Diseño estándar de las pantallas del sistema

1. La cabecera contiene el banner y el menú en base a links.
2. El cuerpo tendrá el contenido de cada funcionalidad, así como los mensajes del sistema en caso de que el usuario tuviese algún error.

Los íconos asociados a la funcionalidad serán los siguientes:

-  Guardar
-  Cargar información
-  Actualizar
-  Eliminar
-  Regresar
-  Limpiar
-  Buscar

Mensajes del sistema

- Lista de todos los mensajes resumidos

3. El pie de página servirá para poner cualquier información adicional que el usuario deba conocer en todo momento.

Diseño estándar del Servidor de Mapas:

El servidor de mapas que mostrará la información cartográfica cumplirá los mismos requerimientos de usabilidad, diseño, rendimiento, etc. antes expuestos, a continuación se muestra el diseño estándar del servidor de mapas.

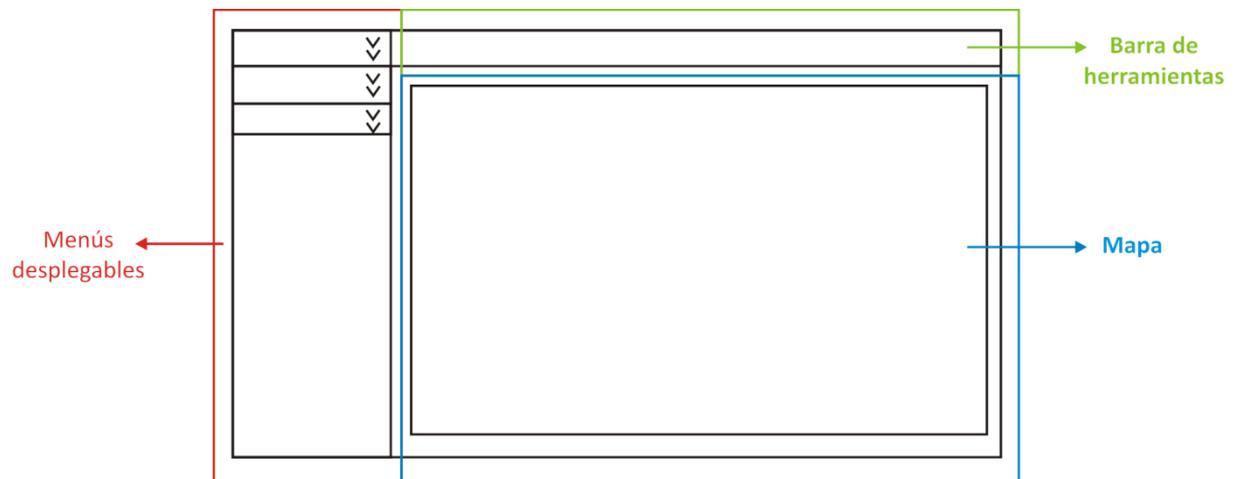


Gráfico 42. Diseño estándar del Servidor de Mapas

1. La barra de herramientas comprenderán las funcionalidades básicas del mapa como son: Acercamiento, alejamiento, vista total, historial de vistas, dibujo y solicitud de información.
2. En los menús desplegables se encontrará la estructura de las capas en árboles, así como la simbología y posición geográfica.
3. La zona del mapa mostrará la información cartográfica disponible.

3.5. DEFINICIÓN DE LA ARQUITECTURA DEL SISTEMA

La arquitectura del software consiste en un conjunto de patrones y abstracciones coherentes que proporciona el marco de referencia necesario para guiar la construcción del software para un sistema de información, así como los

fundamentos para que analistas, diseñadores, programadores, etc., trabajen en una línea común que permitirá obtener los objetivos del sistema, cubriendo todas las necesidades.

Para el desarrollo de este sistema se utilizará el Modelo Vista Controlador (MVC), a continuación se explica dicho modelo:

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

A continuación se detalla más específicamente cada componente:

- **Modelo:** Esta es la representación específica de la información con la cual el sistema opera. El modelo se limita a lo relativo de la vista y su controlador facilitando las presentaciones visuales complejas.
- **Vista:** Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- **Controlador:** Este responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y, probablemente, a la vista.

Se tomo este modelo pues separa los diferentes elementos antes presentados permitiendo tener un sistema flexible y portable.

3.6. RECOLECCIÓN, LEVANTAMIENTO Y DEPURACIÓN DE INFORMACIÓN GEOGRÁFICA PARA CREACIÓN DE RUTAS

3.6.1. Cartografía base

El modelo de gestión de entregas se basa en la información cartográfica del Centro de la Ciudad de Cuenca. La información cartográfica hace referencia a aquellos datos espacialmente referenciados, estos almacenan información cartográfica y alfanumérica. A través de la información cartográfica se puede conocer la localización exacta de los distintos elementos en el espacio y con respecto a otros (topología), con la alfanumérica se obtienen los datos sobre las características o atributos del elemento geográfico.

Dicha información cartográfica se estructura a través de mapas temáticos, para lo cual dispondremos de un SIG (Sistema de información geográfico) que nos permita la gestión, obtención, análisis y depuración de los datos.

Para la generación de nodos y bloques de calles es necesario contar con una información geográfica precisa, a continuación se detallan los procedimientos seguidos para la obtención de dichos datos.

A través de la Geomática se obtendrá la infraestructura geográfica que hace referencia a la base espacial sobre la que se apoya y organiza toda información geográfica que ofrece funciones de carácter métrico, geométrico, topológico, temático y de interoperabilidad espacial. Dicha infraestructura está formada por la red geodésica del centro de ciudad de Cuenca.

Para el proyecto se tomará como cartografía base las capas temáticas en formato shape del proyecto SI_CUENCA, información gratuita entregada por la Universidad del Azuay, año 2008. El SIG a utilizar es ArcGIS para el procesamiento y depuración de los datos.

Los shapfiles originales del proyecto SI_CUENCA se encuentran con el datum Provisional de Sudamérica de 1956 UTM zona 17S, escala 1:1000. La ciudad de Cuenca se encuentra en la Zona 17 Sur. Entre los shapfiles a utilizar del proyecto SI_CUENCA se encuentran:

-  Manzanas.shp
-  Parr_urb.shp
-  Vias.shp

La cartografía del Centro de la Ciudad de Cuenca se basa en un detallado proceso de análisis y depuración de los shapfiles manzanas, parroquias urbanas y vías correspondientes a SI_CUENCA. Los datos temáticos fundamentales, tendrán la consideración de cartografía temática. La depuración de la cartografía proporcionada es de suma importancia ya que contendrá los nodos que servirán para la deducción de las rutas óptimas, por lo cual se debe tener datos espaciales precisos. A continuación, en el Gráfico 43 se presenta el Mapa de la Ciudad de Cuenca elaborado a partir de los shapfiles antes mencionados con el datum PSAD56 Z17S.

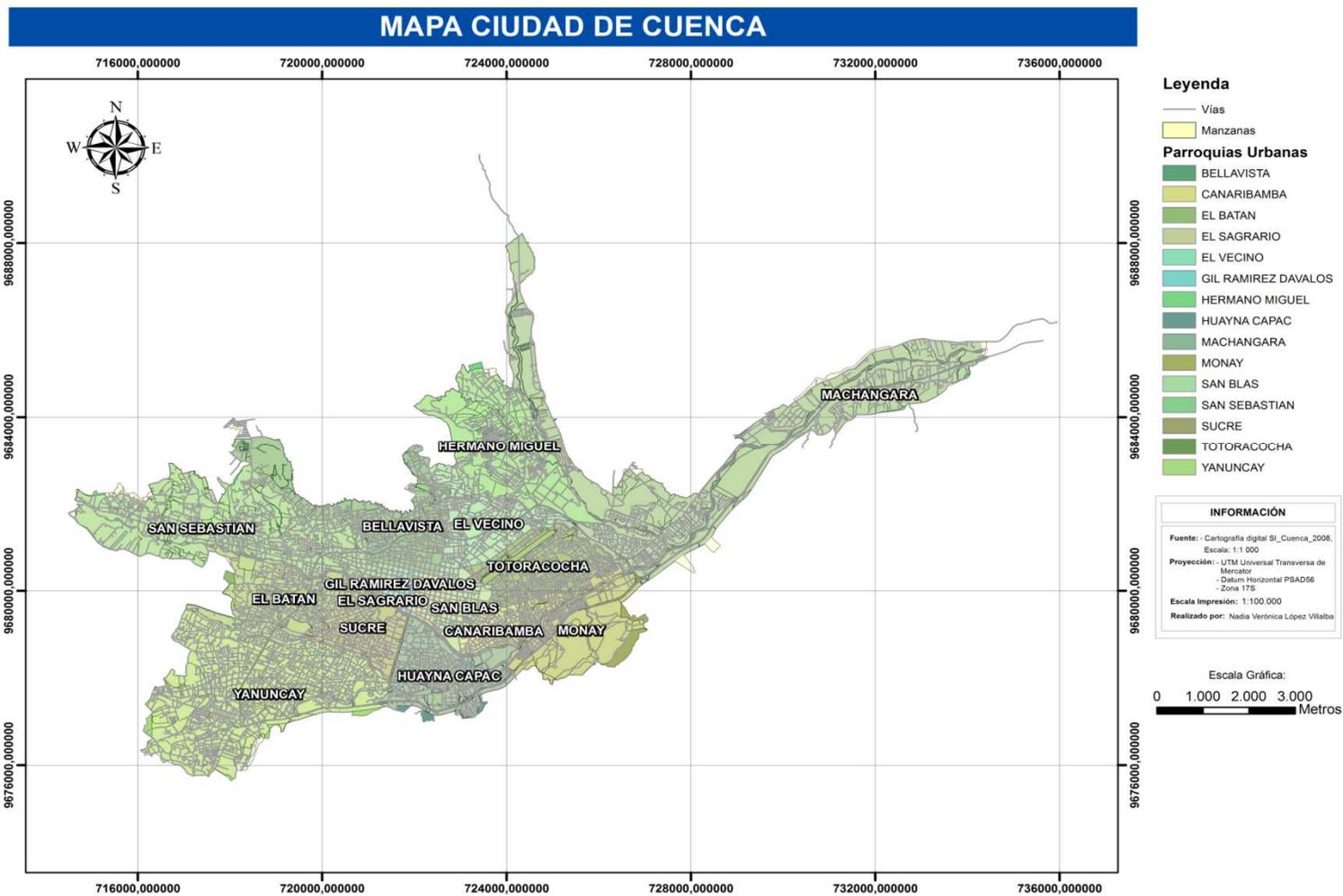


Gráfico 43. Mapa de la Ciudad de Cuenca, Datum Horizontal PSAD56 Z17S, Escala 1:1 000

A continuación se procederá a analizar los datos de las capas a utilizar:
 En la capa de vías se encuentran los siguientes atributos a ser empleados:

FID	Shape *	PARROQUIA	NOMBRE APE
402	Polyline M	MACHANGARA	PANAMERICANA NORTE
403	Polyline M	MACHANGARA	PANAMERICANA NORTE
404	Polyline M	MACHANGARA	PANAMERICANA NORTE
405	Polyline M	MACHANGARA	BATAILLON YAGUACHI
406	Polyline M	MACHANGARA	BATAILLON YAGUACHI
407	Polyline M	MACHANGARA	BATAILLON VENCEDORES
408	Polyline M	MACHANGARA	TRATADO DE GIRON
409	Polyline M	MACHANGARA	BATAILLON CARACAS
410	Polyline M	MACHANGARA	TRATADO DE GIRON
411	Polyline M	MACHANGARA	BATAILLON PICHINCHA
412	Polyline M	MACHANGARA	BATAILLON PICHINCHA
413	Polyline M	MACHANGARA	TRATADO DE GIRON
414	Polyline M	MACHANGARA	TRATADO DE GIRON
415	Polyline M	MACHANGARA	TRATADO DE GIRON
416	Polyline M	MACHANGARA	ESCUADRON CEDE,0
417	Polyline M	MACHANGARA	VICTORIA DEL PORTETE
418	Polyline M	MACHANGARA	BATAILLON VENCEDORES
419	Polyline M	MACHANGARA	VICTORIA DEL PORTETE
420	Polyline M	MACHANGARA	VICTORIA DEL PORTETE
421	Polyline M	MACHANGARA	VICTORIA DEL PORTETE
422	Polyline M	BELLAVISTA	RAFAEL MARIA ARIZAGA
423	Polyline M	BELLAVISTA	RAFAEL MARIA ARIZAGA
424	Polyline M	BELLAVISTA	RAFAEL MARIA ARIZAGA
425	Polyline M	BELLAVISTA	RAFAEL MARIA ARIZAGA
426	Polyline M	BELLAVISTA	RAFAEL MARIA ARIZAGA
427	Polyline M	BELLAVISTA	RAFAEL MARIA ARIZAGA
428	Polyline M	BELLAVISTA	RAFAEL MARIA ARIZAGA
429	Polyline M	FI VECINO	DIN BDAVN

Gráfico 44. Tabla de atributos de la capa Vías

Como se puede apreciar en el Gráfico 44, los datos de la columna nombre_ape no se encuentran correctamente depurados, pues existen en la mayoría de los casos errores de sintaxis. Los mismos serán corregidos en el procesamiento de datos.

Además de la información en la tabla de atributos se encontró que en la mayor parte de la ciudad los datos espaciales no se encuentran unidos correctamente, algunos ejemplos de estas fallas se pueden apreciar a continuación en los Gráficos 45, 46 y 47; sin la unión correcta, los nodos no coincidirán para armar la red geodésica del Centro de la Ciudad de Cuenca, estos problemas serán solucionados en la depuración.

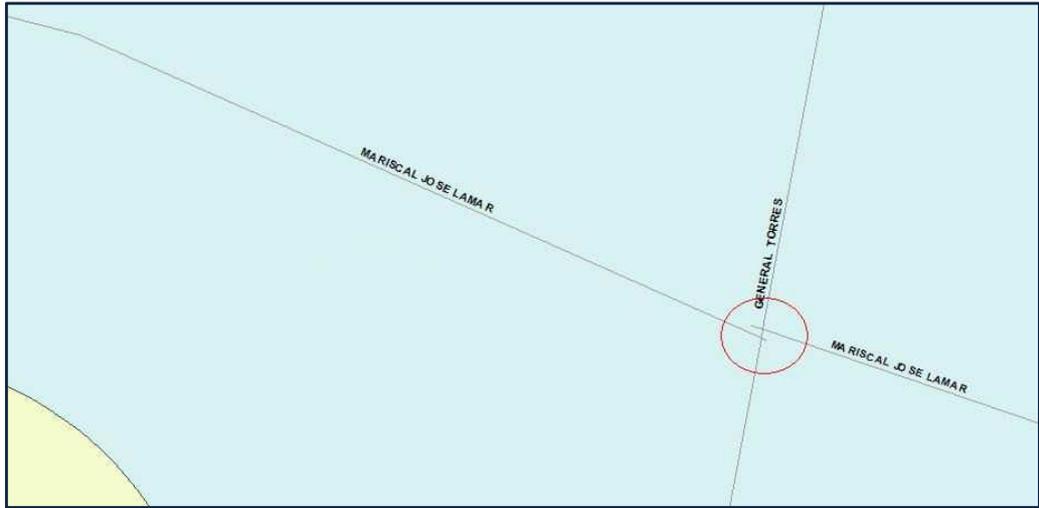


Gráfico 45. Error detectado en la parroquia Gil Ramírez Dávalos

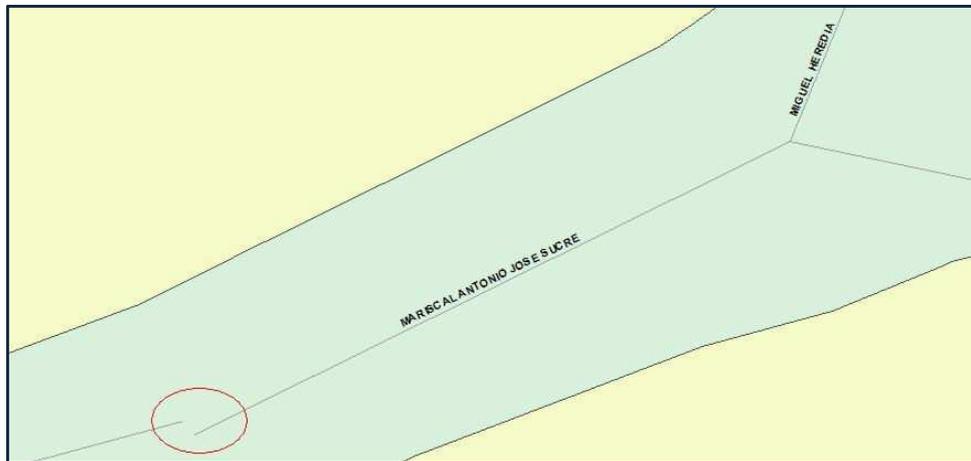


Gráfico 46. Error detectado en la parroquia San Sebastián



Gráfico 47. Error detectado en la parroquia San Blas

Los atributos de las capas de manzanas y parroquias urbanas no son relevantes para el proyecto. Para el estudio se cortará el centro de la ciudad de Cuenca, esto formará la cartografía en la cual se trabajará más adelante. Se proyectará al datum World Geodetic System de 1984 (WGS84) en UTM para su depuración; sin embargo, para finalizar del proceso de elaboración de mapa base se tomará en cuenta que la señal enviada por el GPS se encuentra con coordenadas geográficas, por lo cual, después de la depuración de los datos se realizará una proyección al World Geodetic System de 1984 (WGS84) en coordenadas geográficas (grados, minutos y segundos).

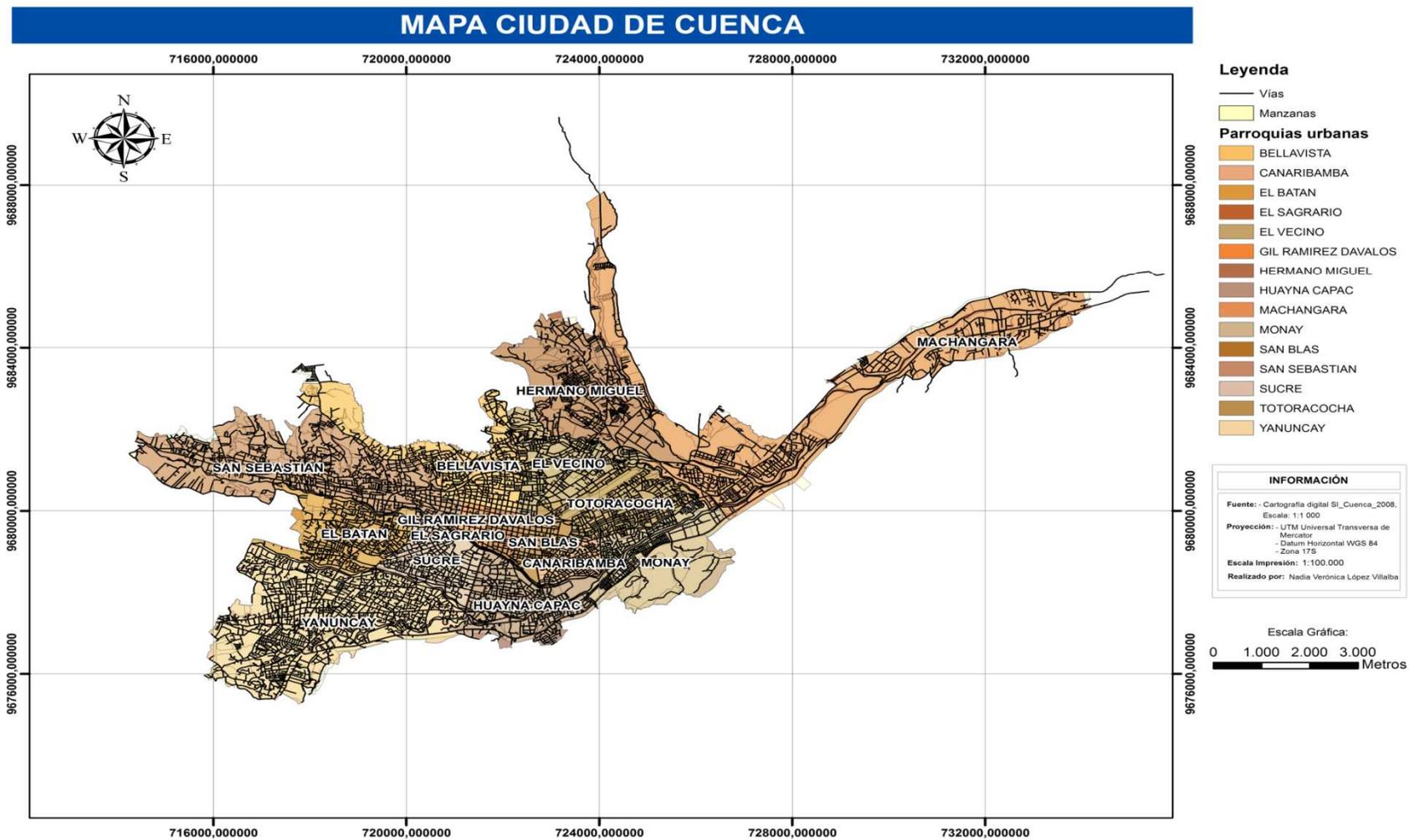


Gráfico 48. Mapa de la Ciudad de Cuenca, Datum Horizontal WGS84 Z17S, Escala 1:1 000

En adelante para el desarrollo del proyecto se utilizarán las nuevas capas generadas; las mismas serán depuradas y se extraerá los datos necesarios para su uso.

3.6.2. Depuración de datos

Una vez la información a utilizar se comenzará a la depuración de los datos espaciales. Los mismos serán extraídos para llenar la base de datos de nodos (TT_POINT) y bloques (TT_BLOCK). Adicionalmente se puede llenar los datos de las calles (TT_STREET).

Para solucionar los problemas de unión de las vías se debe tomar en cuenta que:

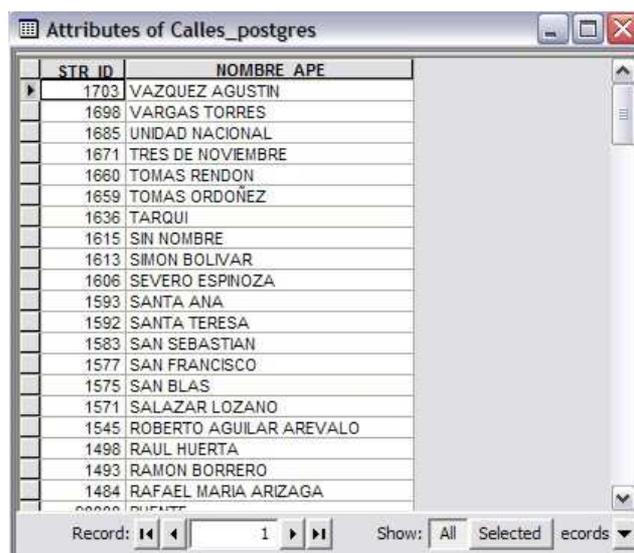
- Las vías es una capa de líneas, estas presentan problemas ya sea que, no están completamente unidas como se apreció en el gráfico 47, y/o sobrepasan la intersección de la línea visualizada en el gráfico 45. Cualquiera de los dos problemas provocan un nodo incorrecto, o simplemente coordenadas flotantes que jamás se interceptarán con otras, generando información innecesaria y a veces redundante.
- La tabla de puntos a llenar TT_POINT contendrá las coordenadas del inicio y final del bloque (línea de calle), así que es necesario estandarizar por cada intersección un solo POI_ID.
- Los bloques son secuencias de líneas, esto quiere decir que un solo bloque puede estar conformada por más de una línea.
- Una calle está conformada por varios bloques, pero su nombre sigue siendo el mismo, así que se agrupará y definirá para cada calle un único código (STR_ID), esto quiere decir que los nombres de las calles no se repetirán.
- Para la depuración de datos es mejor trabajar con el sistema de coordenadas UTM y no las geográficas. Pero posteriormente para el

funcionamiento con el GPS se trabajará con el sistema de coordenadas geográficos.

- Para la obtención de los puntos extremos de cada bloque es necesario pasar la capa de líneas a puntos.

Presentados estos puntos se procederá a la depuración de la información realizando los siguientes pasos:

- a) Estandarización de los nombres de las calles y asignación a un ID:
 - Se cambiarán los caracteres extraños como por ejemplo: ? por ñ, o quitamos los espacios innecesarios, tildes, etc.
 - Se dejará un único nombre y código por calle.
 - Se obtendrá la siguiente información presentada a continuación:



STR_ID	NOMBRE APE
1703	VAZQUEZ AGUSTIN
1698	VARGAS TORRES
1685	UNIDAD NACIONAL
1671	TRES DE NOVIEMBRE
1660	TOMAS RENDON
1659	TOMAS ORDÓÑEZ
1636	TARQUI
1615	SIN NOMBRE
1613	SIMON BOLIVAR
1606	SEVERO ESPINOZA
1593	SANTA ANA
1592	SANTA TERESA
1583	SAN SEBASTIAN
1577	SAN FRANCISCO
1575	SAN BLAS
1571	SALAZAR LOZANO
1545	ROBERTO AGUILAR AREVALO
1498	RAUL HUERTA
1493	RAMON BORRERO
1484	RAFAEL MARIA ARIZAGA
00000	PUENTE

Gráfico 49. Tabla de Atributos de la capa temática Calles_postgres

- Con esta tabla tenemos la información lista de la tabla TT_STREET que será posteriormente ingresada en la base de datos.
- b) Depuración de las líneas cuyos vértices conformarán los vértices de los nodos:

Debido a la gran cantidad de datos, se permitirá trabajar en masa la información espacial de una manera precisa mediante la utilización de la herramienta: ET GeoWizards 9.9 para ArcGIS 9.2, la misma puede ser descargada del internet.

ET GeoWizards es un conjunto de funciones poderosas que ayudan a los usuarios del software de ArcGIS a manipular datos con facilidad.

A continuación con el shape de vías cargado y la tabla depurada de calles correspondiente a TT_STREET se realiza un join data, lo que se pretende es asignar al nombre de la calle su código correspondiente.

En la tabla con el join debemos crear una columna que corresponda al STR_ID, así copiaremos la información de la columna del join a la tabla original. Además se creará la columna de BLK_ID (Id del bloque), en ella procederemos a realizar un field calculator asignando un código único a través del FID, en la siguiente tabla se muestra el resultado de dichos procesos:

FID	Shape	PARROQUIA	NOMBRE APE	STR ID	BLK ID
0	Polyline	EL VECINO	DE LA MERCED	418	31544
1	Polyline	EL VECINO	FEDERICO GUERRERO	666	31545
2	Polyline	SAN BLAS	JESUS ARRIAGA	886	8980
3	Polyline	EL SAGRARIO	ANTONIO VARGAS MACHUCA	100	8986
4	Polyline	MACHANGARA	SIN NOMBRE	1615	13534
5	Polyline	HERMANO MIGUEL	SIN NOMBRE	1615	32212
6	Polyline	HERMANO MIGUEL	SIN NOMBRE	1615	32213
7	Polyline	HERMANO MIGUEL	SIN NOMBRE	1615	701
8	Polyline	HERMANO MIGUEL	SIN NOMBRE	1615	18783
9	Polyline	HERMANO MIGUEL	SIN NOMBRE	1615	9657
10	Polyline	HERMANO MIGUEL	HUAMBOYA	824	14201
11	Polyline	HERMANO MIGUEL	SIN NOMBRE	1615	702
12	Polyline	HERMANO MIGUEL	HUAMBOYA	824	5263
13	Polyline	HERMANO MIGUEL	COJIMIES	325	27660
14	Polyline	HERMANO MIGUEL	COJIMIES	325	5264
15	Polyline	HERMANO MIGUEL	COJIMIES	325	14202
16	Polyline	HERMANO MIGUEL	COJIMIES	325	703
17	Polyline	HERMANO MIGUEL	COJIMIES	325	14203
18	Polyline	HERMANO MIGUEL	COJIMIES	325	27661
19	Polyline	TOTORACOCHA	ANTISANA	81	31621
20	Polyline	HERMANO MIGUEL	SOR. LUISA DE JESUS CORDERO	1621	18784
21	Polyline	HERMANO MIGUEL	CIMARRONES	317	32214
22	Polyline	HERMANO MIGUEL	CIMARRONES	317	32215

Gráfico 50. Tabla de atributos de la capa temática Vías

Este proceso es muy importante para los siguientes pasos, puesto que a partir de esto se definirán los vértices de las líneas formando los nodos, así sabremos que nodo corresponde a que bloque. Después generamos los vértices inicial y final, los datos se verán reflejados en la capa de puntos de esta manera:



Gráfico 51. Información gráfica

La tabla generada es la siguiente:

FID	Shape *	PARROQUIA	NOMBRE APE	BLK ID	COD CALLE
0	Point	EL VECINO	DE LA MERCED	31544	418
1	Point	EL VECINO	DE LA MERCED	31544	418
2	Point	ELVECINO	FEDERICO GUERRERO	31545	666
3	Point	ELVECINO	FEDERICO GUERRERO	31545	666
4	Point	SAN BLAS	JESUS ARRIAGA	8980	886
5	Point	SAN BLAS	JESUS ARRIAGA	8980	886
6	Point	EL SAGRARIO	ANTONIO VARGAS MACHUCA	8986	100
7	Point	EL SAGRARIO	ANTONIO VARGAS MACHUCA	8986	100
8	Point	MACHANGARA	SIN NOMBRE	13534	1615
9	Point	MACHANGARA	SIN NOMBRE	13534	1615
10	Point	HERMANO MIGUEL	SIN NOMBRE	32212	1615
11	Point	HERMANO MIGUEL	SIN NOMBRE	32212	1615
12	Point	HERMANO MIGUEL	SIN NOMBRE	32213	1615
13	Point	HERMANO MIGUEL	SIN NOMBRE	32213	1615
14	Point	HERMANO MIGUEL	SIN NOMBRE	701	1615
15	Point	HERMANO MIGUEL	SIN NOMBRE	701	1615
16	Point	HERMANO MIGUEL	SIN NOMBRE	18783	1615
17	Point	HERMANO MIGUEL	SIN NOMBRE	18783	1615
18	Point	HERMANO MIGUEL	SIN NOMBRE	9657	1615
19	Point	HERMANO MIGUEL	SIN NOMBRE	9657	1615
20	Point	HERMANO MIGUEL	HUAMBOYA	14201	824
21	Point	HERMANO MIGUEL	HUAMBOYA	14201	824
22	Point	HERMANO MIGUEL	SIN NOMBRE	702	1615

Gráfico 52. Tabla de atributos de la capa temática POINT_Vias

Se puede apreciar que para cada BLK_ID se generó dos nodos.

A continuación se creará las columnas correspondientes a las coordenadas X1 y Y1. Se debe tener en cuenta que este es el motivo por el cual se trabaja en el sistema de coordenadas UTM, cuya unidad de medida es el metro, la información obtenida se exportará a Excel, la misma será transformada a un archivo de extensión *.CSV.

El objetivo de la creación del archivo en *.CSV es procesar a través de un procedimiento los datos. Dicho procedimiento utiliza la ecuación de la recta en la cual une los puntos más cercanos dentro de un radio asignado, creándose así una posición única para los puntos.

Una vez procesados los datos se deben exportar a .dbf y posteriormente se grafica sus puntos en ArcGIS.

Se puede apreciar la diferencia entre el archivo original contra el procesado en los siguientes gráficos:

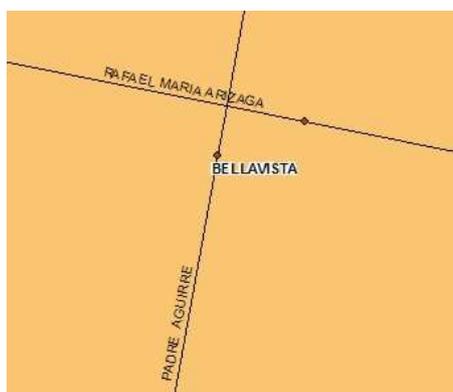


Gráfico 53. Archivo original

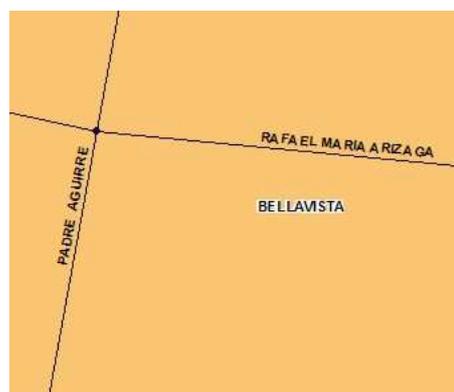


Gráfico 54. Archivo procesado

Se debe tomar en cuenta que este tipo de depuración puede requerir correr el procedimiento de la ecuación de la recta varias veces para obtener el resultado deseado, además se somete a una revisión visual para corroborar los datos.

Con los nodos obtenidos ahora utilizaremos la herramienta ET GeoWizards, ésta nos permitirá generar las líneas a partir de los puntos.

En el siguiente gráfico podemos ver como se generó las líneas en la nueva capa a partir de los nodos.



Gráfico 55. Generación de líneas en la nueva capa a partir de los nodos

A continuación procedemos a realizar un join con el BLK_ID, el mismo nos permitirá llenar las columnas correspondientes al POI_ID0, POI_ID1; así se crea la tabla TT_BLOCK:

FID	Shape	STR ID	BLK ID	POI ID0	POI ID1
0	Polyline	418	31544	19939	19852
1	Polyline	666	31545	19957	19937
2	Polyline	886	8980	19822	19872
3	Polyline	100	8986	19043	19068
4	Polyline	100	31553	19068	19085
5	Polyline	100	31554	18975	18994
6	Polyline	100	26932	18994	19029
7	Polyline	810	23461	18524	18252
8	Polyline	810	27900	18252	17867
9	Polyline	810	23462	17867	17535
10	Polyline	1659	18081	19341	19281
11	Polyline	100	22513	18924	18958
12	Polyline	1615	1363	20046	20074
13	Polyline	1484	4629	17193	17204
14	Polyline	100	18082	18958	18971
15	Polyline	1484	13541	17204	17352
16	Polyline	1484	104	17352	17361
17	Polyline	1484	105	17361	17511
18	Polyline	1484	22584	16098	16238
19	Polyline	1484	9053	16238	16282
20	Polyline	1484	4630	15812	15909
21	Polyline	1484	4631	15909	16098
22	Polyline	100	26933	18971	18975

Gráfico 56. Tabla de atributos de la capa TT_BLOCK_10k_CGSWGS84_Pos

Para finalizar la validación y levantamiento de información procederemos a crear las columnas restantes de la tabla TT_BLOCK y las llenaremos según lo más cercano a la realidad. Para la columna BLK_ID se calculó su geometría utilizando la longitud en metros. A partir de ello se proyecta los shapes al sistema de coordenadas geográficas. En la siguiente tabla se muestra la información levantada, así como el mapa obtenido.

Attributes of tt_block_10k_CGS_WGS84_Postgres

FID	Shape	STR ID	BLK ID	POI ID0	POI ID1	BLK ADDR	BLK OPEN	BLK D
0	Polyline	418	31544	19939	19852	1	1	91,1
1	Polyline	666	31545	19957	19937	0	1	116,59
2	Polyline	886	8980	19822	19872	0	1	94,37
3	Polyline	100	8986	19043	19068	0	1	49,45
4	Polyline	100	31553	19068	19085	0	1	26,68
5	Polyline	100	31554	18975	18994	0	1	40,15
6	Polyline	100	26932	18994	19029	0	1	72,96
7	Polyline	810	23461	18524	18252	2	1	99,85
8	Polyline	810	27900	18252	17867	2	1	138,88
9	Polyline	810	23462	17867	17535	2	1	118,8
10	Polyline	1659	18081	19341	19281	0	1	110,89
11	Polyline	100	22513	18924	18958	0	1	73,56
12	Polyline	1615	1363	20046	20074	2	1	10,77
13	Polyline	1484	4629	17193	17204	1	1	4,67
14	Polyline	100	18082	18958	18971	0	1	28,3
15	Polyline	1484	13541	17204	17352	1	1	57,32
16	Polyline	1484	104	17352	17361	1	1	2,77
17	Polyline	1484	105	17361	17511	1	1	52,36
18	Polyline	1484	22584	16098	16238	1	1	66,75
19	Polyline	1484	9053	16238	16282	1	1	22,15
20	Polyline	1484	4630	15812	15909	1	1	26,31
21	Polyline	1484	4631	15909	16098	1	1	71,09
22	Polyline	100	26933	18971	18975	0	1	7,3

Record: 0 Show: All Selected Records (0 out of 1715 Selected)

Gráfico 57. Tabla de atributos de la capa temática tt_block_10k_CGS_WGS84_Postgres



Gráfico 58. Representación de la tabla TT_BLOCK, TT_POINT y áreas verdes

Con estos shapes en Datum WGS84 - Z17S Coordenadas esféricas se procederá a llenar la base de datos. Finalizando así obtención, análisis, depuración y levantamiento de información geográfica. En el Gráfico 59 se puede apreciar el mapa del centro de la ciudad de Cuenca con el que se trabajará en adelante.

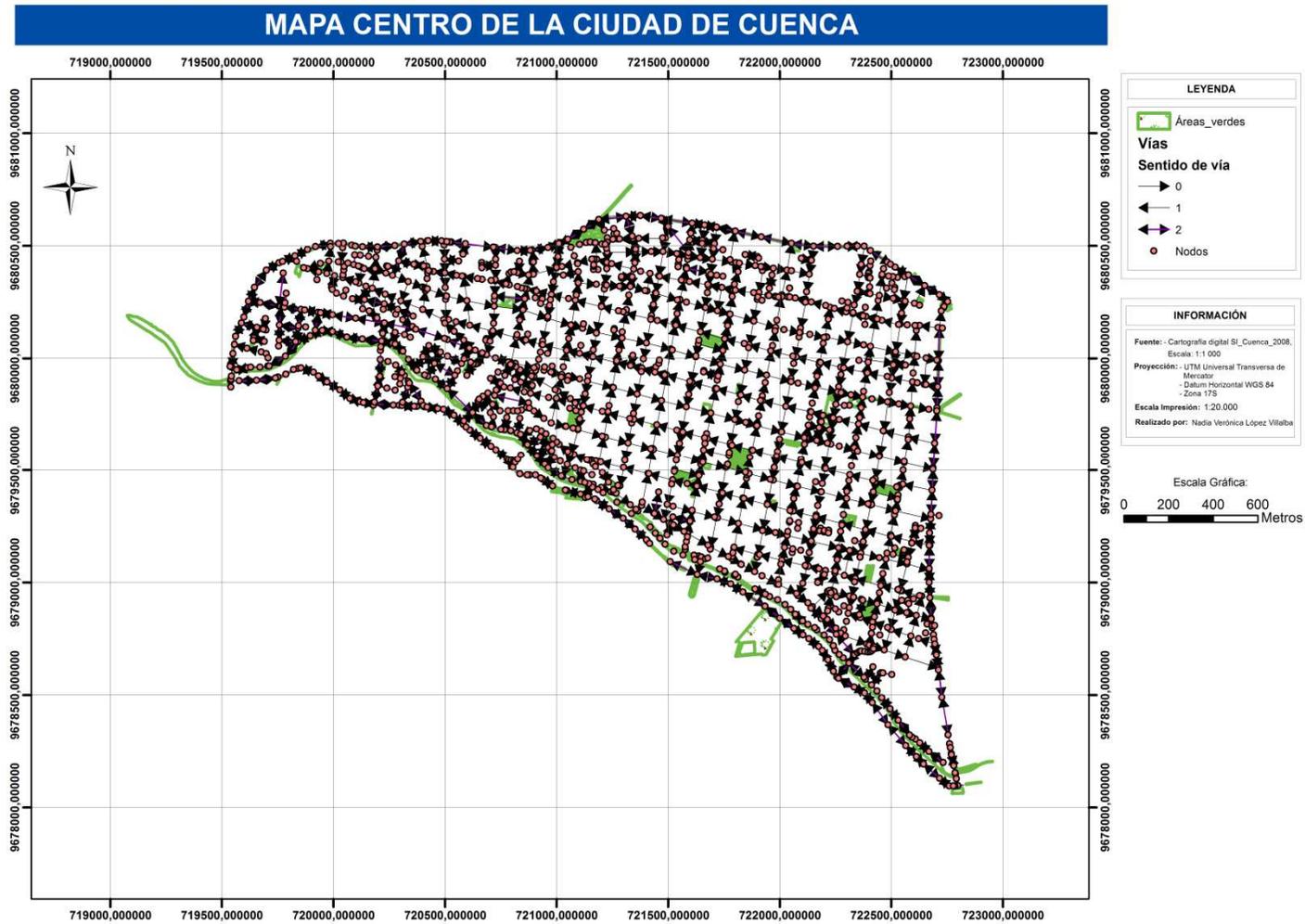


Gráfico 59. Mapa del centro de la Ciudad de Cuenca

CAPITULO IV

PROGRAMACIÓN DEL MODELO PARA GESTIÓN DE ENTREGAS POR RUTAS ÓPTIMAS, MEDIANTE GPS Y CONEXIÓN CELULAR

4.1. INTRODUCCIÓN

En este capítulo se determinan los procedimientos a seguir para realizar la programación del modelo para gestión de entregas, determinando el uso y comportamiento de todas las herramientas mencionadas en el Capítulo II, así como, las librerías fundamentales para la captura de información e inferencia de rutas óptimas, obteniendo de esta manera un sistema inteligente.

4.2. CREACIÓN DE LA BASE DE DATOS

La base de datos constituye el cimiento de todas las aplicaciones, en el Capítulo III se estableció la estructura adecuada y las tablas necesarias para el almacenamiento de la información, denominaremos a la base con el nombre de Traffic. El Anexo 2 es una carpeta que contiene el SQL ejecutado para la creación. A continuación se puede visualizar la estructura de la base creada:

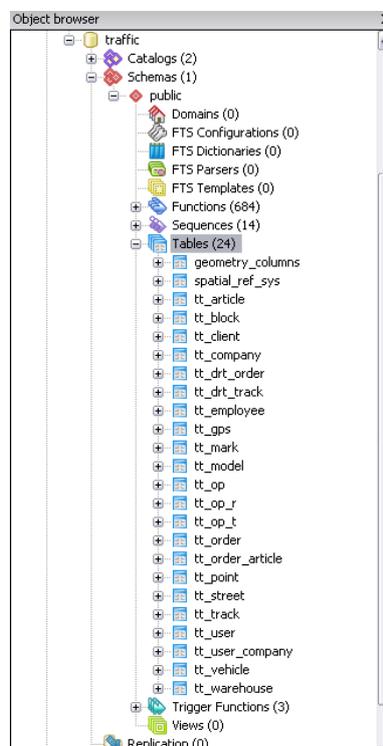


Gráfico 60. Estructura de Base creada

Una vez establecida la base de datos es necesario crear los triggers para la creación automática del dato geométrico para las tablas de TT_BLOCK, TT_CLIENT, TT_WAREHOUSE y TT_TRACK. Para cada uno de los casos se escribirá con el siguiente formato:

```
UPDATE Tabla SET the_geom=GeomFromText( 'POINT(' || longitud || ' ' || latitud || ')', 4326);
```

En donde:

Tabla: Hace referencia a tt_block, tt_client, tt_warehouse y tt_track.

The_geom: La columna geométrica creada para almacenar el geom.

Longitud: Columna correspondiente a la longitud.

Latitud: Columna correspondiente a la latitud.

4326: Datum geográfico correspondiente al WGS84, si se desea en otro sistema se debe cambiar según su código.

4.3. CONFIGURACIÓN DEL GPS

Para realizar el rastreo de vehículos tenemos que tener en cuenta muchos factores como son el modelo de GPS a usar según el vehículo a rastrear, la aplicación de captura de señal, la operadora que utilizará el GPS, entre otros. En el Capítulo II se estableció las herramientas y librerías a utilizar.

Se determinó que es mejor utilizar el servicio GPRS con transmisión de datos al servidor, ya que, el proceso es más óptimo recibiendo directamente la información sin pasar por algún otro proceso como es el SMS.

La configuración del GPS Trax S6 se hizo en base al manual proporcionado por el proveedor en nuestro caso GTE, además se cuenta con la operadora PORTA para obtener el servicio de GPRS. A continuación se describen los pasos para configurar el GPS:

1. Contratar el servicio de operadora para GPRS

Realizar un contrato con la operadora deseada en este caso PORTA para obtener el servicio GPRS, el servicio se debe pagar mensualmente con un

costo determinado por la operadora, se debe proporcionar los siguientes parámetros al proveedor de servicios como son:

- **La dirección IP del servidor:** 201.238.173.196 correspondiente a www.totalhome.com.ec.

- **Puerto:** 20xx

Una vez realizado el contrato, el proveedor de servicios nos dará los siguientes parámetros necesarios para la configuración del GPS:

- **APN:** xxxxxxxxxxxx.PORTA.COM.EC

- **SIMCARD:** 8959301000279931xx

- **IP local del dispositivo:** 172.031.13x.001

- **IP pública del servidor:** 201.238.173.196

- **Puerto del servidor:** 20xx

Por motivos de seguridad se reemplazo ciertos datos por xx para el buen uso de la empresa.

2. Configuración del GPS

Colocar el SIM con el servicio GPRS activo en el dispositivo GPS. GTE provee de un programa para la configuración del GPS, para lo cual debemos conectar el GPS a través del puerto USB a la máquina para realizar su debida configuración:

El programa terminal SERIE y UDP para los equipos TRAX se denomina CommLog:

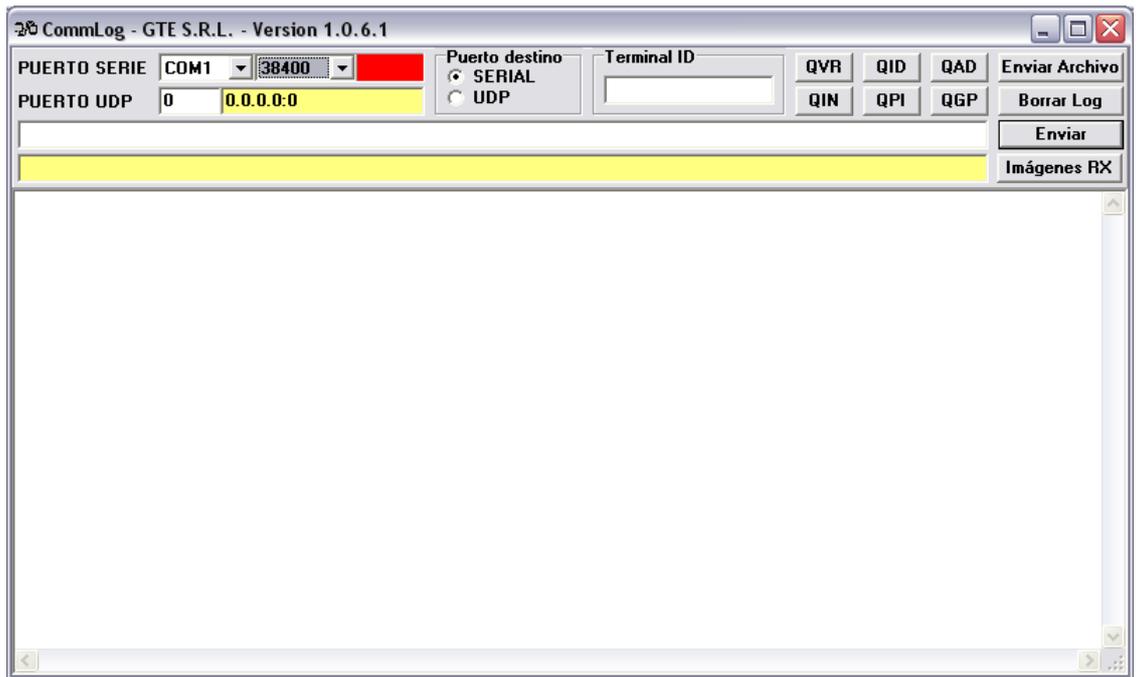


Gráfico 61. CommLog – Programa Terminal Serie y UDP para equipos TRAX

A través de este programa se envía las instrucciones al GPS, se creó un archivo con los siguientes parámetros de configuración para guardar los eventos en log y descargar automáticamente a la IPO:

COMANDOS	INSTRUCCIÓN
>SFD55AA<	Elimina la configuración del equipo
>SGR xxxxxxxxxx.PORTA.COM.EC,PORTA,PORTA<	Set APN
>SIP0 201.238.173.196,010001000020000101<	Set IP0 donde va reportar el móvil
>SDU20xx<	Set puerto UDP 20xx
>SGR xxxxxxxxxx.PORTA.COM.EC,PORTA,PORTA<	Set APN
>SIT6100100<	Set entrada 6 (estado de batería principal) con demora de 10 s para estado activo y 10 s para estado inactivo.
>SIT7005010<	Set entrada 7 (estado de ignición) con demora de 5 s para estado activo y 10 s para estado inactivo.
>SKO072002400960<	Apaga el GPS a los 4 minutos de apagada la ignición, modo reposo a los 16 minutos, dormido total a las 12 horas
>SDC00*030<	Configuración de disparos de cambio de rumbo
>STD00000120000000<	TD00 CADA 120 SEGUNDOS
>STD01000000001000<	TD01 CADA 1000 METROS
>SGS01000100<	GS01 VELOCIDADES SUPERIORES A 100 Kmph
>SBA0102503<	BA01 frenada abrupta. desaceleración de 25kmph en 3 Seg.
>SED08 TD00-+&IN07+ + PH TR0 {STD00*}<	Programación de eventos
>SED09 TD01-+ + PH IP0 {STD01*}<	Programación de eventos tiempo
>SED10 DC00-+ + PH IP0 {SDC00R}<	Programación de eventos ángulo
>SED11 GS01+- + PH IP0<	Cuando la velocidad supera los 100 Kmph
>SED11 BA01-+ + PH IP0<	Cuando existe una frenada abrupta
>SED04 IN06-+ + PH IP0<	Al conectarse la alimentación se envía PI A IP0
>SED05 IN06+- + PH IP0<	Al desconectarse la alimentación se envía PI A IP0
>SED06 IN07-+ + PH IP0<	Cuando se activa la ignición se envía PI A IP0
>SED07 IN07+- + PH IP0<	Cuando sale de ignición se envía PI A IP0
>SEDU55AA<	Actualiza los eventos en la memoria de ejecución
>SGR*<	Reset modem
>SSR55AA*<	Reset equipo

Tabla 10. Parámetros de Configuración para guardar eventos LOG

Enviamos el archivo de configuración al GPS, ahora este se encuentra listo para ser conectado al vehículo.

4.4. APLICACIÓN PARA LA CAPTURA DE PAQUETES DEL REPORTE ENVIADO POR GPS

JPCAP es la biblioteca que se utilizó para la aplicación que captura la señal del GPS a través de un puerto definido. Los datos son capturados continuamente desglosando la información e ingresando a la tabla de la base de datos correspondiente, con esto se tendría el registro del rastreo y ruta que ha seguido el conductor.

Para realizar la captura de los paquetes se desarrolló la aplicación correspondiente en Netbeans IDE 6.1 que contiene la biblioteca JPCAP. A continuación se detalla los pasos a seguir para la captura de paquetes:

1. Obtener la lista de las interfaces de red en la máquina:

Esto nos permitirá capturar los paquetes de una red, para ello, Jpcap proporciona el método `JpcapCaptor.getDeviceList ()`; dicho método devuelve una matriz de objetos `NetworkInterface`.

Un objeto `NetworkInterface` contiene información sobre la interfaz de red correspondiente: su nombre, descripción, las direcciones IP y MAC, y el nombre y la descripción `datalink`.

Con el siguiente código se obtiene la lista de las interfaces de red e imprime en pantalla la información:

```

//Obtiene la lista de interfaces del network
NetworkInterface[] devices = JpcapCaptor.getDeviceList();

//Para cada interfaz de network
for (int i = 0; i < devices.length; i++) {
    //Imprime en pantalla el nombre y descripción
    System.out.println(i+": "+devices[i].name + "(" + devices[i].description+"");

    //Imprime en pantalla el link del nombre y descripción
    System.out.println(" datalink: "+devices[i].datalink_name + "(" +
devices[i].datalink_description+"");

    //Imprime en pantalla el dirección MAC
    System.out.print(" MAC address:");
    for (byte b : devices[i].mac_address)
        System.out.print(Integer.toHexString(b&0xff) + ":");
    System.out.println();
    //Imprime en pantalla la dirección IP, subnet y mascara de la dirección broadcast
    for (NetworkInterfaceAddress a : devices[i].addresses)
        System.out.println(" address:"+a.address + " " + a.subnet + " " + a.broadcast);
}

```

2. Abrir una interfaz de red

Una vez que se obtenga la lista de las interfaces de red se debe elegir la interfaz de red para la captura de paquetes, se puede abrir la interfaz mediante el uso del método `JpcapCaptor.openDevice()`. El siguiente fragmento de código muestra cómo abrir una interfaz de red:

```

NetworkInterface[] devices = JpcapCaptor.getDeviceList();
int index=...; // setea el índice de la interfaz que se desea abrir

//Abre una interfaz con openDevice(NetworkInterface intrface, int snaplen, boolean promisc, int
to_ms)
JpcapCaptor captor=JpcapCaptor.openDevice(device[index], 65535, false, 20);

```

Al llamar al método `JpcapCaptor.openDevice()`, se puede especificar los siguientes parámetros:

Nombre	Propósito
Network Interface	Interfaz de red que se desea abrir.
int snaplen	Número máximo de bytes para capturar a la vez.
boolean promiscs	True si desea abrir la interfaz en modo promiscuo, y false en caso contrario. <ul style="list-style-type: none">- En modo promiscuo se puede capturar todos los paquetes, es decir, aunque su origen o la dirección MAC de destino no es la misma que la dirección MAC de la interfaz que se están abriendo.- En el modo no promiscuo, sólo se puede capturar los paquetes enviados y recibidos por su anfitrión.
int to_ms	Establece un valor de tiempo de captura en milisegundos.

Tabla 11. Parámetros del método `JpcapCaptor.openDevice()`

`JpcapCaptor.openDevice()` devuelve una instancia de `JpcapCaptor`.

3. Captura de paquetes de la interfaz de red

Una vez obtenida una instancia de la `JpcapCaptor`, se puede capturar los paquetes de la interfaz.

Hay dos enfoques principales para la captura de paquetes utilizando una instancia `JpcapCaptor`: el uso de un método callback o capturando paquetes one-by-one.

Se utilizó el método de callback ya que es necesario para la captura continua de paquetes. Se debe aplicar un método de callback mediante la definición de una nueva clase que implementa la interfaz `PacketReceiver`. La interfaz define un método `PacketReceiver receivePacket()`, por lo que necesita para implementar un `receivePacket ()` en su clase.

La siguiente clase implementa un `receivePacket()`, que imprime un paquete capturado.

```

class PacketPrinter implements PacketReceiver {
//this method is called every time Jpcap captures a packet
    public void receivePacket(Packet packet){
        //just print out a captured packet
        System.out.println(packet); }}

```

A continuación, se puede llamar a cualquiera `JpcapCaptor.processPacket ()` o `JpcapCaptor.loopPacket()` para iniciar la captura mediante el método de callback. Se puede especificar el número de paquetes a capturar, -1 para continuar la captura de paquetes hasta el infinito.

```

JpcapCaptor captor=JpcapCaptor.openDevice(device[index], 65535, false, 20);
//call processPacket() to let Jpcap call PacketPrinter.receivePacket() for every packet
capture.captor.processPacket(10,new PacketPrinter());
captor.close();

```

4. Grabar en base de datos

Una vez que se ha establecido los procedimientos y métodos necesarios para la captura debemos realizar el desglose de la información recibida para el ingreso correspondiente en la base de datos. El reporte llega en el siguiente formato:

```

RX: >RPH270111224350-0289884-
07899167350000107000010130009000000000CF06;ID=6113;#LOG:0001;*0B<
RX: >RPH270111224433-0289905-
07899174203000108000010100004000000000CF08;ID=6113;#LOG:0002;*08<
RX: >RPH270111224451-0289970-
078992022610181080000101000040000000048CF10;ID=6113;#LOG:0003;*05<
RX: >RPH270111224530-0289941-
078994012080211080000101000040000000274CF10;ID=6113;#LOG:0004;*0B<
RX: >RPH270111224632-0290158-
078994262470191080000101000990000000520CF10;ID=6113;#LOG:0005;*04<

```

En el Capítulo II, numeral 2.6.4. Reportes Comando PH (Q-R) se detalla el significado de cada tramo.

Se creó la clase `ConnPostgres` para desglosar la información, dentro de la misma se establece la conexión a la base de datos y se creó el método para insertar el track, a continuación tenemos la función de insertar:

```

public boolean insertarTrackPostgres(String fecha, String hora, int age, String chksum, String
digital,
int dir, int evento, int hdop, int id, double lat,
double longitud, int odometer, int posicion,
int satelite, int signal, int speed, int statusconn, int statusreg, String tipoReporte, int ste, String
log)
{
    PreparedStatement pstmt = null;
    Connection conn = null;

    try {
        conn = DriverManager.getConnection(URL, login, password);
        pstmt = conn.prepareStatement("INSERT INTO tt_track
VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)");

        pstmt.setInt(1,id);
        pstmt.setString(2,tipoReporte);
        pstmt.setString(3,fecha);
        pstmt.setString(4,hora);
        pstmt.setDouble(5,lat);
        pstmt.setDouble(6,longitud);
        pstmt.setInt(7,dir);
        pstmt.setInt(8,speed);
        pstmt.setInt(9,ste);
        pstmt.setInt(10,satelite);
        pstmt.setInt(11,age);
        pstmt.setInt(12,posicion);
        pstmt.setInt(13,hdop);
        pstmt.setInt(14,statusconn);
        pstmt.setInt(15,statusreg);
        pstmt.setInt(16,signal);
        pstmt.setInt(17,odometer);
        pstmt.setString(18,digital);
        pstmt.setInt(19,evento);
        pstmt.setString(20,log);
        pstmt.execute();
        System.out.println("Registro insertado.");

    } catch (SQLException ex) {
        ex.printStackTrace();
    }
}

```

```
return false;
} finally {
try {
if (pstm != null) {
pstm.close();
}
if(conn!=null)
conn.close();

} catch (SQLException ex) {
ex.printStackTrace();
return false;
}
}
return true;
}
```

Con ello finaliza la aplicación de captura de señal enviada por el GPS a través del servicio GPRS. A continuación se muestra la pantalla del programa con su funcionalidad:

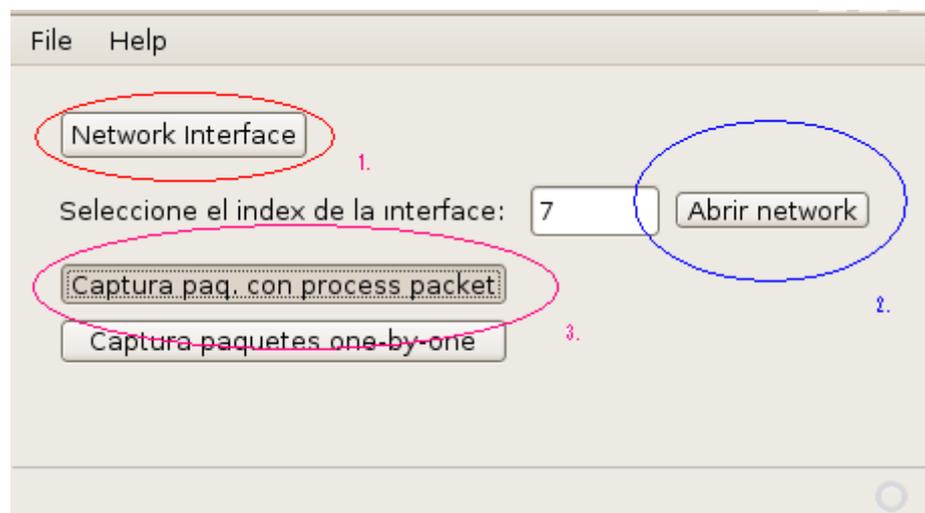
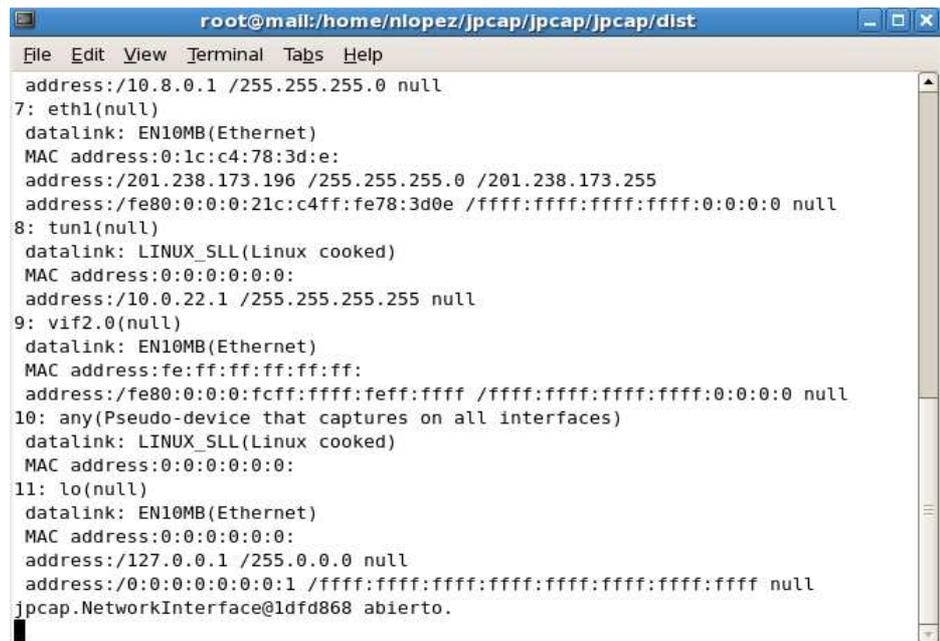


Gráfico 62. Pantalla captura de paquetes del reporte enviado por GPS

1. Para usar la aplicación primero debemos obtener la lista de interfaces con el botón: **Network interface**: Obtiene la lista de las interfaces de red en la máquina. Nos presentará la siguiente pantalla:



```
root@mail:/home/nlopez/jpcap/jpcap/jpcap/dist
File Edit View Terminal Tabs Help
address:/10.8.0.1 /255.255.255.0 null
7: eth1(null)
  datalink: EN10MB(Ethernet)
  MAC address:0:1c:c4:78:3d:e:
  address:/201.238.173.196 /255.255.255.0 /201.238.173.255
  address:/fe80:0:0:0:21c:c4ff:fe78:3d0e /ffff:ffff:ffff:ffff:0:0:0:0 null
8: tun1(null)
  datalink: LINUX_SLL(Linux cooked)
  MAC address:0:0:0:0:0:0:
  address:/10.0.22.1 /255.255.255.255 null
9: vif2.0(null)
  datalink: EN10MB(Ethernet)
  MAC address:fe:ff:ff:ff:ff:ff:
  address:/fe80:0:0:0:fcff:ffff:feff:ffff /ffff:ffff:ffff:ffff:0:0:0:0 null
10: any(Pseudo-device that captures on all interfaces)
  datalink: LINUX_SLL(Linux cooked)
  MAC address:0:0:0:0:0:0:
11: lo(null)
  datalink: EN10MB(Ethernet)
  MAC address:0:0:0:0:0:0:
  address:/127.0.0.1 /255.0.0.0 null
  address:/0:0:0:0:0:0:1 /ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff null
jpcap.NetworkInterface@1dfd868 abierto.
```

Gráfico 63. Resultado Network Interface

2. A continuación elegimos la IP del servidor con el que vamos a trabajar, en nuestro caso el número 7. Abrimos la interfaz de red ingresada con **Abrir network**.
3. Después procedemos a la captura de los paquetes de la interfaz de red continuamente presionando el botón **Captura paquetes con process packet**. Internamente se graban los datos directamente en la base de datos en Postgres.

Para ejecutar el programa realizamos la instrucción:

```
java -jar jpcap.jar
```

4.5. APLICACIÓN PARA LA RECEPCIÓN DE ESTADO DE ORDEN ENTREGA – RECEPCIÓN AL SERVIDOR A TRAVÉS DE SMS

JavaxComm es la biblioteca que se utilizó para la aplicación que recibe un sms con el estado de la orden de entrega – recepción a través del celular SAMSUNG

SGH-E256, el mensaje es procesado y enviado al servidor para su actualización automática.

Se integró la librería comm.jar a Netbeans para obtener acceso a los puertos y por ende la información respectiva al mensaje recibido por el celular en formato texto o PDU dependiendo de la configuración del mismo. Para comunicarse con el módem GSM, es necesario transmitir los comandos AT estándar a través del puerto serie. Para lo cual se realizó los siguientes pasos:

1. Agregar librería javax comm

La nomenclatura <JDK> indica el directorio donde se ha instalado el paquete de desarrollo de Java:

- Copiar el fichero Win32com.dll a <JDK>\jre\bin.
- Copiar el archivo comm.jar a <JDK>\jre\lib\ext.
- Copiar javax.comm a <JDK>\jre\lib este fichero contendrá los controladores que nos permitirá crear nuestros propios drivers de puertos.

2. Configurar el CNMI del celular Samsung

El CNMI define cómo el módem responde cuando se recibe un SMS. Para configurar el CNMI del celular Samsung se debe utilizar la siguiente sintaxis:

$$\text{AT+CNMI} = 2,1,0,0,0$$

Esto permitirá la recepción correcta de los mensajes.

3. Establecer formato de mensaje preferido (+ CMGF)

Existen dos formatos de mensaje de apoyo: el modo de texto y modo PDU.

Para establecer el formato de mensaje preferido se utiliza la siguiente sintaxis:

Sintaxis	Formato
AT+CMGF=1	Texto
AT+CMGF=0	PDU

Posible respuesta del módem GSM: OK

4. Recepción de mensajes

Dependiendo del formato que se utilizará (texto o PDU), la recepción de mensajes difiere; para acceder a las funciones que separan el mensaje recibido de su correspondiente cabecera y cuerpo se creó la siguiente clase:

CLASE CMGR:

La clase CMGR hace referencia a todos los procesos relacionados en la recepción de mensajes CMGR y sus diversos tipos: "REC READ", "REC UNREAD", "UNSENT" y "CMTI". El mensaje que se capta, puede estar en formato texto o PDU. Para cada formato se presenta el ejemplo de la trama de un mensaje:

PDU:

```
+CMGR: "REC READ", "+59392880591", "", "08/11/24,23:21:59-20", 145, 100, 0, 0, "+59395897705", 255, 162
```

```
05000346030186F5B09BFC068DC3EDB43BDC7ECF41F0B71CC40E83E
C697218E47ECF41E470FB3D078DEB65373D0C8AD7CBA0791AE4AEB
BC761109BFD9687DBEF39C8FD06CDC3E272FB3D07B1DFA078BD0C2
ACF4174797A4E2FEB32E10689E06B9EBEE7118340E97DBEF39481D6
E87E7A0305C5E7693CBEDF71C1406B1CBF6B09B1E96BBDF73501E34
4F83DC,5
```

En la siguiente tabla se especifica las partes del mensaje:

Formato PDU		
Cabecera	+CMGR: READ", "+59392880591", " ", "08/11/24,23:21:59- 20",145,100,0,0,"+59395897705",255 ,162	"REC Número de centro de servicio Fecha Remitente Hora
Cuerpo	05000346030186F5B09BFC068DC3E DB43BDC7ECF41F0B71CC40E83E C697218E47ECF41E470FB3D078DE B65373D0C8AD7CBA0791AE4AEB BC761109BFD9687DBEF39C8FD06 CDC3E272FB3D07B1DFA078BD0C 2ACF4174797A4E2FEBC32E10689E 06B9EBEE7118340E97DBEF39481D 6E87E7A0305C5E7693CBEDF71C14 06B1CBF6B09B1E96BBDF73501E3 44F83DC,5 OK	Mensaje en formato PDU El mismo será transformado a formato texto obteniéndose la siguiente cadena: Cuando caminamos por la vida nos damos cuenta que si nunca lloramos no sabemos lo que es tristeza. si nunca caemos jamás aprendemos a levantarnos y si n

Tabla 12. Partes que componen un mensaje

Formato Texto		
Cabecera	+CMGR: READ", "+59399775474", " ", "08/02/25,07:29:18-20",145,4,0,0," +59395897705",255,153	"REC Número de centro de servicio Fecha Remitente Hora
Cuerpo	O5. SI,5 OK	Mensaje en formato texto

Tabla 13. Formatos de Texto

A través de las siguientes funciones correspondientes a Cabecera y Cuerpo se puede cortar el mensaje para su uso:

```
/**
```

```
 * Función que devuelve el texto correspondiente a la cabecera del  
 mensaje. Ej: </br>
```

```
 * +CMGR: "REC READ","+59399775474"," ","08/02/25,07:29:18-  
 20",145,4,0,0,"+59395897705",255,153
```

```
 * @param strmensaje
```

```
 * @return Cadena de caracteres correspondiente a la cabecera del  
 mensaje.
```

```
 */
```

```
public String Cabecera(String strmensaje)
```

```
{
```

```
    String cabMensaje="";
```

```
    int posinicial = strmensaje.indexOf("+CMGR:");
```

```
    int posfinal = strmensaje.indexOf("\n",posinicial);
```

```
    if (posfinal<= strmensaje.length())
```

```
        cabMensaje = strmensaje.substring(posinicial,posfinal);
```

```
    return cabMensaje;
```

```
}
```

```
/**
```

```
 * Función que devuelve el texto correspondiente al cuerpo del  
 mensaje. PDU:
```

```
 * En caso de que el mensaje esté en formato PDU es transformado a  
 texto.
```

```
 * Por ejemplo si recibimos el siguiente mensaje:
```

```
 * +CMGR: "REC READ","+59392880591"," ","08/11/24,23:21:59-  
 20",145,100,0,0,"+59395897705",255,162
```

```
 *
```

```
05000346030186F5B09BFC068DC3EDB43BDC7ECF41F0B71CC40E8  
3EC697218E47ECF41E470FB3D078DEB65373D0C8AD7CBA0791AE4  
AEBBC761109BFD9687DBEF39C8FD06CDC3E272FB3D07B1DFA07  
8BD0C2ACF4174797A4E2FEBC32E10689E06B9EBEE7118340E97DB  
EF39481D6E87E7A0305C5E7693CBEDF71C1406B1CBF6B09B1E96B
```

BDF73501E344F83DC,5

* OK

* La función nos devolverá únicamente el cuerpo del mensaje transformado:

* Cuando caminamos por la vida nos damos cuenta que si nunca lloramos no sabemos lo que es tristeza. si nunca caemos jamas aprendemos a levantarnos y si n

* TEXTO:

* En caso de que el mensaje esté en formato texto, recorta solo la parte correspondiente al mensaje. Por ejemplo si recibimos:

*

* +CMGR: "REC READ", "+59399775474", " ", "08/02/25,07:29:18-20",145,4,0,0,"

* +59395897705",255,153

*

* (,)0 hey!

* ((.)) pis

* ('')(o)(') pis!

* ()'() * * *

* si, si tu..

* ..me estaba acordando d ti y pens en decirt.. Q tengas un lindo dia!

Tqm...,5

* OK

* La función nos devuelve:

*

* (,)0 hey!

* ((.)) pis

* ('')(o)(') pis!

* ()'() * * *

* si, si tu..

* ..me estaba acordando d ti y pens en decirt.. Q tengas un lindo dia!

Tqm..

*

```

* @param strmensaje
* @return Cadena de caracteres correspondiente al cuerpo del
mensaje.
*/
public String CuerpoMensaje(String strmensaje)
{
    String cuerpoMensaje="";
    int posinicial = strmensaje.indexOf("+CMGR:");
    int posfinal = strmensaje.indexOf("\n",posinicial);
    PDU formatPDU = new PDU();
    posinicial = posfinal;
    posfinal = strmensaje.lastIndexOf(",");
    if (posfinal<= strmensaje.length())
        cuerpoMensaje = strmensaje.substring(posinicial,posfinal);
    posinicial = cuerpoMensaje.indexOf("\n");
    cuerpoMensaje = cuerpoMensaje.substring(1);
    boolean bandera = cuerpoMensaje.substring(0,1).matches("0");

    if (bandera == true)
    {
        int          lensms          =
(cuerpoMensaje.substring(0,cuerpoMensaje.length())).length();
        cuerpoMensaje          =
formatPDU.PDU_inverso(cuerpoMensaje.substring(0,(cuerpoMensaje.lengt
h()), lensms).substring(7); //Función PDU_inverso transforma
    }
    cuerpoMensaje = cuerpoMensaje.replace("Ã±","ñ");
    cuerpoMensaje = cuerpoMensaje.replace("Ã©","é");
    cuerpoMensaje = cuerpoMensaje.replace("Â","");
    cuerpoMensaje = cuerpoMensaje.replace("Ã‰","É");
    cuerpoMensaje = cuerpoMensaje.replace("}", "ñ");
    return cuerpoMensaje;
}

```

Con ello podemos cortar la trama del mensaje recibido a través del celular correspondiente a la entrega de la mercancía por parte del conductor.

5. Mensaje estado de orden entrega - recepción

El formato en el que el conductor debe enviar el SMS es el siguiente:

O5.SI

En donde:

O: Hace referencia a la orden, carácter indicador.

5: Número de la orden

SI: Si la orden fue ejecutada con éxito, por defecto se sobreentiende que la orden no se ha realizado aún.

Una vez que el mensaje sea recibido a través del comando CMTI, se procederá a la actualización del estado de la orden en la base de datos.

A continuación se presenta la interfaz correspondiente a la aplicación de recepción de mensajes escritos a través del celular SAMSUNG SGH-E256, debemos tener en cuenta que el celular debe estar conectado al puerto de la máquina y prendido, además de contar con un SIM ya sea de cualquier operadora como MOVISTAR o PORTA.

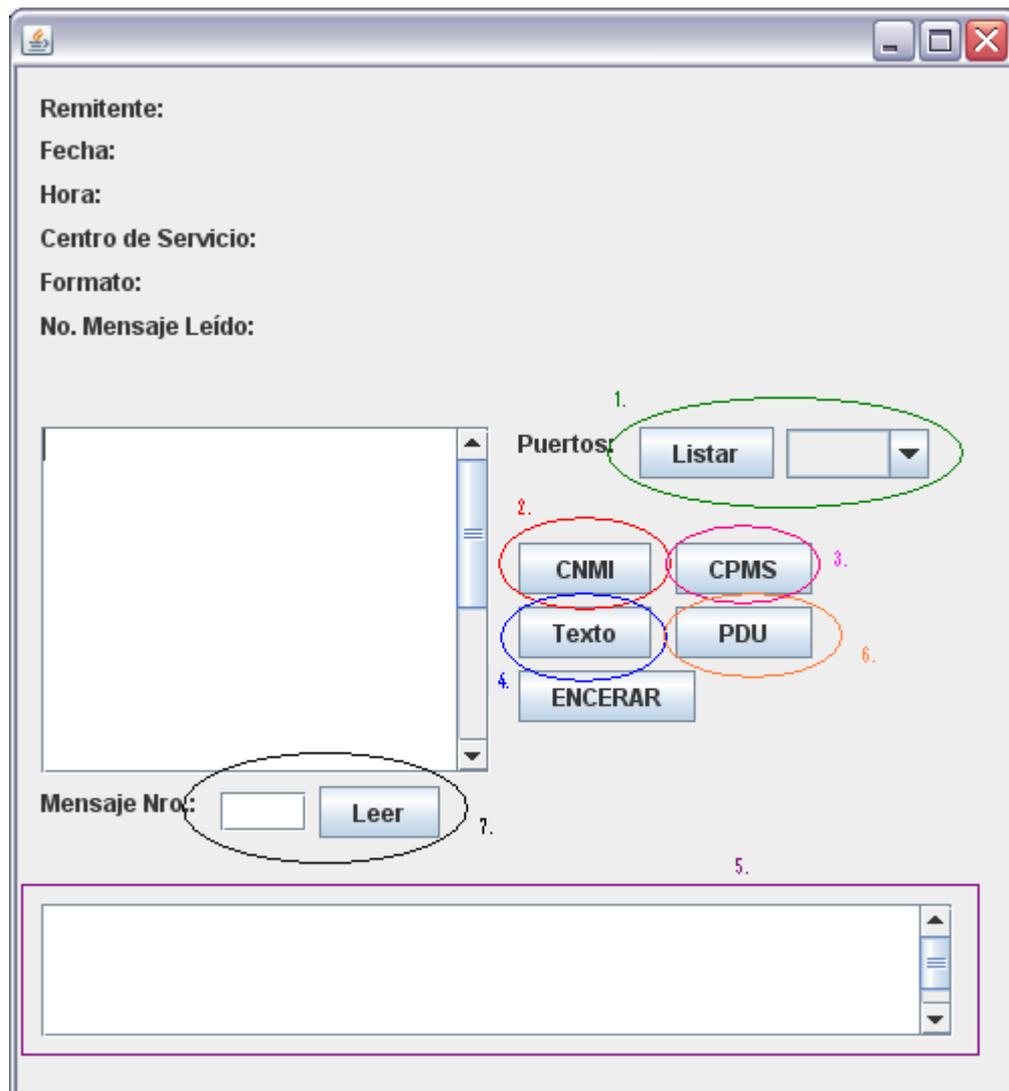


Gráfico 64. Interfaz de aplicación de recepción de mensajes escritos a través del celular SAMSUNG SGH-E256

1. **Listar:** Lista los puertos de la computadora.
2. **CNMI:** Configura el CNMI correspondiente al celular Samsung SGH-E256
3. **CPMS:** Configura el CPMS del celular
4. **Texto:** Setea la recepción de mensajes en formato texto
5. **Pantalla de estado de comandos**
6. **PDU:** Setea la recepción de mensajes en formato PDU
7. **Leer:** Lee un número de mensaje específico almacenado en memoria.

A continuación se describe el uso de la aplicación:

1. Para dejar en uso la aplicación debemos **Listar** los puertos haciendo clic en el botón listar: Se nos cargará los puertos que están activos:

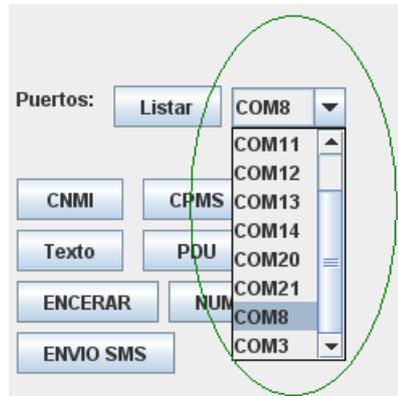


Gráfico 65. Listado de Puertos COM

Seleccionamos el que corresponda al celular Samsung SGH-E256 para lo cual nos vamos a Panel de control – Phone and Modem Options, se nos abrirá una ventana en donde debemos ir a la pestaña Modems:

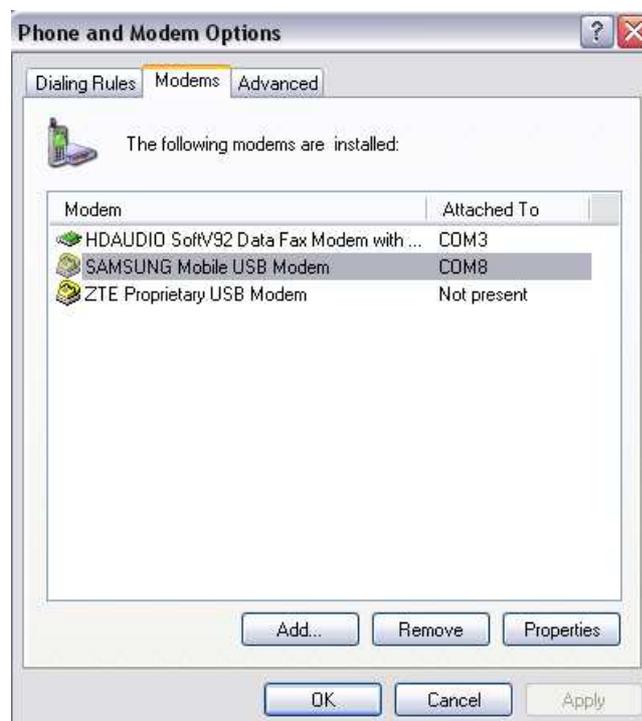


Gráfico 66. Opciones de Modem SAMSUNG SGH-E256

El COM8 va ha ser nuestro puerto.

- Después hacemos clic en **CNMI** y **CPMS**, para su configuración, en la pantalla de comandos de mensajes aparecerá el siguiente texto:



```
+CPMS: 35,35,35,35,35,35
OK
OK
```

- Hacemos clic en el tipo de recepción que queramos Texto o PDU, es preferente configurar en tipo texto, hacemos clic en el botón **Texto**. En la pantalla de comandos aparecerá OK.

El teléfono se encuentra listo para la recepción de mensajes.

- El conductor debe enviar el mensaje con el formato establecido, el celular receptorá el mensaje de la siguiente manera:
Por ejemplo si envía: O3. SI

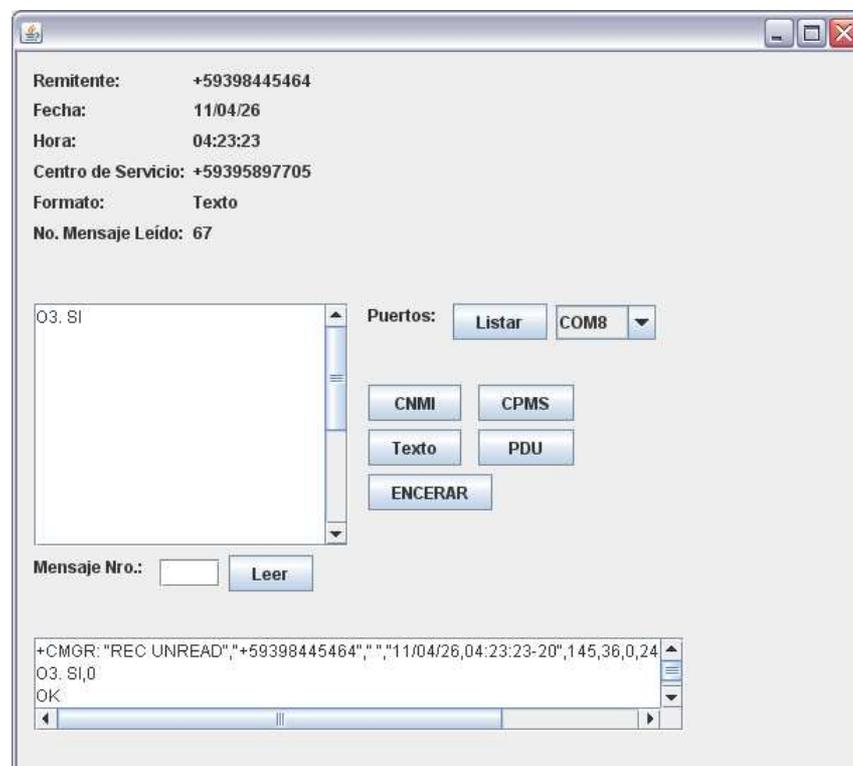


Gráfico 67. Formato de envío de mensajes

Nota: Si se desea leer un mensaje de memoria debemos ingresar el número de mensaje y hacer clic en Leer, por ejemplo:



Gráfico 68. Lectura de mensaje en memoria

4.6. ALGORITMO DE CÁLCULO DE RUTAS ÓPTIMAS

La ruta óptima se basa en un algoritmo creado en Prolog e incorporado a la aplicación del modelo de gestión; a través de la librería jpl.jar podemos utilizar dicho algoritmo.

Para la creación de la ruta óptima a seguir se debe tomar en consideración varios factores como son:

- La lista de órdenes que se debe entregar o recibir lo que se denomina puntos destino.
- El índice de peso de cada bloque creado a través de la siguiente fórmula:

$$w = \text{round}(\text{blk_d} * \text{blk_ind}) / 10$$

En donde:

w : Peso

blk_d : Es la longitud del bloque.

blk_ind : Es el índice referente al peso de cada bloque

- El orden en el que deben ir las órdenes de entrega - recepción.

Una vez establecido los requerimientos se realizó los siguientes pasos:

1. Creación de la base de conocimientos a utilizar:

La base de conocimientos refleja las intersecciones de cada calle, el peso que tiene cada bloque de calle, así como sus nodos, por lo que, se estableció que debe crearse la información según el siguiente formato:

$\text{rt}(\text{blk_id1}, \text{blk_id2})$

Ejemplo:

$\text{rt}(18099, 9001)$: Tiene la información referente a donde puede ir de un bloque a otro, esto expresa los sentidos de las calles, en el ejemplo se dice que el blk_id1 :18099 puede ir al blk_id2 : 9001.

$\text{bl}(\text{blk_id}, w, \text{blk_d}, [\text{latitud_poi_id0}, \text{longitud_poi_id0}, \text{latitud_poi_id0}, \text{longitud_poi_id0}])$.

Ejemplo:

$\text{bl}(109, 40.5, 101.23, [-2.891864, -79.003852, -2.892117, -79.002977])$: Tiene la información de cada bloque con el peso anterior pero adicionalmente esta la latitud y longitud del bloque inicial y bloque final al que se une.

En el Anexo 3 se tiene una carpeta en donde se ubica el archivo de la base de conocimientos creada.

2. Creación del algoritmo:

El algoritmo consiste en la inferencia de la ruta óptima dada una serie de destinos reordenando los mismos a través de permutaciones constantes creando así la ruta global que a través del indicador peso podemos conocer cuál es la ruta óptima.

A continuación se especifica la función principal a la que se debe llamar:

```
/* rroute: Peso, orden y rutas para los destinos
Parametros: S Lugar partida
            Ts Restos de destinos
            W Peso
            Tp Destinos permutados,
            DBr DB de rutas optimas */
rroute([S|Ts],W,[S|Tp],R):-consult('datos.txt'),
      findall(Z,rbetter([S|Ts],Z),DBr),
      permutation(Ts,Tp),
      rsumw([S|Tp],DBr,W,R).
```

En donde a través de:

- **consult('datos.txt')**: Realizamos la consulta del archivo que contiene la base de conocimientos.
- **findall(Z,rbetter([S|Ts],Z),DBr)**: Busca la ruta óptima del destino
- **permutation(Ts,Tp)**: Realiza la permutación de los destinos
- **rsumw([S|Tp],DBr,W,R)**: Va acumulando los pesos respectivos

La función `pbetter(Si,Sf,R)` nos da el mejor camino entre dos puntos, fundamental para conocer la ruta óptima de cada destino:

```

/*pbetter: Mejor camino entre dos puntos */
pbetter(Si,Sf,R):-findall(S,ppath(Si,Sf,S),Z),
                Z \== [],
                pshort(Z,999999,R,R).

```

La función rsumw obtiene la sumatoria de pesos y extrae las rutas de varios destinos:

```

/* rsumw: Sumatoria de pesos y extraccion de rutas de varios
destinos.
Parámetros:      Ti Destino Inicial
                  Tn Destino Siguiete
                  Ts Resto de Destinos
                  DBr DB de rutas
                  Sw Sumatoria de pesos
                  R Ruta
                  Rs Resto de Rutas */
rsumw([_],_,0,[]):-!.
rsumw([Ti,Tn|Ts],DBr,Sw,[R|Rs]):-rselect(Ti,Tn,DBr,W,R),
                                rsumw([Tn|Ts],DBr,Ws,Rs),
                                last(R,Tn),
                                Sw is W+Ws.

```

Con ello tenemos listo el algoritmo, para su funcionamiento debemos enviar de la siguiente manera los parámetros a la función:

```
rroute([Lista de órdenes],W,T,R)
```

Ejemplo:

rroute([34272,18171,13493,20524],W,T,R): La función rroute nos devolverá el peso de cada ruta, el orden de la lista de entregas – recepción y la ruta a seguir.

4.7. APLICACIÓN DE MONITOREO EN MAPSERVER

El servidor de mapas (MapServer) como se especificó anteriormente nos servirá de interfaz para la visualización de la información geográfica y el monitoreo de las rutas recorridas por el conductor de la empresa. Para el desarrollo de esta aplicación se debe realizar los siguientes procesos:

- 1. Elaboración del archivo .map:** Contiene la información referente a las capas temáticas representadas en el árbol de contenidos de MapServer, en este archivo se determina la fuente de información de la que se leerán los datos, sus estilos y filtros determinados como son la escala ya sea de la capa o de las etiquetas de las capas correspondientes. A continuación se presenta un listado de las capas que se mostrarán:

Capa base: Parroquias urbanas cuenca

Capas sobrepuestas:

- Clientes
- Manzanas
- Áreas Verdes
- Vialidad
- Clientes
- Proveedores
- Bodegas
- Ruta rastreo vehículo
- Rutas óptimas

Dentro del archivo .map podemos ver la configuración de cada capa, en el Anexo 4 se puede ver la estructura del .map.

- 2. Definición de la configuración inicial página del servidor de mapas:**

Para configurar inicialmente el servidor de mapas debemos definir los siguientes parámetros:

```

var createMap = function() {
    map = new OpenLayers.Map('mymap', {
        projection: "EPSG:4326",
        controls: [],
        maxExtent: new OpenLayers.Bounds(-79.0369737423846, -
2.91365329121574, -78.9791473467576, -2.87535720468879),
        units: "dd",
        numZoomLevels:20,
        maxScale: 10,
        minScale: 50000
    });
};

```

En donde:

Projection: Define el sistema en el que se encuentran los mapas, en nuestro caso es el GCS_WGS_1984, sistema geográfico de referencia WGS84, determinado por el código EPSG: 4326. Este sistema nos establece las coordenadas geográficas del mapa en grados sexagesimales, con ello se tiene la posición exacta del punto.

MaxExtent: Define los bordes del mapa, se lo realiza en base a la información que se va a cargar, de la siguiente manera:

```

maxExtent: new OpenLayers.Bounds(BBoxMinX, BBoxMinY, BBoxMaxX,.
                                BBoxMaxY)

```

Units: Las unidades del mapa, pueden ser metros (m) o grados sexagesimales (dd), se utilizará el segundo.

numZoomLevels: Número máximo de niveles de acercamiento.

maxScale: Escala máxima.

minScale: Escala mínima

3. Creación de la interfaz

Para la creación de la interfaz se utilizó openlayers y el framework MapFish, el mismo tiene las aplicaciones del servidor de mapas. Se debe estructurar la información en nodos para activar o desactivar las capas y establecer las herramientas a utilizarse, así se definió las siguientes herramientas a utilizar:

 Zoom a la máxima extensión del mapa

 Zoom in

 Alejamiento

 Paneo

 Información

 Dibujar un punto en el mapa

 Dibujar una línea en el mapa

 Dibujar un polígono en el mapa

 Vista anterior

 Vista siguiente

A continuación se presenta la pantalla de presentación de MapServer:

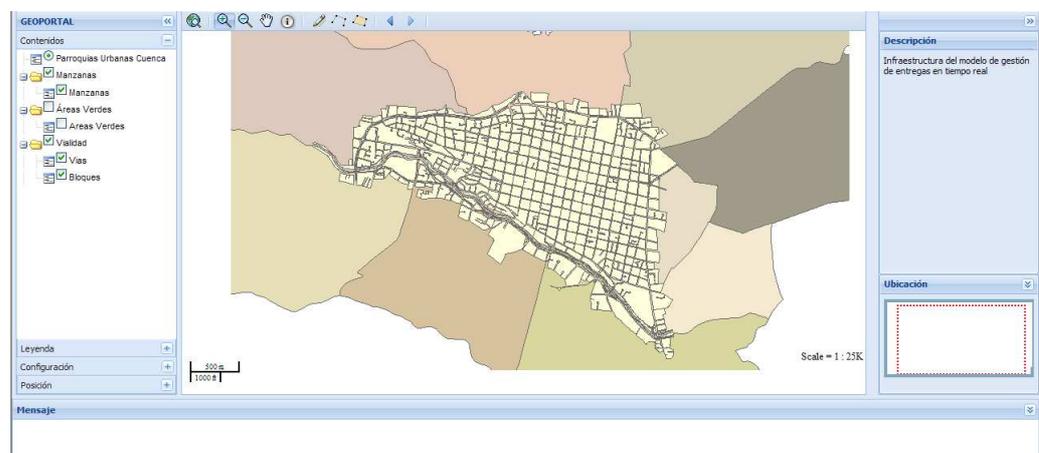


Gráfico 69. Visualización de Mapserver

4. Búsqueda

La aplicación permita la búsqueda de la información referente a las capas cargadas para su visualización.

5. Monitoreo

Con los componentes previamente mencionados se puede realizar el monitoreo de la ruta que actualmente recorre el conductor así como la visualización del resto de capas.

4.8. MODELO DE GESTIÓN DE ENTREGAS POR RUTAS ÓPTIMAS

La creación de la aplicación que se utilizará de interfaz con el usuario se realizó en base al modelo de datos establecido en el Capítulo III. Se toma en consideración que el sistema al que se le vincula tiene todo el módulo de facturación, inventarios, contabilidad, etc. Por lo que en esta aplicación solo se toman las tablas necesarias para la gestión de entregas simulando la información necesaria así como las órdenes de entrega y/o recepción de la mercadería. La aplicación es multiempresa y ya que se encuentra en la web puede ser utilizada desde cualquier lugar.

La aplicación fue creada en Netbeans IDE 6.1, utiliza las librerías de java y adicionalmente la librería .jpl de Prolog. Las clases contenidas en jpl.jar suministran los métodos necesarios para comunicarse con Prolog. En nuestro caso tenemos un programa en Prolog, para invocarlo desde un entrono visual en Java.

Seguridad: El sistema maneja dos tipos de seguridades:

- **Administrador:** Es aquella persona que tendrá acceso total al sistema, se encarga de la administración de la aplicación, así como las configuraciones necesarias de los parámetros del sistema
- **Usuario:** Es el empleado que realizará las transacciones de la aplicación.

A continuación se describen los pasos para la creación de la aplicación:

1. Creación de la interfaz de usuario:

En base a las especificaciones de diseño del Capítulo III se definió la siguiente interfaz, que cuenta de un menú en la parte superior en base al cual el usuario y/o administrador puede navegar. Dependiendo de los privilegios según su rol se tiene acceso a las opciones.

2. Definición de los botones de interacción con el usuario:

Los botones de interacción con el usuario permiten la utilización o mantenimiento de la información, las funciones básicas de cada uno están descritas en el Anexo 5 - Manual de Usuario.



3. Creación de los formularios

Se creó los formularios para el mantenimiento de cada tabla. Se restringe el acceso a los formularios dependiendo del nivel de seguridad que tiene el usuario. Así aquellos que se encuentran en la parte de seguridad solo podrán ser definidos por el administrador.

4. Geolocalización

Para el caso de los clientes, proveedores, bodegas y usuarios se tiene la ubicación geográfica de los mismos en base a las coordenadas que se generan automáticamente a través de la base de datos en Postgres, se utilizó un API de Google para ubicar a cada uno de los mencionados puntos. El API de Google es gratuito y puede ser descargado de la página del mismo, pedirá un registro, en tal caso debemos crear una cuenta en Google, solicitar el key y a su vez reemplazar la siguiente línea en la página .html provista por la misma como se ve a continuación:

```
<script  
src="http://maps.google.com/maps?file=api&v=2&key=AB  
QIAAANLiseyKWZaqTt2RdqZQqghTwM0brOpm-  
AII5BF6PoaKBxRWWERSS5HaZxl6YaQuss5r4VGnpHkVMEw"  
type="text/javascript"></script>
```

Para el uso de la página se pasó los parámetros adecuados de cliente, proveedor, bodega o usuario, dirección, latitud y longitud a la página .html.

En la siguiente ventana se puede ver un ejemplo de la ubicación:

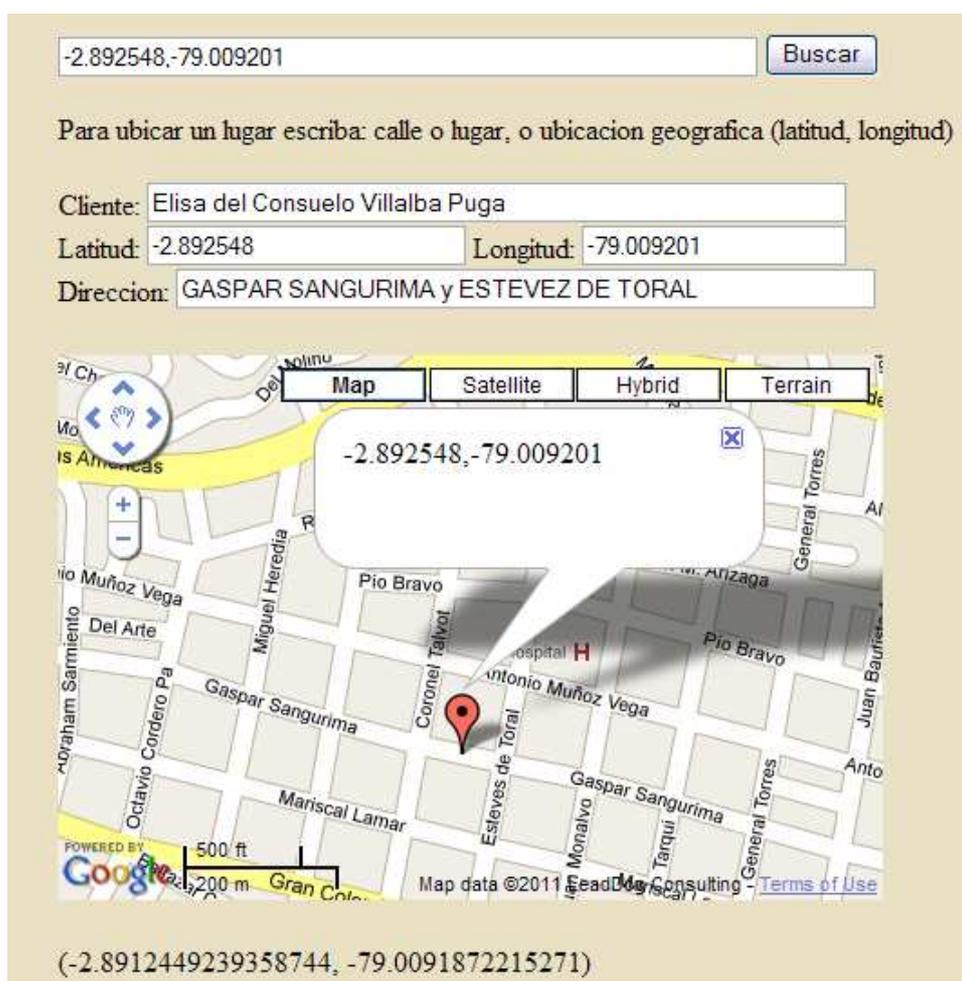


Gráfico 70. Ejemplo de localización

5. Integración de Prolog con java

Como se dijo anteriormente se debe incluir la librería jpl.jar a la aplicación en java. Para utilizar el algoritmo creamos una clase java e importamos las siguientes librerías del jpl:

```
import java.util.Hashtable;
import jpl.*;
import jpl.Query;

public class Op{

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {

        String t1 = "consult('op.pl)";
        Query q3 = new Query(t1);
        System.out.println( t1 + " " + (q3.hasSolution() ? "succeeded" :
"failed") );
        //-----
        String t4 = "rroute([2,7,25,14,24],W,T,R)";
        Query q4 = new Query(t4);
        //-----
        java.util.Hashtable[] ss4 = q4.allSolutions();
        System.out.println( "all solutions of " + t4);
    }
}
```

En main se puede apreciar cómo se llama a Prolog, debemos crear una consulta del .pl, a continuación procedemos a llamar a la función en Prolog correspondiente a las rutas óptimas, por ejemplo se mando como parámetros

`rroute([2,7,25,14,24],W,T,R)`, nos devolverá un query, el mismo puede ser recorrido a través del método `allSolutions()`.

Con ello tenemos la unión con Prolog, a continuación definimos el desgloce de la información adquirida y la grabamos en la base de datos.

6. Ordenes de Entrega-Recepción

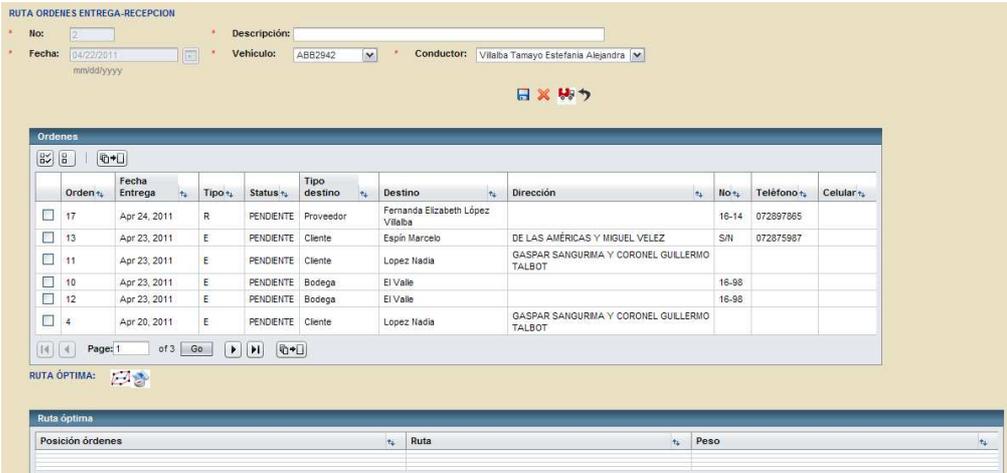
Se realizó dos formularios en donde permiten el ingreso de ordenes de entrega-recepción respectivamente, ahí se definen los artículos que tiene que entregar, la fecha y a quien va dirigida la entrega y lugar de la entrega.

7. Gestión de entregas

La gestión de entregas se define cuando asignamos a un conductor, un vehículo que tiene conectado su GPS respectivo y las ordenes de entrega – recepción a ejecutar en la fecha asignada. Con el botón de agregar ordenes  vamos eligiendo las ordenes que están ‘PENDIENTES’.

Debemos vincular al botón de ruta óptima a la función de cálculo de ruta definida en el punto 5, así dicho botón  nos dará las rutas calculadas ordenadas según su optimización en la tabla de rutas óptimas.

En el siguiente formulario se ve la presentación de la gestión de entregas:



Orden	Fecha Entrega	Tipo	Status	Tipo destino	Destino	Dirección	No.	Teléfono	Celular
<input type="checkbox"/>	17	Apr 24, 2011	R	PENDIENTE	Proveedor	Fernanda Elizabeth López Villaba	16-14	072897865	
<input type="checkbox"/>	13	Apr 23, 2011	E	PENDIENTE	Cliente	Espín Marcelo	S/N	072875987	
<input type="checkbox"/>	11	Apr 23, 2011	E	PENDIENTE	Cliente	Lopez Nadia			
<input type="checkbox"/>	10	Apr 23, 2011	E	PENDIENTE	Bodega	El Valle		16-98	
<input type="checkbox"/>	12	Apr 23, 2011	E	PENDIENTE	Bodega	El Valle		16-98	
<input type="checkbox"/>	4	Apr 20, 2011	E	PENDIENTE	Cliente	Lopez Nadia			

Gráfico 71. Pantalla principal de Gestión de Entregas

Podemos visualizar las rutas geográficamente presionando el botón de localización , se nos desplegará las 3 primeras rutas óptimas en MapServer.

8. Rastreo de vehículos

El rastreo de los vehículos como se planteó anteriormente se da en base al GPS, la información almacenada es visualizada en MapServer, a su vez se incorporó al programa el reporte de la ruta seguida por el conductor con los filtros correspondientes a fecha inicial, fecha final, conductor, vehículo u orden entregada.

9. Actualización estado de la orden

Una vez entregada la orden el conductor deberá enviar un SMS al celular asignado por el administrador con la orden en el formato establecido en la aplicación del celular, automáticamente se actualizará el estado de la orden.

10. Definición de Consulta

Se realizó para cada tabla la consulta de sus datos respectivos en el mismo formulario de mantenimiento, para el caso de las órdenes de entrega-recepción y la gestión de entrega-recepción puede ser visualizada a través del botón de consulta.

11. Ayuda

Se realizó un manual de usuario (véase Anexo 5 - Manual de Usuario) para que el usuario pueda interactuar con la aplicación.

Con ello hemos finalizado la unión de todas las aplicaciones realizando de esta manera una gestión de entregas por rutas óptimas, mediante GPS y conexión celular.

CONCLUSIONES

- La tecnología disponible y el desarrollo nos permite crear una logística para una gestión de entregas óptima que ayude al soporte de toma de decisiones, procesos de análisis y gestión; obteniendo la mejor relación calidad, precio y conservando los plazos óptimos de entrega.
- Los problemas de rutas de vehículos son cada vez más grandes debido a la demanda de servicio que los clientes requieren, creando de esta manera, restricciones operativas que se originan en la posición geográfica de la entrega. Por ello, debemos instruirnos en integrar la tecnología, los sistemas clientes, los recursos y de esta manera tomar mejores decisiones.
- La integración de componentes que permiten el rastreo y localización de vehículos a través de los servicios de GPRS, GSM y GPS, herramientas de inteligencia artificial y desarrollo y monitoreo geográfico de las rutas nos brindan un soporte para crear un modelo de gestión de entregas a través de rutas óptimas con ahorros de tiempo y recursos.
- El modelo de gestión de entregas cuenta con un sistema de monitoreo que representa la información del rastreo vehicular permitiendo el control de los empleados, logística y toma de decisiones que generan una mejora con respecto a la calidad, tiempo y costos.
- El rastreo vehicular nos permite una administración adecuada de los recursos de la empresa, ya sea, del transporte o mercadería. Al integrar dicho rastreo con la inferencia de la ruta óptima se logró mejorar la respuesta a problemas que día a día se van presentando.
- La aplicación celular brinda un soporte para conocer el estado de las órdenes de entrega en tiempo real, ofreciendo un servicio adicional al cliente, que además de un rastreo y localización continua, garantiza un excelente servicio.
- La planificación de rutas a través de la inferencia de caminos óptimos, es un gran beneficio para la empresa, pues reduce el tiempo de entrega-recepción de

órdenes, reduce el consumo de gasolina y la mercadería va monitoreada constantemente conociendo su estado actual, logrando así acelerar los procesos de la empresa optimizando los recursos de la misma.

RECOMENDACIONES

- Se debe tener en cuenta la restricción de la zona en la que va a transitar el vehículo y las capas base de las que se extraerá la información; pues, la generación de nodos y bloques de calles demanda precisión. Por ello, se recomienda la depuración de las capas y generación de información en un sistema de información geográfica.
- Para realizar un rastreo vehicular apropiado es necesario contar con dos SIM's de empresas proveedoras de servicios de comunicación como son MOVISTAR o CLARO, dicho servicio se debe contratar con anticipación y nos ayudará a evitar cortes de servicio de cualquiera de ellas.
- Con respecto a la obtención del reporte de la localización del vehículo se recomienda utilizar el servicio GPRS que vaya directo al servidor, nos ahorrará muchos procesos que con el servicio SMS.
- La base de conocimientos debe estar bien estructurada y depurada, de lo contrario causará retardos innecesarios en la inferencia de rutas óptimas.
- Las direcciones de los clientes deben ser correctas y estar verificadas, de lo contrario no se creará la información necesaria para una entrega óptima.
- Se recomienda configurar al celular Samsung SGH-E256 en modo texto para el estado de orden de entrega-recepción pues la información se procesa directamente. Así, a través del uso del comando CMTI se puede adquirir continuamente los mensajes escritos.
- Los nodos, bloques y puntos de la calles de Cuenca representan una red estructurada en base a parámetros que disminuyen el movimiento vehicular y que facilitarán a la creación de la base de conocimiento para el cálculo de las rutas, por lo que se debe tener una información depurada y alimentada constantemente de calles cerradas, cambio en los sentidos, entre otros.

GLOSARIO

- **AVL** por sus siglas en inglés significa “Automatic Vehicle Location” “Localización automática de Vehículos”. (CosmoTRACS, 2009)
- **Cartografía:** El conjunto de técnicas para la modelización de la superficie terrestre por medio de datos georreferenciados susceptibles de representación gráfica. (BOJA, 2006)
- **Cartografía básica:** La obtenida por procesos directos de observación y medición de superficie terrestre, sirviendo de base y referencia para su uso generalizado como representación gráfica del territorio. (BOJA, 2006)
- **Cartografía derivada:** La realizada a partir de información contenida en cartografía básica mediante procesos de generalización, reducción, selección o simplificación. (BOJA, 2006)
- **Cartografía temática:** La realizada tomando como referencia geométrica otra cartografía básica o derivada, con un contenido relativo a la distribución geográfica de un fenómeno específico. (BOJA, 2006)
- **Conversión de coordenadas:** Cambio de coordenadas basado en una relación uno a uno desde un sistema de coordenadas a otro basado en el mismo datum. Por ejemplo: Entre sistemas de coordenadas geodésicas y cartesianas, o entre coordenadas geodésicas y coordenadas proyectadas, o cambios de unidades tales como de radianes a grados o de pies a metros. (Ministerio del Fomento, 2007)
- **Coordenada:** Cualquiera de los n números de una secuencia que designa la posición de un punto en un sistema n dimensional. Es importante conocer que en un sistema de referencia de coordenadas, los números deben ser dados con unidades. (Ministerio del Fomento, 2007)

- **Datos espaciales:** La información geográfica con una referencia directa o indirecta a una localización por coordenadas o área geográfica. (BOJA, 2006)
- **Datum:** Parámetro o conjunto de parámetros que sirven como referencia o base para el cálculo de otros parámetros. Un Datum define la posición origen, la escala y la orientación de los ejes del sistema de coordenadas. (Ministerio del Fomento, 2007)
- **Datum geodésico:** Datum que describe la relación de un sistema de coordenadas con la Tierra. En la mayoría de los casos, el datum geodésico incluye una definición de elipsoide. (Ministerio del Fomento, 2007)

El datum que se utilizó en Ecuador hasta hace poco fue el “Provisional South American 1956” (PSAD56 también conocido como SAM56), tiene como punto de origen a un sitio denominado “La Canoa” (Venezuela). (Ochoa)

El Datum más utilizado actualmente, impulsado sobre todo por el uso de los GPS, es el World Geodetic System de 1984 (WGS84), y tiene como punto de origen al centro de gravedad de la tierra. (Ochoa)

- **Elipsoide:** Superficie engendrada por la rotación de una elipse alrededor de un eje principal. En la norma internacional ISO 19111 los elipsoides son siempre achatados en el polo esto significa que el eje de rotación es siempre el eje menor. (Ministerio del Fomento, 2007)
- **Geomática:** La técnica para la obtención, análisis, distribución y uso de la información geográfica. (BOJA, 2006)
- **JPL:** Para la comunicación bidireccional entre Java y SWI-Prolog. (Morales, 2007)
- **Latitud geodésica, latitud elipsoidal:** Ángulo que forma el plano ecuatorial con la perpendicular al elipsoide desde un punto dado, se toma positiva hacia el norte. (Ministerio del Fomento, 2007)

- **Longitud geodésica; longitud elipsóidica:** Ángulo que forma el plano meridiano principal con el plano meridiano de un punto dado, se toma positiva hacia el este. (Ministerio del Fomento, 2007)
- **Meridiano:** Intersección de un elipsoide por un plano que contiene el semieje menor del elipsoide. (Ministerio del Fomento, 2007)
- **Metadatos:** Los descriptores de las características técnicas de los datos espaciales. (BOJA, 2006)
- **Posicionamiento:** Las técnicas para la determinación de la localización de un punto sobre la superficie terrestre. (BOJA, 2006)
- **Red geodésica:** El sistema de puntos cuya localización en latitud, longitud y altura son conocidos de forma precisa. (BOJA, 2006)
- **Sistema de coordenadas:** Se define por un nombre, las unidades, la dirección y la secuencia de los ejes. Los sistemas de coordenadas más frecuentes son:
 - a. **Sistema de coordenadas cartesianas tridimensionales:** x,y,z.
 - b. **Sistema de coordenadas geodésicas:** coinciden con las coordenadas elipsóidicas: latitud, longitud y altitud.
 - c. **Sistema de coordenadas sobre una proyección cartográfica.** La definición de cada proyección de la superficie de referencia (elipsoide o esfera) sobre el plano lleva asociada la de las funciones que relacionan la posición de un punto sobre ambas superficies. Esas funciones son tipos de operaciones de conversión.

En la norma ISO 19111, publicada en 2003, se define Sistema de Referencia de Coordenadas (CRS) como un sistema de coordenadas que está referido a la Tierra a través de un datum geodésico. (Fco. Javier Gonzales Matesanz, 2008)

Sistema de coordenadas planas UTM (Universal Transverse de Mercator):

Los sistemas de coordenadas Planas o Proyectadas, se obtienen a partir de la proyección cartográfica de la esfera o el esferoide sobre una superficie plana

(bidimensional). Las coordenadas UTM se expresan en metros y tienen como ejes de referencia la línea del ecuador y la de una meridiana central. (Ochoa)

Sistemas de Coordenadas Esféricas: Conocido también como Sistema de Coordenadas Geográficas (SCG). El sistema considera la referencia de un punto tanto con respecto a la Latitud como con la Longitud, cada una de estas referencias se expresa en grados minutos y segundos. (Ochoa)

- **Sistema de información geográfica:** El conjunto de informaciones georreferenciadas, recursos informáticos, humanos y organizativos que realizan procesos de captación, almacenamiento, análisis y representación de los datos espaciales. (Ministerio del Fomento, 2007)

- **SWI-Prolog:** Ambiente de desarrollo y compilador de Prolog bajo el esquema de licenciamiento del software libre. (Morales, 2007)

- **Transformación de coordenadas:** Cambio de coordenadas desde un sistema de referencia de coordenadas a otro sistema de referencia de coordenadas basado en un datum diferente a través de una relación uno a uno. (Ministerio del Fomento, 2007)

BIBLIOGRAFÍA

(s.f.). Recuperado el 15 de Marzo de 2011, de

<http://www.cs.uns.edu.ar/~grs/InteligenciaArtificial/InteracJavaProlog.pdf>

(s.f.). Obtenido de http://www.swi-prolog.org/packages/jpl/java_api/index.html

(s.f.). Obtenido de http://www.jaist.ac.jp/~fuji/work/docs/html4/jpro_p1_c2.html

(s.f.). Obtenido de

<https://sauron.etsi.urv.es/public/PROPOSTES/pub/pdf/541pub.pdf>

(s.f.). Obtenido de

<http://cybertesis.uach.cl/tesis/uach/2005/bmficis211a/doc/bmficis211a.pdf>

(s.f.). Obtenido de

http://books.google.com.ec/books?hl=es&lr=lang_es&id=YdYXyJ6lLcMC&oi=fnd&pg=PP7&dq=gestion+de+entregas+vehicular&ots=0NiaK59K9B&sig=8S0-OnV87UPpIYXqhyj-kF2nL1U#v=onepage&q&f=false

(s.f.). Recuperado el 21 de Julio de 2010, de

http://books.google.com.ec/books?hl=es&lr=lang_es&id=BCJm9jjgHr4C&oi=fnd&pg=PA77&dq=ruta+optima+prolog&ots=3TPbL_1XVJ&sig=DVTNlA_YiL_4PBtiE9aGyVgFUO8#v=onepage&q&f=false

(s.f.). Obtenido de <http://www.telefonos-moviles.com/gprs/default.asp>

(s.f.). Obtenido de

http://www.nokia.com/EUROPE_NOKIA_COM_3/r2/support/tutorials/6610i/spanish/index_glossary.html

(s.f.). Obtenido de http://www.solomanuales.org/manual_introduccion_al_gprs-manuall31053166.htm

(s.f.). Obtenido de

http://www.pdaexpertos.com/Tutoriales/Internet_movil/Guia_completa_de_GPRS.shtml

(s.f.). Obtenido de

<https://forja.rediris.es/docman/view.php/720/1135/Tutorial%20GPRS.pdf>

(s.f.). Obtenido de <http://hist.library.paho.org/Spanish/EMS/6567.pdf>

(Noviembre de 2003). Recuperado el 21 de Julio de 2010, de

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.101.6648&rep=rep1&type=pdf#page=6>

(Febrero de 2007). Recuperado el 27 de Julio de 2010, de

http://io.us.es/cursos/doctorado/Trabajos_pdf/Jorge_Aguilera_VRPTW.pdf

(17 de Mayo de 2008). Recuperado el 15 de Marzo de 2011, de

<http://hulles.supersized.org/uploads/geewhiz/geewhiz.pdf>

(Febrero de 2008). Recuperado el 21 de Julio de 2010, de

<http://www ldc.usb.ve/~borges/ci3661/prolog/proyecto.pdf>

(2010). Recuperado el 2010, de <http://www.buenastareas.com/ensayos/Bases-De-Datos-De-Conocimiento/1238571.html>

Aruquipa, C. M., Hurtado, M. R., Márquez, G. E., & Saravia, A. R. (13 de Junio de 2007). Recuperado el 15 de Marzo de 2011, de

http://www.postgradoinformatica.edu.bo/enlaces/investigacion/pdf/INGSW3_112.pdf?PHPSESSID=b84491a8a35f8fefa41f066891a7955c

Biblioteca de Ingeniería Eléctrica y Electrónica. (2010). Recuperado el 2010, de

<http://bieec.epn.edu.ec:8180/dspace/bitstream/123456789/1096/7/T10903CAP2.pdf>

BOJA. (9 de Agosto de 2006). Consejería de Obras Públicas y Transporte. Sevilla.

Brazuelo, F. G., & Gallego, D. J. (Enero de 2009). Recuperado el 21 de Julio de

2010, de <http://www.sav.us.es/pixelbit/pixelbit/articulos/n34/4.pdf>

- Calvo, X. B. (21 de Enero de 2005). Recuperado el 21 de Julio de 2010, de <http://upcommons.upc.edu/pfc/bitstream/2099.1/3714/2/35776-1.pdf>
- Chávez, M. B., & Olmedo, D. M. (08 de Agosto de 2006). Recuperado el 01 de Marzo de 2011, de <http://www3.espe.edu.ec:8700/bitstream/21000/402/1/T-ESPE-014994.pdf>
- Collarte, F. B., & Barco, D. M. (2006). Recuperado el 21 de Julio de 2010, de http://cybertesis.upc.edu.pe/upc/2006/collarte_bf/sources/collarte_bf.doc
- CosmoTRACS. (2009). *SlideShare cosmoTRACS Tutorial GPS de Rastreo y Localizacion Satelital en México*. Recuperado el 21 de Julio de 2010, de <http://www.slideshare.net/pasvic70/cosmotracs-tutorial-gps-de-rastreo-y-localizacion-satelital-en-mxico>
- De Los Mozos, J. L., & Moreno, S. L. (s.f.). Recuperado el 21 de Julio de 2010, de <http://ubr.universia.net/pdfs/UBR0042007118.pdf>
- Del Río, M. S. (10 de Marzo de 2006). Recuperado el 21 de Julio de 2010, de https://coit.es/pub/ficheros/p030_resumen_infoglobal_b664f34b.pdf
- Editado por Lockhart, T. (s.f.). Recuperado el 5 de Octubre de 2010, de <http://fcp.unach.mx/manuales/download/Postgres-User.pdf>
- Fco. Javier Gonzales Matesanz, D. H. (2008). *Sigte Servei de Sistemes D'Informació Geografica I Teledetecció Universitat de Girona*. Girona.
- Freitas de Souza, C. O. (s.f.). Recuperado el 11 de Mayo de 2009, de <http://www.midiacom.uff.br/~debora/fsmm/trab-2003-2/msgMM.pdf>
- González, J. G. (30 de Noviembre de 2005). *Foro Académico de la DES de Ingenierías y Arquitectura de la Universidad Michoacana de San Nicolás de Hidalgo*. Recuperado el 11 de Mayo de 2009, de http://www.des_ia.umich.mx/~des_ia/fades05/memorias/FE-028.pdf
- GTE. (2009). *Manual Trax S6*. Buenos Aires, Argentina.

Gutiérrez, Valentina; Palacio, Juan David; Villegas, Juan Guillermo. (2007).

Obtenido de <http://redalyc.uaemex.mx/redalyc/pdf/215/21514507.pdf>

Hernández, S. R., & Carvajal, J. E. (Julio de 2009). Recuperado el 21 de Julio de

2010, de <http://bibdigital.epn.edu.ec/bitstream/15000/1601/1/CD-2324.pdf>

Instituto Geográfico Nacional, Ministerio de Fomento. (2007). Curso de infraestructuras espaciales. España.

Jorrín, S. A. (26 de Mayo de 2010). Recuperado el 17 de Marzo de 2011, de

<http://www1.geograma.com/pdf/GS-ManualUsuario.pdf>

Jpcap Java library for capturing and sending network packets. (s.f.). Recuperado el

2009, de <http://netresearch.ics.uci.edu/kfujii/Jpcap/doc/download.html>

Madeco Materiales de Construcción. (s.f.). Recuperado el 17 de Marzo de 2010, de

<http://www.madeco.com.ec/tienda/?q=node/542>

Maquera, N. G., Abensur, D., & Parracho, A. (s.f.). Recuperado el 11 de Agosto de

2010, de <http://www.cnc-logistica.org/congreso-cnc/documentos/168.pdf>

Ministerio del Fomento, D. G. (5 de Octubre de 2007). Términos y definiciones de la

ISO 19111 Versión 1.0. España.

Morales, I. V. (Julio de 2007). *Instituto Politécnico Nacional Plantilla para la*

representación de Conocimiento Geográfico en el Dominio Espacial. Recuperado el

2010, de <http://148.204.64.119/posgrados/images/sources/cic/tesis/B041258.pdf>

Moya de la Torre, E. J., Pérez, J. M., Janeiro, J. R., García, D. G., & Calvo, O. M.

(s.f.). Recuperado el 11 de Mayo de 2009, de <http://www.cea->

[ifac.es/actividades/jornadas/XXVIII/documentos/1301-](http://www.cea-ifac.es/actividades/jornadas/XXVIII/documentos/1301-)

[Plataforma_TeleControlRemoto.pdf](http://www.cea-ifac.es/actividades/jornadas/XXVIII/documentos/1301-Plataforma_TeleControlRemoto.pdf)

Muñoz, M. (s.f.). Recuperado el 19 de Agosto de 2010, de

<http://dspace.ups.edu.ec/bitstream/123456789/56/10/Capitulo4.pdf>

Murillo, J. D. (Julio de 2005). *Movicuo comunicaciones moviles de última*

generación para la ubicuidad. Recuperado el 05 de 12 de 2010, de

http://patanegra.unex.es/agila/movicuo/archivos/Documentacion_MOVICUO.pdf

Murrugarra, N. L. (s.f.). Recuperado el 15 de Marzo de 2011, de <http://www.seccperu.org/files/ConexJava-Prolog.pdf>

Neira, Á. A. (s.f.). Recuperado el 21 de Julio de 2010, de http://www.item.ntnu.no/fag/tm8100/Pensumstoff2004/GPRS_Tutorial.pdf

Ochoa, I. P. (s.f.). Tutorial ArcGIS.

Ortega, G. M., & García, C. M. (2006). *Estudio de algoritmos de planificación en I.A.* Recuperado el 21 de Julio de 2010, de <http://www.imai-software.com/openlab/data/proyectos/Shop/EstudioTecnico.pdf>

Pereira, W. (s.f.). Recuperado el 23 de Julio de 2010, de <http://cs.mty.itesm.mx/cursos/cita2006/ponencias/adicionales/Arquitectura%20inalambrica%20de%20hardware%20y%20software%20para%20control%20de%20identidad,%20localizacion%20y%20telemetria%20de%20vehiculos.pdf>

Pulgarían, A. J. (2001). Recuperado el 21 de Julio de 2010, de http://pisis.unalmed.edu.co/vieja/cursos/analisis_decisiones/GA/Capitulos.PDF

Roa, G. I., & Merlano, J. C. (s.f.). Recuperado el 21 de Julio de 2010, de <http://reocities.com/Paris/2914/PrimerInformeParcial.pdf>

Rodríguez, A. V. (s.f.). Recuperado el 21 de Julio de 2010, de <http://www.cnc-logistica.org/congreso-cnc/documentos/136.pdf>

Rodríguez, A. V. (Septiembre de 2006). Recuperado el 21 de Julio de 2010, de http://www.adingor.es/Documentacion/CIO/cio2006/docs/000004_final.pdf

Rodríguez, A. V. (2007). Integración de un sig con modelos de cálculo y optimización de rutas de vehículos cvrp y software de gestión de flotas. *XI Congreso de Ingeniería de Organización*, (pág. 8). Madrid.

Sánchez Silva, Á. (11 de 6 de 2006). *Tecana American University*. Recuperado el 23 de Mayo de 2010

Santamaría, F. A. (s.f.). Recuperado el 23 de Julio de 2010, de http://csicolombia.weebly.com/uploads/1/8/3/0/1830887/aprendizaje_con_apoyo_de_tecnologas_mvil_grid.pdf

Seco, A. H. (29 de Octubre de 2001). Recuperado el 11 de Mayo de 2009, de <http://ftp.softnet.tuc.gr/pub/linux/docs/LuCaS/Presentaciones/200103hispalinux/seco/pdf/pasarelas-sms.pdf>

TomTom, sistemas autonavegadores con GPS. (2010). Recuperado el 27 de Julio de 2010, de <http://www.tomtom.com/howdoesitwork/category.php?ID=2&Language=6>
Torres, F. (s.f.). Recuperado el 21 de Julio de 2010, de <http://upcommons.upc.edu/pfc/bitstream/2099.1/3714/2/35776-1.pdf>

Tuckler, E. (s.f.). *Temas generales de Prolog.* Recuperado el Junio de 2010
Vilalta, J. (2002). Recuperado el 01 de Marzo de 2011, de <http://www.vico.org/MuestrarioDiagCU.pdf>

Wikipedia Enciclopedia Libre. (2010). Recuperado el 2010, de http://es.wikipedia.org/wiki/Web_Map_Service