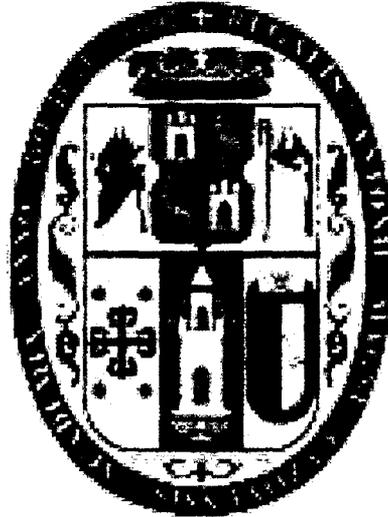


UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO
FACULTAD DE CIENCIAS QUÍMICAS, FÍSICAS Y MATEMÁTICAS
CARRERA PROFESIONAL DE INGENIERÍA INFORMÁTICA Y DE SISTEMAS



**DESARROLLO DE UNA APLICACIÓN WEB ORIENTADA A
SERVICIOS PARA EL MONITOREO DE UNA FLOTA DE
VEHÍCULOS HACIENDO USO DE LA TECNOLOGÍA GPS**

Tesis presentada por:

Br. MARY HELEN CONZA BERROCAL

Para optar al Título Profesional de:

INGENIERO INFORMÁTICO Y DE SISTEMAS

Asesor:

Mgt. EDWIN CARRASCO POBLETE

CUSCO - PERÚ

2013

“Tesis auspiciada por la Universidad Nacional de San Antonio Abad del Cusco”

AGRADECIMIENTOS

Agradezco infinitamente a Dios por sus bendiciones, principalmente por darme la salud y tranquilidad necesaria para la consecución de este sueño anhelado.

Agradezco a mis padres por su apoyo **constante** y su amor incondicional, a mis hermanos Elvis y Nadia, a quienes admiro por su fortaleza y generosidad que no conoce límites, y a mis menores hermanos Joel y Juan quienes alegran mi entorno familiar.

Agradezco a mis amigos, a quienes admiro y estimo bastante, Alex Oviedo, Vladimir Ascue y Franco Rosa, por brindarme sus entendimientos y experiencia, muy en especial a Carlos Borda por su asesoramiento profesional. Aprecio bastante su contribución en el proyecto, pero me siento más agradecida por su contribución en mi desarrollo profesional.

Agradezco también a mis amigos Rubén Holguino y Paúl Flores por apoyarme con la recopilación de datos y las pruebas del proyecto, dando así muestras de su sincero aprecio.

Este triunfo es por ustedes.

Mary Helen.

DEDICATORIA

Dedico este paso importante a dos personas, que son la principal bendición que Dios puso en mi vida, a quienes amo y respeto.

A mi madre, una mujer de gran valor, que con su contraste de severidad y amor me ayudó a forjar mis mejores virtudes.

A mi padre, quien me brindó siempre sabio consejo, quien realiza muchos sacrificios para que mis hermanos y yo emprendamos y salgamos adelante.

Este triunfo es por y para ustedes.

Mary Helen.

PRESENTACIÓN

Señor Decano de la Facultad de Ciencias Químicas, Físicas y Matemáticas de la Universidad Nacional de San Antonio Abad del Cusco.

Señores miembros del jurado.

Al término de mis estudios profesionales y en cumplimiento con la normatividad establecida para optar el Título Profesional de Ingeniero Informático y de Sistemas, pongo a vuestra consideración el presente trabajo de tesis intitulado “DESARROLLO DE UNA APLICACIÓN WEB ORIENTADA A SERVICIOS PARA EL MONITOREO DE UNA FLOTA DE VEHÍCULOS HACIENDO USO DE LA TECNOLOGÍA GPS”, con la finalidad de contribuir al estudio de los sistemas de información geo-referencial y su implementación con tecnologías móviles que hoy por hoy se tienen al alcance.

Esperando que los miembros del jurado eximan las deficiencias que pudieran encontrar y valoren el contenido desarrollado, cuya finalidad es contribuir con el estudio y desarrollo tecnológico de nuestra sociedad.

Br. Mary Helen Conza Berrocal.

RESUMEN

El presente trabajo de investigación trata acerca del desarrollo de una aplicación geo-referencial, cuyo propósito es el de rastrear y monitorear una flota vehicular para una operadora de taxis que labora a nivel local.

El desarrollo metodológico del proyecto se basa en el método descriptivo, esquema usado para recopilar datos y capturar los requerimientos de la empresa en cuestión. Para la aplicación web se emplea las especificaciones del Proceso Unificado de Desarrollo de Software (PUDS).

Los vehículos están equipados con receptores GPS que incluyen módem inalámbrico, estos dispositivos calculan su respectiva posición, hora y velocidad; dichos parámetros son enviados mediante la red celular GSM/GPRS de manera automática a una estación central haciendo uso de los servicios web REST. En la estación central se procesan los parámetros y son almacenados en la base de datos MySQL del servidor web Apache; al mismo tiempo, envía esos datos mediante el uso de web sockets a las máquinas cliente, que en ese momento están rastreando los vehículos en tiempo real, dichas máquinas clientes visualizan la ubicación de los vehículos en un mapa que provee Google Maps.

El resultado de la investigación es un aplicativo que rastree una flota vehicular en tiempo real, a fin de contribuir al estudio de los sistemas de información geo-referencial y su implementación con tecnologías móviles que hoy en día tenemos al alcance.

Palabras clave: Aplicación geo-referencial, rastreo, monitoreo, método descriptivo, PUDS, GPS, receptor GPS, módem inalámbrico, GSM/GPRS, servicio web, REST, estación central, MySQL, servidor web, Apache, web sockets, Google Maps.

ABSTRACT

The research deals with the development of an application geo-refencial, whose purpose is to track and monitor vehicle fleet, for a taxi operator that works locally.

The methodological development of the project is based on the descriptive method, scheme used to collect data and capture the requirements of the company in question. The web application is based on the specifications Unified Process Software Development (PUDS).

The vehicles are equipped with GPS receivers with wireless modem included; these devices calculate their respective position and other parameters, which are sent automatically by GSM/GPRS cellular network signal to a central station using REST web services. At the central station parameters are processed and stored in the MySQL database of Apache web server, at the same time, sends that data to the client machines by using web sockets, which at the time are tracking of the vehicles in real time, such machines customers view vehicle location on a map provided by Google Maps.

The result of the research is an application which tracking a vehicle fleet, to do a contribution to the study of geo-referenced information systems and implementation of mobile technologies that today have handy.

Keywords: Geospatial application, tracking, monitoring, descriptive method, PUDS, GPS, GPS receiver, wireless modem, GSM/GPRS, web services, REST, Central Station, MySQL, web server, Apache, web sockets, Google Maps .

INTRODUCCIÓN

Los sistemas de Rastreo Vehicular Automatizado (AVL – Automatic Vehicle Location) se aplican a los sistemas de localización remota en tiempo real, basados generalmente en el uso de GPS y un sistema de transmisión que es frecuentemente un módem inalámbrico. Este sistema representa una herramienta poderosa en aplicaciones de administración de flotas de transporte, asignación de vehículos de emergencia, sistemas de transporte público, etc. especialmente si se integra con otras aplicaciones relacionadas como sistemas de Call Center, operadoras de taxis, planificadores de ruta, sistemas de bodega y empresas de despacho entre otras.

Este informe presenta los pasos requeridos para la construcción de un aplicativo web de rastreo vehicular para una operadora de taxis que trabaja a nivel local. Para tal fin, se presentan seis capítulos:

En el primer capítulo se identifica el problema referente a la construcción del prototipo, el objetivo general y objetivos específicos, la justificación del proyecto, el alcance, las delimitaciones y el marco metodológico.

En el segundo capítulo se mencionan los conceptos teóricos de las tecnologías implicadas para el desarrollo del prototipo. Conceptos que van desde el desarrollo de la aplicación para que los dispositivos GPS envíen de datos al servidor, hasta conceptos involucrados en la comunicación cliente-servidor en tiempo real.

En los capítulos posteriores se describe la realización del software, utilizando las especificaciones del Proceso Unificado de Desarrollo de Software, la cual consta de cuatro fases: inicio, elaboración, construcción y transición.

Finalmente, se presenta las observaciones, conclusiones, recomendaciones y trabajos futuros.

Se espera que el presente trabajo contribuya con el conocimiento para posteriores investigaciones relacionadas a rastreo satelital en tiempo real, e incentive el interés hacia el desarrollo de aplicaciones geo-referenciales integrados con tecnologías móviles.

CONTENIDO

AGRADECIMIENTOS.....	i
DEDICATORIA	ii
PRESENTACIÓN	iii
RESUMEN	iv
ABSTRACT	v
INTRODUCCIÓN.....	vi
CONTENIDO	vii
ÍNDICE DE ILUSTRACIONES	xii
ÍNDICE DE TABLAS.....	xiv
CAPÍTULO 1. PLANTEAMIENTO DEL PROYECTO	1
1.1 TEMA DE TESIS	2
1.2 PLANTEAMIENTO DEL PROBLEMA	2
1.3 IDENTIFICACIÓN DEL PROBLEMA.....	3
1.4 OBJETIVOS	3
1.4.1 OBJETIVO GENERAL.....	3
1.4.2 OBJETIVOS ESPECÍFICOS	3
1.5 JUSTIFICACIÓN	3
1.6 ALCANCE Y DELIMITACIÓN DEL PROYECTO.....	4
1.6.1 ALCANCE	4
1.6.2 DELIMITACIÓN	4
1.7 MARCO METODOLÓGICO.....	5
1.7.1 METODOLOGÍA DEL PROYECTO.....	5
1.7.2 MÉTODOLÓGÍA DE DESARROLLO DE LA APLICACIÓN	5
CAPÍTULO 2. MARCO TEÓRICO	6
2.1 ANTECEDENTES.....	7

2.1.1	TRANSGPS.....	7
2.1.2	LOJACK STOLEN VEHICLE RECOVERY SYSTEM.....	8
2.1.3	OPENGTS	9
2.2	METODOLOGÍA.....	10
2.2.1	PROCESO DE INVESTIGACIÓN Y DESARROLLO EXPERIMENTAL (I+D)	10
2.2.2	PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE.....	11
2.3	CONCEPTOS TEÓRICOS.....	14
2.3.1	SISTEMA GPS.....	14
2.4	TECNOLOGÍAS.....	16
2.4.1	RECEPTOR GPS.....	16
2.4.2	CLOUD COMPUTING.....	17
2.4.3	ANDROID.....	18
2.4.4	WEB SERVICES.....	22
2.4.5	REST.....	25
2.4.6	JAVA SERVLET.....	27
2.4.7	WEB SOCKET	29
2.5	PROTOCOLOS Y ESTÁNDARES.....	31
2.5.1	ESTÁNDAR NMEA	31
2.5.2	XMLHttpRequest	32
2.6	LENGUAJES DE PROGRAMACIÓN	33
2.6.1	UML	33
2.6.2	JAVA.....	34
2.6.3	HTML.....	35
2.6.4	JAVASCRIPT.....	35
2.6.5	CSS	36
2.6.6	SQL.....	37

2.6.7	XML	37
2.7	HERRAMIENTAS Y FRAMEWORKS	39
2.7.1	TOMCAT	39
2.7.2	DOM	40
2.7.3	AJAX	40
2.7.4	GOOGLE WEB TOOLKIT	42
2.7.5	GOOGLE MAPS	45
2.7.6	MySQL	47
2.7.7	MySQL WORKBENCH	48
2.7.8	ECLIPSE	48
CAPÍTULO 3. FASE DE INICIO		50
3.1	DESCRIPCIÓN DEL PROBLEMA REAL	51
3.2	DESCRIPCIÓN DE LA EMPRESA	52
3.3	PROCESO DEL NEGOCIO	52
3.4	DIAGRAMA BPMN DEL PROCESO DEL NEGOCIO	54
3.5	MODELO DE NEGOCIO	55
3.5.1	PAQUETES DE CASO DE USO DEL NEGOCIO	55
3.5.2	UNIDADES ORGANIZACIONALES	55
3.5.3	IDENTIFICACIÓN DE LOS ACTORES DEL NEGOCIO	56
3.5.4	DIAGRAMA DE CASOS DE USO DEL NEGOCIO	56
3.5.5	DESCRIPCIÓN DE LOS CASOS DE USO DEL NEGOCIO	57
3.5.6	REALIZACIÓN DE LOS CASOS DE USO DEL NEGOCIO	58
3.5.7	LISTA DE REQUERIMIENTOS	59
3.6	CASOS DE USO DEL SISTEMA	60
3.6.1	IDENTIFICACIÓN DE LOS ACTORES	60
3.6.2	DIAGRAMA DE CASOS DE USO DEL SISTEMA	61

3.6.3	MATRIZ DE TRAZABILIDAD.....	62
3.6.4	CASOS DE USO DE ALTO NIVEL	63
3.6.5	CASOS DE USO EXPANDIDO.....	65
3.6.6	DIAGRAMA DE CLASES DE ANÁLISIS.....	70
CAPÍTULO 4. FASE DE ELABORACIÓN		73
4.1	DIAGRAMAS DE SECUENCIA.....	74
4.2	DIAGRAMAS DE COLABORACIÓN	80
4.3	DIAGRAMA DE CLASES	83
4.4	ARQUITECTURA CANDIDATA.....	84
4.4.1	DISEÑO DE LA TOPOLOGÍA DEL SISTEMA	84
4.4.2	DISEÑO DE LA TOPOLOGÍA GENERAL DEL SISTEMA.....	87
CAPÍTULO 5. FASE DE CONSTRUCCIÓN.....		88
5.1	IMPLEMENTACIÓN DEL SISTEMA.....	89
5.1.1	IMPLEMENTACIÓN DEL SERVIDOR.....	90
5.1.2	IMPLEMENTACIÓN EN DISPOSITIVOS GPS.....	92
5.1.3	IMPLEMENTACIÓN EN MÁQUINA CLIENTE	92
5.1.4	MODELO RELACIONAL.....	93
5.1.5	DIAGRAMA DE COMPONENTES	94
5.1.6	DIAGRAMA DE DESPLIEGUE.....	95
5.1.7	INTERFASES DEL SISTEMA A TRAVÉS DE CASOS DE USO REALES.....	96
CAPÍTULO 6. FASE DE TRANSICIÓN.....		107
6.1	IMPLANTACIÓN DEL SISTEMA	108
6.1.1	PREPARACIÓN DE LA VERSIÓN DE ACEPTACIÓN.....	108
6.1.2	INSTALACIÓN DE LA VERSIÓN DE ACEPTACIÓN.....	108
6.1.3	EVALUACIÓN DE LAS PRUEBAS	108
6.1.4	CASOS DE USO DE PRUEBA.....	109

CONCLUSIONES	116
RECOMENDACIONES.....	117
REFERENCIAS	118
REFERENCIA BIBLIOGRÁFICA	118
REFERENCIAS DE PÁGINAS WEB	119
ANEXOS	121

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Interfaz de TransGPS	8
Ilustración 2: Interfaz del sistema OpenGTS	9
Ilustración 3: Ciclo de vida del software	11
Ilustración 4: Sistema de satélites GPS	14
Ilustración 5: Cloud Computing	17
Ilustración 6: Arquitectura de Android	19
Ilustración 7: Comparación entre el funcionamiento de Web Socket y Polling	30
Ilustración 8: Diferencia de una aplicación web clásica y una que implementa Ajax	41
Ilustración 9: Google Maps	46
Ilustración 10: Interfaz de Mysql Workbench	48
Ilustración 11: Interfaz de Eclipse	49
Ilustración 12: Diagrama BPMN del proceso del negocio	54
Ilustración 13: Diagrama de Paquetes del Negocio	55
Ilustración 14: Estructura organizacional del negocio	55
Ilustración 15: Actores del negocio	56
Ilustración 16: Diagrama de casos de uso del negocio	56
Ilustración 17: Diagrama de realización de casos de uso	58
Ilustración 18: Actores del sistema	60
Ilustración 19: Diagrama de casos de uso	61
Ilustración 20: Diagrama de clase de análisis " Configurar parámetros límite"	70
Ilustración 21: Diagrama de clase de análisis " Asignar dispositivo GPS a vehículo"	70
Ilustración 22: Diagrama de clase de análisis " Asignar conductor a vehículo"	71
Ilustración 23: Diagrama de clase de análisis " Rastrear vehículos en tiempo real"	71
Ilustración 24: Diagrama de clase de análisis " Mostrar rutas seguidas por los vehículos"	71
Ilustración 25: Diagrama de secuencia "Configurar parámetros de rastreo"	74
Ilustración 26: Diagrama de secuencia " Asignar dispositivo GPS a vehículo"	75
Ilustración 27: Diagrama de secuencia " Asignar conductor a vehículo"	76
Ilustración 28: Diagrama de secuencia " Rastrear vehículos en tiempo real"	77
Ilustración 29: Diagrama de secuencia " Mostrar rutas seguidas por los vehículos"	78
Ilustración 30: Diagrama de secuencia " Consultar reportes"	79

Ilustración 31: Diagrama de secuencia " Cambiar estado de vehículo"	79
Ilustración 32: Diagrama de colaboración "Rastrear en tiempo real"	80
Ilustración 33: Diagrama de colaboración " Asignar dispositivo GPS a vehículo"	80
Ilustración 34: Diagrama de colaboración " Asignar conductor a vehículo"	80
Ilustración 35: Diagrama de colaboración " Rastrear vehículos en tiempo real"	81
Ilustración 36: Diagrama de colaboración " Mostrar rutas seguidas por los vehículos"	81
Ilustración 37: Diagrama de colaboración " Consultar reportes"	81
Ilustración 38: Diagrama de colaboración " Cambiar estado de vehículo"	82
Ilustración 39: Diagrama de clases del sistema	83
Ilustración 40: Esquema general del sistema	84
Ilustración 41: Diseño de la topología general del sistema	87
Ilustración 42: Diagrama de estructura del modelo de implementación	89
Ilustración 43: Implementación del servidor	90
Ilustración 44: Diseño del modelo relacional	93
Ilustración 45: Diagrama de componentes	94
Ilustración 46: Diagrama de despliegue	95
Ilustración 47: Interfaz del caso de uso real "Visualizar vehículos en tiempo real"	96
Ilustración 48: Interfaz del caso de uso real "Asignar dispositivo GPS a vehículo"	97
Ilustración 49: Interfaz del caso de uso real "Asignar conductor a vehículo"	98
Ilustración 50: Interfaz del caso de uso real "Rastrear vehículos en tiempo real"	99
Ilustración 51: Interfaz del caso de uso real "Visualizar rutas seguidas por los vehículos"	101
Ilustración 52: Interfaz del caso de uso real "Consultar reportes"	103
Ilustración 53: Interfaz del caso de uso real "Cambiar estado de vehículo"	105

ÍNDICE DE TABLAS

Tabla 1: Formato de la sentencia \$GPRMC	31
Tabla 2: Tipos de diagramas oficiales de UML	34
Tabla 3: Descripción de los casos de uso del negocio	57
Tabla 4: Requisitos funcionales	59
Tabla 5: Matriz de trazabilidad	62
Tabla 6: Caso de uso de alto nivel "Configurar parámetros límite"	63
Tabla 7: Caso de uso de alto nivel "Asignar dispositivo GPS a vehículo"	63
Tabla 8: Caso de uso de alto nivel "Asignar conductor a vehículo"	63
Tabla 9: Caso de uso de alto nivel "Rastrear vehículos en tiempo real"	63
Tabla 10: Caso de uso de alto nivel "Mostrar rutas seguidas por los vehículos"	64
Tabla 11: Caso de uso de alto nivel " Consultar reportes"	64
Tabla 12: Caso de uso de alto nivel "Cambiar estado de vehículo"	64
Tabla 13: Caso de uso expandido "Configurar parámetros límite"	65
Tabla 14: Caso de uso expandido "Asignar dispositivo GPS a vehículo"	66
Tabla 15: Caso de uso expandido "Asignar conductor a vehículo"	67
Tabla 16: Caso de uso expandido "Rastrear vehículos en tiempo real"	68
Tabla 17: Caso de uso expandido "Mostrar rutas seguidas por los vehículos"	68
Tabla 18: Caso de uso expandido "Consultar reportes"	69
Tabla 19: Caso de uso expandido "Cambiar estado de vehículo"	70
Tabla 20: Descripción del caso de uso real "Visualizar vehículos en tiempo real"	96
Tabla 21: Detalle de caso de uso Real "Visualizar vehículos en tiempo real"	96
Tabla 22: Descripción del caso de uso real "Asignar dispositivo GPS a vehículo"	97
Tabla 23: Detalle de caso de uso Real "Asignar dispositivo GPS a vehículo."	97
Tabla 24: Descripción del caso de uso real "Asignar conductor a vehículo"	98
Tabla 25: Detalle de caso de uso Real "Asignar conductor a vehículo"	98
Tabla 26: Descripción del caso de uso real "Rastrear vehículos en tiempo real"	99
Tabla 27: Detalle de caso de uso Real "Rastrear vehículos en tiempo real"	100
Tabla 28: Descripción del caso de uso real "Visualizar rutas seguidas por los vehículos"	100
Tabla 29: Detalle de caso de uso Real "Visualizar rutas seguidas por los vehículos"	103

Tabla 30: Descripción del caso de uso real “Consultar reportes”	103
Tabla 31: Detalle de caso de uso Real “Consultar reportes”	104
Tabla 32: Descripción del caso de uso real “Cambiar estado de vehículo”	104
Tabla 33: Detalle de caso de uso Real “Cambiar estado de vehículo”	106
Tabla 34: Caso de prueba "Configurar parámetros límite"	110
Tabla 35: Caso de prueba "Asignar conductor a vehículo"	111
Tabla 36: Caso de prueba "Asignar dispositivo a vehículo"	111
Tabla 37: Caso de prueba "Rastrear vehículos en tiempo real"	112
Tabla 38: Caso de prueba "Mostrar rutas seguidas por los vehículos"	114
Tabla 39: Caso de prueba "Consultar reportes”	114
Tabla 40: Caso de prueba "Cambiar estado del vehículo”	115

CAPÍTULO 1. PLANTEAMIENTO DEL PROYECTO

En este capítulo se identifica el problema referente a la construcción de la aplicación, el objetivo general y objetivos específicos, la justificación del proyecto, el alcance, las delimitaciones y metodología.

1.1 TEMA DE TESIS

“DESARROLLO DE UNA APLICACIÓN WEB ORIENTADA A SERVICIOS PARA EL MONITOREO DE UNA FLOTA DE VEHÍCULOS HACIENDO USO DE LA TECNOLOGÍA GPS”

1.2 PLANTEAMIENTO DEL PROBLEMA

Actualmente, las organizaciones que operan vehículos a nivel local, realizan la monitorización de sus unidades de manera no automática, ya sea llenando fichas de control en cada tramo (como lo hacen las empresas de transporte público) o mediante llamadas (empresas de taxis), estas prácticas conllevan entre otros aspectos, al desorden, la falta de seguridad y a la mala administración de los recursos.

En lo referente a la seguridad no es posible la identificación de los conductores que atentan contra las normas de tránsito, poniendo en cuestionamiento no sólo su propia imagen, sino también la de la organización.

En lo que se refiere a la productividad, los costos del transporte son cada vez más altos debido al permanente aumento del precio del petróleo, por lo que una mala administración de este recurso tan valorado podría mermar enormemente la economía de las organizaciones que poseen unidades vehiculares. Ante la carencia de un control sobre las rutas de los vehículos, los recursos puestos a disposición de los conductores, pueden ser indebidamente aprovechados para fines propios. Otro problema es el incumplimiento de horas establecidas que deben acatar los conductores.

En el caso de las empresas operadoras de taxi, otro problema surge del desconocimiento del operador acerca de la posición de los vehículos que forman su flota al momento en el que un determinado cliente solicita sus servicios. Como es sabido, actualmente el operador pregunta a los conductores vía llamada de radio, por el vehículo que esté más cerca del cliente en cuestión; dicha acción aparte de ser ineficiente, muchas veces no es la más acertada.

Por tales motivos, se pretende solucionar estos problemas, desarrollando un software tomando como caso de estudio a la empresa de transporte “Turismo Ángeles del Cusco SAC”, la cual opera taxis a nivel local.

1.3 IDENTIFICACIÓN DEL PROBLEMA

Actualmente en la ciudad del Cusco, las organizaciones que operan vehículos, realizan el monitoreo de su flota de manera no automática, afectando de esta forma a la seguridad y productividad. La empresa de transporte “Turismo Ángeles del Cusco SAC” también presenta estos inconvenientes.

1.4 OBJETIVOS

1.4.1 OBJETIVO GENERAL

Desarrollar una aplicación web para la gestión y monitoreo de una flota de vehículos, haciendo uso de la tecnología GPS, a fin de mejorar la seguridad e incrementar la productividad.

1.4.2 OBJETIVOS ESPECÍFICOS

- Realizar un análisis técnico de las herramientas de software y hardware que se empleará en el desarrollo de la aplicación a lograr.
- Programar los dispositivos GPS para transferir los datos automáticamente al servidor.
- Implementar el centro de control o estación central, para hacer posible la comunicación con los receptores GPS y los navegadores de las máquinas cliente.
- Implementar la transmisión de datos bidireccional cliente-servidor, para la comunicación en tiempo real.

1.5 JUSTIFICACIÓN

El avance tecnológico ha llegado a niveles antes no imaginados, los cuales están posibilitando el desarrollo de aplicaciones cada vez más sofisticadas y eficientes. Actualmente muchas aplicaciones están adoptando la filosofía de ser independientes de la plataforma, es decir que el sistema operativo en la que uno se

encuentra no debería representar impedimento a la hora de ejecutar dicho programa, haciendo factible la portabilidad de estos aplicativos tanto en desktops, laptops, o tablets que accedan a internet.

Ante la demanda de aplicaciones con características que estén acorde con el avance tecnológico, y ante la necesidad de administrar eficientemente los recursos de transporte, es necesario desarrollar una aplicación para el rastreo y monitoreo de flotas vehiculares con tecnología reciente.

La empresa de transportes “Turismo Ángeles del Cusco SAC” maneja una flota de 50 taxis en el Cusco, dicha cantidad ya no es administrable, por ello la empresa requiere de un sistema que monitoree sus unidades vehiculares. Por su bajo presupuesto se requiere que dicha aplicación sea accesible desde internet.

1.6 ALCANCE Y DELIMITACIÓN DEL PROYECTO

1.6.1 ALCANCE

El presente trabajo de investigación tiene por objeto el desarrollo de una aplicación de monitoreo de una flota vehicular. Además se realizará un análisis técnico de las herramientas software y hardware implicados.

1.6.2 DELIMITACIÓN

Se desarrollará un aplicativo orientado a una organización, y se harán las pruebas respectivas con el usuario, mas no es objeto del presente proyecto llegar a implementarlo, puesto que el costo de implementación es elevado y la informalidad es un problema para las organizaciones a nivel local; como es el caso de la empresa de transportes objeto del presente estudio

Escapa al alcance de la presente investigación la implementación electrónica de los dispositivos GPS, que si bien podrían prestar mayores prestaciones respecto a la seguridad y precisión de datos al presente proyecto, requeriría de una investigación exhaustiva en el área de la electrónica.

1.7 MARCO METODOLÓGICO

1.7.1 METODOLOGÍA DEL PROYECTO

La metodología que se utilizará es el proceso de investigación y desarrollo experimental (I+D), esquema usado para la recolección de información y la construcción de prototipos respectivamente.

La metodología que se utiliza en el desarrollo de este proyecto, tiene como base el método descriptivo, al cual se recurre para recoger, organizar, resumir y analizar los resultados de las observaciones y entrevistas; recopilando datos a fin de tener una idea clara de la situación actual de la empresa en cuestión.

1.7.2 MÉTODOLÓGÍA DE DESARROLLO DE LA APLICACIÓN

Para el desarrollo de la aplicación propuesta, se emplea las especificaciones del Proceso Unificado de Desarrollo de Software (PUDS), que es ampliamente utilizado en la construcción de sistemas que requieren de iteraciones para el entendimiento real de las necesidades del usuario.

CAPÍTULO 2. MARCO TEÓRICO

Se mencionan los conceptos teóricos de las tecnologías implicadas para el desarrollo del prototipo. Conceptos que van desde el desarrollo de la aplicación para que los dispositivos GPS envíen de datos al servidor, hasta conceptos involucrados en la comunicación cliente-servidor en tiempo real.

2.1 ANTECEDENTES

En la recopilación de datos realizada en los archivos de la biblioteca de la Universidad Nacional de San Antonio Abad del Cusco, se pudo constatar que no existen trabajos anteriores vinculados con el presente proyecto de investigación.

A nivel nacional varias empresas de transporte, implementan “Lojack”, el cual es un software de Estados Unidos. En Lima metropolitana se tiene “TransGPS”. Posteriormente se expone a “OpenGTS”, que es un sistema de monitoreo en software libre.

2.1.1 TRANSGPS

Sistema de monitoreo satelital de flota vehicular vía internet en tiempo real; el cual funciona en Lima Metropolitana, realiza monitoreo de flotas vehiculares a tiempo completo.

Fue desarrollado por MRVisual Corp SAC, la cual es una empresa peruana dirigida al transporte de pasajeros, cuentan con una experiencia de 10 años en el rubro, con importantes productos software y equipos de telecomunicaciones, que sirven como herramienta de gestión; y está conformada por profesionales multidisciplinarios orientados al servicio de transporte, educación, medio ambiente y afines.

Es aplicable a buses de transporte de pasajeros, camiones, autos, personas y mascotas.

Incluye acceso a los siguientes reportes: Paradas, kilómetros recorridos, excesos de velocidad, trayecto histórico de rutas ejecutadas. Cuenta con plataformas de visualización cartográfica y fotografía satelital.

Ubicación y visualización inmediata de vehículos, el cliente puede monitorear sus unidades mediante laptop, PC, Iphone o Blackberry/Smartphone, que accedan a Internet.

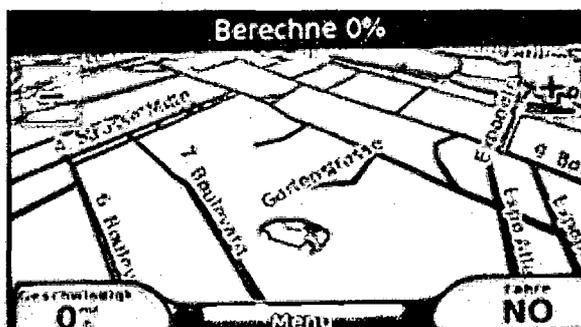


Ilustración 1: Interfaz de TransGPS

Fuente:

http://www.mrvisual.com/index.php?option=com_content&view=article&id=6&catid=3&Itemid=27 [RW001]

2.1.2 LOJACK STOLEN VEHICLE RECOVERY SYSTEM

Lojack es el sistema utilizado por la policía de EE.UU y otros departamentos policiales en diferentes países. Este sistema ha logrado el liderazgo en recuperación de vehículos robados y monitoreo de activos, el éxito ha sido tal que más de 300 000 vehículos han sido recuperados, generando un valor superior a los US\$ 5 mil millones de dólares que no se perdieron.

Hunter inició sus operaciones en Perú en Abril del 2008 y es el licenciatarío exclusivo de la tecnología LoJack en este país.

Actualmente, cuenta con 2 líneas de productos: "Hunter LoJack", orientada especialmente al rastreo y localización de vehículos robados con la tecnología exclusiva de la empresa y "Soluciones tecnológicas corporativas", cuyo propósito es orientar el uso de la tecnología GPS a la administración y control de flotas, carga y mercadería, reduciendo costos y mejorando su productividad.

Fuente: <http://www.hunterlojack.com/website/index.php/acerca-de-hunter/> [RW002]

2.1.3 OPENGTS

OpenGTS, es un sistema de seguimiento GPS en software libre, recupera la información de telemetría desde los dispositivos remotos y proporciona mapas e informes que representan los movimientos de la flota. Esta alternativa en software libre a los sistemas propietarios de control de flotas ha ayudado a los usuarios de más de 70 países a controlar sus coches, camiones, tráiler, autobuses y barcos, así como a localizar gente y mascotas. Fue desarrollado por GTS Enterprise el año 2007.

OpenGTS soporta una configuración flexible e incluye una impresionante lista de funcionalidades. Además de una interfaz web personalizable basada en CSS, incluye un sistema de administración de usuarios que permite al administrador configurar usuarios, cuentas, grupos de vehículos y los propios vehículos.

Posee funcionalidades como perimetraje, geozonas y geolocalización inversa, técnicas que usa OpenGTS para mapear coordenadas en direcciones. Los mapas son cortesía de OpenStreetMap, OpenLayers, Google Maps o GeoServer.

OpenGTS está implementado en Java, lo que significa que para su implementación en una estación central requerirá de versiones actuales de Sun JRE, Apache Ant, Tomcat y los drivers MySQL JDBC.

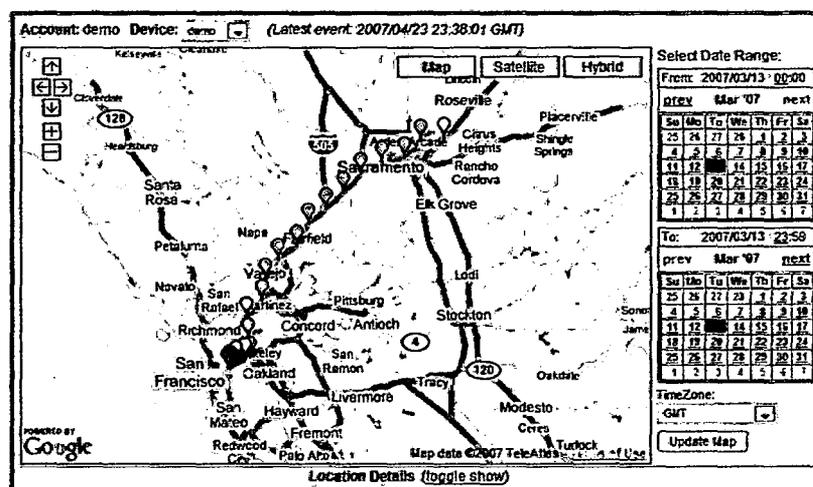


Ilustración 2: Interfaz del sistema OpenGTS

Fuente: <http://opengts.sourceforge.net/> [RW003]

2.2 METODOLOGÍA

2.2.1 PROCESO DE INVESTIGACIÓN Y DESARROLLO EXPERIMENTAL (I+D)

La investigación y el desarrollo experimental (I+D) comprenden el trabajo creativo llevado a cabo de forma sistemática para incrementar el volumen de conocimientos, incluido el conocimiento del hombre, la cultura y la sociedad y el uso de esos conocimientos para crear nuevas aplicaciones.

El término I+D engloba tres actividades:

1. La investigación básica consiste en trabajos experimentales o teóricos que se emprenden principalmente para obtener nuevos conocimientos acerca de los fundamentos de los fenómenos y hechos observables, sin pensar en dales ninguna aplicación o utilización determinada.
2. La investigación aplicada consiste también en trabajos realizados para adquirir nuevos conocimientos; sin embargo, está dirigida fundamentalmente hacia un objetivo práctico específico.
3. El desarrollo experimental consiste en trabajos sistemáticos que aprovechan los conocimientos existentes obtenidos de la investigación y/o la experiencia práctica, y está dirigido a la producción de nuevos materiales, productos o dispositivos; a la puesta en marcha de nuevos procesos, sistemas y servicios, o a la mejora sustancial de los ya existentes. ([RB012], pág 30)

2.2.2 PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE

Es un modelo de desarrollo de software que se caracteriza por estar dirigido por casos de uso, estar centrado en la arquitectura y por ser iterativo e incremental. Está compuesto de cuatro fases denominadas Inicio, Elaboración, Construcción y Transición. Cada una de estas fases es a su vez dividida en una serie de iteraciones (la de inicio sólo consta de varias iteraciones en proyectos grandes). Cada una de estas iteraciones se divide a su vez en una serie de disciplinas que recuerdan a las definidas en el ciclo de vida clásico o en cascada: Análisis de Requisitos, Diseño, Implementación y Prueba.

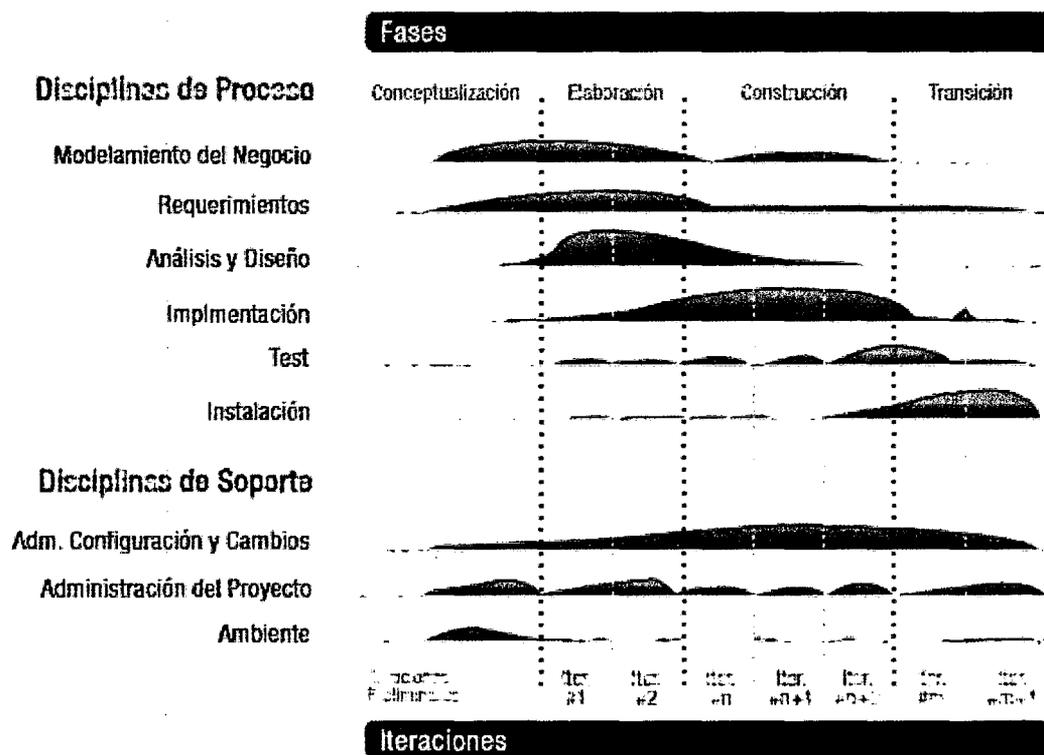


Ilustración 3: Ciclo de vida del software

Fuente: <http://informaticaadsi.blogspot.com/2010/03/informatica-adsi.html> [RW004]

2.2.2.1 FASES DEL CICLO DE VIDA

➤ **Fase de Inicio**

Primera fase del ciclo de vida del software que tiene por objetivo establecer y garantizar una buena base para el desarrollo de las posteriores fases. Los objetivos de esta fase son determinar el ámbito del sistema, delimitar el alcance del sistema y definir una arquitectura posible para el sistema.

➤ **Fase de Elaboración**

El objetivo de esta fase es proporcionar la arquitectura base para el desarrollo del sistema de información, en esta fase se realiza la construcción de los diagramas para el modelo de análisis y diseño, a fin de lograr la estructuración de un diagrama de clases adecuado para el desarrollo correcto del sistema.

➤ **Fase de Construcción**

Tercera fase del ciclo de vida del software, en la que el software es desarrollado a partir de una línea base de la arquitectura ejecutable, hasta el punto en el que se está listo para ser transmitido a las comunidades de usuarios.

➤ **Fase de Transición**

El objetivo de esta fase terminal es la construcción del sistema y que los usuarios puedan comenzar su operatividad con la ejecución del aplicativo, con el propósito de perfeccionar los casos de uso determinados en las fases anteriores. Es decir, esta fase corresponde a la construcción misma del software.

2.2.2.2 FLUJOS DE TRABAJO

➤ **Requisitos.**

Los servicios, restricciones y metas del sistema se definen en detalle y sirven como una especificación del sistema, no pudiéndose requerir nuevos resultados a mitad del proceso de elaboración del software.

➤ **Diseño del Sistema.**

El proceso de diseño del sistema divide los requerimientos en sistemas hardware o software y establece la arquitectura completa del sistema. El diseño del software identifica y describe las abstracciones fundamentales del sistema software y sus relaciones.

➤ **Implementación y prueba de unidades.**

Durante esta etapa, el diseño del software se lleva a cabo como un conjunto o unidades de programas. La prueba de unidades implica verificar que cada una cumpla su especificación.

➤ **Integración y prueba del sistema.**

Los programas o las unidades individuales de programas se integran y prueban como un sistema completo para asegurar que se cumplan los requerimientos del software. Después de las pruebas, el sistema software se entrega al cliente.

➤ **Despliegue**

Por lo general esta es la fase más larga del ciclo de vida del software. El sistema se instala y se pone en funcionamiento práctico. Implica corregir errores no descubiertos en las etapas anteriores del ciclo de vida, mejorar la implementación de las unidades del sistema y resaltar los servicios del sistema una vez que se descubran los nuevos requerimientos.

2.3 CONCEPTOS TEÓRICOS

2.3.1 SISTEMA GPS

El GPS (Global Positioning System) o conocido también como NAVSTAR-GPS es un sistema global de navegación por satélite (GNSS) que permite determinar la posición de los receptores GPS en todo el mundo durante las 24 horas del día, en cualquier condición climatológica.

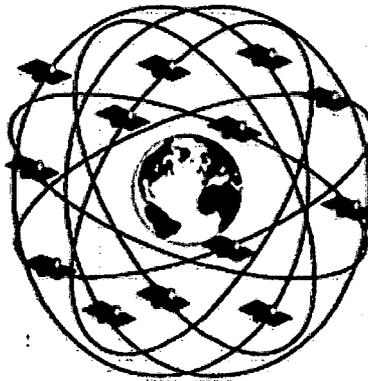


Ilustración 4: Sistema de satélites GPS

El GPS funciona mediante una red de 27 satélites (24 operativos y 3 de respaldo) en órbita sobre el globo, a 20200 Km, con trayectorias sincronizadas para cubrir toda la superficie de la tierra. Cuando se desea determinar la posición, el receptor que se utiliza para ello localiza automáticamente, como mínimo, tres satélites de la red, de los que recibe señales indicando la posición y el tiempo en el reloj de cada uno de ellos. En base a estas señales, el aparato sincroniza el reloj del GPS y calcula el retraso de las mismas; es decir, la distancia al satélite. Por triangulación¹... se consigue una precisión extrema en el

¹ Llamado también método de trilateración inversa, es un método que consiste en determinar la posición absoluta o coordenadas reales de un punto de medición, a partir de la posición relativa de dicho punto respecto a los tres satélites, y las coordenadas o posición de dichos satélites.

reloj del GPS, similar a la de los relojes atómicos² que llevan a bordo cada uno de los satélites. ([RB001], pág. 28).

2.3.1.1 EFEMÉRIDES

Es una información transmitida por los satélites, la cual contiene la predicción precisa de su órbita para una determinada época. Esta información radio difundida por cada satélite, se actualiza cada cierto tiempo, a través de los parámetros orbitales, controlados desde las estaciones en la Tierra. Las efemérides deben ser actualizadas por alteraciones que sufren las órbitas por efecto de la atracción del Sol y la Luna, la diferencia de gravedad entre distintas zonas de la corteza terrestre y el viento solar. Cada satélite transmite sus propias efemérides, a través de las cuales es posible calcular su posición en un sistema de referencia cartesiano geocéntrico.

2.3.1.2 ALMANAQUE

El almanaque es una información expedita de todas las órbitas, de esta manera permite predecir con cierta aproximación, la posición y la hora en que se situarán los satélites. El almanaque se actualiza cada 12,5 minutos, entregando la posición aproximada de todos los satélites sobre el horizonte, renovándose completamente cada 24 horas. Todos los satélites de la constelación transmiten el almanaque completo, cuya función es la planificación de observaciones por parte de los usuarios, también entrega información útil al receptor para conectarse con los satélites disponibles al momento de iniciar una medición.

² Tipo de reloj que para alimentar su contador utiliza una frecuencia de resonancia atómica normal, por lo que mantiene una escala de tiempo continua y estable que se mantiene a una precisión de 10^{-9} segundos por día.

2.4 TECNOLOGÍAS

2.4.1 RECEPTOR GPS

Es el conjunto de elementos (Software y Hardware) que permite determinar la posición, velocidad, tiempo, orientación y precisión entre parámetros necesarios usando la red de satélites GPS.

2.4.1.1 FUNCIONES

- Identificación y seguimiento de los códigos asociados a cada satélite.
- Determinación de la distancia.
- Decodificación de las señales de datos de navegación para obtener las efemérides, el almanaque.
- Realizar las correcciones (correcciones del reloj, correcciones ionosféricas).
- Determinación de la posición y velocidad.
- Validación de los resultados obtenidos y almacenamiento en memoria.
- Presentación de la información.

2.4.2 CLOUD COMPUTING

La computación en nube es un sistema informático basado en Internet y centros de datos remotos para gestionar servicios de información y aplicaciones. La computación en nube permite que los consumidores y las empresas gestionen archivos y utilicen aplicaciones sin necesidad de instalarlas en cualquier computadora con acceso a Internet. Esta tecnología ofrece un uso mucho más eficiente de recursos, como almacenamiento, memoria, procesamiento y ancho de banda, al proveer solamente los recursos necesarios en cada momento.

El término “nube” se utiliza como una metáfora de Internet y se origina en la nube utilizada para representar Internet en los diagramas de red como una abstracción de la infraestructura que representa.

Un ejemplo sencillo de computación en nube es el sistema de documentos y aplicaciones electrónicas Google Docs / Google Apps. Para su uso no es necesario instalar software o disponer de un servidor, basta con una conexión a Internet para poder utilizar cualquiera de sus servicios.

El servidor y el software de gestión se encuentran en la nube (Internet) y son directamente gestionados por el proveedor de servicios. De esta manera, es mucho más simple para el consumidor disfrutar de los beneficios. En otras palabras: la tecnología de la información se convierte en un servicio, que se consume de la misma manera que consumimos la electricidad o el agua.

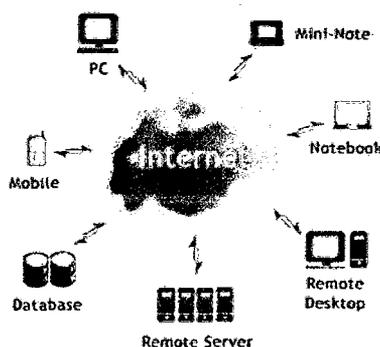


Ilustración 5: Cloud Computing

(Referencia: <http://www.computacionennube.org/computacion-en-nube/> [RW008])

2.4.3 ANDROID

Es un sistema operativo móvil basado en Linux, que junto con aplicaciones middleware está enfocado para ser utilizado en dispositivos móviles como teléfonos inteligentes, tabletas, Smartphones y otros dispositivos. Fue desarrollado inicialmente por Android Inc., una firma comprada por Google en 2005.

2.4.3.1 CARACTERÍSTICAS TÉCNICAS

Como asegura Jesús Gironés en su libro “el gran libro de Android” las características por las que Android se distingue de otras plataformas para móviles son:

- Android tiene una plataforma realmente abierta. Es una plataforma de desarrollo libre basada en Linux y de código abierto. Una de sus grandes ventajas es que se puede usar el sistema sin pagar royalties³.
- Portabilidad Asegurada. Las aplicaciones finales son desarrolladas en Java lo que nos asegura que podrán ser ejecutadas en gran variedad de dispositivos, tanto presentes como futuros. Esto se consigue gracias al concepto de máquina virtual.
- Arquitectura basada en componentes inspirados en Internet. Por ejemplo, el diseño de interfaz de usuario se hace en xml, lo que permite que una misma aplicación se ejecute en un móvil de pantalla reducida o en un netbook.
- Filosofía de dispositivo siempre conectado a Internet
- Gran cantidad de servicios incorporados: por ejemplo, localización basada tanto en GPS como en torres de telefonía móvil. Incorpora potentes bases de datos con SQL. Reconocimiento y síntesis de voz, navegador, mapas...

³ También llamado regalía, es el pago que se efectúa al titular de derechos de autor, patentes, marcas o know-how a cambio del derecho a usarlos o explotarlos

- Alto nivel de seguridad. Los programas se encuentran aislados unos de otros gracias al concepto de ejecución dentro de una caja que incorpora la máquina virtual. Cada aplicación dispone de una serie de permisos que limitan su rango de actuación (servicios de localización, acceso a Internet...)
- Optimización para baja potencia y poca memoria. Por ejemplo, Android utiliza la máquina virtual de Dalvik. Se trata de una implementación de Google de la máquina virtual de Java optimizada para dispositivos móviles.
- Alta calidad de gráficos y sonido: gráficos vectoriales suavizados, animaciones inspiradas en Flash, gráficos en 3 dimensiones basados en OpenGL. Incorpora códec estándar más comunes de audio y vídeo.

([RB003], página 19)

2.4.3.2 ARQUITECTURA

Como se puede ver está formada por cuatro capas. Una de las características más importantes es que todas las capas están basadas en software libre.

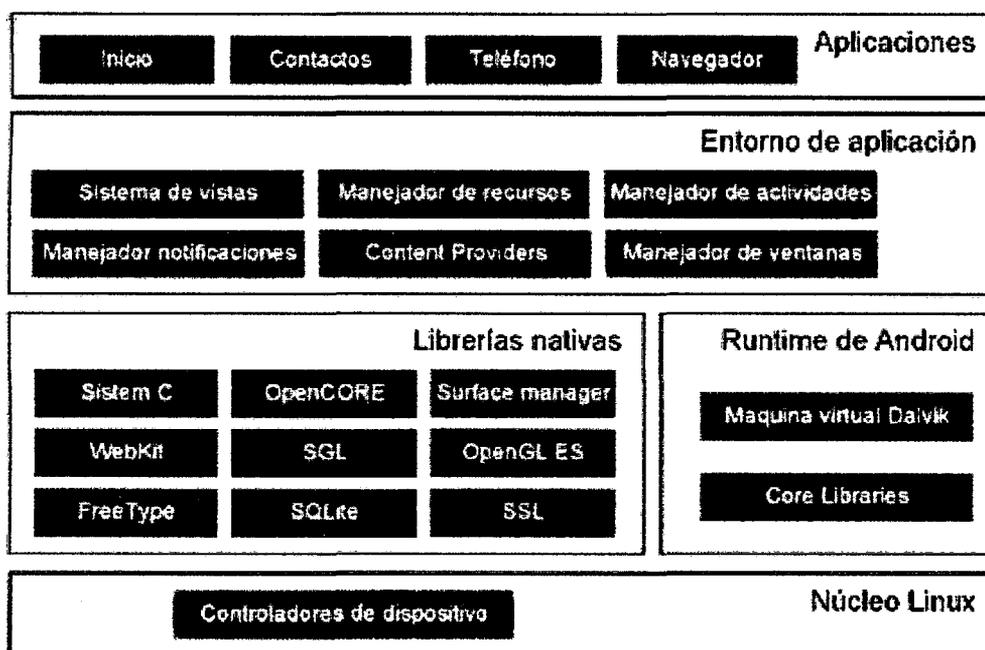


Ilustración 6: Arquitectura de Android

Fuente: [RB003] pág. 20

El núcleo Linux

Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software.

Runtime de Android

Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente. Dalvik ejecuta archivos en el formato Dalvik Executable (.dex), el cual está optimizado para memoria mínima. La Máquina Virtual está basada en registros y corre clases compiladas por el compilador de Java que han sido transformadas al formato.dex por la herramienta incluida "dx".

Librerías nativas

Android incluye un conjunto de bibliotecas de C/C++ usadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del marco de trabajo de aplicaciones de Android; algunas son: System C library (implementación biblioteca C estándar), bibliotecas de medios, bibliotecas de gráficos, 3D y SQLite, entre otras.

Entorno de aplicación

Los desarrolladores tienen acceso completo a los mismos APIs del framework usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework). Este mismo mecanismo permite que los componentes sean reemplazados por el usuario.

Aplicaciones

Las aplicaciones base incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos y otros. Todas las aplicaciones están escritas en lenguaje de programación Java.

2.4.3.3 DISEÑO Y DESARROLLO

Android, al contrario que otros sistemas operativos para dispositivos móviles como iOS o Windows Phone, se desarrolla de forma abierta y se puede acceder tanto al código fuente⁶⁵ como a la lista de incidencias⁶⁶ donde se pueden ver problemas aún no resueltos y reportar problemas nuevos.

El que se tenga acceso al código fuente no significa que se pueda tener siempre la última versión de Android en un determinado móvil, ya que el código para soportar el hardware (controladores) de cada fabricante normalmente no es público, así que faltaría un trozo básico del firmware para poder hacerlo funcionar en dicho terminal, y porque las nuevas versiones de Android suelen requerir más recursos, por lo que los modelos más antiguos quedan descartados por razones de memoria (RAM), velocidad de procesador, etc.

En sus comienzos, Android era eminentemente un sistema operativo pensado para usar con teclado ⁶⁷ y gracias a un cursor poder navegar entre las aplicaciones. Desde su comienzo, Android ha sido altamente personalizable. Poco después, antes del lanzamiento del primer teléfono Android, esta filosofía cambió para convertirse en eminentemente táctil, y poder competir contra el recién lanzado iPhone.

2.4.4 WEB SERVICES

Servicio web es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones, ya sea que estén desarrolladas en lenguajes de programación diferentes o estén sobre diferentes plataformas, sobre una red de ordenadores como el internet.

Las organizaciones OASIS (Organization for the Advancement of Structured Information Standards) y W3C (World Wide Web Consortium) son los comités responsables de la arquitectura y reglamentación de los servicios Web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios Web se ha creado el organismo WS-I⁴, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares.

2.4.4.1 ESTÁNDARES EMPLEADOS

- Web Services Protocol Stack: Así se denomina al conjunto de servicios y protocolos de los servicios Web.
- XML (Extensible Markup Language): Es el formato estándar para los datos que se vayan a intercambiar.
- SOAP (Simple Object Access Protocol) o XML-RPC (XML Remote Procedure Call): Protocolos sobre los que se establece el intercambio.
- Otros protocolos: los datos en XML también pueden enviarse de una aplicación a otra mediante protocolos normales como HTTP (Hypertext Transfer Protocol), FTP (File Transfer Protocol), o SMTP (Simple Mail Transfer Protocol).
- WSDL (Web Services Description Language): Es el lenguaje de la interfaz pública para los servicios Web. Es una descripción basada en XML de los requisitos funcionales necesarios para establecer una comunicación con los servicios Web.

⁴ Es una máquina que atiende las peticiones de los clientes web y les envía los recursos solicitados.

- UDDI (Universal Description, Discovery and Integration): Protocolo para publicar la información de los servicios Web. Permite comprobar que servicios web están disponibles.
- WS-Security (Web Service Security): Protocolo de seguridad estándar aceptada por OASIS (Organization for the Advancement of Structured Information Standards). Garantiza la autenticación de los actores y la confidencialidad de los mensajes enviados.

2.4.4.2 VENTAJAS

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.

2.4.4.3 INCONVENIENTES DE LOS SERVICIOS WEB

- Para realizar transacciones no pueden compararse en su grado de desarrollo con los estándares abiertos de computación distribuida como CORBA (Common Object Request Broker Architecture).
- Su rendimiento es bajo si se compara con otros modelos de computación distribuida, tales como RMI (Remote Method Invocation), CORBA o DCOM (Distributed Component Object Model). Es uno de los inconvenientes derivados de adoptar un formato basado en texto. Y es que entre los objetivos de XML no se encuentra la concisión ni la eficacia de procesamiento.

- Al apoyarse en HTTP, pueden esquivar medidas de seguridad basadas en firewall cuyas reglas tratan de bloquear o auditar la comunicación entre programas a ambos lados de la barrera.

2.4.4.4 MOTIVOS PARA SU IMPLEMENTACIÓN

- La principal razón para usar servicios Web es que se pueden utilizar con HTTP sobre TCP (Transmission Control Protocol) en el puerto 80. Dado que las organizaciones protegen sus redes mediante firewalls que filtran y bloquean gran parte del tráfico de Internet, cierran casi todos los puertos TCP salvo el 80, que es, precisamente, el que usan los navegadores. Los servicios Web utilizan este puerto, por la simple razón de que no resultan bloqueados. Es importante señalar que los servicios web se pueden utilizar sobre cualquier protocolo, sin embargo, TCP es el más común.
- Antes de que existiera SOAP, no había buenas interfaces para acceder a las funcionalidades de otros ordenadores en red. Las que había eran ad hoc y poco conocidas, tales como EDI (Electronic Data Interchange), RPC (Remote Procedure Call), u otras APIs.
- Una tercera razón por la que los servicios Web son muy prácticos es que pueden aportar gran independencia entre la aplicación que usa el servicio Web y el propio servicio. De esta forma, los cambios a lo largo del tiempo en uno no deben afectar al otro. Esta flexibilidad será cada vez más importante, dado que la tendencia a construir grandes aplicaciones a partir de componentes distribuidos más pequeños es cada día más utilizada.

2.4.5 REST

(Transferencia de Estado Representacional - Representational State Transfer) Es una técnica de arquitectura para sistemas hipermedia distribuidos como el internet.

Nació como una alternativa más simple a SOAP⁵ y a WSDL⁶ a los cuales logró desplazar, la clara evidencia está en que varios de los grandes proveedores de Web 2.0 están migrando a esta tecnología, incluyendo a Yahoo, Google y Facebook, quienes marcaron como obsoletos a sus servicios SOAP y WSDL, pasando a usar este modelo por la facilidad que ofrece.

REST define un set de principios arquitectónicos para diseñar servicios web enfocándose en los recursos del sistema, incluyendo cómo se accede al estado de dichos recursos y cómo se transfieren por HTTP hacia clientes escritos en diversos lenguajes. Los sistemas que siguen los principios REST se llaman con frecuencia RESTful.

2.4.5.1 PRINCIPIOS DE ARQUITECTÓNICOS FUNDAMENTALES

- Un protocolo cliente/servidor sin estado: Una aplicación o cliente de servicio web REST debe incluir dentro del encabezado y del cuerpo HTTP de la petición todos los parámetros, contexto y datos que necesita el servidor para generar la respuesta. Una petición completa e independiente hace que el servidor no tenga que recuperar ninguna información de contexto o estado al procesar la petición. De esta manera, el no mantener estado mejora el rendimiento de los servicios web y simplifica el diseño e implementación de los componentes del servidor, ya que la ausencia de estado en el servidor elimina la necesidad de sincronizar los datos de la sesión con una aplicación externa.

⁵ Protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

⁶ Servicios web basados en el Lenguaje de Descripción de Servicios Web (Web Services Description Language)

- Utiliza los métodos HTTP de manera explícita, REST hace que los desarrolladores usen los métodos HTTP explícitamente de manera que resulte consistente con la definición del protocolo. Este principio de diseño básico establece una asociación uno-a-uno entre las operaciones de crear, leer, actualizar y borrar y los métodos HTTP. De acuerdo a esta asociación:
 - se usa POST para crear un recurso en el servidor
 - se usa GET para obtener un recurso
 - se usa PUT para cambiar el estado de un recurso o actualizarlo
 - se usa DELETE para eliminar un recurso
- Usa una sintaxis universal para identificar los recursos. En un sistema REST, cada recurso es direccionable únicamente a través de su URI⁷.
- Representa los recursos en XML, JSON, XHTML. Esto permite que el servicio sea utilizado por distintos clientes escritos en diferentes lenguajes, corriendo en diversas plataformas y dispositivos. El uso de estas representaciones y del encabezado HTTP Accept es un mecanismo conocido como negociación de contenido, el cual le permite a los clientes elegir qué formato de datos puedan leer, y minimiza el acoplamiento de datos entre el servicio y las aplicaciones que lo consumen.

⁷ Aunque se acostumbra llamar URLs a todas las direcciones Web, URI es un identificador más completo, que organiza los recursos en forma de directorios.

2.4.6 JAVA SERVLET

Java Servlets son programas en java que proporcionan un mecanismo simple y consistente para extender la funcionalidad de un web server.

Servlets son precompilados por el programador, una vez puestos en el servidor de páginas (web server) éste los carga, compila y se ejecuta al momento que un usuario remoto lo solicita desde un navegador.

El servidor mantiene el servlet activo en memoria todo el tiempo, a menos que sea reiniciado, por ello el servlet tiene una ventaja y una desventaja notoria. Esto permite que la respuesta sea rápida, por que pueda que otro usuario necesite el mismo servlet, y por otro lado trae una desventaja, si es un servlet no muy requerido por los usuarios, consumirá innecesariamente recursos del computador. Por ello es mejor ser selectivo con los servlet.

Los servlets son objetos que corren dentro y fuera del contexto de un contenedor de servlets (ej: Tomcat) y extienden su funcionalidad. La palabra servlet deriva de otra anterior, applet, que se refería a pequeños programas que se ejecutan en el contexto de un navegador web.

El uso más común de los servlets es generar todas páginas web de forma dinámica a partir de los parámetros de la petición que envíe el navegador web.

2.4.6.1 CARACTERÍSTICAS

Como dice García, Rodríguez e Imaz, las características de la tecnología de los servlets son las siguientes:

1. Son independientes del servidor utilizado y de su sistema operativo, lo que quiere decir que a pesar de estar escritos en Java, el servidor puede estar escrito en cualquier lenguaje de programación, obteniéndose exactamente el mismo resultado que si estuviera en Java.
2. Los servlets pueden llamar a otros servlets, e incluso a métodos concretos de otros servlets. De esa forma se puede distribuir de forma más eficiente el trabajo a realizar. Por ejemplo, se podría tener un servlet encargado de la interacción con los clientes y que llamara a otro servlet para que a su vez se

encargara de la comunicación con una base de datos. De igual forma, los servlets permiten direccionar peticiones de servicios a otros servlets (en la misma máquina o en una máquina remota).

3. Los servlets pueden obtener fácilmente información acerca del cliente (la permitida por el protocolo HTTP), tal como su dirección IP, el puerto que se utiliza en la llamada, el método utilizado (GET, POST,...), etc.
4. Permiten además la utilización de cookies y sesiones, de forma que se puede guardar información específica acerca de un usuario determinado, personalizando de esta forma la interacción cliente-servidor. Una clara aplicación es mantener la sesión con un cliente.
5. Los servlets pueden actuar como enlace entre el cliente y una o varias bases de datos en arquitecturas cliente-servidor de 3 capas (si la base de datos está en un servidor distinto).
6. Asimismo, pueden realizar tareas de proxy para applet. Debido a las restricciones de seguridad, un applet no puede acceder directamente por ejemplo a un servidor de datos localizado en cualquier máquina remota, pero el servlet sí puede hacerlo de su parte.
7. Al igual que los programas CGI, los servlets permiten la generación dinámica de código HTML dentro de una propia página HTML. Así, pueden emplearse servlets para la creación, banners, etc.

([RB002], pág. 15)

2.4.6.2 CICLO DE VIDA DE LOS SERVLET

El ciclo de vida de un Servlet se divide en los siguientes puntos:

- El cliente solicita una petición a un servidor vía URL.
- El servidor recibe la petición.
- Si es la primera, se utiliza el motor de Servlets para cargarlo y se llama al método `init()`.

- Si ya está iniciado, cualquier petición se convierte en un nuevo hilo. Un Servlet puede manejar múltiples peticiones de clientes.
- Se llama al método `service()` para procesar la petición devolviendo el resultado al cliente.
- Cuando se apaga el motor de un Servlet se llama al método `destroy()`, que lo destruye y libera los recursos abiertos.

2.4.7 WEB SOCKET

Normalmente cuando un navegador visita una página web, una solicitud HTTP es enviada al servidor que hospeda a la página. El servidor web recibe esta solicitud y retorna una respuesta. El mecanismo es simple y fácil de implementar, pero no es para nada conveniente si se pretende realizar una aplicación en tiempo real, como son juegos multijugador, aplicaciones de monitorización, chats y herramientas de trabajo colaborativo.

Web socket es un protocolo nuevo para la web bajo TCP, que a diferencia de la conexión bajo HTTP, proporciona un canal de comunicación bidireccional y full-dúplex sobre un único socket TCP. Esta especificación se basa en un canal de comunicación bidireccional entre un cliente y un servidor, mediante el cual pueden enviarse mensajes de un sentido a otro en cualquier momento sin necesidad de que haya una petición HTTP de por medio, por ello brinda una enorme reducción de tráfico en la red y de latencia⁸ en comparación a las soluciones polling⁹ que son soluciones que se emplean a menudo para entregar datos en tiempo real a los clientes.

⁸ Suma de retardos temporales dentro de una red. Un retardo es producido por la demora en la propagación y transmisión de paquetes dentro de la red.

⁹ Esta tecnología el navegador envía una solicitud HTTP en intervalos de tiempo regulares e inmediatamente recibe una respuesta. Esta técnica fue el primer intento para que los navegadores reciban la información en tiempo real, pero obviamente, es una solución burda, puesto que los datos no siempre son predecibles, lo que puede ocasionar la realización de muchas solicitudes en vano, y como resultado muchas conexiones son abiertas y cerradas innecesariamente.

Como las conexiones TCP ordinarias sobre puertos diferentes al 80 son habitualmente bloqueadas por los administradores de redes, el uso de esta tecnología proporciona una solución a este tipo de limitaciones proveyendo una funcionalidad similar a la apertura de varias conexiones en distintos puertos, pero multiplexando diferentes servicios WebSocket sobre un único puerto TCP (a costa de una pequeña sobrecarga del protocolo). WebSocket es soportado actualmente por Google Chrome, Internet Explorer, Firefox, Safari y Opera.

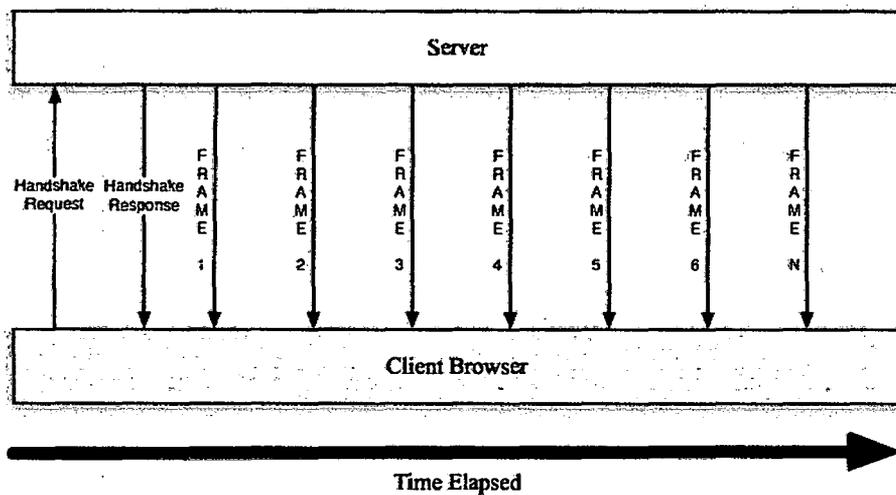


Ilustración 7: Comparación entre el funcionamiento de Web Socket y Polling

Fuente http://www.marakana.com/bookshelf/html5_tutorial/web_socket.html [RW006]

2.5 PROTOCOLOS Y ESTÁNDARES

2.5.1 ESTÁNDAR NMEA

Es un protocolo que se usa para la navegación tanto marítima como terrestre creado por la asociación NMEA¹⁰. Este estándar se creó para el intercambio de información digital entre productos electrónicos marinos.

Es un formato de transmisión de datos entre un receptor GPS y otro dispositivo, este protocolo define la norma para que cada equipo que sea emisor de NMEA pueda ser escuchado por muchos receptores.

2.5.1.1 NMEA 0183

El estándar NMEA 0183 define requisitos de señal eléctrica, protocolo de transmisión de datos y tiempo. Este estándar ha sido adoptado por el comité internacional de electrotécnica de la comunidad Europea como la base para el estándar internacional.

2.5.1.2 SENTENCIA \$GPRMC

La sentencia \$GPRMC fue creada según el estándar NMEA (National Marine Electronics Association) incluye la información necesaria y fundamental para transmitir datos de la posición y el tiempo.

Tiene el siguiente formato:

	1	2	3	4	5	6	7	8	9	10	11	12
\$GPRMC,	hhmmss.ss,	A,	1111.11,	A,	yyyy.yy,	a,	x.x,	x.x,	ddmmyy,	x.x,	A	*hh

Tabla 1: Formato de la sentencia \$GPRMC

Donde:

- 1 Hora en formato UTC (hhmmss.ss)
- 2 Bandera de validación de datos

A = datos válidos

B = datos almacenados

¹⁰ National Marine Electronics Association, es una asociación marina precursora de la comunicación entre equipos, estableciendo un protocolo normalizado de transmisión de datos en diversos equipos de electrónica naval, crearon el estándar NMEA 0183 para los dispositivos GPS.

V = datos inválidos

W = datos no almacenados

- 3 Latitud en grados, minutos y minutos decimales
- 4 Indicador N/S (Norte/Sur)
- 5 Longitud en grados, minutos y minutos decimales
- 6 Indicador E/W (Este/Oeste)
- 7 Velocidad
- 8 Orientación
- 9 Fecha en formato UTC (ddmmyy)
- 10 Variación magnética
- 11 Variación magnética
- 12 Suma de comprobación o checksum¹¹.

2.5.2 XMLHttpRequest

También referida como XMLHTTP (Extensible Markup Language / Hypertext Transfer Protocol), es una interfaz empleada para realizar peticiones HTTP y HTTPS a servidores Web. Para los datos transferidos se usa cualquier codificación basada en texto, incluyendo: texto plano, XML, JSON, HTML y codificaciones particulares específicas. La interfaz se implementa como una clase de la que una aplicación cliente puede generar tantas instancias como necesite para manejar el diálogo con el servidor.

Fuente: <http://www.pergaminovirtual.com.ar/definicion/> [RW011]

¹¹ Su propósito principal es detectar cambios accidentales en una secuencia de datos para proteger la integridad de datos, verificando que no haya discrepancias. La idea es que se transmita el dato junto con su valor hash, de esta forma el receptor puede calcular el valor hash de la secuencia recibida y la puede comparar con el valor hash recibido. Si hay una discrepancia se pueden rechazar los datos o pedir una retransmisión.

2.6 LENGUAJES DE PROGRAMACIÓN

2.6.1 UML

El lenguaje unificado de diagrama o notación (UML) sirve para especificar, visualizar y documentar esquemas de sistemas de software orientado a objetos. UML ayuda a visualizar el diseño y a hacerlo más accesible para otros. UML está controlado por el OMG¹² y es el estándar de descripción de esquemas de software.

UML está diseñado para su uso con software orientado a objetos, y tiene un uso limitado en otro tipo de cuestiones de programación.

UML se compone de muchos elementos de esquematización que representan las diferentes partes de un sistema de software. Los elementos UML se utilizan para crear diagramas, que representa alguna parte o punto de vista del sistema. Soporta los siguientes tipos de diagramas:

DIAGRAMA	OBJETIVO
Actividades	Muestra las actividades, así como los cambios de una a otra actividad junto con los eventos que ocurren en ciertas partes del sistema.
Clases	Muestra las clases, características y relaciones.
Colaboración	Muestra los objetos y sus relaciones, destacando los objetos que participan en el intercambio de mensajes.
Componentes	Muestra los componentes de mayor nivel de la programación.
Implementación	Muestra las instancias de los componentes y sus relaciones.
Objetos	Ejemplo de configuraciones de las instancias.
Paquetes	Estructura jerárquica en tiempo de compilación.
Secuencia	Muestra los objetos y sus múltiples relaciones entre ellos.
Máquinas de	Muestra estados, cambios de estado y eventos en un objeto

¹² Object Management Group

estado	o en parte del sistema.
Casos de uso	Muestra a los actores (otros usuarios del sistema), los casos de uso (las situaciones que se producen cuando utilizan el sistema) y sus relaciones.

Tabla 2: Tipos de diagramas oficiales de UML

2.6.2 JAVA

Java es un lenguaje de programación y la primera plataforma informática creada por Sun Microsystems en 1995. Es la tecnología subyacente que permite el uso de programas punteros, como herramientas, juegos y aplicaciones de negocios. Java se ejecuta en más de 850 millones de ordenadores personales de todo el mundo y en miles de millones de dispositivos, como dispositivos móviles y aparatos de televisión.

Existe un gran número de aplicaciones y sitios Web que no funcionan a menos que Java esté instalado, y muchas más que se crean a diario. Java es rápido, seguro y fiable. De portátiles a centros de datos, de consolas de juegos a súper equipos científicos, de teléfonos móviles a Internet, Java está en todas partes.

La última versión de Java contiene importantes mejoras para el rendimiento, estabilidad y seguridad de las aplicaciones Java que se ejecutan en su equipo. La instalación de la actualización gratuita garantiza que sus aplicaciones Java sigan ejecutándose de forma segura y eficaz.

2.6.2.1 FILOSOFÍA

El lenguaje Java se creó con cinco objetivos principales:

- Debería usar el paradigma de la programación orientada a objetos.
- Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.
- Debería incluir por defecto soporte para trabajo en red.
- Debería diseñarse para ejecutar código en sistemas remotos de forma segura.
- Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.

Su robustez y portabilidad han hecho de su entorno de funcionamiento bastante amplio, es posible programar en dispositivos móviles, navegadores web, sistemas de servidor y aplicaciones de escritorio.

2.6.3 HTML

Hyper Text Markup Language (Lenguaje de marcación de Hipertexto) es el lenguaje de marcas de texto utilizado normalmente en la www (World Wide Web). Fue creado en 1986 por el físico nuclear Tim Berners-Lee; el cual tomo dos herramientas preexistentes: El concepto de Hipertexto (Conocido también como link o ancla) el cual permite conectar dos elementos entre si y el SGML (Lenguaje Estándar de Marcación General) el cual sirve para colocar etiquetas o marcas en un texto que indique como debe verse. HTML no es propiamente un lenguaje de programación como C++, Visual Basic, etc., sino un sistema de etiquetas. HTML no presenta ningún compilador, por lo tanto algún error de sintaxis que se presente éste no lo detectará y se visualizara en la forma como éste lo entienda.

El entorno para trabajar HTML es simplemente un procesador de texto, como el que ofrecen los sistemas operativos Windows (Bloc de notas), UNIX (el editor vi o ed) o el que ofrece MS Office (Word). El conjunto de etiquetas que se creen, se deben guardar con la extensión .htm o .html

Estos documentos pueden ser mostrados por los visores o "browsers" de paginas Web en Internet, como Netscape Navigator, Mosaic, Opera y Microsoft Internet Explorer.

Fuente: <http://www.pergaminovirtual.com.ar/definicion/> [RW011]

2.6.4 JAVASCRIPT

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la

interfaz de usuario y páginas web dinámicas, en bases de datos locales al navegador, aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Fuente: <http://www.pergaminovirtual.com.ar/definicion/> [RW011]

2.6.5 CSS

Hojas de Estilo en Cascada (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla.

CSS se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los Estilos definen la forma de mostrar los elementos HTML y XML. CSS permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento. SQL

Fuente: <http://www.pergaminovirtual.com.ar/definicion/> [RW011]

2.6.6 SQL

Acrónimo de Structured Query Language (Lenguaje de Consulta Estructurado). Es un lenguaje especializado de programación que permite realizar consultas a base de datos.

Los orígenes del SQL están ligados a los de las bases de datos relacionales. En 1970 Dr. Edgar Frank Codd, investigador de IBM¹³, propone el modelo relacional y asociado a este un sublenguaje de acceso a los datos basado en el cálculo de predicados. Basándose en estas ideas los laboratorios de IBM definen el lenguaje SEQUEL (Structured English QUery Language) que más tarde sería ampliamente implementado por el SGBD experimental System R, desarrollado en 1977 también por IBM.

Es un lenguaje de programación interactivo y estandarizado para extraer información y actualizar una base de datos.

Una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar de forma sencilla información de interés de bases de datos, así como hacer cambios en ella.

Fuente: <http://www.pergaminovirtual.com.ar/definicion/> [RW011]

2.6.7 XML

Extensible Markup Language ('lenguaje de marcas extensible'), es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C). Deriva del lenguaje SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML) para estructurar documentos grandes. A diferencia de otros lenguajes, XML da soporte a bases de datos, siendo útil cuando varias aplicaciones se deben comunicar entre sí o integrar información. (Bases de datos Silberschatz).

¹³ International Business Machines, es una empresa multinacional estadounidense de tecnología y consultoría con sede en Armonk, Nueva York.

XML no ha nacido sólo para su aplicación para Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

Fuente: <http://www.pergaminovirtual.com.ar/definicion/> [RW011]

2.7 HERRAMIENTAS Y FRAMEWORKS

2.7.1 TOMCAT

Apache Tomcat funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat es un servidor web con soporte de servlets y JSPs. Tomcat no es un servidor de aplicaciones, como JBoss o JOnAS. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache.

Tomcat puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

2.7.1.1 ESTRUCTURA DE DIRECTORIOS

La jerarquía de directorios de instalación de Tomcat incluye:

- bin - arranque, cierre, y otros scripts y ejecutables.
- common - clases comunes que pueden utilizar Catalina y las aplicaciones web.
- conf - ficheros XML y los correspondientes DTD para la configuración de Tomcat.
- logs - logs de Catalina y de las aplicaciones.
- server - clases utilizadas solamente por Catalina.
- shared - clases compartidas por todas las aplicaciones web.
- webapps - directorio que contiene las aplicaciones web.
- work - almacenamiento temporal de ficheros y directorios.

2.7.2 DOM

El Document Object Model o DOM ('Modelo de Objetos del Documento' o 'Modelo en Objetos para la Representación de Documentos') es esencialmente una interfaz de programación de aplicaciones (API) que proporciona un conjunto estándar de objetos para representar documentos HTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos. A través del DOM, los programas pueden acceder y modificar el contenido, estructura y estilo de los documentos HTML y XML, que es para lo que se diseñó principalmente.

El DOM es una interfaz de programación de aplicaciones para acceder, añadir y cambiar dinámicamente contenido estructurado en documentos con lenguajes como ECMAScript (JavaScript).

Fuente: <http://www.pergaminovirtual.com.ar/definicion/> [RW011]

2.7.3 AJAX

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest¹⁴,

¹⁴ También referida como XMLHTTP (Extensible Markup Language / Hypertext Transfer Protocol), es una interfaz empleada para realizar peticiones HTTP y HTTPS a servidores Web. Para los datos transferidos se usa cualquier codificación basada en texto, incluyendo: texto plano, XML, JSON, HTML

objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores dado que está basado en estándares abiertos como JavaScript y Document Object Model (DOM).

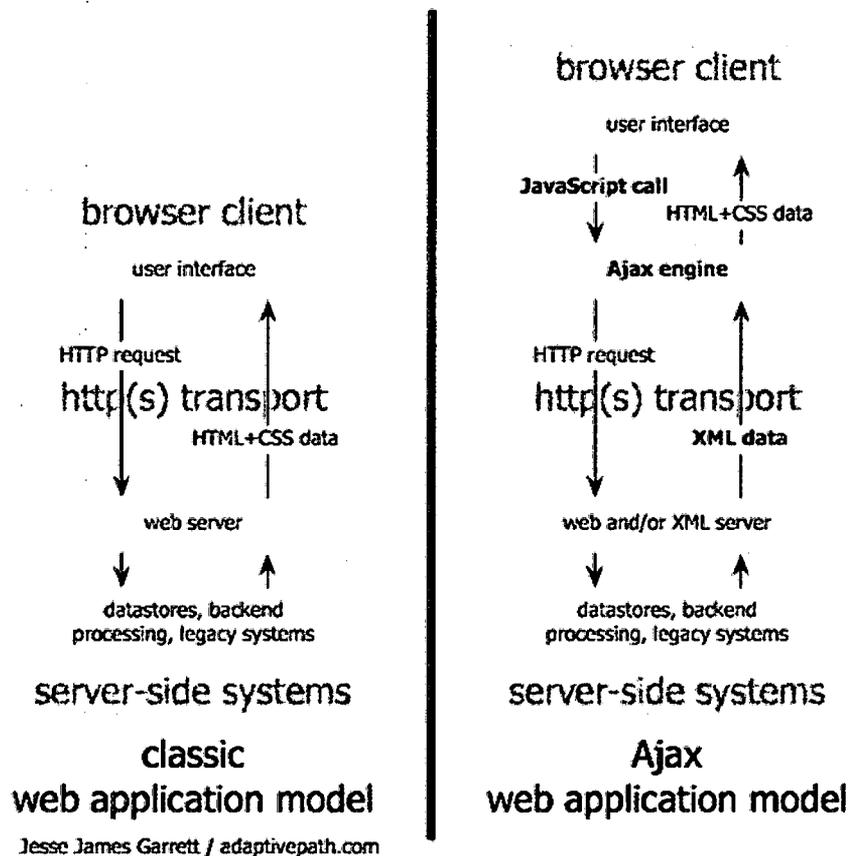


Ilustración 8: Diferencia de una aplicación web clásica y una que implementa Ajax

Fuente: <http://coding.smashingmagazine.com/2010/02/10/some-things-you-should-know-about-ajax/> [RW005]

y codificaciones particulares específicas. La interfaz se implementa como una clase de la que una aplicación cliente puede generar tantas instancias como necesite para manejar el diálogo con el servidor.

2.7.4 GOOGLE WEB TOOLKIT

GWT o Google Web Toolkit es un framework creado por Google que permite ocultar la complejidad de varios aspectos de la tecnología AJAX. Es compatible con varios navegadores, lo cual es notorio ya que cada navegador suele necesitar código específico para lograr un front-end correcto en una aplicación web. El concepto de Google Web Toolkit es bastante sencillo, básicamente lo que se debe hacer es crear el código en Java usando cualquier IDE de Java y el compilador lo traducirá a HTML y JavaScript.

2.7.4.1 DESARROLLO

Con la biblioteca GWT, los desarrolladores pueden crear y depurar aplicaciones AJAX en lenguaje JAVA usando el entorno de desarrollo que prefieran¹⁵. Cuando una aplicación es desplegada, el compilador GWT traduce la aplicación Java a un archivo Javascript, que puede ser ofuscado para optimizar el rendimiento.

GWT no es sólo una interfaz de programación, sino que además proporciona un conjunto de herramientas que permiten desarrollar funcionalidades Javascript de alto rendimiento en el navegador del cliente.

Una aplicación GWT puede ser ejecutada en dos modos:

- Modo host (Hosted mode): La aplicación se ejecuta como código bytecode de Java dentro de la Máquina Virtual de Java (JVM). Este modo es el más usado para desarrollo, soportando el cambio de código en caliente y el depurado.
- Modo web (Web mode): La aplicación se ejecuta como código Javascript y HTML puro, compilado a partir del código Java. Este modo se suele usar para el despliegue de la aplicación.

¹⁵ Existen varios plugins de código abierto para desarrollar en diferentes entornos de desarrollo, como GWT4NB para NetBeans, Cypal Studio for GWT para Eclipse o gwtDeveloper para Jdeveloper.

La utilidad de línea de comandos `applicationCreator` genera automáticamente todos los archivos necesarios para iniciar un proyecto GWT, incluso permite crear un proyecto para Eclipse.

2.7.4.2 ARQUITECTURA GWT

GWT contiene los siguientes componentes:

- **GWT Java-to-JavaScript Compiler:** la función de este componente es traducir el código desarrollado en Java al lenguaje JavaScript. Lo empleamos cuando usamos al GWT en modo web.
- **Hosted Web Browser:** Este componente ejecuta la aplicación Java sin traducirla a JavaScript, en modo host usando la máquina virtual de Java.
- **JRE Emulation Library:** Contiene las bibliotecas más importantes de las clases de Java: `java.lang` en donde se encuentran las clases fundamentales para poder programar en Java y un subconjunto de las clases del paquete `java.util`. `Java.lang` incluye, entre otras, la clase `java.lang.object` que es la clase fundamental de la que heredan o extienden todas las clases en Java. El resto de los paquetes no están soportados por GWT.
- **GWT Web UI Class Library:** Contiene un conjunto de elementos de interfaz de usuario que permite la creación de objetos tales como textos, cajas de texto, imágenes y botones.

2.7.4.3 CARACTERÍSTICAS

- **Componentes de la interfaz de usuario dinámicos y reusables:** los programadores pueden usar clases prediseñadas para implementar comportamientos que de otra manera consumirían mucho tiempo, como arrastrar y soltar o menús en árbol.
- **Simple mecanismo RPC:** Para comunicarse desde el navegador con el servidor web, solamente se necesita definir clases de Java serializables para las peticiones y respuestas. En tiempo de ejecución GWT serializa

automáticamente las peticiones del navegador y de-serializa las respuestas desde el servidor web.

- Gestión del historial del navegador web: GWT al igual que Ajax no necesitan un botón “atrás” del navegador. Por ello no es necesario que se llame a otras páginas para realizar las diferentes acciones, ni recargar el navegador.
- Soporte para depurado de Java: En la fase de desarrollo se tienen la posibilidad de depurar la aplicación con los avanzados sistemas debugging y manipulación de excepciones incluidos en IDEs como Eclipse.
- Compatibilidad con los navegadores.
- Integración con JUnit: lo que permite probar las aplicaciones y depurarlas en un navegador en tiempo de desarrollo.
- Internacionalización
- Los desarrolladores pueden mezclar código escrito en Javascript dentro del código Java usando la Interfaz Nativa Javascript (JSNI).
- Soporte para la API de Google (inicialmente, soporte para Google Gears).
- Es de código abierto.
- Los desarrolladores pueden diseñar y desarrollar sus aplicaciones orientadas a objetos.
- El código Javascript generado puede ser ofuscado para optimizar el rendimiento.
- Existen un numeroso conjunto de bibliotecas desarrolladas por Google y terceros que amplían las funcionalidades de GWT.

2.7.5 GOOGLE MAPS

Google Maps es un servicio de Google que ofrece imágenes vía satélite de todo el planeta, combinadas, en el caso de algunos países, con mapas de sus ciudades, lo que unido a sus posibilidades de programación abierta ha dado lugar a diversas utilidades ofrecidas desde numerosas páginas web.

Desde su lanzamiento en febrero de 2005, la aplicación cartográfica de Google ha conmocionado a la comunidad de desarrolladores. Si bien sus principios técnicos de base eran ya conocidos, incluso utilizados desde hacía tiempo, la aplicación de Google los combina de manera inteligente, y sobre todo ofrece una accesibilidad sin igual.

2.7.5.1 FUNCIONAMIENTO

Esta tecnología es la implementación conjunta de HTML, CSS y JavaScript. Los mapas son solo imágenes que se cargan en el fondo a través de peticiones ejecutadas por la tecnología de AJAX, y se insertan en un <div> en la página HTML. Mientras el usuario navega en el mapa, el API envía información acerca de las nuevas coordenadas y los niveles de “zoom” del mapa a través de AJAX y esto retorna las imágenes.

Varios aspectos de Google Maps son los responsables de su facilidad de uso por cualquier usuario: el sistema de deslizamiento de imagen, acoplado a la carga dinámica de nuevas imágenes; la adaptación del mapa al tamaño de ventana del navegador; la interfaz minimalista; la posibilidad de cambiar de tipo de mapa en un clic...

Según el nivel de zoom, un mapa podría descomponerse en varias decenas de miles de cuadrados. Esta descomposición se realiza automáticamente por un script del lado servidor. Cuando el usuario hace deslizar la imagen en un sentido, JavaScript calcula cuales son los cuadrados involucrados, envía al servidor una pregunta sobre las imágenes en cuestión, y las coloca alineadas junto a las otras.

Cada cuadrado es almacenado en un fichero cuyo nombre indica su longitud, su latitud, y el valor del zoom.

Todo ello necesita, por tanto, una sincronización perfecta entre lo que puede preguntar el código JavaScript, y lo que está efectivamente disponible en el servidor. El programa de recorte inicial de la imagen completa es por tanto extremadamente importante, porque es el responsable, para la precisión del corte y el rigor de su etiquetamiento, del buen desenvolvimiento de los acontecimientos desencadenados por las acciones del usuario.

El API consiste de archivos JavaScript, que contienen las clases, métodos y propiedades, todos ellos muy bien establecidos y documentados, facilitando así la tarea del desarrollador de la aplicación web.

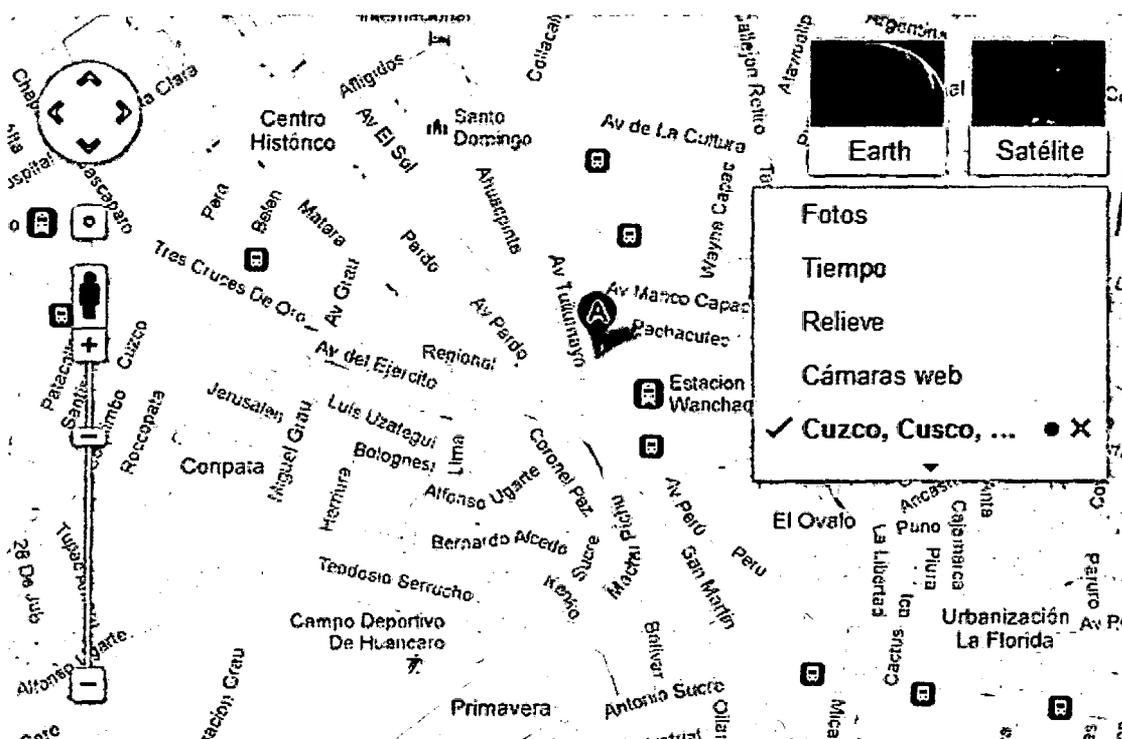


Ilustración 9: Google Maps

Fuente: <https://maps.google.com.pe/> [RW007]

2.7.5.2 SERVICIOS QUE OFRECE

Éstos son algunos de los servicios que ofrece Google Maps:

- Solicitudes de rutas, caminando, conduciendo, tomando transporte público, bicicleta.
- Cálculo de distancias.
- Cálculo de tiempo que tomaría llevar una ruta.
- Solicitudes de coordenadas mediante direcciones.
- Solicitudes de direcciones mediante coordenadas.
- Visualización panorámica de las calles¹⁶.

2.7.6 MySQL

Es un sistema de gestión de bases de datos relacionales multihilo y multiusuario que se caracteriza por su facilidad de uso. Se adapta bien a la administración de datos en un entorno de red, especialmente en arquitecturas cliente/servidor. Se proporciona con muchas herramientas y es compatible con muchos lenguajes de programación. Es uno de los mejores gestores de base de datos Open Source, gracias a su compatibilidad con el servidor de páginas web Apache y su fácil implementación con tecnologías como son PHP.

2.7.6.1 CARACTERÍSTICAS TÉCNICAS

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferente velocidad de operación, soporte físico, capacidad, distribución geográfica.
- Transacciones y claves foráneas.
- Conectividad segura.

¹⁶ Algunos de los servicios de Google maps, como es el caso de éste, no se dan por el momento en la ciudad del Cusco, pero Google está incluyendo con rapidez cada vez a más lugares.

- Replicación.
- Búsqueda e indexación de campos de texto.

2.7.7 MySQL WORKBENCH

MySQL Workbench es una herramienta visual de diseño de bases de datos que integra desarrollo de software, Administración de bases de datos, diseño de bases de datos, creación y mantenimiento para el sistema de base de datos MySQL.

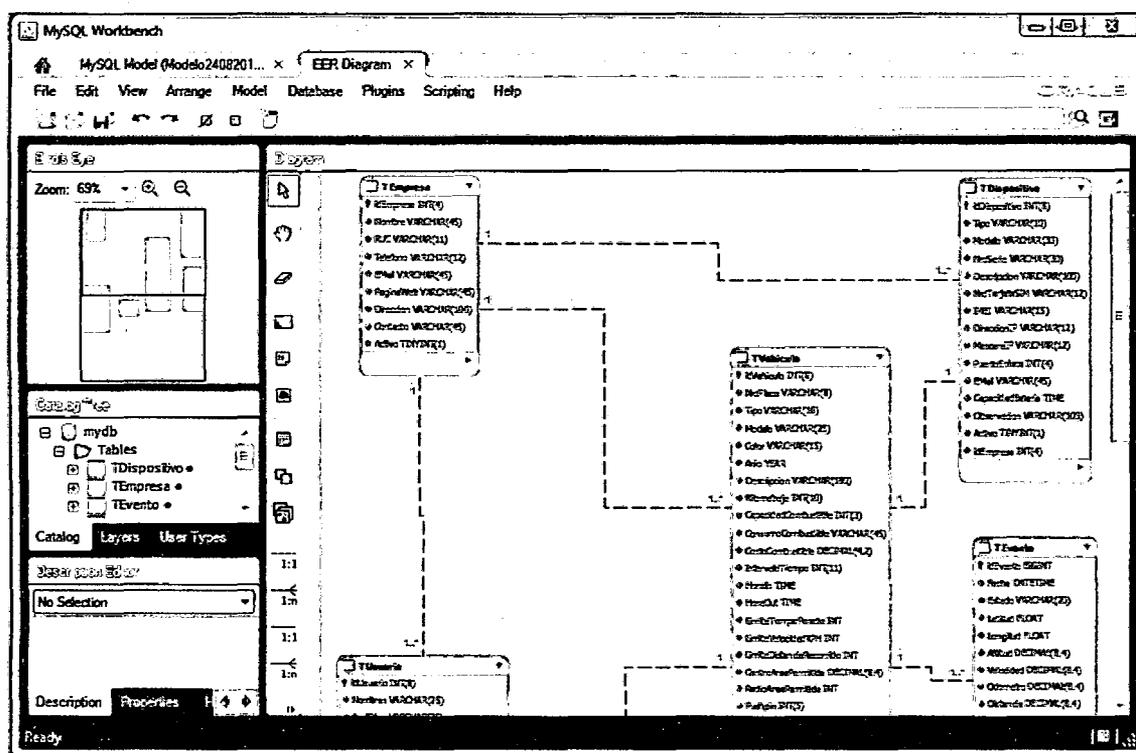


Ilustración 10: Interfaz de Mysql Workbench

2.7.8 ECLIPSE

Eclipse es un programa informático compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el

compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse).

Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

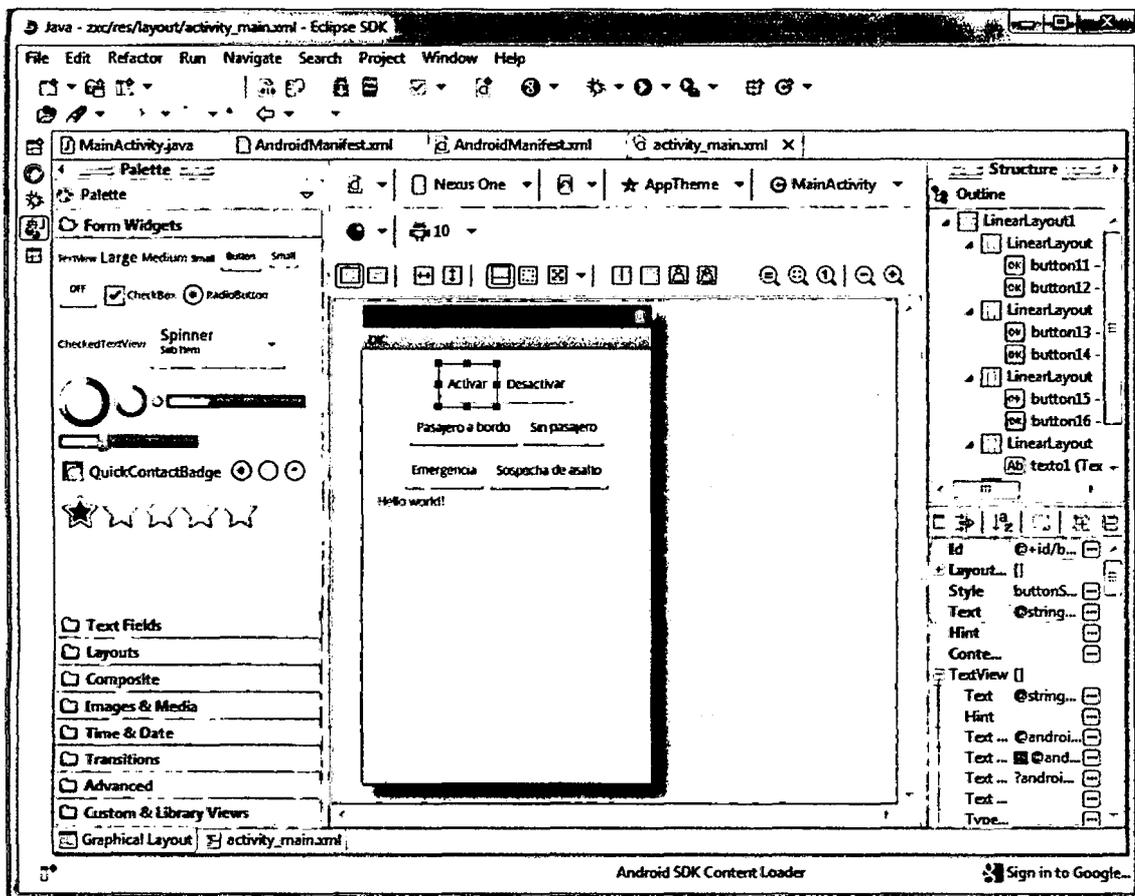


Ilustración 11: Interfaz de Eclipse

CAPÍTULO 3. FASE DE INICIO

Hito principal: Establecer y garantizar una buena base para el desarrollo de las posteriores fases.

Los objetivos de esta fase son:

- ✓ Determinar el ámbito del sistema
- ✓ Delimitar el alcance del sistema
- ✓ Definir una arquitectura posible para el sistema.
- ✓ Determinar los riesgos críticos y dar alternativas de solución para reducirlos.

3.1 DESCRIPCIÓN DEL PROBLEMA REAL

La empresa “Turismo Ángeles del Cusco SAC” adolece de varios problemas, entre ellos, fundamentalmente, para monitorear se debe confiar en la información proporcionada por el conductor a través del uso de radio, que genera problemas que afectan a la productividad y a la seguridad.

En lo referente a la seguridad, actualmente no es posible la identificación de los conductores que atentan contra las normas de tránsito, poniendo en cuestionamiento no sólo su propia imagen, sino también la imagen de la empresa.

En lo referente a la productividad, los operadores actualmente no pueden determinar la ubicación de sus unidades vehiculares, lo que conlleva a los conductores a la posibilidad de aprovechar a su propia conveniencia los recursos puestos a su disposición, afectando a la economía de la organización.

Otro problema que suele pasar por desconocer la ubicación de los vehículos, se debe a la falta de puntualidad del conductor, esto se aplica en el caso de que un cliente solicite el servicio de taxi con horas o días de anticipación. El conductor incluso, aprovechándose de la falta de control de los que adolece la empresa, podría culpar al cliente, alegando que este último fue el que no estuvo a la hora acordada. De hecho no existe forma de corroborar si el conductor fue impuntual o lo fue el cliente, como resultado la empresa en su conjunto pierde fiabilidad.

Actualmente en el Cusco, ninguna empresa operadora de taxis implementa una solución de rastreo satelital. La implementación de este sistema proporcionaría una ventaja competitiva notable para la empresa en cuestión, puesto que proporcionaría a sus clientes puntualidad, seguridad y sobre todo un buen servicio.

3.2 DESCRIPCIÓN DE LA EMPRESA

La empresa “Turismo Ángeles del Cusco SAC” se dedica a ofrecer servicios de transporte a personas particulares y empresas en la ciudad del Cusco.

Actualmente, atiende a clientes corporativos, como son agencias de turismo, empresas de organización de eventos, y a personas particulares, ofreciéndoles confort, puntualidad y sobre todo la garantía de llegar salvos a su lugar de destino.

Inició sus actividades el 15 de diciembre del 2010 y actualmente cuenta con una flota de 5 unidades vehiculares propias de la empresa y 40 unidades que pertenecen a conductores asociados que forman parte de la empresa.

La forma de rastrear los vehículos es mediante aparatos de radio, y su proceso de negocio se expone a continuación.

3.3 PROCESO DEL NEGOCIO

➤ Operar vehículos

Un cliente solicita un vehículo en una determinada dirección, la operadora le indica deja en espera al cliente para consultar la disponibilidad de los vehículos llamando por radio, mientras tanto el cliente puede cansarse de la espera y simplemente cortar la llamada.

En el siguiente paso, hay dos opciones:

- La primera, es que efectivamente esté cerca una unidad vehicular. En este caso el conductor se reporta mediante clave por radio, y se dirige a la dirección que la operadora le indicó. La operadora comunica al cliente que su vehículo ya está llegando.
- La segunda, es que ningún vehículo esté a disposición ni cerca de la zona en cuestión, la operadora informa al cliente, este último se queda frustrado por el hecho de que su espera fue vana.

➤ Brindar seguridad al conductor

Ante un incidente del tipo robo, sospecha de asalto, accidente u otra emergencia, el conductor notifica del hecho mediante una llamada de radio, entonces las demás unidades vehiculares tienen el deber de acudir al conductor implicado.

Los delincuentes conocen de este sistema de seguridad que implementan las operadoras de taxis, por lo que prefieren victimar a conductores independientes.

➤ Brindar seguridad y puntualidad al pasajero

Con el incremento de los índices delictivos de los que adolece actualmente la población del Cusco, las personas prefieren solicitar servicios de las operadoras de taxis, aunque tengan que pagar más por el servicio. La empresa en cuestión brinda a sus clientes la seguridad de llegar a salvos a su destino.

En segundo lugar ante cualquier incidente, del cual no está libre nadie por mucho que no infrinja las normas, la operadora de taxis ofrece la asistencia oportuna a sus pasajeros, puesto que los demás conductores tienen el deber de acudir al conductor y pasajero que resulten implicados.

Si un pasajero solicita los servicios para un día particular y hora particular, la empresa dispone de una unidad vehicular para que le brinde el servicio con puntualidad.

Lamentablemente todo lo anteriormente descrito, es cuestionable, puesto que el operador confía en la información que el conductor brinda, y dicha información resulta imposible de confirmar al no contar con un sistema de rastreo, propiciando así la informalidad y el desorden.

3.4 DIAGRAMA BPMN DEL PROCESO DEL NEGOCIO

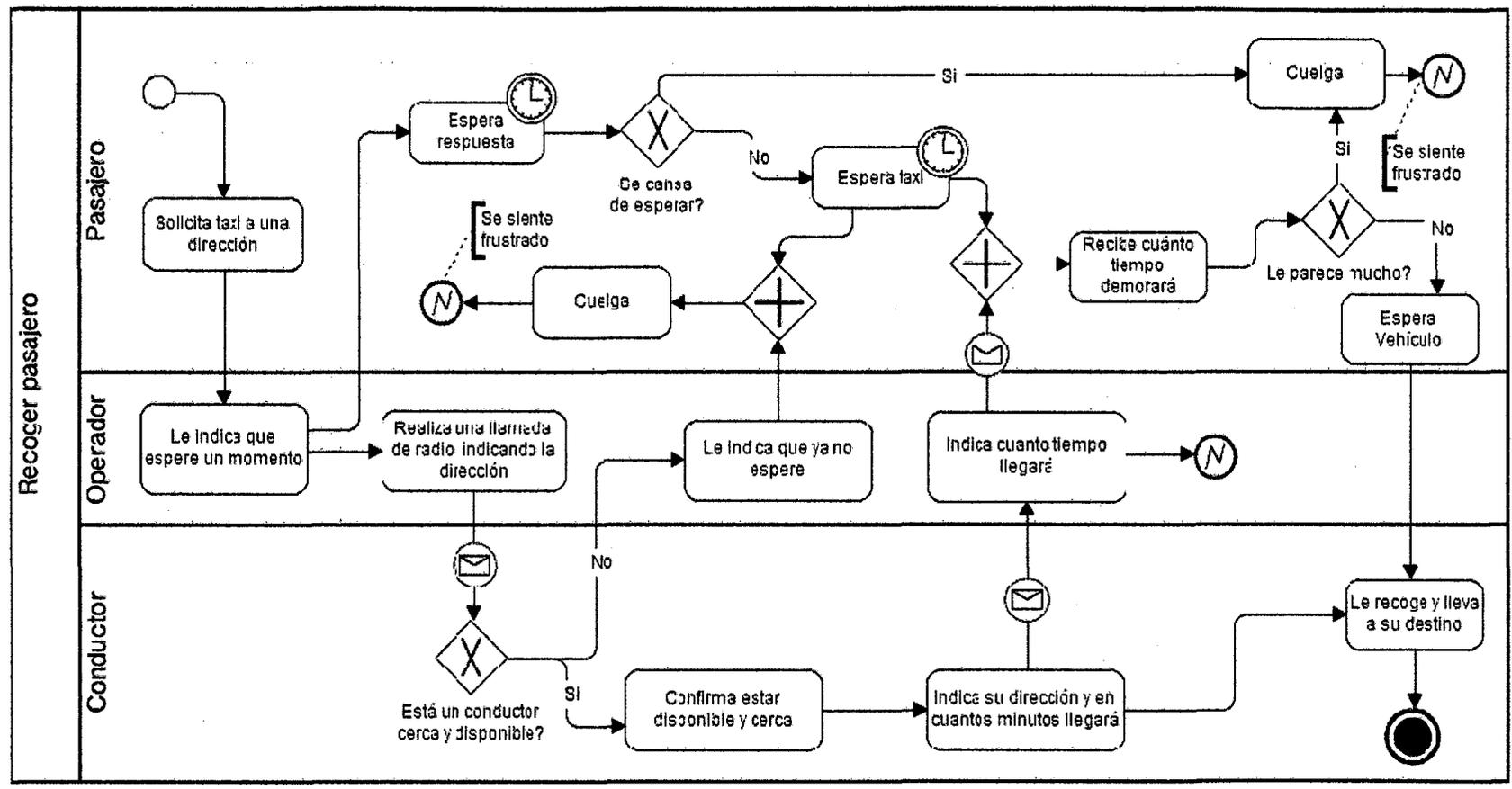


Ilustración 12: Diagrama BPMN del proceso del negocio

3.5 MODELO DE NEGOCIO

El primer paso del proyecto es entender la estructura funcional de la empresa, es decir cómo se desenvuelve tanto en su interior como exteriormente.

3.5.1 PAQUETES DE CASO DE USO DEL NEGOCIO



Ilustración 13: Diagrama de Paquetes del Negocio

3.5.2 UNIDADES ORGANIZACIONALES

La empresa en estudio cuenta con las siguientes áreas:



Ilustración 14: Estructura organizacional del negocio

3.5.3 IDENTIFICACIÓN DE LOS ACTORES DEL NEGOCIO

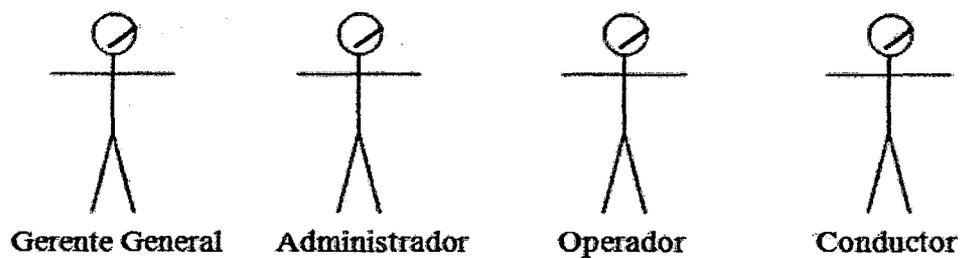


Ilustración 15: Actores del negocio

- **Gerente General.** Persona encargada de la empresa.
- **Administrador.** Persona encargada de administrar y controlar al personal, cumple también las labores de un operador si se requiere.
- **Operador.** Persona que se encarga del seguimiento vehicular exclusivamente.
- **Conductor.** Conductor de la unidad vehicular.

3.5.4 DIAGRAMA DE CASOS DE USO DEL NEGOCIO

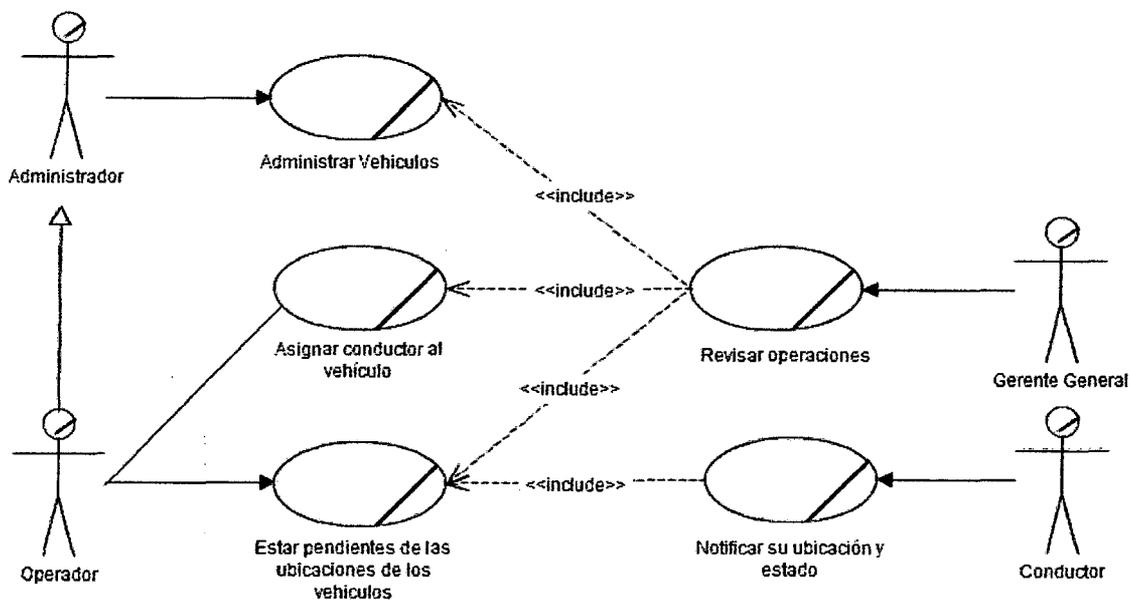


Ilustración 16: Diagrama de casos de uso del negocio

3.5.5 DESCRIPCIÓN DE LOS CASOS DE USO DEL NEGOCIO

CASO DE USO	DESCRIPCIÓN	ENCARGADO
Revisar operaciones.	Revisar las operaciones que realiza el personal.	Gerente General
Administrar vehículos.	Administrar las unidades vehiculares, observando que no haya daños.	Administrador
Estar pendiente de las ubicaciones de los vehículos.	Estar pendientes de las posiciones de los vehículos, contactando con los conductores para que reporten su ubicación.	Operador
Asignar conductor al vehículo.	Estar al corriente de los vehículos y los conductores que los conducen.	Operador
Notificar su ubicación y estado.	Notificar de su posición y de su estado. Estado se refiere a: si se encuentra en servicio, si está con pasajero o se encuentra en una emergencia.	Conductor

Tabla 3: Descripción de los casos de uso del negocio

3.5.6 REALIZACIÓN DE LOS CASOS DE USO DEL NEGOCIO

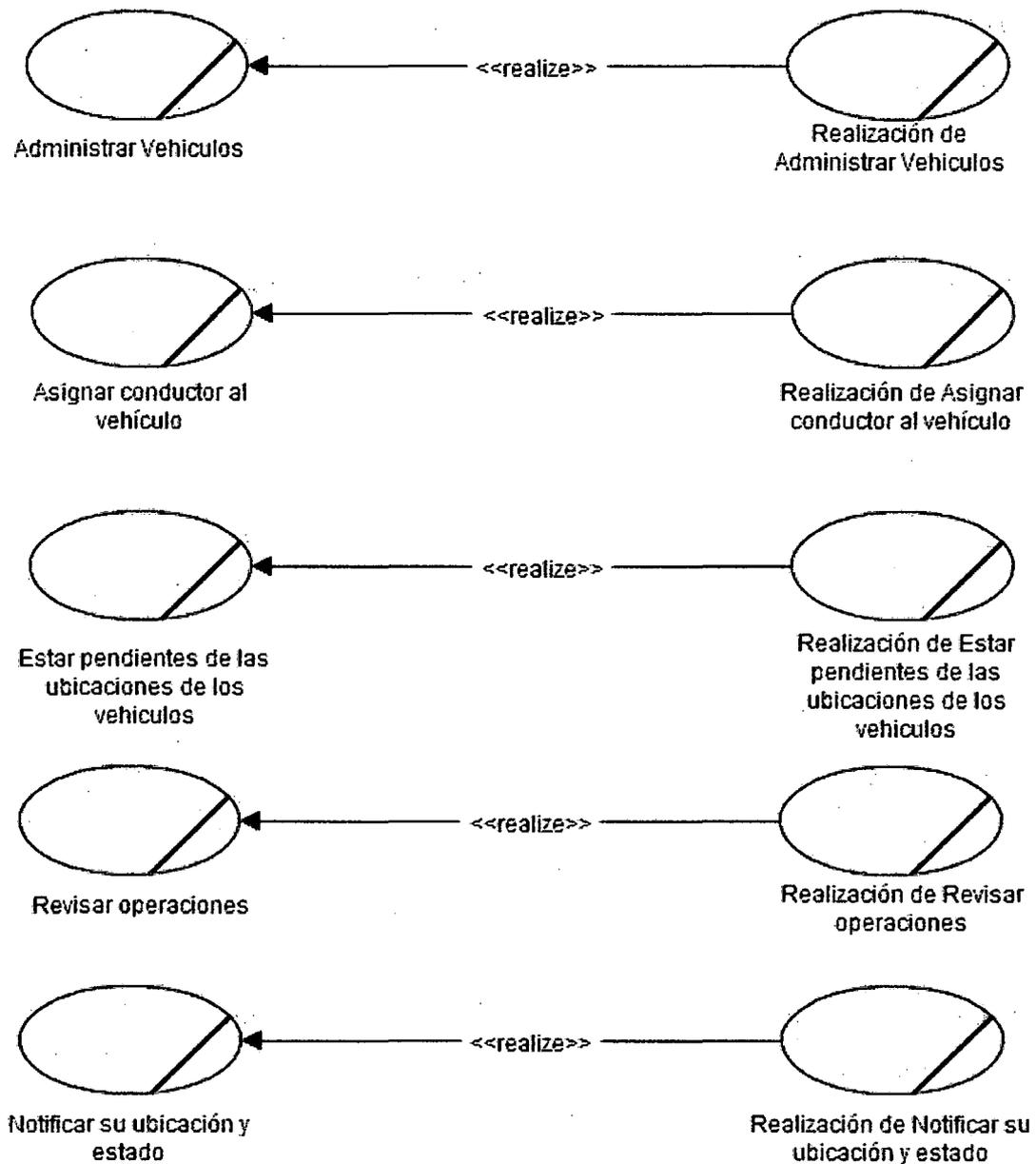


Ilustración 17: Diagrama de realización de casos de uso

3.5.7 LISTA DE REQUERIMIENTOS

3.5.7.1 REQUISITOS FUNCIONALES

CÓDIGO	DESCRIPCIÓN DEL REQUISITO	ENCARGADO	PRIORIDAD
R1	Mantener al corriente de las infracciones cometidas por los conductores.	Administrador	Alta
R2	Permitir asignar un conductor a los vehículos.	Administrador	Alta
R3	En un mapa rastrear los vehículos en tiempo real.	Operador	Alta
R4	En un mapa visualizar las rutas seguidas por los vehículos en una determinada fecha y hora.	Operador	Alta
R5	Permitir al conductor notificar si tiene o no pasajero.	Conductor	Alta
R6	Permitir al conductor notificar si está o no está de servicio.	Conductor	Alta
R7	Permitir al conductor notificar si tiene alguna emergencia.	Conductor	Alta

Tabla 4: Requisitos funcionales

3.5.7.2 REQUISITOS NO FUNCIONALES

- Proporcionar rapidez, logrando tal con un uso eficiente de memoria.
- Proporcionar una interfaz amigable y predecible para la adaptación pronta de los usuarios.
- Fiabilidad en los datos.
- Portabilidad en su máxima expresión, con el simple hecho de tener un navegador y su respectivo plugin de java.
- El sistema debe ser escalable para futuras implementaciones.

3.6 CASOS DE USO DEL SISTEMA

CU1. Configurar parámetros límite.

CU2. Asignar dispositivo GPS a vehículo

CU3. Asignar conductor a vehículo.

CU4. Rastrear vehículos en tiempo real.

CU5. Mostrar rutas seguidas por los vehículos.

CU6. Emitir reportes.

CU7. Cambiar estado de vehículo.

3.6.1 IDENTIFICACIÓN DE LOS ACTORES

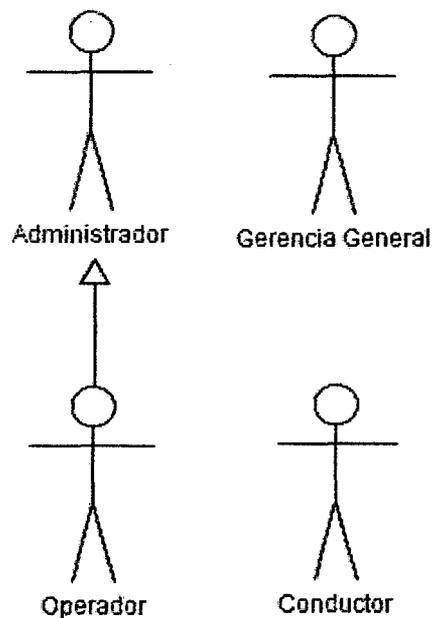


Ilustración 18: Actores del sistema

3.6.2 DIAGRAMA DE CASOS DE USO DEL SISTEMA

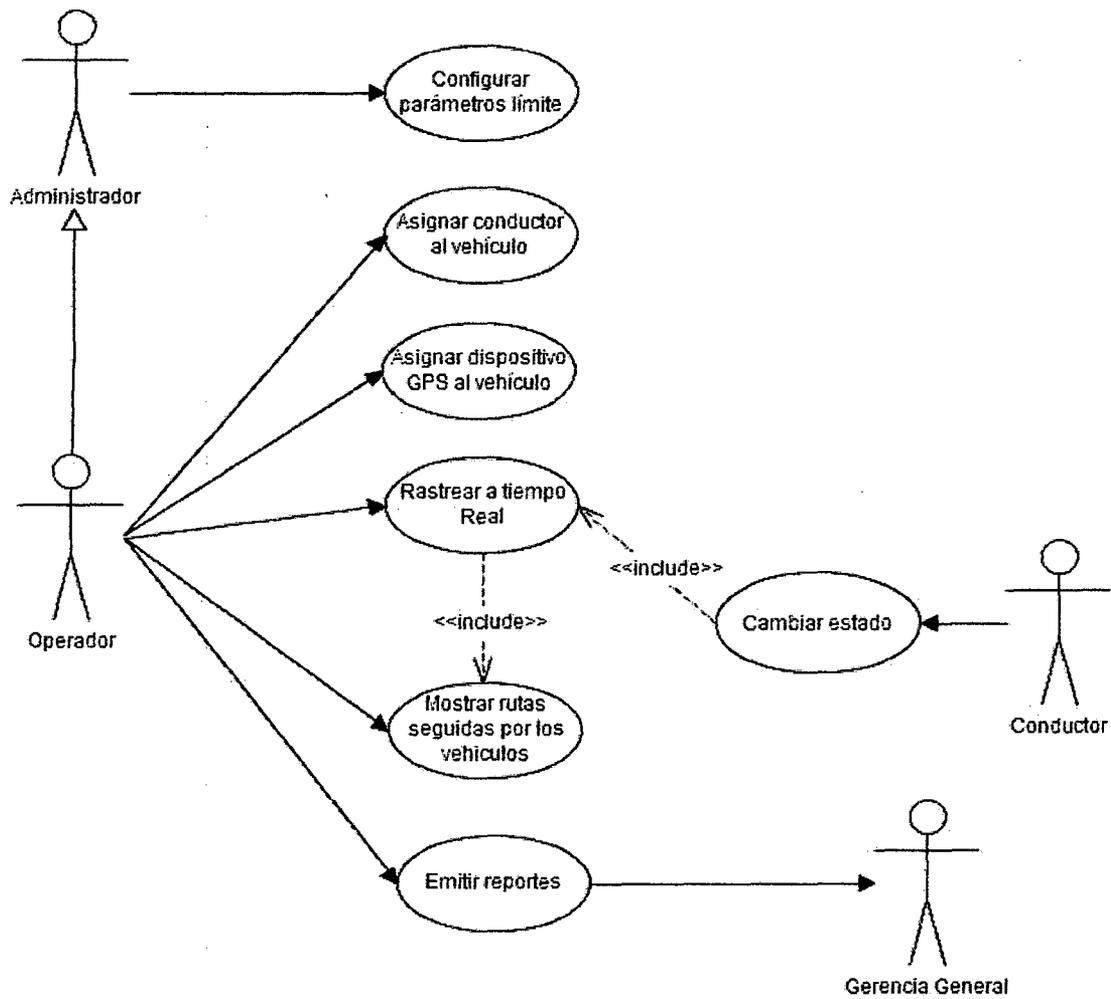


Ilustración 19: Diagrama de casos de uso

Tabla 5: Matriz de trazabilidad

		CASOS DE USO						
		CU1. Configurar parámetros límite.	CU2. Asignar dispositivo GPS a vehículo.	CU3. Asignar conductor a vehículo.	CU4. Rastrear vehículos en tiempo real.	CU5. Mostrar rutas seguidas por los vehículos.	CU6. Emitir reportes.	CU7. Cambiar estado de vehículo.
R E Q U E R I M I E N T O S	R1. Mantener al corriente de las infracciones cometidas por los conductores.	X					X	
	R2. Permitir asignar un conductor a los vehículos.			X				
	R3. En un mapa rastrear los vehículos en tiempo real.		X		X			
	R4. En un mapa visualizar las rutas seguidas por los vehículos en una determinada fecha y hora.		X	X		X		
	R5. Permitir al conductor notificar si tiene o no pasajero.					X		X
	R6. Permitir al conductor notificar si está o no está de servicio.					X		X
	R7. Permitir al conductor notificar si tiene alguna emergencia o sospecha de asalto.					X		X

3.6.3 MATRIZ DE TRAZABILIDAD

3.6.4 CASOS DE USO DE ALTO NIVEL

CASO DE USO	CU1. Configurar parámetros límite
ACTORES	Administrador
PROPOSITO	Configurar los parámetros que no deben transgredir los conductores.
DESCRIPCIÓN	Configurar los parámetros que no deben transgredir los conductores, como son velocidad máxima, tiempo de parada máxima y máxima distancia que puede recorrer un vehículo al día.
REFERENCIAS CRUZADAS	Requisito 1.

Tabla 6: Caso de uso de alto nivel "Configurar parámetros límite"

CASO DE USO	CU2. Asignar dispositivo GPS a vehículo.
ACTORES	Administrador
PROPOSITO	Asignar un dispositivo GPS a un vehículo.
DESCRIPCIÓN	Asigna el dispositivo GPS a un vehículo, dicho dispositivo no debe estar asignado a otro vehículo, de ser el caso debería desasignar el dispositivo del vehículo que lo tiene y luego asignarlo al vehículo deseado.
REFERENCIAS CRUZADAS	Requisito 3 y 4.

Tabla 7: Caso de uso de alto nivel "Asignar dispositivo GPS a vehículo"

CASO DE USO	CU3. Asignar conductor a vehículo
ACTORES	Administrador
PROPOSITO	Asignar un conductor a vehículo.
DESCRIPCIÓN	Asigna un conductor a un vehículo, dicho conductor no debe estar asignado a otro vehículo, de ser el caso debería desasignar el conductor del vehículo que lo tiene y luego asignarlo al vehículo deseado.
REFERENCIAS CRUZADAS	Requisito 2 y 4.

Tabla 8: Caso de uso de alto nivel "Asignar conductor a vehículo"

CASO DE USO	CU4. Rastrear vehículos en tiempo real
ACTORES	Operador
PROPOSITO	Rastrear los vehículos que están de servicio en tiempo real.
DESCRIPCIÓN	El operador, visualiza en un mapa de la interfaz las posiciones de los vehículos que están de servicio, observa en tiempo real su posición, orientación, si tiene pasajero, y si tiene una emergencia. En una tabla de la misma interfaz, muestra la velocidad, el tiempo de parada y la dirección por cada vehículo.
REFERENCIAS CRUZADAS	Requisito 3, 5, 6 y 7.

Tabla 9: Caso de uso de alto nivel "Rastrear vehículos en tiempo real"

CASO DE USO	CU5. Mostrar rutas seguidas por los vehículos
ACTORES	Operador
PROPOSITO	Visualizar las rutas seguidas por los vehículos en una determinada fecha y hora.
DESCRIPCIÓN	El operador, visualiza en un mapa de la interfaz las posiciones de los vehículos seleccionados en una determinada fecha y hora. Se muestra su posición en el mapa de la interfaz respectiva. En una tabla de la misma interfaz, se muestra el detalle, como es la hora, la velocidad, el tiempo de parada y dirección de la calle.
REFERENCIAS CRUZADAS	Requisito 4.

Tabla 10: Caso de uso de alto nivel "Mostrar rutas seguidas por los vehículos"

CASO DE USO	CU6. Consultar reportes
ACTORES	Gerente General y Administrador
PROPOSITO	Consultar reportes acerca de las infracciones cometidas por el usuario.
DESCRIPCIÓN	Realizar consultas al sistema sobre las infracciones cometidas por los las unidades vehiculares de manera resumida y a la vez detallada respecto a un proceso del negocio cuando sea necesario.
REFERENCIAS CRUZADAS	Requisito 1.

Tabla 11: Caso de uso de alto nivel " Consultar reportes"

CASO DE USO	CU7. Cambiar estado de vehículo
ACTORES	Conductor
PROPOSITO	El conductor informa su estado a la estación central.
DESCRIPCIÓN	El conductor mientras está de servicio, desde el dispositivo móvil integrado en su vehículo, pone en conocimiento de los operadores de sus estados: "En servicio/Fuera de servicio", "Con pasajero/Sin pasajero", "Emergencia" y "Fuera de riesgo".
REFERENCIAS CRUZADAS	Requisito 5, 6 y 7.

Tabla 12: Caso de uso de alto nivel "Cambiar estado de vehículo"

3.6.5 CASOS DE USO EXPANDIDO

CASO DE USO	CU1. Configurar parámetros límite	
ACTORES	Administrador	
OBJETIVO	Configurar los parámetros que no deben transgredir los conductores.	
DESCRIPCIÓN	El administrador configura los parámetros que no deben transgredir los conductores, como son velocidad máxima, tiempo de parada máxima y máxima distancia que puede recorrer un vehículo al día.	
DESCRIPCIÓN INICIAL PASO A PASO		
Pre condición: Solicitar datos y validarlos.		
ACTOR	SISTEMA	
1. Selecciona la opción de configurar parámetros límite.	2. Muestra los parámetros antes establecidos, si no existen, muestra unos parámetros por defecto.	
3. El usuario procede a modificar y guarda los cambios.	4. Guarda la configuración ingresada por el usuario.	
Post Condición: las configuraciones realizadas se tomarán en cuenta para las futuras consultas de excesos y transgresiones.		
Cursos Alternos: Si el usuario no ingresa los datos correctamente se muestra un mensaje de error, para que vuelva a intentarlo.		

Tabla 13: Caso de uso expandido "Configurar parámetros límite"

CASO DE USO	CU2. Asignar dispositivo GPS a vehículo.	
ACTORES	Administrador	
OBJETIVO	Asignar un dispositivo GPS a un vehículo.	
DESCRIPCIÓN	El administrador asigna un dispositivo GPS a un vehículo.	
DESCRIPCIÓN INICIAL PASO A PASO		
Pre condición:		
- Mostrar lista de vehículos.		

- Mostrar lista de dispositivos GPS.	
ACTOR	SISTEMA
1. Selecciona la opción de asignar dispositivo GPS a vehículo.	2. Muestra una lista de vehículos.
3. Selecciona un vehículo.	4. Muestra el vehículo seleccionado, y muestra la lista de dispositivos GPS que no están asignadas a ningún vehículo.
5. Selecciona un dispositivo y acepta la asignación.	6. Ejecuta la operación de asignación.
Post Condición: Asignación realizada.	
Cursos Alternos: Si el dispositivo está asignado a otro vehículo, el usuario debe primero desasignar el dispositivo del vehículo que lo tiene y luego asignarlo al vehículo deseado.	

Tabla 14: Caso de uso expandido "Asignar dispositivo GPS a vehículo"

CASO DE USO	CU3. Asignar conductor a vehículo
ACTORES	Administrador
OBJETIVO	Asignar un conductor a un vehículo.
DESCRIPCIÓN	El administrador asigna un conductor a un vehículo.
DESCRIPCIÓN INICIAL PASO A PASO	
Pre condición:	
<ul style="list-style-type: none"> - Mostrar lista de vehículos. - Mostrar lista de conductores. 	
ACTOR	SISTEMA
1. Selecciona la opción de asignar conductor a vehículo.	2. Muestra una lista de vehículos.
3. Selecciona un vehículo.	4. Muestra el vehículo seleccionado, y muestra la lista de conductores que no están asignados a ningún vehículo.
5. Selecciona un conductor y acepta	6. Ejecuta la operación de asignación.

la asignación.	
Post Condición: Asignación realizada.	
Cursos Alternos: Si el conductor está asignado a otro vehículo, el usuario debe primero desasignar el conductor del vehículo que lo tiene y luego asignarlo al vehículo deseado.	

Tabla 15: Caso de uso expandido "Asignar conductor a vehículo"

CASO DE USO	CU4. Rastrear vehículos en tiempo real	
ACTORES	Operador	
OBJETIVO	Rastrear los vehículos que están de servicio en tiempo real.	
DESCRIPCIÓN	El operador, visualiza en un mapa de la interfaz las posiciones de los vehículos que están de servicio, observa en tiempo real su posición, orientación, si tiene pasajero, y si tiene una emergencia. En una tabla de la misma interfaz, se muestra la velocidad, el tiempo de parada y la dirección por cada vehículo.	
DESCRIPCIÓN INICIAL PASO A PASO		
Pre condición: Obtener los vehículos que están de servicio.		
ACTOR	SISTEMA	
1. Selecciona la opción de visualizar vehículos en tiempo real.	2. Muestra la lista de vehículos que están de servicio.	
3. Selecciona de la lista de vehículos las unidades que desea rastrear.	3. Muestra en un mapa un marcador ¹⁷ por cada vehículo que está de servicio, y muestra el detalle en una tabla.	
Post Condición: Representar a los vehículos en el mapa y mostrar el detalle en una tabla.		
Cursos Alternos: El usuario tiene la posibilidad de elegir los vehículos a rastrear, entonces tiene la posibilidad de seleccionar únicamente los vehículos		

¹⁷ También llamado pushpin, es la representación de un objeto en el mapa.

de su interés.

Tabla 16: Caso de uso expandido "Rastrear vehículos en tiempo real"

CASO DE USO	CU5. Mostrar rutas seguidas por los vehículos	
ACTORES	Operador	
OBJETIVO	Visualiza en un mapa las posiciones de los vehículos seleccionados en una determinada fecha y hora.	
DESCRIPCIÓN	El operador, visualiza en un mapa de la interfaz las posiciones de los vehículos seleccionados en una determinada fecha y hora. Se muestra su posición en el mapa. En una tabla de la misma interfaz, se muestra el detalle, como es la hora, la velocidad, el tiempo de parada y dirección de la calle.	
DESCRIPCIÓN INICIAL PASO A PASO		
Pre condición: Obtener los vehículos.		
ACTOR	SISTEMA	
1. Selecciona la opción de mostrar rutas seguidas por los vehículos.	2. Muestra la lista de vehículos.	
3. Selecciona de la lista de vehículos las unidades que desea mostrar sus rutas. Además selecciona una fecha y un intervalo de horas.	3. Muestra en un mapa una ruta por cada vehículo, en una determinada fecha y entre el intervalo de horas.	
Post Condición: Representar las rutas de los vehículos en el mapa y mostrar el detalle en una tabla. Ese detalle se muestra la hora, la velocidad, el tiempo de parada, y la dirección de la calle.		
Cursos Alternos: El usuario tiene la posibilidad de elegir los vehículos a rastrear, entonces tiene la posibilidad de seleccionar únicamente los vehículos de su interés.		

Tabla 17: Caso de uso expandido "Mostrar rutas seguidas por los vehículos"

CASO DE USO	CU6. Consultar reportes	
ACTORES	Gerente General o Administrador	
OBJETIVO	Consultar reportes sobre las transgresiones de velocidad y tiempo de parada, y sobre el consumo de combustible.	
DESCRIPCIÓN	El gerente general puede realizar la consulta sobre las transgresiones de velocidad y tiempo de parada, y sobre el consumo de combustible; en el momento que crea conveniente.	
DESCRIPCIÓN INICIAL PASO A PASO		
Pre condición: Información proporcionada correctamente.		
ACTOR	SISTEMA	
1. Selecciona la opción de reporte.	1. Muestra una ventana que le pide datos necesarios para la consulta.	
2. Ingresar datos para la consulta.	1. Muestra el resultado en una tabla de la interfaz.	
Post Condición: Muestra el reporte		
Cursos Alternos: Ninguna.		

Tabla 18: Caso de uso expandido "Consultar reportes"

CASO DE USO	CU7. Cambiar estado de vehículo	
ACTORES	Conductor	
OBJETIVO	Modificar el estado del conductor.	
DESCRIPCIÓN	El conductor mientras está de servicio, desde el dispositivo móvil integrado en su vehículo, pone en conocimiento de los operadores de sus estados: "En servicio/Fuera de servicio", "Con pasajero/Sin pasajero" y "Emergencia/Fuera de riesgo".	
DESCRIPCIÓN INICIAL PASO A PASO		
Pre condición: El dispositivo debe estar configurado para emitir señal al dominio donde se encuentra el servidor.		

ACTOR	SISTEMA
1. Selecciona su estado actual: "En servicio/Fuera de servicio", "Con pasajero/Sin pasajero" y "Emergencia /Fuera de riesgo.	2. Notifica a la estación central. Dependiendo del estado se notifica en la interfaz de las máquinas cliente, que representará de diferente forma cada estado.
Post Condición: Representar a los vehículos en el mapa.	
Cursos Alternos: Ninguno	

Tabla 19: Caso de uso expandido "Cambiar estado de vehículo"

3.6.6 DIAGRAMA DE CLASES DE ANÁLISIS

CU1.CONFIGURAR PARÁMETROS LÍMITE

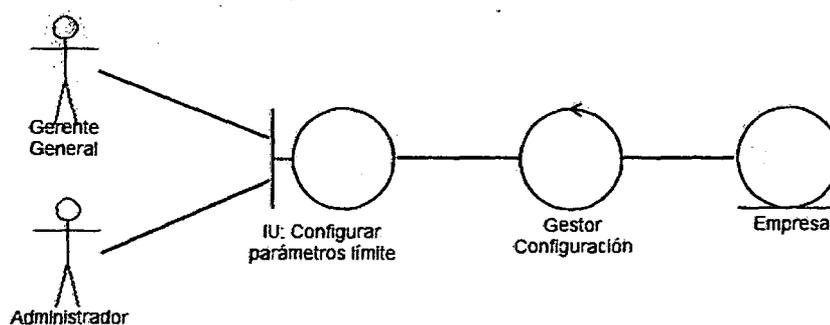


Ilustración 20: Diagrama de clase de análisis " Configurar parámetros límite"

CU2. ASIGNAR DISPOSITIVO GPS A VEHICULO

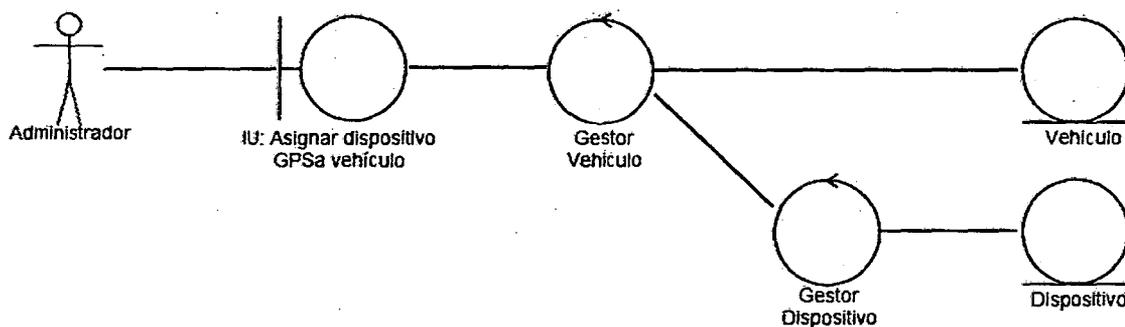


Ilustración 21: Diagrama de clase de análisis " Asignar dispositivo GPS a vehículo"

CU3. ASIGNAR CONDUCTOR A VEHICULO

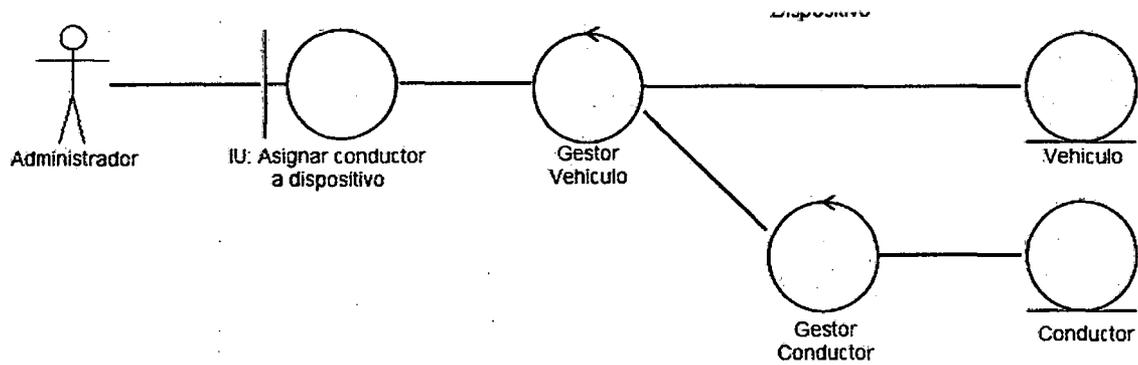


Ilustración 22: Diagrama de clase de análisis "Asignar conductor a vehículo"

CU4. RASTREAR VEHÍCULOS EN TIEMPO REAL

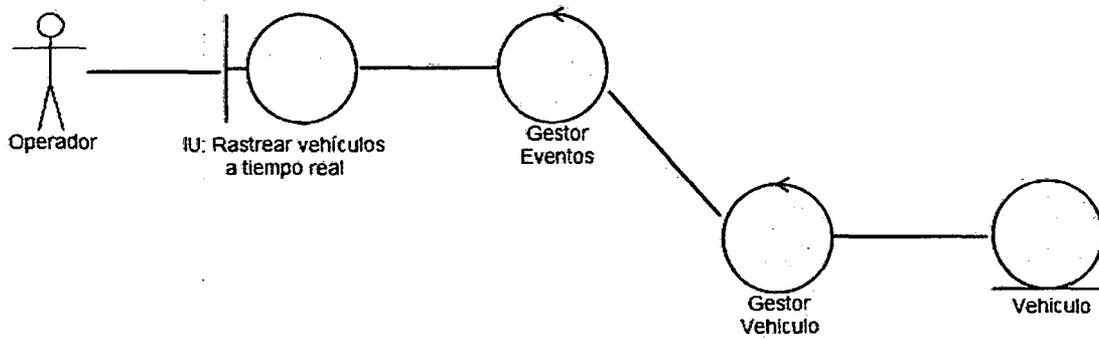


Ilustración 23: Diagrama de clase de análisis "Rastrear vehículos en tiempo real"

CU5. MOSTRAR RUTAS SEGUIDAS POR LOS VEHÍCULOS

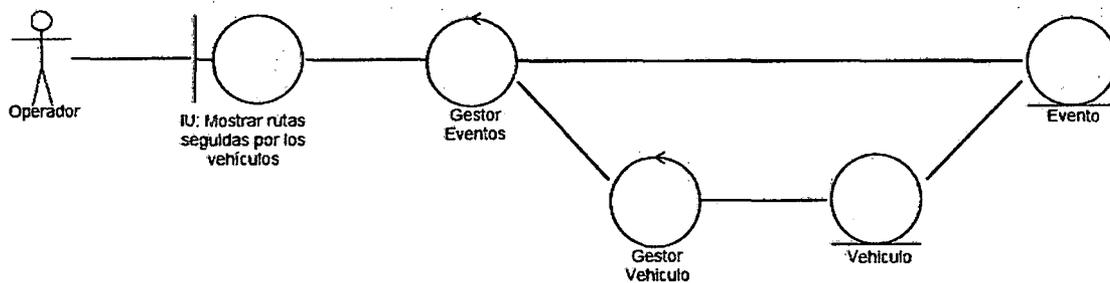


Ilustración 24: Diagrama de clase de análisis "Mostrar rutas seguidas por los vehículos"

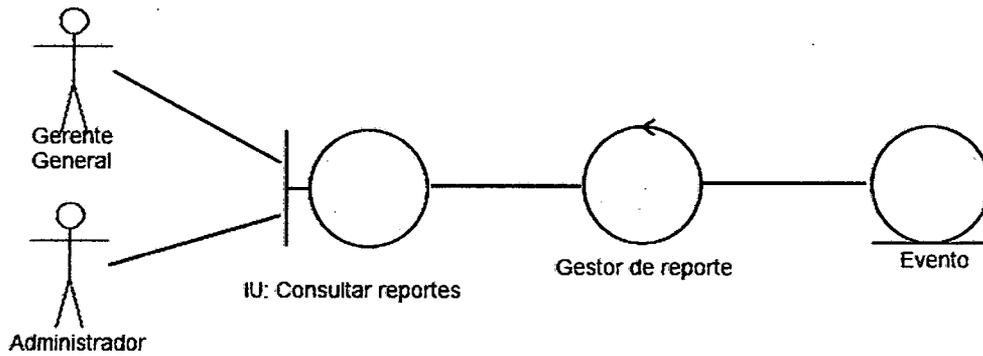
CU6. CONSULTAR REPORTES

Diagrama de clase de análisis " Consultar reportes"

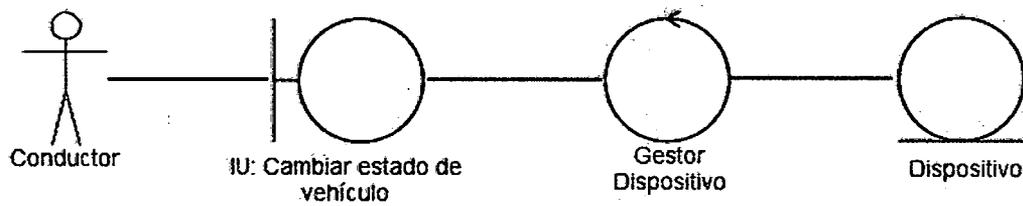
CU7. CAMBIAR ESTADO DE VEHICULO

Diagrama de clase de análisis " Cambiar estado de vehículo"

CAPÍTULO 4. FASE DE ELABORACIÓN

Hito principal: Definir la arquitectura base para el desarrollo del sistema de información y mitigar los riesgos identificados en el anterior capítulo, la construcción de los diagramas para el modelo de análisis y diseño para la estructuración de un diagrama de clases adecuado para el desarrollo del prototipo del proyecto.

4.1 DIAGRAMAS DE SECUENCIA

CUI.CONFIGURAR PARÁMETROS LÍMITE

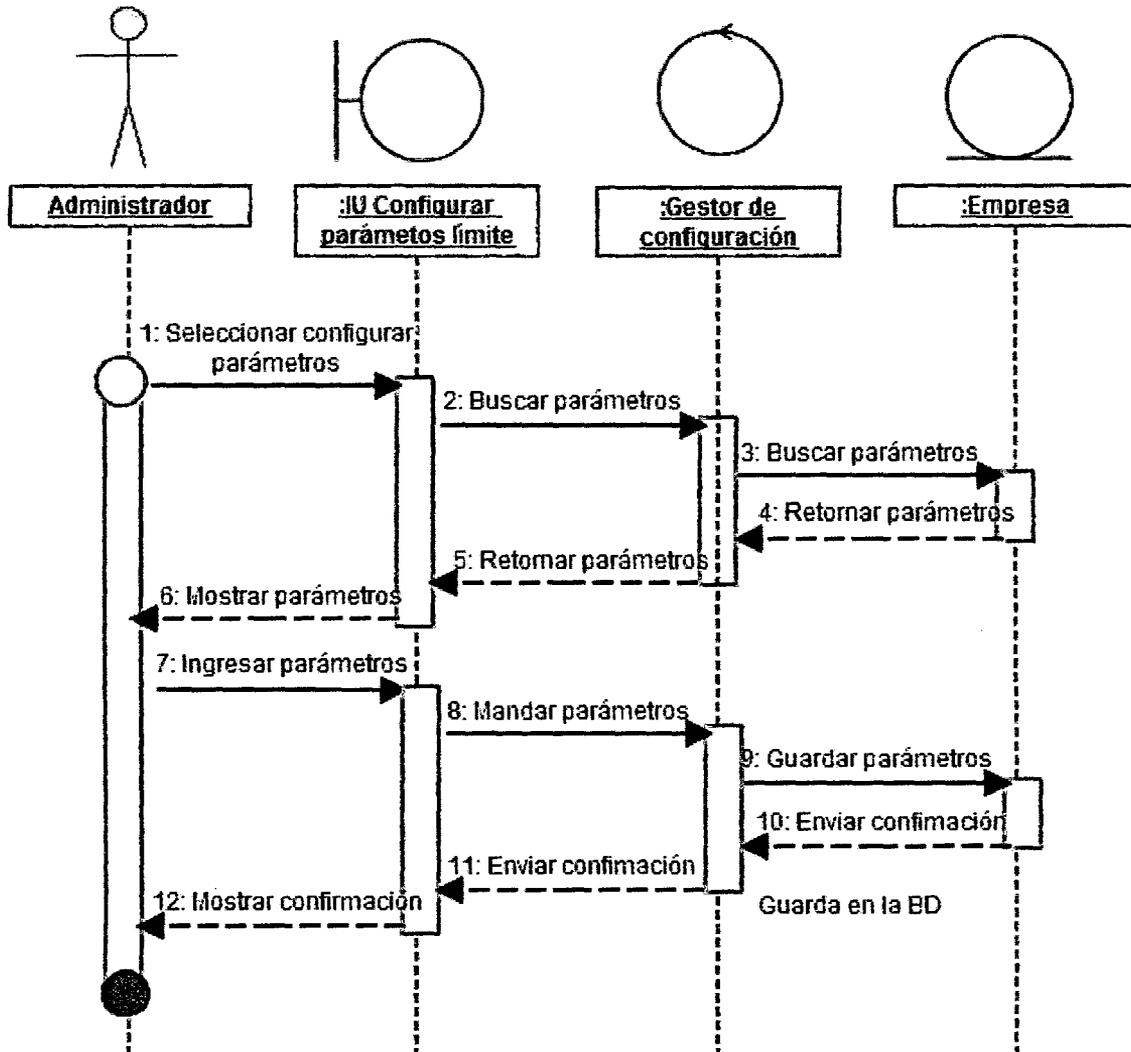


Ilustración 25: Diagrama de secuencia "Configurar parámetros de rastreo"

CU2. ASIGNAR DISPOSITIVO GPS A VEHICULO

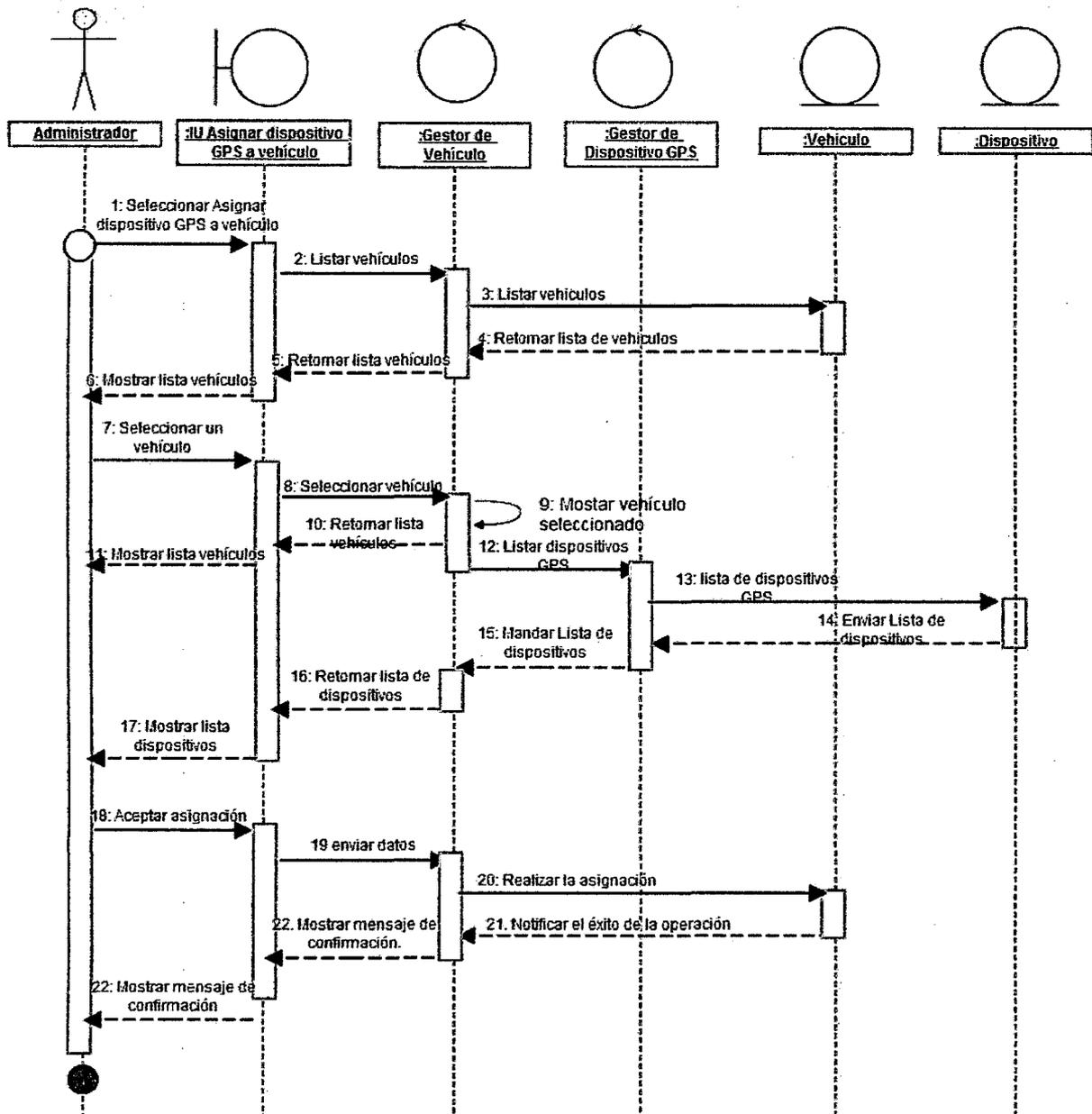


Ilustración 26: Diagrama de secuencia " Asignar dispositivo GPS a vehículo"

CU3. ASIGNAR CONDUCTOR A VEHICULO

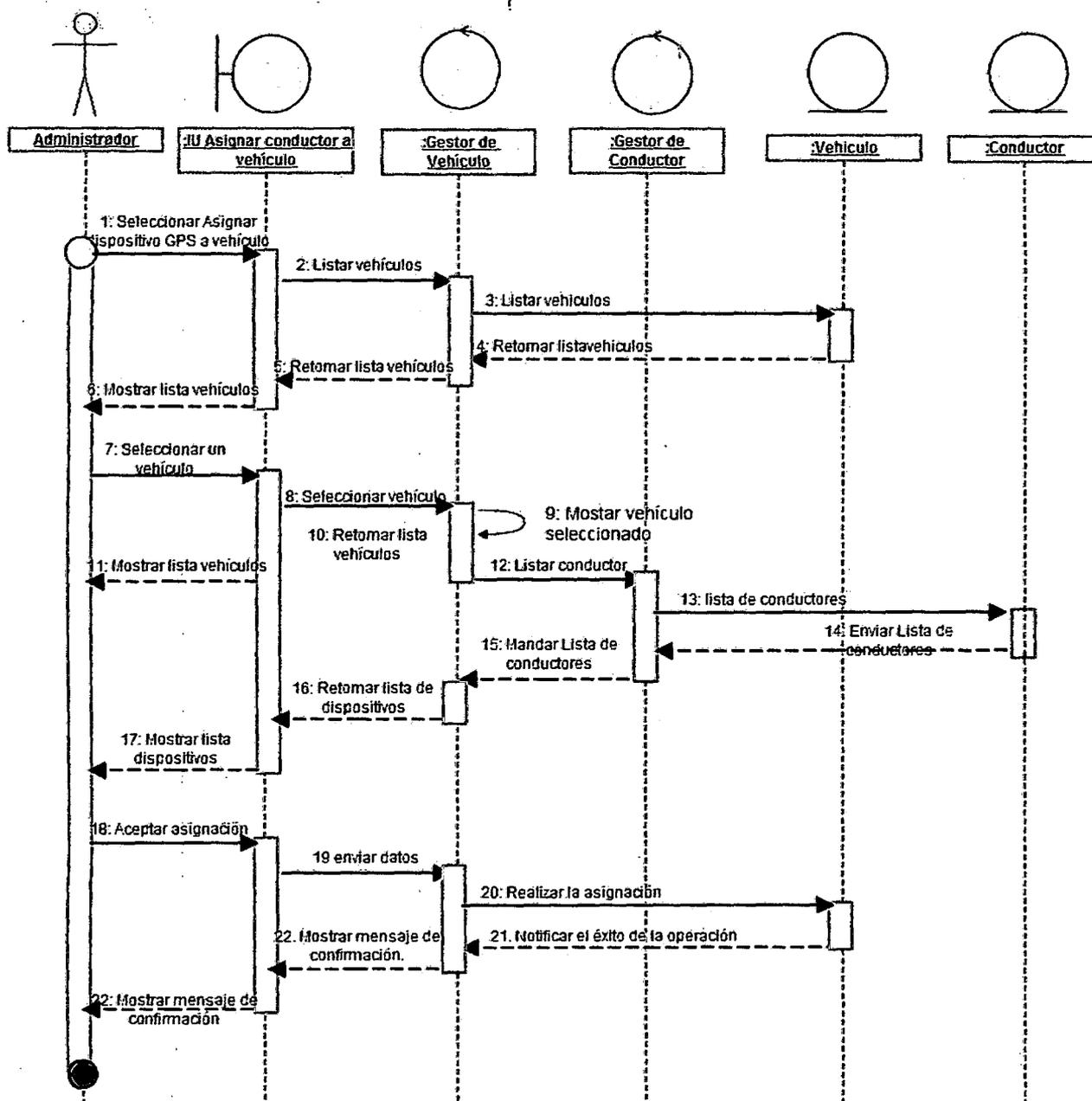


Ilustración 27: Diagrama de secuencia " Asignar conductor a vehículo"

CU4. RASTREAR VEHÍCULOS EN TIEMPO REAL

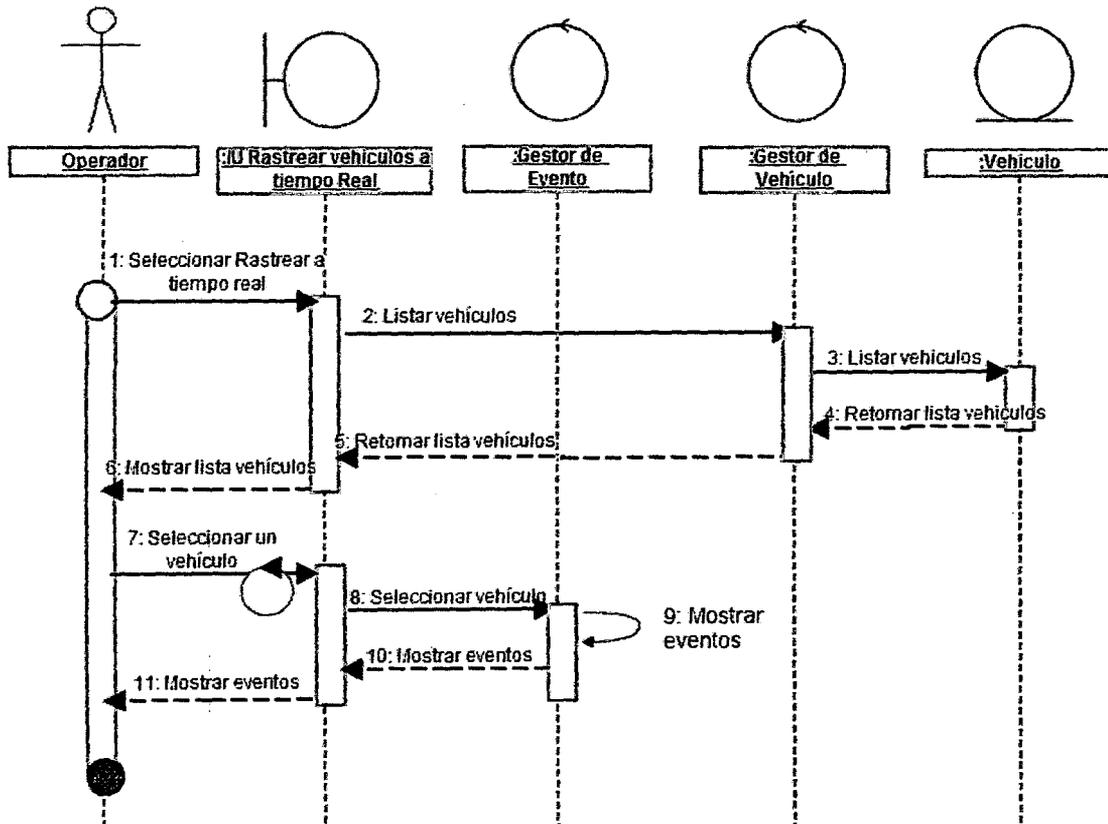


Ilustración 28: Diagrama de secuencia " Rastrear vehículos en tiempo real"

CU5. MOSTRAR RUTAS SEGUIDAS POR LOS VEHÍCULOS

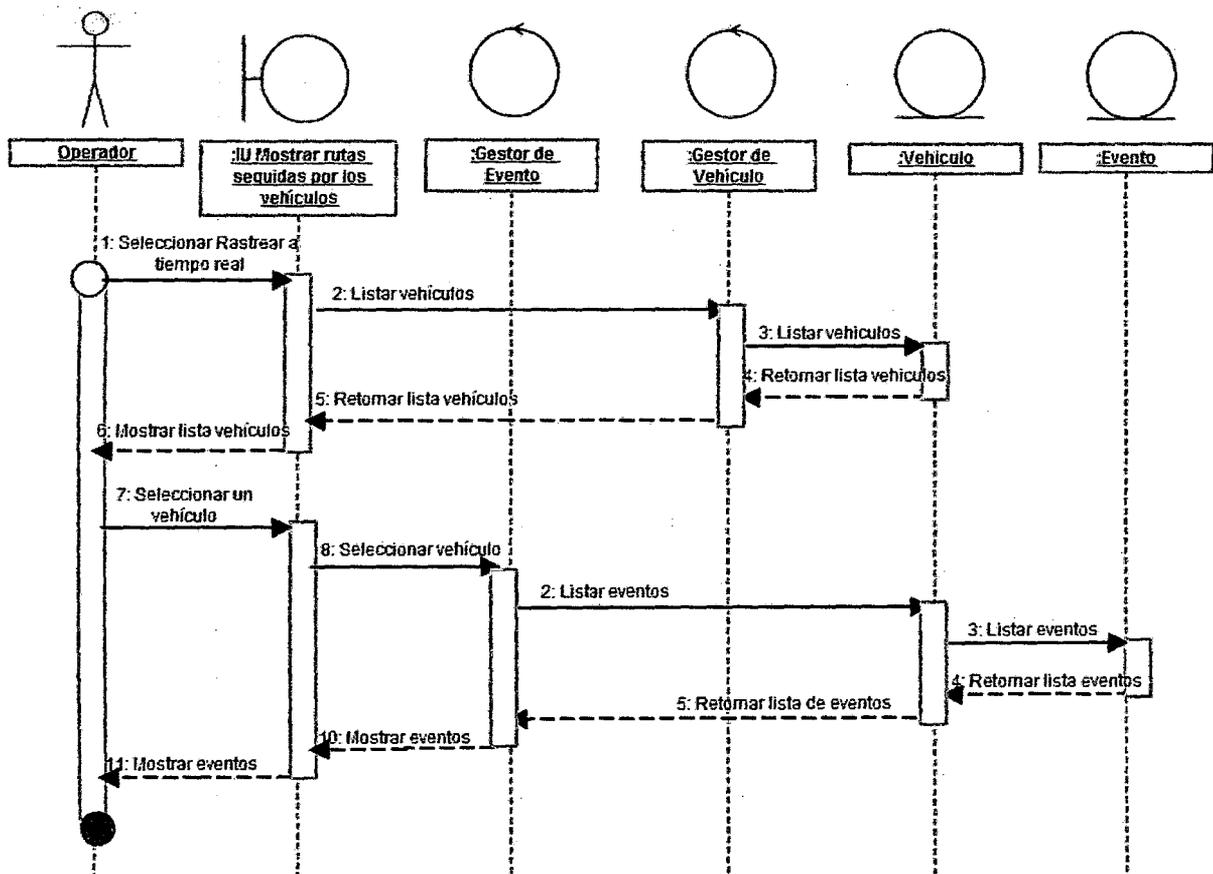


Ilustración 29: Diagrama de secuencia "Mostrar rutas seguidas por los vehículos"

CU6. CONSULTAR REPORTES

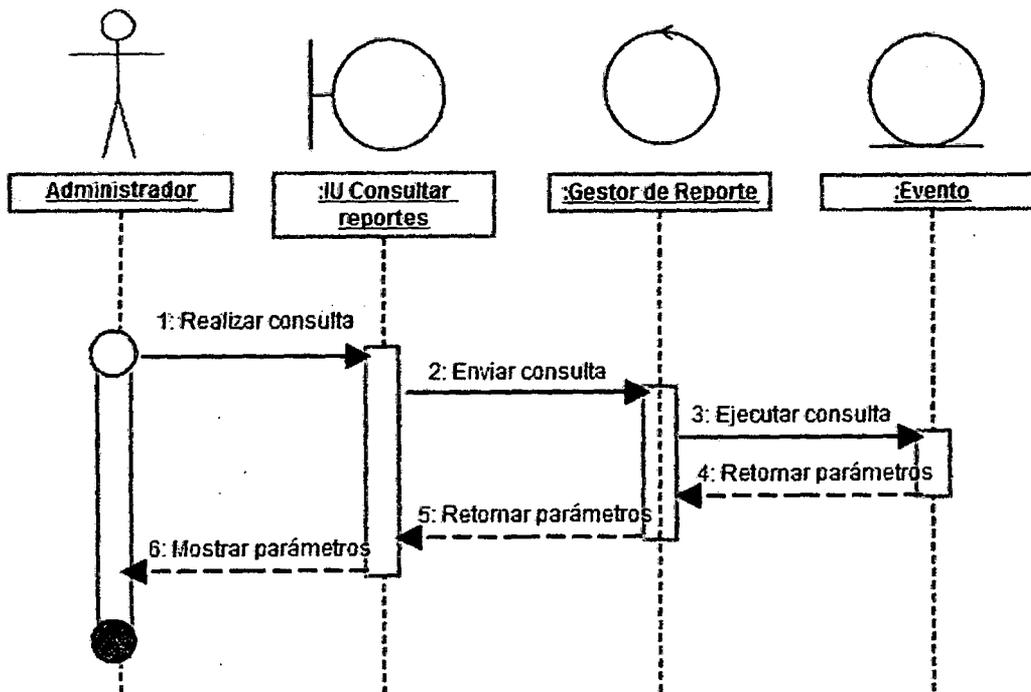


Ilustración 30: Diagrama de secuencia " Consultar reportes"

CU7. CAMBIAR ESTADO DE VEHICULO

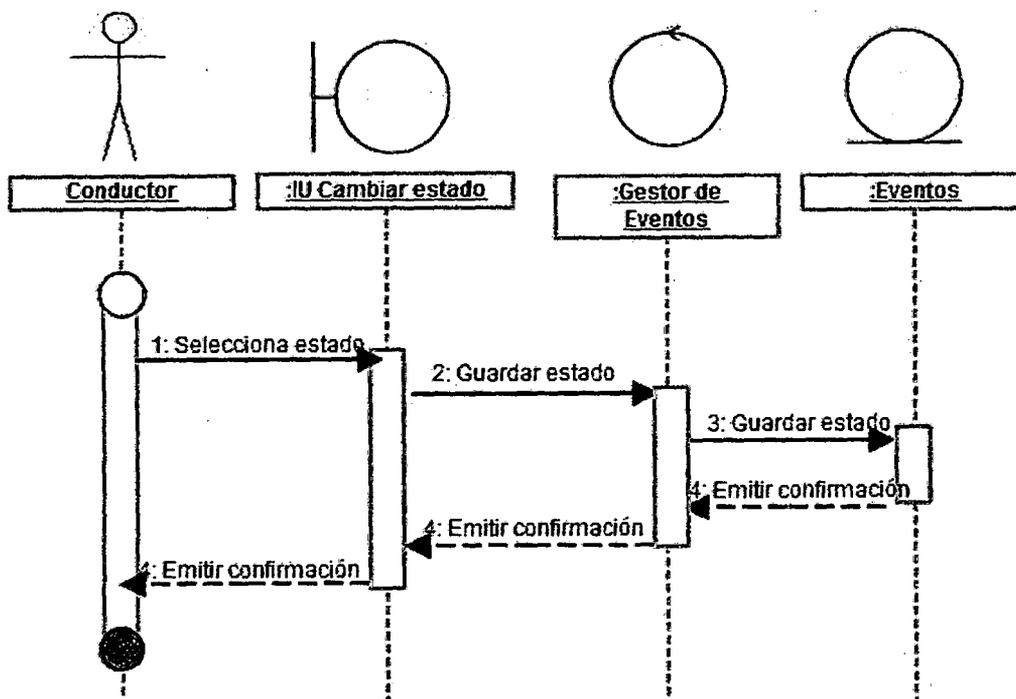


Ilustración 31: Diagrama de secuencia " Cambiar estado de vehículo"

4.2 DIAGRAMAS DE COLABORACIÓN

CU1.CONFIGURAR PARÁMETROS LÍMITE

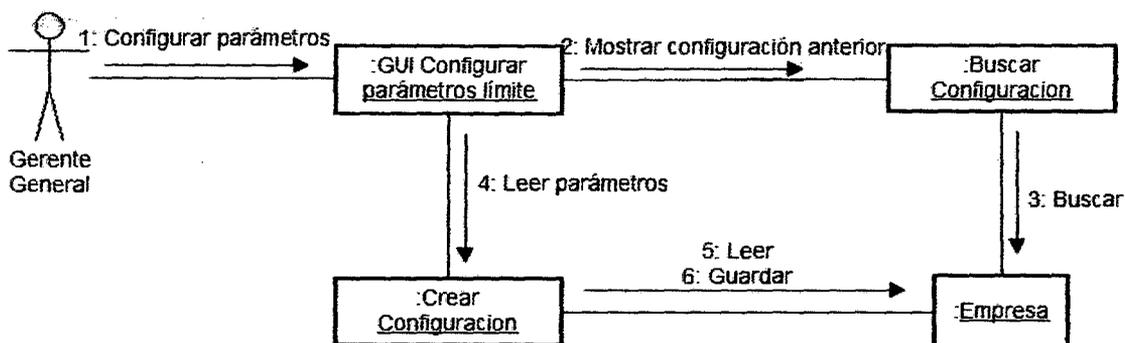


Ilustración 32: Diagrama de colaboración "Rastrear en tiempo real"

CU2. ASIGNAR DISPOSITIVO GPS A VEHICULO



Ilustración 33: Diagrama de colaboración "Asignar dispositivo GPS a vehículo"

CU3. ASIGNAR CONDUCTOR A VEHICULO

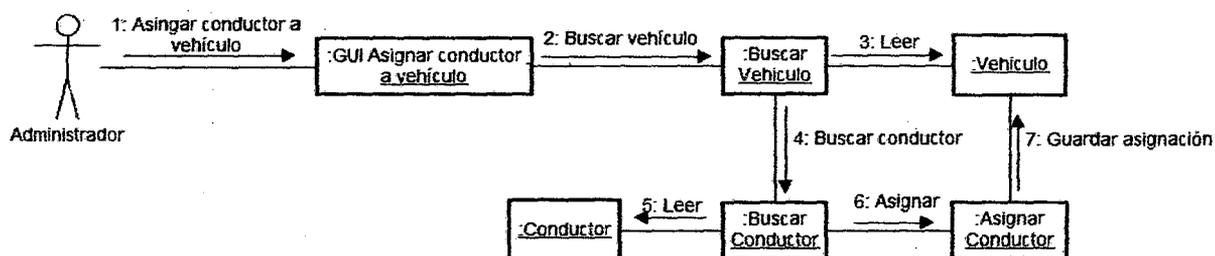


Ilustración 34: Diagrama de colaboración "Asignar conductor a vehículo"

CU4. RASTREAR VEHÍCULOS EN TIEMPO REAL

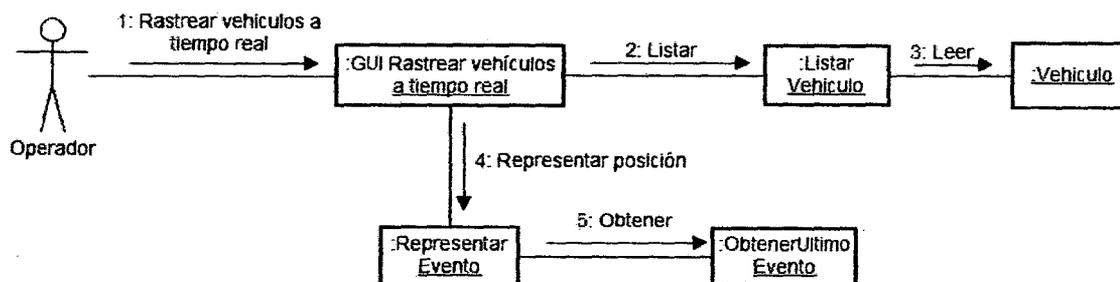


Ilustración 35: Diagrama de colaboración " Rastrear vehículos en tiempo real"

CU5. MOSTRAR RUTAS SEGUIDAS POR LOS VEHÍCULOS

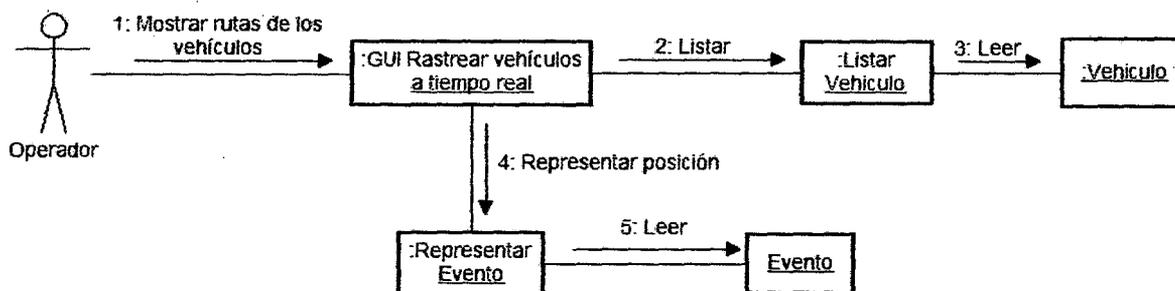


Ilustración 36: Diagrama de colaboración " Mostrar rutas seguidas por los vehículos"

CU6. CONSULTAR REPORTES

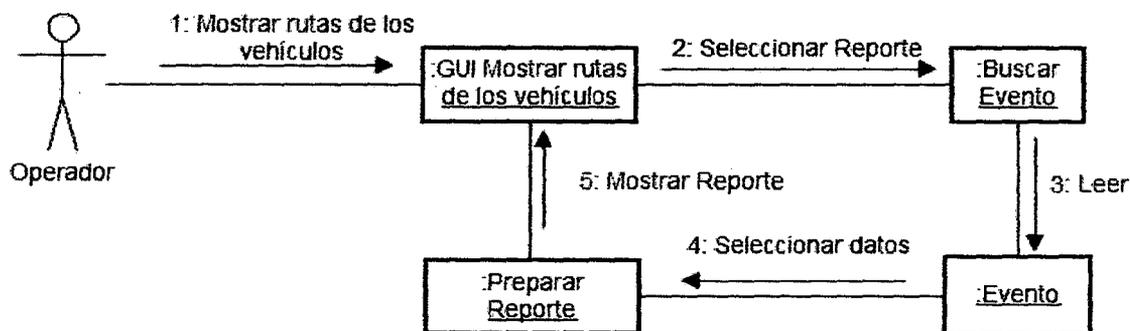
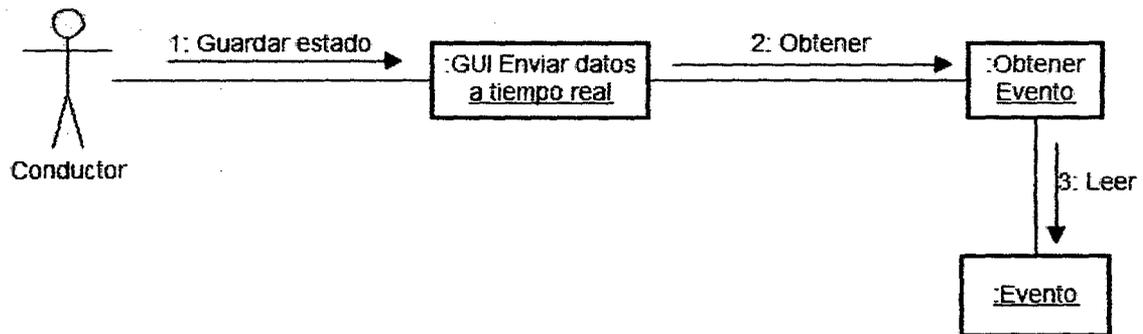


Ilustración 37: Diagrama de colaboración " Consultar reportes"

CU7. CAMBIAR ESTADO DE VEHICULO**Ilustración 38: Diagrama de colaboración " Cambiar estado de vehículo"**

4.3 DIAGRAMA DE CLASES

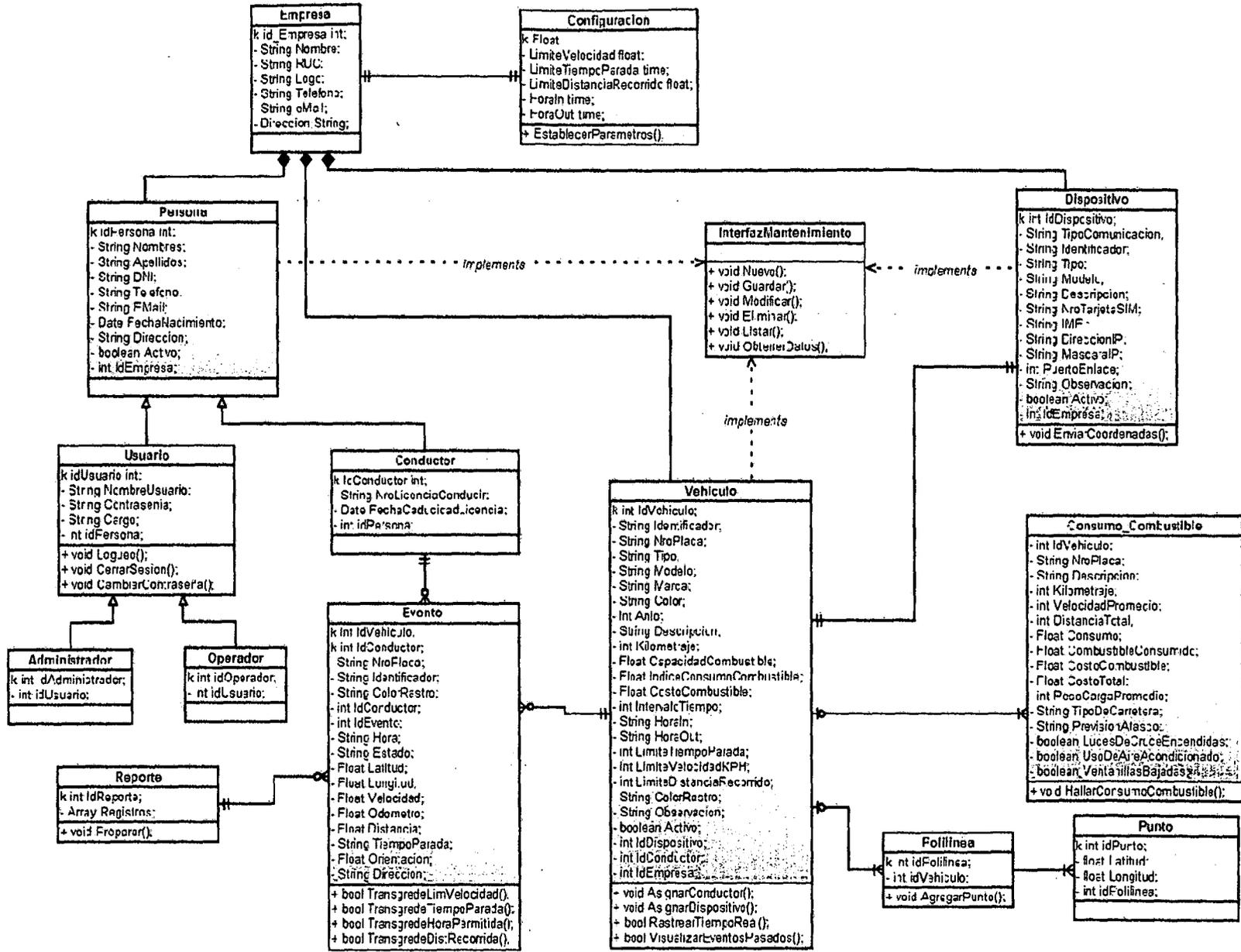


Ilustración 39: Diagrama de clases del sistema

4.4 ARQUITECTURA CANDIDATA

Esta sección describe los componentes usados y sus configuraciones, así también la arquitectura de comunicación que será implementada, reflejando las funciones que competen al lado del software y del hardware.

4.4.1 DISEÑO DE LA TOPOLOGÍA DEL SISTEMA

El sistema de rastreo y monitoreo vehicular necesita tener bien definida su estructura básica de funcionamiento, de acuerdo a los equipos y tecnologías a utilizarse. En la parte de diseño del sistema se ha optado por las herramientas tecnológicas y equipos que se consideran adecuados para el total cumplimiento de los objetivos.

4.4.1.1 TOPOLOGÍA REFERENTE AL HARDWARE

4.4.1.1.1 ESQUEMA DE LA TOPOLOGÍA

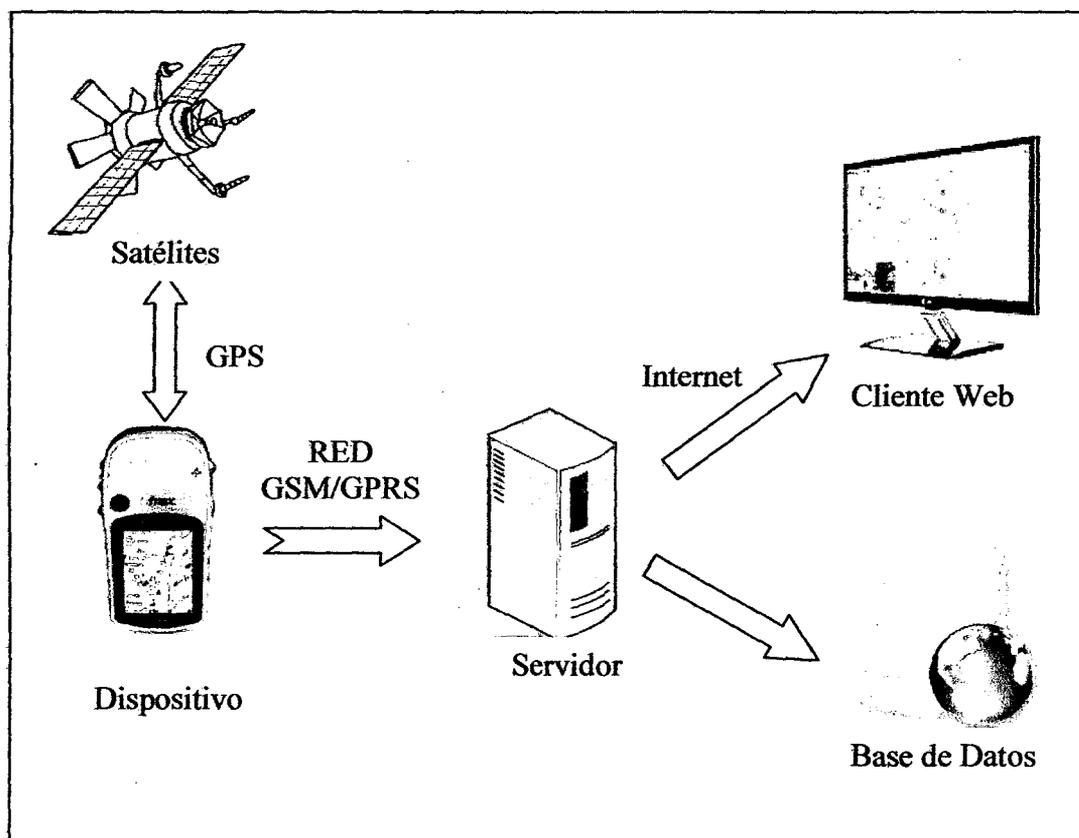


Ilustración 40: Esquema general del sistema

- Satélites.** Permite al receptor GPS obtener las coordenadas.
- Receptor GPS.** En este caso un teléfono móvil con sistema operativo Android, es el encargado de calcular la posición, luego enviarlas a través de la red celular GSM/GPRS a un servidor. El dispositivo hallará y lanzará su posición cada intervalo de tiempo.
- Servidor.** El servidor está permanentemente en escucha, manteniendo un puerto donde recibirá las señales de los dispositivos GPS. Al recibir las señales, en primer lugar, envía inmediatamente a los clientes. En segundo lugar, casi simultáneamente, almacena los parámetros en la BD para una futura reproducción. Ejecuta la aplicación web para que se pueda acceder a ella mediante un navegador web. El servidor estará alojado en la nube, es decir en el servidor de una empresa especializada en ofrecer alojamiento a sistema web, debido a que es más económico mantenerlo y ofrece las garantías de no perder datos.
- Base de datos.** Para almacenar los datos del sistema, de los conductores, dispositivos GPS, usuarios, vehículos, rutas.
- Cliente web.** Desde cualquier navegador web, se puede acceder a la aplicación web, sólo hace falta tener conexión a internet.

4.4.1.2 DESCRIPCIÓN DE LOS COMPONENTES

A continuación se da una especificación acerca del tipo de equipos que se necesitan para lograr el funcionamiento del sistema propuesto.

4.4.1.2.1 MÓVIL EQUIPADO CON GPS

Cada vehículo tendrá asignado un equipo móvil con las siguientes especificaciones:

- Sistema operativo Android a partir de la versión 2.2 o superior.

- Que esté implementado con un receptor GPS satelital.
- Móvil con acceso a internet, de preferencia se sugiere los planes que ofrecen las operadoras de telefonía móvil, que tienen planes desde 30 soles mensuales.
- Que esté conectada permanentemente a la energía eléctrica del carro, ya sea a la toma del mechero del vehículo o incluso conectado directamente a la batería del vehículo para evitar tenerlo que cargar cuando la batería esté descargada.
- Configurar el dispositivo para que no se apague automáticamente ni entre en modo suspendido, asegurándose de tal forma que la aplicación a implementar funcione sin perturbaciones.

4.4.1.2.2 ESTACIÓN CENTRAL

El servidor está alojado en una VPC que ofrece Amazon, en esta VPC se requiere fundamentalmente:

- Apache Tomcat 7.2
- Mysql 5.5 server y client.
- Jdk 7 (1.7)
- PhpMyAdmin

4.4.1.2.3 MÁQUINAS CLIENTE

Puede ser una computadora, laptop o tablet con internet, no es necesario tener instalado ningún software, tampoco importa el sistema operativo, nada más basta con tener un navegador ya sea Google Chrome, Firefox, Internet Explorer, Opera o Safari, con el plugin de java instalado (jvm).

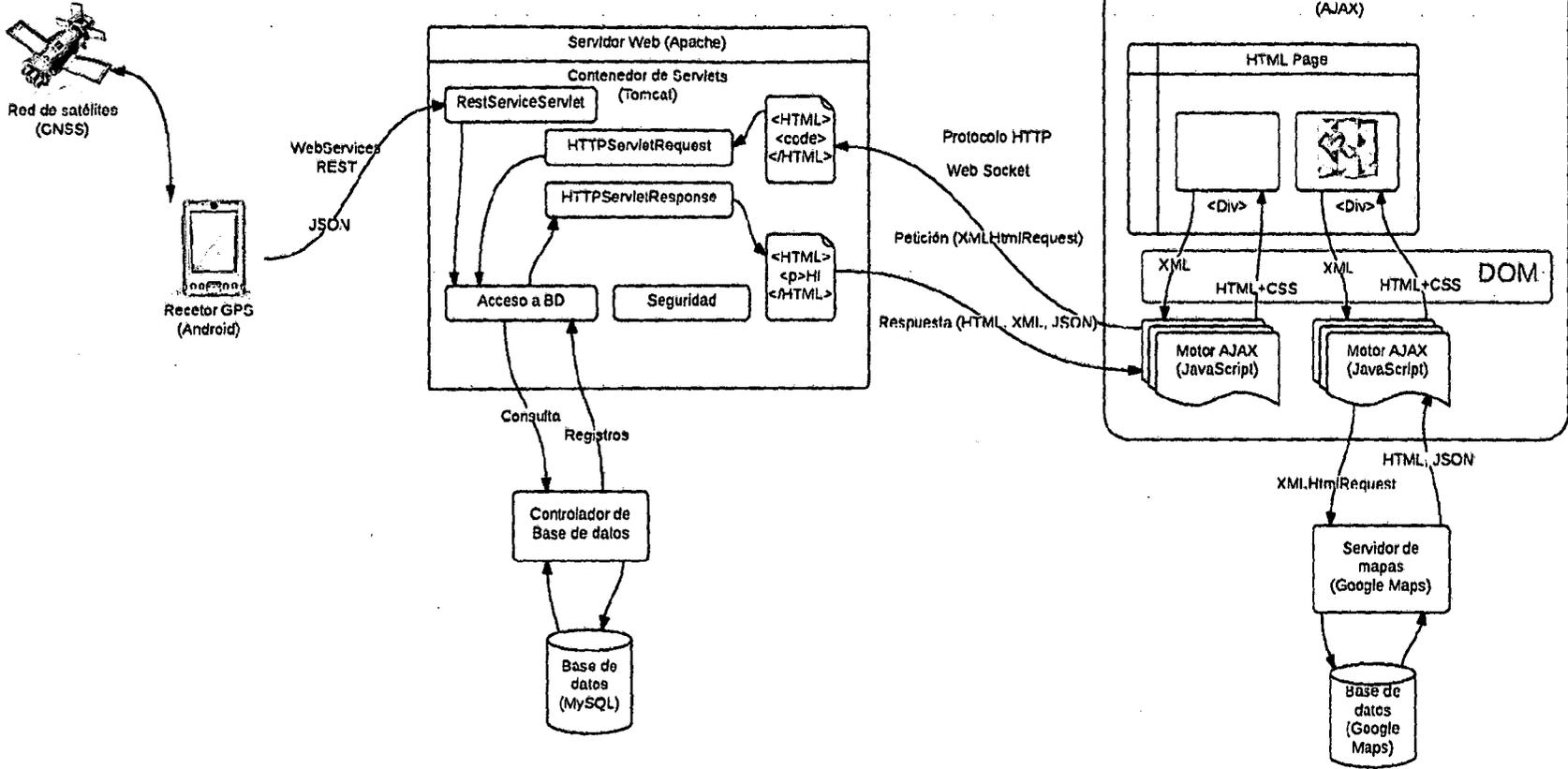


Ilustración 41: Diseño de la topología general del sistema

CAPÍTULO 5. FASE DE CONSTRUCCIÓN

Hito principal: Construcción del sistema web y que los usuarios puedan comenzar su operatividad con la ejecución del aplicativo, con el propósito de perfeccionar los casos de uso determinados en las fases anteriores. Es decir, esta fase corresponde a la construcción misma del aplicativo web.

5.1 IMPLEMENTACIÓN DEL SISTEMA

En esta etapa se describe las características del sistema web, identificando para esto sus asociaciones y descripción de los atributos que tendrá en base a su diagrama de clases, así como la descripción de los casos de uso con las interfaces ya diseñadas.

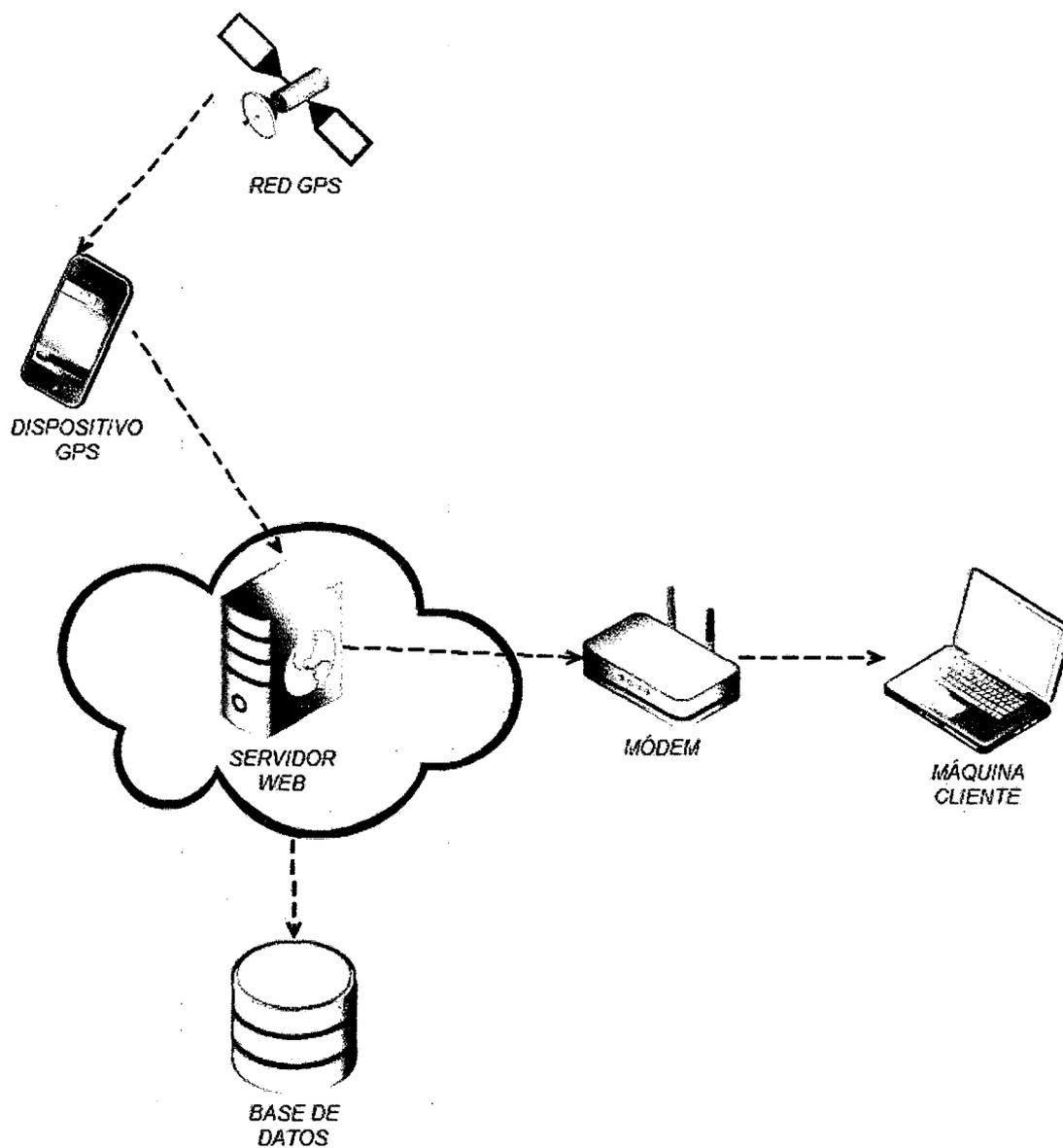


Ilustración 42: Diagrama de estructura del modelo de implementación

5.1.1 IMPLEMENTACIÓN DEL SERVIDOR

Para subir esta aplicación a internet se vio por conveniente la creación de un VPC¹⁸ de Amazon, el cual puede funcionar como un servidor remoto, en el que se puede configurar e instalar las dependencias que la aplicación exige, con la misma facilidad con la que uno configura su propio servidor local.

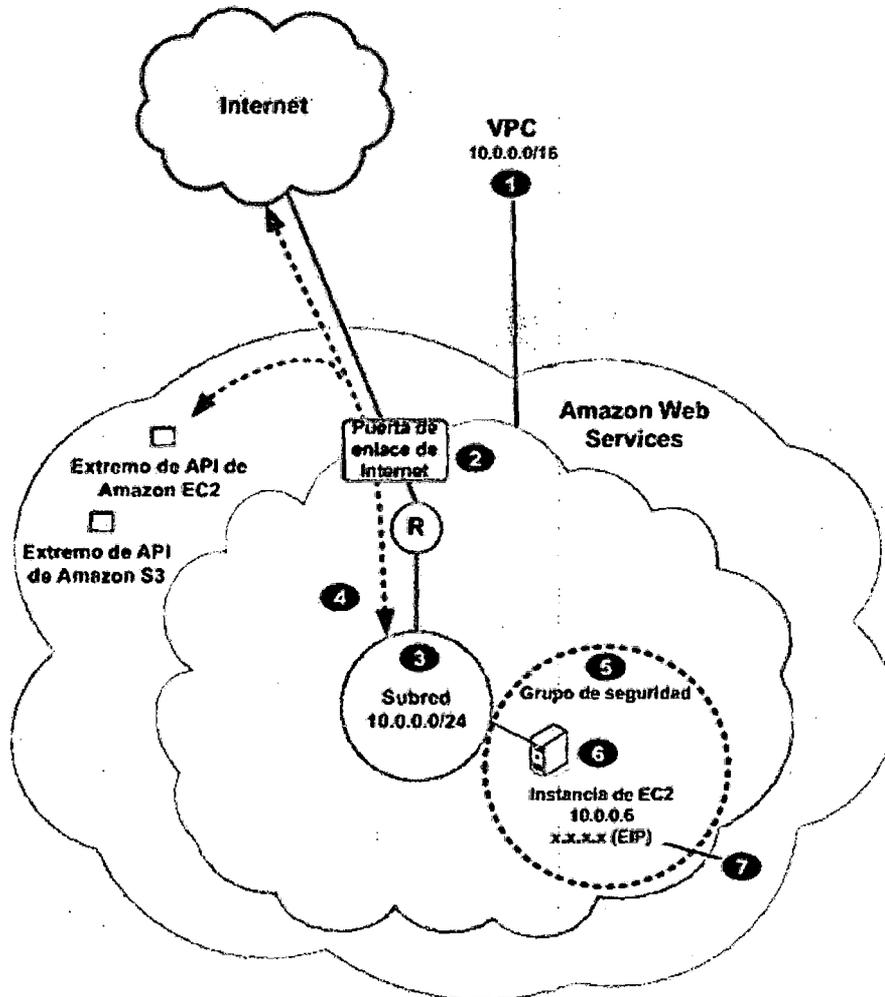


Ilustración 43: Implementación del servidor

Fuente: Amazon.org

Donde se tiene:

¹⁸ Virtual Private Cloud, entorno configurable y dinámico de recursos públicos en la nube que requiere el uso de protocolos de encriptación, túneles y otros procedimientos de seguridad para transferir datos entre una empresa privada y un proveedor de servicios en la nube. Además permite crear una topología de red virtual, en la que se incluyen subredes y enrutamientos.

1. VPC, que consiste de una parte aislada de la nube de AWS.
2. Puerta de enlace de internet, que conecta la VPC directamente a internet y proporciona acceso a otros recursos de AWS.
3. Subred de Amazon VPC, que consiste de un segmento de rango de direcciones IP de la VPC en el que lanza instancias de Amazon EC2¹⁹. Las subredes permiten agrupar instancias en función de las necesidades operativas y de seguridad.
4. Enrutamiento de la VPC, para permitir que el tráfico fluya entre la subred e Internet.
5. Grupo de seguridad para controlar el tráfico entrante y saliente para las instancias que se lanza.
6. Instancia en la subred que puede ser Linux o Windows; la instancia tiene una dirección IP privada del rango de direcciones de la subred.
7. Dirección IP elástica a la instancia. Una IP elástica es una dirección pública estática que puede asignar a cualquier instancia de la VPC. Esta asignación asocia a la instancia una dirección IP pública en lugar de su dirección privada. La IP elástica es necesaria para que la instancia VPC asociada se pueda acceder desde internet.

Para la aplicación fue necesaria sólo una instancia de Linux, en la que se instalaron las siguientes dependencias:

- Apache Tomcat 7.2
- Mysql 5.5 server y client.
- Jdk 7 (1.7)
- PhpMyAdmin

¹⁹ Amazon Elastic Compute Cloud (Amazon EC2), es un servicio web que permite lanzar y gestionar instancias de servidores de Linux/UNIX y Windows en centros de datos de Amazon.

La dirección IP pública asignada por Amazon la aplicación del presente proyecto es 50.112.185.33.

Posteriormente accediendo a Apache Tomcat se subió la aplicación a internet, llamada "RastreoGPSv2.2", y la base de datos con el nombre bdRastreoGPS, a la que se puede acceder mediante la dirección: <http://ec2-50-112-185-33.us-west-2.compute.amazonaws.com:8080/Rastreo>.

5.1.2 IMPLEMENTACIÓN EN DISPOSITIVOS GPS

La aplicación creada se llama RastreoGPS.apk el cual requiere de una versión e Android 2.2 o superior.

5.1.3 IMPLEMENTACIÓN EN MÁQUINA CLIENTE

La aplicación se puede acceder desde cualquier computador (Computadora de escritorio, laptop, tablet) que tenga instalado cualquiera de estos navegadores:

- Google Chrome (versión 4.0 o superior)
- Internet Explorer (versión 9.0 o superior)
- Firefox (versión 8.0 o superior)
- Safari (versión 5.0 o superior)
- Opera (versión 10.7 o superior)

Y dicho navegador debe tener instalado el plugin de java (jvm).

5.1.4 MODELO RELACIONAL

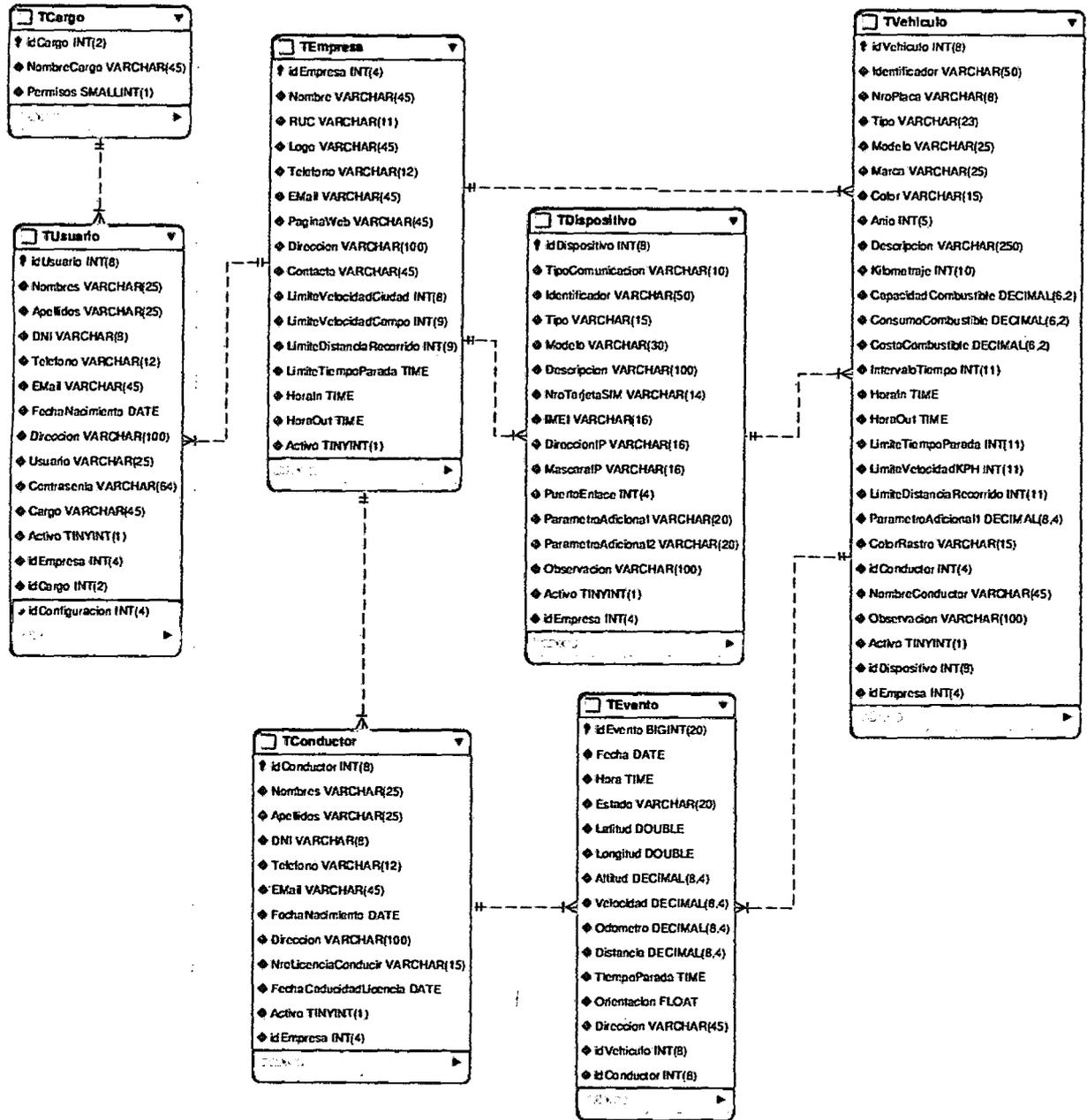


Ilustración 44: Diseño del modelo relacional

5.1.5 DIAGRAMA DE COMPONENTES

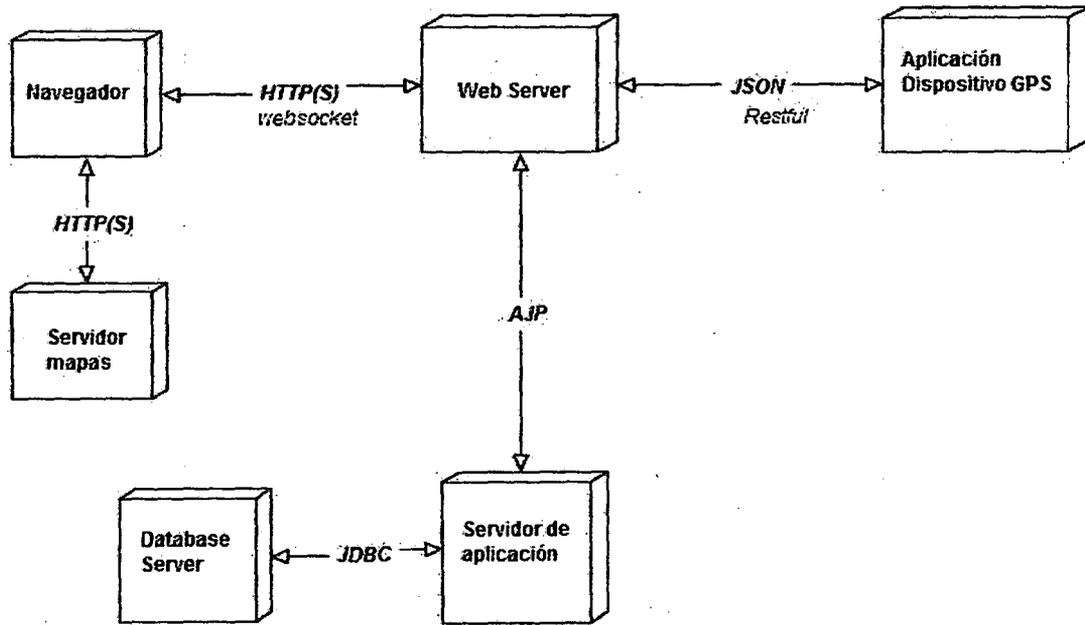
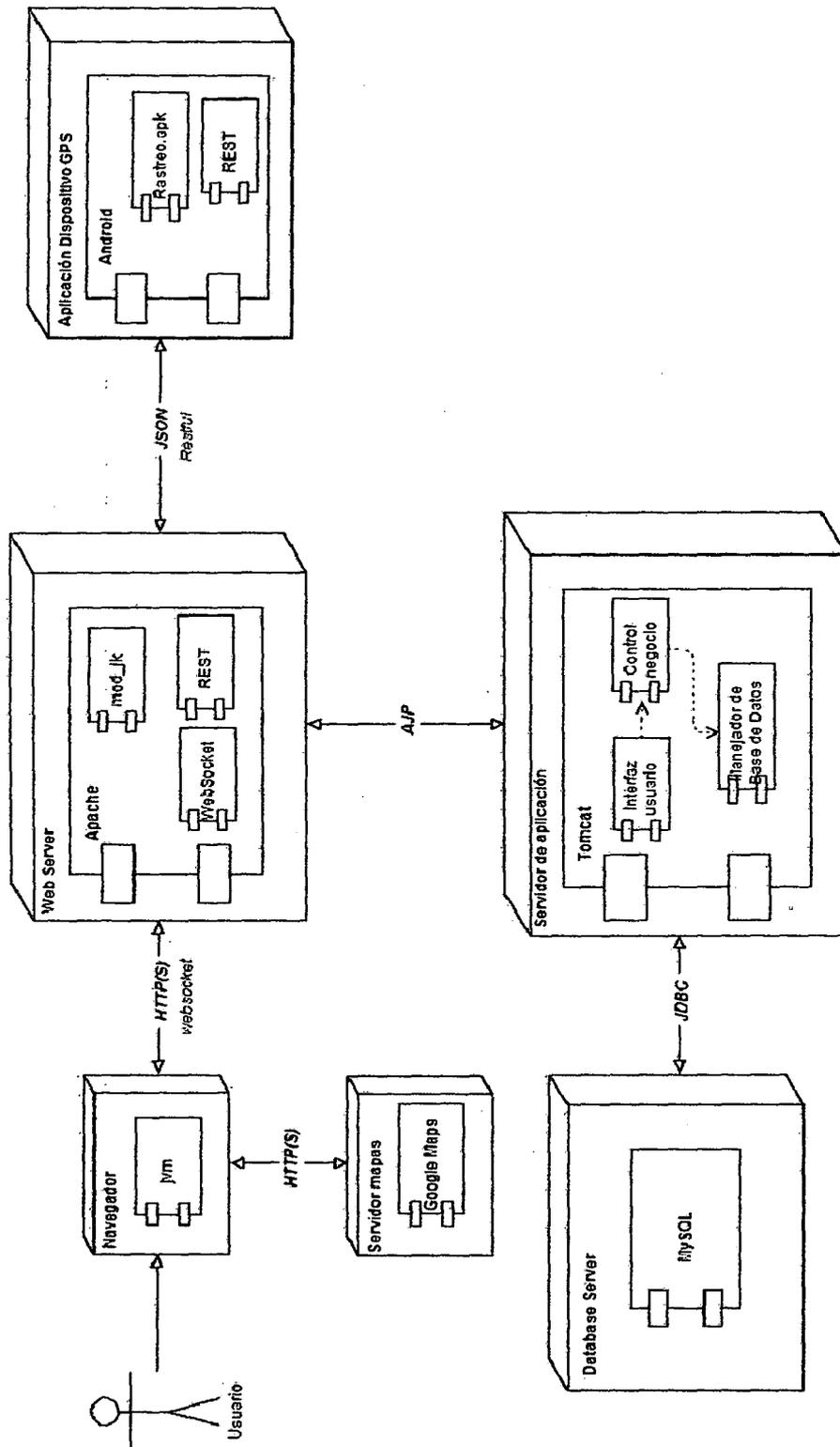


Ilustración 45: Diagrama de componentes

5.1.6 DIAGRAMA DE DESPLIEGUE



Huistración 46: Diagrama de despliegue

5.1.7 INTERFASES DEL SISTEMA A TRAVÉS DE CASOS DE USO REALES

Ahora se presenta la aplicación utilizando la interfaz final, para los casos de uso planteados en la fase de elaboración donde se diseñó el prototipo del sistema, en tal sentido se describe cada caso de uso mostrado en la aplicación web.

5.1.7.1 INTERFAZ DEL CASO DE USO REAL "CONFIGURAR PARÁMETROS LÍMITE"

Caso de uso:	CUI. Configurar parámetros límite
Actores :	Administrador o gerente general.
Propósito:	Configurar los parámetros que no deben transgredir los conductores.
Resumen:	El administrador configura los parámetros que no deben transgredir los conductores, como son velocidad máxima, tiempo de parada máxima y máxima distancia que puede recorrer un vehículo al día.

Tabla 20: Descripción del caso de uso real "Visualizar vehículos en tiempo real"

Ilustración 47: Interfaz del caso de uso real "Visualizar vehículos en tiempo real"

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1.- El usuario ingresa en "A" el límite de velocidad, en "B" el límite de distancia al día, en "C" el límite del tiempo de parada.	
2.- Da click en "D" para guardar la configuración.	
	3.- Guarda los parámetros límite.

Tabla 21: Detalle de caso de uso Real "Visualizar vehículos en tiempo real"

5.1.7.2 INTERFAZ DEL CASO DE USO REAL “ASIGNAR DISPOSITIVO GPS A VEHÍCULO”

Caso de uso:	Asignar dispositivo GPS a vehículo
Actores :	Administrador o delegado operador
Propósito:	Asignar a un vehículo un dispositivo GPS.
Resumen:	Puede que se realice con dispositivos GPS que no estén permanentemente en el vehículo. Esta interfaz permite cambiar de dispositivo GPS fácilmente.

Tabla 22: Descripción del caso de uso real “Asignar dispositivo GPS a vehículo”

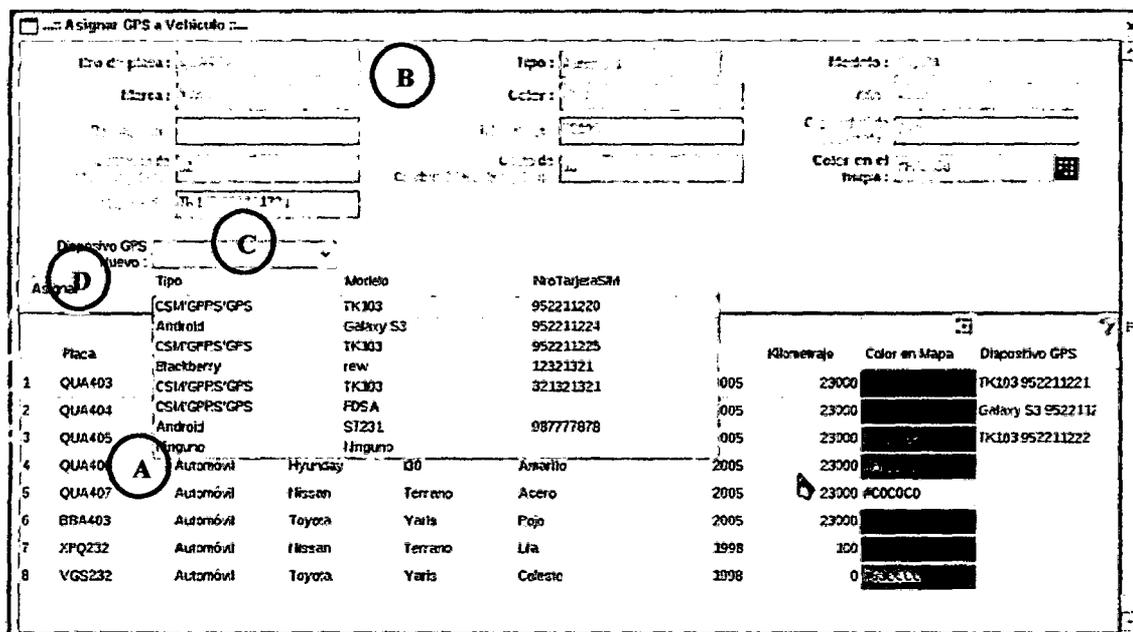


Ilustración 48: Interfaz del caso de uso real “Asignar dispositivo GPS a vehículo”

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1.- Selecciona en A el vehículo en cuestión.	
	2.-En B se muestra el detalle.
3.- En C selecciona al dispositivo a asignar.	
4.- Le da en D para asignar dicho dispositivo.	
	5.- El sistema le asigna.

Tabla 23: Detalle de caso de uso Real “Asignar dispositivo GPS a vehículo.”

5.1.7.3 INTERFAZ DEL CASO DE USO REAL "ASIGNAR CONDUCTOR A VEHÍCULO"

Caso de uso:	Asignar conductor a vehículo
Actores :	Administrador o delegado operador
Propósito:	Asignar a un vehículo un conductor.
Resumen:	Suele pasar en algunas empresas que el vehículo no es de uso exclusivo de un conductor. Esta interfaz permite cambiar de conductor fácilmente.

Tabla 24: Descripción del caso de uso real "Asignar conductor a vehículo"

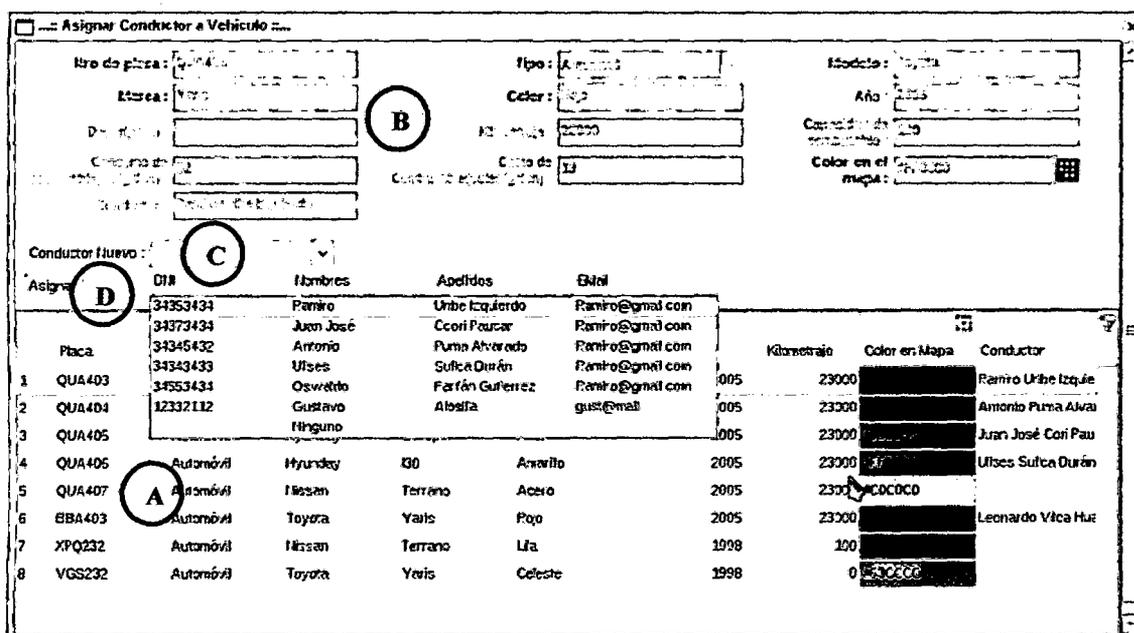


Ilustración 49: Interfaz del caso de uso real "Asignar conductor a vehículo"

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1.- Selecciona en A el vehículo en cuestión.	
2.- En B se muestra el detalle.	
3.- En C selecciona al conductor a asignar.	
4.- Le da en D que significa conformidad.	
	5.- El sistema le asigna.

Tabla 25: Detalle de caso de uso Real "Asignar conductor a vehículo"

5.1.7.4 INTERFAZ DEL CASO DE USO REAL "RASTREAR VEHÍCULOS EN TIEMPO REAL"

Caso de uso:	CU4. Rastrear vehículos en tiempo real
Actores :	Administrador o delegado operador.
Propósito:	Visualizar la ruta que están siguiendo los vehículos en el momento exacto de la consulta.
Resumen:	Esta interfaz permite la visualización de las rutas seguidas por los vehículos en tiempo real.

Tabla 26: Descripción del caso de uso real "Rastrear vehículos en tiempo real"

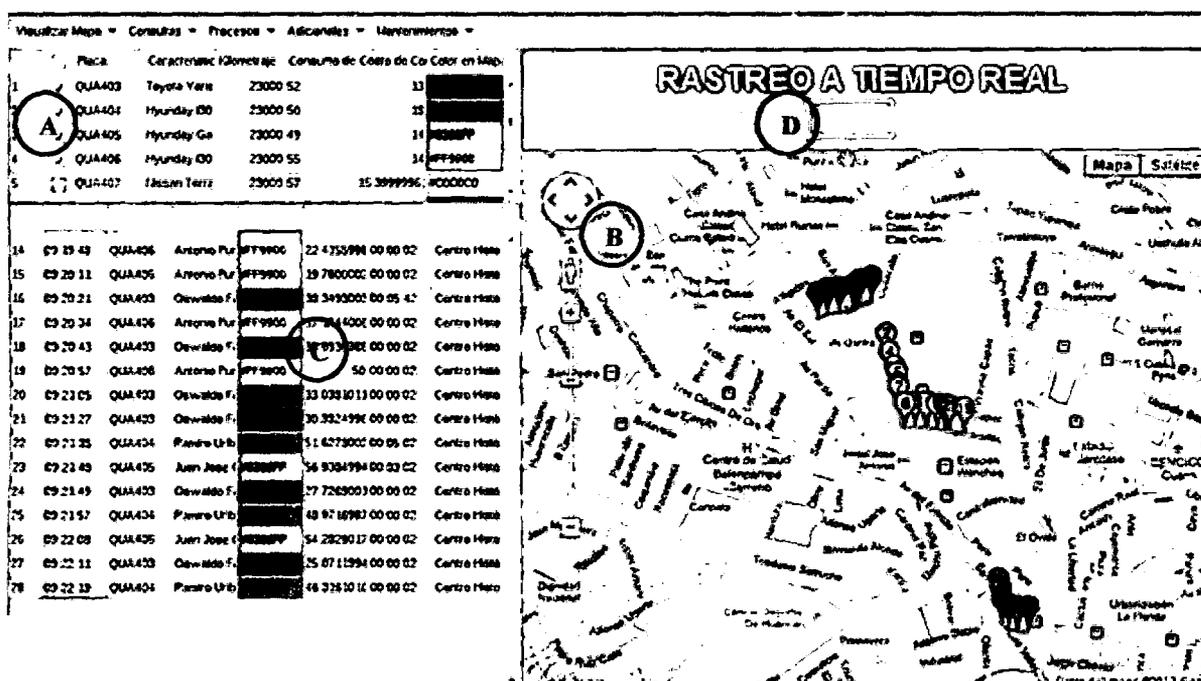


Ilustración 50: Interfaz del caso de uso real "Rastrear vehículos en tiempo real"

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1.- El usuario selecciona la opción de rastreo en tiempo real.	
	<p>2.- El sistema en A muestra los vehículos que están de servicio.</p> <p>3.-El sistema muestra los marcadores de los vehículos y se traza en B.</p> <p>4.- En la tabla C se muestra la lista de los eventos. En la columna 5 se puede apreciar las transgresiones de velocidad, y en la columna 6 las transgresiones de los tiempos de parada, se enfatizan resaltándolas con fondo rojo.</p>

Tabla 27: Detalle de caso de uso Real "Rastrear vehículos en tiempo real"

5.1.7.5 INTERFAZ DEL CASO DE USO REAL "VISUALIZAR RUTAS SEGUIDA POR LOS VEHÍCULOS"

Caso de uso:	CU5. Mostrar rutas seguidas por los vehículos
Actores :	Administrador o delegado operador.
Propósito:	Visualizar la ruta que están siguiendo los vehículos en el momento exacto de la consulta.
Resumen:	El operador, visualiza en un mapa las posiciones de los vehículos seleccionados en una determinada fecha y hora. Se muestra su posición en el mapa. En una tabla aparte, se muestra el detalle, como es la hora, la velocidad, el tiempo de parada y dirección de la calle.

Tabla 28: Descripción del caso de uso real "Visualizar rutas seguidas por los vehículos"

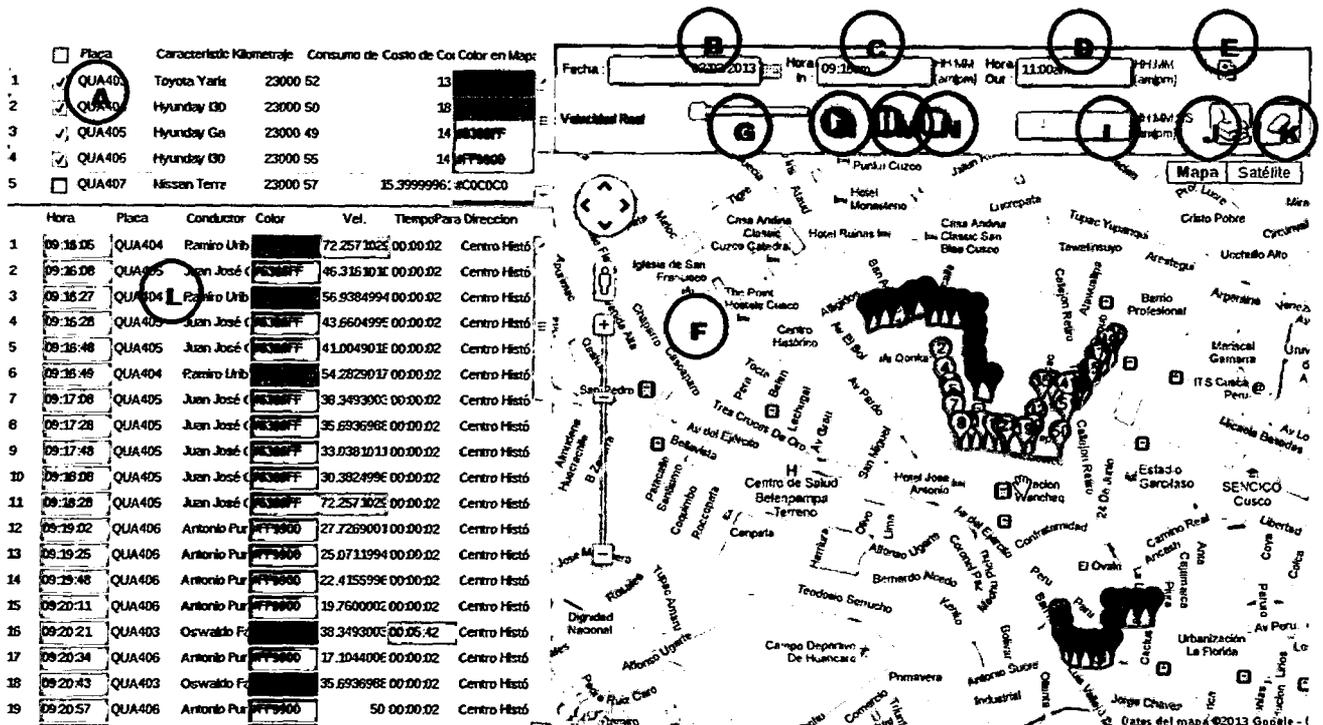


Ilustración 51: Interfaz del caso de uso real “Visualizar rutas seguidas por los vehículos”

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
<p>1.- El usuario selecciona, de la lista de vehículos que tiene la empresa en la que se desempeña, los vehículos a rastrear en A.</p> <p>2.- En B elije la fecha a consultar, en C y D los intervalos de horas.</p> <p>3- Luego presiona el botón en E.</p>	
	<p>4.-El sistema muestra en L la lista de eventos y en F la visualización completa de los eventos en el mapa de Google Maps, los puntos (marcadores) están identificados con una numeración correspondiente a la lista L de eventos. El usuario puede ver la información de los marcadores simplemente colocando el</p>

	<p>mouse encima del correspondiente marcador.</p> <p>5.- En L se puede apreciar en la columna 5 las transgresiones de velocidad, y en la columna C las transgresiones de los tiempos de parada resaltadas de rojo.</p>
<p>6.- El usuario puede seleccionar la velocidad de reproducción en G.</p> <p>En H se muestra tres controles: play, parar, stop.</p> <p>7.- Selecciona el botón H el primero es para dar play a la reproducción.</p>	
	<p>8.- En el mapa se colocan los marcadores uno a uno según la reproducción a la vez que los eventos de la lista se van seleccionando según la hora que indica I.</p>
<p>9.- Selecciona el botón M para parar la reproducción.</p>	
	<p>10.- La ubicación de los marcadores para en F y la lista L para de seleccionarse. El usuario puede ahora cambiar nuevamente la velocidad de reproducción en G si así lo desea.</p>
<p>11.- Se para totalmente la reproducción presionando el último control de H.</p>	
	<p>12.- La interfaz se reinicia y permite al usuario realizar otra consulta sobre la misma interfaz, iniciando así un nuevo ciclo de este caso de uso.</p>
<p>13.- Si está demasiado copado de marcadores el mapa, el usuario puede limpiar el mapa haciendo click en J.</p>	

	14. Se limpia las rutas y los marcadores del mapa.
15.- Selecciona K para imprimir la consulta.	
	16.- Imprime la consulta como se muestra en el ANEXO9

Tabla 29: Detalle de caso de uso Real "Visualizar rutas seguidas por los vehículos"

5.1.7.6 INTERFAZ DEL CASO DE USO REAL "CONSULTAR REPORTE"

Caso de uso:	CU4. Consultar reportes
Actores :	Administrador o delegado operador.
Propósito:	Realizar consultas sobre las infracciones cometidas.
Resumen:	Realizar consultas al sistema sobre las infracciones cometidas por los las unidades vehiculares de manera concisa.

Tabla 30: Descripción del caso de uso real "Consultar reportes"

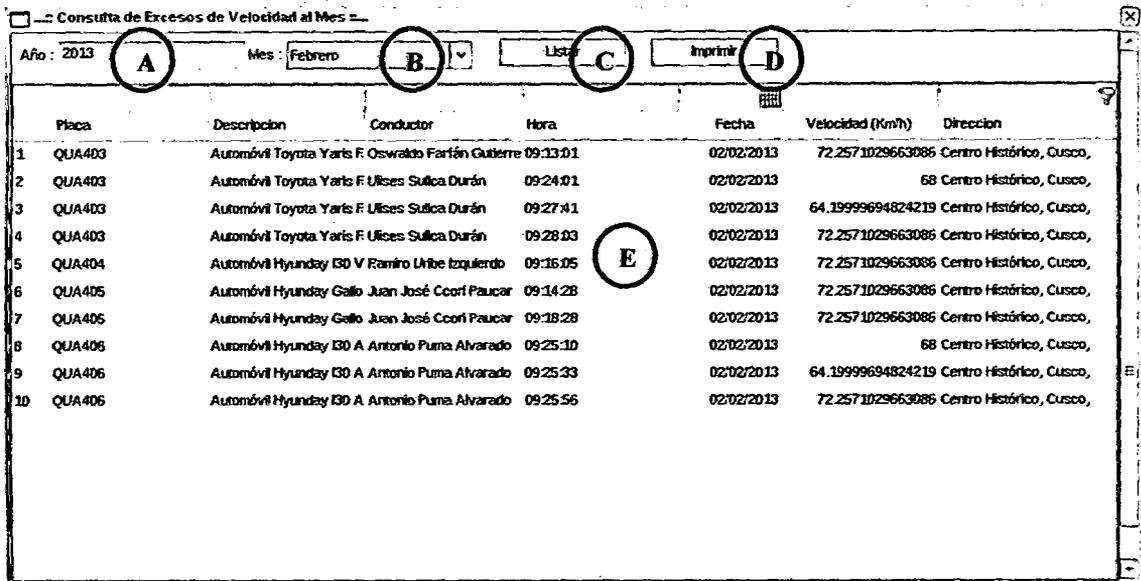


Ilustración 52: Interfaz del caso de uso real "Consultar reportes"

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1.- El usuario selecciona año en A y mes en B, luego en C los lista.	
	2.- El sistema muestra el reporte de la consulta en E.
3.- El usuario selecciona en D si desea imprimir.	
	2.- El sistema le permite imprimir.

Tabla 31: Detalle de caso de uso Real "Consultar reportes"

5.1.7.7 INTERFAZ DEL CASO DE USO REAL "CAMBIAR ESTADO DE VEHICULO"

Caso de uso:	CU4. Cambiar estado de vehículo
Actores :	Conductor.
Propósito:	El conductor informa su estado a la estación central, mediante un dispositivo Android.
Resumen:	El conductor mientras está de servicio, desde el dispositivo móvil integrado en su vehículo, pone en conocimiento de los operadores de sus estados: "En servicio/Fuera de servicio", "Con pasajero/Sin pasajero", "SOS" y "Fuera de riesgo".

Tabla 32: Descripción del caso de uso real "Cambiar estado de vehículo"

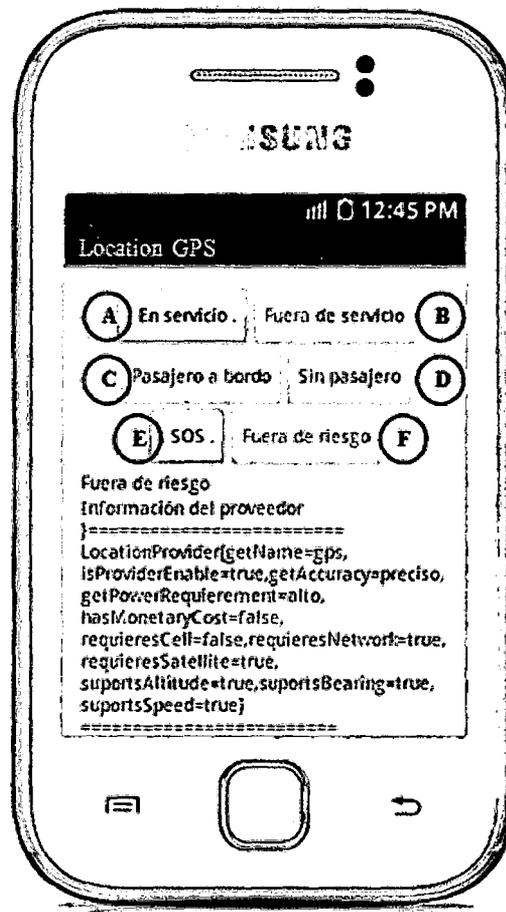


Ilustración 53: Interfaz del caso de uso real "Cambiar estado de vehículo"

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1.- El conductor selecciona A si está en servicio, en B si está fuera de servicio.	
	2.- La aplicación Android envía el cambio de estado a la estación central. 3.- Muestra a las máquinas cliente de los operadores.
4.- El conductor selecciona C si está con pasajero a bordo, en D si ya no tiene pasajero.	
	4.- La aplicación Android envía el cambio de estado a la estación central. 6.- Muestra a las máquinas cliente de los operadores.

7.- El conductor selecciona E si tiene alguna emergencia, puede ser un accidente, robo o asalto.	
	8.- La aplicación Android envía el cambio de estado a la estación central. 9.- Muestra a las máquinas cliente de los operadores.

Tabla 33: Detalle de caso de uso Real "Cambiar estado de vehículo"

CAPÍTULO 6. FASE DE TRANSICIÓN

Hito principal: Garantizar que se tiene una versión del producto listo para ser usado por los usuarios, se deja operativa la aplicación web y se realizan pruebas del software.

6.1 IMPLANTACIÓN DEL SISTEMA

En esta etapa se describe las características del aplicativo web, identificando para esto sus asociaciones y descripción de sus atributos todo esto ya en base de su diagrama de clases, así como la descripción de los casos de uso con las ventanas definidas en las interfaces de la aplicación web.

6.1.1 PREPARACIÓN DE LA VERSIÓN DE ACEPTACIÓN

Esta sección lleva a cabo la captura de ciertos puntos a recabar como son:

- Preparación de la documentación detallada y visual a través de la red-internet, como preparación a los usuarios en el proceso de las pruebas.
- Señalar las instrucciones que deberán seguir los usuarios durante las pruebas hechas en la red.
- Sincronizar de manera adecuada para la interacción de usuarios en diferentes sitios de acceso al aplicativo web.

6.1.2 INSTALACIÓN DE LA VERSIÓN DE ACEPTACIÓN

La instalación del aplicativo se realiza en un servidor virtual, cualquier modificación o mejora en el software será realizada por la suscrita, para la empresa de transporte “Turismo Ángeles del Cusco SAC” no habrían los inconvenientes de tener que instalar nuevo software o esperar buen tiempo para mejorar las funcionalidades del aplicativo, puesto que el aplicativo lo único que requiere del lado del cliente es un navegador web.

6.1.3 EVALUACIÓN DE LAS PRUEBAS

Se recopilan a continuación las pruebas realizadas durante el desarrollo de la tesis, estas prueban consisten en dos tipos básicos que son pruebas de laboratorio y pruebas de campo, las mismas que han sido divididas para un mejor análisis y entendimiento de los resultados obtenidos.

Para detectar posibles falencias que se presenten cuando se implemente el conjunto de tecnologías, primero se secciona las pruebas por componente.

El primer componente evaluado fue el GPS, el cual necesita aproximadamente de 1 a 3 minutos, luego de encendido, para la recepción de señales provenientes de los satélites. Posteriormente se tomó varias mediciones con el propósito de comprobar que los datos de posicionamiento y velocidad sean reales.

Luego se activo la aplicación RastreoGPS en el dispositivo GPS, y se constató de que la señal se envía al servidor vía REST de manera satisfactoria.

Se constató que el servidor recibe las señales de los dispositivos GPS, de manera satisfactoria. Hasta aquí queda probada la comunicación de los dispositivos con el servidor.

Posteriormente, desde diferentes equipos se accedió a la dirección url <http://ec2-50-112-185-33.us-west-2.compute.amazonaws.com> del aplicativo, y se corroboró que en tiempo real pueden ver la posición de los vehículos.

Luego se corroboró la consistencia de datos, para verificar la fiabilidad del sistema, que también resulto satisfactorio.

6.1.4 CASOS DE USO DE PRUEBA

Para realizar estas pruebas se somete algunos casos de prueba que verifiquen la evaluación del sistema web, estas pruebas se realizaron de acuerdo a la programación del cronograma.

CONFIGURAR PARÁMETROS LÍMITE	
Descripción	El administrador o gerente general ingresa los parámetros que no deben transgredir los conductores, como son velocidad máxima, tiempo de parada máxima y máxima distancia que puede recorrer un vehículo al día.
Condiciones de ejecución	La condición de ejecución del caso de prueba es que el usuario haya sido validado como administrador o gerente.
Entrada	<ul style="list-style-type: none"> • Se pulsa "Mantenimientos" en el menú superior. • Se ingresa a "Configuración de parámetros límite". • El sistema muestra la interfaz propia de configuración

	<p>de parámetros límite.</p> <ul style="list-style-type: none"> • Ingresa los parámetros correspondientes. • Si ingresa mal un dato muestra una señal de alerta para que vuelva a reingresarlo. • Le da click en el botón "Modificar configuración". • Se guarda dicha configuración.
Resultado esperado	Los parámetros se almacenan en la base de datos y quedan establecidos para posteriores procesos.
Evaluación de prueba	Prueba superada con éxito.

Tabla 34: Caso de prueba "Configurar parámetros límite"

ASIGNAR CONDUCTOR A VEHÍCULO	
Descripción	El administrador asigna un conductor a un vehículo, dicho conductor no debe estar asignado a otro vehículo, de ser el caso debería desasignar el conductor del vehículo que lo tiene y luego asignarlo al vehículo deseado.
Condiciones de ejecución	La condición de ejecución del caso de prueba es que el usuario haya sido validado como administrador.
Entrada	<ul style="list-style-type: none"> • Se pulsa "Procesos" en el menú superior. • Se ingresa a "Asignar conductor a vehículo". • El sistema muestra la interfaz propia de la asignación de conductores a vehículos. • Se selecciona el vehículo al que se desee asignar otro conductor. • El sistema muestra información detallada de dicho vehículo. • El sistema muestra una lista conductores disponibles, es decir, que no hayan sido asignados previamente a ningún vehículo. • Se selecciona uno de los conductores disponibles.

	<ul style="list-style-type: none"> • Se almacena en la base de datos.
Resultado esperado	La asignación queda almacenada en la base de datos y quedan establecidos para posteriores procesos.
Evaluación de prueba	Prueba superada con éxito.

Tabla 35: Caso de prueba "Asignar conductor a vehículo"

ASIGNAR DISPOSITIVO GPS A VEHÍCULO	
Descripción	El administrador asigna un dispositivo GPS a un vehículo, dicho dispositivo no debe estar asignado a otro vehículo, de ser el caso debería desasignar el dispositivo del vehículo que lo tiene y luego asignarlo al vehículo deseado.
Condiciones de ejecución	La condición de ejecución del caso de prueba es que el usuario haya sido validado como administrador.
Entrada	<ul style="list-style-type: none"> • Se pulsa "Procesos" en el menú superior. • Se ingresa a "Asignar dispositivo a vehículo". • El sistema muestra la interfaz propia de la asignación de dispositivos a vehículos. • Se selecciona el vehículo al que se desee asignar otro dispositivo. • El sistema muestra información detallada de dicho vehículo. • El sistema muestra una lista dispositivos disponibles, es decir, que no hayan sido asignados previamente a ningún vehículo. • Se selecciona uno de los dispositivos disponibles. • Se almacena en la base de datos.
Resultado esperado	La asignación queda almacenada en la base de datos y quedan establecidos para posteriores procesos.
Evaluación de prueba	Prueba superada con éxito.

Tabla 36: Caso de prueba "Asignar dispositivo a vehículo"

RASTREAR VEHÍCULOS EN TIEMPO REAL	
Descripción	El operador, visualiza en un mapa de la interfaz las posiciones de los vehículos que están de servicio, observa en tiempo real su posición, orientación, si tiene pasajero, y si tiene una emergencia. En una tabla de la misma interfaz, muestra la velocidad, el tiempo de parada y la dirección por cada vehículo.
Condiciones de ejecución	Las condiciones de ejecución del caso de prueba es que el operador haya sido validado correctamente y esté activo.
Entrada	<ul style="list-style-type: none"> • Se pulsa "Visualizar unidades" en el menú superior. • Se ingresa a "Vehículos en tiempo real". • El sistema muestra una lista de los vehículos. • Selecciona los vehículos que desea rastrear. • El sistema muestra la interfaz propia del rastreo vehicular, mostrando los vehículos correspondientes, en sus posiciones y velocidad, tiempo de parada y estado del vehículo.
Resultado esperado	Se visualiza en un mapa de la interfaz, las ubicaciones de los vehículos en tiempo real.
Evaluación de prueba	Prueba superada con éxito.

Tabla 37: Caso de prueba "Rastrear vehículos en tiempo real"

MOSTRAR RUTAS SEGUIDAS POR LOS VEHÍCULOS	
Descripción	El operador, visualiza en un mapa de la interfaz las posiciones de los vehículos seleccionados en una determinada fecha y hora. Se muestra su posición en el mapa de la interfaz respectiva. En una tabla de la misma interfaz, se muestra el detalle, como es la hora, la velocidad, el tiempo de parada y dirección de la calle.
Condiciones de ejecución	Las condiciones de ejecución del caso de prueba es que el operador haya sido validado correctamente y esté activo.
Entrada	<ul style="list-style-type: none"> • Se pulsa "Visualizar unidades" en el menú superior. • Se ingresa a "Replay de vehículos". • El sistema muestra una lista de los vehículos. • Selecciona los vehículos que desea saber sus rutas. • Selecciona la fecha en la cual desea hacer la consulta. • Selecciona un intervalo de horas. • Pulsa el botón de "aceptar". • El sistema muestra en la interfaz una tabla con la descripción de las posiciones del vehículo, la hora, la velocidad, el tiempo de parada y dirección de la calle. • El sistema muestra en la interfaz un mapa con la representación de las rutas seguidas. • Selecciona la velocidad de reproducción del movimiento del vehículo. • Selecciona "play" • El sistema arranca la reproducción del movimiento del vehículo. • Selecciona "stop" • El sistema frena la reproducción del movimiento del vehículo.

	<ul style="list-style-type: none"> • Selecciona "quit" • El sistema borra de la interfaz el mapa y también la tabla de detalle.
Resultado esperado	Se visualiza en un mapa de la interfaz, las ubicaciones de los vehículos en determinada fecha e intervalo de horas.
Evaluación de prueba	Prueba superada con éxito.

Tabla 38: Caso de prueba "Mostrar rutas seguidas por los vehículos"

CONSULTAR REPORTES	
Descripción	El gerente general o administrador consulta reportes sobre las transgresiones de velocidad y tiempo de parada, y sobre el consumo de combustible.
Condiciones de ejecución	Las condiciones de ejecución del caso de prueba es que el usuario haya sido validado como administrador o gerente general.
Entrada	<ul style="list-style-type: none"> • Selecciona la opción de "Consultas" del menú superior. • Da click a la consulta que desea realizar. • El sistema muestra la respectiva interfaz. • Ingresa datos para la consulta. • Muestra el resultado en una tabla de la interfaz. • Si desea imprime la consulta • El sistema muestra una interfaz para imprimir la consulta.
Resultado esperado	Los parámetros se almacenan en la base de datos y se visualizan para los operadores en tiempo real.
Evaluación de prueba	Prueba superada con éxito.

Tabla 39: Caso de prueba "Consultar reportes"

CAMBIAR ESTADO DEL VEHÍCULO	
Descripción	El conductor ingresa su estado en la interfaz del dispositivo GPS.
Condiciones de ejecución	Las condiciones de ejecución del caso de prueba es que el conductor y el dispositivo GPS estén correctamente ingresados y activos en la base de datos del sistema.
Entrada	<ul style="list-style-type: none"> • El conductor selecciona su estado: “En servicio/Fuera de servicio”, “Con pasajero/Sin pasajero”, “Emergencia” y “Fuera de riesgo”. • El dispositivo GPS notifica a la estación central.
Resultado esperado	Los parámetros se almacenan en la base de datos y se visualizan para los operadores en tiempo real.
Evaluación de prueba	Prueba superada con éxito.

Tabla 40: Caso de prueba "Cambiar estado del vehículo"

CONCLUSIONES

- Se logró desarrollar una aplicación web para la gestión y monitoreo de una flota de vehículos, que hace uso de la tecnología GPS. No se pudo constatar fehacientemente que tanto la productividad como la seguridad mejorarán con la implantación del mismo, puesto que no se logró implantar el software por la informalidad que actualmente adolece la empresa caso de estudio del proyecto propuesto. Pero, vale decir que, con los reportes que el software genera además del monitoreo permanente, los administradores podrían sancionar las faltas, la impuntualidad e incumplimiento de los conductores, lo que reduciría los problemas de seguridad y productividad.
- Se analizó las herramientas de software y hardware que posibilitaron el desarrollo satisfactorio del prototipo propuesto.
- Se programó los dispositivos GPS para que transfieran datos al servidor automáticamente mediante la red GSM/GPRS, dicho objetivo se logró gracias al manejo de los servicios REST. Se constató que este servicio a parte de proporcionar mejor rendimiento que los servicios SOAP, es simple de diseñar e implementar, al minimizar el acoplamiento de datos entre el servicio y las instancias que lo consumen.
- Se implementó el centro de control en la nube virtual que ofrece Amazon. Esta nube virtual proporcionó la flexibilidad necesaria para la instalación de las dependencias que hacen posible la comunicación con los dispositivos GPS y los navegadores de las máquinas cliente.
- Se ha logrado implementar la transmisión de datos bidireccional cliente-servidor, para la comunicación en tiempo real, con la tecnología web socket, la cual brinda una enorme reducción de tráfico en la red, al no requerir que los navegadores realicen peticiones constantemente para mantenerse actualizados de los cambios que acontecen en el servidor.

RECOMENDACIONES

- Se sugiere realizar el diseño e implementación de un prototipo de sistema de localización de vehículos usando una red de radios y analizar en términos de seguridad, eficiencia y costos con respecto a los sistemas que usan la señal de internet.
- Se recomienda hacer un estudio comparativo sobre el desempeño de los gestores de base de datos geográficos, en términos de capacidad, eficiencia y rendimiento.
- Se recomienda la investigación de sistema de rastreo con implementaciones electrónicas, que le darían mayor funcionalidad y potencialidad.
- Se recomienda comparar la eficiencia de los servidores VPC en las aplicaciones en tiempo real.
- Se recomienda la investigación comparativa de las tecnologías en tiempo real, haciendo énfasis en la eficiencia, tiempo de respuesta, y costo de implementación.
- Se recomienda la realización de un análisis de costos y un estudio de mercado a nivel local, para verificar la factibilidad de la implantación de este tipo de sistemas.

REFERENCIAS

REFERENCIA BIBLIOGRÁFICA

- [RB001] ELIZONDO HERNÁNDEZ, Enriqueta; FONSECA MONTERUBIO, Angeles; MORALES DE HERNÁNDEZ, Martha (2009). *Observación y Orientación Comunicaciones*. México: Palibrio.
- [RB002] GARCÍA DE JALÓN, Javier; RODRIGUEZ, José Ignacio, Imaz, Aitor. (1999). *Aprenda Servlets de Java como si estuviera en primero*. San Sebastián, España:TECNUM.
- [RB003] GIRONÉS, Jesús Tomás. (2011). *El gran libro de ANDROID* (Primera edición ed.). Barcelona, España: Marcombo.
- [RB004] KEREKI, Federico. (2010). *Essential GWT: Building for the Web with Google Web Toolkit 2*. España: Pearson Education.
- [RB005] LETHAM, Lawrence. (2001). *GPS fácil* (Primera Edición ed.). Barcelona, España: Paidotribo.
- [RB006] RICHARDSON, Leonard; RUBY, Sam. (2007). *RESTfull Web Services* (Primera edición ed.). Estados Unidos:Mike Loukides.
- [RB007] OCDE - Organización para la Cooperación y Desarrollo Económicos. (2003). *Manual de Francasti*. París: FECYT - Federación Española Ciencia Y Tecnología.

REFERENCIAS DE PÁGINAS WEB

- [RW001] Citectran Centro de investigación tecnológica de transporte. Sistema de monitoreo vehicular TRANSGPS. 2010 [3 páginas] Disponible en: http://www.mrvisual.com/index.php?option=com_content&view=article&id=6&catid=3&Itemid=27. Consultado en marzo 22, 2013.
- [RW002] Hunter Lojack Soluciones tecnológicas de seguridad. Lojack stolen vehicle recovery system. 2012 [3 páginas] Disponible en: http://www.mrvisual.com/index.php?option=com_content&view=article&id=6&catid=3&Itemid=27. Consultado en diciembre 12, 2012.
- [RW003] OpenGTS. The OpenGTS Project. 2011. [2 páginas]. Disponible en: <http://opengts.sourceforge.net/>. Consultado en diciembre 2, 2012.
- [RW004] Informática ADSI. Proceso Unificado de Desarrollo de Software. 2010 [1 página] Disponible en: <http://informaticaadsi.blogspot.com/2010/03/informatica-adsi.html>. Consultado en diciembre 2, 2012.
- [RW005] HEILMANN, Chistian. Developing Sites With AJAX: Design Challenges and Common Issues. 2010 [1 página]. Disponible en: <http://coding.smashingmagazine.com/2010/02/10/some-things-you-should-know-about-ajax/> . Consultado en febrero 2, 2012.
- [RW006] Mozilla Developer Network y contribuyentes individuales. Web socket. 2011. [4 páginas]. Disponible en: <https://developer.mozilla.org/es/docs/WebSockets>. Consultado en enero 23, 2013.
- [RW007] Google Developers. Api de Google Maps/ Google Earth. [8 páginas]. Disponible en: http://www.google.com/intl/es_ALL/help/terms_maps.html . Consultado en noviembre 12, 2012.
- [RW008] Amazon Web Services Inc. o sus afiliados. Virtual Private Cloud (VPC) of Amazon Web Services. 2013 [3 páginas]. Disponible en: <http://aws.amazon.com/es/vpc/> . Consultado en febrero 27, 2013.

- [RW009] Sramp software developer. Traductor de sentencias \$GPRMC. 2013 [1 página]. Disponible en: <http://www.sramp.com/tools/nmeatranslator.html> . Consultado en diciembre 23, 2012.
- [RW010] Oracle Technology Network. Java Servlet Technology, 2012. [4 páginas]. Disponible en: <http://www.oracle.com/technetwork/java/index-jsp-135475.html> . Consultado en febrero 23, 2012.
- [RW011] Perfamino virtual. Definiciones informáticas, 2012 [5 páginas]. Disponible en: <http://www.pergaminovirtual.com.ar/definicion/> Consultado en abril 24,2012

ANEXOS

ANEXO 1: GLOSARIO DE TÉRMINOS

➤ AJAX

Acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones. Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales.

➤ Aplicación web

Es un sitio web que contiene páginas con contenido sin determinar parcialmente o en su totalidad. El contenido final de estas páginas se determina sólo cuando un visitante solicita una página del servidor Web.

➤ Controlador de base de datos

Es un software que actúa como intérprete entre una aplicación Web y una base de datos. Los datos de una base de datos se almacenan en un formato propio de dicha base de datos. Un controlador de base de datos permite a la aplicación Web leer y manipular datos que, de otro modo, resultarían indescifrables.

➤ CGI

Interfaz de entrada común, es una importante tecnología de la World Wide Web que permite a un cliente (navegador web) solicitar datos de un programa ejecutado en un servidor web. CGI especifica un estándar para transferir datos entre el cliente y el programa. Es un mecanismo de comunicación entre el

servidor web y una aplicación externa cuyo resultado final de la ejecución son objetos MIME. Las aplicaciones que se ejecutan en el servidor reciben el nombre de CGIs.

Las aplicaciones CGI fueron una de las primeras prácticas de crear contenido dinámico para las páginas web. En una aplicación CGI, el servidor web pasa las solicitudes del cliente a un programa externo. Este programa puede estar escrito en cualquier lenguaje que soporte el servidor, aunque por razones de portabilidad se suelen usar lenguajes de script. La salida de dicho programa es enviada al cliente en lugar del archivo estático tradicional.

CGI ha hecho posible la implementación de funciones nuevas y variadas en las páginas web, de tal manera que esta interfaz rápidamente se volvió un estándar, siendo implementada en todo tipo de servidores web.

➤ EC2 de Amazon

Es un servicio web que permite lanzar y gestionar instancias de servidores de Linux/UNIX y Windows en centros de datos de Amazon.

➤ EDGE

Es una tecnología de la telefonía móvil celular, que Enhanced Data rates for GSM of Evolution (Tasas de Datos Mejoradas para la evolución de GSM), es decir, el anterior mejorado, permite has un máximo de conexión de 236 Kbps, es decir 0,236 “Megas”. Actúa como puente entre las redes 2G y 3G. EDGE se considera una evolución del GPRS (General Packet Radio Service). debido a que puede funcionar en un máximo de 473,6 kbits por segundo. Esta tecnología funciona con redes GSM.

➤ GPRS

Servicio general de paquetes vía radio creado en la década de los 80 es una extensión del Sistema Global para Comunicaciones Móviles (Global System for Mobile Communications o GSM) para la transmisión de datos mediante conmutación de paquetes. Existe un servicio similar para los teléfonos móviles, el sistema IS-136. Permite velocidades de transferencia de 56 a 144 kbps.

➤ GSM

Global System for Mobile communications (Sistema Global para las comunicaciones Móviles), es el sistema de teléfono móvil digital más utilizado y el estándar para teléfonos móviles. La mayoría de las redes GSM utilizan 850MHz y 1900Mhz. Es de triple banda.

➤ MIME

Multipurpose Internet Mail Extensions (en español "extensiones multipropósito de correo de internet") son una serie de convenciones o especificaciones dirigidas al intercambio a través de Internet de todo tipo de archivos (texto, audio, vídeo, etc.) de forma transparente para el usuario. Una parte importante del MIME está dedicada a mejorar las posibilidades de transferencia de texto en distintos idiomas y alfabetos.

➤ Puerta de enlace

Es un dispositivo o una computadora que sirve como enlace entre dos redes informáticas, es decir, es el dispositivo que conecta y dirige el tráfico de datos entre dos redes o más. Generalmente en las casas u oficinas, ese dispositivo es el router y Cable-Modem o DSL-Modem que conecta la red local de la casa (LAN) con Internet (WAN). En las empresas, muchas veces es una computadora la que dirige el tráfico de datos entre la red local y la red exterior, y, generalmente, también actúa como servidor proxy y firewall.

Este dispositivo, al conectar dos redes de IP, poseerá una dirección IP privada: que servirá para identificarse dentro de la red local, y una dirección IP pública: que servirá para identificarse dentro de la red exterior.

➤ Relojes Atómicos

Tipo de reloj tipo de reloj que para alimentar su contador utiliza una frecuencia de resonancia atómica normal, mantienen una escala de tiempo continua y estable manteniendo una precisión de 10^{-9} segundos por día.

➤ Servicio de mapas

El servicio de mapas es la manera en la que se puede publicar mapas en la web. El mapa se crea en el servidor y, a continuación, se publica el servicio de mapas en la aplicación que lo invoque.

Un servicio de mapas facilita los mapas, las entidades y los datos de atributos disponibles dentro de muchos tipos de aplicaciones de cliente.

➤ Servidor de Aplicaciones

Es un software que ayuda al servidor Web a procesar las páginas que contienen scripts o etiquetas del lado del servidor. Cuando se solicita al servidor una página de este tipo, el servidor Web pasa la página al servidor de aplicaciones para su procesamiento antes de enviarla al navegador. Entre los servidores de aplicaciones más habituales se encuentran Macromedia ColdFusion, Macromedia JRun Server, Microsoft .NET Framework, IBM WebSphere y Apache Tomcat.

➤ Servidor web

Es un software que suministra páginas web en respuesta a las peticiones de los navegadores Web. La petición de una página se genera cuando un visitante hace clic en un vínculo de una página Web en el navegador, elige un marcador en el navegador o introduce un URL en el cuadro de texto Dirección del navegador. Entre los servidores Web más utilizados se encuentran Microsoft Internet Information Server, Microsoft Personal Web Server, Apache HTTP Server, Netscape Enterprise Server y Sun ONE Web Server.

➤ Sistema de administración de base de datos

(DBMS o sistema de base de datos) es un software que se utiliza para crear y manipular bases de datos. Entre los sistemas de bases de datos más habituales figuran Microsoft Access, Oracle 9i y MySQL.

➤ RIA

Son aplicaciones web que tienen la mayoría de las características de las aplicaciones de escritorio tradicionales. Estas aplicaciones utilizan un

navegador web estandarizado para ejecutarse y por medio de complementos o mediante una máquina virtual se agregan las características adicionales.

Normalmente en las aplicaciones web, hay una recarga continua de páginas cada vez que el usuario pulsa sobre un enlace. De esta forma se produce un tráfico muy alto entre el cliente y el servidor, llegando muchas veces a recargar la misma página con un cambio mínimo.

En los entornos RIA, en cambio, no se producen recargas de página, ya que desde el principio se carga toda la aplicación, y sólo se produce comunicación con el servidor cuando se necesitan datos externos como datos de una base de datos o de otros ficheros externos.

➤ SOAP

Protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

➤ Triangulación

Llamado también método de trilateración inversa, es un método que consiste en determinar la posición absoluta o coordenadas reales de un punto de medición, a partir de la posición relativa de dicho punto respecto a los tres satélites, y las coordenadas o posición de dichos satélites.

➤ URI

Aunque se acostumbra llamar URLs a todas las direcciones Web, URI es un identificador más completo, que organiza los recursos en forma de directorios.

➤ VPC

Es un entorno configurable y dinámico de recursos públicos en la nube que requiere el uso de protocolos de encriptación, túneles y otros procedimientos de seguridad para transferir datos entre una empresa privada y un proveedor de servicios en la nube. Además permite crear una topología de red virtual, en la que se incluyen subredes y enrutamientos.

➤ XMLHttpRequest

También referida como XMLHTTP (Extensible Markup Language / Hypertext Transfer Protocol), es una interfaz empleada para realizar peticiones HTTP y HTTPS a servidores Web. Para los datos transferidos se usa cualquier codificación basada en texto, incluyendo: texto plano, XML, JSON, HTML y codificaciones particulares específicas. La interfaz se implementa como una clase de la que una aplicación cliente puede generar tantas instancias como necesite para manejar el diálogo con el servidor.

ANEXO 2: GLOSARIO DE ACRÓNIMOS

AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
CORBA	Common Object Request Broker Architecture
CSS	Cascading Style Sheets
DCOM	Distributed Component Object Model
DOM	Modelo de Objetos del Documento
ECJ	Eclipse compiler for Java
EDI	Electronic Data Interchange
FTP	File Transfer Protocol
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPRMC	Standard Recommended Minimum Specific GNSS Data.
GWT	Google Web Toolkit
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
JDBC	Java Database connectivity
JDK	Java Development Kit
JDT	Java Development Toolkit
JSDK	Java Servlet Development Kit
JRE	Java Runtime Environment
JSNI	Interfaz Nativa Javascript
JSON	JavaScript Object Notation
JVM	Java Virtual Machine
MIME	Multipurpose Internet Mail Extensions
NDK	Native Development Kit
NMEA	National Marine Electronics Associations
OASIS	Organization for the Advancement of Structured Information Standards
PC	Personal computer
PDA	Personal digital assistant
RAM	Random Access Memory
REST	Representational State Transfer

RMI	Remote Method Invocation
RPC	Remote Procedure Calls
RIA	Rich Internet Applications
SAX	Simple API for XML
SDK	Software Developers Kit
SMS	Short Message Service
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
SSJS	Server-side Javascript
TCP	Transmission Control Protocol
UDDI	Universal Description Discovery and Integration
URL	Uniform resource locator
URI	Uniform resource identifier
W3C	World Wide Web Consortium
WSDL	Web Services Description Language
XML	eXtensible Markup Language

ANEXO 3: CÁLCULO DE CONSUMO DE COMBUSTIBLE

Para calcular el consumo de combustible en galones, se tiene la siguiente fórmula:

$$Cc=Dr/i$$

Donde:

Dr: Distancia Recorrida (Km)

i : Índice del consumo de Combustible (Km/Galón)

Este parámetro se puede calcular como indica el siguiente anexo. ANEXO3

Cc: Combustible total consumido (Galones)

Para calcular el costo total del combustible consumido:

$$Ct=Pc*Cc$$

Donde:

Pc : Precio de un galón de combustible

Ct : Costo Total

ANEXO 4: CÁLCULO DEL ÍNDICE DE CONSUMO DE COMBUSTIBLE (Km/Galón)

La mayoría de veces los indicadores del consumo de combustible que estiman los fabricantes de vehículos se encuentran por debajo del consumo real, esto debido a factores como el tráfico, el clima, el desgaste del vehículo, el tipo de carretera, etc. Para poder sacar una medida más o menos exacta de kilómetros que avanza el vehículo por galón, se recomienda hacer esta prueba realizando las actividades rutinarias en las condiciones más normales.

El procedimiento a seguir para calcular el índice de consumo de combustible es el siguiente:

1. Cargar más de 10 galones de gasolina al vehículo.
2. Anotar el kilometraje que indica el odómetro.
3. Realizar el recorrido rutinario con el vehículo.
4. Anotar el kilometraje que indica el odómetro al finalizar los 10 galones de gasolina.
5. Dividir el resultado entre 10.

Se recomienda realizar esta prueba como mínimo 10 veces siguiendo una rutina normal, mientras más veces se repita el procedimiento el resultado tendrá mayor precisión. Finalmente hallar el promedio de dichos resultados. La cifra resultante se puede comparar con la cifra de las especificaciones del vehículo.

ANEXO 5: REPORTE GENERADO DE EXCESO DE VELOCIDAD AL DÍA

26/03/13

Wrapper HTML for Rastro

EXCESOS DE VELOCIDAD DEL DÍA

FECHA: 06 de Enero de 2013

	Placa	Descripcion	Conductor	Hora	Velocidad (Km/h)	Direccion
1	QUA403	Automóvil Toyota Yaris Rojo-2005	Oswaldo Farfán Gutierrez	09:13:01	72.2571029663086	Centro Histórico, Cusco, Perú
2	QUA403	Automóvil Toyota Yaris Rojo-2005	Ulises Sulca Durán	09:24:01	68	Centro Histórico, Cusco, Perú
3	QUA403	Automóvil Toyota Yaris Rojo-2005	Ulises Sulca Durán	09:27:41	64.19999694824219	Centro Histórico, Cusco, Perú
4	QUA403	Automóvil Toyota Yaris Rojo-2005	Ulises Sulca Durán	09:28:03	72.2571029663086	Centro Histórico, Cusco, Perú
5	QUA404	Automóvil Hyundai I30 Verde-2005	Ramiro Uribe Izquierdo	09:16:05	72.2571029663086	Centro Histórico, Cusco, Perú
6	QUA405	Automóvil Hyundai Galloper Azul- 2005	Juan José Ccori Paucar	09:14:28	72.2571029663086	Centro Histórico, Cusco, Perú
7	QUA405	Automóvil Hyundai Galloper Azul- 2005	Juan José Ccori Paucar	09:18:28	72.2571029663086	Centro Histórico, Cusco, Perú
8	QUA406	Automóvil Hyundai I30 Amarillo- 2005	Antonio Puma Alvarado	09:25:10	68	Centro Histórico, Cusco, Perú
9	QUA406	Automóvil Hyundai I30 Amarillo- 2005	Antonio Puma Alvarado	09:25:33	64.19999694824219	Centro Histórico, Cusco, Perú
10	QUA406	Automóvil Hyundai I30 Amarillo- 2005	Antonio Puma Alvarado	09:25:56	72.2571029663086	Centro Histórico, Cusco, Perú

ANEXO 6: REPORTE GENERADO DE EXCESO DE VELOCIDAD DEL MES

26/03/13

Wrapper HTML for Rastreo

EXCESOS DE VELOCIDAD DEL MES

FECHA: Febrero del 2013

	Placa	Descripcion	Conductor	Hora	Fecha	Velocidad (Km/h)	Direccion
1	QUA403	Automóvil Toyota Yaris Rojo-2005	Oswaldo Farfán Gutiérrez	09:13:01	02/02/2013	72.2571029663086	Centro Histórico, Cusco, Perú
2	QUA403	Automóvil Toyota Yaris Rojo-2005	Ulises Sulca Durán	09:24:01	02/02/2013	68	Centro Histórico, Cusco, Perú
3	QUA403	Automóvil Toyota Yaris Rojo-2005	Ulises Sulca Durán	09:27:41	02/02/2013	64.19999694824219	Centro Histórico, Cusco, Perú
4	QUA403	Automóvil Toyota Yaris Rojo-2005	Ulises Sulca Durán	09:28:03	02/02/2013	72.2571029663086	Centro Histórico, Cusco, Perú
5	QUA404	Automóvil Hyundai I30 Verde-2005	Ramiro Uribe izquierdo	09:15:05	02/02/2013	72.2571029663086	Centro Histórico, Cusco, Perú
6	QUA405	Automóvil Hyundai Galloper Azul-2005	Juan José Ccori Paucar	09:14:28	02/02/2013	72.2571029663086	Centro Histórico, Cusco, Perú
7	QUA405	Automóvil Hyundai Galloper Azul-2005	Juan José Ccori Paucar	09:18:28	02/02/2013	72.2571029663086	Centro Histórico, Cusco, Perú
8	QUA406	Automóvil Hyundai I30 Amarillo-2005	Antonio Puma Alvarado	09:25:10	02/02/2013	68	Centro Histórico, Cusco, Perú
9	QUA406	Automóvil Hyundai I30 Amarillo-2005	Antonio Puma Alvarado	09:25:33	02/02/2013	64.19999694824219	Centro Histórico, Cusco, Perú
10	QUA406	Automóvil Hyundai I30 Amarillo-2005	Antonio Puma Alvarado	09:25:56	02/02/2013	72.2571029663086	Centro Histórico, Cusco, Perú

ANEXO 7: REPORTE GENERADO DE TIEMPO DE PARADA DEL DÍA

25/03/13

Wrapper HTML for Rastreo

EXCESOS DE TIEMPO DE PARADA DEL DÍA

FECHA: 06 de Enero de 2013

	Placa	Descripcion	Conductor	Hora	Tiempo Parada	Direccion
1	QUA403	Automóvil Toyota Yaris Rojo-2005	Oswaldo Farfán Gutiérrez	09:20:21	00:05:42	Centro Histórico, Cusco, Perú
2	QUA403	Automóvil Toyota Yaris Rojo-2005	Ulises Sulca Durán	09:24:01	00:04:02	Centro Histórico, Cusco, Perú
3	QUA404	Automóvil Hyundai I30 Verde-2005	Ramiro Uribe Izquierdo	09:21:35	00:05:02	Centro Histórico, Cusco, Perú
4	QUA405	Automóvil Hyundai Galoper Azul-2005	Juan José Coori Paucar	09:21:48	00:03:02	Centro Histórico, Cusco, Perú

ANEXO 8: REPORTE GENERADO DE TIEMPO PARADA DEL MES

26/03/13 Wrapper HTML for Rastreo

EXCESOS DE TIEMPO DE PARADA DEL MES

FECHA: Febrero del 2013

	Placa	Descripcion	Conductor	Hora	Fecha	Tiempo Parada	Direccion
1	QUA403	Automóvil Toyota Yaris Rojo-2005	Oswaldo Farián Gutierrez	09:20:21	02/02/2013	00:05:42	Centro Histórico, Cusco, Perú
2	QUA403	Automóvil Toyota Yaris Rojo-2005	Ulises Sulca Durán	09:24:01	02/02/2013	00:04:02	Centro Histórico, Cusco, Perú
3	QUA404	Automóvil Hyundai I30 Verde-2005	Ramiro Uribe Izquierdo	09:21:35	02/02/2013	00:05:02	Centro Histórico, Cusco, Perú
4	QUA405	Automóvil Hyundai Galloper Azul-2005	Juan José Ccori Paucar	09:21:48	02/02/2013	00:03:02	Centro Histórico, Cusco, Perú

ANEXO 9: REPORTE GENERADO DE REPLAY DE VEHÍCULOS

26/03/13

Wrapper HTML for Rastreo

REPLAY DE VEHÍCULOS

FECHA: 06 de Enero de 2013

HORA IN: 09:20:00 HORA OUT: 10:00:00

HORA EN QUE SE PARÓ EL REPRODUCTOR: 09:20:00

LISTA DE VEHÍCULOS

	Placa	Características	Kilometraje	Consumo de combustible(km/galón)	Costo de Combustible(soles/galón)	Color en Mapa
1	QUA403	Toyota Yaris Rojo	23000 52			13 #FF0000
2	QUA404	Hyundai B0 Verde	23000 50			18 #25A955
3	QUA405	Hyundai Galloper Azul	23000 49			14 #5366FF
4	QUA406	Hyundai B0 Amarillo	23000 55			14 #FF9900
5	QUA407	Nissan Terrano Acero	23000 57		15.399999618530273	#000000
6	88A403	Toyota Yaris Rojo	23000 53		13.199999803265137	#FF00FF
7	XPQ232	Nissan Terrano Lila	100 54			15 #636399
8	VG5232	Toyota Yaris Celeste	0 0			0 #630000

LISTA DE EVENTOS

Hora	Placa	Conductor	Color	Latitud	Longitud	Vel	Odometro	Distancia	TiempoParada	Direccion
09:20:11	QUA406	Antonio Puma Alvarado	#FF9900	-13.524262428283691	-71.97177124023438	19.780000228881836	10000	96 00:00:02		Centro Histórico, Cusco, Perú
09:20:21	QUA403	Oswaldo Farfán Guarez	#FF0000	-13.520079612731934	-71.97600555419922	38.349300384521484	10000	228 00:05:42		Centro Histórico, Cusco, Perú
09:20:34	QUA406	Antonio Puma Alvarado	#FF9900	-13.524408340454102	-71.97105407714844	17.104400634765625	10000	114 00:00:02		Centro Histórico, Cusco, Perú
09:20:43	QUA403	Oswaldo Farfán Guarez	#FF0000	-13.519906044006348	-71.97553934326172	35.69369888305664	10000	245 00:00:02		Centro Histórico, Cusco, Perú
09:20:57	QUA406	Antonio Puma Alvarado	#FF9900	-13.524513244628906	-71.97030639648438	50	10000	132 00:00:02		Centro Histórico, Cusco, Perú
09:21:05	QUA403	Oswaldo Farfán Guarez	#FF0000	-13.519874572753906	-71.97519869140625	33.03810119628906	10000	264 00:00:02		Centro Histórico, Cusco, Perú
09:21:27	QUA403	Oswaldo Farfán Guarez	#FF0000	-13.519739151000977	-71.974608375	30.38249969482422	10000	282 00:00:02		Centro Histórico, Cusco, Perú
09:21:35	QUA404	Ramiro Unibe Izquierdo	#25A955	-13.532002449035645	-71.9686279296875	51.62730026245117	10000	444 00:05:02		Centro Histórico, Cusco, Perú
09:21:48	QUA405	Juan José Coon Pauca	#6366FF	-13.524408340454102	-71.97088623046875	56.938499450683594	10000	858 00:03:02		Centro Histórico, Cusco, Perú
09:21:49	QUA403	Oswaldo Farfán Guarez	#FF0000	-13.51959228515625	-71.97435760498047	27.726900300708008	10000	300 00:00:02		Centro Histórico, Cusco, Perú
09:21:57	QUA404	Ramiro Unibe Izquierdo	#25A955	-13.532085418701172	-71.9682846089336	48.97169876098633	10000	460 00:00:02		Centro Histórico, Cusco, Perú
09:22:08	QUA405	Juan José Coon Pauca	#6366FF	-13.524492263793945	-71.97032928486797	54.282901763916016	10000	874 00:00:02		Centro Histórico, Cusco, Perú
09:22:11	QUA403	Oswaldo Farfán Guarez	#FF0000	-13.519477844238281	-71.97388458251953	25.071199417114258	10000	318 00:00:02		Centro Histórico, Cusco, Perú
09:22:19	QUA404	Ramiro Unibe Izquierdo	#25A955	-13.532075881958008	-71.9678726196289	46.31610107421875	10000	476 00:00:02		Centro Histórico, Cusco, Perú
09:22:28	QUA405	Juan José Coon Pauca	#6366FF	-13.524397850036621	-71.96998896191406	51.62730026245117	10000	890 00:00:02		Centro Histórico, Cusco, Perú
09:22:33	QUA403	Oswaldo Farfán Guarez	#FF0000	-13.519822120666304	-71.97378540039062	22.415599822998047	10000	336 00:00:02		Centro Histórico, Cusco, Perú
09:22:41	QUA404	Ramiro Unibe Izquierdo	#25A955	-13.532116889953613	-71.96754455566406	43.660499572753906	10000	492 00:00:02		Centro Histórico, Cusco, Perú
09:22:48	QUA405	Juan José Coon Pauca	#6366FF	-13.523835182189941	-71.9698486328125	48.97169876098633	10000	906 00:00:02		Centro Histórico, Cusco, Perú
09:22:55	QUA403	Oswaldo Farfán Guarez	#FF0000	-13.519811630249023	-71.973388671875	19.780000228881836	10000	354 00:00:02		Centro Histórico, Cusco, Perú

ANEXO 10: REPORTE GENERADO DE CONSUMO DE COMBUSTIBLE DEL MES

26/03/13 Wrapper HTML for Rastreo

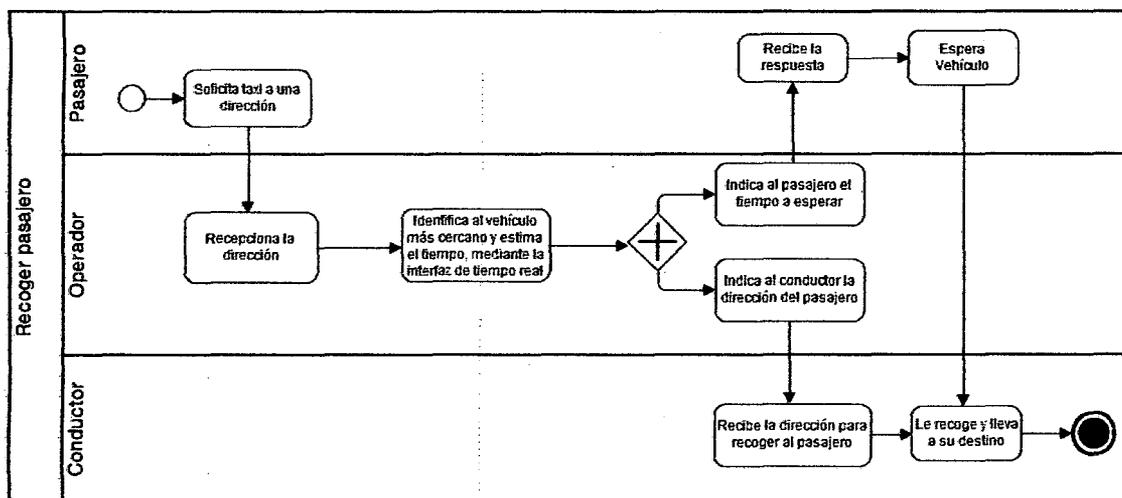
REPORTE DE CONSUMO DE COMBUSTIBLE DEL MES

FECHA: Febrero del 2013

No	No Placa	Descripcion	Kilometraje	Velocidad Promedio (Km/h)	Distancia Total Km	Consumo de combustible (km/galón)	Combustible Consumido (galones)	Costo Combustible (Soles/Galón)	Costo Total (Soles)
1	QUA-403	Automóvil Toyota Yaris Rfp-2005	23000	20	636	52	0	13	0.1599999964237213
2	QUA-404	Automóvil Hyundai I30 Verde-2005	23000	54	242	50	0	18	0.09000000357627869
3	QUA-405	Automóvil Hyundai Galloper Azul-2005	23000	30	1018	49	0	14	0.28999999165534973
4	QUA-406	Automóvil Hyundai I30 Amarillo-2005	23000	52	240	55	0	14	0.05999999865889549

ANEXO 10: MODELO DE NEGOCIO IMPLEMENTANDO EL SISTEMA

Se puede apreciar mediante el siguiente diagrama la mejora sustancial de implementar el sistema respecto del modelo de negocio del cual se partió en la sección 3.4 del presente documento.



Como se puede apreciar el proceso se lleva a cabo con eficiencia y fluidez, el operador define directamente el vehículo más cercano y simplemente notifica al conductor la dirección y al pasajero el tiempo que debe esperar, evitando así que el cliente se sienta defraudado por una espera vana.

La implementación de este sistema proporcionaría a la empresa una ventaja competitiva bastante considerable, ya los clientes recibirían un mejor servicio, confiable y puntual. Ante tales ventajas los clientes aumentarían, por el servicio que al momento ninguna operadora de taxis a nivel local ofrece, respuestas inmediatas y servicio puntual.

En lo referente a la seguridad, los conductores al saberse rastreados no despilfarrarían los recursos en sus intereses personales, y serían puntuales puesto que el sistema almacena los detalles de su recorrido, generando así la ansiada productividad.

Los operadores actualmente no pueden determinar si sus conductores transgreden normas, con la implementación de este sistema, los operadores identificarían a estos últimos, proporcionando así mayor seguridad para sus clientes.

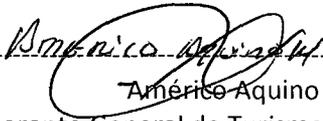
CERTIFICADO

LA EMPRESA DE TRANSPORTE "TURISMO ÁNGELES DEL CUSCO SAC" CERTIFICA:

Que la señorita **Mary Helen Conza Berrocal** identificada con DNI N° 46019594, desarrolló **UNA APLICACIÓN WEB ORIENTADA A SERVICIOS PARA EL MONITOREO DE UNA FLOTA DE VEHÍCULOS HACIENDO USO DE LA TECNOLOGÍA GPS**, y se realizó una prueba del sistema alcanzado los objetivos establecidos.

Se emite el siguiente certificado para los fines que vean por conveniente.

Cusco, 20 de Junio del 2013.



Américo Aquino Merma
Gerente General de Turismo Ángeles Cusco SAC
RUC: 2049039439