

# **HORUS - Sistema Prototipo de Monitoreo y Control Vehicular Asistido por *GPS***



## **ANEXOS**

**Juan Andrés Cárdenas Díaz  
Libardo Andrés Samboní Rubio**

*Universidad del Cauca*

**Facultad de Ingeniería Electrónica y Telecomunicaciones  
Departamento de Telecomunicaciones  
GRIAL – Grupo de Radio e InALámbricas  
Señales y Sistemas de Acceso y Difusión Basados en Radio.  
Popayán, Noviembre de 2010**

# **HORUS - Sistema Prototipo de Monitoreo y Control Vehicular Asistido por GPS**



Trabajo de Grado para optar al Título de Ingeniero en Electrónica y Telecomunicaciones

## **ANEXOS**

**Juan Andrés Cárdenas Díaz  
Libardo Andrés Samboní Rubio**

Director: Ing. Víctor Manuel Quintero Flórez

*Universidad del Cauca*

**Facultad de Ingeniería Electrónica y Telecomunicaciones  
Departamento de Telecomunicaciones  
GRIAL – Grupo de Radio e InALámbricas  
Señales y Sistemas de Acceso y Difusión Basados en Radio.  
Popayán, Noviembre de 2010**

# TABLA DE CONTENIDO

ANEXO A. IMPLEMENTACIÓN DETALLADA DE HORUS.....	1
A.1 ASPECTOS COMPLEMENTARIOS DEL CAPÍTULO 1 – MARCO TEÓRICO.....	1
A.1.1 Funcionamiento del GPS .....	1
A.1.2 Elementos comunes del software CAD .....	4
A.2 ASPECTOS COMPLEMENTARIOS DEL CAPÍTULO 2 - DESCRIPCIÓN DETALLADA DE HORUS .....	6
A.2.1 Descripción Detallada de los Casos de Uso.....	6
A.3 ASPECTOS COMPLEMENTARIOS DEL CAPÍTULO 3 – IMPLEMENTACIÓN DE HORUS .....	25
A.3.1 Módulo GPS.....	25
A.3.1.1 Descripción de pines del módulo GPS. ....	25
A.3.1.2 Protocolo de comunicación del módulo GPS.....	26
A.3.1.3 Configuración del GPS.....	27
A.3.2 Códigos de Línea.....	30
A.3.2.1 Codificación Digital Unipolar.....	30
A.3.2.2 Codificación Digital Polar .....	31
A.3.2.2.1 No Retorno a Cero (NRZ).....	31
A.3.2.2.2 Retorno a Cero (RZ, Return to Zero) .....	32
A.3.2.2.3 Bifase .....	32
A.3.2.3 Codificación Digital Bipolar.....	32
A.3.2.3.1 Código de Alta Densidad de 3 Ceros (HDB3, High Density Bipolar) ....	33
A.3.2.3.1.1 Características y Funcionamiento de HDB3 .....	33
A.3.2.3.1.2 Generación de la codificación HDB3 .....	35
A.3.3 Descripción de Tarjetas.....	39
A.3.3.1 Tarjetas Unidad Móvil.....	39
A.3.3.1.1 Módulos Software de la Unidad Móvil.....	39
A.3.3.1.2 Tarjeta de Control y Despliegue Móvil .....	42
A.3.3.1.3 Tarjeta de Adaptación Radio .....	42
A.3.3.1.3.1 Generador de niveles HDB3.....	43
A.3.3.1.3.2 Filtro Limitador de Banda.....	44
A.3.3.1.3.3 Filtro eliminador DC.....	46
A.3.3.1.3.4 Amplificador.....	47
A.3.3.1.3.5 Detector de pulsos positivos y detector de pulsos negativos .....	48
A.3.3.1.3.6 Circuito Impreso de la Tarjeta de Adaptación Radio. ....	49
A.3.3.2 Tarjeta de la Unidad de Control, Conmutación y	

Acondicionamiento de la Estación de Control (UCCA-EC).....	50
A.3.3.2.1 Módulos Software de la tarjeta UCCA-EC .....	50
A.3.3.2.2 Circuito Impreso de la Tarjeta UCCA-EC .....	51
A.3.4 Medición de la Desviación de Frecuencia Radio Motorola PRO 3100 .....	51
A.3.4.1 Medición con Señal de Entrada Binaria.....	51
A.3.4.2 Medición de la desviación de frecuencia con señal de entrada HDB3 .....	57
A.3.5 Descripción Detallada de las Interfaces Implementadas en la Arquitectura de HORUS para el Despliegue de Servicios .....	62
A.3.5.1 Navegador Web .....	63
A.3.5.2 Servlets .....	63
A.3.5.3 Java Server Pages.....	64
A.3.5.4 Java Script .....	65
A.3.5.5 Google Maps JavaScript API V3. ....	65
A.3.5.5.1 Clase LatLng.....	66
A.3.5.5.2 Clase MapTypeld. ....	66
A.3.5.5.3 Clase Map.....	67
A.3.5.5.4 Clase Marker.....	69
A.3.5.5.5 Clase Polyline. ....	71
A.3.5.5.6 Clase InfoWindow. ....	72
A.3.5.6 DaoGen.....	74
A.3.5.7 JDBC .....	74
A.3.5.8 GiovyNet Driver.....	75
A.3.5.9 RS-232.....	75
A.3.5.10 Servicios Web. ....	76
A.3.5.11 MidMaps .....	76
A.3.5.11.1 Clase GoogleMapsCoordinates.....	77
A.3.5.11.2 Clase GoogleMapsMarker .....	77
A.3.5.11.3 Clase GoogleMapsPath.....	78
A.3.5.11.4 Clase GoogleStataicMap.....	79
A.3.5.12 JSMS .....	81
A.3.5.13 SMSC.....	81
A.3.6 Diseño Físico de la Base de Datos Implementada en HORUS.....	82
A.3.6.1 Tabla Usuarios.....	82
A.3.6.2 Tabla Vehículos .....	83
A.3.6.3 Tabla Clientes .....	83
A.3.6.4 Tabla Conductores.....	83
A.3.6.5 Tabla Direcciones .....	84
A.3.6.6 Tabla Tareas.....	84
A.3.6.7 Tabla Emergencias .....	85
A.3.6.8 Tabla Recorridos.....	85
A.3.7 Descripción del Funcionamiento del Protocolo de HORUS .....	86



A.4 ASPECTOS COMPLEMENTARIOS DEL CAPÍTULO 4 – PRUEBAS Y RESULTADOS.....	89
A.4.1 Relación de Potencia Recibida con Voltaje Medido en el Pin RSSI del Radio Motorola Pro 3100.....	89
A.4.1.1 Cálculo de Eb/No .....	93
A.4.2 Formula de Haversine.....	96
A.4.3 Herramientas Utilizadas Durante la Fase de Implementación y Pruebas del Sistema.....	97
 ANEXO B. MANUAL DE USUARIO .....	 102
B.1 ASPECTOS GENERALES.....	102
B.1.1 Inicio de Sesión.....	102
B.1.2 Consultar Posición Geográfica .....	103
B.1.3 Graficar Recorrido.....	104
B.1.4 Consultar Información asociada al vehículo .....	105
B.2 INSTRUCCIONES DE USO PARA ADMINISTRADORES.....	105
B.2.1 Crear cuentas de usuario .....	107
B.2.2 Modificar cuentas de usuario .....	108
B.2.3 Eliminar cuentas de usuario .....	109
B.3 INSTRUCCIONES DE USO PARA CLIENTES .....	109
B.3.1 Cliente Web .....	109
B.3.2 Cliente Móvil .....	110
B.3.2.1 Validación de Usuario .....	111
B.3.2.2 Consultar posición geográfica .....	112
B.3.2.3 Graficar recorrido .....	114
B.3.2.4 Consultar información asociada al vehículo .....	116
B.4 INSTRUCCIONES DE USO PARA OPERADORES .....	116
B.4.1 Localizar direcciones.....	117
B.4.2 Despachar Vehiculos .....	117
B.4.3 Enviar mensajes a unidades móviles .....	120
B.4.4 Gestión de tareas.....	120
B.4.4.1 Buscar tareas.....	121
B.4.4.2 Agregar Tareas .....	122
B.4.4.3 Modificar Tareas .....	122
B.4.4.4 Eliminar Tareas.....	123
B.5 INSTRUCCIONES DE USO PARA CONDUCTORES.....	124
B.5.1 Funciones de la Interfaz Física de la Unidad Móvil.....	125



---

B.5.1.1 Inicio de Sesión.....	125
B.5.1.2 Visualización de Mensaje de Texto .....	127
B.5.1.3 Cierre de Sesión .....	128
B.5.1.4 Envío de la Señal de Pánico .....	128

## LISTADO DE FIGURAS

Figura A.1. Representación de la localización de un satélite. ....	2
Figura A.2. Intercepción de dos esferas. ....	2
Figura A.3. Posición en 3D a partir de la localización de 3 satélites. ....	2
Figura A.4. Principio de funcionamiento del GPS. ....	3
Figura A.5. Interacción de las capacidades y características de un sistema CAD. ....	6
Figura A.6. Diagrama detallado de casos de uso para HORUS.....	7
Figura A.7. Diagrama de pines del módulo GPS EM-406A.....	25
Figura A.8. Conexión del módulo GPS a un computador usandoun MAX232 y un conector DB9 Hembra. ....	28
Figura A.9. Configuración inicial de SiRF Demo. ....	28
Figura A.10. Parámetros GPS mostrados por el software SiRF Demo. ....	29
Figura A.11. Configuración del módulo GPS. ....	30
Figura A.12. NRZ Unipolar. ....	31
Figura A.13. Codificación bipolar de una señal binaria. ....	32
Figura A.14. Ancho de banda utilizado por codificaciones tipo NRZ Unipolar, AML y HDB3. ....	33
Figura A.15. Posición de bit donde se generan las violaciones. ....	34
Figura A.16. Codificación HDB3. ....	34
Figura A.17. Conversor Digital-Analógico. ....	35
Figura A.18. Amplificador operacional en configuración sumador inversor. ....	35
Figura A.19. Módulo generador de los niveles HDB3. ....	36
Figura A.20. Módulos software que componen a la unidad móvil. ....	40
Figura A.21. Circuito impreso de la tarjeta de control y despliegue móvil. ....	42
Figura A.22 (a). Voltaje de salida 0.249 v a partir de bits 0 – 0 de entrada. ....	43
Figura A.22 (b). Voltaje de salida -0.249 V a partir de bits 1 – 0 de entrada.....	25
Figura A.22 (c). Voltaje de salida aproximadamente 0 V a partir de bits 0 – 1 de entrada.....	25
Figura A.23. Simulación del filtro limitador de banda. ....	45
Figura A.24 (a). Señal de entrada (1000 Hz) vs. Señal de salida del filtro limitador de banda. ....	45
Figura A.24 (b). Señal de entrada (2000 Hz) vs. Señal de salida del filtro limitador de banda.....	26
Figura A.24 (c). Señal de entrada (3000 Hz) vs. Señal de salida del filtro limitador de banda.....	27
Figura A.24 (d). Señal de entrada (5000 Hz) vs. Señal de salida del filtro limitador de banda.....	30
Figura A.24 (e). Señal de entrada (10000 Hz)vs. Señal de salida del filtro limitador de banda.....	30

Figura A.14 (f). Señal de entrada (15000 Hz) vs. Señal de salida del filtro limitador de banda.....	31
Figura A.25. Simulación del filtro eliminador DC.....	46
Figura A.26. Señal de entrada vs. Señal de salida filtro eliminador DC. ....	47
Figura A.27. Simulación del Amplificador. ....	47
Figura A.28. Simulación del detector de pulsos positivos. ....	48
Figura A.29. Señal de entrada con valor $\pm 5V$ (amarilla).....	48
Figura A.30. Señal de entrada vs. señal de salida detector de pulsos positivos (Simulación).....	49
Figura A.31. Circuito impreso de la tarjeta de adaptación radio.....	49
Figura A.32. Módulos software que componen la tarjeta de la UCCA-EC. ....	50
Figura A.33. Circuito impreso de la tarjeta de la UCCA-EC.....	51
Figura A.34. Esquema de conexión del generador de señales, radio convencional de dos vías y analizador de redes.....	52
Figura A.35 (a). Frecuencia de la señal portadora 151.1619 MHz. (Span 1 MHz) .....	52
Figura A.35 (b). Frecuencia de la señal portadora 151.16266 MHz. (Span 20 KHz).....	52
Figura A.36 (a). Espectro de la señal al transmitir señal binaria de 200 mVPP y frecuencia 300 Hz. (Función MAX HOLD) .....	53
Figura A.36 (b). Espectro de la señal al transmitir señal binaria de 200 mVPP y frecuencia 300 Hz. (Función NORMAL).....	53
Figura A.37. Señales moduladas con señal de entrada 200 mVPP (curva azul) y 400 mVPP (curva negra).....	53
Figura A.38 (a). Nivel de señal de entrada: 400 mV. Desviación de frecuencia: 2.554 KHz. (MAX HOLD).....	54
Figura A.38 (b). Nivel de señal de entrada: 400 mV. Desviación de frecuencia: 2.554 KHz. (NORMAL).....	54
Figura A.38 (c). Nivel de señal de entrada: 500 mV. Desviación de frecuencia: 2.954 KHz. (MAXHOLD).....	54
Figura A.38 (d). Nivel de señal de entrada: 500 mV. Desviación de frecuencia: 2.954 KHz. (NORMAL).....	54
Figura A.38 (e). Nivel de señal de entrada: 600 mV. Desviación de frecuencia: 3.513 KHz. (MAX HOLD).....	55
Figura A.38 (f). Nivel de señal de entrada: 600 mV. Desviación de frecuencia: 3.513 KHz. (NORMAL).....	55
Figura A.38 (g). Nivel de señal de entrada: 800 mV. Desviación de frecuencia: 4.311KHz. (MAX HOLD).....	55
Figura A.38 (h). Nivel de señal de entrada: 800 mV. Desviación de frecuencia: 4.3 KHz. (NORMAL).....	55
Figura A.38 (i). Nivel de señal de entrada: 1000 mV. Desviación de frecuencia: 5.908 KHz. (MAX HOLD).....	56



Figura A.38 (j). Nivel de señal de entrada: 1000 mV. Desviación de frecuencia: 5.908 KHz. (NORMAL).....	56
Figura A.38 (k). Nivel de señal de entrada: 1200 mV. Desviación de frecuencia: 7.026 KHz. (MAX HOLD).....	56
Figura A.38 (l). Nivel de señal de entrada: 1200 mV. Desviación de frecuencia: 7.026 KHz. (NORMAL).....	56
Figura A.39. Medición 1. Señal modulada que presenta los estados HDB3 al transmitir secuencia de ceros consecutivos con una velocidad de 333 bps y 500 mVPP. ....	57
Figura A.40 (a). Medición 2. ....	58
Figura A.40 (b). Medición 3.....	58
Figura A.40 (c). Medición 4. ....	59
Figura A.41. Espectro de la señal presenta los estados HDB3 al transmitir secuencia de unos consecutivos con una velocidad de 333 bps y 500 mVPP. ....	60
Figura A.42. Valores de frecuencia medidos al ubicar marcadores M1 y M2 a 30 dB por debajo de los máximos de la señal. ....	61
Figura A.43. Valores de frecuencia medidos al ubicar marcadores M1 y M2 a 20 dB por debajo de los máximos de la señal. ....	62
Figura A.44. Intercambio de tramas entre Servidor, Central y Móvil. ....	87
Figura A.45. Estructura de la trama definida en HORUS. ....	87
Figura A.46. Medidor de intensidad de campo y antena Yaesu para FT411.....	90
Figura A.47. Montaje de los equipos para realizar la prueba de medición de campo eléctrico.....	90
Figura A.48. Implementación en Java de la formula de Haversine. ....	97
Figura B.1. Interfaz de HORUS para validar el ingreso de los usuarios al sistema. ....	102
Figura B.2 (a). Error generado al seleccionar incorrectamente el tipo de usuario.....	103
Figura B.3 (b). Error generado al ingresar incorrectamente la contraseña de la cuenta.....	103
Figura B.4 (c). Error generado al ingresar incorrectamente el nombre de la cuenta. .	103
Figura B.5. Icono para consultar posición geográfica. ....	103
Figura B.6. Servicio de HORUS para consultar posición geográfica.....	104
Figura B.7. Icono para graficar recorrido. ....	104
Figura B.8. Servicio de HORUS para graficar el recorrido realizado por un vehículo. ....	104
Figura B.9. Icono para consultar información asociada al vehículo. ....	105
Figura B.10. Interfaz web para consultar la información asociada a un vehículo. ....	105
Figura B.11. Interfaz para administración de vehículos. ....	106
Figura B.12. Alerta implementada por el sistema para notificar un error en	

el proceso de localización de vehículos. ....	106
Figura B.13. Botón para ingresar a la interfaz de gestión de usuarios. ....	107
Figura B.14. Interfaz de HORUS para gestión de usuarios. ....	107
Figura B.15 (a). Notificación de éxito durante el proceso de agregar usuarios. ....	108
Figura B.15 (b). Notificación de éxito durante el proceso de agregar usuarios. ....	108
Figura B.16. Elementos implementados para facilitar la búsqueda de usuarios. ....	108
Figura B.17(a). Tipo de usuario incorrecto. ....	108
Figura B.17(b). Usuario no encontrado. ....	108
Figura B.18 (a). Notificación de éxito en el proceso de modificar usuarios. ....	109
Figura B.18 (b). Notificación de erro en el proceso de modificar usuarios. ....	109
Figura B.19 (a). Notificación de éxito en el proceso de eliminar usuarios. ....	109
Figura B.19 (b). Notificación de error en el proceso de eliminar usuarios. ....	109
Figura B.20. Interfaz web para clientes. ....	110
Figura B.21. Partes de un teléfono celular. ....	110
Figura B.22 (a). Validación de usuarios. ....	111
Figura B.22 (b). Conexión con el servidor. ....	111
Figura B.23. Proceso de validación exitoso. ....	112
Figura B.24. Alarmas utilizadas para reportar errores durante el proceso de validación desde la aplicación móvil. ....	112
Figura B.25 (a). Selección del tipo de mapa. ....	113
Figura B.25 (b). Consultar posición. ....	113
Figura B.25 (c). Mapa con la posición del vehículo. ....	113
Figura B.26. Menú de opciones. ....	114
Figura B.27. Error en la conexión con el servidor de mapas. ....	114
Figura B.28 (a). Visualizar Recorrido. ....	115
Figura B.28 (b). Mapa con el recorrido del vehículo. ....	115
Figura B.29 (a). Información del vehículo. ....	116
Figura B.29 (b). Información del conductor. ....	116
Figura B.30. Interfaz web para operadores de HORUS. ....	117
Figura B.31. Elementos involucrados en la localización de direcciones. ....	117
Figura B.32. Localización de direcciones. ....	118
Figura B.33. Elementos involucrados en el proceso de despacho vehicular. ....	120
Figura B.34. Elementos involucrados en el proceso de envió de mensajes, ....	120
Figura B.35. Interfaz para gestión de tareas. ....	121
Figura B.36. Elementos involucrados en la búsqueda de tareas. ....	121
Figura B.37. Error en búsqueda de tareas asignadas. ....	121
Figura B.38. Elementos involucrados durante el proceso de asignación de tareas desde la aplicación web. ....	122
Figura B.39 (a). Proceso de asignación de tareas exitoso. ....	122
Figura B.39 (b). Error durante l proceso de asignación de tarea. ....	122
Figura B.40. Elementos involucrados en el proceso de modificar tareas. ....	123
Figura B.41 (a). Proceso de modificación de tareas exitoso. ....	123

Figura B.41 (b). Error durante el proceso de modificación de tareas .....	123
Figura B.42 (a). Notificación de éxito durante el proceso de eliminar tareas.....	123
Figura B.42 (b). Notificación de error durante el proceso de eliminar tareas.....	123
Figura B.43. Interfaz física de la unidad móvil. ....	124
Figura B.44. Mensaje “Buscando Red” de inicio de unidad móvil. ....	125
Figura B.45. Inicio de sesión. ....	125
Figura B.46. Pantalla de ingreso de identificador de usuario. ....	126
Figura B.47. Pantalla de ingreso de contraseña de usuario.....	126
Figura B.48. Pantalla de inicio se sesión exitosa. ....	127
Figura B.49. Pantalla por defecto del sistema con identificación de usuario validada. ....	127
Figura B.50. Pantalla de nuevo mensaje de texto. ....	127
Figura B.51. Pantalla de cierre se sesión. ....	128
Figura B.52. Pantalla de confirmación de cierre de sesión. ....	128
Figura B.53. Pantalla de cancelación de cierre de sesión.....	128

## LISTADO DE TABLAS

Tabla A.1 (a). Descripción detallada para el caso de uso Validar Usuario.....	8
Tabla A.1 (b). Descripción detallada para el caso de uso Consultar Posición Geográfica.....	<b>¡Error!</b>
<b>Marcador no definido.</b>	
Tabla A.1 (c). Descripción detallada para el caso de uso Desplegar Recorrido.....	<b>¡Error! Marcador no definido.</b>
Tabla A.1 (d). Descripción detallada para el caso de uso Buscar Dirección.....	<b>¡Error! Marcador no definido.</b>
Tabla A.1 (e). Descripción detallada para el caso de uso Despachar Vehículo.....	<b>¡Error! Marcador no definido.</b>
Tabla A.1 (f). Descripción detallada para el caso de uso Consultar Información Asociada al Vehículo.....	<b>¡Error! Marcador no definido.</b>
Tabla A.1 (g). Descripción detallada para el caso de uso Enviar Mensajes.....	<b>¡Error! Marcador no definido.</b>
Tabla A.1 (h). Descripción detallada para el caso de uso Solicitar Información vía SMS.....	<b>¡Error! Marcador no definido.</b>
Tabla A.1 (i). Descripción detallada para el caso de uso Validar Conductor.....	<b>¡Error! Marcador no definido.</b>
Tabla A.1 (j). Descripción detallada para el caso de uso Consultar Coordenadas.....	<b>¡Error! Marcador no definido.</b>
Tabla A.1 (k). Descripción detallada para el caso de uso Activar Alarma.....	<b>¡Error! Marcador no definido.</b>
Tabla A.1 (l). Descripción detallada para el caso de uso Buscar Cuenta Usuario.....	<b>¡Error! Marcador no definido.</b>
Tabla A.1 (m). Descripción detallada para el caso de uso Crear Cuenta.....	<b>¡Error! Marcador no definido.</b>
Tabla A.1 (n). Descripción detallada para el caso de uso Modificar Cuenta.....	<b>¡Error! Marcador no definido.</b>
Tabla A.1 (o). Descripción detallada para el caso de uso Eliminar	

Cuenta.....**¡Error! Marcador no definido.**

Tabla A.1 (p). Descripción detallada para el caso de uso Buscar Tareas.....**¡Error! Marcador no definido.**

Tabla A.1 (q). Descripción detallada para el caso de uso Asignar Tareas.....**¡Error! Marcador no definido.**

Tabla A.1 (r). Descripción detallada para el caso de uso Modificar Tarea.....**¡Error! Marcador no definido.**

Tabla A.1 (s). Descripción detallada para el caso de uso Eliminar Tarea.  
.....**¡Error! Marcador no definido.**

Tabla A.2. Funciones de los pines del módulo GPS. .... 26

Tabla A.3. Mensajes de salida NMEA. .... 26

Tabla A.4. Descripción de los valores de un mensaje RMC. .... 27

Tabla A.5. Voltajes de salida respecto a bits de entrada. .... 38

Tabla A.6. Voltajes de salida en función a los bits de entrada del  
amplificador operacional. .... 43

Tabla A.7. Desviación de frecuencia para diferentes voltajes pico a pico  
de la de señal de datos de entrada..... 57

Tabla A.8. Frecuencias correspondientes a estados HDB3..... 58

Tabla A.9. Resumen de desviación de frecuencia con secuencia de ceros  
como señal de entrada. .... 59

Tabla A.10. Frecuencias correspondientes a estados HDB3..... 60

Tabla A.11. Métodos de la clase *LatLng* implementados en Horus. .... 66

Tabla A.12. Constantes de la clase *MapTypeId* implementadas en Horus. .... 67

Tabla A.13. Métodos de la clase *Map* implementados en Horus. .... 67

Tabla A.14. Eventos de la clase *Map* implementados en Horus. .... 68

Tabla A.15. Propiedades de la clase *Map* implementadas en Horus. .... 69

Tabla A.16. Métodos de la clase *Marker* implementados en Horus. .... 70

Tabla A.17. Eventos de la clase *Marker* implementados en Horus. .... 70

Tabla A.18. Propiedades de la clase *Marker* implementadas en Horus. .... 71

Tabla A.19. Métodos de la clase *Polyline* implementados en Horus..... 71

Tabla A.20. Eventos de la clase *Polyline* implementados en Horus..... 72

Tabla A.21. Propiedades de la clase *Polyline* implementadas en Horus..... 72

Tabla A.22. Métodos de la clase *InfoWindow* implementados en Horus..... 73

Tabla A.23. Eventos de la clase *InfoWindow* implementados en Horus..... 73

Tabla A.24. Propiedades de la clase *InfoWindow* implementados en Horus..... 74

Tabla A.25. Características de las normas definidas para comunicación serial. .... 76

Tabla A.26. Métodos de la clase *GoogleMapsMarker* implementados en Horus. .... 78

Tabla A.27. Atributos de la clase *GoogleMapsMarker* implementados en Horus..... 78

Tabla A.28. Métodos de la clase *GoogleMapsPath* implementados en Horus..... 78

Tabla A.29. Métodos de la clase *GoogleStaticMap* implementados en Horus. .... 80

Tabla A.30. Atributos de la clase <i>GoogleStaticMap</i> implementados en Horus.....	81
Tabla A.31. Relación entre voltaje del pin RSSI y el nivel medido de campo eléctrico. ....	91
Tabla A.32. Relación entre voltaje del pin RSSI, nivel medido de campo eléctrico y potencia de recepción.....	92
Tabla A.33. Relación entre voltaje del pin RSSI y la potencia de recepción. ....	92
Tabla A.34. Potencia de recepción en prueba de vehículo con unidad móvil alejado del lóbulo principal de antena de estación de control. ....	93
Tabla A.35. Relación Eb/No para las locaciones de la prueba de vehículo con unidad móvil en dirección al lóbulo principal de antena de estación de control.....	94
Tabla A.36. Potencia de recepción en prueba de vehículo con unidad móvil alejado del lóbulo principal de antena de estación de control. ....	95
Tabla A.37. Relación Eb/No para las locaciones de la prueba de vehículo con unidad móvil alejado del lóbulo principal de antena de estación de control.....	95
Tabla A.38. Herramientas software utilizadas durante la fase de implementación y pruebas de HORUS. ....	101
Tabla B.1. Iconos implementados por HORUS para identificar lugares. ....	119
Tabla B.2. Asignación de caracteres del teclado alfanumérico. ....	126

# ANEXO A

## IMPLEMENTACIÓN DETALLADA DE HORUS

### A.1 ASPECTOS COMPLEMENTARIOS DEL CAPÍTULO 1 – MARCO TEÓRICO

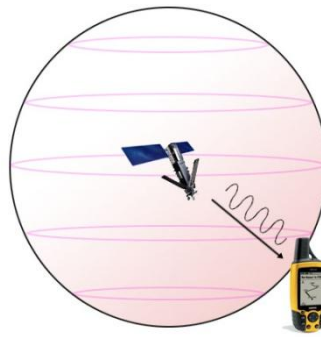
#### A.1.1 Funcionamiento del GPS

A continuación se describe el funcionamiento del GPS en 6 pasos (1):

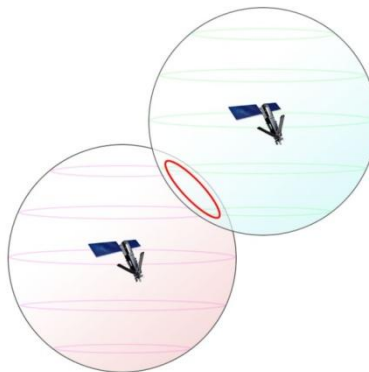
- a. El fundamento básico de la técnica de navegación empleada en el GPS es la medida simultánea de las distancias a varios satélites cuya posición se estima a partir de las efemérides (5 parámetros orbitales Keplerianos<sup>1</sup>), parámetros que son transmitidos por los propios satélites. La colección de efemérides de toda la constelación se completa cada 12 minutos y se guarda en el receptor GPS.
- b. El receptor GPS funciona midiendo su distancia a los satélites, ello permite establecer varias ecuaciones, al menos cuatro, cuya solución permite obtener la latitud, longitud y altitud del receptor, así como la sincronización de su reloj; información que es utilizada para calcular su posición. Esta distancia se mide calculando el tiempo que la señal tarda en llegar al receptor. Conocido ese tiempo y basándose en el hecho de que la señal viaja a la velocidad de la luz (salvo algunas correcciones que se aplican), se calcula la distancia entre el receptor y el satélite.
- c. Cada satélite indica que el receptor se encuentra en un punto en la superficie de la esfera, con centro en el propio satélite y de radio la distancia total hasta el receptor, tal como se muestra en la figura A.1.
- d. Al obtener información de dos satélites el receptor se encuentra en la intersección de dos esferas, tal como se muestra en la figura A.2.

---

<sup>1</sup> Parámetros keplerianos: parámetros utilizados para definir las formas de las órbitas satelitales, situar satélites en sus respectivas órbitas e indicar la posición de la órbita con relación a la tierra.

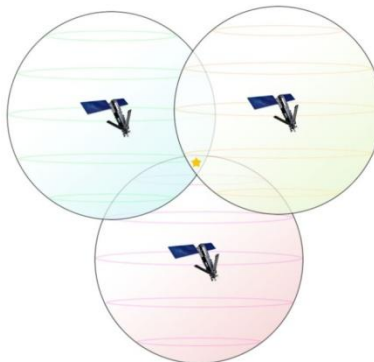


**Figura A.1.** Representación de la localización de un satélite.



**Figura A.2.** Intersección de dos esferas.

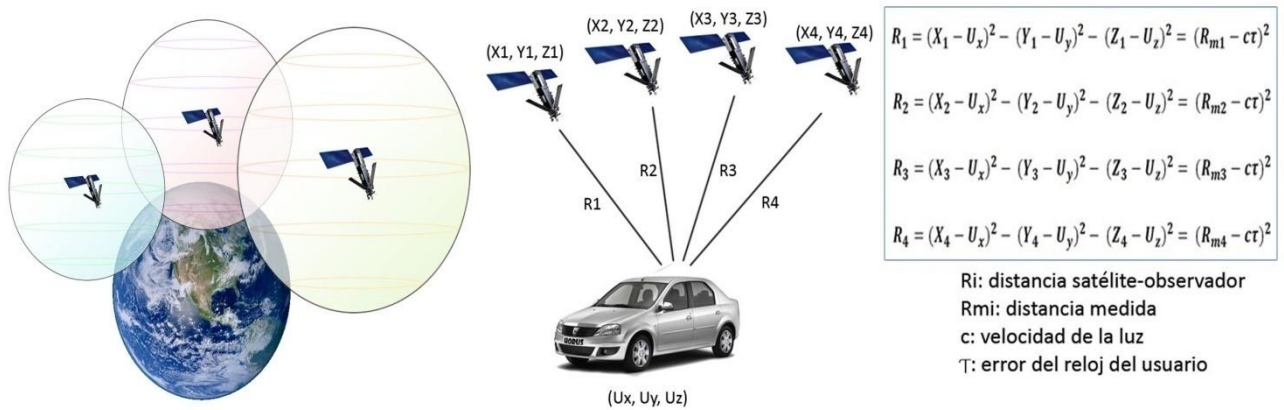
- e. Si se adquiere la misma información de un tercer satélite la nueva esfera sólo corta la circunferencia anterior en dos puntos. Uno de ellos se puede descartar porque ofrece una posición absurda. De esta manera se obtiene la posición en tercera dimensión (3D). Sin embargo, dado que el reloj incorporado en los receptores GPS no está sincronizado con los relojes atómicos de los satélites GPS, los dos puntos determinados no son precisos. La figura A.3 muestra la forma como se obtiene la posición en 3D.



**Figura A.3.** Posición en 3D a partir de la localización de 3 satélites.



- f. Teniendo información de un cuarto satélite, se elimina el inconveniente de la falta de sincronización entre el reloj del receptor GPS y los relojes de los satélites. Y es en este momento cuando el receptor GPS determina una posición 3-D exacta (latitud, longitud y altitud). La figura A.4. resume el principio de funcionamiento de los sistemas de navegación por satélite.



**Figura A.4.** Principio de funcionamiento del GPS.

Para el funcionamiento del sistema se usan dos tipos de códigos, los denominados C/A, utilizados por usuarios civiles, que emplean una frecuencia de transmisión de 1.023 MHz, y los denominados códigos P o de precisión, sólo permitidos para uso militar, que trabajan a 10.23 MHz. La precisión instrumental (sin considerar otras fuentes de error) depende de la frecuencia del código. Se estima que la precisión instrumental obtenida en la medida de las distancias es de varias decenas de metros con códigos C/A y de algunos metros con códigos P.

Las ventajas de este método son su exactitud y su uso masivo, además de no depender de ningún operador concreto. Sus limitaciones están en entornos cerrados y zonas urbanas donde el posicionamiento puede ser inexacto o incluso inviable (2). Los principales problemas que se pueden encontrar son el efecto multitrayecto<sup>2</sup>, consumo considerable de tiempo para recibir las tramas que conforman la señal emitida por los satélites y la falta de fiabilidad<sup>3</sup> de la señal.

<sup>2</sup> Fenómeno consistente en la propagación de una onda por varios caminos diferentes. Ello se debe a los fenómenos de reflexión y de difracción.

<sup>3</sup> La fiabilidad de un sistema es la probabilidad de que éste funcione o realice una cierta función, bajo condiciones establecidas y durante un período de tiempo determinado.

### A.1.2 Elementos comunes del software CAD

Podría considerarse como software CAD, desde un programa que permita llevar un registro de incidentes o eventos a cargo de una agencia, hasta un complejo conjunto de programas y conexiones externas que permitan apoyar plenamente todo lo que la agencia hace para atender los incidentes y dar respuesta a los eventos. La complejidad del software CAD estaría determinada por el tamaño de la agencia, el número de despachadores y el personal de trabajo necesario para brindar apoyo.

El software implementado en los sistemas CAD, tienen elementos o componentes esenciales que dan soporte a la labor del despachador (3).

- **Información sobre incidentes:** Consiste en una base de datos que permite acceder a la información relacionada con un incidente (ubicación, número de casa, nombre de la calle, tipo de incidente y otros datos recolectados por el despachador u operador (a)). Cuando el incidente se registra en la base de datos, el software CAD generalmente compara la ubicación de las entradas anteriores, y muestra una alerta si el nuevo incidente está relacionado con un registro de incidentes existente.
- **Interfaz de emergencia:** El computador donde está instalado el software CAD está conectado electrónicamente a las instituciones capaces de dar respuesta inmediata a una emergencia (Policía, Bomberos, Cruz Roja, Defensa Civil, Sistema Nacional de Prevención y Atención de Desastres, etc.), de modo que la información relacionada con la emergencia se muestra en pantalla y puede ser transferida a la institución más adecuada según el tipo de emergencia.
- **Verificación de la ubicación:** Una vez que el incidente es reportado y la ubicación es ingresada al software CAD, esta es comparada con una base de datos de localización estándar (archivo geográfico), que incluye dentro de sus parámetros de búsqueda los números de las casas, los nombres de las calles y los nombres de los barrios con el fin de facilitar el proceso de búsqueda y la verificación del lugar del incidente sin que se presenten inconsistencias.
- **Despliegue de Incidentes:** Una vez introducido un incidente, el software CAD muestra una lista de los incidentes pendientes, actuales y pasados, de acuerdo con la asignación del despachador. Los incidentes son por lo general, ordenados por fecha, hora y la prioridad, pero también por la ubicación, tipo de incidente u otros criterios, para que la atención sea más rápida y eficiente.
- **Información de las unidades:** Los sistemas CAD tiene una base de datos del personal y de las unidades de campo, que se utiliza para visualizar la lista de las unidades activas y las disponibles. Esta base de datos incluye el identificador de la



unidad asignada al personal (puede ser un código compuesto de números, caracteres o mixto) y sus capacidades especiales (antiexplosivos, paramédicos, transporte, etc.). La base de datos está vinculada a la base de datos de incidentes, lo que permite al operador consultar el estado actual de la unidad (activo o inactivo). De esta manera mediante el uso de los comandos o botones de la pantalla, el despachador puede cambiar el estado de una unidad y asignarle una atención a determinado incidente.

- **Asignación de atenciones:** Con la información de los incidentes y las unidades, el despachador puede vincular un incidente a una o más unidades. La mayoría del software implementado en sistemas CAD proporcionan una recomendación a las unidades encargadas de la atención, según el riesgo asociado al tipo de incidente (alto, medio o bajo) y el tipo de unidad encargada de atenderlo.
- **Marcas de tiempo:** Siempre que el despachador asigna una atención, se debe registrar el tiempo utilizado para darle respuesta y solución al incidente, con el fin de compararlo con los registros de la unidad y calcular la eficiencia del sistema.
- **Generación de reportes:** Para ayudar a analizar el evento o incidente y el desempeño de las unidades, el software CAD permite la elaboración automática de los reportes, listando y organizando toda la información relacionada con el incidente por rangos de fecha y hora.
- **Enlaces externos:** El computador y el software CAD pueden estar vinculados con otros sistemas informáticos, incluido el sistema de emergencias 911 en el caso de Estado Unidos, bases de datos locales, departamentales o nacionales con información de carácter legal (órdenes de registro de vehículos robados, licencias de conducir, tarjetas de propiedad, antecedentes legales, etc.), sistemas de radio, terminales móviles de datos, localizadores, etc.
- **Implementación de mapas:** Muchos centros de comunicación modernos han instalado sistemas de cartografía informática que se pueden interconectar con archivos o bases de datos geográficas de una zona determinada, con el fin de brindar la posibilidad de visualizar en mapas las unidades y los lugares del incidente.
- **Mantenimiento:** El software le permite al administrador del sistema crear y editar los diferentes archivos y reportes que le dan soporte al sistema; además permite generar archivos de respaldo (*back-up*) para salvar dicha información.
- **Seguridad:** El software CAD debe proporcionar un método flexible de asignación de niveles de seguridad a todos los diversos tipos de información que maneja, con el fin de permitir que el administrador del sistema asigne a los usuarios permisos de acceso

para consultar la información del sistema y restringir las funciones dentro del mismo.

La figura A.5 muestra cómo interactúan las características y capacidades de un sistema CAD. Los módulos más importantes del software se muestran en color verde, los sub-módulos en color amarillo y las bases de datos de cada modulo en color rojo. Los sistemas y bases de datos externas se muestran en color azul.

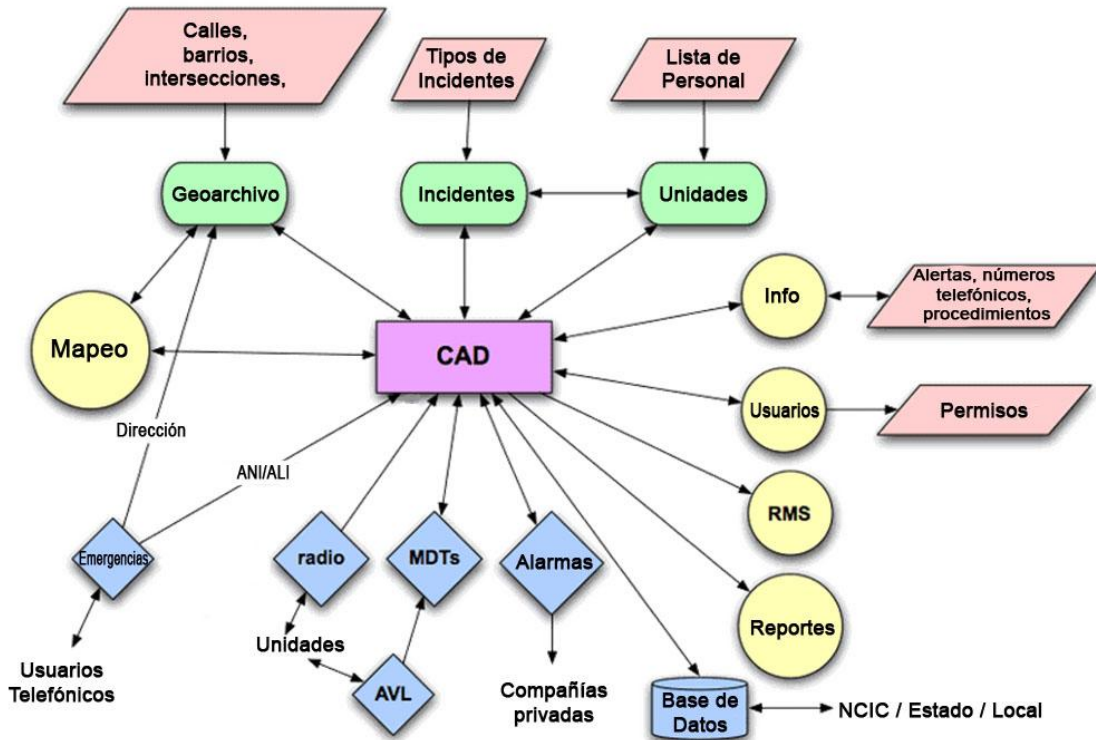


Figura A.5. Interacción de las capacidades y características de un sistema CAD.

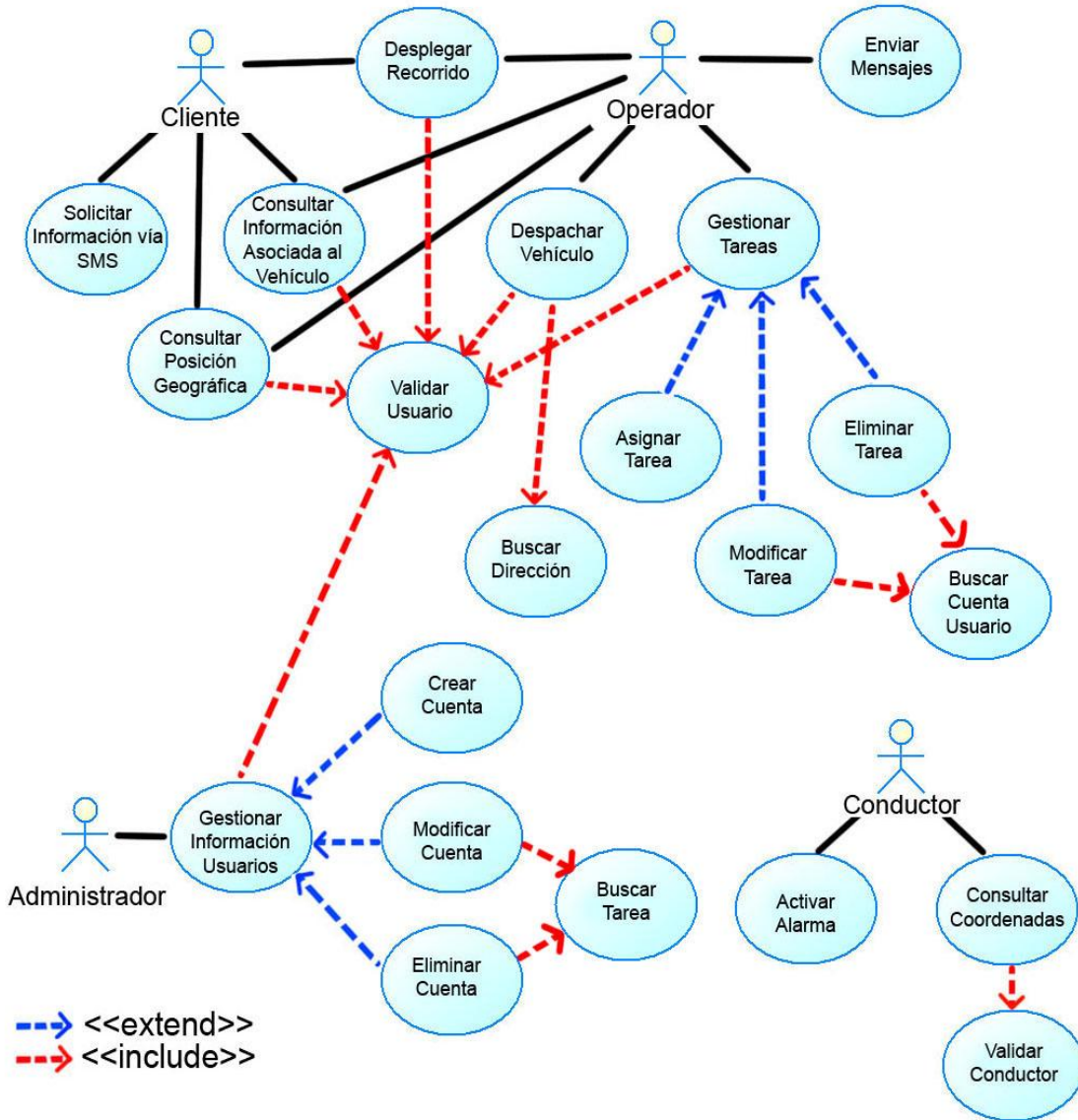
## A.2 ASPECTOS COMPLEMENTARIOS DEL CAPÍTULO 2 - DESCRIPCIÓN DETALLADA DE HORUS

En esta sección se explican detalladamente aspectos relacionados con el diseño del sistema prototipo de control y monitoreo vehicular asistido por GPS - HORUS.

### A.2.1 Descripción Detallada de los Casos de Uso

En la figura 2.3 del documento principal se muestran los casos de uso generales del sistema, donde se menciona únicamente aquellos que son de mayor prioridad para el funcionamiento del mismo; sin embargo, la ejecución de algunos de estos casos de

uso depende de otros catalogados en un nivel prioridad menor, haciendo necesario mencionar todos los casos de uso que se definieron e implementaron dentro del sistema para el despliegue de los servicios ofrecidos por HORUS. La figura A.6 muestra el diagrama detallado de casos de uso para el sistema.



**Figura A.6.** Diagrama detallado de casos de uso para HORUS.

Según las convenciones utilizadas en la figura A.6, las líneas continuas de color negro indican los casos de uso iniciados por los actores del sistema, las flechas punteadas de color azul representan casos de uso que son derivaciones o extensiones de un caso de uso general y las flechas punteadas de color rojo representan casos de uso que son prerequisites de un caso de uso general.

En las tablas A.1 (a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s) se muestra la descripción detallada para cada caso de uso mencionado en la figura A.1.

<b>VALIDAR USUARIO</b>
<b>Iniciador</b>
Cliente, Operador o Administrador
<b>Precondiciones</b>
El usuario debe tener registrada una cuenta con la información requerida por el sistema.
<b>Flujo de eventos</b>
<ul style="list-style-type: none"> <li>• <b>Flujo principal:</b> <ol style="list-style-type: none"> <li>1. El usuario selecciona el tipo de usuario e ingresa el nombre y la contraseña de su cuenta.</li> <li>2. El sistema compara los datos ingresados con la información almacenada en la base de datos y genera una respuesta.</li> <li>3. El sistema despliega en pantalla la interfaz correspondiente al tipo de usuario que se registro en el sistema.</li> </ol> </li>   <li>• <b>Flujos Alternativos:</b> <ol style="list-style-type: none"> <li>2. Si el proceso de validación no resulta exitoso, el sistema despliega en pantalla una alerta implementada en Java Script notificando el error en caso de que la validación del usuario se realice desde la aplicación web o desplegara una alarma en caso de que se esté realizando desde la aplicación móvil. Las posibles respuestas son: <ul style="list-style-type: none"> <li>○ <i>USUARIO</i>: Indica que los datos ingresados no corresponden al tipo de usuario seleccionado.</li> <li>○ <i>PASSWORD</i>: Indica que la contraseña de la cuenta no es correcta.</li> <li>○ <i>ERROR</i>: Indica que no existe una cuenta asociada a los datos ingresados.</li> </ul> </li> </ol> </li>   <li>• <b>Excepciones:</b> <ul style="list-style-type: none"> <li>○ Si la conexión entre el servidor y la base no es exitosa, se producirá una excepción cuando se pretendan realizar las consultas.</li> <li>○ Si la conexión entre el servidor central y el dispositivo móvil no es exitosa, se producirá una excepción cuando el stub implementado en la aplicación móvil intente utilizar los métodos del servicio web implementado en el servidor.</li> </ul> </li> </ul>
<b>Post-condiciones</b>
Si ocurre una falla en el servidor central que interrumpa el flujo normal de las aplicaciones, se debe validar nuevamente el ingreso de los usuarios al sistema.
<b>Recursos Especiales Implementados</b>
<ul style="list-style-type: none"> <li>• Base de Datos: <ul style="list-style-type: none"> <li>○ Tabla Usuarios.</li> </ul> </li> </ul>
<b>Interfaces de Usuario Relacionadas</b>
<ul style="list-style-type: none"> <li>• inicio.jsp</li> <li>• InicioForm.java</li> </ul>

**Tabla A.1 (a).** Descripción detallada para el caso de uso Validar Usuario.

<b>CONSULTAR POSICIÓN GEOGRÁFICA</b>	
<b>Iniciador</b>	Cliente, Operador o Administrador
<b>Precondiciones</b>	Los datos de la cuenta de los usuarios deben haber sido validados para permitirles el ingreso al sistema.
<b>Flujo de eventos</b>	<ul style="list-style-type: none"><li>• <b>Flujo principal:</b><ol style="list-style-type: none"><li>1. El usuario selecciona la opción de consultar la posición geográfica del vehículo.</li><li>2. El sistema obtiene de la base de datos las coordenadas angulares de posicionamiento enviadas por el dispositivo GPS de la unidad móvil y foto del vehículo.</li><li>3. El sistema transforma las coordenadas obtenidas de la bases de datos en números decimales y las envía hasta el servidor de mapas de Google.</li><li>4. El servidor de mapas retorna una imagen donde se señala con un icono la posición del vehículo.</li><li>5. El sistema despliega en la interfaz web, la imagen obtenida del servidor de mapas y permite consultar sobre la imagen, los datos de posicionamiento y la foto del vehículo a través de ventanas emergentes.</li></ol></li><li>• <b>Flujos Alternativos:</b><ol style="list-style-type: none"><li>1. Los usuarios tipo administrador y operador pueden consultar la posición geográfica de cualquier vehículo, para lo cual el sistema les permite ingresar el ID del vehículo a localizar.</li><li>2. En caso de que el ID de un vehículo ingresado por un administrador o por un operador no se encuentre registrado en la base de datos, el sistema notificara al usuario a través de una alerta implementada en Java Script para el caso de la aplicación web y a través de una alarma para el caso de la aplicación móvil.</li><li>3. Para el caso de la aplicación móvil, se despliega una alarma notificando que no hay conexión con el servidor de mapas de <i>Google</i>, en caso de que se presente esta situación y no se pueda obtener la imagen.</li></ol></li><li>• <b>Excepciones:</b><ul style="list-style-type: none"><li>○ Si la conexión entre el servidor y la base no es exitosa, se producirá una excepción cuando se pretendan realizar las consultas.</li><li>○ Si la conexión entre el servidor central y el dispositivo móvil no es exitosa, se producirá una excepción cuando el stub implementado en la aplicación móvil intente utilizar los métodos del servicio web implementado en el servidor.</li><li>○ Si la conexión entre el dispositivo móvil y el servidor de mapas de Google no es exitosa, se producirá una excepción cuando se envíen las peticiones para obtener las imágenes de los mapas.</li></ul></li></ul>
<b>Post-condiciones</b>	Una vez realizadas las consultas y el usuario cierre la sesión de su cuenta, el sistema debe garantizar un nivel de seguridad mínimo y validar de nuevo el ingreso al sistema, en caso de que así lo requieran.
<b>Recursos Especiales Implementados</b>	<ul style="list-style-type: none"><li>• Unidad Móvil.</li><li>• Unidad de Control, Conmutación y Acondicionamiento EC.</li><li>• Radios Convencionales de dos Vías.</li><li>• Google Maps API.</li><li>• MIDMaps API.</li><li>• Base de Datos:</li></ul>

<ul style="list-style-type: none"> <li>○ Tabla Vehículos.</li> <li>○ Tabla Recorridos</li> <li>○ Tabla Usuarios.</li> </ul>
<b>Interfaces de Usuario Relacionadas</b>
<ul style="list-style-type: none"> <li>• googlemaps.jsp</li> <li>• barralocalizar.jsp</li> <li>• barraoperador.jsp</li> <li>• GoogleMapsCanvas.java</li> </ul>

**Tabla A.1 (b).** Descripción detallada para el caso de uso Consultar Posición Geográfica.

DESPLEGAR RECORRIDO
<b>Iniciador</b>
Cliente, Operador o Administrador
<b>Precondiciones</b>
Los datos de la cuenta de los usuarios deben haber sido validados para permitirles el ingreso al sistema.
<b>Flujo de eventos</b>
<ul style="list-style-type: none"> <li>• <b>Flujo principal:</b> <ol style="list-style-type: none"> <li>1. El usuario selecciona la opción de desplegar recorrido.</li> <li>2. El sistema obtiene de la base de datos las diez últimas coordenadas angulares de posicionamiento enviadas por el dispositivo GPS de la unidad móvil y foto del vehículo.</li> <li>3. El sistema calcula la distancia recorrida durante el intervalo de tiempo donde se obtuvieron las coordenadas.</li> <li>4. El sistema transforma las coordenadas obtenidas de la bases de datos en números decimales y las envía hasta el servidor de mapas de Google.</li> <li>5. El servidor de mapas retorna una imagen donde se muestra con una línea, el recorrido realizado por el vehículo, señalando con un icono el comienzo y el final.</li> <li>6. El sistema despliega en la interfaz web, la imagen obtenida del servidor de mapas y permite consultar a través de ventanas emergentes, la velocidad promedio del recorrido, la distancia recorrida, el tiempo empleado en realizar el recorrido y la foto del vehículo.</li> </ol> </li> <li>• <b>Flujos Alternativos:</b> <ol style="list-style-type: none"> <li>1. Los usuarios tipo administrador y operador pueden consultar el recorrido de cualquier vehículo, para lo cual el sistema les permite ingresar el ID del vehículo a localizar.</li> <li>2. En caso de que el ID de un vehículo ingresado por un administrador o por un operador no se encuentre registrado en la base de datos, el sistema notificara al usuario a través de una alerta implementada en Java Script para el caso de la aplicación web y a través de una alarma para el caso de la aplicación móvil.</li> <li>4. Para el caso de la aplicación móvil, se despliega una alarma notificando que no hay conexión con el servidor de mapas de <i>Google</i>, en caso de que se presente esta situación y no se pueda obtener la imagen.</li> </ol> </li> <li>• <b>Excepciones:</b> <ul style="list-style-type: none"> <li>○ Si la conexión entre el servidor y la base no es exitosa, se producirá una excepción cuando se pretendan realizar las consultas.</li> <li>○ Si la conexión entre el servidor central y el dispositivo móvil no es exitosa, se producirá una excepción cuando el <i>stub</i> implementado en la aplicación móvil intente utilizar los métodos del servicio web implementado en el servidor.</li> <li>○ Si la conexión entre el dispositivo móvil y el servidor de mapas de Google no es</li> </ul> </li> </ul>



exitosa, se producirá una excepción cuando se envíen las peticiones para obtener las imágenes de los mapas.
<b>Post-condiciones</b>
Una vez realizadas las consultas y el usuario cierre la sesión de su cuenta, el sistema debe garantizar un nivel de seguridad mínimo y validar de nuevo el ingreso al sistema, en caso de que así lo requieran.
<b>Recursos Especiales Implementados</b>
<ul style="list-style-type: none"> <li>• Unidad Móvil.</li> <li>• Unidad de Control, Conmutación y Acondicionamiento EC.</li> <li>• Radios Convencionales de dos Vías.</li> <li>• Google Maps API.</li> <li>• MIDMaps API.</li> <li>• Base de Datos: <ul style="list-style-type: none"> <li>○ Tabla Vehículos.</li> <li>○ Tabla Recorridos</li> <li>○ Tabla Usuarios.</li> </ul> </li> </ul>
<b>Interfaces de Usuario Relacionadas</b>
<ul style="list-style-type: none"> <li>• googlemaps.jsp</li> <li>• barralocalizar.jsp</li> <li>• barraoperador.jsp</li> <li>• GoogleMapsPathCanvas.java</li> </ul>

**Tabla A.1 (c).** Descripción detallada para el caso de uso Desplegar Recorrido.

<b>BUSCAR DIRECCIÓN</b>
<b>Iniciador</b>
Operador
<b>Precondiciones</b>
El operador previamente debe haber obtenido el permiso para ingresar al sistema una vez se hayan validado sus datos.
<b>Flujo de eventos</b>
<ul style="list-style-type: none"> <li>• <b>Flujo principal:</b> <ol style="list-style-type: none"> <li>1. El operador ingresa en el sistema la dirección o una palabra clave asociada al lugar que se desea ubicar en el mapa y selecciona la opción "Buscar Dirección".</li> <li>2. El sistema verifica la existencia de la dirección o de la palabra clave en la base de datos.</li> <li>3. El sistema obtiene las coordenadas decimales de la base de datos y las envía al servidor.</li> <li>4. El servidor de mapas retorna una imagen donde se señala con un icono el lugar a localizar.</li> <li>5. El sistema despliega en la interfaz web, la imagen obtenida del servidor de mapas, la posición del vehículo, una breve descripción y una imagen asociada a e lugar.</li> </ol> </li> <li>• <b>Flujos Alternativos:</b> <ol style="list-style-type: none"> <li>2. En caso de que la dirección o que la palabra calve ingresada por un operador, no se encuentre registrada en la base de datos, el sistema notificara al usuario a través de una alerta implementada en Java Script.</li> </ol> </li> </ul>

<ul style="list-style-type: none"> <li>• <b>Excepciones:</b> <ul style="list-style-type: none"> <li>○ Si la conexión entre el servidor y la base no es exitosa, se producirá una excepción cuando se pretendan realizar las consultas.</li> <li>○ Si la conexión entre el servidor de mapas de Google y el servidor central no es exitosa, se producirá una excepción cuando se envíen las peticiones para obtener las imágenes de los mapas.</li> </ul> </li> </ul>
<b>Post-condiciones</b>
Una vez realizadas las consultas y el usuario cierre la sesión de su cuenta, el sistema debe garantizar un nivel de seguridad mínimo y validar de nuevo el ingreso al sistema, en caso de que así lo requieran.
<b>Recursos Especiales Implementados</b>
<ul style="list-style-type: none"> <li>• Google Maps API.</li> <li>• Base de Datos: <ul style="list-style-type: none"> <li>○ Tabla Direcciones.</li> </ul> </li> </ul>
<b>Interfaces de Usuario Relacionadas</b>
<ul style="list-style-type: none"> <li>• googlemaps.jsp</li> <li>• barraoperador.jsp</li> </ul>

**Tabla A.1 (d).** Descripción detallada para el caso de uso Buscar Dirección.

DESPACHAR VEHICULO
<b>Iniciador</b>
Operador
<b>Precondiciones</b>
El operador previamente debe haber obtenido el permiso para ingresar al sistema una vez se hayan validado sus datos.
<b>Flujo de eventos</b>
<ul style="list-style-type: none"> <li>• <b>Flujo principal:</b> <ol style="list-style-type: none"> <li>3. El operador ingresa en el sistema la dirección de lugar donde proviene la solicitud de despacho.</li> <li>4. El operador notifica vía radio la solicitud a los vehículos e ingresa en el sistema el ID de los cinco primeros vehículos que se reporten como disponibles.</li> <li>5. El operador selecciona la opción despachar vehículo.</li> <li>6. El sistema obtiene de la base de datos las coordenadas angulares de los vehículos y del lugar donde proviene la solicitud de despacho para calcular la distancia existente entre cada uno de los vehículos reportados como disponibles y este lugar.</li> <li>7. El sistema envía al servidor de mapas las coordenadas.</li> <li>8. El servidor de mapas retorna una imagen donde se señala con un icono la posición de los vehículos y del lugar, trazando una línea entre este último y el icono del vehículo q se encuentra a la menor distancia.</li> <li>9. El sistema despliega en la interfaz web, la imagen obtenida del servidor de mapas, el ID del vehículo que se debe despachar para atender la solicitud y la distancia existente entre este vehículo y el lugar de donde proviene la solicitud.</li> </ol> </li> <li>• <b>Flujos Alternativos:</b> <ol style="list-style-type: none"> <li>10. En caso de que la dirección o que el ID de un vehículo ingresado por un operador no se encuentre registrado en la base de datos, el sistema notificara al usuario a través de una alerta implementada en Java Script.</li> </ol> </li> </ul>

<ul style="list-style-type: none"> <li>• <b>Excepciones:</b> <ul style="list-style-type: none"> <li>○ Si la conexión entre el servidor y la base no es exitosa, se producirá una excepción cuando se pretendan realizar las consultas.</li> <li>○ Si la conexión entre el servidor de mapas de Google y el servidor central no es exitosa, se producirá una excepción cuando se envíen las peticiones para obtener las imágenes de los mapas.</li> </ul> </li> </ul>
<b>Post-condiciones</b>
Una vez realizadas las consultas y el usuario cierre la sesión de su cuenta, el sistema debe garantizar un nivel de seguridad mínimo y validar de nuevo el ingreso al sistema, en caso de que así lo requieran.
<b>Recursos Especiales Implementados</b>
<ul style="list-style-type: none"> <li>• Unidad Móvil.</li> <li>• Unidad de Control, Conmutación y Acondicionamiento EC.</li> <li>• Radios Convencionales de dos Vías.</li> <li>• Google Maps API.</li> <li>• Base de Datos: <ul style="list-style-type: none"> <li>○ Tabla Vehículos.</li> <li>○ Tabla Recorridos</li> <li>○ Tabla Direcciones.</li> </ul> </li> </ul>
<b>Interfaces de Usuario Relacionadas</b>
<ul style="list-style-type: none"> <li>• googlemaps.jsp</li> <li>• barraoperador.jsp</li> </ul>

**Tabla A.1 (e).** Descripción detallada para el caso de uso Despachar Vehículo.

CONSULTAR INFORMACION ASOCIADA AL VEHICULO
<b>Iniciador</b>
Cliente, Operador o Administrador
<b>Precondiciones</b>
El operador previamente debe haber obtenido el permiso para ingresar al sistema una vez se hayan validado sus datos.
<b>Flujo de eventos</b>
<ul style="list-style-type: none"> <li>• <b>Flujo principal:</b> <ol style="list-style-type: none"> <li>1. El usuario selecciona la opción de consultar información.</li> <li>2. El sistema obtiene de la base de datos la información asociada al vehículo, su propietario y su conductor en el caso de la aplicación web y únicamente del vehículo y su conductor en el caso de la aplicación móvil.</li> <li>3. El sistema despliega una tabla en pantalla con la información obtenida de la base de datos y las respectivas imágenes para el caso de la aplicación web. En el caso de la aplicación móvil, solo se muestra en un texto la información asociada al vehículo y al conductor.</li> </ol> </li> <li>• <b>Flujos Alternativos:</b> <ol style="list-style-type: none"> <li>1. Los usuarios tipo administrador y operador pueden consultar la información asociada a cualquier vehículo, para lo cual el sistema les permite ingresar el ID del vehículo.</li> <li>2. En caso de el ID de un vehículo ingresado por un usuario no se encuentre registrado</li> </ol> </li> </ul>

<p>en la base de datos, el sistema notificara al usuario a través de una alerta implementada en Java Script en el caso de la aplicación web.</p> <ul style="list-style-type: none"> <li>• <b>Excepciones:</b> <ul style="list-style-type: none"> <li>○ Si la conexión entre el servidor y la base no es exitosa, se producirá una excepción cuando se pretendan realizar las consultas.</li> <li>○ Si la conexión entre el servidor central y el dispositivo móvil no es exitosa, se producirá una excepción cuando el <i>stub</i> implementado en la aplicación móvil intente utilizar los métodos del servicio web implementado en el servidor.</li> </ul> </li> </ul>
<b>Post-condiciones</b>
Una vez realizadas las consultas y el usuario cierre la sesión de su cuenta, el sistema debe garantizar un nivel de seguridad mínimo y validar de nuevo el ingreso al sistema, en caso de que así lo requieran.
<b>Recursos Especiales Implementados</b>
<ul style="list-style-type: none"> <li>• Base de Datos: <ul style="list-style-type: none"> <li>○ Tabla Vehículos.</li> <li>○ Tabla Usuarios</li> <li>○ Tabla Conductores</li> </ul> </li> </ul>
<b>Interfaces de Usuario Relacionadas</b>
<ul style="list-style-type: none"> <li>• información.jsp</li> <li>• barraoperador.jsp</li> <li>• InfoForm.java</li> </ul>

**Tabla A.1 (f).** Descripción detallada para el caso de uso Consultar Información Asociada al Vehículo.

ENVIAR MENSAJES
<b>Iniciador</b>
Operador
<b>Precondiciones</b>
El operador previamente debe haber obtenido el permiso para ingresar al sistema una vez se hayan validado sus datos.
<b>Flujo de eventos</b>
<ul style="list-style-type: none"> <li>• <b>Flujo principal:</b> <ol style="list-style-type: none"> <li>1. El operador ingresa el ID del vehículo al que va dirigido el mensaje, escribe un mensaje con un tamaño máximo de 32 caracteres y pulsa el botón “Enviar Mensaje”.</li> <li>2. El sistema verifica la existencia del vehículo en la base de datos.</li> <li>3. El sistema envía el mensaje a la unidad móvil a través de la unidad de control conmutación y acondicionamiento, y de los radios convencionales de dos vías; utilizando el protocolo definido para transmisión de datos.</li> <li>4. La unidad móvil informa la llegada del mensaje activando la alarma sonora y desplegándolo en el LCD.</li> </ol> </li> <li>• <b>Flujos Alternativos:</b> <ol style="list-style-type: none"> <li>1. Si el mensaje contiene alguno de los caracteres utilizados para diferenciar los campos de información según el protocolo diseñado (“:”, “&gt;”, “+”), el sistema despliega en pantalla un mensaje indicando que no se pueden incluir en el cuerpo del mensaje.</li> <li>3. En caso de que el servidor central no reciba confirmación de que el mensaje enviado</li> </ol> </li> </ul>

<p>fue recibido exitosamente, lo retransmitirá hasta cuatro veces más. Si después de realizar esos intentos todavía no se recibe confirmación, el vehículo será considerado como fuera del área de cobertura</p> <ul style="list-style-type: none"> <li>• <b>Excepciones:</b> <ul style="list-style-type: none"> <li>○ Si la conexión entre el servidor y la base no es exitosa, se producirá una excepción cuando se pretendan realizar las consultas.</li> <li>○ El mensaje enviado desde la aplicación web puede no llegar al destinatario si el vehículo está en un área por fuera de la cobertura del sistema de comunicaciones implementado.</li> </ul> </li> </ul>
<b>Post-condiciones</b>
Una vez realizadas las operaciones y el operador cierre la sesión de su cuenta, el sistema debe garantizar un nivel de seguridad mínimo y validar de nuevo el ingreso al sistema, en caso de que así lo requieran.
<b>Recursos Especiales Implementados</b>
<ul style="list-style-type: none"> <li>• Unidad Móvil.</li> <li>• Unidad de Control, Conmutación y Acondicionamiento EC.</li> <li>• Radios Convencionales de dos Vías.</li> <li>• Base de Datos: <ul style="list-style-type: none"> <li>○ Tabla Vehículos.</li> <li>○ Tabla Usuarios</li> <li>○ Tabla Conductores</li> </ul> </li> </ul>
<b>Interfaces de Usuario Relacionadas</b>
<ul style="list-style-type: none"> <li>• Mensajesop.jsp</li> </ul>

**Tabla A.1 (g).** Descripción detallada para el caso de uso Enviar Mensajes.

SOLICITAR INFORMACION VIA SMS
<b>Iniciador</b>
Cliente
<b>Precondiciones</b>
El cliente debe tener un dispositivo móvil con capacidad para enviar y recibir mensajes de texto cortos (SMS).
<b>Flujo de eventos</b>
<ul style="list-style-type: none"> <li>• <b>Flujo principal:</b> <ol style="list-style-type: none"> <li>1. Para solicitar información de algún vehículo vía SMS, el cliente envía un mensaje de texto a la estación central con el ID del vehículo.</li> <li>2. El sistema recibe el SMS y obtiene el ID del vehículo para verifica su existencia en la base de datos.</li> <li>3. El sistema envía como respuesta al usuario un mensaje de texto con la información básica del vehículo y su conductor.</li> </ol> </li> <li>• <b>Flujos Alternativos:</b> <ol style="list-style-type: none"> <li>2. En caso de que el vehículo no se encuentre registrado en la base de datos, el sistema envía como respuesta un mensaje de texto indicándolo.</li> </ol> </li> <li>• <b>Excepciones:</b> <ul style="list-style-type: none"> <li>○ Si la conexión entre el servidor y la base no es exitosa, se producirá una excepción</li> </ul> </li> </ul>

<ul style="list-style-type: none"> <li>o cuando se pretendan realizar las consultas.</li> <li>o El éxito del envío y la recepción del mensaje depende de la red de telefonía móvil.</li> </ul>
<b>Post-condiciones</b>
Ninguna
<b>Recursos Especiales Implementados</b>
<ul style="list-style-type: none"> <li>• JSMS</li> <li>• SMSC</li> <li>• Dispositivo Móvil.</li> <li>• Base de Datos: <ul style="list-style-type: none"> <li>o Tabla Vehículos.</li> <li>o Tabla Conductores.</li> <li>o Tabla Usuarios.</li> </ul> </li> </ul>
<b>Interfaces de Usuario Relacionadas</b>
<ul style="list-style-type: none"> <li>• Interfaz Servidor Central – Red de Telefonía Móvil.</li> </ul>

**Tabla A.1 (h).** Descripción detallada para el caso de uso Solicitar Información vía SMS.

<b>VALIDAR CONDUCTOR</b>
<b>Iniciador</b>
Conductor
<b>Precondiciones</b>
<ul style="list-style-type: none"> <li>• La comunicación entre la unidad móvil y el servidor central se debe haber establecido con anterioridad.</li> <li>• El vehículo debe estar registrado en el sistema.</li> </ul>
<b>Flujo de eventos</b>
<ul style="list-style-type: none"> <li>• <b>Flujo principal:</b> <ol style="list-style-type: none"> <li>1. El conductor ingresa el nombre y la contraseña de su cuenta a través de la unidad móvil.</li> <li>2. El sistema envía los datos hasta el servidor central donde se compara con la información almacenada en la base de datos y se genera una respuesta.</li> <li>3. El sistema envía la respuesta al vehículo para que sea visualizada por el conductor a través de la unidad móvil.</li> <li>4. Se autoriza el ingreso del conductor al sistema.</li> </ol> </li> <li>• <b>Flujos Alternativos:</b> <ol style="list-style-type: none"> <li>2. Si el proceso de validación no resulta exitoso, el sistema despliega en la pantalla de la unidad móvil un mensaje indicándolo.</li> </ol> </li> <li>• <b>Excepciones:</b> <ul style="list-style-type: none"> <li>o Si la conexión entre el servidor y la base no es exitosa, se producirá una excepción cuando se pretendan realizar las consultas.</li> <li>o Los datos enviados desde o hacia el vehículo, podrían perderse en caso de que se encuentre por fuera del área de cobertura del sistema de comunicaciones implementado.</li> </ul> </li> </ul>

<b>Post-condiciones:</b>
Si el vehículo sale del área de cobertura del sistema, el conductor deberá validarse nuevamente en el sistema.
<b>Recursos Especiales Implementados</b>
<ul style="list-style-type: none"> <li>• Unidad Móvil.</li> <li>• Radios Convencionales de dos vías.</li> <li>• Unidad de Control, Conmutación y Acondicionamiento EC.</li> <li>• Base de Datos:             <ul style="list-style-type: none"> <li>○ Tabla Vehículos.</li> <li>○ Tabla Conductores.</li> <li>○ Tabla Usuarios.</li> </ul> </li> </ul>
<b>Interfaces de Usuario Relacionadas</b>
<ul style="list-style-type: none"> <li>• LCD</li> <li>• Teclado</li> </ul>

**Tabla A.1 (i).** Descripción detallada para el caso de uso Validar Conductor.

CONSULTAR COORDENADAS
<b>Iniciador</b>
Conductor
<b>Precondiciones</b>
<ul style="list-style-type: none"> <li>• La comunicación entre la unidad móvil y el servidor central se debe haber establecido con anterioridad.</li> <li>• El vehículo debe estar registrado en el sistema.</li> <li>• El conductor debe estar validado en el sistema.</li> </ul>
<b>Flujo de eventos</b>
<ul style="list-style-type: none"> <li>• <b>Flujo principal:</b> <ol style="list-style-type: none"> <li>1. El conductor selecciona la opción de consultar coordenadas a través del teclado de la unidad móvil.</li> <li>2. La unidad móvil despliega en pantalla las coordenadas de posicionamiento obtenidas desde el dispositivo GPS.</li> </ol> </li> <li>• <b>Excepciones:</b> <ul style="list-style-type: none"> <li>○ Si la conexión entre el GPS y el microcontrolador de la unidad móvil no es exitosa, se producirá una excepción cuando se soliciten las coordenadas geográficas.</li> </ul> </li> </ul>
<b>Post-condiciones</b>
Ninguna
<b>Recursos Especiales Implementados</b>
<ul style="list-style-type: none"> <li>• Unidad Móvil</li> </ul>
<b>Interfaces de Usuario Relacionadas</b>

<ul style="list-style-type: none"> <li>• LCD</li> <li>• Teclado</li> </ul>
--

**Tabla A.1 (j).** Descripción detallada para el caso de uso Consultar Coordenadas.

<b>ACTIVAR ALARMA</b>
<b>Iniciador</b>
Conductor
<b>Precondiciones</b>
<ul style="list-style-type: none"> <li>• La comunicación entre la unidad móvil y el servidor central se debe haber establecido con anterioridad.</li> <li>• El vehículo debe estar registrado en el sistema.</li> </ul>
<b>Flujo de eventos</b>
<ul style="list-style-type: none"> <li>• <b>Flujo principal:</b> <ol style="list-style-type: none"> <li>1. El conductor presiona el botón de pánico de la unidad móvil.</li> <li>2. El sistema envía la información relacionada con la emergencia hasta el servidor central, donde es almacenada en la base de datos.</li> <li>3. El sistema obtiene las coordenadas de la emergencia desde la base de datos y las envía al servidor de mapas.</li> <li>4. El servidor de mapas devuelve una imagen, señalando con un icono las coordenadas donde se encuentra el vehículo que activo el botón de pánico.</li> <li>5. El sistema despliega en pantalla la imagen enviada por el servidor de mapas y permite visualizar mediante una ventana emergente la imagen del vehículo y las coordenadas de su posición.</li> </ol> </li> <li>• <b>Excepciones:</b> <ul style="list-style-type: none"> <li>○ Si la conexión entre el servidor y la base no es exitosa, se producirá una excepción cuando se pretendan realizar las consultas.</li> <li>○ Los datos enviados desde o hacia el vehículo, podrían perderse en caso de que se encuentre por fuera del área de cobertura del sistema de comunicaciones implementado.</li> <li>○ Si la conexión entre el servidor de mapas de Google y el servidor central no es exitosa, se producirá una excepción cuando se envíen las peticiones para obtener las imágenes de los mapas.</li> </ul> </li> </ul>
<b>Post-condiciones</b>
Si el vehículo sale del área de cobertura del sistema, el conductor deberá validarse nuevamente en el sistema.
<b>Recursos Especiales Implementados</b>
<ul style="list-style-type: none"> <li>• Unidad Móvil</li> </ul>
<b>Interfaces de Usuario Relacionadas</b>
<ul style="list-style-type: none"> <li>• LCD y Teclado – Unidad Móvil.</li> </ul>

**Tabla A.1 (k).** Descripción detallada para el caso de uso Activar Alarma.



BUSCAR CUENTA USUARIO	
<b>Iniciador</b>	Administrador
<b>Precondiciones</b>	El administrador previamente debe haber obtenido el permiso para ingresar al sistema una vez se hayan validado sus datos.
<b>Flujo de eventos</b>	<ul style="list-style-type: none"> <li>• <b>Flujo principal:</b> <ol style="list-style-type: none"> <li>1. El administrador selecciona el tipo de usuario a buscar e ingresa la cedula asociada a la cuenta del usuario.</li> <li>2. El sistema recorre la tabla usuarios de la base de datos y despliega en pantalla la información asociada al usuario.</li> </ol> </li> <li>• <b>Flujos Alternativos:</b> <ol style="list-style-type: none"> <li>2. Si el proceso de búsqueda no resulta exitoso, el sistema despliega en pantalla una alerta implementada en Java Script notificando el error. Las posibles respuestas son:                             <ul style="list-style-type: none"> <li>○ <i>ERROR</i>: Indica que no existe una cuenta asociada a la cedula ingresada.</li> <li>○ <i>TIPO_USUARIO</i>: Indica que la cedula ingresada no corresponde a ninguna cuenta según el tipo de usuario seleccionado.</li> </ul> </li> </ol> </li> <li>• <b>Excepciones:</b> <ul style="list-style-type: none"> <li>○ Si la conexión entre el servidor y la base no es exitosa, se producirá una excepción cuando se pretendan realizar las consultas.</li> </ul> </li> </ul>
<b>Post-condiciones:</b>	Si ocurre una falla en el servidor central que interrumpa el flujo normal de las aplicaciones, se debe validar nuevamente el ingreso de los usuarios al sistema.
<b>Recursos Especiales Implementados</b>	<ul style="list-style-type: none"> <li>• Base de Datos:                             <ul style="list-style-type: none"> <li>○ Tabla Usuarios.</li> </ul> </li> </ul>
<b>Interfaces de Usuario Relacionadas</b>	<ul style="list-style-type: none"> <li>• gestionusuarios.jsp</li> </ul>

**Tabla A.1 (I).** Descripción detallada para el caso de uso Buscar Cuenta Usuario.

CREAR CUENTA	
<b>Iniciador</b>	Administrador
<b>Precondiciones</b>	El administrador previamente debe haber obtenido el permiso para ingresar al sistema una vez se hayan validado sus datos.
<b>Flujo de eventos</b>	<ul style="list-style-type: none"> <li>• <b>Flujo principal:</b> <ol style="list-style-type: none"> <li>1. El administrador selecciona el tipo de usuario para crear la cuenta.</li> </ol> </li> </ul>

<ol style="list-style-type: none"> <li>2. El sistema despliega en pantalla los campos de texto para ingresar la información correspondiente a la cuenta.</li> <li>3. El administrador ingresa la información del usuario y presiona el botón "Agregar Usuario".</li> <li>4. El sistema crea un nuevo registro en la base de datos con la información del usuario.</li> </ol> <ul style="list-style-type: none"> <li>• <b>Flujos Alternativos:</b> <ol style="list-style-type: none"> <li>3. Si el proceso de creación de un nuevo registro no resulta exitoso, el sistema despliega en pantalla una alerta implementada en Java Script notificando el error. La respuestas enviada es: <ul style="list-style-type: none"> <li>○ <i>ERROR</i>: Indica que no fue posible crear un nuevo registro.</li> </ul> </li> </ol> </li> <li>• <b>Excepciones:</b> <ul style="list-style-type: none"> <li>○ Si la conexión entre el servidor y la base no es exitosa, se producirá una excepción cuando se pretendan realizar las consultas.</li> </ul> </li> </ul>
<b>Post-condiciones:</b>
Si ocurre una falla en el servidor central que interrumpa el flujo normal de las aplicaciones, se debe validar nuevamente el ingreso de los usuarios al sistema.
<b>Recursos Especiales Implementados</b>
<ul style="list-style-type: none"> <li>• Base de Datos: <ul style="list-style-type: none"> <li>○ Tabla Usuarios.</li> </ul> </li> </ul>
<b>Interfaces de Usuario Relacionadas</b>
<ul style="list-style-type: none"> <li>• gestionusuarios.jsp</li> </ul>

**Tabla A.1 (m).** Descripción detallada para el caso de uso Crear Cuenta.

<b>MODIFICAR CUENTA</b>
<b>Iniciador</b>
Administrador
<b>Precondiciones</b>
El administrador previamente debe haber obtenido el permiso para ingresar al sistema una vez se hayan validado sus datos.
<b>Flujo de eventos</b>
<ul style="list-style-type: none"> <li>• <b>Flujo principal:</b> <ol style="list-style-type: none"> <li>1. El administrador selecciona el tipo de usuario de la cuenta que se va a modificar, ingresa la cedula del usuario y presiona el botón "Buscar".</li> <li>2. El sistema despliega en pantalla los campos de texto con la información correspondiente a la cuenta del usuario.</li> <li>3. El administrador modifica los campos necesarios y presiona el botón "Modificar Usuario".</li> <li>4. El sistema actualiza la información en el registro correspondiente a la cuenta del usuario.</li> </ol> </li> <li>• <b>Flujos Alternativos:</b> <ol style="list-style-type: none"> <li>4. Si el proceso de modificación de un registro no resulta exitoso, el sistema despliega en pantalla una alerta implementada en Java Script notificando el error. La respuesta enviada es: <ul style="list-style-type: none"> <li>○ <i>ERROR</i>: Indica que no fue posible modificar el registro.</li> </ul> </li> </ol> </li> </ul>

<ul style="list-style-type: none"> <li>• <b>Excepciones:</b> <ul style="list-style-type: none"> <li>○ Si la conexión entre el servidor y la base no es exitosa, se producirá una excepción cuando se pretendan realizar las consultas.</li> </ul> </li> </ul>
<b>Post-condiciones:</b>
Si ocurre una falla en el servidor central que interrumpa el flujo normal de las aplicaciones, se debe validar nuevamente el ingreso de los usuarios al sistema.
<b>Recursos Especiales Implementados</b>
<ul style="list-style-type: none"> <li>• Base de Datos: <ul style="list-style-type: none"> <li>○ Tabla Usuarios.</li> </ul> </li> </ul>
<b>Interfaces de Usuario Relacionadas</b>
<ul style="list-style-type: none"> <li>• gestionusuarios.jsp</li> </ul>

**Tabla A.1 (n).** Descripción detallada para el caso de uso Modificar Cuenta.

<b>ELIMINAR CUENTA</b>
<b>Iniciador</b>
Administrador
<b>Precondiciones</b>
El administrador previamente debe haber obtenido el permiso para ingresar al sistema una vez se hayan validado sus datos.
<b>Flujo de eventos</b>
<ul style="list-style-type: none"> <li>• <b>Flujo principal:</b> <ol style="list-style-type: none"> <li>1. El administrador selecciona el tipo de usuario de la cuenta que se va a modificar, ingresa la cedula del usuario y presiona el botón “Buscar”.</li> <li>2. El sistema despliega en pantalla los campos de texto con la información correspondiente a la cuenta del usuario.</li> <li>3. El administrador presiona el botón “Eliminar Usuario”.</li> <li>4. El sistema elimina de la base de datos el registro correspondiente a la cuenta del usuario.</li> </ol> </li> <li>• <b>Flujos Alternativos:</b> <ol style="list-style-type: none"> <li>5. Si el proceso de creación de modificación de un registro no resulta exitoso, el sistema despliega en pantalla una alerta implementada en Java Script notificando el error. La respuesta enviada es: <ul style="list-style-type: none"> <li>○ <i>ERROR</i>: Indica que no fue posible eliminar el registro.</li> </ul> </li> </ol> </li> <li>• <b>Excepciones:</b> <ul style="list-style-type: none"> <li>○ Si la conexión entre el servidor y la base no es exitosa, se producirá una excepción cuando se pretendan realizar las consultas.</li> </ul> </li> </ul>
<b>Post-condiciones:</b>
Si ocurre una falla en el servidor central que interrumpa el flujo normal de las aplicaciones, se debe validar nuevamente el ingreso de los usuarios al sistema.
<b>Recursos Especiales Implementados</b>
<ul style="list-style-type: none"> <li>• Base de Datos:</li> </ul>

<ul style="list-style-type: none"> <li>○ Tabla Usuarios.</li> </ul>
<b>Interfaces de Usuario Relacionadas</b>
<ul style="list-style-type: none"> <li>• gestionusuarios.jsp</li> </ul>

**Tabla A.1 (o).** Descripción detallada para el caso de uso Eliminar Cuenta.

BUSCAR TAREAS
<b>Iniciador</b>
Operador
<b>Precondiciones</b>
El operador previamente debe haber obtenido el permiso para ingresar al sistema una vez se hayan validado sus datos.
<b>Flujo de eventos</b>
<ul style="list-style-type: none"> <li>• <b>Flujo principal:</b> <ol style="list-style-type: none"> <li>1. El operador ingresa la fecha en la cual se asigno la tarea a buscar y presiona el botón "Buscar".</li> <li>2. El sistema recorre la tabla tareas de la base de datos y despliega en pantalla mediante una tabla la información de las tareas asignadas en la fecha ingresada.</li> </ol> </li> <li>• <b>Flujos Alternativos:</b> <ol style="list-style-type: none"> <li>3. Si el proceso de búsqueda no resulta exitoso, el sistema despliega en pantalla una alerta implementada en Java Script notificando el error. Las respuesta enviada es: <ul style="list-style-type: none"> <li>○ <i>ERROR</i>: Indica que no existe una tarea asociada a la fecha ingresada.</li> </ul> </li> </ol> </li> <li>• <b>Excepciones:</b> <ul style="list-style-type: none"> <li>○ Si la conexión entre el servidor y la base no es exitosa, se producirá una excepción cuando se pretendan realizar las consultas.</li> </ul> </li> </ul>
<b>Post-condiciones:</b>
Si ocurre una falla en el servidor central que interrumpa el flujo normal de las aplicaciones, se debe validar nuevamente el ingreso de los usuarios al sistema.
<b>Recursos Especiales Implementados</b>
<ul style="list-style-type: none"> <li>• Base de Datos: <ul style="list-style-type: none"> <li>○ Tabla tareas.</li> </ul> </li> </ul>
<b>Interfaces de Usuario Relacionadas</b>
<ul style="list-style-type: none"> <li>• plantilla_operador.jsp</li> <li>• gestión_tareas.jsp</li> <li>• buscar_tareas.jsp</li> </ul>

**Tabla A.1 (p).** Descripción detallada para el caso de uso Buscar Tareas.

<b>ASIGNAR TAREA</b>
<b>Iniciador</b>
Operador
<b>Precondiciones</b>
El operador previamente debe haber obtenido el permiso para ingresar al sistema una vez se hayan validado sus datos.
<b>Flujo de eventos</b>
<ul style="list-style-type: none"> <li>• <b>Flujo principal:</b> <ol style="list-style-type: none"> <li>1. El operador presiona el botón “Agregar” en la interfaz de gestión de tareas.</li> <li>2. El sistema despliega en pantalla una tabla con los campos de texto para ingresar la información relacionada con la tarea a asignar.</li> <li>3. El administrador ingresa la información y presiona el botón “Agregar”.</li> <li>4. El sistema crea un nuevo registro en la base de datos con la información de la tarea.</li> </ol> </li> <li>• <b>Flujos Alternativos:</b> <ol style="list-style-type: none"> <li>5. Si el proceso de creación de un nuevo registro no resulta exitoso, el sistema despliega en pantalla una alerta implementada en Java Script notificando el error. La respuestas enviada es:                             <ul style="list-style-type: none"> <li>○ <i>ERROR</i>: Indica que no fue posible crear un nuevo registro.</li> </ul> </li> </ol> </li> <li>• <b>Excepciones:</b> <ul style="list-style-type: none"> <li>○ Si la conexión entre el servidor y la base no es exitosa, se producirá una excepción cuando se pretendan realizar las consultas.</li> </ul> </li> </ul>
<b>Post-condiciones:</b>
Si ocurre una falla en el servidor central que interrumpa el flujo normal de las aplicaciones, se debe validar nuevamente el ingreso de los usuarios al sistema.
<b>Recursos Especiales Implementados</b>
<ul style="list-style-type: none"> <li>• Base de Datos:                             <ul style="list-style-type: none"> <li>○ Tabla tareas.</li> </ul> </li> </ul>
<b>Interfaces de Usuario Relacionadas</b>
<ul style="list-style-type: none"> <li>• plantilla_operador.jsp</li> <li>• gestión_tareas.jsp</li> <li>• agregar_tarea.jsp</li> </ul>

**Tabla A.1 (q).** Descripción detallada para el caso de uso Asignar Tareas.

<b>MODIFICAR TAREA</b>
<b>Iniciador</b>
Operador
<b>Precondiciones</b>
El operador previamente debe haber obtenido el permiso para ingresar al sistema una vez se hayan validado sus datos.
<b>Flujo de eventos</b>
<ul style="list-style-type: none"> <li>• <b>Flujo principal:</b> <ol style="list-style-type: none"> <li>1. El operador ingresa el ID de la tarea a modificar y presiona el botón “Modificar”.</li> </ol> </li> </ul>

<ol style="list-style-type: none"> <li>2. El sistema despliega en pantalla una tabla con la información correspondiente a la tarea seleccionada.</li> <li>3. El operador modifica los campos necesarios y presiona el botón “Actualizar”.</li> <li>4. El sistema actualiza la información en el registro correspondiente a la tarea seleccionada.</li> </ol> <ul style="list-style-type: none"> <li>• <b>Flujos Alternativos:</b> <ol style="list-style-type: none"> <li>4. Si el proceso de modificación de un registro no resulta exitoso, el sistema despliega en pantalla una alerta implementada en Java Script notificando el error. La respuesta enviada es: <ul style="list-style-type: none"> <li>○ <i>ERROR</i>: Indica que no fue posible modificar el registro.</li> </ul> </li> </ol> </li> <li>• <b>Excepciones:</b> <ul style="list-style-type: none"> <li>○ Si la conexión entre el servidor y la base no es exitosa, se producirá una excepción cuando se pretendan realizar las consultas.</li> </ul> </li> </ul>
<b>Post-condiciones:</b>
Si ocurre una falla en el servidor central que interrumpa el flujo normal de las aplicaciones, se debe validar nuevamente el ingreso de los usuarios al sistema.
<b>Recursos Especiales Implementados</b>
<ul style="list-style-type: none"> <li>• Base de Datos: <ul style="list-style-type: none"> <li>○ Tabla tareas.</li> </ul> </li> </ul>
<b>Interfaces de Usuario Relacionadas</b>
<ul style="list-style-type: none"> <li>• plantilla_operador.jsp</li> <li>• gestión_tareas.jsp</li> <li>• modificar_tarea.jsp</li> </ul>

**Tabla A.1 (r).** Descripción detallada para el caso de uso Modificar Tarea.

ELIMINAR TAREA
<b>Iniciador</b>
Operador
<b>Precondiciones</b>
El operador previamente debe haber obtenido el permiso para ingresar al sistema una vez se hayan validado sus datos.
<b>Flujo de eventos</b>
<ul style="list-style-type: none"> <li>• <b>Flujo principal:</b> <ol style="list-style-type: none"> <li>1. El administrador, ingresa el ID de la tarea a eliminar y presiona el botón “Eliminar”.</li> <li>2. El sistema elimina de la base de datos el registro correspondiente al ID de la tarea ingresado.</li> </ol> </li> <li>• <b>Flujos Alternativos:</b> <ol style="list-style-type: none"> <li>3. Si el proceso de creación de modificación de un registro no resulta exitoso, el sistema despliega en pantalla una alerta implementada en Java Script notificando el error. La respuesta enviada es: <ul style="list-style-type: none"> <li>○ <i>ERROR</i>: Indica que no fue posible eliminar el registro.</li> </ul> </li> </ol> </li> <li>• <b>Excepciones:</b></li> </ul>

<ul style="list-style-type: none"> <li>○ Si la conexión entre el servidor y la base no es exitosa, se producirá una excepción cuando se pretendan realizar las consultas.</li> </ul>
<b>Post-condiciones:</b>
Si ocurre una falla en el servidor central que interrumpa el flujo normal de las aplicaciones, se debe validar nuevamente el ingreso de los usuarios al sistema.
<b>Recursos Especiales Implementados</b>
<ul style="list-style-type: none"> <li>• Base de Datos:             <ul style="list-style-type: none"> <li>○ Tabla tareas.</li> </ul> </li> </ul>
<b>Interfaces de Usuario Relacionadas</b>
<ul style="list-style-type: none"> <li>• plantilla_operador.jsp</li> <li>• gestión_tareas.jsp</li> </ul>

**Tabla A.1 (s).** Descripción detallada para el caso de uso Eliminar Tarea.

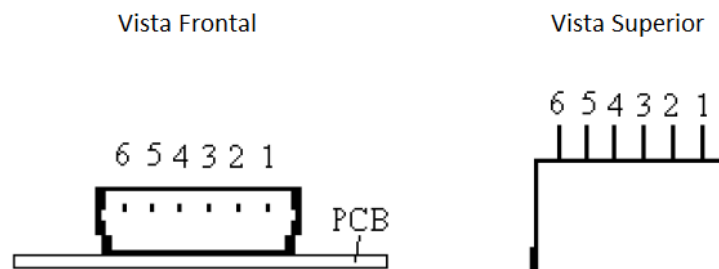
### A.3 ASPECTOS COMPLEMENTARIOS DEL CAPÍTULO 3 – IMPLEMENTACIÓN DE HORUS.

En esta sección se explican detalladamente aspectos relacionados con la implementación del sistema prototipo de control y monitoreo vehicular asistido por GPS - HORUS.

#### A.3.1 Módulo GPS

##### A.3.1.1 Descripción de pines del módulo GPS.

En la figura A.7 se muestra la distribución de pines del módulo, y en la tabla A.2 se presentan las funciones de cada uno.



**Figura A.7.** Diagrama de pines del módulo GPS EM-406A.

PIN	Función
1	GND
2	Vin
3	Rx
4	Tx
5	GND
6	1 PPS

**Tabla A.2.** Funciones de los pines del módulo GPS.

- **Vin:** Voltaje de alimentación del módulo (4.5 a 6.5 V).
- **Tx:** canal de transmisión para la salida de datos de navegación y medida del módulo GPS.
- **Rx:** canal de recepción para el ingreso de comandos de configuración del módulo GPS.
- **GND:** provee la tierra de la placa del módulo GPS.
- **1 PPS:** este pin provee una señal de salida de un pulso por segundo, la cual está sincronizada con la señal de reloj del GPS.

### A.3.1.2 Protocolo de comunicación del módulo GPS

Como protocolo de comunicación con el módulo GPS se usó el protocolo NMEA. El protocolo NMEA especifica diferentes tipos de mensajes que ofrecen información variada, como velocidad, coordenadas, fecha y hora, número de satélites, etc. En la tabla A.3 se muestran los datos entregados por cada tipo de mensaje manejado por el protocolo NMEA.

TIPO DE MENSAJE	DESCRIPCIÓN
<b>GGA</b>	Hora, posición y datos de determinación de la posición.
<b>GLL</b>	Latitud, longitud, hora y estado.
<b>GSA</b>	Modo de operación del receptor GPS, número de satélites usados para el cálculo de la posición.
<b>GSV</b>	Número de satélites en vista, elevación, ángulo azimutal y relación señal a ruido.
<b>MSS</b>	Relación señal a ruido, potencia de recepción de la señal, frecuencia.
<b>RMC</b>	Hora, fecha, posición, curso y velocidad.
<b>VTG</b>	Curso y velocidad relativa a la tierra.
<b>ZDA</b>	Señal de un pulso por segundo.

**Tabla A.3.** Mensajes de salida NMEA.

Para el desarrollo de HORUS se utilizó únicamente el mensaje RMC<sup>4</sup> entregado por el módulo GPS, debido a que es información suficiente relacionada con la posición y

<sup>4</sup> RMC: Recommended Minimum Specific GNSS Data.



velocidad.

A continuación se presenta un mensaje RMC a manera de ejemplo, y en la tabla A.4 se hace una descripción de los valores de cada campo:

**\$GPRMC,161229.487,A,3723.2475,N,12158.3416,W,0.13,309.62,120598,\*10**

NOMBRE	VALOR	DESCRIPCIÓN
ID del mensaje	\$GPRMC	Encabezado RMC
Hora Universal	161229.487	hhmmss.sss
Estado	A	A=Datos Válidos ; V=Datos Inválidos
Latitud	3723.2475	ddmm.mmmm <sup>5</sup>
Indicador N/S	N	N=norte ; S=Sur
Longitud	12158.3416	dddmm.mmmm
Indicador E/W	W	E=este ; W=oeste
Velocidad	0.13	Velocidad en nudos
Curso	309.62	
Fecha	120598	ddmmaa
Variación Magnética	Vacío	No soportado por el módulo
Checksum	*10	
<CR><LF>		Caracteres de control indicadores de fin del mensaje

**Tabla A.4.** Descripción de los valores de un mensaje RMC.

### A.3.1.3 Configuración del GPS

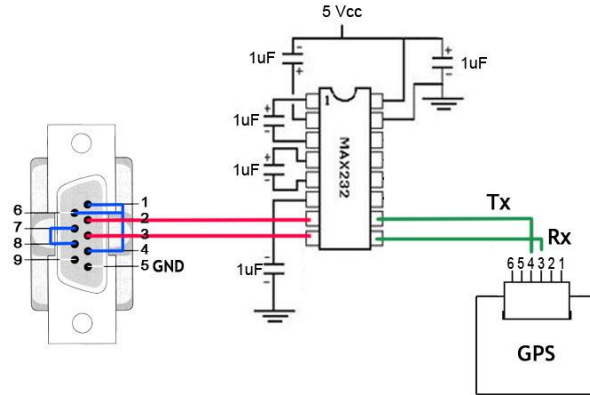
Para realizar la conexión entre el módulo GPS y el computador, se utilizó el puerto serial. En la figura A.8 se muestra el circuito utilizado, el cual hace uso de un conector DB9 hembra conectado al puerto serial. El conector se encuentra en configuración *Null Modem*, es decir se cortocircuitaron algunas líneas de transmisión y recepción de tal forma que el computador considera como un modem al dispositivo conectado y para la comunicación solo se utilizan las líneas de transmisión (Tx) y recepción (Rx).

Se hizo uso del circuito integrado MAX232 para convertir los niveles del estándar RS232 (15 a -15V) a señales de 0 y 3V.

Por configuración de fábrica, el módulo GPS entrega los mensajes GSA, RMC, GGA y GSV. Asimismo, la tasa por defecto del dispositivo es de 4800 baudios. Para el caso de HORUS, el módulo se configuró para que entregara únicamente el mensaje RMC a una tasa de 9600 baudios. Para realizar dicha configuración, se utilizó el software SiRFDemo, propietario de la compañía fabricante del módulo GPS, el cual es una

<sup>5</sup> d: grados (*degrees*); m: minutos (*minutes*); s: segundos (*seconds*). Sistema de Coordenadas Universal Transversal de Mercator (UTM, *Universal Transverse Mercator*).

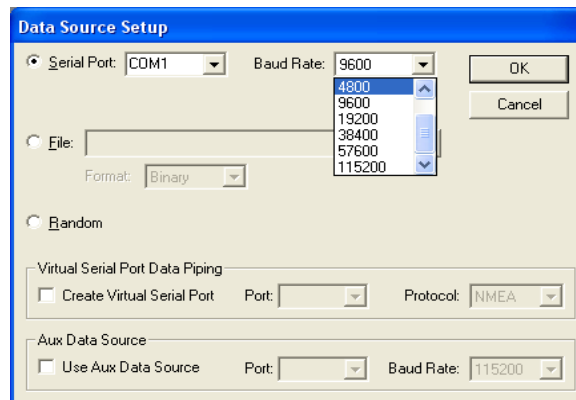
herramienta que permite monitorear en tiempo real la operación de un dispositivo GPS y obtener información relacionada, tal como satélites localizados, fuerza de la señal y posición actual.



**Figura A.8.** Conexión del módulo GPS a un computador usando un MAX232 y un conector DB9 Hembra.

Por configuración de fábrica, el módulo GPS entrega los mensajes GSA, RMC, GGA y GSV. Asimismo, la tasa por defecto del dispositivo es de 4800 baudios. Para el caso de HORUS, el módulo se configuró para que entregara únicamente el mensaje RMC a una tasa de 9600 baudios. Para realizar dicha configuración, se utilizó el software SiRFDemo, propietario de la compañía fabricante del módulo GPS, el cual es una herramienta que permite monitorear en tiempo real la operación de un dispositivo GPS y obtener información relacionada, tal como satélites localizados, fuerza de la señal y posición actual.

Al iniciar SiRF Demo, se configuró el puerto serial y la velocidad del computador a la misma configurada por el GPS, la cual es por defecto 4800. En la figura A.9 se observa esta configuración.



**Figura A.9.** Configuración inicial de SiRF Demo.

Al iniciar la conexión entre el software y el módulo GPS, se aprecian diversos parámetros relacionados con el funcionamiento del dispositivo de localización, los cuales se presentan en la figura A.10. La ventana “Signal View” muestra los satélites con sus respectivos niveles de potencia de recepción que el módulo GPS detecta. Los que se encuentran de color verde, son aquellos satélites que están siendo rastreados por el módulo GPS y que se usan para realizar el cálculo de posición; los azules son satélites rastreados, pero que no están siendo usados para calcular posición; y los de color rojo son aquellos satélites que no son detectados por el módulo GPS, pero se conoce su posición aproximada debido a una configuración del GPS.

En la ventana “Radar View” se observan los satélites anteriormente descritos, identificados por sus respectivos colores. La ventana “Debug View - NMEA” es la de interés del trabajo de grado, y es la que muestra la respuesta del módulo GPS. En este caso, el GPS entrega por defecto los mensajes GPGLA, GPGSA, GPGSV y GPRMC.

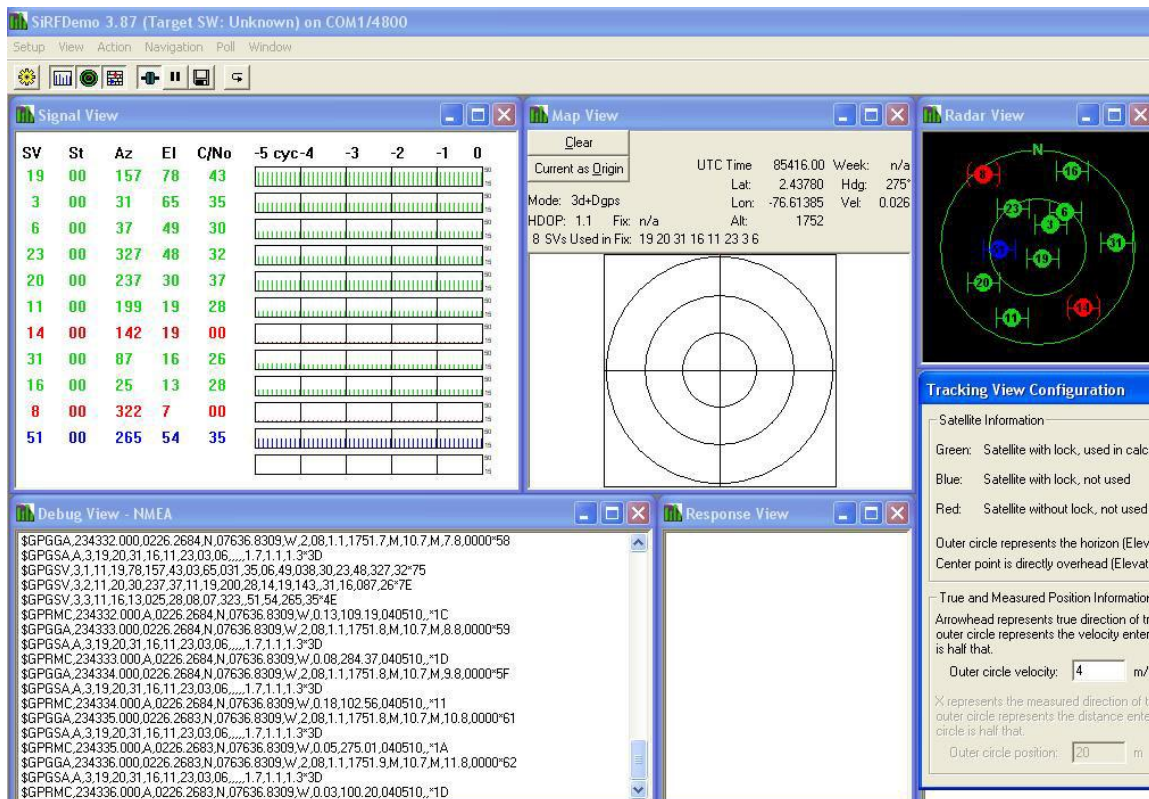
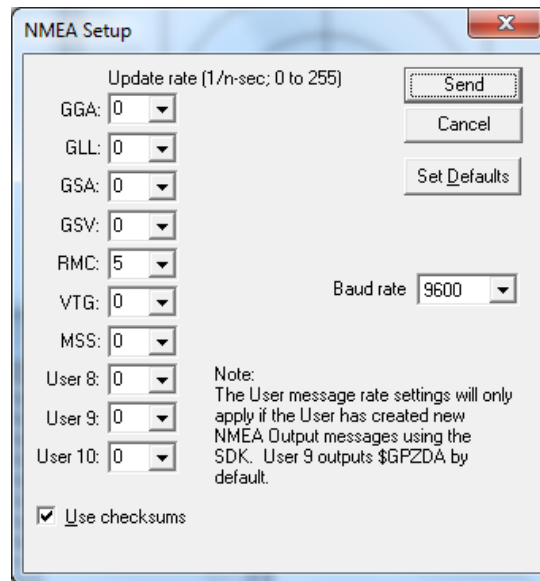


Figura A.10. Parámetros GPS mostrados por el software SiRF Demo.

En el menú “Action” del programa, se selecciona la opción “Switch to NMEA protocol” y aparece la ventana mostrada en la figura A.11.



**Figura A.11.** Configuración del módulo GPS.

En esta ventana se presentan todos los mensajes manejados por el protocolo NMEA, los valores numéricos que aparecen al lado de cada mensaje se refiere a la periodicidad con la que se desea el mensaje sea entregado por el modulo GPS, dada en segundos. Solamente se activo el mensaje RMC, con una periodicidad de 5 segundos.

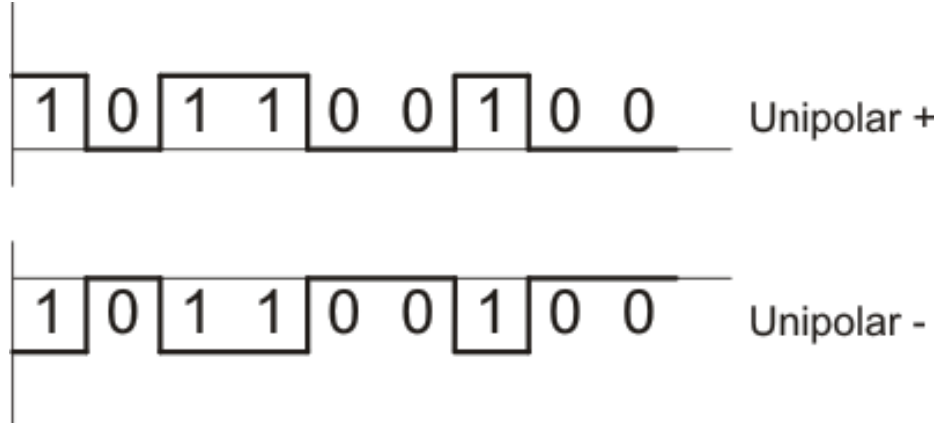
### A.3.2 Códigos de Línea

Los códigos de línea surgen ante la necesidad de transmitir señales digitales a través de diversos medios de transmisión. Una señal digital es una secuencia de pulsos de tensión discretos y discontinuos, donde cada pulso es un elemento de la señal.

Los códigos de línea son utilizados en un sistema de comunicaciones para realizar el transporte digital de datos. La representación de la onda se suele realizar mediante un número determinado de impulsos. Estos impulsos representan los “unos” y los “ceros” digitales. Los tipos más comunes de codificación en línea son el unipolar, polar y bipolar.

#### A.3.2.1 Codificación Digital Unipolar

Usa una sola polaridad, codificando únicamente uno de los estados binarios (el “uno”) que toma una polaridad positiva (o puede ser negativa), es decir que para todo el tren de pulsos toma un mismo valor de voltaje. El estado “cero” se representa por 0 voltios. La figura A.12 muestra la codificación unipolar No Retorno a Cero (NRZ, *Non Return to Zero*).



**Figura A.12.** NRZ Unipolar.

A pesar de ser una codificación muy sencilla permite representar los conceptos usados con sistemas de codificación más complejos y permite examinar los tipos de problemas que se deben resolver en los sistemas de transmisión digital.

### **A.3.2.2 Codificación Digital Polar**

La codificación polar utiliza dos niveles de voltaje: uno positivo y otro negativo.

#### **A.3.2.2.1 No Retorno a Cero (NRZ)**

El voltaje de la señal no vuelve a cero entre bits consecutivos de valor “uno”. Este tipo de código no es recomendable en largas distancias principalmente por dos motivos. En primer lugar presentan niveles residuales de corriente continua y en segundo lugar por la posible ausencia de suficientes números de transiciones de señal que permitan la recuperación fiable de una señal de temporización.

- NRZ - L: El nivel de la señal depende del tipo de bit que representa. Generalmente, un valor de voltaje positivo indica un “cero” y un valor de voltaje negativo indica un “uno”, aunque también puede darse el caso contrario. El nivel de señal depende del estado del bit.
- NRZ - I: Es una inversión del nivel de voltaje que representa un bit “uno”. Es la transición entre el valor de voltaje positivo y negativo lo que representa un bit “uno”, y no los voltajes en sí mismos. Un bit “cero” se representa sin ningún cambio. Es mejor que NRZ-L debido a que existe una sincronización implícita gracias al cambio de señal cada vez que se encuentra un “uno”.

### A.3.2.2.2 Retorno a Cero (RZ, Return to Zero)

Utiliza tres valores: positivo, negativo y cero. Un bit 'uno' se representa por una transición de positivo a cero y un bit 'cero' se representa con la transición de negativo a cero, con retorno de voltaje 0 a la mitad de tiempo de bit.

### A.3.2.2.3 Bifase

En este método, la señal cambia en medio del intervalo del bit, pero no retorna a cero, sino que continúa el resto del intervalo con la polaridad opuesta. Existen dos tipos de codificación Bifase:

- Manchester: Una transición de polaridad de positiva a negativa representa el valor binario "cero", y una transición de negativa a positiva representa un "uno".
- Manchester Diferencial: Necesita dos cambios de señal para representar el bit "cero", pero sólo uno para representar el bit "uno". Es decir, una transición de polaridad inversa a la del bit previo representa el "cero" y una transición igual representa el "uno".

### A.3.2.3 Codificación Digital Bipolar

La codificación digital bipolar, utiliza tres estados: Positivo, Negativo y Cero. El nivel de voltaje cero se utiliza para representar un bit "cero". Los bits "uno" se codifican como valores positivo y negativo de forma alternada. Si el primer "uno" se codifica con una amplitud positiva, el segundo lo hará con una amplitud negativa, el tercero positiva y así sucesivamente. Siempre se produce una alternancia entre los valores de amplitud para representar los bits "uno", aunque estos bits no sean consecutivos. La figura A.13 muestra la codificación bipolar de un tren de pulsos.

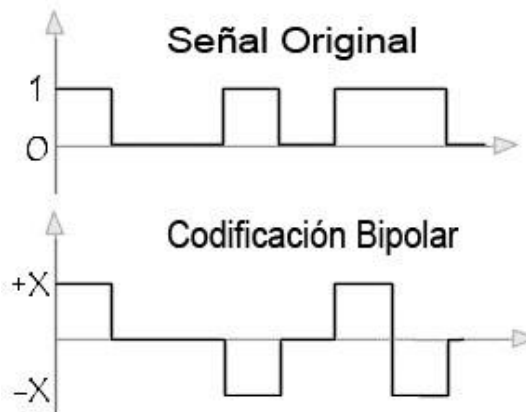
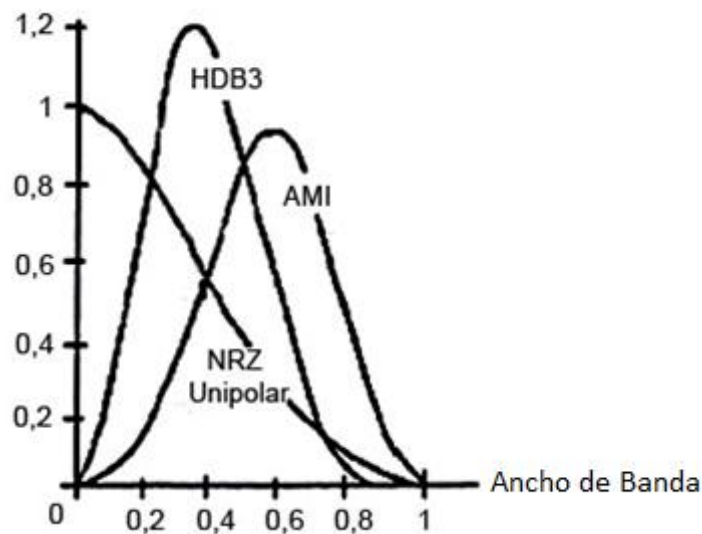


Figura A.13. Codificación bipolar de una señal binaria.

### A.3.2.3.1 Código de Alta Densidad de 3 Ceros (HDB3, High Density Bipolar)

#### A.3.2.3.1.1 Características y Funcionamiento de HDB3

En la codificación de línea HDB3, los unos lógicos tienen polaridades inversas, y los ceros lógicos toman el valor de voltaje cero. Este tipo de señal no tiene componente continua ni de bajas frecuencias. La codificación HDB3 ocupa un ancho de banda menor que el usado por AMI y el NRZ unipolar, aproximadamente del 80%. La figura A.14 muestra esta comparación de anchos de banda. Otra ventaja de la codificación HDB3 frente al NRZ unipolar es la carencia de la componente DC.



**Figura A.14.** Ancho de banda utilizado por codificaciones tipo NRZ Unipolar, AMI y HDB3.

Se presenta el inconveniente que cuando aparece una larga cadena de ceros se puede perder el sincronismo al no distinguir un bit de los adyacentes. Con el fin de evitar esta situación, se establece que en las cadenas de 4 ceros el cuarto sea reemplazado por un bit llamado “bit de violación” el cual tiene el mismo valor de voltaje de un uno lógico.

En las violaciones posteriores (cadenas de 4 ceros) se hace el reemplazo por una de estas dos secuencias:

**000V**  
**B00V**

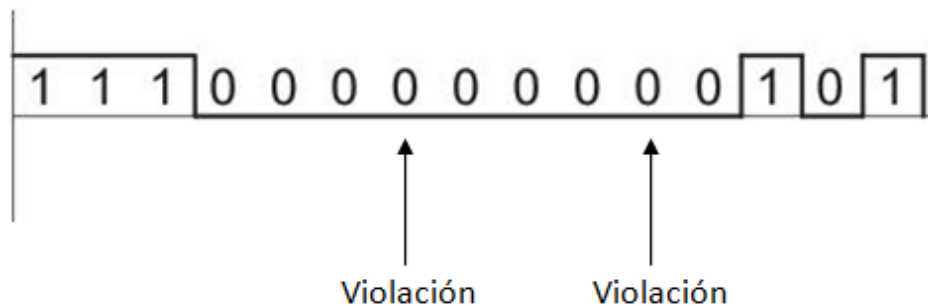
Donde:

- **V:** bit de violación, de igual polaridad al pulso precedente.
- **B:** bit de relleno, el cual es un pulso de polaridad inversa al anterior.

Para decidir la secuencia a utilizar, se deben contar la cantidad de unos existentes entre la última violación y la actual, si la cantidad es par se emplea “B00V”, si es impar entonces “000V”. Para determinar la secuencia de la primera violación se cuenta la cantidad de unos que hay desde el inicio de la trama hasta que la violación ocurra y se determina si es una cantidad par o impar con el fin de seleccionar la secuencia correspondiente, o bien puede establecerse arbitrariamente que la secuencia “000V” corresponderá siempre a la primera violación.

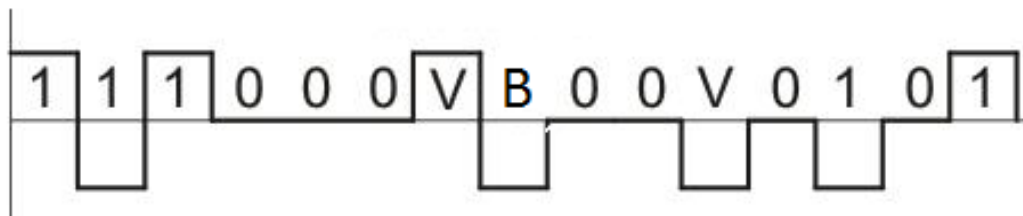
El primer pulso de una violación lleva la misma polaridad del último 1 transmitido de manera que se detecte que es un bit de violación. En la combinación “B00V” el bit de violación y el de relleno poseen la misma polaridad.

En la figura A.15 se muestra una señal a transmitir, y se han marcado las posiciones de las violaciones que se presentan, cada cuatro ceros:



**Figura A.15.** Posición de bit donde se generan las violaciones.

Al codificar con HDB3, las secuencias de unos tienen polaridades inversas, y los ceros lógicos se representan con valor de voltaje cero. En cuanto a las violaciones, los cuatro primeros ceros se sustituyen por “000V” debido a que se trata de la primera violación y los cuatro siguientes por “B000V” debido a que entre la última violación y ésta hay un número par de unos (en este caso no hay ningún uno, por lo que se toma como un valor par). En la figura A.16 se muestra la señal obtenida.



**Figura A.16.** Codificación HDB3.



### A.3.2.3.1.2 Generación de la codificación HDB3

Para explicar cómo se realiza el proceso de codificación HDB3, se recurre al concepto de conversión digital-analógica. Básicamente la conversión basa en realizar la suma ponderada de los diversos bits que conforman la señal binaria. El valor relativo de cada uno viene dado por la correspondiente potencia de dos, mostrada en la ecuación B.1. En la figura A.17 se muestra el concepto de múltiples entradas de bits al convertor digital analógico, y una salida analógica.

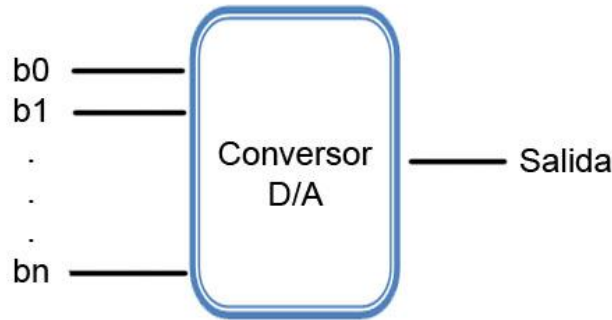


Figura A.17. Convertor Digital-Analógico.

$$S = \sum_{k=0}^n b_k 2^k = A_o (b_0 + b_1 2^1 + b_2 2^2 + \dots + b_n 2^n) \quad (\text{B.1})$$

Donde  $A_o$  es un factor de conversión que permite cambiar los estados lógicos en sus correspondientes valores de voltaje.

La conversión digital-analógica puede realizarse de forma sencilla utilizando un amplificador operacional y configurándolo en forma de sumador inversor. En la entrada inversora del amplificador se conectan las señales de bits que se convertirán en señales analógicas. En la figura A.18 se muestra un amplificador operacional en configuración sumador inversor, con dos entradas de bits.

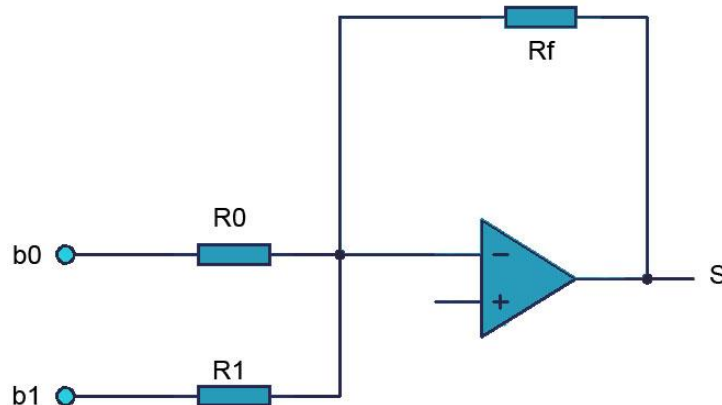


Figura A.18. Amplificador operacional en configuración sumador inversor.

Al ser un sumador inversor, los niveles obtenidos tienen valores negativos, por tal razón es necesario adicionar un voltaje positivo ya que HDB3 requiere de un nivel positivo. Esto se logra adicionando un voltaje de referencia positivo ( $V_{ref}$ ) en la entrada no inversora del amplificador operacional, de manera que la salida del circuito está dada por la ecuación B.2:

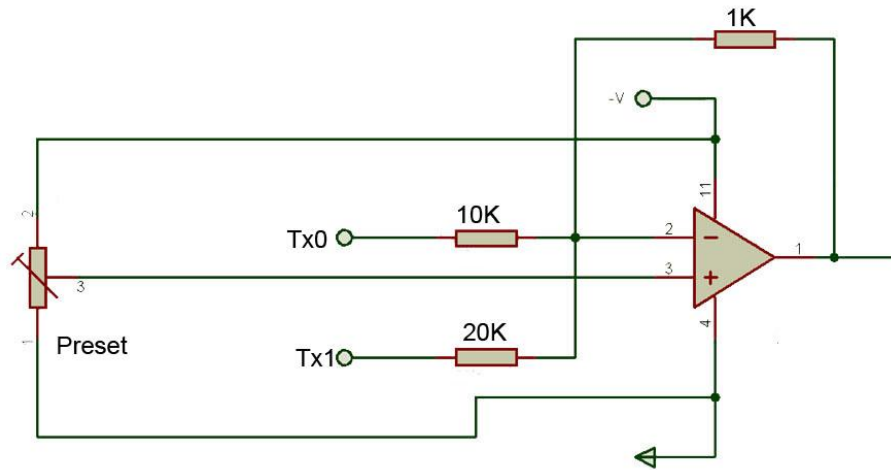
$$S_{b_1b_0} = -\frac{R_f}{R_0}b_0 - \frac{R_f}{R_1}b_1 + V_{ref} \quad (B.2)$$

Lo cual corresponde con el esquema de conversión digital-analógica. La ecuación 1.3 se establece al definir la ecuación B.2 en forma de la ecuación B.1:

$$S_{b_1b_0} = -A_o(b_0 + b_12^1) + V_{ref} \quad (B.3)$$

Para el caso de la generación de los niveles HDB3 implementado en HORUS, se definieron como niveles de salida +250 mV, 0 V y -250 mV. El módulo generador de niveles HDB3 implementado en la tarjeta de adaptación radio cuenta con el amplificador operacional en configuración sumador inversor, al cual se le aplican dos entradas de bits en su entrada inversora provenientes de los pines 35 y 36 del microcontrolador. Se calculó un valor  $A_o$  y un voltaje de referencia ( $V_{ref}$ ) de tal forma que se obtuvieran los niveles de voltajes requeridos.

En la figura A.19 se muestra la configuración del módulo.



**Figura A.19.** Módulo generador de los niveles HDB3.

Los pines Tx0 y Tx1 son las entradas, con las cuales se determina la salida del circuito. Para hallar los valores, se define la ecuación B.4 en base a la ecuación B.3.

$$S_{Tx_0Tx_1} = -A_o(b_0 + b_12^1) + V_{ref} \quad (B.4)$$

Teniendo Tx0 y Tx1 en valor “0”, se obtiene el valor más alto de voltaje (Marca Positiva). Haciendo uso de la ecuación B.4 se define la ecuación B.5:

$$S_{00} = -A_o(0 + 0) + Vref \quad (B.5)$$

$$S_{00} = Vref \quad (B.6)$$

Debido a que la marca positiva ( $S_{00}$ ) es igual al voltaje de referencia, éste se fijó en un valor de +250 mV, debido a que este valor corresponde a la marca positiva de la codificación HDB3 utilizada por HORUS. La ecuación B.7 muestra el voltaje de referencia definido como 250 mV.

$$Vref = 250 \text{ mV} \quad (B.7)$$

Este valor se fija de manera experimental utilizando un *preset*, el cual posee resistencia variable y se ajusta hasta encontrar el valor de voltaje correcto.

Asimismo, se definieron las combinaciones de entradas digitales que generan las señales de salida deseadas. Reemplazando Tx0=0 y Tx1=1 en la ecuación B.4 se obtiene la ecuación B.8:

$$S_{01} = -A_o(1 + 0) + Vref \quad (B.8)$$

Donde  $b_0 = Tx_1 = 1 = 5 \text{ V}$  correspondiente a un uno lógico, por lo tanto se define la ecuación B.9:

$$S_{01} = -A_o(5V) + Vref \quad (B.9)$$

El valor de  $A_o$  se encuentra igualando a cero debido a que el estado Tx0=0 y Tx1=1 corresponde al estado 0 V, entonces:

$$-A_o(5V) + Vref = 0$$

$$A_o(5V) = Vref$$

Reemplazando el voltaje de referencia definido en la ecuación B.7 se obtiene la ecuación B.10:

$$A_o(5V) = 250\text{mV}$$

$$A_o = \frac{250mV}{5V}$$

$$A_o = 0.05 \quad (B.10)$$

Reemplazando en la ecuación B.9 los valores obtenidos de B.7 y B.10 se tiene:

$$S_{01} = -0.05(5V) + 250mV$$

$$S_{01} = 0V \quad (B.11)$$

Para el valor -250 mV (Marca Negativa), usando la ecuación B.4 se obtiene la ecuación B.12:

$$S_{10} = -A_o(0 + 1 * 2) + Vref \quad (B.12)$$

Donde  $b_1 = Tx_0 = 1 = 5V$  correspondiente a un uno lógico, por lo tanto al reemplazar B.7 y B.10 en la ecuación B.12 se tiene:

$$S_{10} = -0.05(10V) + 250mV$$

$$S_{10} = -500mV + 250mV$$

$$S_{10} = -250mV \quad (B.13)$$

La tabla A.5 muestra las combinaciones de bits necesarias en las entradas Tx0 y Tx1 del amplificador operacional, con el fin de generar los niveles de voltaje definidos para HORUS y generar la codificación HDB3 de la señal. Estos valores corresponden a las ecuaciones B.6, B.11 y B.13.

Bits de Entrada (Tx0 – Tx1)	Voltaje de Salida
<b>0 - 0</b>	+ 250 mV
<b>0 - 1</b>	0 V
<b>1 - 0</b>	- 250 mV

**Tabla A.5.** Voltajes de salida respecto a bits de entrada.

Para encontrar los valores de las resistencias utilizadas se usan las ecuaciones B.14, B.15 y B.16.

$$A_o = \frac{Rf}{R0} \quad (B.14)$$

Al reemplazar B.10 en B.14 se tiene:

$$A_o = \frac{R_f}{R_0} = 0.05$$

$$R_f = 0.05 (R_0)$$

$$R_0 = 20 R_f \quad (\text{B.15})$$

Se escogió un valor para  $R_f$  de  $1 \text{ K}\Omega$ , por lo que se tiene un valor para  $R_0 = 20 \text{ K}\Omega$ . La ganancia para la entrada  $b_1$  es el doble, entonces:

$$R_1 = \frac{R_0}{2} = 10R_f \quad (\text{B.16})$$

$$R_1 = 10\text{K}\Omega$$

### **A.3.3 Descripción de Tarjetas.**

La unidad móvil y la estación de control están implementadas en tarjetas que les permiten realizar las funciones necesarias para establecer la comunicación entre los diferentes componentes del sistema. En esta sección se describen los módulos software y hardware que componen las tarjetas.

#### **A.3.3.1 Tarjetas Unidad Móvil.**

##### **A.3.3.1.1 Módulos Software de la Unidad Móvil.**

La unidad móvil compuesta por el receptor GPS, la unidad de control y acondicionamiento y la unidad de despliegue de información se encarga de controlar, organizar, adaptar, recibir, enviar y desplegar la información manejada por el sistema en el extremo móvil del sistema (vehículo). Para ello, se define el microcontrolador como el dispositivo controlador de los elementos que conforman a la unidad móvil. El software programado en el PIC se dividió en diferentes módulos que facilitaron estructurar el código desarrollado y la programación del sistema. En la figura A.20 se muestran los módulos software que componen la unidad móvil.

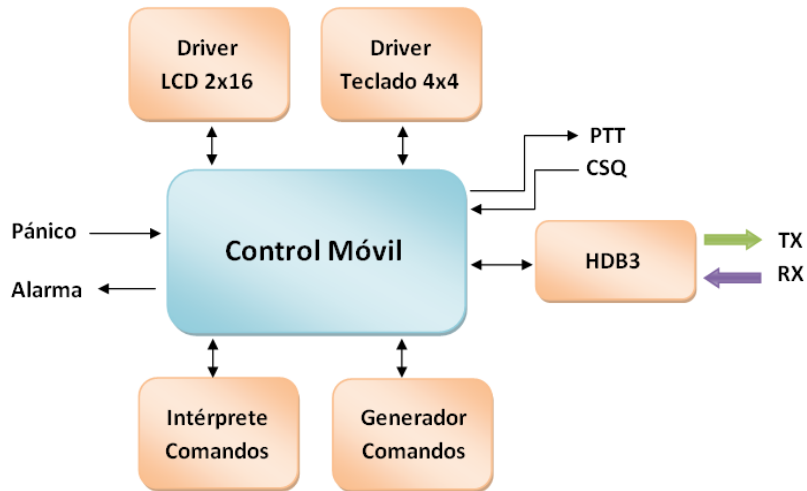


Figura A.20. Módulos software que componen a la unidad móvil.

- **Driver LCD 2x16:** módulo encargado de controlar la pantalla de cristal liquido 2x16 utilizada para desplegar información al conductor. El LCD tiene una interfaz de 3 pines de control y 8 de datos, los cuales son usados para desplegar los caracteres en el mismo. El módulo tiene 2 formas de configurar el bus de datos: utilizando los 8 pines de datos o utilizando sólo cuatro de ellos. En el caso de HORUS se utilizaron 4 pines de datos, lo cual reduce el número de conexiones hardware necesarias para implementar el sistema.
- **Driver Teclado 4x4:** módulo encargado de controlar el teclado matricial 4x4 usado en la interfaz de usuario de la unidad móvil. El método de lectura software del teclado consiste en rotar un estado “cero” a través de los cuatro pines de las filas, el cual es detectado en una de las columnas al momento de presionar una tecla. La codificación de la tecla se hace verificando en qué fila y columna está el cero.
- **Control Móvil:** módulo encargado de controlar y comunicar a los otros módulos que componen la unidad móvil. Aquí se encuentra el flujo principal del programa, se coordinan las acciones y se utilizan las funciones implementadas en los demás módulos.
- **Intérprete de Comandos:** módulo encargado de comprender las tramas del protocolo de HORUS recibidas y procesar la información de una manera más sencilla para que sea entendida fácilmente por los otros módulos.
- **Generador de Comandos:** módulo encargado de unir la información generada por los otros módulos y organizarla en las tramas definidas por el protocolo de HORUS.

Se han definido funciones que generan los comandos dependiendo de la información que se desee enviar.

- **HDB3:** módulo encargado de codificar/decodificar la información en HDB3 a nivel software. La trama es almacenada en memoria para codificarla/decodificarla. La codificación se realiza a través de funciones software programadas en base al funcionamiento de HDB3, alternando la polaridad de los estados “uno” e introduciendo el tipo de violación necesario dependiendo del caso. Los valores son entregados a la etapa hardware “Generador de Niveles HDB3”, a través de los pines 35 y 36 del microcontrolador.

La decodificación consiste en detectar las transiciones que ocurren en los pines 33 y 34 del microcontrolador. Cuando existe una transición de estado en el pin 33 se interpreta como una marca positiva, cuando la hay en el pin 34 se interpreta como una marca negativa. Cuando no ocurre una transición en ninguno de los pines, y transcurre un tiempo de bit<sup>6</sup>, se interpreta ese bit como un espacio (estado “cero”).

Cuando se detecta un “uno” en cualquiera de los pines 33 o 34, se usa en el código software una bandera para indicar que este bit se encuentra en un estado de identificación “pendiente”, al llegar el siguiente “uno” se compara su polaridad con el “uno” de estado pendiente. Si las polaridades son alternas, se identifica al “uno” pendiente como una marca, si las polaridades de los bits son iguales se cuentan cuántos ceros pasaron entre ellos para determinar a qué tipo de violación corresponde.

Pánico representa la señal de entrada que se genera cuando el conductor presiona el botón de pánico de la unidad móvil. Esta señal tiene prioridad sobre cualquier otro proceso que se esté llevando a cabo en la unidad móvil.

Alarma representa la señal enviada al *buzzer* para generar la señal sonora al momento de recibir un mensaje proveniente de la central.

PTT representa la señal enviada al pin 13 del radio, con el fin de ponerlo en un estado de transmisión.

CSQ representa la señal proveniente del pin 5 del radio, indicando que hay presencia de señal portadora.

Como se explicó en el documento principal, a nivel hardware la unidad móvil está dividida en dos tarjetas:

---

<sup>6</sup> El tiempo de bit para el caso de HORUS es de 500 ms, dado que la velocidad de transferencia de datos del sistema es de 2000 bps.

- Tarjeta de control y despliegue móvil.
- Tarjeta de adaptación radio.

A continuación se presentan los circuitos impresos de cada tarjeta. También se hace una descripción más detallada de las etapas hardware que conforman la tarjeta de adaptación radio.

### A.3.3.1.2 Tarjeta de Control y Despliegue Móvil

Los circuitos impresos de las tarjetas se generaron con la herramienta software ARES 7 Professional<sup>7</sup>, a partir de las conexiones creadas en ISIS 7 Professional. En la figura A.21 se muestra el circuito impreso de la tarjeta de control y despliegue móvil, donde se interconectan los componentes de las etapas hardware descritas en la sección 3.1.4.1 del documento principal.

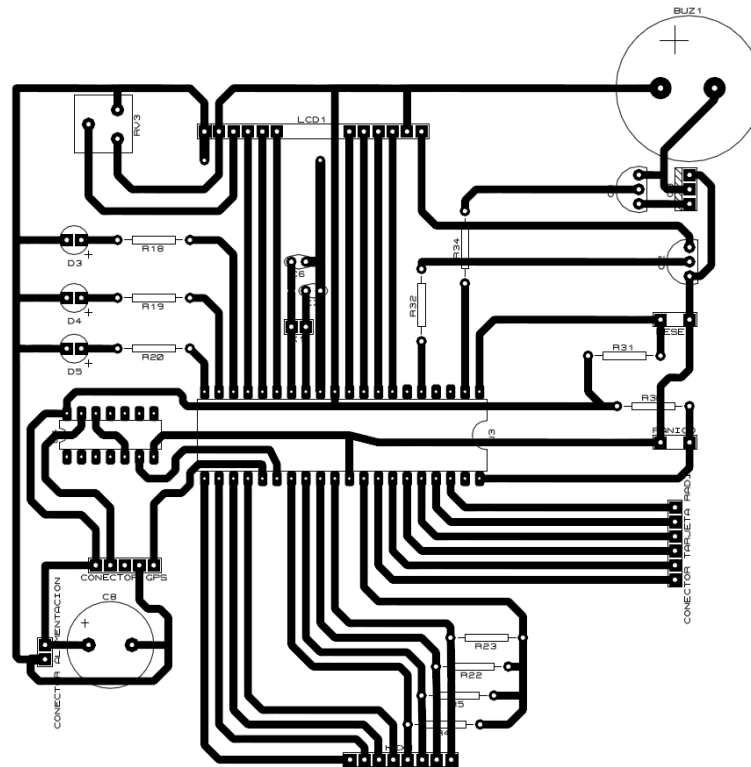


Figura A.21. Circuito impreso de la tarjeta de control y despliegue móvil.

### A.3.3.1.3 Tarjeta de Adaptación Radio

Esta tarjeta se encarga de realizar la codificación/decodificación HDB3 y acondicionar la señal para ser enviada a través del radio convencional de dos vías. A continuación

<sup>7</sup> <http://www.labcenter.com/>



se hace una descripción del funcionamiento de las etapas que conforman esta tarjeta.

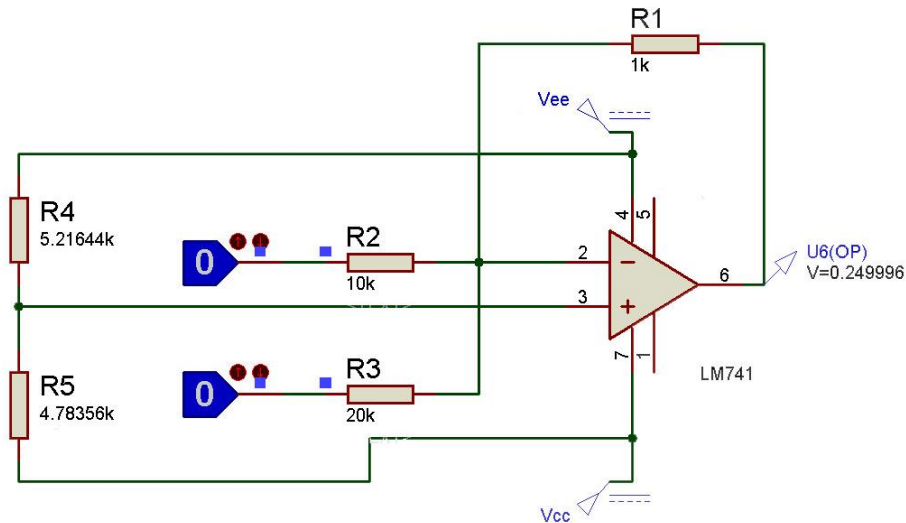
### A.3.3.1.3.1 Generador de niveles HDB3

En la sección 3.1.4.2 del documento principal se presentó el diagrama circuital del amplificador en configuración sumador inversor, utilizado para generar los diferentes niveles HDB3, usando una combinación de bits provenientes del PIC. En la tabla A.6 se resumen los valores de voltaje generados a partir de las combinaciones de bits de entrada.

Bits de Entrada (Tx0 – Tx1)	Voltaje de Salida
0 - 0	+ 250 mV
0 - 1	0 V
1 - 0	- 250 mV

**Tabla A.6.** Voltajes de salida en función a los bits de entrada del amplificador operacional.

En las figuras A.22 (a, b y c) se presenta la simulación del funcionamiento del circuito implementado en HORUS para generar los valores de la tabla A.6, con las diferentes combinaciones de bits a la entrada del sumador inversor. Las resistencias R4 y R5 representan el *preset* utilizado en el montaje real.



**Figura A.22 (a).** Voltaje de salida 0.249 v a partir de bits 0 – 0 de entrada.

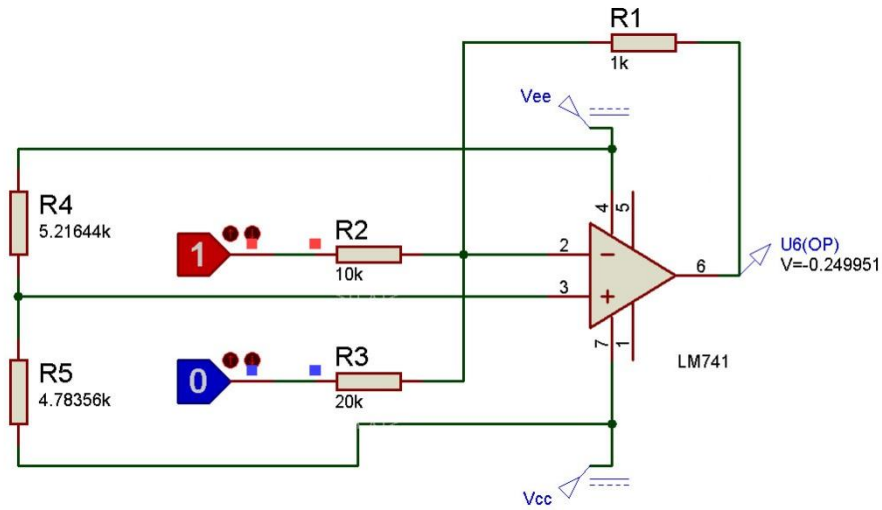


Figura A.22 (b). Voltaje de salida -0.249 V a partir de bits 1 – 0 de entrada.

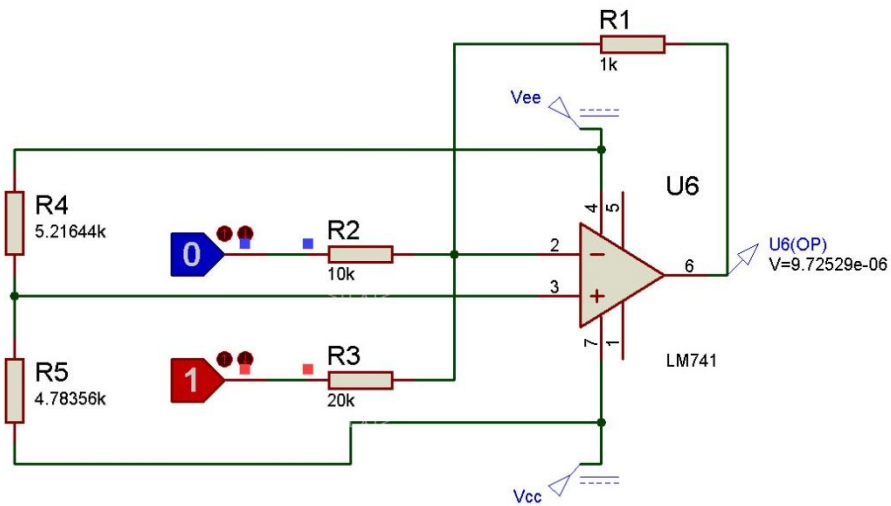
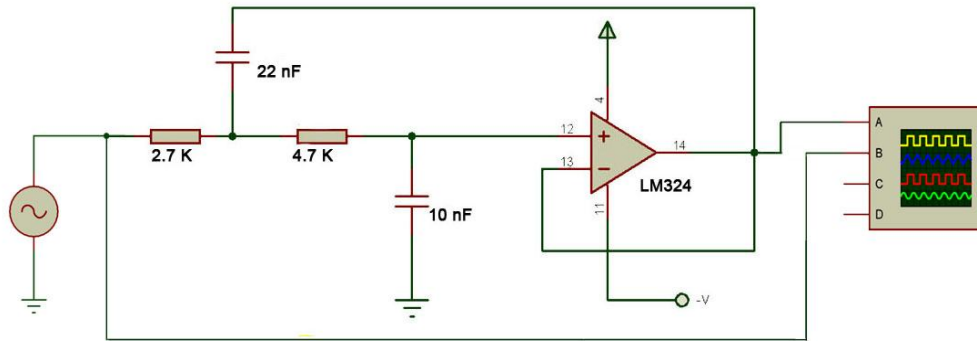


Figura A.22 (c). Voltaje de salida aproximadamente 0 V a partir de bits 0 – 1 de entrada.

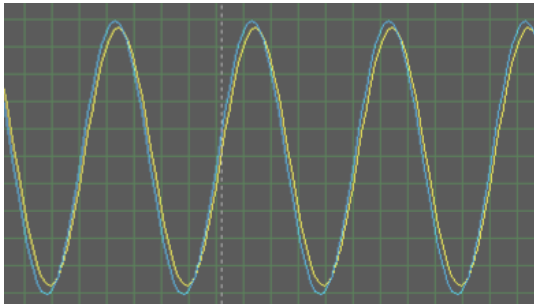
#### A.3.3.1.3.2 Filtro Limitador de Banda

Este filtro está encargado de limitar el espectro de la información enviada a un ancho de banda de 3000 Hz. Para simular esta etapa, se utilizó un generador de señales, el cual se conectó a la entrada del filtro, y un osciloscopio con el canal A conectado a la salida del filtro, y el canal B a la entrada. La figura A.23 muestra la simulación del filtro limitador de banda.

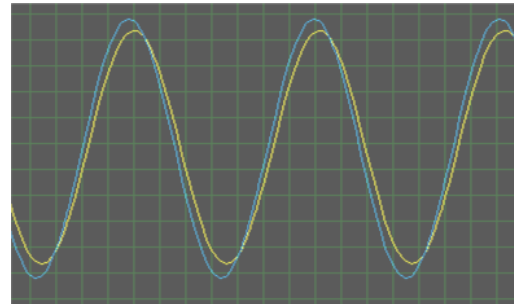


**Figura A.23.** Simulación del filtro limitador de banda.

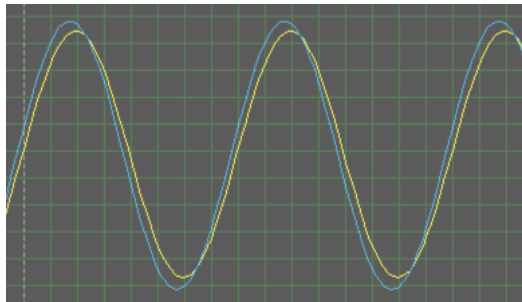
La frecuencia del generador de señales se incrementó regularmente, con el objetivo de visualizar la señal que se tenía a la salida. Para señales de entrada con frecuencias menores a 3000 Hz, se obtenía la misma señal a la salida del filtro. Sin embargo, al incrementar la frecuencia de la señal de entrada por encima de 3000 Hz, se observó a la salida del filtro una señal más atenuada, cumpliéndose el objetivo del filtro. En las figuras A.24 (a, b, c, d, e y f) se muestran la señal de entrada (curva azul) y la señal de salida (curva amarilla).



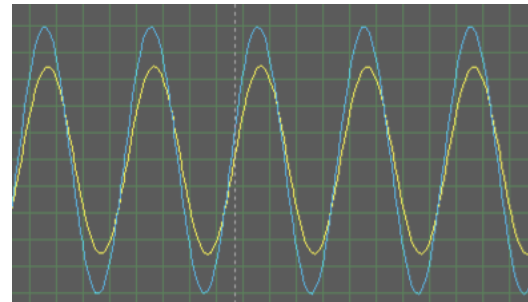
**Figura A.24 (a).** Señal de entrada (1000 Hz) vs. Señal de salida del filtro limitador de banda.



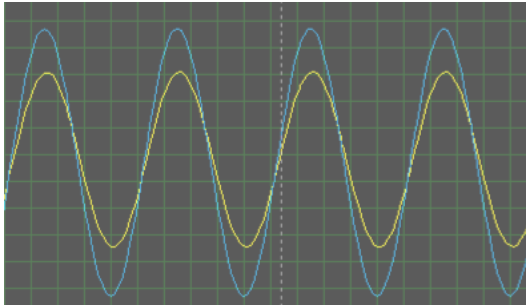
**Figura A.24 (b).** Señal de entrada (2000 Hz) vs. Señal de salida del filtro limitador de banda.



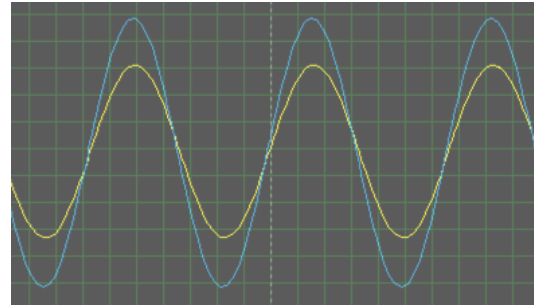
**Figura A.24 (c).** Señal de entrada (3000 Hz) vs. Señal de salida del filtro limitador de banda.



**Figura A.24 (d).** Señal de entrada (5000 Hz) vs. Señal de salida del filtro limitador de banda.



**Figura A.24 (e).** Señal de entrada (10000 Hz) vs. Señal de salida del filtro limitador de banda.

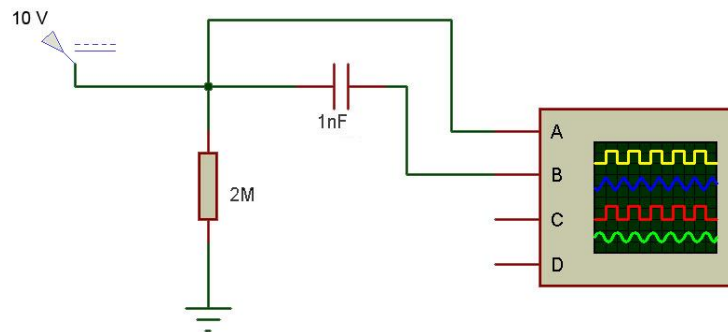


**Figura A.14 (f).** Señal de entrada (15000 Hz) vs. Señal de salida del filtro limitador de banda.

A medida que la frecuencia de la señal de entrada se incrementa, el filtro atenúa cada vez más la la señal. Con una señal de entrada de frecuencia menor o igual a 3000 Hz, la señal de salida salida presenta casi la misma amplitud que la de entrada, mientras que para los valores de frecuencia superiores a los 3000 Hz se tiene una señal de salida mas tenue respecto a la de entrada.

#### A.3.3.1.3.3 Filtro eliminador DC

Filtro pasa-alto utilizado para eliminar cualquier nivel DC que introduzca el radio. Para simular este circuito, se utilizó una señal DC de 10 V, como entrada al filtro. A la entrada y salida del filtro se conectó un osciloscopio con el fin de comparar las señales. La figura A.25 muestra la simulación del filtro eliminador DC.



**Figura A.25.** Simulación del filtro eliminador DC.

En la figura A.26 se muestra la interfaz del osciloscopio utilizado en la simulación. La curva amarilla representa la señal de entrada de 10 VDC, y la curva azul representa la señal de salida, ubicada en el centro de referencia del osciloscopio, con valor 0 VDC.

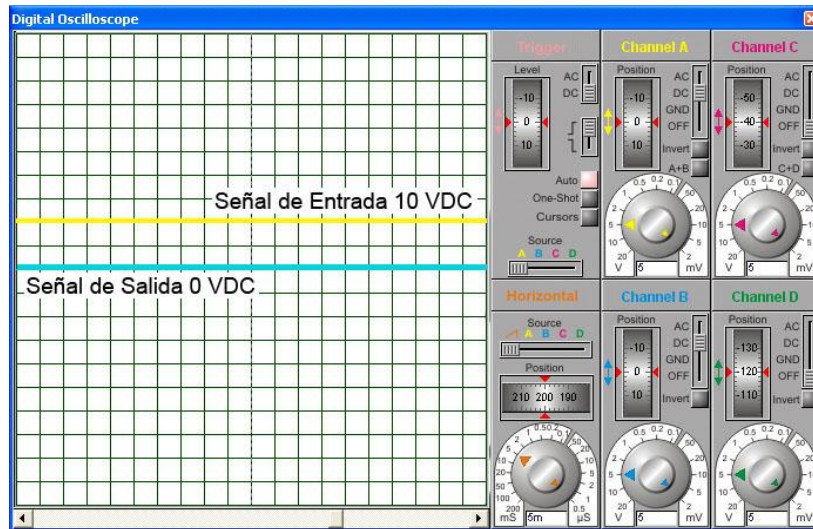


Figura A.26. Señal de entrada vs. Señal de salida filtro eliminador DC.

#### A.3.3.1.3.4 Amplificador

La etapa amplificadora se encarga de incrementar el voltaje de  $\pm 250$  mV a  $\pm 5$  V. La figura A.27 muestra la simulación del amplificador donde se visualiza un voltaje de entrada al amplificador de 250 mV y un voltaje de salida de 4.957 V.

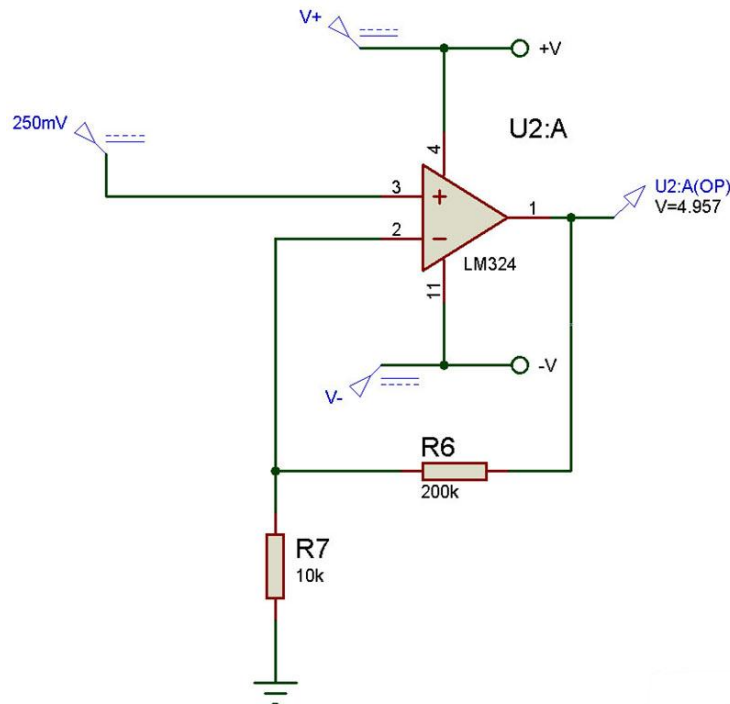


Figura A.27. Simulación del Amplificador.

### A.3.3.1.3.5 Detector de pulsos positivos y detector de pulsos negativos

Para realizar la simulación de los comparadores con histéresis encargados de realizar la detección de los pulsos positivos y negativos, se utilizó un tren de pulsos como señal de entrada, con valores pico - 5 V y 5 V. Se presenta la simulación realizada para el detector de pulsos positivos, ya que el procedimiento de detección de pulsos negativos es análogo. El objetivo de la simulación fue comprobar que a la salida del circuito se obtuvieran sólo los pulsos positivos.

En la figura A.28 se muestra la simulación del detector de pulsos positivos, donde se tiene el generador del tren de pulsos conectado a la entrada del circuito, junto con el canal A del osciloscopio. A la salida del detector se ha conectado el canal B del osciloscopio con el fin de comparar las señales de entrada y salida.

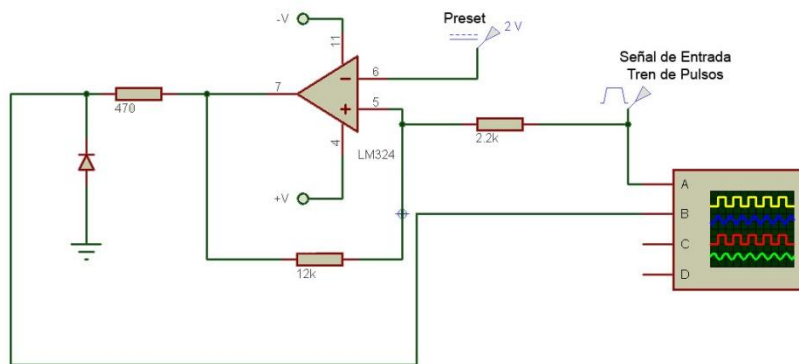


Figura A.28. Simulación del detector de pulsos positivos.

La figura A.29 muestra la señal de entrada (señal amarilla) que se tiene en el canal A del osciloscopio. El canal B donde se detecta la señal de salida se tiene en GND, para mostrar la polaridad +5 y -5 que presenta la señal de entrada en el canal A.

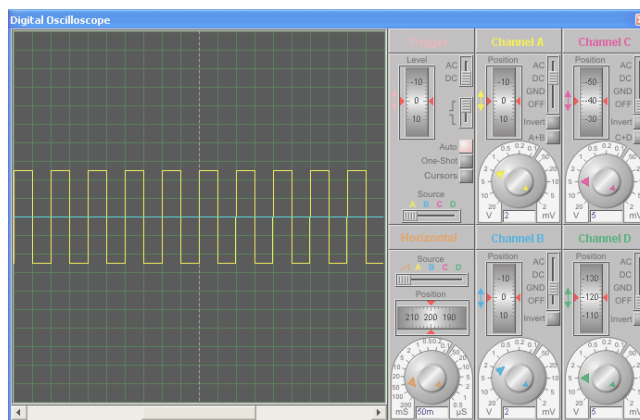
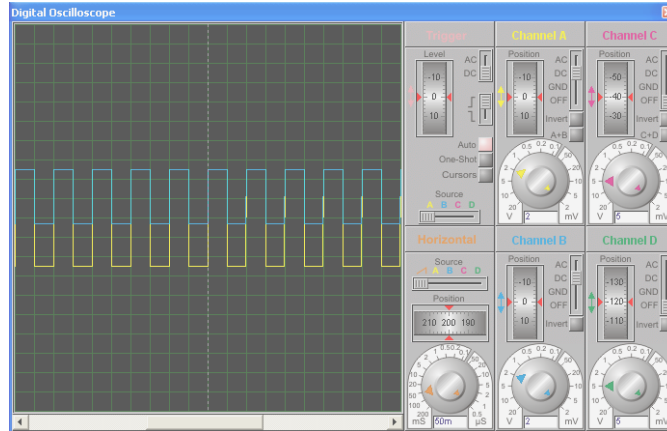


Figura A.29. Señal de entrada con valor  $\pm 5V$  (amarilla).

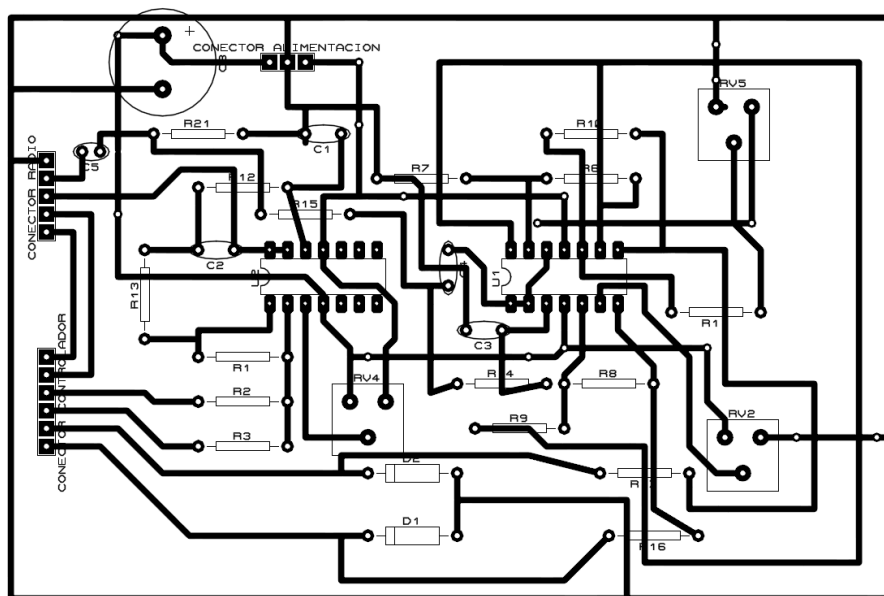
La figura A.30 muestra la señal de entrada (amarilla) y la señal de salida (azul) del detector de pulsos positivos. A la salida se obtienen pulsos de valor +5 V, lográndose la detección de únicamente pulsos positivos.



**Figura A.30.** Señal de entrada vs. señal de salida detector de pulsos positivos (Simulación).

#### A.3.3.1.3.6 Circuito Impreso de la Tarjeta de Adaptación Radio.

Los elementos de las etapas hardware de la tarjeta de adaptación radio descritas en la sección 3.1.4.2 del documento principal se interconectan en una Tarjeta Impresa (PCB, *Printed Circuit Board*). La figura A.31 muestra el circuito impreso de la tarjeta de adaptación radio.

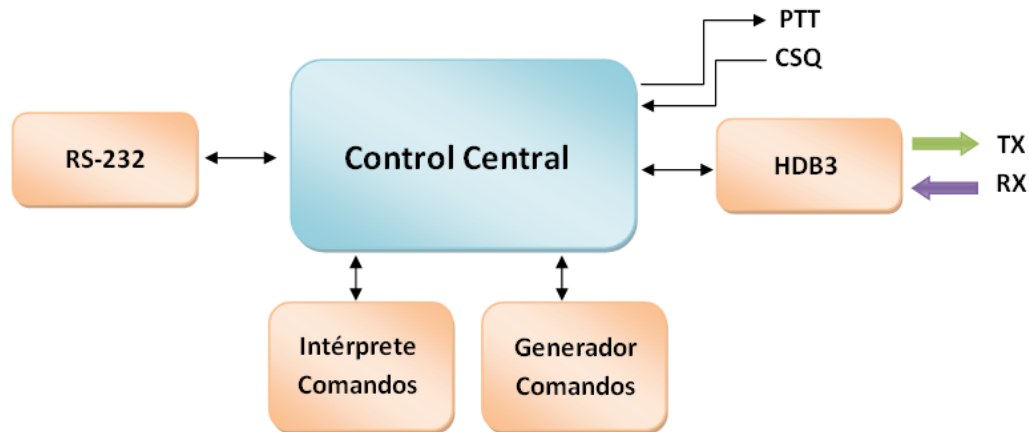


**Figura A.31.** Circuito impreso de la tarjeta de adaptación radio.

### A.3.3.2 Tarjeta de la Unidad de Control, Conmutación y Acondicionamiento de la Estación de Control (UCCA-EC)

#### A.3.3.2.1 Módulos Software de la tarjeta UCCA-EC

La tarjeta UCCA-EC es una interfaz entre el radio convencional de dos vías y los demás componentes del bloque estación de control, encargada de controlar, organizar, adaptar, recibir y enviar la información manejada por el sistema en el extremo fijo (central de despacho) del sistema. El microcontrolador es el dispositivo de control principal de la tarjeta, donde se encuentran programados los módulos software definidos para esta tarjeta. En la figura A.32 se muestran los módulos software que componen la tarjeta UCCA-EC.



**Figura A.32.** Módulos software que componen la tarjeta de la unidad de control, conmutación y acondicionamiento de la estación de control.

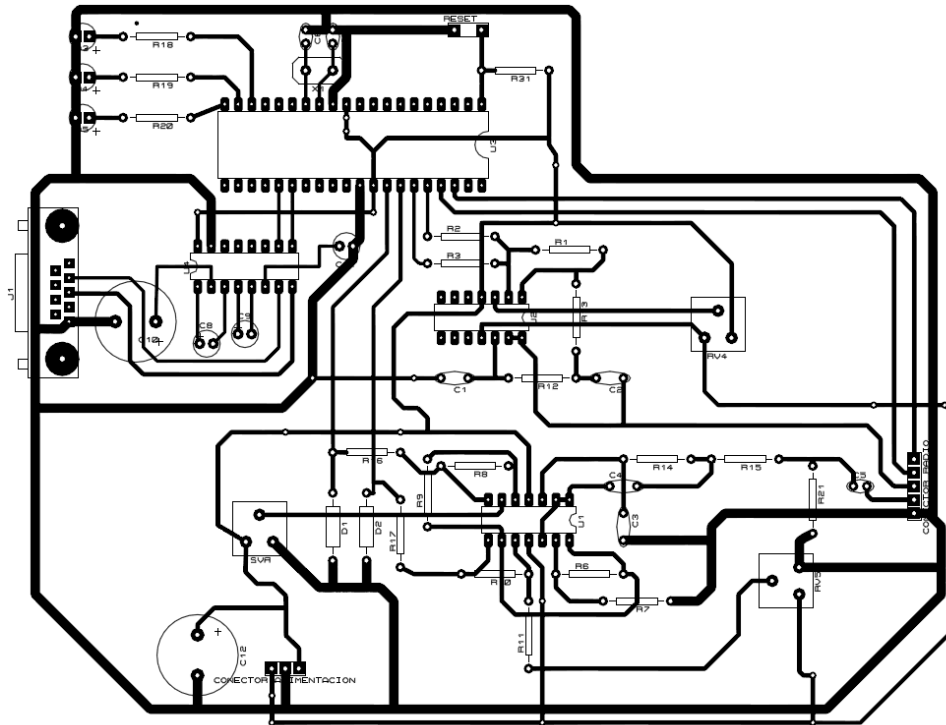
Los módulos HDB3, intérprete de comandos y generador de comandos realizan las mismas funciones descritas en la definición de los módulos software que componen la unidad móvil. Asimismo, las señales PTT y CSQ se utilizan de la misma manera.

- **Control Central:** módulo encargado de controlar y comunicar a los otros módulos software que componen la tarjeta UCCA-EC. Aquí se encuentra el flujo principal del programa, se coordinan las acciones y se utilizan las funciones implementadas en los demás módulos.
- **RS-232:** módulo encargado de realizar la comunicación entre la tarjeta y el servidor central a través del estándar RS-232. En este módulo se definen las funciones que permiten hacer uso del estándar.



### A.3.3.2 Circuito Impreso de la Tarjeta UCCA-EC

La figura A.33 muestra el circuito impreso de la tarjeta UCCA-EC donde se interconectan los elementos que componen las etapas descritas en la sección 3.3.1.4.3 del documento principal.



**Figura A.33.** Circuito impreso de la tarjeta de la unidad de control, conmutación y acondicionamiento de la estación de control.

### A.3.4 Medición de la Desviación de Frecuencia Radio Motorola PRO 3100

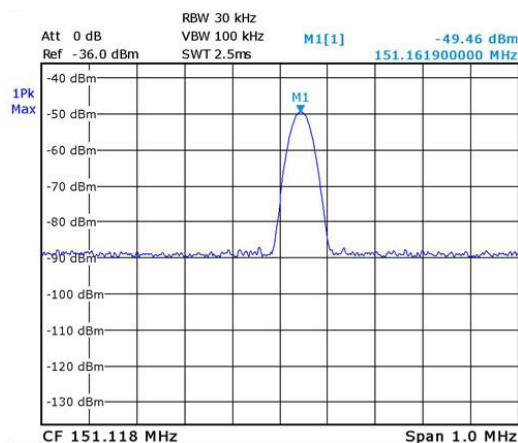
#### A.3.4.1 Medición con Señal de Entrada Binaria

Para realizar las mediciones de frecuencia de la señal modulada por el radio Motorola PRO 3100, se utilizó un Analizador de Redes ZVL marca Rohde & Schwarz. Para la prueba se conectó un generador de señales cuadradas al pin 14 del puerto de datos del radio, y se transmitió la señal, la cual fue captada por el analizador de redes. La figura A.34 presenta el esquema de conexión.

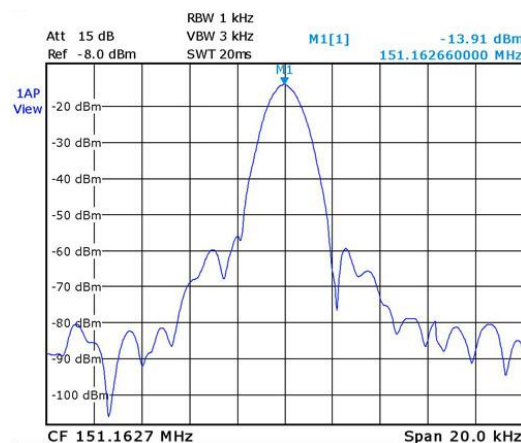


**Figura A.34.** Esquema de conexión del generador de señales, radio convencional de dos vías y analizador de redes.

Inicialmente, con el objetivo de encontrar la frecuencia de portadora, se presionó el PTT del micrófono del radio, donde se midió una frecuencia de 151.16266 MHz, la cual está dentro del rango 136 a 174 MHz especificado para el radio configurado en la banda de VHF. En las figuras A.35 (a y b) se muestra la frecuencia de la señal portadora en el dominio de la frecuencia.



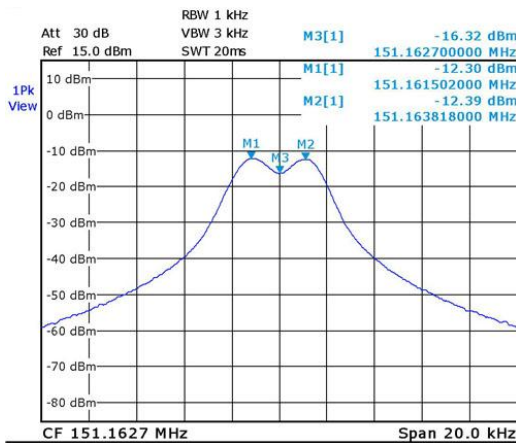
**Figura A.35 (a).** Frecuencia de la señal portadora 151.1619 MHz. (Span 1 MHz)



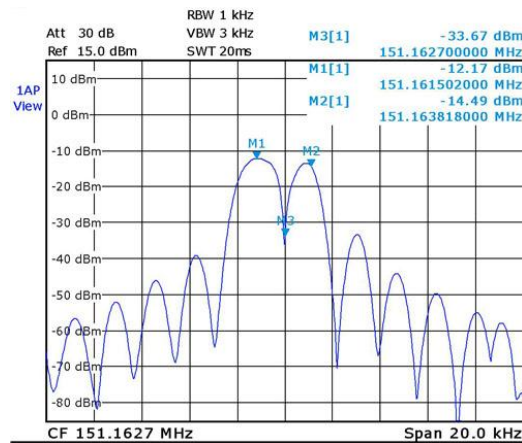
**Figura A.35 (b).** Frecuencia de la señal portadora 151.16266 MHz. (Span 20 KHz)

Posteriormente, se realizó la transmisión de la señal binaria. En las figuras A.36 (a y b) se muestra el espectro de la señal captada por el analizador de redes al fijar una señal binaria de amplitud 200 mVPP y frecuencia de 300 Hz a la entrada del radio. En cada figura se visualizan tres marcadores: M1, M2 y M3. M3 representa la frecuencia central, mientras que M1 y M2 representan el máximo corrimiento alrededor de la frecuencia central (desviación de frecuencia) generada al modular en FM. La desviación de frecuencia se determina como el valor absoluto de la diferencia entre la frecuencia

central y una de las frecuencias de marca (M1 o M3).



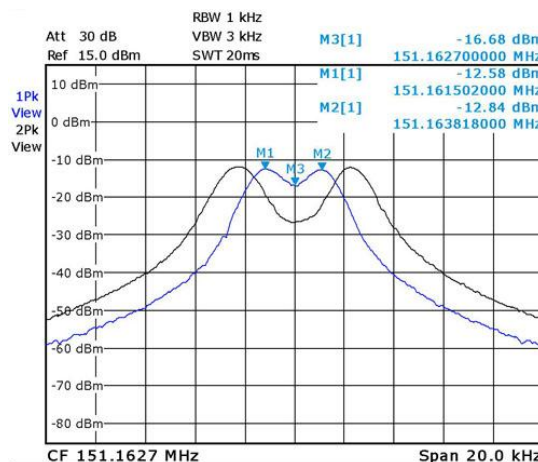
**Figura A.36 (a).** Espectro de la señal al transmitir señal binaria de 200 mVPP y frecuencia 300 Hz. (Función MAX HOLD<sup>8</sup>)



**Figura A.36 (b).** Espectro de la señal al transmitir señal binaria de 200 mVPP y frecuencia 300 Hz. (Función NORMAL<sup>9</sup>)

La desviación de frecuencia para este caso es de 1.118 KHz, medida como la diferencia entre las frecuencias de M2 y M3.

Cuando el voltaje pico a pico de la señal binaria de entrada aumenta, la desviación de frecuencia también lo hace. En la figura A.37 se muestra la diferencia de desplazamiento de frecuencia que existe entre dos señales cuando se tiene señal de entrada de 200 mVPP (curva azul) y 400 mVPP (curva negra).

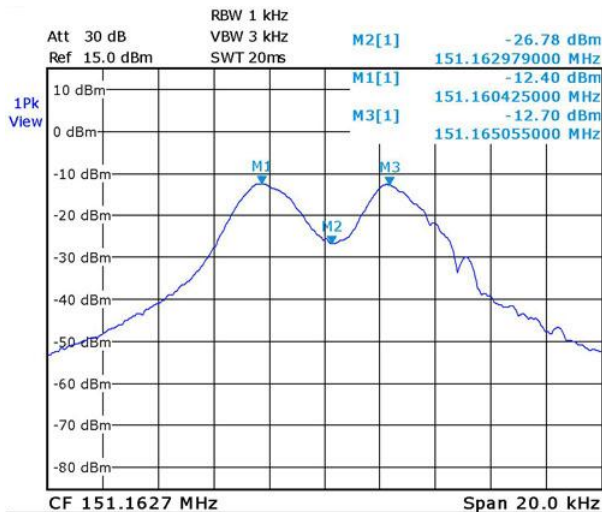


**Figura A.37.** Señales moduladas con señal de entrada 200 mVPP (curva azul) y 400 mVPP (curva negra).

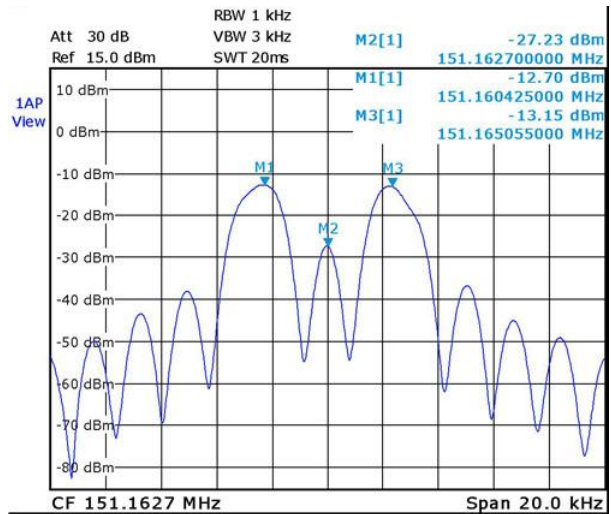
<sup>8</sup> MAX HOLD: función que retiene los máximos de la señal, diferenciándose claramente los picos de la misma y manteniéndose en un estado constante.

<sup>9</sup> NORMAL: función que muestra la señal en su estado normal, tal como el analizador la capta en tiempo real.

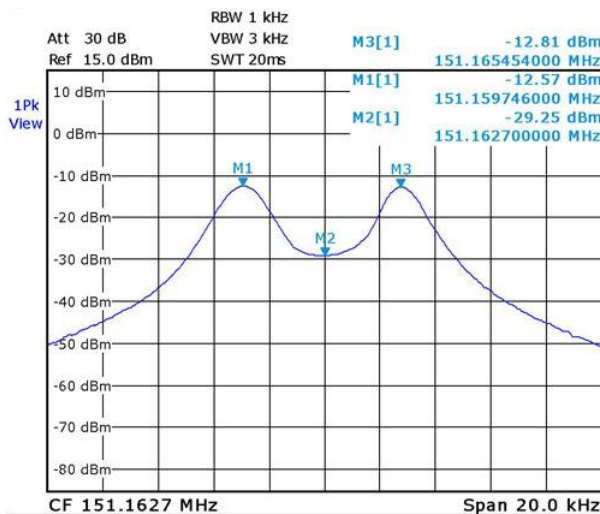
Las figuras A.38 (a, b, c, d, e, f, g, h, i, j, k y l) muestran el espectro de la señal modulada por el radio Motorola PRO 3100, cuando se aplica una señal binaria de frecuencia 300 Hz en el pin de transmisión de datos, con diferentes valores de voltaje pico a pico. En este caso el marcador M2 representa la frecuencia central, mientras que M1 y M3 representan el máximo corrimiento de la frecuencia central.



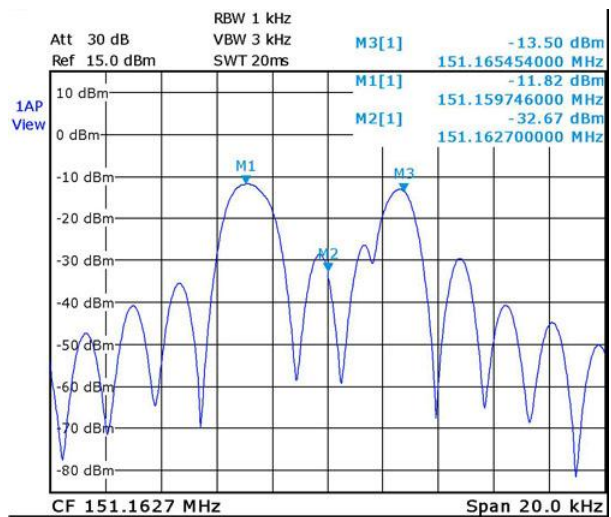
**Figura A.38 (a).** Nivel de señal de entrada: 400 mV. Desviación de frecuencia: 2.554 KHz. (MAX HOLD).



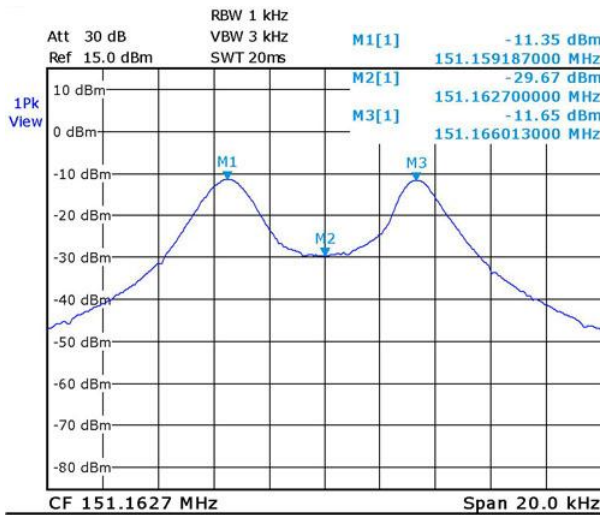
**Figura A.38 (b).** Nivel de señal de entrada: 400 mV. Desviación de frecuencia: 2.554 KHz. (NORMAL)



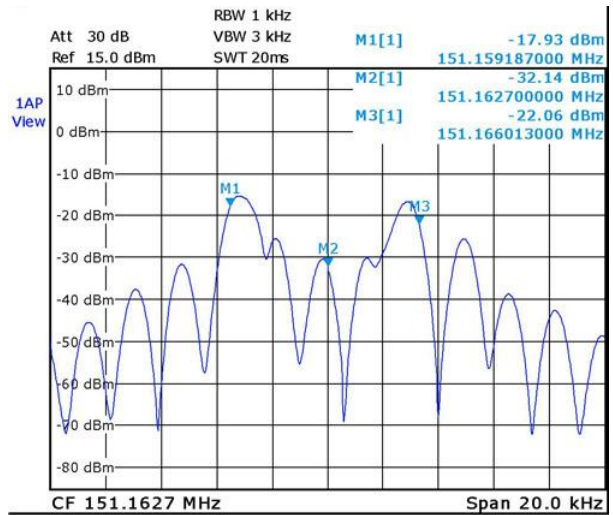
**Figura A.38 (c).** Nivel de señal de entrada: 500 mV. Desviación de frecuencia: 2.954 KHz. (MAX HOLD)



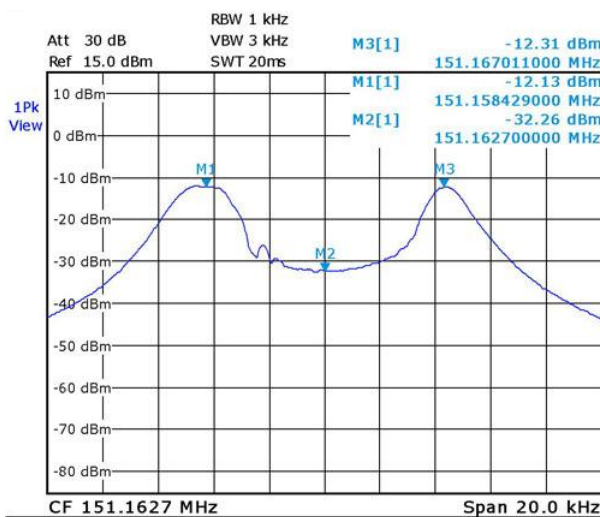
**Figura A.38 (d).** Nivel de señal de entrada: 500 mV. Desviación de frecuencia: 2.954 KHz. (NORMAL)



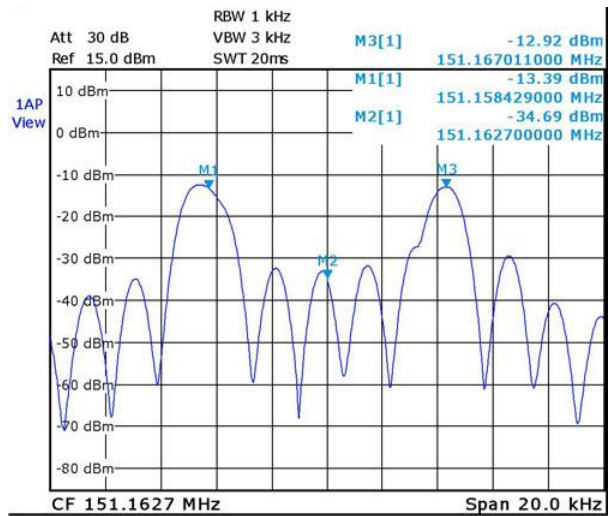
**Figura A.38 (e).** Nivel de señal de entrada: 600 mV. Desviación de frecuencia: 3.513 KHz. (MAX HOLD)



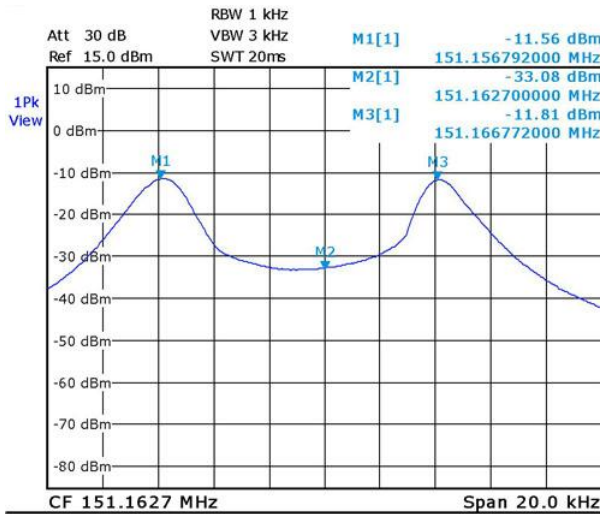
**Figura A.38 (f).** Nivel de señal de entrada: 600 mV. Desviación de frecuencia: 3.513 KHz. (NORMAL)



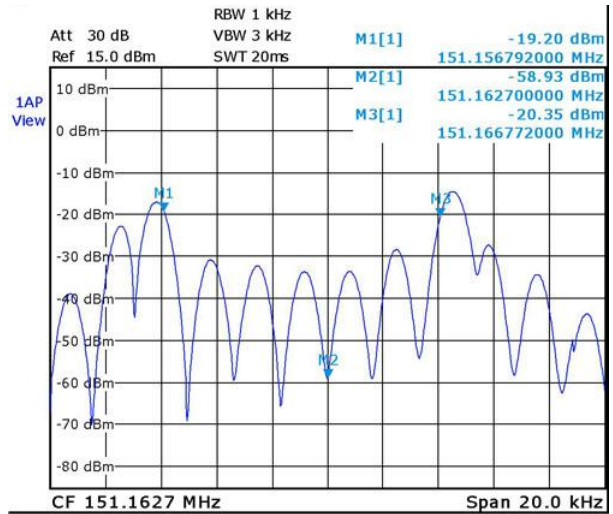
**Figura A.38 (g).** Nivel de señal de entrada: 800 mV. Desviación de frecuencia: 4.311 KHz. (MAX HOLD)



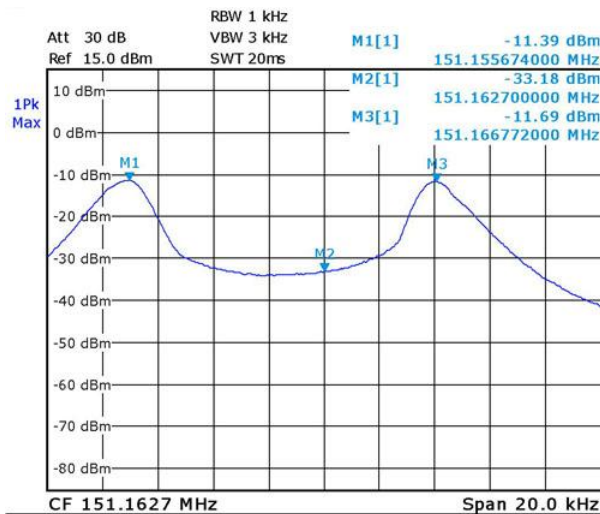
**Figura A.38 (h).** Nivel de señal de entrada: 800 mV. Desviación de frecuencia: 4.311 KHz. (NORMAL)



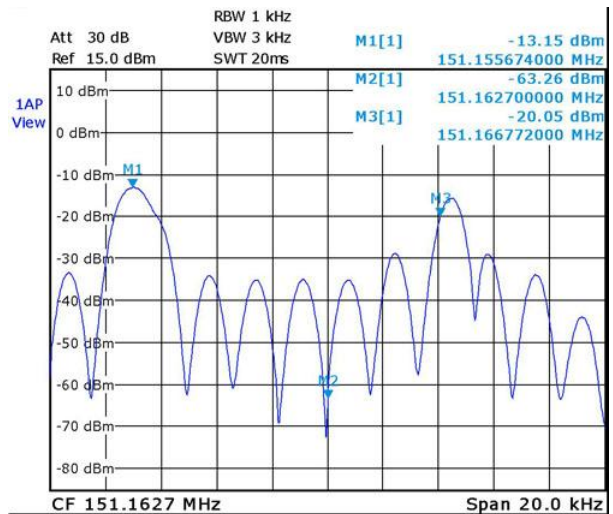
**Figura A.38 (i).** Nivel de señal de entrada: 1000 mV. Desviación de frecuencia: 5.908 KHz. (MAX HOLD)



**Figura A.38 (j).** Nivel de señal de entrada: 1000 mV. Desviación de frecuencia: 5.908 KHz. (NORMAL)



**Figura A.38 (k).** Nivel de señal de entrada: 1200 mV. Desviación de frecuencia: 7.026 KHz. (MAX HOLD)



**Figura A.38 (l).** Nivel de señal de entrada: 1200 mV. Desviación de frecuencia: 7.026 KHz. (NORMAL)

La tabla A.7 resume las diferentes desviaciones de frecuencia obtenidas para los respectivos valores de voltaje pico. La desviación de frecuencia se midió entre los marcadores que presentarían el máximo corrimiento.

Voltaje de Entrada de la señal de Datos (mVpp)	Desviación máxima de Frecuencia $\Delta f$ (KHz)
200	1.118
400	2.554
500	2.954
600	3.513
800	4.311
1000	5.908
1200	7.026

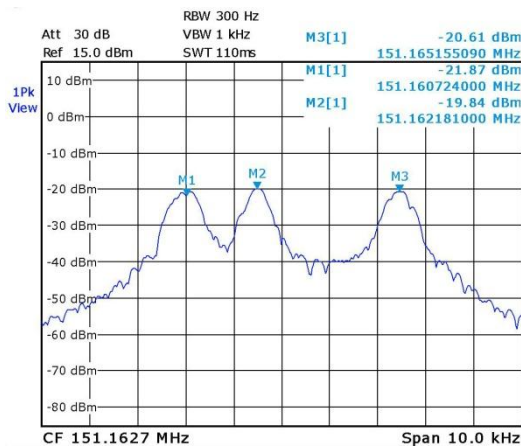
**Tabla A.7.** Desviación de frecuencia para diferentes voltajes pico a pico de la de señal de datos de entrada.

Para HORUS se definió una señal de 500 mVPP como entrada al radio Motorola PRO3100. Según la tabla A.7, la desviación de frecuencia correspondiente a este valor de voltaje es de 2.954 KHz. Sin embargo este valor se encontró al fijar como entrada al radio una señal cuadrada producida con el generador de funciones.

Para encontrar el valor de desviación de frecuencia correspondiente a la señal de entrada HDB3 usada en HORUS, se realizó la prueba que se explica a continuación.

#### A.3.4.2 Medición de la desviación de frecuencia con señal de entrada HDB3

En esta prueba se fijó la señal con codificación HDB3, generada por la tarjeta de adaptación radio, como entrada al pin 14 del radio Motorola PRO3100, por lo que en el analizador de redes se observaron tres picos, correspondientes a los tres niveles de la codificación. Esta señal tiene un valor de 500 mVPP. Inicialmente se transmitió una secuencia de ceros continuos a una velocidad de 333 bps. En la figura A.39 se presenta la señal captada por el analizador de redes.



**Figura A.39.** Medición 1. Señal modulada que presenta los estados HDB3 al transmitir secuencia de ceros consecutivos con una velocidad de 333 bps y 500 mVPP.

Se definieron tres marcadores, donde M1 indica la frecuencia modulada correspondiente a la marca negativa (-250 mV), M2 la del espacio (0V) y M3 la de la marca positiva (+250 mV). La tabla A.8 muestra estos valores.

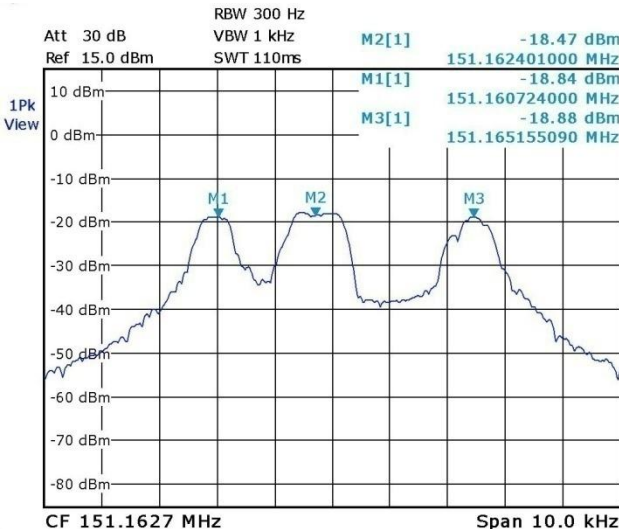
Estado HDB3	Frecuencia (MHz)
Marca negativa (-250 mV)	151.160724
Espacio (0V)	151.162181
Marca positiva (+250 mV)	151.16515509

**Tabla A.8.** Frecuencias correspondientes a estados HDB3.

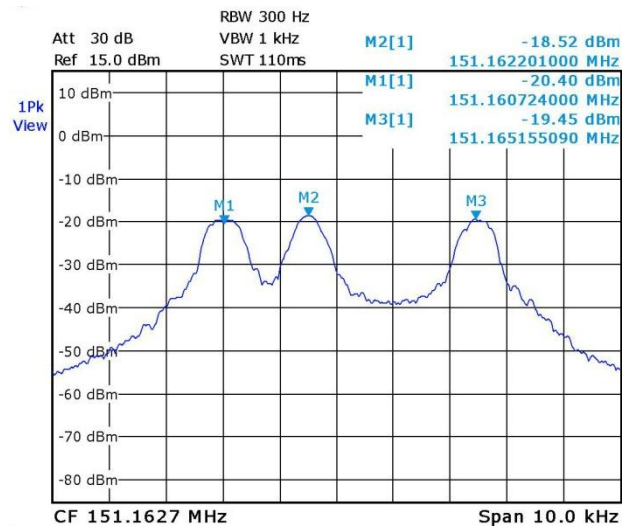
Se observa que los valores de las frecuencias de marca no son simétricos respecto a la frecuencia central de 151.1627 MHz. Asimismo, la frecuencia modulada para el espacio, no es exactamente la misma frecuencia central, sino que existe una desviación de 0.519 KHz. Esto se debe a que el circuito generador de los niveles HDB3 no es de alta precisión y realmente no se generan valores de voltaje exactos. Aun con una señal HDB3 de entrada al radio, se genera un pequeño nivel DC, que afecta la señal desplazando su espectro.

Se determinó una desviación de frecuencia de 2.97409 KHz como la diferencia entre la posición del marcador M3 y M2. La diferencia entre los marcadores M2 y M1 produce como resultado una desviación de frecuencia de 1.457 KHz.

Se realizaron tres mediciones más, en las figuras A.40 (a, b y c) se muestran los marcadores M1, M2 y M3 correspondientes a las frecuencias de la marca negativa, espacio y marca positiva, respectivamente.

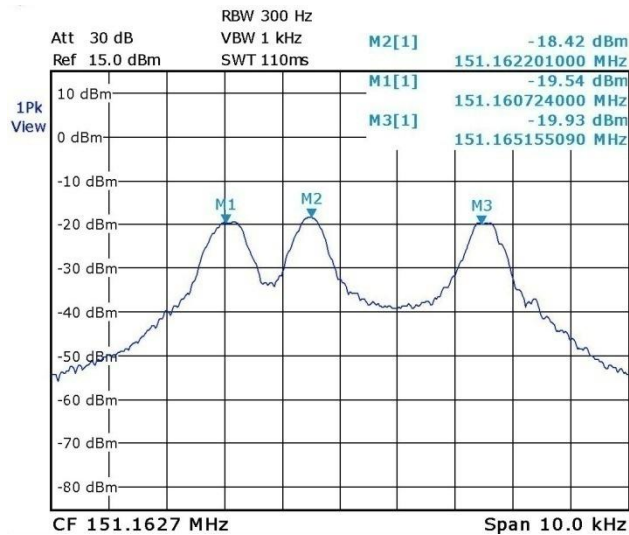


**Figura A.40 (a).** Medición 2.



**Figura A.40 (b).** Medición 3.





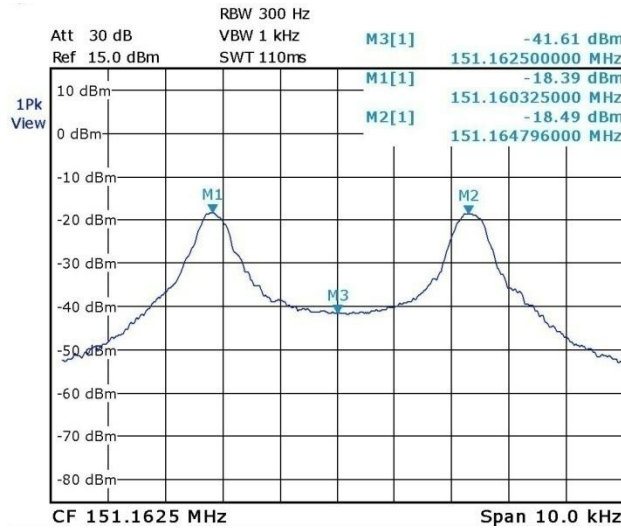
**Figura A.40 (c).** Medición 4.

La tabla A.9 muestra el resumen de la desviación de frecuencia hallada para las medidas realizadas utilizando la secuencia de ceros consecutivos. La desviación de frecuencia se calculó hallando la diferencia entre los marcadores que presentaron la máxima separación (M3 y M2).

Número de medición	Desviación Máxima de Frecuencia (KHz)
Medición 1	2.97409
Medición 2	2.75409
Medición 3	2.95409
Medición 4	2.95409

**Tabla A.9.** Resumen de desviación de frecuencia con secuencia de ceros como señal de entrada.

De igual forma, se realizó la transmisión de una secuencia de unos consecutivos, a una velocidad de 333 bps, para visualizar si los valores de frecuencia modulados para cada estado HDB3 son iguales a los obtenidos al enviar la secuencia de ceros. La figura A.41 muestra la el espectro de la señal y en la tabla A.10 se resumen los valores de frecuencia obtenidos para cada estado HDB3.



**Figura A.41.** Espectro de la señal presenta los estados HDB3 al transmitir secuencia de unos consecutivos con una velocidad de 333 bps y 500 mVPP.

Estado HDB3	Frecuencia (MHz)
Marca negativa (-250 mV)	151.160325
Espacio (0V)	151.1625
Marca positiva (+250 mV)	151.164796

**Tabla A.10.** Frecuencias correspondientes a estados HDB3.

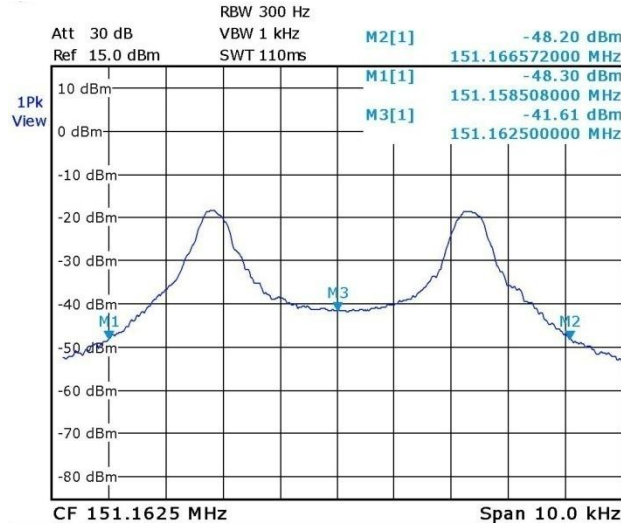
Comparando los valores de las tablas A.8 y A.10 se ve que las frecuencias de las marcas y espacio son muy similares, lo que quiere decir que la desviación de frecuencia no varía significativamente al en enviar una secuencia consecutiva de ceros o unos al codificarlos en HDB3.

Debido a que el estado ESPACIO no es utilizado la transmisión de unos consecutivos, en este caso no se presenta un máximo en el marcador central M3, el cual indica una frecuencia central de 151.1625 MHz. Comparando este valor con la frecuencia de espacio de la tabla A.10 se determina una diferencia de 0.319 KHz, lo que evidencia una mínima desviación de la frecuencia central al enviar una secuencia de ceros o unos.

La desviación máxima de frecuencia para el caso de transmisión de la secuencia de unos consecutivos es de 2.296 KHz, calculada como la diferencia entre los marcadores M2 y M3.

Con el objetivo de hallar el ancho de banda práctico necesario para transmitir la señal HDB3, se tomaron los valores de frecuencia medidos para las marcas en el caso de la transmisión de unos consecutivos. Los marcadores M1 y M2 correspondientes a las marcas de HDB3, se movieron a lo largo de la señal, hasta localizarse a 30 dB por

debajo de los valores máximos de la señal<sup>10</sup>. La figura A.42 muestra los valores de frecuencia medidos.



**Figura A.42.** Valores de frecuencia medidos al ubicar marcadores M1 y M2 a 30 dB por debajo de los máximos de la señal.

El ancho de banda práctico se calculó al medir la distancia entre M2 y M1, lo cual se define en la ecuación B.17.

$$B_{\text{Práctico}} = M2 - M1 \quad (\text{B.17})$$

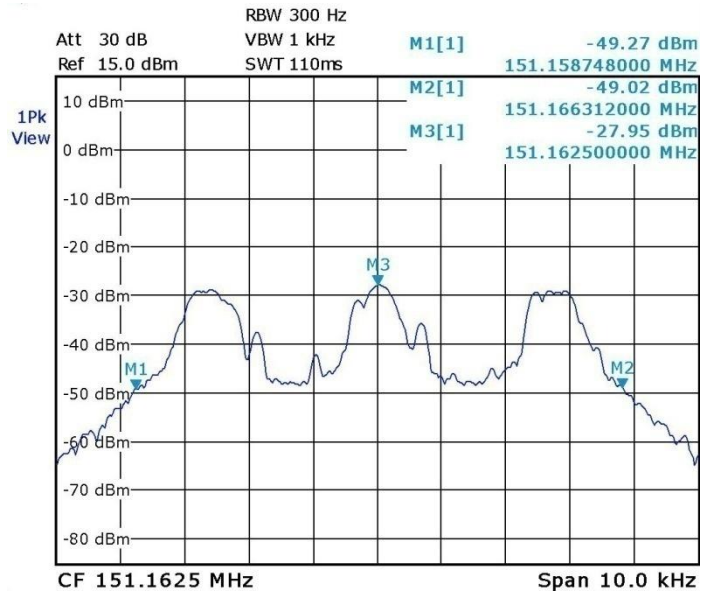
Reemplazando los valores de M2 y M1 en la ecuación B.17 se tiene:

$$B_{\text{Práctico}} = 151.166572 \text{ MHz} - 151.158508 \text{ MHz}$$

$$B_{\text{Práctico}} = 8.064 \text{ KHz}$$

Por lo tanto, el ancho de banda ocupado para transmitir la señal HDB3 es de 8.064 KHz, quedando un ancho de banda de 4.436 KHz para los canales de guarda. Asimismo, se calculó el ancho de banda de manera práctica, moviendo los marcadores M1 y M2 hasta ubicarlos en los puntos donde se tuviera una diferencia de 20 dB respecto a los máximos. En este caso se usaron las mediciones obtenidas al transmitir la secuencia de ceros consecutivos. La figura A.43 muestra los valores de las frecuencias en dichos puntos.

<sup>10</sup> Considerando que para valores superiores a 30 dB la señal presenta una atenuación muy grande, por lo que la potencia de las componentes de frecuencia no es muy grande y no son tenidas en cuenta para el cálculo del ancho de banda.



**Figura A.43.** Valores de frecuencia medidos al ubicar marcadores M1 y M2 a 20 dB por debajo de los máximos de la señal.

A través de la ecuación B.17 se calcula el ancho de banda para este caso.

$$B_{\text{Práctico}} = 151.166312 \text{ MHz} - 151.158748 \text{ MHz}$$
$$B_{\text{Práctico}} = 7.564 \text{ KHz}$$

Por lo tanto, el ancho de banda ocupado para transmitir la señal HDB3 es de 7.564 KHz, quedando un ancho de banda de 4.936 KHz para los canales de guarda.

En el documento principal se encontró el ancho de banda de manera teórica, con un resultado de 11.94818 KHz. Comparando este valor con los valores prácticos obtenidos se observan diferencias de 3.88418 KHz, para el caso del ancho de banda encontrado utilizando el concepto de 30 dB por debajo de los máximos de la señal, y 4.38418 KHz para el caso del ancho de banda encontrado utilizando el concepto de 20 dB por debajo de los máximos de la señal.

### **A.3.5 Descripción Detallada de las Interfaces Implementadas en la Arquitectura de HORUS para el Despliegue de Servicios**

A pesar que en el capítulo tres del documento principal se realiza una descripción general de las interfaces implementadas en la arquitectura de HORUS para el despliegue de servicios, es necesario explicar de manera detallada las diferentes tecnologías y herramientas que conforman las interfaces y que son de gran relevancia para su correcto funcionamiento.

### A.3.5.1 Navegador Web

Los navegadores web (*browsers*), son programas que permiten ver la información que contiene una página web, generalmente código escrito en Lenguaje de Marcación de Hipertexto (HTML<sup>11</sup>, *HyperText Markup Language*). El navegador interpreta el código, en el que está escrita la página web y lo presenta en pantalla permitiendo al usuario interactuar con su contenido y navegar hacia otros lugares de la red mediante enlaces o hipervínculos (4).

La funcionalidad básica de un navegador web es permitir la visualización de documentos de texto, posiblemente con recursos multimedia incrustados. Los documentos pueden estar ubicados en el computador donde está el usuario o en internet, pero también pueden estar en cualquier otro dispositivo que esté conectado al computador del usuario y que dispone de las capacidades necesarias para la transmisión de los documentos.

### A.3.5.2 Servlets

Un *servlet* es un programa que se ejecuta en un servidor o contenedor JEE<sup>12</sup>, especialmente diseñado para ofrecer contenido dinámico, generalmente HTML, desde un servidor web y a partir de los parámetros de la solicitud que se realice a través del navegador web.

La API usada para escribir *servlets*, no incluye ninguna información acerca de cómo son cargados, ni el ambiente en el cual se ejecutan, ni el protocolo usado para transmitir los datos del usuario. Lo anterior permite a los *servlets* ser implementados por diferentes servidores Web (5).

El principal componente de la API *servlet* es la interfaz *HttpServlet*. Todos los *servlets* implementan esta interfaz directamente a través de la clase donde son definidos. Esta interfaz está provista de métodos que manipulan a los *servlets* y la comunicación con sus clientes. Cuando un *servlet* es llamado desde un cliente, este recibe dos objetos: *ServletRequest* y *ServletResponse*. La interfaz *ServletRequest* se encarga de la comunicación desde el cliente al servidor, mientras que la interfaz *ServletResponse* atiende la comunicación desde el *servlet* al cliente.

La interfaz *ServletRequest* permite al *servlet* acceder a información como, los nombres

---

<sup>11</sup> El Lenguaje de Marcación de Hipertexto, es el lenguaje de marcación predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

<sup>12</sup> *Java Platform, Enterprise Edition* o Java EE, es una plataforma de programación para desarrollar y ejecutar aplicaciones software en lenguaje de programación Java con arquitectura distribuida de N niveles, basándose ampliamente en componentes software modulares que se ejecutan sobre un servidor de aplicaciones.

de los parámetros pasados por el cliente, el protocolo usado por el cliente, los nombres de los *host* remotos que hacen la solicitud y el servidor que la recibe. Esta interfaz permite a los *servlets* el acceso a métodos que permiten manejar la presentación de la respuesta como salida en el navegador, a través de los cuales se consiguen los datos desde el cliente que usa protocolos como HTTP POST<sup>13</sup> (5).

La interfaz *ServletResponse* proporciona al *servlet* los métodos para contestarle al cliente. Permite al *servlet* configurar la forma de salida de los datos para el cliente como *ServletOutputStream* que permite enviar la réplica de datos como respuesta. Las clases e interfaces descritas conforman a un *servlet* básico. Pero existen métodos adicionales que provee la API con la capacidad para controlar sesiones o múltiples conexiones, entre muchas más aplicaciones, por ejemplo:

- Procesamiento de datos enviados con el Protocolo de Transferencia de Hipertexto (HTTP<sup>14</sup>, *Hypertext Transfer Protocol*) a partir de una forma HTML: Un *servlet* puede ser parte de una orden de entrada al sistema de procesamiento de datos, realizando la respectiva actualización en la base de datos.
- Interacción múltiple entre personas: Un *servlet* puede manipular y sincronizar múltiples solicitudes al mismo tiempo.
- Redireccionamiento de peticiones: Los *servlets* pueden reenviar las peticiones a otros servidores y *servlets*. Esto permite que al ser usados se balancee la carga entre varios servidores que atiendan la misma tarea, según el tipo de proceso que deban atender.

### A.3.5.3 Java Server Pages

Las JSP Son una tecnología orientada a crear páginas web con programación en Java<sup>15</sup>, que permite generar contenido dinámico, en forma de documentos HTML, XML<sup>16</sup> o de otro tipo. Con JSP es posible crear aplicaciones web que se ejecuten en diferentes servidores web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma (6). Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar *scripts* de servidor en sintaxis Java.

---

<sup>13</sup> Este protocolo, al igual que el protocolo HTTP GET codifica los parámetros de solicitud como pares nombre y valor de la solicitud HTTP.

<sup>14</sup> HTTP es el protocolo usado en cada transacción de la web. HTTP define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web (clientes y servidores) para comunicarse.

<sup>15</sup> Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90.

<sup>16</sup> El Lenguajes Extensible de marcas (XML, *Extensible Markup Language*), es un metalenguaje extensible de etiquetas desarrollado por el W3C (*World Wide Web Consortium*), considerado como una manera de definir y describir otros lenguajes de programación.

El motor de las páginas JSP está basado en servlets. Los *servlets* generan archivos con extensión “.jsp” que incluyen, dentro de la estructura de etiquetas HTML, las sentencias Java a ejecutar en el servidor. Antes de que sean funcionales los archivos, el motor JSP lleva a cabo una fase de traducción de esa página en un servlet, implementado en un archivo “.class” (código byte de Java). Esta fase de traducción se lleva a cabo eventualmente cuando se recibe la primera solicitud de la página JSP, aunque existe la opción de precompilar el código para evitar ese tiempo de espera la primera vez que un cliente solicita la página.

El desempeño de una página JSP es el mismo que tendría el servidor equivalente, ya que el código es compilado como cualquier otra clase Java. A su vez, la máquina virtual compilará dinámicamente a código de máquina las partes de la aplicación que lo requieran durante el proceso de interpretación del archivo por un navegador web (7). Esto hace que JSP tenga un buen desempeño y sea más eficiente que otras tecnologías web que ejecutan el código de una manera puramente interpretada.

#### **A.3.5.4 Java Script**

*Java Script* es un lenguaje de programación que no requiere de compilación, los programas desarrollados en este lenguaje son ejecutados por el navegador. *Java Script* no es orientado a objetos, esto quiere decir que se puede programar sin necesidad de crear clases, tal como se realiza en los lenguajes de programación estructurada (8).

Entre las acciones típicas que se pueden realizar en *Java Script* se tienen dos vertientes. Por un lado los efectos especiales sobre páginas web, para crear contenidos dinámicos y elementos de la página con movimiento, cambios de color o cualquier otro dinamismo. Por el otro, *Java Script* permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que se puede crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo (9).

*Java Script* ha tenido influencia de múltiples lenguajes y se diseñó con una sintaxis similar al lenguaje de programación Java, aunque más fácil de utilizar para personas que no programan (9). Todos los navegadores modernos interpretan el código *Java Script* integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje *Java Script* de una implementación del Modelo de Objetos para la Representación de Documentos (DOM, *Document Object Model*).

#### **A.3.5.5 Google Maps JavaScript API V3.**

El JavaScript de Google Maps API le permite integrar Google Maps en páginas web propietarias. La versión 3 de esta API es especialmente diseñada para ser más rápida

y más aplicable a dispositivos móviles, así como las tradicionales aplicaciones web.

El API proporciona una serie de utilidades para la manipulación de mapas y añadir contenido al mapa a través de una variedad de servicios, lo que le permite crear aplicaciones robustas en cualquier sitio web. Es un servicio gratuito, disponible para cualquier sitio web que es gratuito para los consumidores.

La dirección URL <http://maps.google.com/maps/api/js> señala la ubicación de un archivo *JavaScript* que carga todos los símbolos y las definiciones necesarias para el uso de la API de Google Maps V3. La página debe contener una secuencia de comandos de etiquetas que apunten a esta dirección (10).

Para mostrar el mapa en una página web, se debe reservar un lugar para él. Generalmente, esto se hace creando un elemento `<div>` y obteniendo una referencia a este elemento según el modelo de objetos del documento (DOM). A continuación se realiza una descripción de las clases y métodos de la API de Google Maps que se implementaron durante la etapa de desarrollo de la aplicación web para HORUS.

#### A.3.5.5.1 Clase LatLng

Un objeto de la clase `LatLng` representa un punto en el mapa a partir de sus coordenadas geográficas, latitud y longitud.

- **Constructor:** La implementación de este método se realiza de la siguiente manera: `LatLng(lat:number,lng:number,noWrap?:boolean)`. El valor de la latitud debe estar entre -90 grados y 90 grados, y la longitud se debe ajustar entre -180 grados y 180 grados.
- **Métodos:** La tabla A.11 muestra la descripción e implementación para los métodos de clase `LatLng`.

MÉTODO	VALOR DE RETORNO	DESCRIPCIÓN
<code>equals(other:LatLng)</code>	booleano	Compara objetos tipo <code>LatLng</code>
<code>lat()</code>	numérico	Retorna la latitud en grados
<code>lng()</code>	numérico	Retorna la longitud en grados
<code>toString()</code>	string	Convierte un tipo de dato a string

Tabla A.11. Métodos de la clase `LatLng` implementados en Horus.

#### A.3.5.5.2 Clase MapTypeId.

Las constantes de esta clase permiten identificar los diferentes tipos de mapa soportados por la tercera versión de la API de Google Maps. La tabla A.12 muestra las



posibles constantes y una breve descripción.

CONSTANTE	DESCRIPCIÓN
HYBRID	Este tipo de mapa muestra una capa transparente de las calles principales sobre imágenes de satélite.
ROADMAP	Este tipo de mapa muestra un plano normal de las calles principales.
SATELLITE	Este tipo de mapa muestra imágenes de satélite.
TERRAIN	Este tipo de mapa muestra imágenes con características físicas tales como el terreno y la vegetación.

Tabla A.12. Constantes de la clase MapTypeI implementadas en Horus.

### A.3.5.5.3 Clase Map.

- **Constructor:** Este método permite crear un mapa en el interior del contenedor HTML determinado, a través de una etiqueta <div>. Su implementación se define como Map(mapDiv:Node,opts?:MapOptions), donde las opciones del mapa hacen referencia a sus propiedades.
- **Métodos:** En la tabla A.13 se mencionan y describen los métodos implementados en la aplicación web de HORUS a través de la clase Map.

MÉTODO	VALOR DE RETORNO	DESCRIPCIÓN
getCenter()	LatLng	Retorna un objeto de tipo LatLng con las coordenadas del centro del mapa.
getDiv()	Nodo	Retorna el contenedor HTML donde se despliega el mapa.
getMapTypeI()	MapTypeI	Retorna el tipo de mapa.
getZoom()	Numérico	Retorna un número entre 0 y 20 indicando el nivel de zoom del mapa.
setCenter (latlng:LatLng)	Ninguno	Permite fijar las coordenadas del centro del mapa, recibe como parámetro un objeto de la clase LatLng.
setMapTypeI (mapTypeI:MapTypeI)	Ninguno	Permite fijar el tipo de mapa.
setOptions (opciones:MapOptions)	Ninguno	Permite fijar múltiples opciones y características del mapa.
setZoom (zoom: número)	Ninguno	Permite fijar el nivel de zoom del mapa.

Tabla A.13. Métodos de la clase Map implementados en Horus.

- **Eventos:** En la tabla A.14 se mencionan y describen los eventos implementados en la aplicación web de HORUS a través de la clase *Map*.

EVENTO	DESCRIPCIÓN
<b>center_changed</b>	Este evento se desencadena cuando cambia el centro del mapa.
<b>click</b>	Este evento se desencadena cuando el usuario hace clic en el mapa (pero no al hacer clic en un marcador o InfoWindow).
<b>dblclick</b>	Este evento se desencadena cuando el usuario hace doble clic en el mapa.
<b>drag</b>	Este evento permanece activo mientras el usuario arrastra el mapa.
<b>dragend</b>	Este evento se desencadena cuando el usuario deja de arrastrar el mapa.
<b>dragstart</b>	Este evento se desencadena cuando el usuario comienza a arrastrar el mapa.
<b>idle</b>	Este evento se desencadena cuando el mapa queda inactivo después de la toma panorámica o zoom.
<b>maptypeid_changed</b>	Este evento se desencadena cuando cambia la propiedad <code>mapTypeId</code> .
<b>mousemove</b>	Este evento se desencadena cuando el usuario mueve el ratón sobre el contenedor del mapa.
<b>mouseout</b>	Este evento se desencadena cuando el cursor del ratón sale del contenedor del mapa.
<b>mouseover</b>	Este evento se desencadena cuando el ratón del usuario entra en el contenedor del mapa.
<b>zoom_changed</b>	Este evento se desencadena cuando el zoom del mapa los cambios de propiedad.

Tabla A.14. Eventos de la clase *Map* implementados en Horus.

- **Propiedades:** En la tabla A.15 se mencionan y describen las propiedades de la clase *Map*. Estas propiedades se definen en el momento en que se inicializa el mapa.

PROPIEDAD	TIPO	DESCRIPCIÓN
<b>backgroundColor</b>	String	Color utilizado para el fondo <div>, este color se ve cuando el mapa no ha sido cargado completamente. Esta opción sólo se establece cuando el mapa se ha inicializado.
<b>center</b>	LatLng	El primer centro del mapa.
<b>disableDefaultUI</b>	Booleano	Activa o desactiva toda la interfaz de usuario predeterminada. Se puede reemplazar individualmente.
<b>disableDoubleClickZoom</b>	Booleano	Activa o desactiva el zoom y el centro con doble click. Habilitado de forma predeterminada.
<b>draggable</b>	Booleano	Si es false, evita que el mapa de ser arrastrado. Esta opción está activada por defecto.

<b>draggableCursor</b>	String	El nombre o la url del cursor para mostrar en un objeto que pueda ser arrastrado.
<b>draggingCursor</b>	String	El nombre o la url del cursor que se muestra cuando un objeto es arrastrado.
<b>keyboardShortcuts</b>	Booleano	Si es false, evita que el mapa sea controlado por el teclado. Los atajos del teclado están habilitados por defecto.
<b>mapTypeControl</b>	Booleano	Si es false, deshabilita el control para cambiar el de tipo de mapa. Esta habilitado por defecto.
<b>navigationControl</b>	Booleano	Si es false, deshabilita el control de navegación. Esta habilitado por defecto.
<b>scaleControl</b>	Booleano	Si es false, deshabilita el control de la escala del mapa. Esta habilitado por defecto.
<b>scrollwheel</b>	Booleano	Si es falso, desactiva la perilla de desplazamiento de zoom del mapa. La perilla de desplazamiento está activada por defecto.
<b>zoom</b>	Numérico	El nivel inicial de zoom requerido para el mapa.

Tabla A.15. Propiedades de la clase *Map* implementadas en Horus.

#### A.3.5.5.4 Clase Marker.

- **Constructor:** Crea un marcador con las opciones especificadas, el cual se añadirá al mapa durante la construcción, es necesario tener en cuenta, que la posición del marcador debe ser establecida previamente. La implementación del constructor es de la siguiente manera: `Marker (opta?:MarkerOptions)`, donde las opciones hacen referencia sus propiedades.
- **Métodos:** En la tabla A.16 se mencionan y describen los métodos implementados en la aplicación web de HORUS a través de la clase *Marker*.

MÉTODO	VALOR DE RETORNO	DESCRIPCIÓN
<b>getIcon ()</b>	String	Retorna una cadena de caracteres especificando el nombre de la imagen utilizada como icono para señalar una coordenada en el mapa.
<b>getMap ()</b>	Map	Retorna un objeto de la clase Map.
<b>getPosition()</b>	LatLng	Retorna las coordenadas donde se encuentra ubicado el marcador.
<b>getShadow(</b>	String	Retorna una cadena de caracteres con el nombre de la imagen utilizada para darle al icono el efecto de sombra.
<b>getTitle()</b>	String	Retorna una cadena de caracteres con el nombre asignado al marcador.

<b>getVisible()</b>	Booleano	Retorna un tipo de dato booleano indicando el estado de visibilidad del marcador.
<b>setIcon(icon:string MarkerImage)</b>	Ninguno	Fija un icono el mapa para señalar una coordenada específica.
<b>setMap(map:Map StreetViewPanorama)</b>	Ninguno	Representa el marcador especificado en el mapa. Si el mapa se establece en null, el marcador se eliminará.
<b>setOptions (options:MarkerOptions)</b>	Ninguno	Fija las propiedades del marcador.
<b>setPosition (latlng:LatLng)</b>	Ninguno	Fija las coordenadas del marcador a través de un objeto de la clase LatLng.
<b>setShadow (shadow:string MarkerImage)</b>	Ninguno	Fija la imagen designada para dar efecto de sombra al icono del marcador.
<b>setVisible (visible: Boolean)</b>	Ninguno	Si es false, se oculta el marcador. El marcador esta visible por defecto.

**Tabla A.16.** Métodos de la clase *Marker* implementados en Horus.

- **Eventos:** En la tabla A.17 se mencionan y describen los eventos implementados en la aplicación web de HORUS a través de la clase *Marker*.

EVENTO	DESCRIPCIÓN
<b>click</b>	Este evento se desencadena cuando se hace click sobre el icono del marcador.
<b>dblclick</b>	Este evento se desencadena cuando se hace doble click sobre el icono del marcador.
<b>drag</b>	Este evento se activa repetidamente mientras el usuario arrastra el marcador.
<b>dragend</b>	Este evento se desencadena cuando el usuario deja de arrastrar el marcador.
<b>dragstart</b>	Este evento se desencadena cuando el usuario comienza a arrastrar el marcador.
<b>icon_changed</b>	Este evento se desencadena cuando se cambia el icono de marcador en sus propiedades.
<b>mouseout</b>	Este evento se desencadena cuando el ratón sale de la zona del icono de marcador.
<b>mouseover</b>	Este evento se desencadena cuando el ratón entra el área del icono de marcador.
<b>position_changed</b>	Este evento se desencadena cuando cambia la posición del marcador en sus propiedades.
<b>title_changed</b>	Este evento se desencadena cuando cambia el título del marcador en sus propiedades.
<b>visible_changed</b>	Este evento se desencadena cuando se producen cambios en la visibilidad del marcador.

**Tabla A.17.** Eventos de la clase *Marker* implementados en Horus.

- **Propiedades:** En la tabla A.18 se mencionan y describen las propiedades de la clase *Marker*.

PROPIEDAD	TIPO	DESCRIPCIÓN
<b>clickable</b>	Booleano	Si es true, el marcador reacciona ante eventos del ratón.
<b>cursor</b>	String	El cursor del ratón.
<b>draggable</b>	Booleano	Si es true, el marcador se puede arrastrar. El valor por defecto es false.
<b>flat</b>	Booleano	Si es true, la sombra marcador no se mostrará.
<b>icon</b>	String	Icono para el primer plano.
<b>map</b>	Map	Mapa en el que se mostrará marcador.
<b>position</b>	LatLng	Posición del marcador.
<b>shadow</b>	String	Sombras de la imagen.
<b>title</b>	cadena	Texto sobre el marcador.
<b>visible</b>	booleano	Si es true, el marcador es visible.

**Tabla A.18.** Propiedades de la clase *Marker* implementadas en Horus.

#### A.3.5.5.5 Clase Polyline.

Una polilínea es una superposición lineal de segmentos de línea conectados en el mapa.

- **Constructor:** Crea una polilínea de acuerdo a las opciones especificadas, la ruta de acceso, el estilo de la traza (color y tamaño) y las posiciones; las cuales se definen mediante una serie de objetos *LatLng* o mediante un vector de coordenadas *LatLng*. La implementación del constructor es de la siguiente manera: `Polyline(opts?:PolylineOptions)`, donde las opciones hacen referencia a sus propiedades.
- **Métodos:** En la tabla A.19 se mencionan y describen los métodos implementados en la aplicación web de HORUS a través de la clase *Polyline*.

MÉTODO	VALOR DE RETORNO	DESCRIPCIÓN
<b>getMap ()</b>	Map	Retorna el mapa en el cual se adjunta la polilínea.
<b>getPath ()</b>	Array<LatLng>	Recupera la traza de la polilínea.
<b>setMap (Map:Map)</b>	Ninguno	Representa la polilínea en el mapa especificado. Si el mapa se establece en null, la polilínea será removida.
<b>setOptions (options:PolylineOptions)</b>	Ninguno	Permite fijar las opciones o propiedades de la polilínea.
<b>setPath(path:Array&lt;LatLng&gt;)</b>	Ninguno	Permite establecer la ruta del a traza.

**Tabla A.19.** Métodos de la clase *Polyline* implementados en Horus.

- **Eventos:** En la tabla A.20 se mencionan y describen los eventos implementados en la aplicación web de HORUS a través de la clase *Polyline*.

EVENTO	DESCRIPCIÓN
<b>click</b>	Este evento se desencadena cuando se hace click sobre una polilínea.
<b>dblclick</b>	Este evento se desencadena cuando se hace doble click sobre una polilínea.
<b>mousedown</b>	Este evento se desencadena cuando el ratón se mueve por debajo de una polilínea.
<b>mouseover</b>	Este evento se desencadena cuando el ratón se mueve sobre una polilínea.
<b>mouseup</b>	Este evento se desencadena cuando el ratón se mueve por encima de una polilínea.
<b>rightclick</b>	Este evento se desencadena al hacer clic derecho sobre una polilínea.

**Tabla A.20.** Eventos de la clase *Polyline* implementados en Horus.

- **Propiedades:** En la tabla A.21 se mencionan y describen las propiedades de la clase *Polyline*.

PROPIEDAD	TIPO	DESCRIPCIÓN
<b>clickable</b>	Booleano	Indica si la polilínea soporta eventos del ratón asociados al clic. El valor predeterminado es true.
<b>map</b>	Map	Mapa en el que se mostrará la polilínea.
<b>path</b>	Array<LatLng>	La secuencia ordenada de coordenadas de la polilínea. Esta ruta se puede especificar utilizando una matriz simple de <i>LatLngs</i> , o un vector de <i>LatLngs</i> . Es posible insertar o extraer <i>LatLngs</i> en el vector de coordenadas y este actualizará automáticamente la polilínea en el mapa.
<b>strokeColor</b>	String	El color del trazo según el formato tradicional de HTML, es decir. "# FFAA00".
<b>strokeOpacity</b>	Numérico	La opacidad de la línea; valor entre 0.0 y 1.0.
<b>strokeWeight</b>	Numérico	El ancho del trazo en píxeles.

**Tabla A.21.** Propiedades de la clase *Polyline* implementadas en Horus.

#### A.3.5.5.6 Clase InfoWindow.

Una *infowindow* o ventana de información, es una superposición que contiene información y se ve como una burbuja. Frecuentemente está asociado a un marcador.

- **Constructor:** Crea una ventana de información con las características establecidas. Una *infowindow* se puede colocar en un mapa en una posición particular o por encima de un marcador, con base en los parámetros especificados en las opciones. Cuando se activa una *infowindow*, el mapa se

mueve para hacerse visible cuando se abre. Después de construir una *infowindow*, el usuario puede hacer clic en el botón de cierre para quitarlo del mapa, o el desarrollador puede llamar al método “close()” para generar el mismo efecto. La implementación del constructor es de la siguiente manera: `InfoWindow(opta?:InfoWindowOptions)`, donde las opciones hacen referencia a sus propiedades.

- **Métodos:** En la tabla A.22 se mencionan y describen los métodos implementados en la aplicación web de HORUS a través de la clase *InfoWindow*.

MÉTODO	VALOR DE RETORNO	DESCRIPCIÓN
<b>close ()</b>	Ninguno	Cierra la ventana de información.
<b>getContenido ()</b>	String Nodo	Retorna la información contenida en la ventana de información, puede ser una cadena de caracteres o texto en formato HTML.
<b>getPosition ()</b>	LatLng	Retorna las coordenadas donde se despliega la ventana de información.
<b>open(map:Map, anchor?:MVCObject)</b>	Ninguno	Abre la ventana de información en el mapa. Opcionalmente, una ventana de información se puede asociar a un marcador o a cualquier objeto que implemente la propiedad “position”.
<b>setContent(content:string Node)</b>	Ninguno	Permite fijar el contenido de la ventana de información, puede ser una cadena de caracteres o texto en formato HTML.
<b>setOptions (options:InfoWindowOptions)</b>	Ninguno	Permite fijar las propiedades de la ventana de información.
<b>setPosition(position:LatLng)</b>	Ninguno	Permite fijar la posición de la ventana de información.

**Tabla A.22.** Métodos de la clase *InfoWindow* implementados en Horus.

- **Eventos:** En la tabla A.23 se mencionan y describen los eventos implementados en la aplicación web de HORUS a través de la clase *InfoWindow*.

EVENTO	DESCRIPCIÓN
<b>closeclick</b>	Este evento se dispara cuando se hace clic sobre el botón que cierra la ventana de información.
<b>content_changed</b>	Este evento se dispara cuando las propiedades del contenido de la ventana de información cambian.
<b>position_changed</b>	Este evento se dispara cuando las propiedades de la posición cambian.

**Tabla A.23.** Eventos de la clase *InfoWindow* implementados en Horus.

- **Propiedades:** En la tabla A.24 se mencionan y describen las propiedades de la clase *InfoWindow*.

PROPIEDAD	TIPO	DESCRIPCIÓN
<b>content</b>	String Node	Contenido que se mostrará en la ventana de información. Puede ser un elemento HTML, una cadena de texto, o una cadena que contiene HTML. El tamaño de la ventana de información depende del contenido.
<b>disableAutoPan</b>	Booleano	Si es true, la ventana de información se despliega automáticamente cuando se carga el mapa. Si la variable se fija como false, se desactiva esta propiedad.
<b>maxWidth</b>	Numérico	Permite establecer el ancho máximo de la ventana de información, independientemente del ancho del contenido.
<b>position</b>	LatLng	Hace referencia a la coordenada donde se desplegará la ventana de información.

**Tabla A.24.** Propiedades de la clase *InfoWindow* implementados en Horus.

### A.3.5.6 DaoGen

Como su nombre lo indica, el generador de código de objetos para acceso a datos (DaoGen) es un programa diseñado para facilitar la programación de bases de datos. DaoGen genera el código fuente completo de trabajo extensible para clases de acuerdo con el patrón de diseño para Objetos de Acceso a Datos (DAO, *Data Access Object*). Actualmente DaoGen genera código fuente para los lenguajes Java y PHP<sup>17</sup> que implementan bases de datos en el Lenguaje de Consulta Estructurado (SQL<sup>18</sup>, *Structured Query Language*) genéricas a partir de la información básica relacionada con la estructura de la base de datos que el cliente proporciona (nombre y número de filas que componen las tablas y llaves primarias) (11).

El código fuente obtenido a partir de DaoGen interactúa perfectamente con la base de datos, la idea es evitar que el usuario tenga que escribir la funcionalidad básica para cada tabla de la base de datos que necesite su aplicación. DaoGen puede ser utilizado directamente desde un navegador web y no requiere de ningún instalador.

### A.3.5.7 JDBC

La API de Java para conexión de bases de datos (JDBC) es el estándar de la industria para la conectividad entre el lenguaje de programación Java y una amplia gama de

<sup>17</sup> PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor.

<sup>18</sup> SQL es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas.



bases de datos basadas en SQL (12).

Esta API se presenta como una colección de interfaces Java y métodos de gestión de controladores de conexión hacia cada modelo específico de base de datos. Un controlador de conexiones hacia un modelo de base de datos en particular es un conjunto de clases que implementan las interfaces Java y que utilizan los métodos de registro para declarar los tipos de Localizadores de Recursos Uniformes (URL<sup>19</sup>, *Uniform Resource Locator*) que pueden manejar (13). Para utilizar una base de datos particular, el usuario ejecuta su programa junto con la librería de conexión apropiada al modelo de su base de datos, y accede a ella estableciendo una conexión, para ello provee el localizador a la base de datos y los parámetros de conexión específicos. A partir de allí puede realizar cualquier tipo de tarea con la base de datos a la cual tenga acceso: consulta, actualización, creación, modificación, borrado de tablas y ejecución de procedimientos almacenados en la base de datos.

#### **A.3.5.8 GiovyNet Driver**

Es una solución Java de código abierto protegida por la Licencia Pública General para Librerías de GNU (GNU LGPL<sup>20</sup>, *GNU Lesser General Public License*) que se utiliza para el control de puertos seriales RS-232. GiovyNet Driver trabaja bajo sistemas operativos Linux o Windows y está compuesto por tres archivos: un archivo ".jar", un archivo ".dll" y un archivo ".so"; los dos últimos archivos contienen la lógica para manipular el puerto serie de acuerdo con el sistema operativo y el archivo ".jar" identifica el sistema operativo, si es Linux, utiliza las funciones de archivo ".so", pero si es Windows, usa las funciones del archivo ".dll" (14).

#### **A.3.5.9 RS-232.**

RS-232 es una interfaz que designa una norma para el intercambio serie de datos binarios entre un Equipo Terminal de Datos (DTE, *Data Terminal Equipment*) y un Equipo de Comunicación de Datos (DCE, *Data Communication Equipment*) (15).

La interfaz RS-232 está diseñada para distancias cortas, de hasta 15 metros según la norma, y para velocidades de comunicación bajas, de no más de 20 Kbps. A pesar de ello, muchas veces se utiliza a mayores velocidades con un resultado aceptable. La interfaz trabaja en comunicación asíncrona o síncrona y tipos de canal *simplex*, *half duplex* o *full duplex*. En un canal *simplex* los datos siempre viajarán en una dirección,

---

<sup>19</sup> Una URL podría concebirse como la extensión del nombre completo de un archivo. Las URLs posibilitan el direccionamiento de personas, ficheros y de una gran variedad de información, disponible según los distintos protocolos o servicios de internet.

<sup>20</sup> La Licencia Pública General para Librerías de GNU es una licencia de software creada por la Free Software Foundation que brinda a los usuarios la libertad de compartir y modificar el software.

por ejemplo desde DCE a DTE. En un canal *half duplex*, los datos viajan en una u otra dirección, pero sólo durante un determinado periodo de tiempo; luego el canal debe ser conmutada antes que los datos viajan en la otra dirección. En un canal *full duplex*, los datos pueden viajar en ambos sentidos simultáneamente. La tabla A.25 muestra las características de las normas definidas para comunicación serial.

CARACTERISTICAS	NORMA			
	RS-232	RS-423	RS-422	RS-485
<b>Modo</b>	Simple	Simple	Diferencial	Diferencial
<b>Numero de Transmisores</b>	1	1	1	32
<b>Numero de Receptores</b>	1	10	10	32
<b>Longitud Máxima</b>	15 m	1200 m	1200 m	1200 m
<b>Velocidad Máxima</b>	20 Kbps	100 Kbps	10 Mbps	10 Mbps
<b>Salida del Transmisor</b>	$\pm 5V_{min}$	$\pm 3.6V_{min}$	$\pm 2V_{min}$	$\pm 1.5V_{min}$
	$\pm 15V_{max}$	$\pm 6V_{max}$	---	---
<b>Carga al Transmisor</b>	3K $\Omega$ a 7K $\Omega$	450 $\Omega$ min	100 $\Omega$ min	60 $\Omega$ min
<b>R de entrada al receptor</b>	3K $\Omega$ a 7K $\Omega$	4K $\Omega$ min	4K $\Omega$ min	12K $\Omega$ min
<b>Sensibilidad del receptor</b>	$\pm 3V$	$\pm 200mV$	$\pm 200mV$	$\pm 200mV$

**Tabla A.25.** Características de las normas definidas para comunicación serial.

#### A.3.5.10 Servicios Web.

La W3C<sup>21</sup> (*World Wide Web Consortium*) define Servicio Web como un conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la web, mediante la adopción de estándares abiertos. Estas aplicaciones o tecnologías intercambian datos entre sí con el objetivo de ofrecer servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la web (16).

Estos servicios proporcionan mecanismos de comunicación estándar entre diferentes aplicaciones, que interactúan entre sí para presentar información dinámica al usuario. Los servicios web permiten la comunicación entre diferentes máquinas, con diferentes plataformas y entre distintos programas (17).

#### A.3.5.11 MidMaps

MidMaps es una pequeña librería que permite integrar Google Maps en las

<sup>21</sup> El World Wide Web Consortium, abreviado W3C, es un consorcio internacional que produce recomendaciones para la World Wide Web.

aplicaciones Java ME<sup>22</sup> (18) mediante la API de mapas estáticos de Google (Google Statics Maps API), la cual retorna una imagen (GIF<sup>23</sup>, PNG<sup>24</sup> o JPEG<sup>25</sup>) como respuesta a una solicitud HTTP estándar realizada a través de una dirección URL en la que se especifican los parámetros y atributos del mapa (19).

Para cada solicitud es necesario especificar la ubicación del mapa, el tamaño de la imagen, el nivel de acercamiento, el tipo de mapa y la ubicación de marcadores opcionales en lugares determinados del mapa. Además, es posible etiquetar los marcadores con caracteres alfanuméricos, de modo que sea posible hacer referencia a ellos. A continuación se realiza una descripción de las clases y métodos de la librería MIDMaps que se implementaron durante la etapa de desarrollo de la aplicación móvil para HORUS.

#### A.3.5.11.1 Clase *GoogleMapsCoordinates*

- **Constructor:** Recibe como parámetros la longitud y la latitud en decimales asociadas a la coordenada. Su implementación está definida de la siguiente manera:

```
GoogleMapsCoordinates(double lat, double lng)
```

#### A.3.5.11.2 Clase *GoogleMapsMarker*

- **Constructor:** Recibe como parámetro la coordenada en el mapa donde se ubicara el marcador a través de un objeto de la clase *GoogleMapsMarker*. Su implementación está definida de la siguiente manera:

```
GoogleMapsMarker(GoogleMapsCoordinates coordinates).
```

- **Métodos:** En la tabla A.26 se mencionan y describen los métodos implementados en la aplicación móvil para HORUS a través de la clase *GoogleMapsMarker*.

<sup>22</sup> Java ME o Plataforma Java, Edición Micro (Java Platform, Micro Edition), es una colección de API's de Java para el desarrollo de software para dispositivos de recursos limitados, como los Asistentes Digitales Personales (PDA, Personal Digital Assistant), teléfonos móviles y otros.

<sup>23</sup> GIF es un formato gráfico utilizado ampliamente en la Web, tanto para imágenes como para animaciones.

<sup>24</sup> PNG (*Portable Network Graphics*) es un formato gráfico basado en un algoritmo de compresión sin pérdida para *bitmaps* no sujeto a patentes.

<sup>25</sup> JPEG además de ser un método de compresión, es considerado como un formato de archivo. JPEG es el formato de imagen más común utilizado por las cámaras fotográficas digitales y otros dispositivos de captura de imagen.

MÉTODO	DESCRIPCIÓN
<b>setColor(java.lang.String color)</b>	Permite fijar un color al marcador.
<b>setLabel(char label)</b>	Permite fijar un tamaño al marcador.
<b>setSize(java.lang.String size)</b>	Permite fijar un caracter como identificador al marcador.

**Tabla A.26.** Métodos de la clase *GoogleMapsMarker* implementados en Horus.

- **Atributos:** En la tabla A.27 se mencionan y describen los atributos de la clase *GoogleMapsMarker* implementados en la aplicación móvil para HORUS.

ATRIBUTO	TIPO	DESCRIPCIÓN
<b>NO_LABEL</b>	static char	Variable que indica marcador sin identificador.
<b>SIZE_MID</b>	static java.lang.String	Variable que indica tamaño mediano para el marcador.
<b>SIZE_SMALL</b>	static java.lang.String	Variable que indica tamaño pequeño para el marcador.
<b>SIZE_TINY</b>	static java.lang.String	Variable que indica marcador delgado.

**Tabla A.27.** Atributos de la clase *GoogleMapsMarker* implementados en Horus.

#### A.3.5.11.3 Clase *GoogleMapsPath*.

- **Constructor:** No recibe parámetros, un objeto de esta clase hace referencia a un conjunto de segmentos de línea unidos y superpuestos en el mapa. Su implementación está definida de la siguiente manera: *GoogleMapsPath()*.
- **Métodos:** En la tabla A.28 se mencionan y describen los métodos implementados en la aplicación móvil para HORUS a través de la clase *GoogleMapsPath*.

MÉTODO	DESCRIPCIÓN
<b>addPoint(GoogleMapsCoordinates coordinates)</b>	Permite agregar un punto al vector de coordenadas con base en el cual se trazan los segmentos de línea.
<b>setColor(java.lang.String color)</b>	Permite fijar un color al segmento de línea.
<b>setFillColor(java.lang.String color)</b>	Permite fijar un color de relleno al segmento de línea
<b>setWeight(int weight)</b>	Permite fijar un grosor a la línea trazada.

**Tabla A.28.** Métodos de la clase *GoogleMapsPath* implementados en Horus.

#### A.3.5.11.4 Clase GoogleStataicMap.

- **Constructor:** Esta clase hereda sus atributos de la clase GoogleMaps cuyo constructor se define de la siguiente manera: GoogleMaps(java.lang.String apiKey), donde el parámetro *apiKey* representa la clave que Google Maps brinda para implementar la API de *Google Static Maps*.
- **Métodos:** En la tabla A.29 se mencionan y describen los métodos implementados en la aplicación móvil para HORUS a través de la clase *GoogleStaticMap*.

MÉTODO	DESCRIPCIÓN
<b>addMarker(GoogleMapsMarker marker)</b>	Permite agregar al mapa un marcador previamente creado.
<b>addPath(GoogleMapsPath path)</b>	Permite agregar al mapa un segmento de línea o polilínea previamente creada.
<b>getImage()</b>	Permite enviar la petición HTTP al servidor de Google Maps para obtener la imagen del mapa con las propiedades previamente definidas.
<b>move(int direction)</b>	Con base en el parámetro ingresado, este método redefine la petición HTTP para solicitar una nueva imagen con un desplazamiento en una determinada dirección.
<b>removeMarker(GoogleMapsMarker marker)</b>	Permite remover un marcador de la imagen de mapa.
<b>removePath(GoogleMapsPath path)</b>	Permite remover un segmento de línea o polilínea de la imagen de mapa.
<b>setCenter(GoogleMapsCoordinates coordinates)</b>	Permite fijar el centro del mapa.
<b>setHandler(GoogleStaticMapHandler handler)</b>	Permite fijar el manejador de eventos a través de la Interface GoogleStaticMapHandler de la librería MIDMaps.
<b>setMapFormat(java.lang.String format)</b>	Permite fijar el formato de la imagen en la que se va a solicitar el mapa al servidor de Google Maps.
<b>setMapSize(int width, int height)</b>	Permite fijar el tamaño del mapa, recibe como parámetros el ancho y el largo.
<b>setMapType(java.lang.String type)</b>	Permite fijar el tipo de mapa, según los atributos definidos en esta misma clase.

<b>setZoom(int zoomLevel)</b>	Permite fijar el nivel de acercamiento para la imagen del mapa.
<b>update()</b>	Actualiza las propiedades del mapa.
<b>zoomIn()</b>	Aumenta el nivel de acercamiento del mapa.
<b>zoomOut()</b>	Disminuye el nivel de acercamiento del mapa.

**Tabla A.29.** Métodos de la clase *GoogleStaticMap* implementados en Horus.

- **Atributos:** En la tabla A.30 se mencionan y describen los atributos de la clase *GoogleStaticMap* implementados en la aplicación móvil para HORUS.

ATRIBUTO	TIPO	DESCRIPCIÓN
<b>COLOR_BLACK</b>	static java.lang.String	Hace referencia al color negro en los marcadores y polilíneas.
<b>COLOR_BLUE</b>	static java.lang.String	Hace referencia al color azul en los marcadores y polilíneas.
<b>COLOR_BROWN</b>	static java.lang.String	Hace referencia al color café en los marcadores y polilíneas.
<b>COLOR_GRAY</b>	static java.lang.String	Hace referencia al color gris en los marcadores y polilíneas.
<b>COLOR_GREEN</b>	static java.lang.String	Hace referencia al color verde en los marcadores y polilíneas.
<b>COLOR_ORANGE</b>	static java.lang.String	Hace referencia al color naranja en los marcadores y polilíneas.
<b>COLOR_PURPLE</b>	static java.lang.String	Hace referencia al color morado en los marcadores y polilíneas.
<b>COLOR_RED</b>	static java.lang.String	Hace referencia al color rojo en los marcadores y polilíneas.
<b>COLOR_WHITE</b>	static java.lang.String	Hace referencia al color blanco en los marcadores y polilíneas.
<b>COLOR_YELLOW</b>	static java.lang.String	Hace referencia al color amarillo en los marcadores y polilíneas.
<b>ERROR_CONNECTION</b>	static int	Dependiendo del valor indica el tipo de error presentado en la conexión entre el dispositivo móvil y el servidor de mapas de Google.
<b>FORMAT_GIF</b>	static java.lang.String	Hace referencia al formato “.gif” en la imagen del mapa obtenida desde el servidor de Google.
<b>FORMAT_JPG</b>	static java.lang.String	Hace referencia al formato “.jpg” en la imagen del mapa obtenida desde el servidor de Google.
<b>FORMAT_PNG</b>	static java.lang.String	Hace referencia al formato “.png” en la imagen del mapa obtenida desde el servidor de Google.
<b>TYPE_HYBRID</b>	static java.lang.String	Hace referencia al tipo de mapa híbrido. Muestra una capa

		transparente de las calles principales sobre imágenes de satélite.
<b>TYPE_ROADMAP</b>	static java.lang.String	Hace referencia al tipo de mapa callejero. Muestra un plano normal de las calles principales.
<b>TYPE_SATELLITE</b>	static java.lang.String	Hace referencia al tipo de mapa satelital. Muestra imágenes de satélite.
<b>TYPE_TERRAIN</b>	static java.lang.String	Hace referencia al tipo de mapa geografico. Muestra imágenes con características físicas tales como el terreno y la vegetación.

**Tabla A.30.** Atributos de la clase *GoogleStaticMap* implementados en Horus.

#### A.3.5.12 JSMS

JSMS (*Java Short Message Service*) es una API para enviar y recibir mensajes cortos (SMS) y Mensajes Multimedia (MMS<sup>26</sup>, *Multimedia Messages*). Los mensajes pueden ser enviados y recibidos mediante el uso de cualquier dispositivo GSM y también mediante los protocolos de comunicación más comunes de un centro de mensajes cortos (SMSC) (20).

La API JSMS ha sido diseñada con enfoque en una arquitectura modular, lo cual permite una fácil integración de otros medios de transporte de SMS.

#### A.3.5.13 SMSC

Los mensajes de texto son procesados por un SMSC que se encarga de almacenarlos hasta que son enviados. Cuando un usuario envía un SMS, o lo recibe, se incluyen con su carga útil o cuerpo del mensaje (*Payload*) al menos los siguientes parámetros (21):

- Fecha de envío (*time stamp*).
- Validez del mensaje, desde una hora hasta una semana.
- Número de teléfono del remitente y del destinatario.
- Número del SMSC que ha originado el mensaje.

De este modo se asegura el correcto procesamiento del mensaje en el SMSC y a lo largo de toda la cadena. Las funciones del SMSC son:

---

<sup>26</sup> Sistema de mensajería multimedia es un estándar de mensajería que le permite a los teléfonos móviles enviar y recibir contenidos multimedia, incorporando sonido, video, fotos o cualquier otro contenido disponible en el futuro.

- Recibir y almacenar los SMS enviados por los usuarios o por otras fuentes (avisos del operador, sistemas de publicidad, alertas de correo electrónico, etc.) hasta que puedan ser enviados.
- Verificar los permisos del usuario para enviar mensajes, en comunicación con el Registro Local de Ubicación (HLR<sup>27</sup>, *Home Location Register*) de origen.
- Verificar si el usuario al que se envía el mensaje está disponible mediante consulta al Registro de Ubicación de Visitantes (VLR<sup>28</sup>, *Visitor Location Register*) de destino; si está operativo, el mensaje se envía, y si no se almacena temporalmente en el SMSC.
- Verificar periódicamente el estado de los usuarios que tienen mensajes pendientes.

### A.3.6 Diseño Físico de la Base de Datos Implementada en HORUS

A continuación se describen los atributos e identificadores de las entidades que fueron implementadas en el DBMS PostgreSQL v8.4 y que físicamente representan en memoria una tabla donde se almacena la información.

#### A.3.6.1 Tabla Usuarios

En esta tabla se almacena la información más significativa de los usuarios del sistema, según los requerimientos del sistema prototipo de rastreo y control vehicular. A continuación se mencionan los campos o columnas que la componen y una breve descripción:

- **Cedula (PK<sup>29</sup>):** Este campo corresponde a la clave primaria de la tabla.
- **Nombre:** Nombre del usuario (primer y segundo nombre).
- **Apellido:** Apellido del usuario (primer y segundo apellido).
- **Login:** Nombre de la cuenta de usuario con la que se ingresa al sistema.
- **Password:** Contraseña de la cuenta de usuario con la que se valida el ingreso al sistema.
- **Género:** Género del usuario. Se almacena M para masculino y F para femenino.
- **Teléfono:** Número de teléfono del usuario.
- **Celular:** Número del teléfono celular del usuario.
- **Dirección:** Dirección de residencia del usuario.
- **Email:** Correo electrónico del usuario.
- **Tipo:** Tipo de usuario según los requerimientos de HORUS (cliente, administrador, operador o conductor).

<sup>27</sup> Base de datos que permite localizar los usuarios de una determinada red GSM.

<sup>28</sup> Registro utilizado para seguir la pista de localización de un usuario en movimiento dentro de una red GSM de telefonía móvil celular.

<sup>29</sup> Hace referencia a la clave primaria o llave primaria de la tabla (PK, *Primary Key*).



- **Foto:** Nombre del archivo donde esta almacenada la foto digital del usuario.
- **Nacimiento:** Fecha de nacimiento del usuario.
- **Ciudad:** Ciudad de residencia del usuario.

Cabe resaltar que los anteriores datos se consideran comunes para todos los usuarios del sistema.

### A.3.6.2 Tabla Vehículos

En esta tabla se almacena la información más significativa de los vehículos asociados al sistema. A continuación se mencionan los campos o columnas que la componen y una breve descripción:

- **Id [PK]:** Este campo corresponde a la clave primaria de la tabla. Cada vehículo registrado en HORUS tendrá asociado un identificador numérico.
- **Propietario [FK<sup>30</sup>]:** Numero de cedula del propietario del vehículo. Hace referencia al campo “cedula” de la tabla cliente.
- **Conductor [FK]:** Numero de cedula del conductor del vehículo. Hace referencia al campo “id\_conductor” de la tabla conductores.
- **Placa:** Caracteres y números de la placa del vehículo.
- **Modelo:** Modelo del vehículo (año de fabricación).
- **Tipo:** Tipo de vehículo (automóvil, camión campero o furgoneta).
- **Marca:** Marca comercial del vehículo.
- **Foto:** Nombre del archivo donde esta almacenada la foto digital del vehículo.

### A.3.6.3 Tabla Clientes

En esta tabla se relacionan los usuarios de HORUS que han sido registrados como clientes con los vehículos. A continuación se listan los campos o columnas que la componen y una breve descripción:

- **Cédula [PK]:** Este campo corresponde a la clave primaria de la tabla.
- **Id\_vehículo [FK]:** Este campo corresponde a la clave foránea de la tabla. Número que identifica cada vehículo registrado en HORUS y hace referencia al campo “id” de la tabla vehículos.

### A.3.6.4 Tabla Conductores

En esta tabla se relacionan los usuarios de HORUS que han sido registrados como conductores y los vehículos. A continuación se mencionan los campos o columnas que

---

<sup>30</sup> Hace referencia a la clave secundaria o llave foránea de la tabla (FK, *Foreign Key*).

la componen y una breve descripción:

- **Id\_conductor [PK]:** Este campo corresponde a la clave primaria de la tabla.
- **Id\_vehículo [FK]:** Este campo corresponde a la clave foránea de la tabla. Número que identifica cada vehículo registrado en HORUS y hace referencia al campo “id” de la tabla vehículos.
- **Licencia:** Número de la licencia de conducción.

#### A.3.6.5 Tabla Direcciones

En esta tabla se relaciona la información de las características de lugares reconocidos de la ciudad y sus datos de localización geográfica, como una herramienta para personalizar el mapa según la información del sistema. A continuación se mencionan los campos o columnas que la componen y una breve descripción:

- **Dirección [PK]:** Este campo corresponde a la clave primaria de la tabla. Tipo de dato compuesto que almacena la dirección de un lugar específico.
- **Latitud:** Latitud en coordenadas decimales correspondientes a la dirección de referencia.
- **Longitud:** Longitud en coordenadas decimales correspondientes a la dirección de referencia.
- **Icono:** Nombre y el formato de la imagen que contiene el icono utilizado para representar determinado lugar en el mapa.
- **Palabra\_clave:** Palabra asociada al lugar o a la dirección. Se implementa con el fin de facilitar la búsqueda de lugares en el mapa.
- **Imagen:** Nombre y formato de la imagen que contiene la imagen del lugar asociado a la dirección.
- **Descripción:** Breve descripción del lugar asociado a la dirección.

#### A.3.6.6 Tabla Tareas

En esta tabla se almacena la información necesaria para realizar la gestión de tareas asignadas a vehículos y conductores desde la aplicación web a la aplicación que tienen acceso los operadores de HORUS. A continuación se mencionan los campos o columnas que la componen y una breve descripción:

- **Id\_tarea [PK]:** Este campo corresponde a la clave primaria de la tabla. Se representa con un valor numérico, único e irrepetible asociado a la tarea.
- **Tipo:** Valor numérico que corresponde al tipo de tarea a realizar (recoger pasajero o recoger mercancía).
- **Descripción:** Breve descripción de la tarea.

- **Origen:** Origen de la tarea. Identificación del operador que la asigna.
- **Destino:** Dirección del destino asociado a la tarea.
- **Idcond\_asign [FK]:** Cedula del conductor a quien se le asigna una tarea. Hace referencia al campo “id\_conductor” de la tabla conductores.
- **Estado:** Estado de la tarea (atendido o pendiente).
- **Idv\_asign [FK]:** Numero de identificación del vehículo en el que se va a realizar la tarea asignada. Hace referencia al campo “id” de la tabla vehículos.

#### A.3.6.7 Tabla Emergencias

En esta tabla se almacena la información necesaria para realizar la gestión de emergencias reportadas por los conductores de los vehículos. A continuación se mencionan los campos o columnas que la componen y una breve descripción:

- **Id\_emergencia [PK]:** Este campo corresponde a la clave primaria de la tabla. Valor numérico único e irrepetible asociado a la emergencia.
- **Idv\_emergencia [FK]:** Identificación del vehículo que reporta la emergencia. Hace referencia al campo “id” de la tabla vehículos.
- **Conductor [FK]:** Cedula del conductor que reportó la emergencia. Hace referencia al campo “id\_conductor” de la tabla conductores.
- **Hora:** Hora en que se reportó la emergencia.
- **Fecha:** Fecha en que se reportó la emergencia.
- **Lat\_emergencia:** Latitud del lugar donde se presentó la emergencia en el formato de coordenadas enviados por el GPS según la trama NMEA.
- **Hlat\_emergencia:** Hemisferio de la latitud del lugar donde se reportó la emergencia, N para latitud norte y S para latitud sur.
- **Long\_emergencia:** En este campo se almacena la longitud del lugar donde se presentó la emergencia en el formato de coordenadas enviados por el GPS según la trama NMEA.
- **Hlong\_emergencia:** Hemisferio de la longitud del lugar donde se reportó la emergencia, W para longitud oeste y E para longitud este.
- **Estado:** Corresponde al estado de la emergencia (atendido o pendiente).

#### A.3.6.8 Tabla Recorridos

En esta tabla se almacena toda la información relacionada con la ubicación geográfica de los vehículos. A continuación se mencionan los campos o columnas que la componen y una breve descripción:

- **Id\_recorrido [PK]:** Este campo corresponde a la clave primaria de la tabla. Valor numérico único e irrepetible asociado a la posición del vehículo en un instante de tiempo determinado.
- **Latitud:** Latitud de la posición del vehículo en el formato de coordenadas enviados por el GPS según la trama NMEA en un instante de tiempo determinado.
- **Hemisferio\_lat:** Hemisferio de la posición del vehículo en un lugar determinado, N para latitud norte y S para latitud sur.
- **Longitud:** Longitud de la posición del vehículo en el formato de coordenadas enviados por el GPS según la trama NMEA en un instante de tiempo determinado.
- **Hemisferio\_long:** Hemisferio de la posición del vehículo en un lugar determinado, W para longitud oeste y E para longitud este.
- **Id\_vehiculo [FK]:** Hace referencia al campo “id” de la tabla vehículo y almacena la identificación del vehículo asociado a la posición.
- **Hora:** Hora en la que el dispositivo GPS midió la posición geográfica.
- **Velocidad:** Velocidad del vehículo en el momento en que el dispositivo GPS midió la posición geográfica.
- **Fecha:** Fecha en la que el servidor central recibe la posición geográfica enviada por las unidades móviles.
- **rumbo:** Rumbo de navegación del vehículo, enviado por el dispositivo GPS según formato de la trama NMEA.

### A.3.7 Descripción del Funcionamiento del Protocolo de HORUS

Para probar el funcionamiento del protocolo implementado en HORUS, se hizo uso del software de simulación ISIS Professional. En esta herramienta se simularon los componentes servidor, central y móvil que se comunican en el sistema.

La figura A.44 muestra el proceso desarrollado al iniciar HORUS, mostrando las tramas intercambiadas entre servidor, central y móvil. Se han enumerado las tramas del 1 al 22 de acuerdo al orden en que son enviadas. A continuación se realiza una descripción de las tramas enviadas en cada paso del proceso.

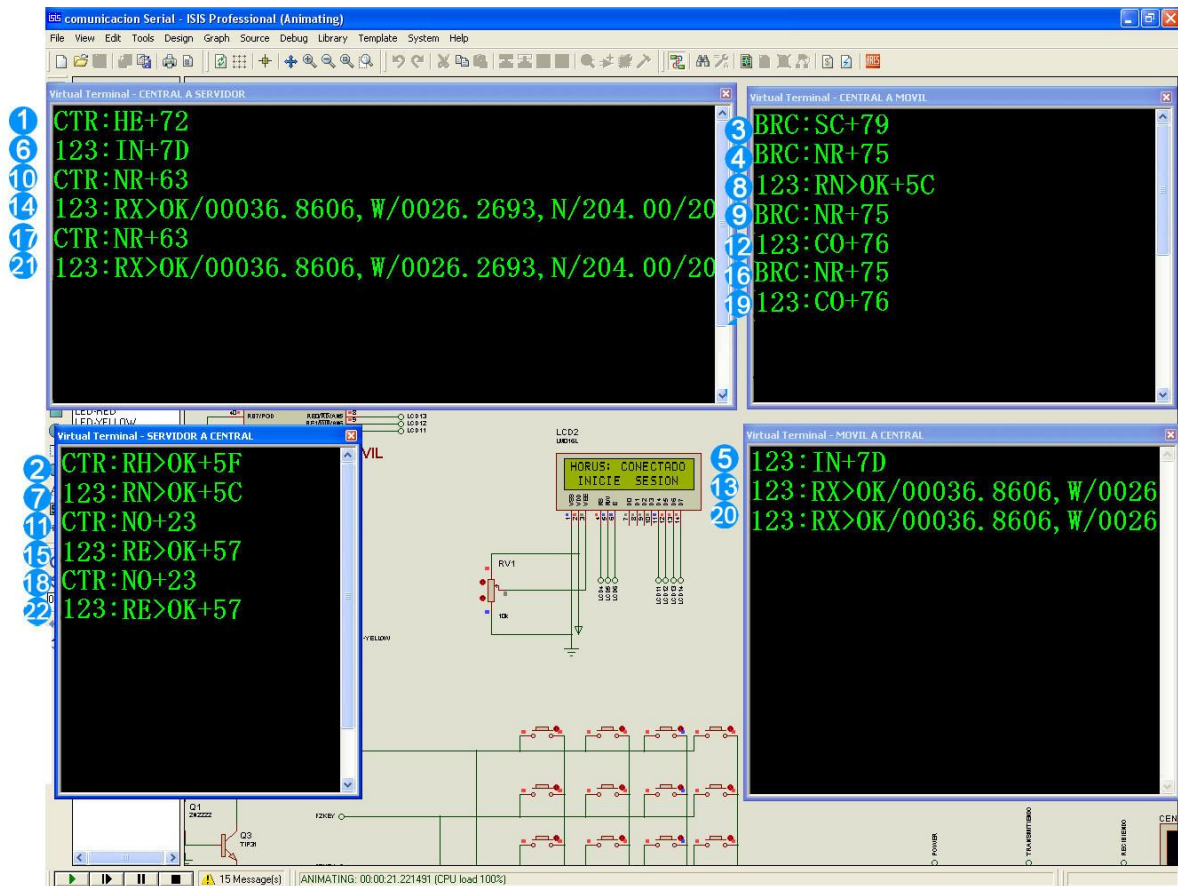


Figura A.44. Intercambio de tramas entre Servidor, Central y Móvil.

En la figura A.45 se muestra la estructura de la trama de datos definida en el protocolo de HORUS. En todas las tramas se envía el preámbulo definido.

Preámbulo	ID Origen/Destino	:	Tipo de Solicitud/Respuesta	>	Resultado /Información	+	Checksum
-----------	-------------------	---	-----------------------------	---	------------------------	---	----------

Figura A.45. Estructura de la trama definida en HORUS.

A continuación se describen las tramas enviadas en cada paso del proceso mostrado en la figura A.44 y la información que se ocupa en cada campo de la trama mostrada en la figura A.45:

1. La central envía mensaje de *Hello* hacia el servidor para determinar que éste se encuentre *online*.

El campo "ID Origen/Destino" de la trama se encuentra en "CTR", indicando que es la central quien envía el mensaje.

Está el separador ":" y el campo "Tipo de Solicitud/Respuesta" contiene "HE", correspondiente al indicador de mensaje de *Hello*.

En este caso se trata de una trama básica, por lo que el campo “Resultado/Información” no se tiene en cuenta, y tampoco hace falta la presencia del separador “>”.

Finalmente se tiene el separador “+” y el valor del *Checksum* calculado.

2. El servidor responde a la central su estado activo.

El campo “ID Origen/Destino” es “CTR”, indicando que la trama va dirigida hacia la central.

Separador “:” y el campo “Tipo de Solicitud/Respuesta” contiene “HE”, como respuesta al *Hello* enviado por la central.

Separador “>” y el campo “Resultado/Información” contiene “OK” notificando su estado activo.

Separador “+” y finalmente al valor del *Checksum*.

Asimismo las demás tramas muestran el contenido de cada campo en el protocolo implementado en HORUS, dependiendo de su origen y destino. A continuación se continúa explicando el proceso de una manera resumida:

3. La central envía un mensaje *broadcast* a los móviles informando disponibilidad del servidor.
4. La central envía un mensaje *broadcast* a los móviles preguntando si alguno tiene algún tipo de solicitud.
5. El móvil con ID=123 responde a la central con una solicitud de ingreso a la red.
6. La central reenvía al servidor la solicitud de ingreso del móvil 123.
7. El servidor responde positivamente a la central ante la solicitud de ingreso del móvil 123.
8. La central retransmite la respuesta positiva del servidor hacia el móvil 123, permitiendo su acceso a la red.
9. La central envía otro mensaje *broadcast* preguntando si algún otro móvil tiene una solicitud.
10. La central pregunta al servidor si tiene solicitudes.
11. Servidor responde a central que no tiene solicitudes.

En cualquier caso que el servidor no responda, se regresa al paso 10 hasta cuatro veces más, y si en ninguna de éstas se recibe respuesta entonces se considera desconectado al servidor y se envía un *broadcast* a los móviles informando que éste no está disponible. Esto hace que se envíe un mensaje de *Hello* periódicamente por parte de la central hasta recibir respuesta. Los ingresos a la red y/o inicios de sesión de los móviles se pierden. Cuando el servidor responde ante la

central, se envía un mensaje *broadcast* a los móviles indicando la disponibilidad del servidor (paso 3).

12. La central pide coordenadas al móvil 123.

13. Móvil 123 responde enviando sus coordenadas.

La información enviada por el móvil en esta trama es el mensaje *GPRMC* del protocolo NMEA especificado en la sección 3.1.1.2.

14. La central reenvía coordenadas de 123 al servidor.

15. El servidor confirma a la central la recepción de coordenadas.

16. Nuevamente se repite el proceso en que la central pregunta a los móviles si tienen una solicitud.

17. Central pregunta a servidor se tiene nueva solicitud.

18. Servidor responde a central que no tiene solicitudes.

19. Central pide coordenadas al móvil 123.

20. Móvil 123 responde con el envío de sus coordenadas.

21. Central reenvía coordenadas del móvil 123 al servidor.

22. Servidor confirma la recepción de coordenadas.

Este proceso se sigue repitiendo, de tal forma que cuando haya un nuevo móvil, éste solicite el ingreso a la red cuando la central pregunte por nuevas solicitudes de móviles, se confirme su posibilidad de acceso por parte del servidor y se soliciten sus coordenadas.

Con esta prueba se encontraron y corrigieron errores de comunicación entre los componentes mostrados, y se identificaron errores en el software programado inicialmente en el microcontrolador. Al corregir estas fallas se consiguió un mejor manejo de la información y un mejor funcionamiento práctico del protocolo planteado.

#### **A.4 ASPECTOS COMPLEMENTARIOS DEL CAPÍTULO 4 – PRUEBAS Y RESULTADOS.**

En esta sección se explican aspectos complementarios relacionados con las pruebas realizadas con el fin de caracterizar el funcionamiento y desempeño del sistema prototipo de control y monitoreo vehicular asistido por GPS - HORUS.

##### **A.4.1 Relación de Potencia Recibida con Voltaje Medido en el Pin RSSI del Radio Motorola Pro 3100.**

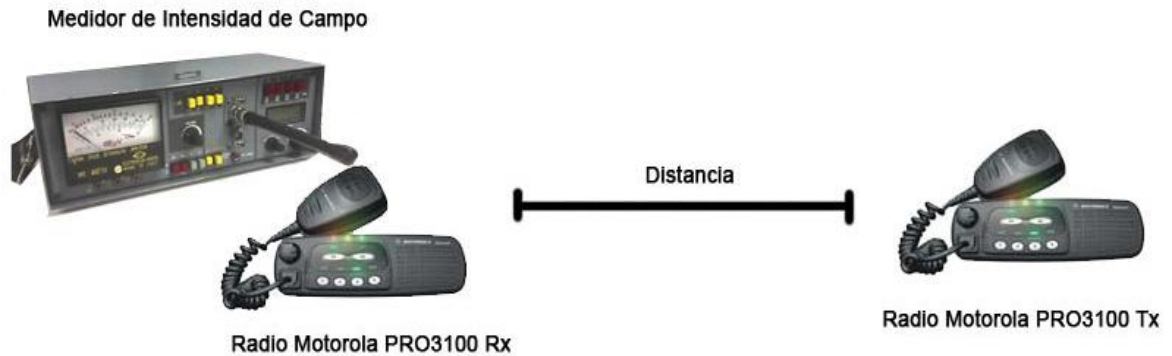
Para encontrar una relación entre el voltaje medido en el pin RSSI del radio Motorola PRO 3100 y la potencia de recepción, se realizó una prueba donde se midió un nivel

de voltaje asociado al campo eléctrico recibido por una antena. Para ello, se utilizó un medidor de intensidad de campo y una antena Yaesu para FT411, con patrón de radiación omnidireccional utilizada en VHF. En la figura A.46 se muestran estos dos elementos.



**Figura A.46.** Medidor de intensidad de campo y antena Yaesu para FT411.

En la figura A.47 se muestra el montaje realizado con los elementos para realizar la prueba.



**Figura A.47.** Montaje de los equipos para realizar la prueba de medición de campo eléctrico.

La prueba consistió en realizar transmisiones con uno de los radios, ubicado a diferentes distancias del medidor de intensidad de campo y del otro radio. Inicialmente se conectó la antena Yaesu al radio receptor con el fin de medir el voltaje en el pin RSSI, y posteriormente se conectó la antena al medidor de intensidad del campo, buscando determinar una relación entre el voltaje medido en el pin RSSI y el voltaje mostrado por el medidor de intensidad de campo. En la tabla A.31 se presenta la relación entre el voltaje del pin RSSI y el voltaje medido.



RSSI (V)	Nivel Medido de Voltaje (dBuV)
2.385	69
2.38	67
2.362	60
2.359	59
2.263	57.5
2.173	46.5
2.164	46
2.126	41
1.953	40
1.921	36
1.895	35
1.810	32
1.791	25
1.786	23

**Tabla A.31.** Relación entre voltaje del pin RSSI y el nivel medido de campo eléctrico.

Para encontrar la relación entre el pin RSSI y potencia recibida, fue necesario encontrar la intensidad de campo eléctrico partir del nivel voltaje medido. La ecuación B.18 muestra la relación existente entre el nivel medido de voltaje y la frecuencia de la señal, especificada en el manual del medidor de intensidad de campo.

$$E[dBuV/m] = Le[dBuV] - 33.6 + 20\text{Log}(f[MHz]) \quad (B.18)$$

Donde:

$E$ : Intensidad de campo eléctrico.

$Le$ : Nivel de voltaje medido.

$f$ : Frecuencia de la señal.

Tomando el primer valor de nivel medido de voltaje de la tabla A.31, y la frecuencia 151.162 MHz y reemplazándolos en la ecuación B.18, se tiene:

$$E = 69 - 33.6 + 20\text{Log}(151.162)$$

$$E = 78.9889 \text{ dBuV/m}$$

La ecuación B.19 muestra la relación de potencia recibida, en función de la intensidad de campo eléctrico y frecuencia de la señal.

$$P_{Rx}[dBm] = E[dBuV/m] - 20\text{Log}(f[MHz]) - 77.2 \quad (B.19)$$

Reemplazando en la ecuación B.19 la intensidad de campo hallada a través de la ecuación B.18 y la frecuencia de la señal, se tiene:

$$P_{Rx} = 78.9889 - 20\text{Log}(151.162) - 77.2$$

$$P_{Rx} = -41.8 \text{ dBm}$$

Se procede de la misma manera para todos los valores de la tabla A.31 para hallar la potencia de recepción. La tabla A.32 muestra la relación entre el nivel de voltaje medido y la potencia de recepción.

RSSI (V)	Nivel Medido de Voltaje (dBuV)	Potencia de Recepción (dBm)	Potencia de Recepción (mW)
2.385	69	-41.8	6.60693E-05
2.38	67	-43.8	4.16869E-05
2.362	60	-50.8	8.31764E-06
2.359	59	-51.8	6.60693E-06
2.263	57.5	-53.3	4.67735E-06
2.173	46.5	-64.3	3.71535E-07
2.164	46	-64.8	3.31131E-07
2.126	41	-69.8	1.04713E-07
1.953	40	-70.8	8.31764E-08
1.921	36	-74.8	3.31131E-08
1.895	35	-75.8	2.63027E-08
1.810	32	-78.8	1.31826E-08
1.791	25	-85.8	2.63027E-09
1.786	23	-87.8	1.65959E-09

**Tabla A.32.** Relación entre voltaje del pin RSSI, nivel medido de campo eléctrico y potencia de recepción.

Teniendo los valores de la tabla A.32, es posible hallar la potencia recibida para las cuatro locaciones de la prueba descrita en la sección 4.2.1. En esta prueba se midió el voltaje del pin RSSI del radio, para los cuales se determinó una potencia de recepción interpolando los valores de la tabla A.32. Los valores de potencia de recepción encontrados para las locaciones se muestran en la tabla A.33.

Locación	RSSI Señal (V)	Potencia Señal Rx (dBm)	Potencia Señal Rx (mW)
Parqueadero FIET	2.364	-50.0222	9.94901E-06
Oficina de Tránsito	2.346	-52.0031	6.30507E-06
Estadio Ciro López	2.343	-52.05	6.23735E-06
Villa del viento	2.104	-69.9272	1.0169E-07
SENA (Sede Norte)	2.049	-70.2451	9.45127E-08

**Tabla A.33.** Relación entre voltaje del pin RSSI y la potencia de recepción.

#### A.4.1.1 Cálculo de Eb/No

La energía de bit ( $E_b$ ) puede determinarse a partir de la ecuación B.20.

$$E_b[dBw/Hz] = P_{Rx}[dBw] - 10\text{Log}(R[bps]) \quad (\text{B.20})$$

En la tabla A.34 se presenta la potencia de recepción medida en las locaciones parque parqueadero FIET, oficina de tránsito, estadio Ciro López, barrio villa del viento y SENA sede norte. Estos valores se hallaron a partir de la tabla A.33, teniendo en cuenta las ecuaciones B.21 y B.22.

$$P[dBw] = 10\text{Log}\left(\frac{P[w]}{1w}\right) \quad (\text{B.21})$$

$$P[dBm] = 10\text{Log}\left(\frac{P[mw]}{1mw}\right) \quad (\text{B.22})$$

Locación	Potencia Rx (dBm)	Potencia Rx (mW)	Potencia de Rx (dBw)
Parqueadero FIET	-50.0222	$9.94901 \times 10^{-6}$	-80.0222
Oficina de Tránsito	-52.0031	$6.30507 \times 10^{-6}$	-82.0031
Estadio Ciro López	-52.05	$6.23735 \times 10^{-6}$	-82.05
Villa del viento	-69.9272	$1.0169 \times 10^{-7}$	-99.9272
SENA (Sede Norte)	-70.2451	$9.45127 \times 10^{-8}$	-100.2451

**Tabla A.34.** Potencia de recepción en prueba de vehículo con unidad móvil alejado del lóbulo principal de antena de estación de control.

Haciendo uso de la ecuación (E.20) se halla la energía de bit para cada locación. La velocidad de transmisión de datos es de 2000 bps.

- Parqueadero FIET

$$E_b = -80.0222 - 10\text{Log}(2000)$$

$$E_b = -113.0324 \text{ dBw/Hz}$$

- Oficina de Tránsito

$$E_b = -82.0031 - 10\text{Log}(2000)$$

$$E_b = -115.0134 \text{ dBw/Hz}$$

- Estadio Ciro López

$$E_b = -82.05 - 10\text{Log}(2000)$$

$$Eb = - 115.0602dBw/Hz$$

- Barrio Villa del Viento

$$Eb = -99.9272 - 10Log(2000)$$

$$Eb = -132.93745 dBw/Hz$$

- SENA (Sede Norte)

$$Eb = -100.2451 - 10Log(2000)$$

$$Eb = - 133.2553 dBw/Hz$$

La Densidad Espectral de Potencia de Ruido<sup>31</sup> (No) puede determinarse con la ecuación B.23.

$$No[dBw/Hz] = -204[dBw] + NF[dB] \quad (B.23)$$

Asumiendo una figura de ruido para el radio Motorola PRO3100 de 7dB, se tiene:

$$No = -204 + 7$$

$$No = -197 dBw/Hz$$

La tabla A.35 muestra la relación Eb/No para cada locación.

Locación	Eb/No (dB)
Parqueadero FIET	83.9676
Oficina de Tránsito	81.9866
Estadio Ciro López	81.9398
Villa del viento	64.0325
SENA (Sede Norte)	63.7447

**Tabla A.35.** Relación Eb/No para las locaciones de la prueba de vehículo con unidad móvil en dirección al lóbulo principal de antena de estación de control.

En la tabla A.36 se presenta la potencia de recepción medida en las locaciones parque Benito Juárez, barrio las Américas e ISS sede Villamarista. De igual forma se calculó la relación Eb/No.

<sup>31</sup> Debida al ruido térmico sobre un ancho de banda de 1 Hz.

Locación	Potencia Rx (dBm)	Potencia Rx (mW)	Potencia de Rx (dBw)
Parque Benito Juárez	-93.666	$4.29932 \times 10^{-10}$	-123.666
Barrio las Américas	-91.838	$6.54938 \times 10^{-10}$	-121.838
ISS Sede Villamarista	-91.271	$7.46277 \times 10^{-10}$	-121.271

**Tabla A.36.** Potencia de recepción en prueba de vehículo con unidad móvil alejado del lóbulo principal de antena de estación de control.

Haciendo uso de la ecuación (E.20) se halla la energía de bit para cada locación. La velocidad de transmisión de datos es de 2000 bps.

- Parque Benito Juárez:

$$Eb = -123.666 - 10\text{Log}(2000)$$

$$Eb = -156.676 \text{ dBw/Hz}$$

- Barrio las Américas:

$$Eb = -121.838 - 10\text{Log}(2000)$$

$$Eb = -154.848 \text{ dBw/Hz}$$

- ISS Sede Villamarista:

$$Eb = -121.271 - 10\text{Log}(2000)$$

$$Eb = -154.281 \text{ dBw/Hz}$$

La tabla A.37 muestra la relación Eb/No para cada locación.

Locación	Eb/No (dB)
Parque Benito Juárez	40.324
Barrio las Américas	42.152
ISS Sede Villamarista	42.719

**Tabla A.37.** Relación Eb/No para las locaciones de la prueba de vehículo con unidad móvil alejado del lóbulo principal de antena de estación de control.

#### A.4.2 Formula de Haversine

La fórmula de Haversine es una ecuación que permite calcular la distancia máxima entre dos puntos del globo terrestre conociendo su latitud y longitud. Su nombre se deriva de la función matemática  $\text{haversin}(\theta)$ .

Según la fórmula, para cualquier par de puntos sobre una esfera (22):

$$\text{haversin}\left(\frac{d}{R}\right) = \text{haversin}(\varphi_1 - \varphi_2) + \cos(\varphi_1)\cos(\varphi_2)\text{haversin}\Delta\lambda \quad (\text{B.24})$$

Donde:

- $\text{haversin}(\theta) = \text{sen}^2(\theta/2) = \{(1 - \cos(\theta))/2\}$ ,
- $d$ : distancia entre los dos puntos sobre la esfera [m].
- $R$ : radio de la esfera [m].
- $\varphi_1$ : latitud del punto 1.
- $\varphi_2$ : la latitud del punto 2.
- $\Delta\lambda$ : es la diferencia de longitud.

El argumento de la función  $\text{haversin}$  debe estar en radianes. En grados,  $\text{haversin}(d/R)$  de la fórmula se convertiría en  $\text{haversin}(180 * d / \pi R)$ . Al despejar la variable “d” de la ecuación (B.24) se obtiene la ecuación B.25:

$$d = R * \text{haversin}^{-1}(h) = 2R * \arcsin(\sqrt{h}) \quad (\text{B.25})$$

Donde:

- $h = \text{haversin}(d/R)$

Esta fórmula es sólo una aproximación cuando se aplica a la Tierra, porque la Tierra no es una esfera perfecta; el radio  $R$  de la Tierra varía de 6.356.78 kilómetros en los polos hasta 6378.14 kilómetros en el ecuador. Hay pequeñas correcciones, típicamente del orden de 0.1% (suponiendo la media geométrica  $R = 6.367.45$  kilómetros), a causa de la ligera forma elíptica del planeta.

La formula de Haversine se utilizó para calcular la distancia existente entre las coordenadas de un vehículo y un lugar específico, con el fin de despachar un vehículo según el criterio de la distancia más corta.

Esta fórmula se implementó en Java dentro del método  $\text{calcularDistancia}()$  de la clase *HorusManager* en el paquete control de la aplicación, desarrollada para el servidor central; el método recibe como parámetros la latitud y longitud de dos coordenadas

geográficas y retorna como resultado en un tipo de dato *String*, la distancia en kilómetros existente entre las dos coordenadas ingresadas. En la figura A.48 se muestra la implementación en Java de la fórmula de Haversine.

```
public String calcularDistancia(String lat1, String long1, String lat2, String long2) {  
  
    String rtaDistancia = "";  
  
    double valorLat1 = Float.parseFloat(lat1);  
    double valorLong1 = Float.parseFloat(long1);  
    double valorLat2 = Float.parseFloat(lat2);  
    double valorLong2 = Float.parseFloat(long2);  
  
    double earthRadius = 6371; // Km  
    double dLat = Math.toRadians(valorLat2 - valorLat1);  
    double dLng = Math.toRadians(valorLong2 - valorLong1);  
    double a = Math.sin(dLat / 2) * Math.sin(dLat / 2) +  
        Math.cos(Math.toRadians(valorLat1)) * Math.cos(Math.toRadians(valorLat2))  
        * Math.sin(dLng / 2) * Math.sin(dLng / 2);  
    double c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));  
    double dist = earthRadius * c; //Distancia en Kilometros  
    float rtaDist = (float) dist;  
  
    rtaDistancia = "" + rtaDist;  
  
    return rtaDistancia;  
}
```

**Figura A.48.** Implementación en Java de la fórmula de Haversine.

### A.4.3 Herramientas Utilizadas Durante la Fase de Implementación y Pruebas del Sistema.

Durante la fase de implementación, pruebas y desarrollo de las aplicaciones implementadas en HORUS se utilizaron diferentes herramientas software que permitieron diseñar y modelar la estructura de los programas; generar el respectivo código, compilarlo y facilitar el despliegue de los servicios a través de componentes hardware; tal como se muestra en la arquitectura del capítulo 3 del documento principal.

En la tabla A.38 se observan las herramientas utilizadas y una breve descripción, donde se menciona de manera cualitativa el uso que se les dio y su comportamiento durante el proceso de diseño, elaboración e implementación de HORUS.

HERRAMIENTA	OBSERVACIONES
<b>ENTORNOS DE DESARROLLO INTEGRADO (IDE)</b>	
<b>Net Beans v6.5</b>	Es una herramienta diseñada para escribir, compilar, depurar y ejecutar programas. Es un Entorno de Desarrollo Integrado (IDE <sup>32</sup> , <i>Integrated Development Environment</i> ) de código abierto escrito completamente en Java que soporta el desarrollo de todos los tipos de aplicación en este lenguaje (J2SE, aplicaciones web, EJB y aplicaciones móviles). Este IDE se utilizó para desarrollar las aplicaciones “HorusWebApp” y “HorusMobileApp”.
<b>PCWHD – CCS v4.104</b>	Este IDE se utilizó para desarrollar los programas de los microcontroladores PIC18f452 que conforman la unidad móvil y la UCCA - EC; incluye un editor de texto, el compilador, el intérprete, y el depurador de código para el lenguaje C++. Tiene la capacidad para trabajar independientemente o en integración con el IDE de MPLAB (entorno de desarrollo integrado de las empresas MICROCHIP).
<b>MPLAB IDE v8.36</b>	Es un conjunto de herramientas libres e integradas para el desarrollo de aplicaciones destinadas a microcontroladores PIC; se ejecuta como una aplicación de 32 bits en Microsoft Windows, es fácil de usar e incluye una serie de componentes de software libre para el rápido desarrollo de aplicaciones y depuración de código. Esta herramienta se utilizó para depurar el código de HORUS desarrollado sobre el IDE CCS.
<b>INTERFAZ DE PROGRAMACION DE APLICACIONES (API)</b>	
<b>Google Maps API v3.0</b>	El <i>JavaScript</i> de <i>Google Maps</i> API permitió integrar <i>Google Maps</i> en los archivos JSP que conforman la interfaz gráfica de la aplicación web desarrollada para HORUS. La versión 3 de esta API es especialmente diseñada para ser más rápida y más aplicable a los dispositivos móviles, así como las tradicionales aplicaciones de navegador de escritorio.
<b>MIDMaps API v1.0</b>	Esta librería permitió integrar <i>Google Maps</i> en las aplicaciones Java ME mediante la API de mapas estáticos de <i>Google</i> ( <i>Google Statics Maps API</i> ). A través del envío de solicitudes HTTP desde el dispositivo móvil, donde se incluían los parámetros y atributos del mapa, se obtuvieron las imágenes correspondientes a la posición y localización de los vehículos.
<b>JSMS v2.2.8.</b>	Esta API fue implementada para incluir en las aplicaciones de HORUS el servicio de envío y recepción de mensajes cortos (SMS). La API <i>JSMS</i> ha sido diseñada con enfoque en una arquitectura modular, lo cual permite que además de GSM se integren fácilmente otros medios o tecnologías de transporte de mensajes cortos, lo que hace al proceso transparente para el

<sup>32</sup> Un entorno de desarrollo integrado es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaces gráficas. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, puede utilizarse para varios.



	usuario.
<b>GiovyNetDriver v1.3</b>	Esta API permitió desarrollar la interfaz para la comunicación entre el servidor central y la unidad de control, conmutación y acondicionamiento EC. Es una solución Java de código abierto que se utiliza para el control de puertos seriales RS-232.
<b>SERVIDORES</b>	
<b>Glassfish v2.0.</b>	Glassfish es el servidor de aplicaciones Java basado en la Edición Empresarial de la Plataforma Java (JEE, <i>Java Platform Enterprise Edition</i> ) este servidor permite la publicación de aplicaciones web gracias a su contenedor web que soporta las tecnologías web de servlets y JSP, así como también componentes empresariales gracias a su contenedor de <i>Enterprise JavaBeans</i> (EJB) que permiten publicar servicios basados en tecnologías EJB y en la API para Persistencia de Java (JPA, <i>Java API Persistence</i> ), entre otras. Este servidor se utilizó para realizar las pruebas de despliegue de la aplicación web desarrollada para HORUS sobre el <i>localhost</i> .
<b>Apache Tomcat v6.0.18</b>	Apache Tomcat es un servidor web con soporte para <i>servlets</i> y JSP. No es considerado como un servidor de aplicaciones ya que funciona como servidor web por sí mismo. Este servidor fue utilizado para realizar las pruebas de despliegue de la aplicación web desarrolladas para HORUS sobre internet.
<b>PostgreSQL v8.4</b>	Este fue el servidor de base de datos de libre distribución utilizado para almacenar toda la información asociada a HORUS. Permite que mientras se escribieran procesos en una tabla, otros accedieran a la misma tabla o a otra tabla diferente sin que se presentaran bloqueos. Este servidor provee soporte para una amplia variedad de tipos nativos como números de precisión arbitraria, texto de longitud ilimitada, direcciones IP (IPv4 e IPv6), direcciones MAC y arreglos, entre otros.
<b>EMULADORES</b>	
<b>Proteus 7 Professional</b>	<p>PROTEUS es un entorno integrado diseñado para la realización completa de proyectos de construcción de equipos electrónicos en todas sus etapas: diseño, simulación, depuración y construcción. En HORUS se utilizaron dos elementos fundamentales en la etapa de diseño y construcción de la unidad móvil y la unidad de control, conmutación y acondicionamiento EC:</p> <ul style="list-style-type: none"> <li>• ISIS v7.5, la herramienta para la elaboración avanzada de esquemas electrónicos, que incorpora una librería de más de 6.000 modelos de dispositivos digitales y analógicos.</li> <li>• ARES v7.5, la herramienta para la elaboración de tarjetas de circuito impreso con posicionador automático de elementos y generación automática de pistas, que permite el uso de hasta 16 capas.</li> </ul>
<b>Filterlab v2.0</b>	Es una herramienta software gratuita que permite realizar el diseño de filtros activos. Genera el diagrama circuital de los filtros con los respectivos valores de sus componentes, así como

	una gráfica de la respuesta en frecuencia. Este software fue utilizado con el fin de hallar el valor de los componentes de los filtros implementados en las tarjetas del sistema.
<b>Eltima Virtual Serial Port Driver v6.0</b>	Durante la etapa de pruebas esta herramienta brinda la posibilidad de crear puertos virtuales con la misma funcionalidad que los puertos de hardware real. Este software garantiza la conexión segura y rápida entre los puertos virtuales y permite crear en el sistema tantos puertos seriales virtuales como sea posible para satisfacer las necesidades del usuario. Todo el funcionamiento depende de la velocidad del procesador; ahorrando tiempo, dinero y esfuerzo, debido a que el enlace entre los puertos serie virtuales es mucho más rápido y más seguro que la verdadera conexión hardware. Una vez creado el puerto serial virtual puede tener cualquier nombre (COM por defecto) o pueden tener el mismo nombre que el puerto serie real.
<b>Hercules v3.2.3</b>	Esta herramienta es un software libre que permite realizar conexiones a través del puerto serial (RS-485 y RS-232), terminal UDP/IP y terminal TCP/IP Cliente-Servidor. Este software se utilizó para realizar diferentes pruebas de funcionamiento del protocolo de comunicación del sistema, del lado de la central de despacho, donde se encuentra el computador conectado a través del puerto serial.
<b>Free Serial Port Monitor v3.3.1</b>	Esta herramienta se utilizó con éxito para el monitoreo del flujo de datos entre dispositivos serie y aplicaciones de Windows. La utilidad de monitoreo de esta herramienta permitió espiar, capturar, ver y analizar el flujo de datos a través de cada uno de los puertos seriales a través de los cuales se estableció la conexión entre la Unidad de Control, Conmutación y Acondicionamiento EC y la aplicación "CalculoBER.jar", utilizada para medir la tasa de errores de bits en HORUS.
<b>Sony Ericsson SDK<sup>33</sup> v2.5.0.6.</b>	Estas herramientas permitieron simular el comportamiento de las aplicaciones desarrolladas para los dispositivos móviles, considerando diferentes fabricantes, diferentes modelos de teléfonos celulares y las características asociadas a cada uno de estos. La aplicación móvil desarrollada para clientes móviles de HORUS, fue simulada con estos programas y a diferencia del Kit de Desarrollo de Software (SDK, <i>Software Development Kit</i> ) de Nokia para la serie 40, que no soporta conexión a través de servicios web, la aplicación funcionó perfectamente en las demás herramientas software.
<b>Nokia Series 40 SDK 6th Edition v1.0.</b>	
<b>Nokia N97 SDK v1.0.</b>	
<b>Sun Java Wireless Toolkit v2.5.2</b>	
<b>NAVEGADORES WEB</b>	
<b>Internet Explorer v7.0.</b>	Un navegador interpreta el código en el que está escrita la página web y lo presenta en pantalla, permitiendo al usuario interactuar con su contenido y navegar hacia otros lugares de la red mediante enlaces o hipervínculos. La aplicación web desarrollada para HORUS fue probada exitosamente en estos tres navegadores; las imágenes, botones, tablas, campos de
<b>Google Chrome v6.0.</b>	

<sup>33</sup> Un kit de desarrollo de software es generalmente un conjunto de herramientas de desarrollo que le permite a un programador crear aplicaciones para un sistema concreto.

<b>Mozilla Firefox v3.6.8.</b>	texto y tablas que conforman las interfaces graficas no sufrieron modificaciones; además ninguno de los navegadores solicitó la instalación de algún componente adicional para visualizar las interfaces.
<b>DISPOSITIVOS MOVILES</b>	
<b>Nokia N93.</b>	De acuerdo las características funcionales de este dispositivo y gracias a que estaba disponible dentro de los equipos del Departamento de Telemática de la Universidad del Cauca, la aplicación desarrollada para clientes móviles de HORUS, se probó en un ambiente real, obteniendo resultados similares a los obtenidos cuando se utilizó el emulador del SDK de Nokia para el dispositivo de referencia N97. La única diferencia radicó en los tiempos de conexión entre la aplicación móvil y los servidores definidos en la arquitectura (servidor central y servidor de mapas de <i>Google</i> ); ya que en el caso de la simulación estos tiempos dependían de la velocidad de procesamiento del computador donde estaba instalado el emulador y en la prueba real dependían de la velocidad de conexión ofrecida por la red.

**Tabla A.38.** Herramientas software utilizadas durante la fase de implementación y pruebas de HORUS.

# ANEXO B

## MANUAL DE USUARIO

### B.1 ASPECTOS GENERALES

En esta sección del documento se explican las instrucciones de uso que son comunes a todos los usuarios de la aplicación web de HORUS.

#### B.1.1 Inicio de Sesión

Cualquier usuario que pretenda utilizar los servicios de HORUS, debe estar previamente registrado en el sistema y su información personal debe estar almacenada en la base de datos del mismo. Bajo estas condiciones el sistema podrá validar el ingreso de los usuarios a través de la interfaz web diseñada para este fin. La figura B.1 muestra la interfaz web inicio de sesión.



Figura B.1. Interfaz de HORUS para validar el ingreso de los usuarios al sistema.

Para ingresar al sistema debe seleccionar el tipo de usuario, ingresar el nombre de la cuenta de usuario, la contraseña de la cuenta y presionar el botón “Ingresar”. Si los datos ingresados son correctos, el sistema automáticamente desplegara en pantalla la interfaz correspondiente al tipo de usuario seleccionado; en caso contrario el sistema desplegara una alerta informando que se ha cometido un error durante el proceso de validación. Las figuras B.2 (a, b, c) muestran las alertas que notifican los posibles errores que se pueden cometer.



**Figura B.2 (a).** Error generado al seleccionar incorrectamente el tipo de usuario.



**Figura B.3 (b).** Error generado al ingresar incorrectamente la contraseña de la cuenta.



**Figura B.4 (c).** Error generado al ingresar incorrectamente el nombre de la cuenta.

### B.1.2 Consultar Posición Geográfica

Este servicio permite observar en un mapa la posición geográfica de un vehículo y las respectivas coordenadas. La figura B.3 muestra el icono asociado a esta funcionalidad.



**Figura B.5.** Icono para consultar posición geográfica.

Una vez presionado el icono, el sistema automáticamente ubica un marcador en el mapa indicando la posición geográfica del vehículo. Para conocer las coordenadas y la velocidad a la que se desplaza el vehículo, se debe hacer clic sobre el marcador y el sistema automáticamente desplegara los datos a través de una ventana emergente sobre el mapa. La figura B.6 muestra un ejemplo de esta función.



Figura B.6. Servicio de HORUS para consultar posición geográfica.

### B.1.3 Graficar Recorrido

Este servicio permite observar en un mapa el recorrido realizado por un vehículo hasta una hora antes de que sea iniciado el servicio. La figura B.7 muestra el icono asociado a esta funcionalidad.



Figura B.7. Icono para graficar recorrido.

Una vez presionado el icono, el sistema automáticamente despliega en el mapa una línea indicando el recorrido realizado por el vehículo, donde se señala con marcadores el inicio y el fin del recorrido. Para conocer información asociada al recorrido, se debe hacer clic sobre el marcador que indica el fin del recorrido y el sistema automáticamente desplegará los datos relacionados con la distancia, la velocidad y el tiempo; a través de una ventana emergente sobre el mapa. La figura B.8 muestra un ejemplo de esta función.



Figura B.8. Servicio de HORUS para graficar el recorrido realizado por un vehículo.

### B.1.4 Consultar Información asociada al vehículo

Este servicio permite consultar las características de un vehículo, la información relacionada con el propietario y la información relacionada con el conductor. La figura B.9 muestra el icono asociado a esta funcionalidad.



Figura B.9. Icono para consultar información asociada al vehículo.

Una vez presionado el icono, el sistema automáticamente despliega en pantalla una interfaz donde se distribuyen en tablas y asociados a su respectiva imagen, los datos del vehículo, su propietario y su conductor. La figura B.10 muestra un ejemplo de esta interfaz.

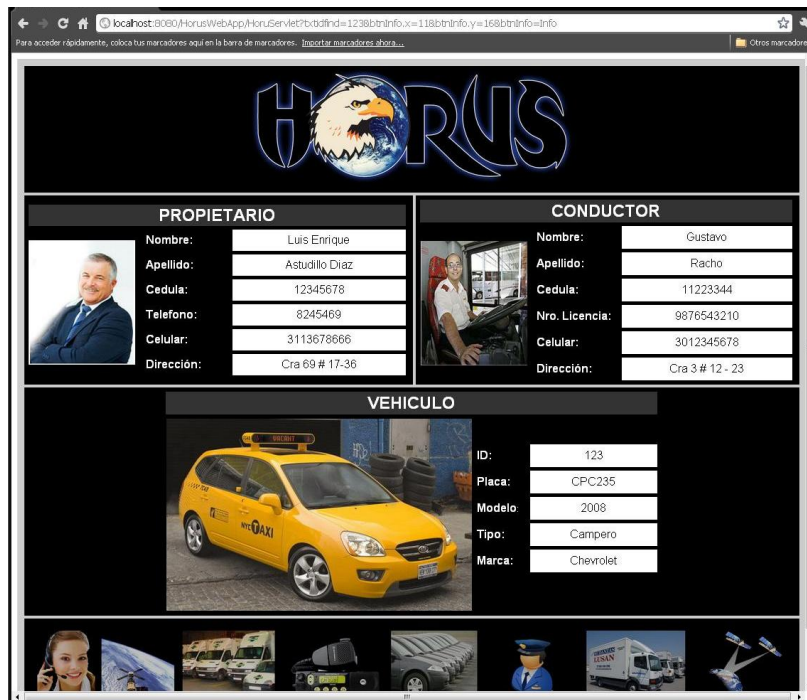


Figura B.10. Interfaz web para consultar la información asociada a un vehículo.

## B.2 INSTRUCCIONES DE USO PARA ADMINISTRADORES

Una vez los datos de usuario hayan sido validados y el ingreso al sistema haya sido autorizado, el sistema despliega en pantalla la interfaz de administración para vehículos asociados a HORUS, en esta interfaz se encuentran los iconos descritos en







Figura B.13. Botón para ingresar a la interfaz de gestión de usuarios.

La figura B.14 muestra la interfaz web que despliega el sistema cuando se presiona el botón “Gestión Usuarios”.

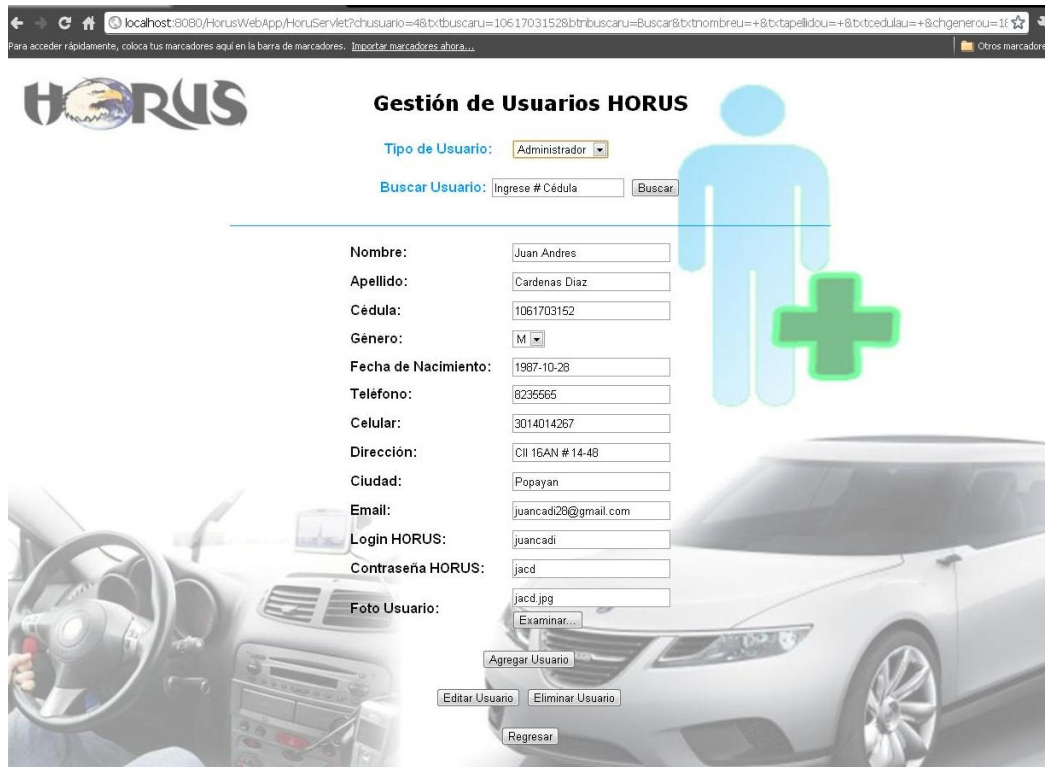
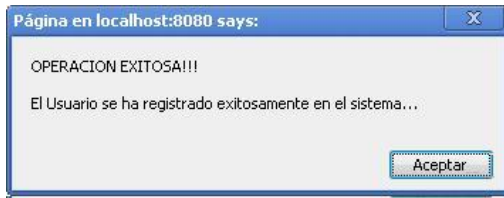


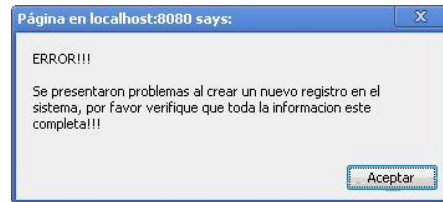
Figura B.14. Interfaz de HORUS para gestión de usuarios.

### B.2.1 Crear cuentas de usuario

Para crear una cuenta de usuario, se debe seleccionar el tipo de usuario en el menú de la interfaz web, llenar todos los campos de texto con la información que solicita el sistema y finalmente presionar el botón “Agregar Usuario”. Las figuras B.14 (a, b) muestran la alerta que notifica el éxito del proceso y la alerta que notifica un error en caso de que se presente.



**Figura B.15 (a).** Notificación de éxito durante el proceso de agregar usuarios.



**Figura B.15 (b).** Notificación de errores durante el proceso de agregar usuarios.

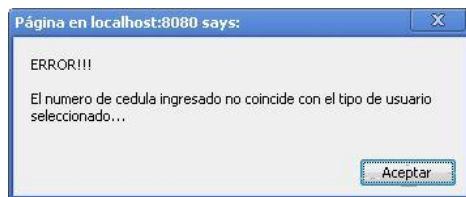
## B.2.2 Modificar cuentas de usuario

Para modificar una cuenta de usuario, se debe seleccionar el tipo de usuario en el menú de la interfaz web, ingresar el número de identificación del usuario y presionar el botón “Buscar”. La figura B.16 muestra la imagen de los elementos de la interfaz web que permiten realizar la búsqueda de usuarios.



**Figura B.16.** Elementos implementados para facilitar la búsqueda de usuarios.

En caso de que el usuario no se encuentre registrado en el sistema se despliega en pantalla una alerta notificándolo. Las figuras B.17 (a, b) muestran las alertas desplegadas.

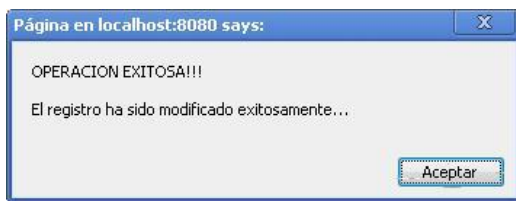


**Figura B.17(a).** Tipo de usuario incorrecto.

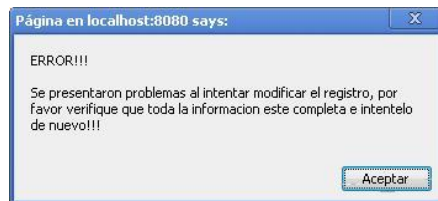


**Figura B.17 (b).** Usuario no encontrado.

Una vez encontrado el usuario, el sistema despliega en pantalla los campos de texto con la información del usuario. De esta manera el administrador puede modificar los datos que considere necesarios y presionar el botón “Modificar Usuario”, para que el registro de la base de datos sea actualizado. Las figuras B.18 (a, b) muestran la alerta que notifica el éxito del proceso y la alerta que notifica un error en caso de que se presente.



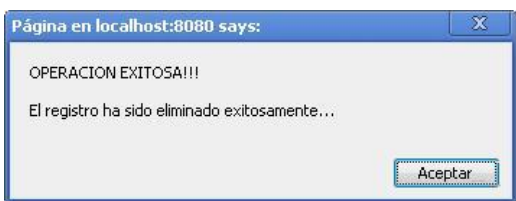
**Figura B.18 (a).** Notificación de éxito en el proceso de modificar usuarios.



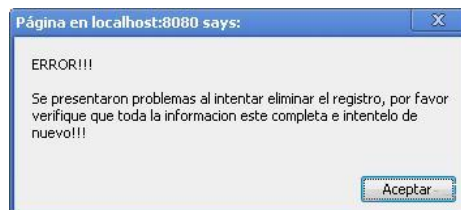
**Figura B.18 (b).** Notificación de error en el proceso de modificar usuarios.

### B.2.3 Eliminar cuentas de usuario

Para eliminar una cuenta de usuario, se debe seleccionar el tipo de usuario en el menú de la interfaz web, ingresar el número de identificación del usuario y presionar el botón “Buscar”. Una vez el sistema haya encontrado el usuario, finalmente se debe presionar el botón “Eliminar Usuario”. Las figuras B.19 (a, b) muestran la alerta que notifica el éxito del proceso y la alerta que notifica un error en caso de que se presente.



**Figura B.19 (a).** Notificación de éxito en el proceso de eliminar usuarios.



**Figura B.19 (b).** Notificación de error en el proceso de eliminar usuarios.

## B.3 INSTRUCCIONES DE USO PARA CLIENTES

Los usuarios tipo cliente tienen la posibilidad de utilizar los servicios de HORUS a través de la aplicación web o a través de la aplicación móvil. Según sea el caso, consultar las instrucciones de uso que aparecen en las secciones B.3.1 y B.3.2 respectivamente.

### B.3.1 Cliente Web

Una vez los datos de usuario hayan sido validados y el ingreso al sistema haya sido autorizado, el sistema relaciona automáticamente al cliente con el ID del vehículo asociado a su cuenta y despliega en pantalla la interfaz web para clientes. Para utilizar los servicios de HORUS se debe presionar cualquiera de los iconos descritos en las secciones B.2.1, B.2.2 y B.2.3. La figura B.20 muestra la interfaz web para clientes.

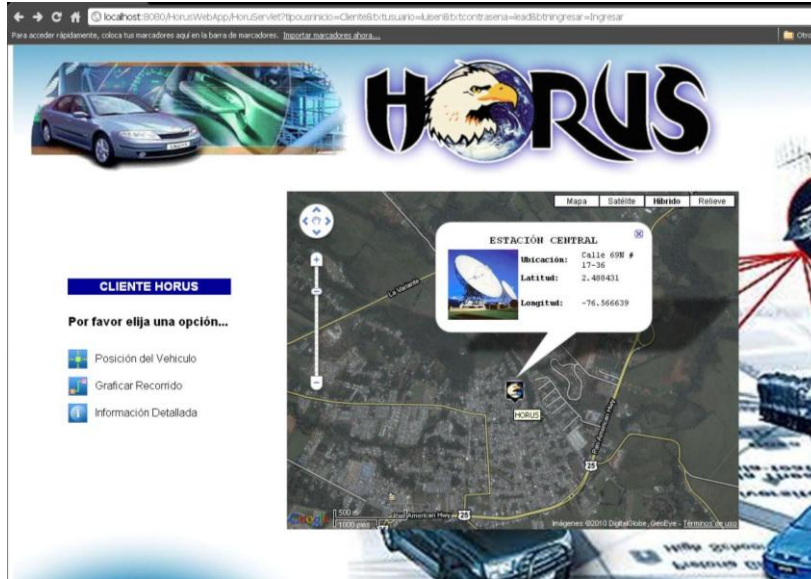


Figura B.20. Interfaz web para clientes.

### B.3.2 Cliente Móvil

La aplicación de HORUS para clientes móviles, permite realizar el monitoreo de un vehículo desde un teléfono celular con capacidades de acceso a la red, CLDC v1.0 o superior y MIDP v2.0 o superior. La figura B.21 muestra la descripción de las partes más relevantes de un dispositivo móvil para la aplicación de HORUS.



Figura B.21. Partes de un teléfono celular.

### B.3.2.1 Validación de Usuario

Segundos después de que se haya desplegado la pantalla de bienvenida, el sistema muestra en la pantalla del dispositivo móvil la interfaz para validar el ingreso de usuarios tipo cliente. Para ingresar los datos de validación se deben seguir los siguientes pasos:

- a. Seleccionar el campo de texto “Login” mediante la tecla inferior de navegación.
- b. Ingresar el nombre de la cuenta de usuario a través del teclado alfanumérico.
- c. Seleccionar el campo de texto “Password” mediante la tecla inferior de navegación.
- d. Ingresar la contraseña de la cuenta de usuario a través del teclado alfanumérico.
- e. Presionar la tecla programable del lado izquierdo (Ingresar).

En cualquier momento se puede presionar la tecla programable del lado derecho (Salir) para cerrar la aplicación.

Una vez realizado este procedimiento, el sistema despliega en pantalla una animación indicando que los datos están siendo enviados hasta el servidor central para realizar su respectiva validación. Las figuras B.22 (a, b) muestran el proceso de validación de usuarios a través de los dispositivos móviles.



Figura B.22 (a). Validación de usuarios.



Figura B.22 (b). Conexión con el servidor.

El sistema notifica al usuario si el proceso de validación resulta exitoso. La figura B.23 muestra la alarma implementada en el dispositivo móvil para indicar el caso de éxito.



**Figura B.23.** Proceso de validación exitoso.

Considerando que la aplicación móvil fue desarrollada única y exclusivamente para los usuarios tipo cliente, el sistema notificara un error en el proceso de validación en los siguientes casos:

- El *login* o el *password* fueron ingresados incorrectamente.
- Los datos a validar corresponden a un tipo de usuario diferente al tipo cliente.
- Los datos a validar no aparecen registrados en el sistema.

Según sea el caso. La figura B.24 muestra las imágenes de las alarmas desplegadas para notificar errores durante el proceso de validación.



**Figura B.24.** Alarmas utilizadas para reportar errores durante el proceso de validación desde la aplicación móvil.

### B.3.2.2 Consultar posición geográfica

Una vez el cliente ingresa al sistema, se despliegan en pantalla las opciones para seleccionar el tipo de mapa en el que se pretende visualizar la posición o el recorrido

de los vehículos y las opciones de geolocalización que ofrece la aplicación de HORUS para dispositivos móviles. Para consultar la posición geográfica de un vehículo se deben seguir los siguientes pasos:

- a. Seleccionar el tipo de mapa utilizando las teclas de navegación superior e inferior.
- b. Presionar la tecla de acción sobre la opción seleccionada.
- c. Seleccionar la opción “Consultar Posición” utilizando la tecla inferior o superior de navegación y presionar la tecla de acción.
- d. Presionar la tecla programable del lado izquierdo (Ver Mapa).

En cualquier momento se puede presionar la tecla programable del lado derecho (Salir) para cerrar la aplicación.

Una vez realizado este procedimiento, el sistema despliega en pantalla la imagen de un mapa donde se señala mediante un marcador la posición del vehículo. Las figuras B.25 (a, b, c) muestran el proceso a realizar para consultar la posición de un vehículo a través de un dispositivo móvil.



**Figura B.25 (a).** Selección del tipo de mapa.



**Figura B.25 (b).** Consultar posición.



**Figura B.25 (c).** Mapa con la posición del vehículo.

Las teclas de navegación del dispositivo móvil permite desplazar la imagen del mapa en los cuatro sentidos (arriba, abajo, izquierda y derecha) y la tecla programable del lado derecho permite desplegar un menú de opciones donde se puede incrementar o disminuir el nivel de acercamiento del mapa a través de las opciones “Zoom (+)” y “Zoom (-)”. La figura B.26 muestra las opciones desplegadas en el menú de la tecla programable del lado derecho.



**Figura B.26.** Menú de opciones.

En caso de que no se pueda establecer la conexión entre el dispositivo móvil y el servidor de mapas de *Google*, el sistema despliega una alarma notificando el error en el proceso de conexión. La figura B.27 muestra la alarma implementada por HORUS para notificar este tipo de errores.



**Figura B.27.** Error en la conexión con el servidor de mapas.

### B.3.2.3 Graficar recorrido

Una vez el cliente ingresa al sistema, se despliegan en pantalla las opciones para seleccionar el tipo de mapa en el que se pretende visualizar la posición o el recorrido de los vehículos y las opciones de geolocalización que ofrece la aplicación de HORUS para dispositivos móviles. Para consultar la posición geográfica de un vehículo se deben seguir los siguientes pasos:



- a. Seleccionar el tipo de mapa utilizando las teclas de navegación superior e inferior.
- b. Presionar la tecla de acción sobre la opción seleccionada.
- c. Seleccionar la opción “Visualizar Recorrido” utilizando la tecla inferior o superior de navegación y presionar la tecla de acción.
- d. Presionar la tecla programable del lado izquierdo (Ver Mapa).

En cualquier momento se puede presionar la tecla programable del lado derecho (Salir) para cerrar la aplicación.

Una vez realizado este procedimiento, el sistema despliega en pantalla la imagen de un mapa donde visualiza mediante una línea el recorrido del vehículo y se señala mediante un marcador el inicio y el fin del recorrido. Las figuras B.28 (a, b) muestran el proceso a realizar para visualizar el recorrido de un vehículo a través de un dispositivo móvil una vez haya sido seleccionado el tipo de mapa como se muestra en la figura B.25 (a).

Las teclas de navegación del dispositivo móvil permite desplazar la imagen del mapa en los cuatro sentidos (arriba, abajo, izquierda y derecha) y la tecla programable del lado derecho permite desplegar un menú de opciones donde se puede incrementar o disminuir el nivel de acercamiento del mapa a través de las opciones “Zoom (+)” y “Zoom (-)”. La figura B.26 muestra las opciones desplegadas en el menú de la tecla programable del lado derecho.

En caso de que no se pueda establecer la conexión entre el dispositivo móvil y el servidor de mapas de *Google*, el sistema despliega una alarma notificando el error en el proceso de conexión. La figura B.27 muestra la alarma implementada por HORUS para notificar este tipo de errores.



Figura B.28 (a). Visualizar Recorrido.



Figura B.28 (b). Mapa con el recorrido del vehículo.

### B.3.2.4 Consultar información asociada al vehículo

Una vez el sistema haya desplegado la imagen del mapa, es posible consultar la información asociada al vehículo presionando la tecla programable del lado izquierdo (Información Vehículo). La figura B.29 (a, b) muestra en la pantalla del dispositivo móvil la información asociada al vehículo y su conductor.



Figura B.29(a). Información del vehículo.



Figura B.29 (b). Información del Conductor.

## B.4 INSTRUCCIONES DE USO PARA OPERADORES

Una vez los datos de usuario hayan sido validados y el ingreso al sistema haya sido autorizado, el sistema despliega en pantalla la interfaz para gestión y despacho de vehículos asociados a HORUS, en esta interfaz se implementan los iconos descritos en las secciones B.2.1, B.2.2 y B.2.3, un botón adicional para ingresar a la interfaz de gestión de tareas, un botón para despachar vehículos y un botón para enviar mensajes de texto a las unidades móviles de los vehículos, entre otros elementos. La figura B.30 muestra la interfaz web que HORUS despliega para los operadores del sistema y posteriormente se describen las instrucciones según las funcionalidades que brinda esta interfaz.

En la figura B.30 se puede visualizar un icono en el mapa que señala la posición de la central de despacho HORUS; además, se puede observar a través de una ventana emergente sus coordenadas, la dirección asociada y una imagen alusiva al lugar.

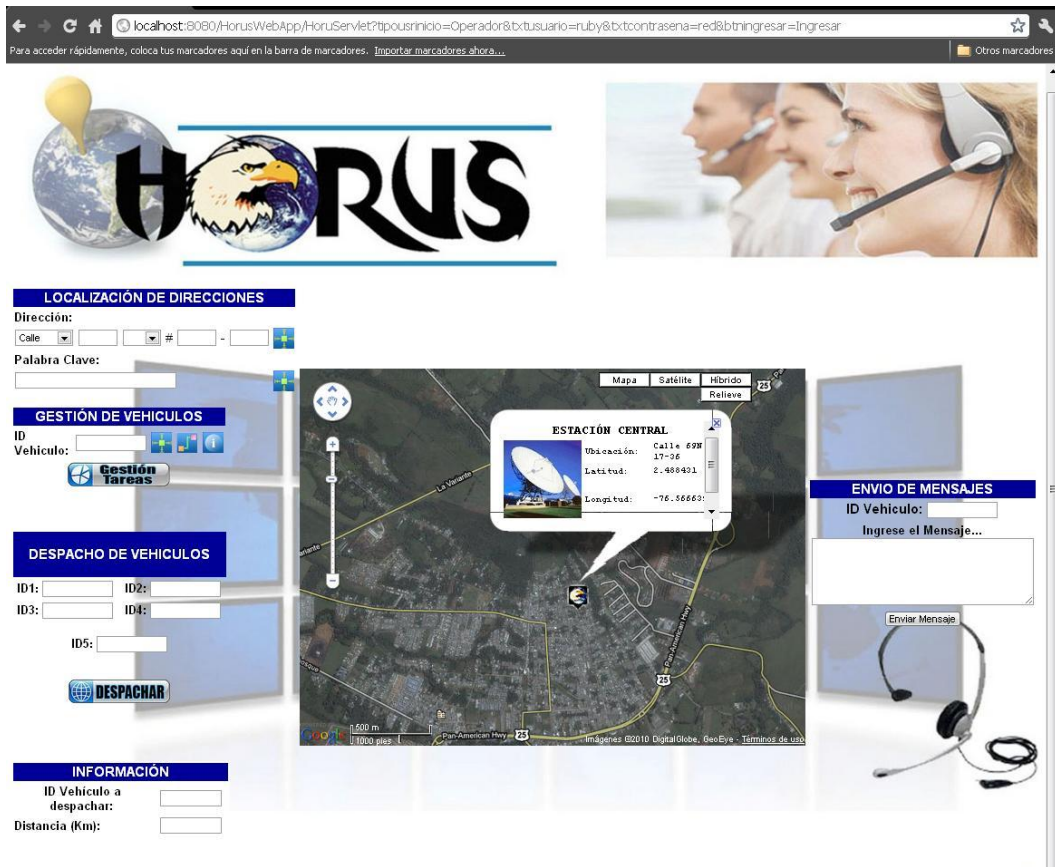


Figura B.30. Interfaz web para operadores de HORUS.

#### B.4.1 Localizar direcciones

Una dirección en el mapa se puede localizar de dos formas diferentes. La primera consiste en ingresar la dirección según el formato tradicional, especificando la calle o la carrera y los números que la identifican; y la segunda consiste en ingresar una palabra clave asociada al lugar que se pretende localizar. Posteriormente se debe presionar el icono de la figura B.2.1 y el sistema ubicara automáticamente un icono en el mapa según las coordenadas del lugar a localizar. La figura B.31 muestra los elementos utilizados en el proceso de localizar direcciones.









Figura B.31. Elementos involucrados en la localización de direcciones.













La figura B.32 muestra un ejemplo de localización de direcciones a través de la interfaz web de HORUS para operadores.



**Figura B.32.** Localización de direcciones.

Los iconos utilizados para señalar la posición de un lugar en el mapa varían dependiendo del lugar a localizar. La tabla B.1 muestra los iconos implementados en HORUS y una breve descripción.

ICONO	DESCRIPCIÓN
	Hace referencia a la central de HORUS.
	Hace referencia a vehículos particulares.
	Hace referencia a taxis.
	Indica que se ha activado el botón de pánico.
	Hace referencia a estaciones de gasolina.
	Hace referencia a la dirección de una vivienda particular.
	Hace referencia a colegios.
	Hace referencia a universidades.
	Hace referencia a restaurantes
	Hace referencia a penitenciarías.
	Hace referencia a estaciones de policía.
	Hace referencia a bares.

	Hace referencia a venta de comida rápida.
	Hace referencia a cafeterías.
	Hace referencia a veterinarias.
	Hace referencia a hospitales.
	Hace referencia a gimnasios.
	Hace referencia a centros de salud.
	Hace referencia a consultorios médicos
	Hace referencia a discotecas.
	Hace referencia a karaokes.
	Hace referencia a cementerios.
	Hace referencia a iglesias.
	Hace referencia a balnearios.

**Tabla B.1.** Iconos implementados por HORUS para identificar lugares.

#### B.4.2 Despachar vehículos

La figura B.32 muestra los elementos de la interfaz web involucrados en el despacho de vehículos. Para despachar vehículos se deben seguir los siguientes pasos:

- a. Localizar en el mapa la dirección que representa el origen de una solicitud o el destino del despacho.
- b. Ingresar mediante los campos de texto de la interfaz web, los identificadores de los primeros cinco vehículos que se reporten como disponibles.
- c. Presionar el botón “Despachar”

Una vez realizado este procedimiento, el sistema desplegará en pantalla el ID del vehículo que se encuentra más cercano al lugar donde se originó la solicitud de despacho y la distancia aproximada que existe entre estos. Además se crea un nuevo registro en la base de datos asignando una tarea al vehículo despachado.



**Figura B.33.** Elementos involucrados en el proceso de despacho vehicular.

### B.4.3 Enviar mensajes a unidades móviles

Para enviar un mensaje de máximo 32 caracteres a una unidad móvil, se debe ingresar el ID de la unidad móvil a la que va dirigido el mensaje y se debe escribir el mensaje en el campo de texto designado para tal fin. Finalmente se debe presionar el botón “Enviar Mensaje”. La figura B.34 muestra los elementos de la interfaz involucrados en el proceso de envío de mensajes.



**Figura B.34.** Elementos involucrados en el proceso de envío de mensajes,

### B.4.4 Gestión de tareas

Una vez se presione el botón “Gestionar Tareas”, el sistema despliega en pantalla la interfaz para gestionar las tareas asignadas a los vehículos. La figura B.35 muestra la interfaz web donde se pueden observar elementos para buscar tareas según la fecha en la que fueron asignadas, eliminarlas o modificarlas mediante su identificador (ID); o asignar nuevas tareas a un vehículo.

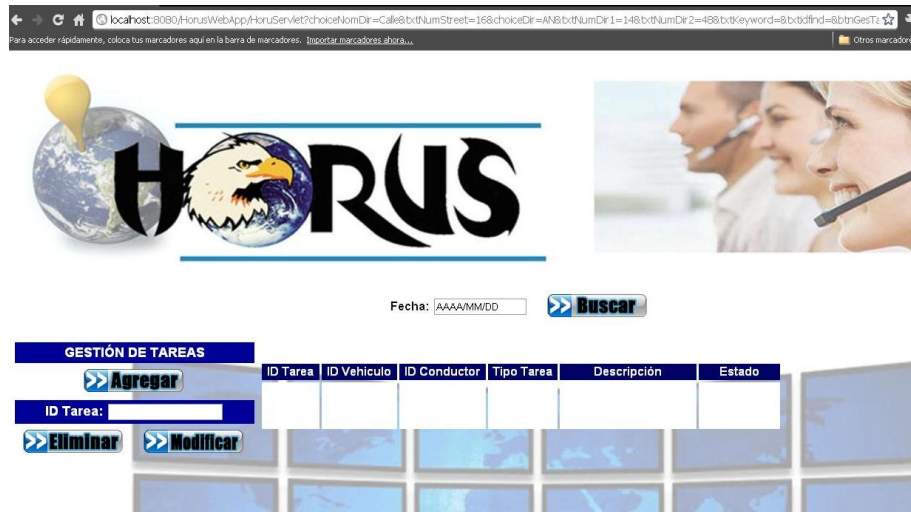


Figura B.35. Interfaz para gestión de tareas.

#### B.4.4.1 Buscar tareas

Para realizar la búsqueda de tareas previamente registradas, se debe ingresar la fecha en la que fue asignada, según el formato que indica el campo de texto “Fecha” (AAAA/MM/DD) y presionar el botón “Buscar”. Una vez realizada esta acción el sistema desplegará mediante una tabla la información asociada a todas las tareas que hayan sido asignadas en la fecha ingresada. La figura B.36 muestra los elementos involucrados en el proceso de búsqueda de tareas.



Figura B.36. Elementos involucrados en la búsqueda de tareas.

En caso de que la fecha ingresada no coincida con ninguno de los registros de la base de datos del sistema, se desplegará en pantalla un alerta notificando el error. La figura B.37 muestra la alerta implementada en este caso.



Figura B.37. Error en búsqueda de tareas asignadas.

#### B.4.4.2 Agregar Tareas

Para asignar una nueva tarea, se debe presionar el botón “Agregar” de la interfaz principal para gestión de tareas mostrada en la figura B.35. Automáticamente el sistema desplegará una tabla con campos de texto para ingresar los datos de identificación del vehículo al que se pretende asignar la tarea, el número de identificación del conductor, el tipo de tarea asignada, una breve descripción y además se debe fijar el estado de la tarea (pendiente o atendido).

Una vez ingresados los datos se debe presionar el botón “Agregar” y el sistema creará un nuevo registro en la base de datos con la información de la nueva tarea asignada. La figura B.38 muestra los elementos de la interfaz web involucrados en el proceso de agregar tareas.



**Figura B.38.** Elementos involucrados durante el proceso de asignación de tareas desde la aplicación web.

Las figuras B.39 (a, b) muestran la alerta que notifica el éxito o el error en caso de que se presente, durante el proceso de asignación de tareas.



**Figura B.39 (a).** Proceso de asignación de tareas exitoso.



**Figura B.39 (b).** Error durante el proceso de asignación de tareas.

#### B.4.4.3 Modificar Tareas

Para modificar una tarea, se debe ingresar el número que identifica el registro de una tarea en el campo de texto “ID Tarea” y presionar el botón “Modificar” de la interfaz principal para gestión de tareas mostrada en la figura B.35. En caso de no conocer el identificador de la tarea, el sistema brinda la posibilidad de buscar el registro asociado mediante a partir de la fecha en la que fue asignada, véase sección B.3.4.1 del anexo B.

Una vez realizado este procedimiento, el sistema desplegará en pantalla una tabla con

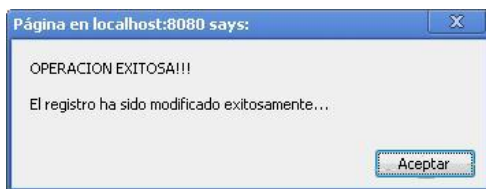


la información correspondiente a la tarea, cuyos campos podrán ser modificados según sea el caso; el único campo de la tabla que no es modificable es el correspondiente al identificador de tarea. Una vez realizada las modificaciones, se debe presionar el botón “Actualizar” par que los cambios sean almacenados en el registro. La figura B.40 muestra los elementos involucrados en el procedo de modificar tareas.

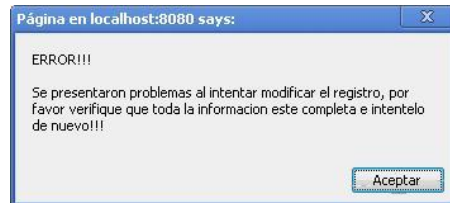
El sistema desplegara en pantalla una alerta notificando el éxito o alguna irregularidad presentada durante el proceso de modificación de registros. Las figuras B.41 (a, b) muestran las alertas implementadas para tal fin.



**Figura B.40.** Elementos involucrados en el proceso de modificar tareas.



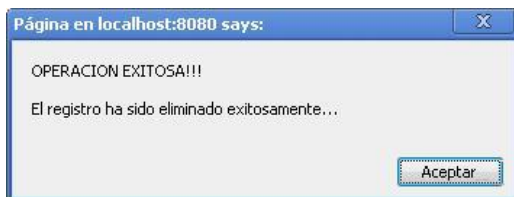
**Figura B.41 (a).** Proceso de modificación de tareas exitoso.



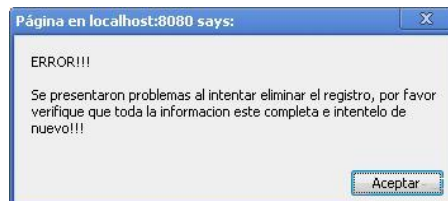
**Figura B.41 (b).** Error durante el proceso de modificación de tareas.

#### B.4.4.4 Eliminar Tareas

Para eliminar una tarea, se debe ingresar el número que identifica el registro de una tarea en el campo de texto “ID Tarea” y presionar el botón “Eliminar” de la interfaz principal para gestión de tareas mostrada en la figura B.35. En caso de no conocer el identificador de la tarea, el sistema brinda la posibilidad de buscar el registro asociado mediante a partir de la fecha en la que fue asignada, véase sección B.3.4.1 del anexo B.



**Figura B.42 (a).** Notificación de éxito durante el proceso de eliminar tareas.



**Figura B.42 (b).** Notificación de error durante el proceso de eliminar tareas.

El sistema desplegará en pantalla una alerta notificando el éxito o alguna irregularidad presentada durante el proceso de eliminación de registros. Las figuras B.42 (a, b) muestran las alertas implementadas para tal fin.

## B.5 INSTRUCCIONES DE USO PARA CONDUCTORES

HORUS presenta una interfaz de usuario que permite al conductor iniciar sesión en el sistema, recibir mensajes de texto provenientes de la central y enviar una señal de pánico hacia la central en caso de presentarse una situación de emergencia. La figura B.43 muestra la interfaz física de la unidad móvil.

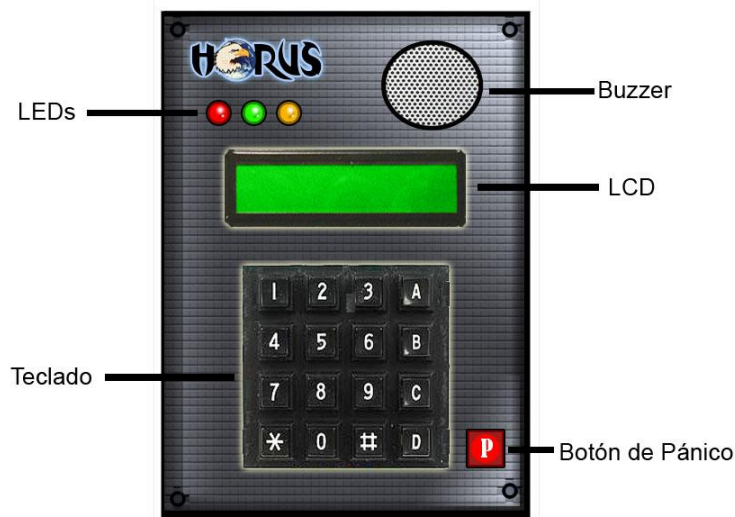


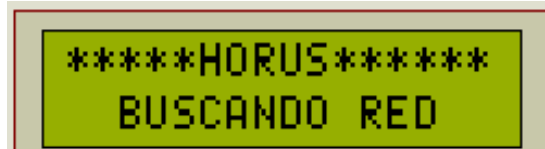
Figura B.43. Interfaz física de la unidad móvil.

- **LEDs:** Diodos emisores de luz utilizados como indicadores de estado de la interfaz física de la unidad móvil: Rojo: encendido; Amarillo: canal ocupado; Verde: transmitiendo.
- **Teclado:** Teclado alfanumérico utilizado para interactuar con el menú de la interfaz. Dependiendo del número de veces consecutivas que una tecla sea presionada, se puede introducir una letra o un número, del modo en que se hace en un teléfono móvil.
- **Buzzer:** Zumbador mecánico que emite una señal sonora durante 5 segundos para notificar al conductor del vehículo que ha llegado un mensaje de texto enviado desde la central.
- **LCD:** Despliegue de cristal líquido de 2 líneas y 32 caracteres que permite la visualización de la información.
- **Botón de Pánico:** Pulsador que permite enviar una señal de pánico desde el vehículo hacia la estación de control con el fin de reportar situaciones de emergencia.

## B.5.1 Funciones de la Interfaz Física de la Unidad Móvil

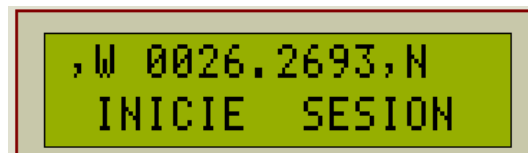
### B.5.1.1 Inicio de Sesión

Al inicializar la interfaz física de la unidad móvil, el vehículo no se encuentra asociado a la red, por lo que automáticamente aparece un mensaje “buscando red”. En este estado, la unidad móvil solicita a la estación central el ingreso a la red. La figura B.44 muestra el mensaje “buscando red” al inicializar la unidad móvil.



**Figura B.44.** Mensaje “Buscando Red” de inicio de unidad móvil.

Cuando la central permite a la unidad móvil el ingreso a la red, se inicia el proceso de posicionamiento del vehículo. Las coordenadas son desplegadas en la primera línea de la pantalla, desplazándose de derecha a izquierda, y enviadas hacia la central. La segunda línea de la pantalla solicita al conductor el inicio de sesión. La figura B.45 muestra la pantalla de inicio de sesión de la unidad móvil.



**Figura B.45.** Inicio de sesión.

Para realizar el inicio de sesión, se hace uso del teclado alfanumérico. Se debe presionar la tecla “★” para empezar a introducir la identificación de usuario. La tabla B.2 muestra los caracteres posibles a introducir con cada tecla del teclado alfanumérico.

Tecla	CARACTERES/FUNCION				
1	.	,	:	;	1
2	A	B	C	2	
3	D	E	F	3	
4	G	H	I	4	
5	J	K	L	5	
6	M	N	O	6	
7	P	Q	R	S	7
8	T	U	V	8	
9	W	X	Y	Z	9
0	-	0			
★	OK				

#	Cancelar
A	Anterior
B	Cerrar Sesión
C	Borrar
D	Borrar Todo

**Tabla B.2.** Asignación de caracteres del teclado alfanumérico.

Se debe buscar la tecla con la letra o número que se desea introducir y presionarla cuantas veces sea necesario para que el caracter aparezca. Se debe presionar la tecla en intervalos menores a medio segundo, ya que pasado este tiempo se grabará el último caracter que aparezca y el cursor pasará a la siguiente posición. Al presionar otra tecla, el caracter que aparece en pantalla se guarda y el cursor avanza a la siguiente posición, mostrando la letra correspondiente a la tecla que se acaba de presionar.

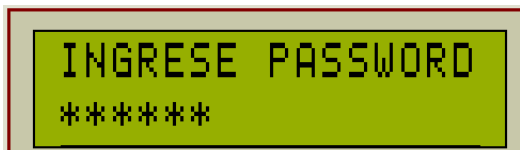
Si se desea borrar el último caracter introducido se debe presionar la tecla “C”, y si se desea borrar toda la cadena introducida, se debe presionar la tecla “D”.

El identificador de usuario puede ser de hasta 16 caracteres, de tipo alfanumérico. Después de introducir el ID de usuario se debe presionar la tecla “★” para pasar a la pantalla de ingreso de contraseña de usuario. La figura B.46 muestra la pantalla de ingreso de ID y el registro del identificador de usuario de nombre “Gustavo”.



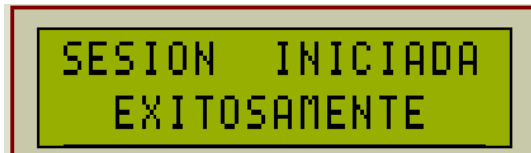
**Figura B.46.** Pantalla de ingreso de identificador de usuario.

Al confirmar la identificación se usuario, aparece la pantalla de ingreso de contraseña de usuario. La contraseña de usuario puede ser de hasta 16, y sólo puede ser de tipo numérico. La forma de ingreso de la contraseña se realiza de la misma forma en que se hace para el ingreso del identificador de usuario. La figura B.47 muestra la pantalla de ingreso de la contraseña de usuario.



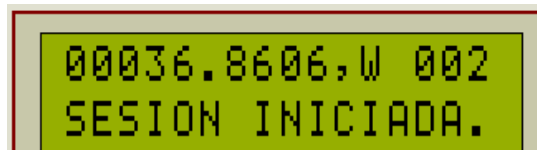
**Figura B.47.** Pantalla de ingreso de contraseña de usuario.

Después de confirmar el ingreso de la contraseña de usuario a través de la tecla “★”, la información de identificador y contraseña de usuario es enviada hasta la estación de control y validada en el servidor central. Si el conductor está registrado tiene permitido acceder al sistema, se muestra una pantalla de inicio de sesión exitosa durante 5 segundos. En caso contrario, aparece un mensaje de error y se solicitan nuevamente los datos del usuario. La figura B.48 muestra la pantalla de inicio de sesión de usuario exitosa.



**Figura B.48.** Pantalla de inicio de sesión exitosa.

Después de que el usuario es validado en el sistema y se ha desplegado la pantalla de inicio de sesión exitosa, se muestra la pantalla por defecto del sistema, donde la primera línea corresponde a las coordenadas geográficas del vehículo, las cuales son desplazadas de derecha a izquierda, y en la segunda línea se muestra el mensaje “Sesión Iniciada”. La figura B.49 muestra la pantalla por defecto del sistema después de la validación de usuario.



**Figura B.49.** Pantalla por defecto del sistema con identificación de usuario validada.

### **B.5.1.2 Visualización de Mensaje de Texto**

Cuando se recibe un mensaje de texto enviado por la central, el *buzzer* emite una señal sonora durante 5 segundos. La pantalla por defecto del sistema cambia automáticamente a una pantalla donde en la primera línea se muestra la notificación “Nuevo Mensaje” y en la segunda se muestra el mensaje, desplazándolo de derecha a izquierda. La figura B.50 muestra la pantalla de nuevo mensaje de texto.



**Figura B.50.** Pantalla de nuevo mensaje de texto.

### B.5.1.3 Cierre de Sesión

En cualquier momento el conductor puede cerrar la sesión. Para ello, se debe presionar la tecla “B”, lo que da paso al mensaje en pantalla “Presione ★ para cerrar sesión”. Este mensaje se desplaza de derecha a izquierda. En la figura B.51 se muestra la pantalla de cierre de sesión.

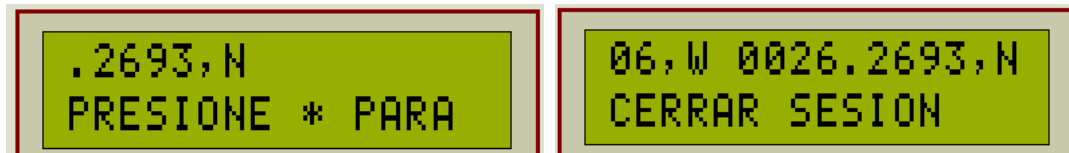


Figura B.51. Pantalla de cierre se sesión.

Si la tecla “★” no se presiona en un lapso de 20 segundos, la pantalla por defecto del sistema vuelve a aparecer. Si tecla “★” es presionada, aparece un mensaje de confirmación de cierre de sesión, donde se muestran las opciones para confirmar o cancelar el cierre de sesión desplazándolas de derecha a izquierda. Si se desea cerrar sesión de debe pulsar la tecla “★” nuevamente. La figura B.52 muestra la pantalla de confirmación de cierre de sesión.

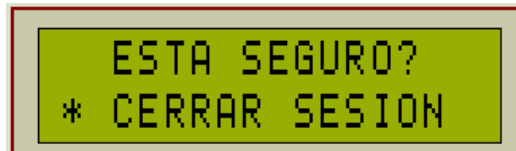


Figura B.52. Pantalla de confirmación de cierre de sesión.

Si se desea cancelar el cierre de sesión de debe presionar la tecla “#”. En la figura B.53 se muestra la pantalla de cancelación de cierre de sesión.

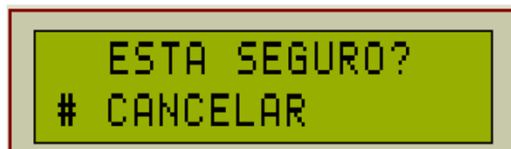


Figura B.53. Pantalla de cancelación de cierre de sesión.

### B.5.1.4 Envío de la Señal de Pánico

En caso de presentarse una situación de emergencia debe presionarse el botón de pánico. Al hacer esto, el dispositivo envía la señal de emergencia hacia la central de despacho para notificar la eventualidad.

## REFERENCIAS BIBLIOGRÁFICAS

1. L. Martinez, M. Urios. *Tecnologías de Localización y Posicionamiento para Servicios Basados en Localización (LBS)*. [En línea] 2006. [Citado el: 03 de 02 de 2010.] <http://www.coit.es/publicaciones/bit/bit154/68-70.pdf>.
2. Oficina de Coordinación Nacional de Posicionamiento, Navegación, y Cronometría por Satélite. *Sistema de Posicionamiento Global al Servicio del Mundo*. [En línea] 2010. [Citado el: 05 de 02 de 2010.] <http://www.gps.gov/spanish.html>.
3. Dispatch Magazine On-Line. *Computer-Aided Dispatch Software Resources*. [En línea] 2010. [Citado el: 12 de 04 de 2010.] <http://www.911dispatch.com/info/cad/index.html>.
4. Introducción al mundo de los estándares web. “¿Cómo funciona Internet?” [En línea] 2010. [Citado el: 07 de 07 de 2010.] <http://mosaic.uoc.edu/ac/le/es/m1/ud2/index.html>.
5. Servlets. *Servlet Simple*. [En línea] 2010. [Citado el: 12 de 08 de 2010.] <http://manuales.dgsca.unam.mx/webdina/servlets.htm>.
6. Alvarez, M. *La tecnología Java para la creación de páginas web con programación en el servidor*. [En línea] 2010. [Citado el: 08 de 08 de 2010.] <http://www.desarrolloweb.com/articulos/831.php>.
7. Oracle. *JavaServer Pages Technology*. [En línea] 2010. [Citado el: 08 de 08 de 2010.] <http://java.sun.com/products/jsp/>.
8. Script, Diferencias entre Java y JavaScript. *Diferencias entre Java y JavaScript*. [En línea] 2010. [Citado el: 08 de 08 de 2010.] <http://www.desarrolloweb.com/articulos/492.php>.
9. Web Developers Notes. *JavaScript*. [En línea] 2010. [Citado el: 08 de 08 de 2010.] [http://www.webdevelopersnotes.com/basics/languages\\_on\\_the\\_internet.php3](http://www.webdevelopersnotes.com/basics/languages_on_the_internet.php3).
10. Google API Javascript Mapa V3. *Tutorial*. [En línea] [Citado el: 15 de 06 de 2010.] <http://code.google.com/intl/es-419/apis/maps/documentation/javascript/>.
11. Data Access Object (DAO) Code Generator. *DaoGen Manual*. [En línea] [Citado el: 02 de 06 de 2010.] <http://titaniclinux.net/daogen/>.

12. Tecnologías de Internet. *J2SE - JDBC: Java DataBase Connectivity*. [En línea] [Citado el: 12 de 05 de 2010.] [http://www.aulambra.com/java/java\\_jdbc.asp](http://www.aulambra.com/java/java_jdbc.asp) .
13. Oracle. *JDBC Overview*. [En línea] 2010. [Citado el: 15 de 05 de 2010.] <http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136101.html> .
14. GiovyNet – Java Electronics Solutions. *GiovyNet Driver*. [En línea] [Citado el: 19 de 06 de 2010.] <http://java.giovy.net.com/Giovy.net>.
15. Alegsa. *Definición de RS-232*. [En línea] 1998. [Citado el: 08 de 14 de 2010.] <http://www.alegsa.com.ar/Dic/RS-232.php>.
16. W3C – World Wide Web Consortium. *Guía breve de servicios web*. [En línea] [Citado el: 02 de 08 de 2010.] <http://www.w3c.es/divulgacion/guiasbreves/ServiciosWeb>
17. Alegsa. *Definición de Servicios Web*. [En línea] [Citado el: 02 de 08 de 2010.] <http://www.alegsa.com.ar/Dic/servicio%20web.php>
18. Jappit.com. *MidMaps: Google Maps Java ME library*. [En línea] [Citado el: 02 de 08 de 2010.] <http://www.jappit.com/blog/midmaps-google-maps-java-me-library/> .
19. API de Google Static Maps. *Guía para programadores del API de Google Static Maps*. [En línea] [Citado el: 03 de 08 de 2010.] <http://code.google.com/intl/es-419/apis/maps/documentation/staticmaps/>
20. ObjectXP. *JSMS Overview*. [En línea] [Citado el: 12 de 06 de 2010.] <http://www.objectxp.com/en/products/jsms/>
21. Academic dictionaries and encyclopedias . *Short message service center*. [En línea] [Citado el: 18 de 07 de 2010.] <http://en.academic.ru/dic.nsf/enwiki/1229654>.
22. The Math Forum. *Deriving the Haversine Formula*. [En línea] [Citado el: 02 de 09 de 2010.] <http://mathforum.org/library/drmath/view/51879.html>