



REPÚBLICA DEL ECUADOR

Escuela Politécnica Nacional

" E S C I E N T I A H O M I N I S S A L U S "

La versión digital de esta tesis está protegida por la Ley de Derechos de Autor del Ecuador.

Los derechos de autor han sido entregados a la "ESCUELA POLITÉCNICA NACIONAL" bajo el libre consentimiento del (los) autor(es).

Al consultar esta tesis deberá acatar con las disposiciones de la Ley y las siguientes condiciones de uso:

- Cualquier uso que haga de estos documentos o imágenes deben ser sólo para efectos de investigación o estudio académico, y usted no puede ponerlos a disposición de otra persona.
- Usted deberá reconocer el derecho del autor a ser identificado y citado como el autor de esta tesis.
- No se podrá obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo original.

El Libre Acceso a la información, promueve el reconocimiento de la originalidad de las ideas de los demás, respetando las normas de presentación y de citación de autores con el fin de no incurrir en actos ilegítimos de copiar y hacer pasar como propias las creaciones de terceras personas.

Respeto hacia sí mismo y hacia los demás.

ESCUELA POLITÉCNICA NACIONAL

**FACULTAD DE INGENIERÍA ELÉCTRICA Y
ELECTRÓNICA**

**DESARROLLO DE UN SISTEMA PROTOTIPO DE CONSULTA DE
PARQUEADEROS LIBRES**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
ELECTRÓNICA Y CONTROL**

ENRIQUE XAVIER PORRAS TORRES
xavept@gmail.com

DIRECTOR: ING. JAVIER PRADO
javier.prado@updatecom.com.ec

CO-DIRECTOR: DR. LUIS CORRALES
luisco5049@yahoo.com

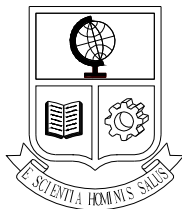
Quito, Junio 2011

DECLARACIÓN

Yo Enrique Xavier Porras Torres, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Enrique Xavier Porras Torres



ESCUELA POLITECNICA NACIONAL
"E SCIENTIA HOMINIS SALUS"

La versión digital de esta tesis está protegida por la Ley de Derechos del Autor del Ecuador.

Los derechos de autor han sido entregados a la "ESCUELA POLITÉCNICA NACIONAL" bajo el libre consentimiento del autor.

Al consultar esta tesis deberá acatar las disposiciones de la Ley y de las siguientes condiciones de uso:

- Cualquier uso que haga de estos documentos o imágenes deben ser sólo para efectos de investigación o estudio académico, y usted no puede ponerlos a disposición de otra persona.
- Usted deberá reconocer el derecho del autor a ser identificado y citado como el autor de esta tesis.
- No se podrá obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo original.

El libre Acceso a la información, promueve el reconocimiento de la originalidad de las ideas de los demás, respetando las normas de presentación y de citación de autores con el fin de no incurrir en actos ilegítimos de copiar y hacer pasar como propias las creaciones de terceras personas.

Respeto hacia sí mismo y hacia los demás

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por Enrique Xavier Porras Torres, bajo nuestra supervisión.

Ing. Javier Prado
DIRECTOR DEL PROYECTO

Dr. Luis Corrales
CO-DIRECTOR DEL PROYECTO

AGRADECIMIENTO

Deseo brindar mi más sincero y cordial agradecimiento al Ing. Javier Prado, al Dr. Luis Corrales, a todos quienes conforman UPDATECOM, a la Escuela Politécnica Nacional, mi hermosa familia y a mis amigos, “los guambras”, quienes estuvieron junto a mi durante toda esta enriquecedora travesía, a todos ustedes, gracias por su tiempo, el apoyo, la confianza y la oportunidad depositada en mi persona. Que Dios los bendiga.

Xavier

DEDICATORIA

Quiero dedicar este trabajo a mis padres, Manuelito y Carlotita, y a toda mi familia por haber sido fuente de apoyo e inspiración, pero de forma muy especial a mis hermanos Eduardo y Lucía, por haber apostado a este proyecto, mi vida, y por ser más que un ejemplo de vida mi razón de ser de cada día. Que Dios los cobije siempre de bendiciones.

Xavier

ÍNDICE DE CONTENIDOS

DECLARACIÓN.....	i
CERTIFICACIÓN.....	iii
AGRADECIMIENTO.....	iv
DEDICATORIA.....	v
CONTENIDO.....	vi
ÍNDICE DE FIGURAS.....	ix
ÍNDICE DE TABLAS.....	xii
RESUMEN.....	xiv
PRESENTACIÓN.....	xv

CONTENIDO

CAPÍTULO 1

MARCO TEÓRICO

1.1. PROBLEMÁTICA DEL SISTEMA PROTOTIPO.....	1
1.2. SOLUCIÓN PROPUESTA.....	1
1.3. HERRAMIENTAS DE SOLUCIÓN.....	3
1.3.1. GSM (GLOBAL SYSTEM FOR MOBILE COMMUNICATION)..	3
1.3.1.1. Arquitectura de red GSM.....	3
1.3.2. SISTEMA CELULAR.....	5
1.3.3. SMS (SHORT MESSAGE SERVICE).....	5
1.3.3.1. Servicio SMS.....	5
1.3.3.2. Acceso a los servicios SMS.....	6
1.3.4. EMULADOR DE BASE DE DATOS LOCAL.....	7
1.3.5. ADMINISTRADOR DE LA BASE DE DATOS: PHPMYADMIN..	7
1.3.6. BASE DE DATOS MYSQL.....	8
1.3.7. WEB HOSTING SERVICE.....	9
1.3.7.1. Tipos de alojamiento en un web hosting.....	10
1.3.7.2. Dominios.....	12

1.3.8. SISTEMAS DE COORDENADAS.....	12
1.3.8.1. Coordenadas Proyectadas.....	12
1.3.8.2. Coordenadas Cartesianas.....	13
1.3.8.3. Coordenadas Geográficas.....	14
1.3.9. BLUETOOTH.....	15
1.3.9.1. Clasificación Bluetooth.....	16

CAPÍTULO 2

DESCRIPCIÓN DEL HARDWARE PARA EL DESARROLLO DEL PROTOTIPO

2.1. INTRODUCCIÓN.....	17
2.2. PDA (ASISTENTE DIGITAL PERSONAL).....	17
2.2.1. SISTEMAS OPERATIVOS PARA EQUIPOS EMBEBIDOS.....	18
2.3. SISTEMA DE POSICIONAMIENTO GLOBAL “GPS”.....	19
2.3.1. FUNCIONAMIENTO DE UN GPS.....	19
2.3.2. CAUSAS DE ERROR DE PRECISIÓN DEL GPS.....	21
2.3.3. SELECCIÓN DEL HARDWARE GPS.....	23
2.3.4. TRAMA GPS SEGÚN NORMA NMEA 0183.....	26
2.3.4.1. Trama RMC.....	27
2.4. MODEM GSM.....	28
2.4.1. Módems para circuito impreso (CI).....	29
2.4.2. Módems para PC.....	29
2.4.2.1. Internos.....	29
2.4.2.2. Externos.....	30
2.4.3. Tipos de conexión del módem.....	30

CAPÍTULO 3

DESARROLLO DEL SOFTWARE

3.1. INTRODUCCIÓN.....	33
3.2. VISUAL STUDIO .NET.....	33
3.3. DESARROLLO DE LA BASE DE DATOS EN MYSQL.....	35

3.4. DESARROLLO DE LA APLICACIÓN CLIENTE.....	38
3.4.1. ALGORITMO DE LA APLICACIÓN CLIENTE.....	39
3.4.2. SIMULACIÓN DE LA INFORMACIÓN PARA LA BASE DE DATOS.....	40
3.4.3. SEGURIDAD DE LA APLICACIÓN CLIENTE.....	41
3.4.4. OPCIONES DE LA APLICACIÓN CLIENTE.....	41
3.4.5. ALARMAS DE DISPONIBILIDAD.....	45
3.5. DESARROLLO Y CONFIGURACIÓN DE LA APLICACIÓN SERVIDOR.....	47
3.5.1. INTERFAZ CON MÓDEMS.....	48
3.5.2. INTERFAZ CON MÓDEMS GSM.....	49
3.5.3. COMANDOS AT.....	49
3.5.3.1. Listado de mensajes.....	50
3.5.4. ALGORITMO DE LA APLICACIÓN SERVIDOR.....	50
3.5.5. SEGURIDAD DE LA APLICACIÓN SERVIDOR.....	51
3.5.6. PROCESAMIENTO DE LA TRAMA SMS.....	54
3.5.6.1. Cálculo de la distancia en grandes círculos.....	55
3.5.6.2. Simulación del cálculo de distancias.....	57
3.5.7. OPCIONES DE LA APLICACIÓN SERVIDOR.....	59
3.6. DESARROLLO Y CONFIGURACIÓN DE LA APLICACIÓN PDA.....	62
3.6.1. ALGORITMO DE LA APLICACIÓN PDA.....	62
3.6.2. TRAMA PARA ENCAPSULADO DE DATOS.....	65
3.6.3. PROCESO DE CONSULTA (ENVÍO Y RECEPCIÓN DE DATOS).....	66
3.6.3.1. Mapas para dispositivos móviles.....	68
3.6.4. INFORMACIÓN REFERENTE AL SISTEMA PICO Y PLACA...	71
3.6.5. FACTORES DE CONSIDERACIÓN.....	79

CAPÍTULO 4

PRUEBAS Y RESULTADOS DEL PROTOTIPO

4.1. INTRODUCCIÓN.....	81
------------------------	----

4.2. PRUEBAS PARA LA APLICACIÓN CLIENTE.....	81
4.3. PRUEBAS PARA LA APLICACIÓN SERVIDOR.....	82
4.4. PRUEBAS PARA LA APLICACIÓN PDA.....	83

CAPÍTULO 5 CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones.....	84
5.2 Recomendaciones.....	85
REFERENCIAS BIBLIOGRÁFICAS.....	86

ANEXO A	HTC modelo HD2
ANEXO B	Dell GPS Navigation System NAVTEQ Bluetooth Receiver BT-309
ANEXO C	Modem ZTE-MF100 Datacard FAQ
ANEXO D	Código Fuente Sistema PROPARK

ÍNDICE DE FIGURAS

CAPÍTULO 1 MARCO TEÓRICO

Figura 1-1	Estructura del sistema de consulta PROPARK.....	2
Figura 1-2	Arquitectura GSM a nivel de aplicación.....	4
Figura 1-3	Parte de un sistema celular.....	5
Figura 1-4	Servicio SMS.....	6
Figura 1-5	Panel de control XAMPP.....	7
Figura 1-6	Panel de control phpMyAdmin.....	8
Figura 1-7	Coordenadas Proyectadas.....	13
Figura 1-8	Coordenadas Cartesianas.....	14
Figura 1-9	Representación de Coordenadas Geográficas de un punto...	15

CAPÍTULO 2

DESCRIPCIÓN DEL HARDWARE PARA EL DESARROLLO DEL PROTOTIPO

Figura 2-1	PDA: HTC HD2.....	19
Figura 2-2	(DOP) Dilución de la precisión geométrica.....	20
Figura 2-3	La señal del satélite “1” viaja menos a través de la atmósfera, dando como resultado menor distorsión. El satélite “2” está menos en el horizonte, sin embargo, el resultado es una distorsión atmosférica significativa.....	21
Figura 2-4	La distorsión en señales de radio son corregidas por la combinación de señales satelitales (1 y 2) con información de corrección transmitida vía estaciones terrestres GPS.....	22
Figura 2-5	Errores de Efemérides.....	22
Figura 2-6	Error por efecto múltiple.....	23
Figura 2-7	Adquisición de datos del GPS/Bluetooth mediante el HiperTerminal.....	24
Figura 2-8.	Emparejamiento GPS/Bluetooth y PDA (HTC HD2).....	25
Figura 2-9	Confirmación de emparejamiento.....	25
Figura 2-10	GPS-Bluetooth DELL NAVTEQ ON BOARD Modelo BT-309....	26
Figura 2-11	Utilización de un módem GSM desde un sistema digital.....	28
Figura 2-12	Módem GSM ZT-MF100.....	31

CAPÍTULO 3

DESARROLLO DEL SOFTWARE

Figura 3-1	Microsoft Visual Studio 2008.....	33
Figura 3-2	Emulador PDA WM 6.5 Profesional.....	34
Figura 3-3	Especificaciones plan web hosting.....	35
Figura 3-4	Panel de control: www.xavierporras.com	38
Figura 3-5	Pantalla phpMyAdmin: Base de datos subida en la web.....	39
Figura 3-6	Algoritmo de funcionamiento: Aplicación cliente.....	37
Figura 3-7	Pantalla de inicio a la aplicación de escritorio para los parqueaderos.....	40

Figura 3-8	Pantalla de ingreso a la aplicación cliente.....	41
Figura 3-9	Pantalla menú principal.....	42
Figura 3-10	Pantalla opción: Ingreso de plaza.....	43
Figura 3-11	Pantalla opción: Eliminar plaza.....	43
Figura 3-12	Pantalla opción: Ver usuarios.....	44
Figura 3-13	Pantalla de salida.....	44
Figura 3-14	Pantalla: Alarma plazas libres por terminarse.....	45
Figura 3-15	Pantalla: Alarma parqueadero casi desocupado.....	45
Figura 3-16	Pantalla: Alarma parqueadero lleno.....	46
Figura 3-17	Pantalla: Alarma parqueadero vacío.....	46
Figura 3-18	Mensaje de validación si la placa ya está registrada.....	47
Figura 3-19	Módem GSM conectado al servidor.....	47
Figura 3-20	Prueba de funcionamiento del módem.....	49
Figura 3-21	Algoritmo de funcionamiento: Aplicación servidor.....	51
Figura 3-22	Pantalla de inicio a la aplicación de escritorio para el servidor..	52
Figura 3-23	Pantalla de ingreso a la aplicación servidor: Identificación de usuario.....	53
Figura 3-24	Software propietario de comprobación del módem GSM.....	53
Figura 3-25	Interface para aplicación de escritorio del servidor.....	54
Figura 3-26	Comparación del geoide y el esferoide.....	56
Figura 3-27	Ejemplo de conversión de formato de coordenadas.....	57
Figura 3-28	Simulación del cálculo de la distancia realizado por PROPARK.	58
Figura 3-29	Cálculo de distancia por www.tutiempo.net	58
Figura 3-30	Pantalla: Programa servidor con direcciones habilitadas.....	59
Figura 3-31	Pantalla: Ingreso de parqueadero a la base de datos.....	60
Figura 3-32	Pantalla: Eliminación de parqueadero de la base de datos.....	60
Figura 3-33	Pantalla: Información total de los parqueaderos.....	61
Figura 3-34	Pantalla: Cambio de clave de acceso a las respectivas aplicaciones “cliente” del sistema.....	61
Figura 3-35	Mensaje de error del servidor si los permisos de ingreso no están habilitados.....	62
Figura 3-36	Algoritmo: Aplicación PDA.....	63
Figura 3-37	Pantalla del menú de Inicio de Windows Mobile.....	64

Figura 3-38	Pantalla de Inicio PROPARK.....	64
Figura 3-39	Menú principal aplicación PDA.....	65
Figura 3-40	Petición de parqueadero más cercano.....	66
Figura 3-41	Mensaje de confirmación de petición realizada.....	66
Figura 3-42	SMS enviado por el servidor PROPARK.....	67
Figura 3-43	Pantalla de visualización de la ubicación del automotor.....	67
Figura 3-44	Pantalla: Información general del sistema pico y placa.....	71
Figura 3-45	Tipos de parqueaderos según el sistema pico y placa.....	72
Figura 3-46	Parqueadero de borde.....	73
Figura 3-47	Ejemplo opción parqueadero de borde Carapungo.....	74
Figura 3-48	Parqueaderos Intermodales.....	75
Figura 3-49	Ejemplo parqueadero intermodal: Trolebús la Y.....	75
Figura 3-50	Parqueaderos Internos.....	76
Figura 3-51	Ejemplo parqueadero interno: Cumandá.....	76
Figura 3-52	Información Zona Azul.....	77
Figura 3-53	Configuración de una tecla de la PDA.....	78
Figura 3-54	Mensaje de despedida de la aplicación.....	78
Figura 3-55	Respuesta de la aplicación si el GPS no está listo.....	79
Figura 3-56	Mensaje de la aplicación si el GPS no está encendido.....	80
Figura 3-57	Mensaje de no disponibilidad del GPS luego de 3 intentos.....	80

CAPÍTULO 4

PRUEBAS Y RESULTADOS DEL PROTOTIPO

Figura 4-1	Aplicación cliente instalada.....	81
Figura 4-2	Aplicación Servidor instalada.....	82
Figura 4-3	Aplicación PDA instalada.....	83

ÍNDICE DE TABLAS

CAPÍTULO 1

MARCO TEÓRICO

Tabla 1-1	Clasificación Bluetooth según los tipos de energía y rango.....	16
-----------	---	----

CAPÍTULO 2

DESCRIPCIÓN DEL HARDWARE PARA EL DESARROLLO DEL PROTOTIPO

Tabla 2-1	Desglose trama \$GPRMC.....	27
-----------	-----------------------------	----

CAPÍTULO 3

INTEGRACIÓN DE DISPOSITIVOS Y DESARROLLO DEL SOFTWARE

Tabla 3-1	Tabla “parqueaderos”.....	36
Tabla 3-2	Tabla “registros”.....	37
Tabla 3-3	Tabla “servidorPropark”.....	37
Tabla 3-4	Formato de trama para encapsulado de datos.....	65

RESUMEN

El presente proyecto de titulación detalla el diseño de un prototipo de sistema para consulta de plazas libres de parqueaderos, el cual permite integrar y complementar los procesos de automatización que se implementen en los mismos, ayudando de esta manera a optimizar las medidas tomadas por el cabildo regente y el sector privado en la parte de aparcamiento.

Con este objetivo, se desarrolla un conjunto de aplicaciones que facilitan compartir información referente a disponibilidad de puestos en los recintos registrados al prototipo entre los administradores de estos centros y el dueño del automotor. Esta transacción la puede realizar el usuario mediante un PDA que consulta una base de datos con la ayuda de mensajes de texto.

Como resultado se consigue un mejor ordenamiento vehicular y por ende que el usuario final del sistema pueda llegar rápidamente, y con certeza de excedencia, al estacionamiento más cercano a su vehículo.

PRESENTACIÓN

Debido al crecimiento del parque automotor, el problema del tráfico es una realidad que se enfrenta día a día y con ella la falta de espacio donde estacionar dicho conglomerado vehicular; razón por la cual, el sector público y privado se ha visto en la necesidad de adoptar un conjunto de acciones dentro de las cuales se contemplan medidas frente al problema de falta de parqueaderos y mejoramiento en la eficiencia de los ya existentes.

Estas medidas buscan básicamente contribuir con soluciones prácticas a los problemas de movilidad ocasionados por los conductores que ocupan de forma no adecuada el espacio público como estacionamiento permanente y particular de sus vehículos.

La aceptación por parte de los usuarios respecto a estas medidas fue positiva, pero con el paso del tiempo se pudo apreciar las debilidades de las mismas, ya que, al existir una alta demanda de vehículos en las entradas a los estacionamientos (públicos y privados), estos se congestionaban indeseablemente, factor crítico al momento de movilizarse por las zonas aledañas.

Aunque ya de por sí, el ingresar a dichos recintos produce un estancamiento en la movilidad vehicular, el problema se agudiza cuando los usuarios luego de una larga espera al momento de llegar a la entrada se encontraban con la noticia de que el parqueadero ya estaba lleno.

El sistema PROPARK se presenta como una propuesta a este problema, facilitando una herramienta de consulta, a cualquier usuario que cuente con un PDA (SO Windows Mobile), referente al parqueadero más cercano a su ubicación y su disponibilidad.

Se desea aclarar que lejos de la realidad está pensar que la creación de un sistema de consulta motive a los ciudadanos a sacar sus vehículos y saturar las estrechas calles de la ciudad, sino más bien, el proyecto busca que los vehículos sean mejor estacionados y no ocupen un indeseado lugar en las vías, dejando más espacio libre al tránsito de personas, vehículos y bicicletas.

Es así que el sistema de consulta propuesto busca ser un instrumento de desarrollo urbano para la recuperación del espacio público, el impulso del desarrollo económico local y el direccionamiento de la movilidad en la ciudad.

En el capítulo 1, se especifica un marco teórico el cual contempla de manera general la problemática y el planteamiento de la solución por parte del sistema PROPARK referente al aparcamiento vehicular, además revisa las principales características de los protocolos de comunicación GSM, SMS y Bluetooth así como de ciertas herramientas usadas en el desarrollo de la aplicación como lo son: el emulador de bases de datos XAMPP, el administrador de bases de datos PHPMyAdmin, la base de datos MySQL, el Web Hosting Service, y los principales sistemas de coordenadas para representar la tierra.

En el capítulo 2, se analiza y se justifica la selección del hardware utilizado para la implementación de la solución propuesta.

En el capítulo 3, se muestra el desarrollo del software y la interfaz gráfica de las diferentes aplicaciones que componen el sistema prototipo.

En el capítulo 4, se determinan los resultados a las pruebas hechas al sistema prototipo PROPARK.

En el capítulo 5, se presentan las conclusiones obtenidas y recomendaciones sugeridas una vez concluido el proyecto.

Finalmente, se adjuntan los anexos que pormenorizan las características técnicas de los dispositivos: PDA, GPS/Bluetooth, módem GSM; y se concluye anexando el código fuente de las aplicaciones desarrolladas.

CAPÍTULO 1

MARCO TEÓRICO

1.1 PROBLEMÁTICA DEL SISTEMA PROTOTIPO

Actualmente, la impresión de que las vías de la ciudad han colapsado es permanente. Un análisis funcional de este problema indica que simplemente existe insuficiencia de vías para permitir un eficiente desplazamiento de los vehículos en la ciudad, problema que se traduce en congestión, el mismo que llega a una situación de colapso el momento de buscar un lugar en donde poder parquearse.

Frente a este último punto, el sector público y privado adoptó un conjunto de medidas que contemplan la creación de nuevos parqueaderos y la mejora en la eficiencia de los existentes. A pesar de estas medidas, el ingresar los vehículos a dichos recintos producen un estancamiento en la movilidad vehicular y el problema se agudiza cuando los usuarios, luego de una larga espera, al momento de llegar a la entrada se encuentran con la noticia de que el parqueadero ya está lleno.

1.2 SOLUCIÓN PROPUESTA

Frente a esta coyuntura se propone el diseño de un sistema prototipo de consulta de parqueaderos libres, el cual busca facilitar la información de disponibilidad de los parqueaderos que se registren a este sistema, permitiendo conocer el número de plazas libres y como resultado del mismo descongestionar las vías para el normal desenvolvimiento de los dueños de los automotores en las calles.

De manera general el sistema de consulta propuesto, denominado de aquí en adelante PROPARK (Ver Figura 1-1), basado en la estructura computacional cliente-servidor¹, cuenta con tres aplicaciones denominadas:

- 1) Cliente
- 2) Servidor
- 3) PDA

¹ Modelo que permite descentralizar el procesamiento y recursos, sobre todo, de cada uno de los servicios y de la visualización de la Interfaz Gráfica de Usuario. [14]

- 1) Cliente.- Esta aplicación almacena y actualiza dentro de una base de datos ubicada en la web, el número de plazas libres y ocupadas de los parqueaderos que se encuentren registrados a este sistema, así como la placa, fecha y hora de ingreso de los automotores que accedan en los mismos.
- 2) Servidor.- El registro de los parqueaderos al sistema se lo realiza con ayuda de esta aplicación, la cual además recepta y valida la información de los mensajes de texto que arriban a un módem GSM conectado al computador donde corre esta aplicación.
- 3) PDA.- La aplicación instalada en un PDA realiza la adquisición de las coordenadas, de donde se desea hacer la consulta, a un GPS que se encuentra dentro de un vehículo mediante el estándar Bluetooth. Esta información es procesada para posteriormente ser enviada en un mensaje de texto al módem GSM instalado en el servidor.

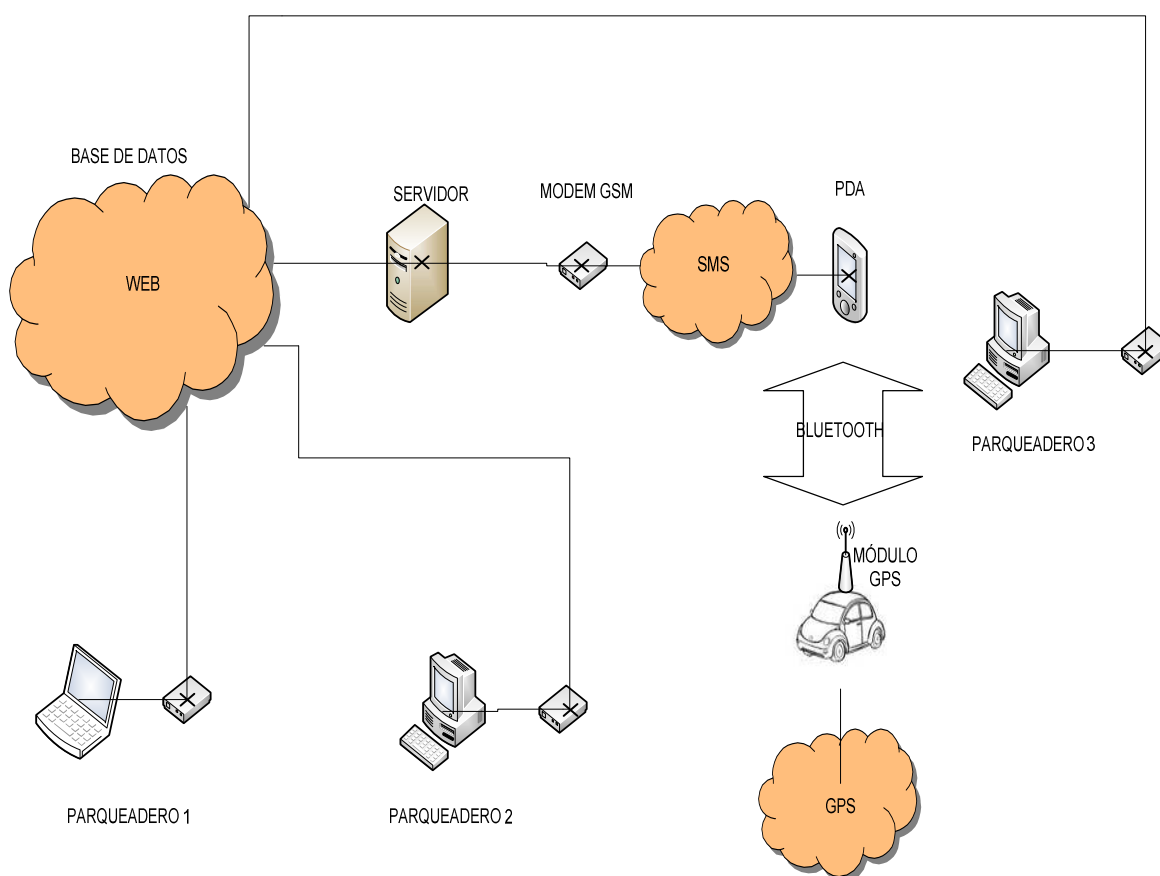


Figura 1-1. Estructura del sistema de consulta PROPARK

NOTA: La validación de la información en la aplicación servidor, en primera instancia, deberá constatar la validez de los datos del SMS enviado por el PDA para luego determinar mediante un algoritmo el parqueadero más cercano a las coordenadas recibidas y posteriormente enviar de vuelta al PDA la dirección y la disponibilidad del mismo mediante un mensaje texto.

1.3 HERRAMIENTAS DE SOLUCIÓN

1.3.1 GSM (GLOBAL SYSTEM FOR MOBILE COMMUNICATION) [1]

Es un sistema de comunicaciones móviles que ofrece los siguientes servicios:

- Transmisión/recepción de voz
- Transmisión/recepción de datos
- Envío/recepción de mensajes cortos (Short Message Service, SMS)

1.3.1.1 Arquitectura de red GSM

En la Figura 1-2 se muestra de manera resumida la arquitectura de la red GSM. Esta arquitectura es más compleja y dispone de más elementos que los presentados.

El objetivo es describir el servicio SMS a nivel de aplicación, sin entrar en demasiados detalles de la red subyacente.

La arquitectura GSM está constituida por tres partes:

1. *Subsistema Radio (RSS, Radio SubSystem)*, Cubre la comunicación entre las estaciones móviles (MS, Mobile Station) y las estaciones base (BTS, Base Transceiver Station). El interfaz radio entre ellas se denomina Um.
2. *El subsistema de estaciones base (BSS)*, Controla las comunicaciones de radio del terminal móvil. Está en contacto con el subsistema de red NSS (Network Subsystem) conectando al móvil con otros usuarios, está constituido por los siguientes elementos:

- BTS (Base Transceiver Station): Gestiona las comunicaciones de las estaciones móviles. Constituido por emisor, receptor y antena. Procesa los canales de radio (interfaz Um).
- BSC (Base Station Controller): Realiza el control de las BTSs. Este subsistema hace de interfaz entre la parte de radio y la parte de red.

3. *Subsistema de red y conmutación (NSS, Network and Switching Subsystem).*-

Realiza la conmutación, gestión de la movilidad, interconexión con otras redes y control del sistema. Esta es la parte más compleja, siendo sus elementos fundamentales los siguientes:

- MSC (Mobile Services Switching Center), es el centro de conmutación entre otras muchas funciones.
- GMSC (Gateway Mobile Services Switching Center): Realiza la conexión con otras redes
- Bases de datos: Subsistema que se encarga de controlar diversas funciones del equipo móvil. Sus principales elementos son:
 - HLR (Home Location Register)
 - VLR (Visitor Location Register)
 - EIR (Equipment Identity Register)

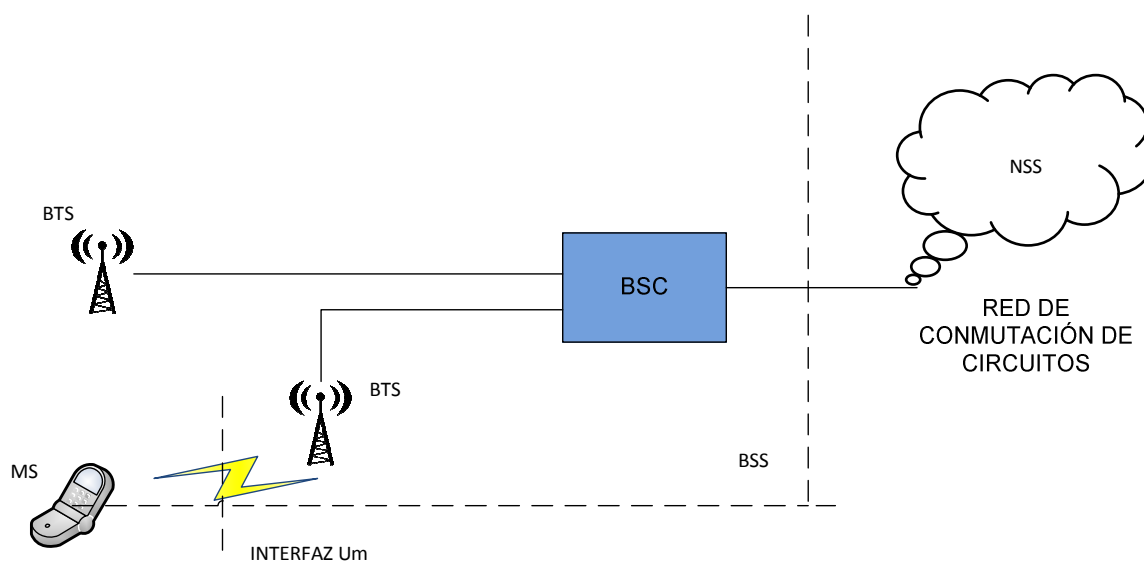


Figura 1-2. Arquitectura GSM a nivel de aplicación

1.3.2 SISTEMA CELULAR [13]

El sistema GSM está basado en una arquitectura celular que permite una mejor utilización del espectro a base de replicar unas unidades fundamentales o celdas, con las que se tesela² el área que se quiere cubrir.

El tamaño de las celdas es un parámetro de diseño que se calcula en base al número medio de usuarios (tráfico medio generado) y el porcentaje de utilización de la estación base. Cuanto menor sea el radio de las celdas, mayor cantidad de usuarios puede soportar el sistema, sin embargo mayor es el coste en infraestructura.

En la Figura 1-3 se muestra un ejemplo de una parte de un sistema celular. Todas las estaciones base (BTS) están interconectadas a través del subsistema de red (NSS).

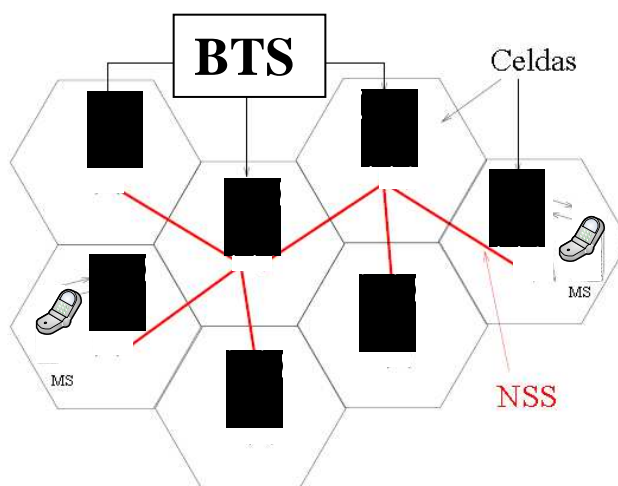


Figura 1-3. Parte de un sistema celular

1.3.3 SMS (SHORT MESSAGE SERVICE) [2]

1.3.3.1 Servicio SMS

El servicio SMS permite transferir un mensaje de texto entre una estación móvil (MS³) y otra entidad (SME⁴) a través de un centro de servicio (SC⁵) (Ver Figura 1-4).

² Un teselado o teselación es una regularidad o patrón de figuras que cubre o pavimenta completamente una superficie plana que no deja huecos y no superpone figuras [3]

³ Mobile Station [1]

⁴ Short Messaging Entity [1]

⁵ Service Center [1]

Diseñado originariamente como parte del estándar digital GSM en la actualidad se encuentra disponible en una amplia variedad de redes, incluyendo las redes 3G.

El servicio final ofrecido es una comunicación punto-punto entre la estación móvil (MS) y la entidad (SME). La entidad puede ser otra estación móvil o puede estar situado en una red fija como un módem. En el caso de envío de un mensaje entre dos móviles, ambas partes son estaciones móviles.

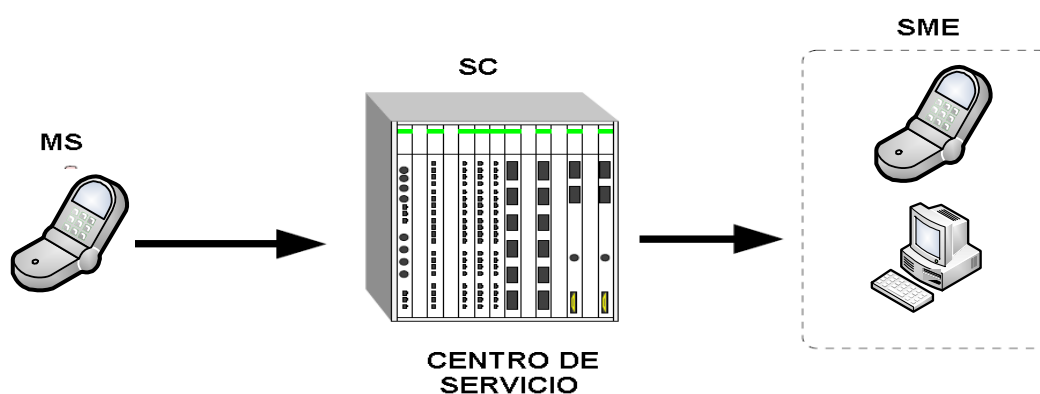


Figura 1-4. Servicio SMS

1.3.3.2 Acceso a los servicios SMS

Actualmente ha aparecido gran cantidad de servicios basados en mensajes cortos. Para poder ofrecer estos servicios ha sido necesario diseñar software y hardware que puedan acceder a los servicios SMS. Esto se ha podido conseguir de varias maneras:

1. Algunos teléfonos se pueden conectar directamente a un PC y mediante un software propietario se puede acceder a los datos de móvil, así como enviar y recibir mensajes SMS. El principal problema de esta solución es que no es abierta, y los fabricantes no proporcionan suficiente información como para poder realizar aplicaciones con ellos. Es necesario realizar ingeniería inversa.
2. Utilización de un MÓDEM GSM. Mediante un MÓDEM GSM se puede conectar cualquier sistema digital a la red GSM, no sólo para enviar

mensajes SMS sino también para transmitir datos. Esta es la solución empleada en el presente trabajo

1.3.4 EMULADOR DE BASE DE DATOS LOCAL [4]

XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente de la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script⁶: PHP⁷ y Perl⁸ (Ver Figura 1-5).

El nombre proviene del acrónimo de “X” (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl.

El programa está liberado bajo la licencia GNU⁹ y actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris, y MacOS X.

Estas características fueron decisivas dentro de este proyecto, razón por la cual este servicio se utilizó para emular una base de datos antes de subirla al servidor web.

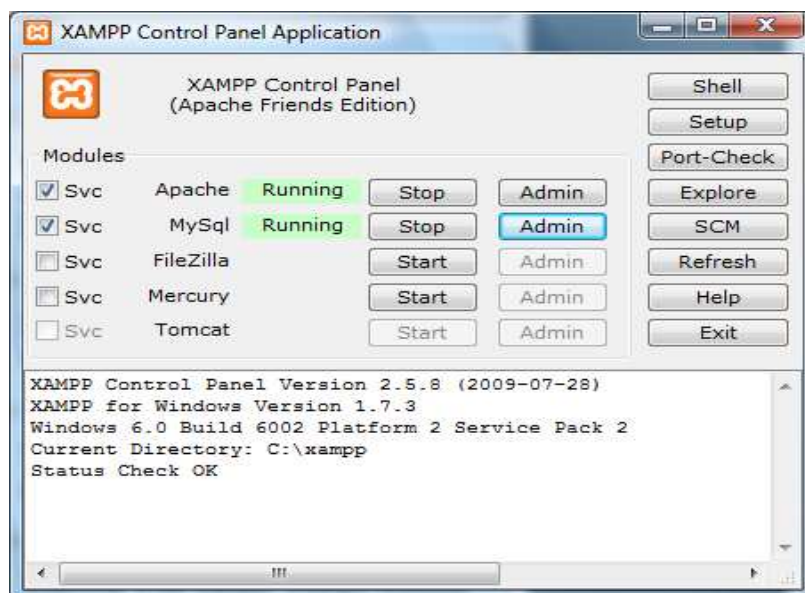


Figura 1-5. Panel de control XAMPP

⁶ Llamado también archivo de procesamiento por lotes, es un programa interpretado que por lo regular se almacena en un archivo de texto plano, realiza diversas tareas como combinar componentes e interactuar con el sistema operativo o con el usuario. [6]

⁷ Hypertext Preprocessor [7]

⁸ Practical Extraction and Report Language [8]

⁹ General Public License, licencia orientada a proteger la libre distribución, modificación y uso de software.

1.3.5 ADMINISTRADOR DE LA BASE DE DATOS: PHPMYADMIN [5]

PhpMyAdmin es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web (Ver Figura 1-6). Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios.

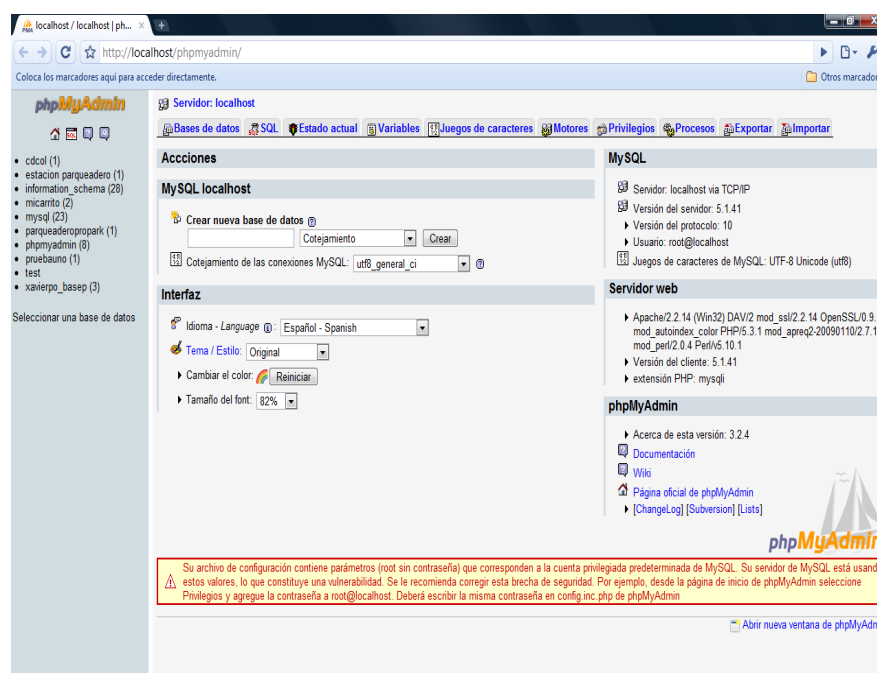


Figura 1-6. Panel de control phpMyAdmin

1.3.6 BASE DE DATOS MYSQL [9]

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL.

MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. Dentro de sus características se puede destacar el amplio subconjunto del

lenguaje SQL, lenguaje de consulta estructurado el cual es declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas.

Una de sus ventajas es el manejo del álgebra y el cálculo relacional permitiendo efectuar consultas con el fin de recuperar, de una forma sencilla, información de interés de una base de datos, así como también hacer cambios sobre ella.

Dentro del subconjunto reservado de comandos los más utilizados son:

- **INSERT**
Agrega uno o más registros a una (y sólo una) tabla en una base de datos relacional.
- **UPDATE**
Es utilizada para modificar los valores de un conjunto de registros existentes en una tabla.
- **DELETE**
Borra uno o más registros existentes en una tabla.

1.3.7 WEB HOSTING SERVICE [10]

Un web hosting es un tipo de servicio de alojamiento de internet que permite a individuos y organizaciones realizar su propia página web accesible a través de la World Wide Web (www).

Los "Web hosts" son compañías que proporcionan espacio en un servidor de su propiedad o arrendado para uso de sus clientes, así como el abastecimiento de conectividad a internet, por lo general en un centro de datos (data center).

El ámbito de aplicación de los servicios de alojamiento es muy variado. La más básica es la página web con una escala de alojamiento de archivos pequeños, donde los archivos pueden ser cargados por un protocolo de transferencia de archivos (FTP, File Transfer Protocol) y donde los archivos se entregan generalmente a la Web "como son" o con escaso procesamiento.

Muchos proveedores de servicios de Internet (ISP, Internet Service Provider) ofrecen este servicio gratuito a sus suscriptores. Las personas también pueden

obtener el alojamiento de su página web de otros proveedores de servicios alternativos.

Una sola página de alojamiento es suficiente sólo para páginas web personales. Un sitio complejo requiere un paquete más amplio que proporcione base de datos de apoyo y desarrollo de plataformas de aplicaciones. Estas facilidades permiten a los clientes escribir o instalar scripts para aplicaciones como foros y gestión de contenidos.

El 'host' también puede proporcionar una interfaz o un panel de control para manejar el servidor Web y la instalación de scripts, así como otros servicios como el correo electrónico.

1.3.7.1 Tipos de alojamiento en un web hosting

Los servicios de alojamiento de internet pueden correr en servidores web. Muchas compañías que no son proveedores de servicio de internet también necesitan un computador conectado permanentemente a la web para que puedan proveer detalles de sus servicios para cualquier interesado.

Entre los tipos de alojamiento a un web hosting se tiene:

- a) Alojamiento gratuito
- b) Alojamiento compartido (Shared hosting)
- c) Alojamiento de imágenes
- d) Alojamiento revendedor (Reseller)
- e) Servidores Virtuales (VPS)
- f) Servidores dedicados
- g) Colocación (Housing)
- h) Alojamiento web en la nube (Cloud hosting)

a) Alojamiento gratuito

El alojamiento gratuito es extremadamente limitado cuando se lo compara con el alojamiento de pago. Estos servicios generalmente agregan publicidad en los sitios y tienen un espacio y tráfico limitado.

b) Alojamiento compartido (Shared hosting)

En este tipo de servicio se alojan clientes de varios sitios en un mismo servidor. Resulta una alternativa muy buena para pequeños y medianos clientes, En resumen las desventajas son: disminución de los recursos del servidor, de velocidad, de desempeño, de seguridad y de estabilidad.

c) Alojamiento de imágenes

Este tipo de hospedaje se ofrece para guardar imágenes en internet, la mayoría de estos servicios son gratuitos y las páginas se valen de la publicidad colocadas en su página al subir la imagen.

d) Alojamiento revendedor (Reseller)

Este servicio de alojamiento está diseñado para grandes usuarios o personas que venden el servicio de hospedaje a otras personas. Estos paquetes cuentan con gran cantidad de espacio y de dominios disponibles para cada cuenta.

e) Servidores virtuales (VPS, Virtual Private Server): mediante el uso de una máquina virtual

La empresa ofrece el control de un ordenador aparentemente no compartido. Así se pueden administrar varios dominios de forma fácil y económica, además de elegir los programas que se ejecutan en el servidor. Por ello, es el tipo de producto recomendado para empresas de diseño y programación web.

f) Servidores dedicados

El término servidor dedicado se refiere a una forma avanzada de alojamiento web en la cual el cliente alquila o compra un ordenador completo, y por tanto tiene el control total y la responsabilidad de administrarlo. El cuidado físico de la máquina y de la conectividad a Internet es tarea de la empresa de alojamiento, que suele tenerlo en un centro de datos.

La principal desventaja de elegir un servidor dedicado, es el costo del servicio, también hay que tomar en cuenta que se requiere de un administrador con amplios conocimientos en el tema del manejo de servidores ya que las configuraciones principales, repercuten el correcto funcionamiento del mismo.

g) Colocación (Housing)

Este servicio consiste básicamente en vender o alquilar un espacio físico de un centro de datos para que el cliente coloque ahí su propio ordenador. La empresa le proporciona la corriente y la conexión a Internet, pero el ordenador servidor lo elige completamente el usuario (incluyendo el hardware de soporte).

h) Alojamiento web en la nube (Cloud hosting)

El alojamiento web en la "nube" (Cloud Hosting) está basado en las tecnologías más innovadoras que permiten a un gran número de máquinas actuar como un sistema conectado a un grupo de medios de almacenamiento, tiene ventajas considerables sobre las soluciones de web hosting tradicionales tal como el uso de recursos.

La seguridad de un sitio web alojado en la "nube" (cloud) está garantizada por numerosos servidores en lugar de sólo uno. La tecnología Cloud también elimina cualquier limitación física para el crecimiento en tiempo real y hace que la solución sea extremadamente flexible.

Bajo estas características se eligió este tipo de alojamiento el cual satisface las necesidades para esta aplicación.

1.3.7.2 Dominios

Algunos planes de alojamiento (pero no los gratuitos) incluyen un nombre de dominio para que sea más fácil acceder al sitio. Si no viene incluido, es el usuario quien tiene que registrar un dominio mediante un registrador o bien usar un subdominio de la misma compañía.

1.3.8 SISTEMAS DE COORDENADAS [11]

Un sistema de coordenadas es una creación humana que permite la definición de la posición de un objeto relativo en otros objetos y al centro de coordenadas.

Los sistemas de coordenadas más utilizados para representar la superficie de la tierra son:

- Coordenadas proyectadas
- Coordenadas cartesianas
- Coordenadas geográficas

1.3.8.1 Coordenadas Proyectadas

Un sistema de coordenadas proyectadas está definido en una superficie plana, de dos dimensiones (Ver Figura 1-7).

En este sistema, las localizaciones son identificadas por coordenadas x , y en una cuadrícula, con origen al centro de dicha cuadrícula. Cada posición tiene dos valores que la referencian a la localización central. Un valor especifica la posición horizontal y el otro valor la posición vertical. Los dos valores son llamados coordenada X y coordenada Y .

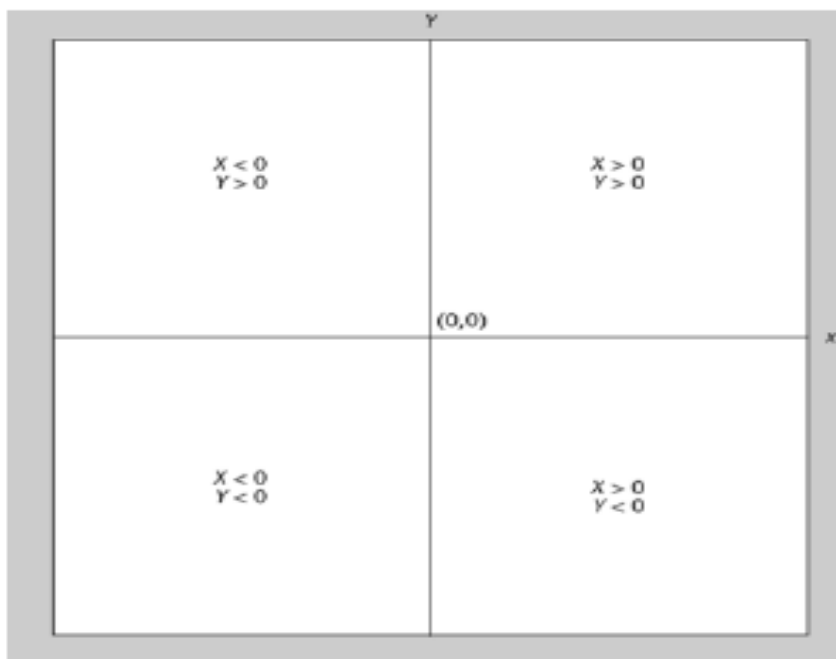


Figura 1-7. Coordenadas Proyectadas

1.3.8.2 Coordenadas Cartesianas [12]

En este sistema de coordenadas una posición se define en un espacio tridimensional por las coordenadas (x, y, z) (Ver Figura 1-8).

Los ejes de coordenadas se determinan de la siguiente forma:

- El eje X pasa por el centro de la tierra y por el meridiano principal de Greenwich.
- El eje Y forma un ángulo de 90° con los otros dos ejes.
- El eje Z pasa por el centro de la tierra y por los polos.

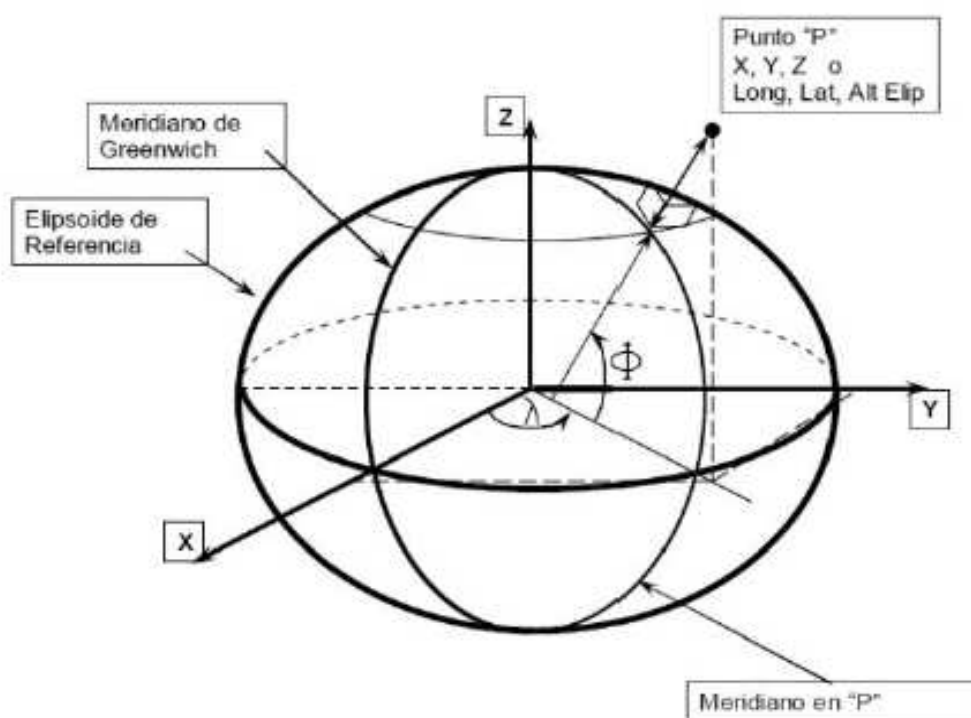


Figura 1-8. Coordenadas Cartesianas

1.3.8.3 Coordenadas Geográficas

Utilizan una superficie esférica tridimensional para definir las localizaciones sobre la superficie terrestre. En este sistema cualquier punto sobre la superficie terrestre se determina con dos ángulos medidos desde el centro de la tierra, que se llaman latitud y longitud (Ver Figura 1-9).

La longitud de un punto es el ángulo medido a lo largo del ecuador desde cualquier punto de la tierra. Las líneas verticales de igual longitud son círculos

máximos que pasan por los polos y se llaman meridianos. La longitud se mide con respecto a un meridiano principal, que es el meridiano de Greenwich (longitud cero).

La latitud de un punto es el ángulo medido desde el centro de la tierra hacia el norte, entre el ecuador y la posición de un punto sobre la superficie terrestre. Las líneas horizontales de igual latitud se llaman paralelos. Se toma el ecuador terrestre como la línea de latitud cero.

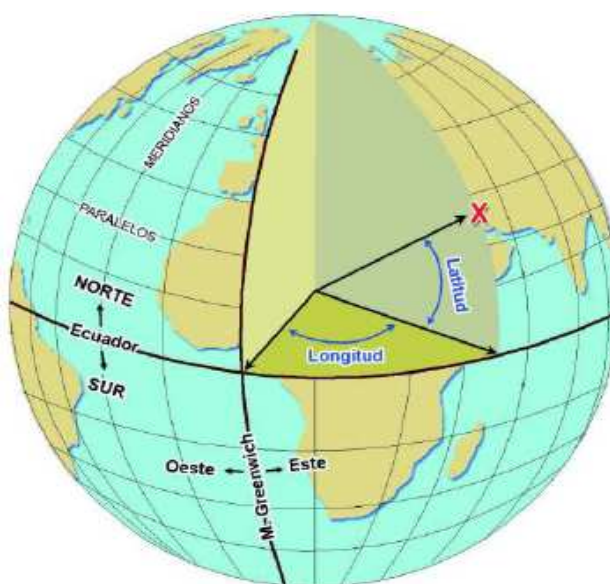


Figura 1-9. Representación de las Coordenadas Geográficas de un punto

1.3.9 BLUETOOTH [21]

Es una especificación industrial para redes inalámbricas de área personal (WPANs) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM (Industrial, Scientific and Medical) de los 2,4 GHz.

Gracias a este protocolo, los dispositivos que lo implementan pueden comunicarse entre ellos cuando se encuentran dentro de su alcance. La comunicación realizada por radiofrecuencia permite que los dispositivos no tengan que estar alineados y puedan incluso estar en habitaciones separadas si la potencia de transmisión lo permite.

1.3.9.1 Clasificación Bluetooth

Existen varias clasificaciones pero para el presente caso la más idónea hace referencia a su potencia de transmisión (Ver Tabla 1-2), siendo totalmente compatibles los dispositivos de una clase con los de las otras.

Tabla 1-1. Clasificación Bluetooth según los tipos de energía y rango

Clase	Potencia máxima permitida (mW)	Potencia máxima permitida (dBm)	Rango aproximado (m)
Clase 1	100	20	~100
Clase 2	2.5	4	~25
Clase 3	1	0	~1

Luego de este preámbulo se puede concluir que, la utilización de los sistemas de posicionamiento global (GPS), se ha vuelto cada vez más convencional, encontrando así más personas que sin tener una gran capacidad financiera, disponen de algún dispositivo para conocer la posición en tiempo real de un usuario o vehículo, y si a esto se le añade la versatilidad para la implementación de soluciones de ingeniería como la del protocolo Bluetooth, el servicio SMS, la base de datos MySQL y de un software especializado, se pueden obtener resultados prácticos que ayuden a solventar problemas cotidianos como el presentado en este capítulo.

CAPÍTULO 2

DESCRIPCIÓN DEL HARDWARE PARA EL DESARROLLO DEL PROTOTIPO

2.1 INTRODUCCIÓN

Durante las últimas décadas el desarrollo de las tecnologías ha evolucionado de manera muy rápida, a tal punto que se han creado nuevas formas de comunicación, que cada vez son más aceptadas por el mundo actual. Razón por la cual, para mejorar el servicio y la eficiencia de la solución se decidió integrar tecnologías soportadas en equipo estandarizado como el que se detalla a continuación.

2.2 PDA (ASISTENTE DIGITAL PERSONAL) [15]

Un PDA es un ordenador de bolsillo, el cual originalmente fue diseñado como agenda electrónica con un sistema de reconocimiento de escritura.

Hoy en día estos dispositivos, pueden realizar muchas de las funciones que hace una computadora de escritorio con la ventaja de ser portátil.

Un PDA típico tiene al menos una pantalla táctil para ingresar información, una tarjeta de memoria para almacenarla y al menos un sistema de conexión inalámbrica.

La mayoría de los PDA actuales poseen conectividad 3G, lo que permite el acceso a internet de alta velocidad en prácticamente cualquier lugar, además de la funcionalidad como teléfono móvil avanzado.

Los PDAs son usados para almacenar información que puede ser consultada a cualquier hora y en cualquier lugar. Por lo general, estos dispositivos son utilizados de manera doméstica, sin embargo también se pueden encontrar en otros campos.

Muchos PDAs son usados en vehículos para poder utilizar diversas aplicaciones, y es por esto que cada vez es más común encontrar dispositivos Bluetooth

instalados dentro de vehículos nuevos, siendo este precedente inspiración para proponer la presente.

2.2.1 SISTEMAS OPERATIVOS PARA EQUIPOS EMBEBIDOS

Hoy en día tenemos los siguientes sistemas operativos y equipos competidores:

- Dispositivos Symbian OS presente en las gamas altas de teléfonos móviles de Nokia y Sony Ericsson.
- Research in Motion con sus Blackberry, más teléfonos inteligentes que PDAs, pero que han copado una parte importante del Mercado corporativo a la vez que incorporaban prestaciones de PDA.
- Dispositivos Pocket Pc con HP como líder de fabricantes acompañado por otras empresas de informática como Dell o Acer, a quienes se han incorporado los fabricantes de Taiwán como High Tech Computer que van copando el mercado del teléfono inteligente con sus marcas propias (como Qtek) o fabricando para terceros y, sobre todo, operadores de telefonía móvil.
- Dispositivos Palm OS, hoy en día mantenido casi en solitario por Palm, pero que hasta hace poco ha tenido importantes fabricantes como Sony.
- Dispositivos con iPhone OS actualmente iOS4, son los nuevos y revolucionarios dispositivos de Apple, el iPhone, iPhone 3G, iPhone 3GS, iPhone 4 y el iPad touch.
- Dispositivos con Android, sistema operativo de Google, basado en el núcleo Linux, utilizado por HTC y actualmente Motorola entre otros, es el nuevo fuerte competidor del iPhone, tiene la gran ventaja de ser Open Source.
- Dispositivos con Windows Mobile, sistema operativo usado principalmente por HTC como el HTC Touch Diamond o el HTC Touch HD, principales competidores del iPhone, ya que igualan o superan sus características.

Debido a lo anteriormente expuesto se decidió seleccionar la HTC HD2 (Ver Figura 2-1) debido a su tecnología táctil capacitiva con SO Windows Mobile y sobre todo por la potencia de procesamiento de 1GHz (Ver Anexo A) la cual

garantiza así una respuesta rápida y fluida de la aplicación diseñada para el mismo.



Figura 2-1. PDA: HTC HD2

2.3 SISTEMA DE POSICIONAMIENTO GLOBAL “GPS” [16]

El Sistema de Posicionamiento Global (“GPS”, Global Positioning System o NAVSTAR, NAVigation System and Ranging), es un sistema de navegación global por satélite (“GNSS”, Global Navigation Satellite System) que permite determinar el posicionamiento y localización de un objeto en cualquier parte del globo terrestre.

2.3.1 FUNCIONAMIENTO DE UN GPS

Su funcionamiento se basa en una red de 24 satélites puestos en órbita sobre el globo terráqueo con trayectorias sincronizadas para cubrir toda la superficie del mismo, utilizando el método de trilateración¹⁰ inversa (triangulación) el receptor localiza automáticamente como mínimo 3 satélites de dicha red.

¹⁰ Método matemático para determinar las posiciones relativas de objetos usando geometría de triángulos de forma análoga a la triangulación.

El receptor GPS utiliza la información enviada por los satélites (tiempo en la que emitieron las señales, localización de los mismo) para tratar de sincronizar su propio reloj con el reloj atómico que poseen los satélites para luego calcular la distancia de cada satélite respecto al punto de medición y finalmente determinar la posición relativa respecto de los satélites involucrados.

Cada satélite indica que el receptor se encuentra en un punto en la superficie de la esfera, con centro en dicho satélite y de radio la distancia total hasta el receptor.

Obteniendo información de dos satélites se nos indica que el receptor se encuentra sobre la circunferencia que resulta cuando se intersecan las dos esferas.

Es así como obteniendo información de más de tres satélites, eliminamos el inconveniente de la falta de sincronización de los relojes pudiendo así determinar una posición 3-D exacta (latitud, longitud y altitud) que además utiliza el principio de dilución geométrica¹¹ (GDop o DOP Dilución de precisión) (Ver Figura 2-2) para tener una mayor precisión.

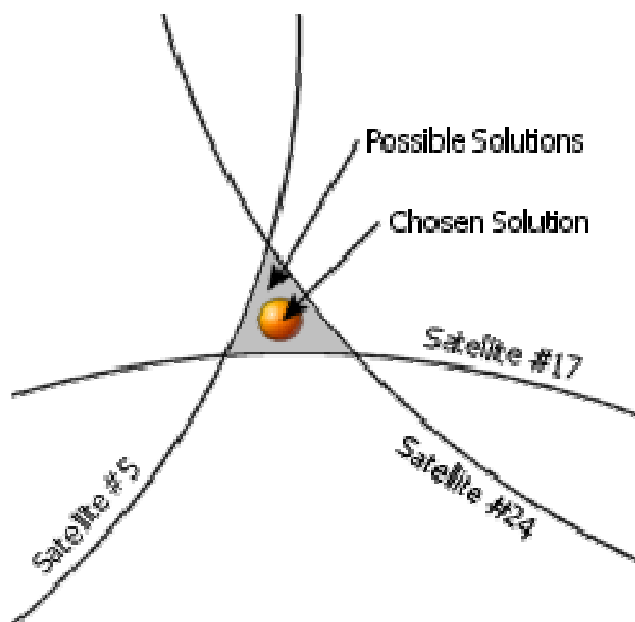


Figura 2-2. (DOP) Dilución de la precisión geométrica

¹¹ Proceso de unión de dos o más polígonos mediante la eliminación de los lados comunes.

2.3.2 CAUSAS DE ERROR DE PRECISIÓN DEL GPS [17]

Existen varios fenómenos que pueden causar una mala precisión. Por ejemplo, cuando las señales de radio por satélite son transmitidas, estas son distorsionadas por la tropósfera y especialmente por la ionósfera. De hecho, los satélites que vuelan muy por debajo del horizonte no son buenos para conseguir una buena resolución (obtención de coordenadas) porque las señales viajan a través de la atmósfera.

Algunos dispositivos GPS pueden inclusive excluir estos satélites de una solución para evitar los problemas de precisión que ellos podrían causar (Ver Figura 2-3).

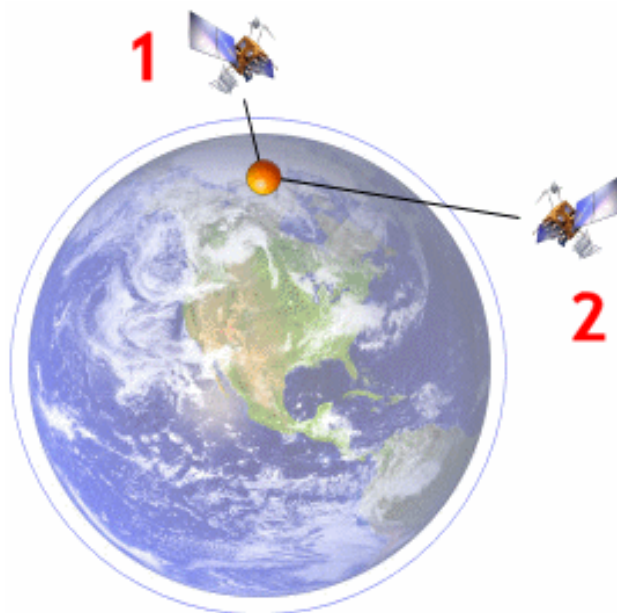


Figura 2-3. La señal del satélite “1” viaja menos a través de la atmósfera, dando como resultado menor distorsión. El satélite “2” está menos en el horizonte, sin embargo, el resultado es una distorsión atmosférica significativa.

Afortunadamente, la distorsión atmosférica puede ser medida y corregida en su mayor parte. Esto se logra mediante el uso de estaciones terrestres GPS (Ver Figura 2-4).

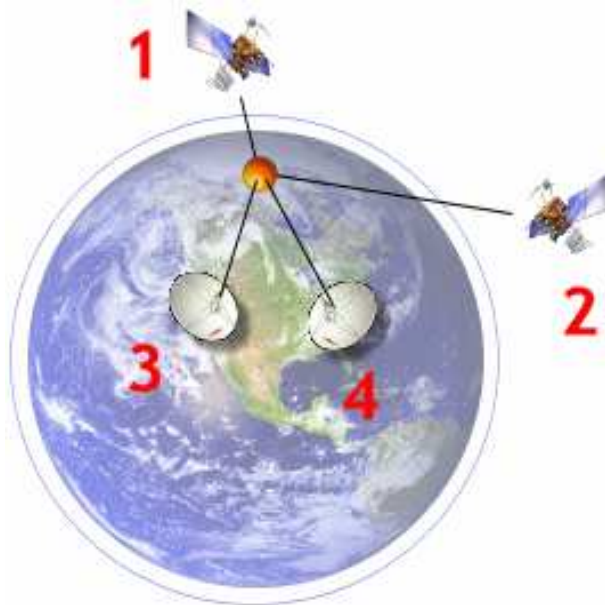


Figura 2-4. La distorsión en señales de radio son corregidas por la combinación de señales satelitales (1 y 2) con información de corrección transmitida vía estaciones terrestres GPS (3 y 4)

También existen errores de precisión si los satélites se salen de su órbita (errores de efemérides, (Ver Figura 2-5) o por señales retardadas (denominado efecto múltiple, (Ver Figura 2-6) producidas por rebote de las señales en edificios u otros obstáculos.

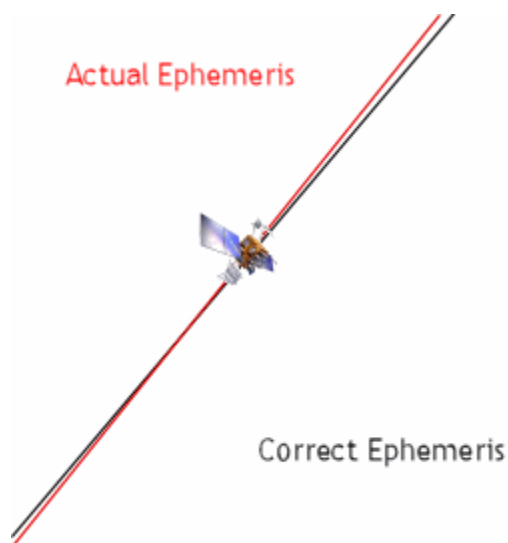


Figura 2-5. Errores de Efemérides

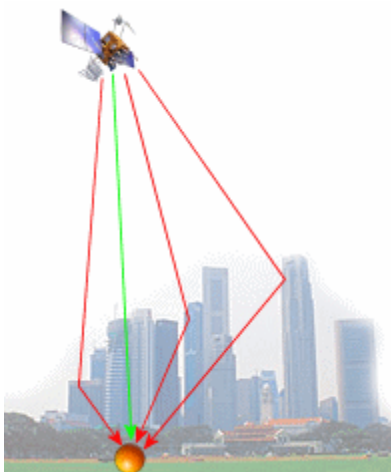


Figura 2-6. Error por efecto múltiple

2.3.3 SELECCIÓN DEL HARDWARE GPS [18]

Hay tres obstáculos que se deben superar antes de poder empezar a descifrar la secuencia de la trama GPS y el uso de los datos en la aplicación:

1. Conseguir la conexión correcta de hardware
2. Lectura del puerto COM desde una aplicación
3. Portabilidad del GPS

1. Conseguir la conexión correcta de hardware

Las conexiones de hardware de GPS vienen en tres variedades comunes:

- puerto Serie
- USB
- Bluetooth

Para leer los datos de un GPS, hay que hacer que estas conexiones se parezcan a los puertos COM de la PC ya que si el GPS se conecta a la PC mediante un puerto serie, el problema se resume a una comunicación serial RS-232.

En este caso se eligió el GPS/Bluetooth, debido a que se busca sobre todo, brindar seguridad al conductor en el momento de utilizar la aplicación y como resultado de esta elección fue el eliminar alambres que podrían causar enredos con la palanca de cambios y/o el volante, situaciones que al final se traducirían en posibles accidentes de tránsito; el costo, la compatibilidad entre dispositivos, baja interferencia y sobre todo la disponibilidad que hoy se presenta como parte de diferentes equipos (como un PDA) son algunas de las ventajas por las cuales se seleccionó este tipo de conexión.

Es así como ciertos fabricantes nos brindan ya de facto un puerto COM auto detectable, que mediante una clave de emparejamiento permite conectarse directamente con otros dispositivos.

2. Lectura del puerto COM desde una aplicación

Ahora que se tiene el puerto COM creado hoy en día hay docenas de controles ActiveX y otros .NET dedicados a hacer que la lectura y la escritura en un puerto COM sean más fáciles, teniendo como una herramienta sencilla para la comprobación de la recepción de datos del GPS/Bluetooth el HyperTerminal (Ver Figura 2-7)

```

GPS-BLUETOOTH - HyperTerminal
Archivo Edición Ver Llamar Transferir Ayuda

$GPGGA,021357.302,0014.7276,S,07831.9645,W,0,00,50.0,2881.4,M,13.8,M,0.0,0000*75
$GPGSA,A,1,,,,,,,,,50.0,50.0,50.0*05
$GPRMC,021357.302,V,0014.7276,S,07831.9645,W,.,,241210,.,*1E
$GPGGA,021358.302,0014.7276,S,07831.9645,W,0,00,50.0,2881.4,M,13.8,M,0.0,0000*7A
$GPGSA,A,1,,,,,,,,,50.0,50.0,50.0*05
$GPGSV,3,1,11,18,54,083,00,22,54,016,00,31,53,280,00,30,46,183,00*7B
$GPGSV,3,2,11,25,40,068,00,21,31,164,00,14,20,353,00,16,13,204,00*76
$GPGSV,3,3,11,29,10,128,00,06,08,230,00,12,05,046,00*43
$GPRMC,021358.302,V,0014.7276,S,07831.9645,W,.,,241210,.,*11
$GPGGA,021359.302,0014.7276,S,07831.9645,W,0,00,50.0,2881.4,M,13.8,M,0.0,0000*7B
$GPGSA,A,1,,,,,,,,,50.0,50.0,50.0*05
$GPRMC,021359.302,V,0014.7276,S,07831.9645,W,.,,241210,.,*10
$GPGGA,021400.301,0014.7276,S,07831.9645,W,0,00,50.0,2881.4,M,13.8,M,0.0,0000*73
$GPGSA,A,1,,,,,,,,,50.0,50.0,50.0*05
$GPRMC,021400.301,V,0014.7276,S,07831.9645,W,.,,241210,.,*18
$GPGGA,021401.301,0014.7276,S,07831.9645,W,0,00,50.0,2881.4,M,13.8,M,0.0,0000*72
$GPGSA,A,1,,,,,,,,,50.0,50.0,50.0*05
$GPRMC,021401.301,V,0014.7276,S,07831.9645,W,.,,241210,.,*19

0:00:00 conectado Autodetect. 9600 8-N-1 DESPLAZAR MAY NUM Capturar Imprimir

```

Figura 2-7. Adquisición de datos del GPS/Bluetooth mediante el HiperTerminal

NOTA: Antes de empezar el desarrollo de la aplicación para el PDA se debe emparejar dicho dispositivo con el GPS/Bluetooth a fin de asignar un puerto COM a la aplicación (Ver Figura 2-8).

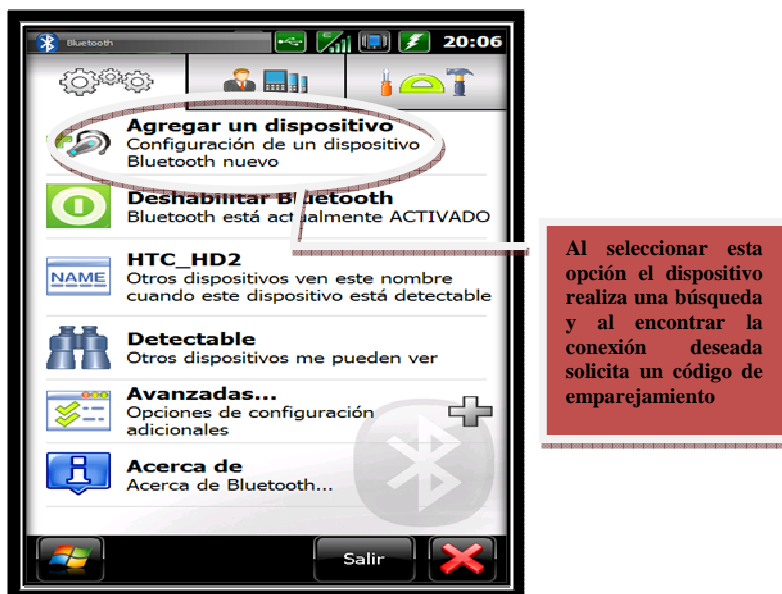


Figura 2-8. Emparejamiento GPS/Bluetooth y PDA (HTC HD2)

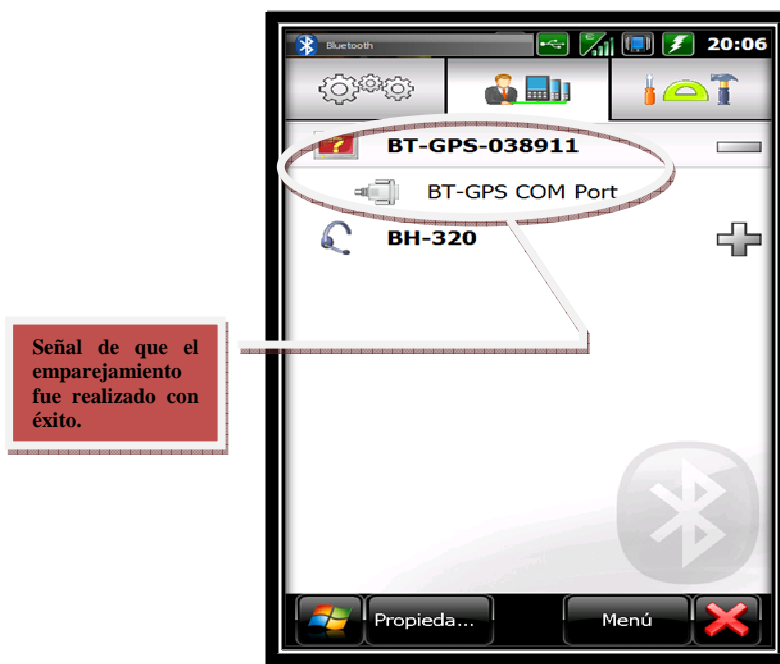


Figura 2-9. Confirmación de emparejamiento

3. Portabilidad del GPS

Desafortunadamente, los fabricantes de GPS a menudo eligen qué partes del estándar utilizado sus equipos soportarán. Para afrontar la situación, se debe decidir qué tramas se van a utilizar como fuentes primarias de la posición, velocidad, rumbo, navegación y datos de satélite acorde a la situación.

Para el caso específico se seleccionó el receptor GPS-Bluetooth Dell NAVTEQ ON BOARD Modelo BT-309 debido a su conexión USB, el puerto COM que posee de facto y sobre todo por contar como fuente primaria de datos parte del estándar NMEA 0183. (Ver Anexo B) (Ver Figura 2-10).



Figura 2-10. GPS-Bluetooth DELL NAVTEQ ON BOARD Modelo BT-309

2.3.4 TRAMA GPS SEGÚN NORMA NMEA 0183 [19]

El estándar NMEA 0183 ha sido desarrollado por la *National Marine Electronics Association* para establecer comunicación de todo tipo de instrumentos marinos entre sí. Este estándar fue adoptado por todos los fabricantes de receptores GPS a fin de poder conectarles otros instrumentos de navegación.

El flujo de datos NMEA 0183 consiste en una serie de "tramas" delimitadas por un carácter de nueva línea. Cada trama comienza con un identificador de seis caracteres, siendo el primero "\$".

El estándar define docenas de frases, pero como se mencionó, sólo una fracción se aplica directamente a los dispositivos GPS. En el presente caso, los adquiridos son:

- \$ GPGGA.- Sistema de posicionamiento global de datos fijos
- \$ GPGSA.- DOP del GPS y los satélites activos
- \$ GPGSV.- GPS satélites a la vista
- \$ GPRMC.- Especificaciones mínimas recomendadas GPS para la transmisión de datos

Dentro de las tramas mencionadas, es óptimo elegir el campo RMC ya que contiene toda la información necesaria para la solución del proyecto.

2.3.4.1 Trama RMC [20]

A continuación un ejemplo y el significado de cada campo dentro de la trama RMC (Ver Tabla 2-1):

Tabla 2-1. Desglose trama \$GPRMC

\$ GPRMC, 111111, 2, 3333.33, 4, 55555.55, 6, 777.7, 888.8, 999999, aaa.a, b *cc	
Donde:	
111111	tiempo de ajuste UTC ¹² (Time Stamp) en formato hhmmss (hora,minuto, segundo)
2	es la validez del ajuste ("A" = válido, "V" = no válido)
3333.33	es la latitud actual en formato ddmn.mm (grados, minutos)
4	es el hemisferio de latitud (N = norte, "S" = sur)
55555.55	es la longitud actual en formato dddmm.mm (grados, minutos)
6	es el hemisferio de longitud ("E" = este, "W" = oeste)
777.77	es la velocidad en nudos
888.8	es el verdadero curso en grados

¹² Escala de tiempo universal coordinado

999999	es la fecha de ajuste UTC (Date Stamp) en formato DDMMAA (día, mes, año)
aaa.a	es la variación magnética en grados
b	es la dirección de la variación magnética ("E" = este, "W" = oeste)
*cc	es la suma de comprobación "CHECKSUM"

2.4 MODEM GSM [1]

Existen dos tipos de módems (Ver Figura 2-11), según la aplicación que queramos realizar.

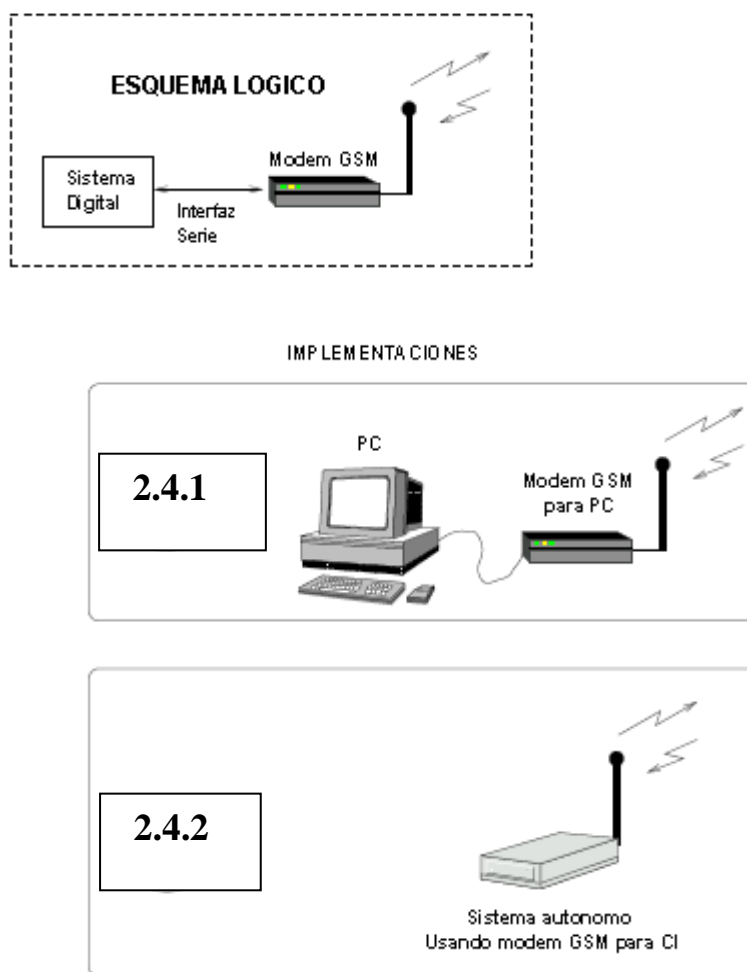


Figura 2-11. Utilización de un módem GSM desde un sistema digital

2.4.1 Módems para circuito impreso (CI)

Son módems de reducido tamaño y perfectamente apantallados que están preparados para ser incorporados dentro de un circuito impreso y que permiten desarrollar un hardware específico y que no depende de un PC.

2.4.2 Módems para PC

Tienen un tamaño también bastante reducido; la distinción principal que se suele hacer es entre módems internos y externos, aunque recientemente han aparecido módems llamados “módems software”, más conocidos como “winmódems” o “linuxmódems”, que han complicado un poco el panorama.

También existen los módems para XDSL¹³, RDSI, etc. Y los que se usan para conectarse a través de cable coaxial de 75 ohms (cable módems)

2.4.2.1 Internos

Consisten en una tarjeta de expansión sobre la cual están dispuestos los diferentes componentes que forman el módem. Existen para diversos tipos de conector:

- Bus ISA: debido a las bajas velocidades que se manejan en estos aparatos, durante muchos años se utilizó en exclusiva este conector, hoy en día en desuso.
- Bus PCI: el formato más común en la actualidad, todavía en uso.
- AMR: en algunas placas, económicos pero poco recomendables por su bajo rendimiento. Hoy es una tecnología obsoleta. La principal ventaja de estos módems reside en su mayor integración con el ordenador, ya que no ocupan espacio sobre la mesa y reciben energía eléctrica directamente del propio ordenador. Además, suelen ser algo más baratos debido a que carecen de carcasa y transformador, especialmente si son PCI (en este caso, son casi todos del tipo “módem software”). Por el contrario, son algo más complejos de instalar y la información sobre su estado sólo puede obtenerse por software.

¹³ X Digital Subscriber Line [22]

2.4.2.2 Externos

Similares a los anteriores, pero externos al ordenador o PDA. La ventaja de estos módems reside en su fácil portabilidad entre ordenadores diferentes (algunos de ellos más fácilmente transportables y pequeños que otros), además de que es posible saber el estado del módem mediante los leds de estado que incorporan. Por el contrario, ocupan más espacio que los internos.

2.4.3 Tipos de conexión del módem [23]

- La conexión de los módems telefónicos externos con el ordenador se realiza generalmente mediante uno de los puertos serie tradicionales o COM (RS232), por lo que se usa la UART¹⁴ del ordenador, que deberá ser capaz de proporcionar la suficiente velocidad de comunicación. Estos módems necesitan un enchufe para su transformador.
- Modems PC Card: son módems en forma de tarjeta que se utilizaban en portátiles antes de la llegada del USB (PCMCIA¹⁵). Su tamaño es similar al de una tarjeta de crédito algo más gruesa, pero sus capacidades son las mismas que los modelos estándares.
- Existen modelos para puerto USB, de conexión y configuración aún más sencillas, que no necesitan toma de corriente.

Hay modelos tanto para conexión mediante telefonía fija, como para telefonía móvil,

En el presente caso se utilizó el módem USB ZT-MF100 (Ver Figura 2-12), el cual es un módem 3G multi-modo, aplicable a todas las redes GSM / GPRS entre otras.

Integra funciones de módem y teléfono móvil y combina la comunicación móvil con el internet perfectamente.

La capacidad de procesamiento de datos del USB MÓDEM ZTE es a través de las redes móviles mencionadas, esto permitió romper con las limitaciones de tiempo y zona, otorgando al proyecto una comunicación sin

¹⁴ Universal Asynchronous Receiver-Transmitter [24]

¹⁵ Personal Computer Memory Card [25]

cables a cualquier hora y lugar. Este módem soporta muchos tipos de computadoras portátiles u otras computadoras que ofrezcan conexión USB.

Posee como requerimientos:

SISTEMA OPERATIVO: Windows 2000/XP/Vista/7

CPU: 500MHz Pentium o superior

DISCO DURO: 20G o superior, (100Mb espacio libre en disco)

RAM: 128M o superior



Figura 2-12. Módem GSM ZT-MF100

- Módems software, HSP (Host Signal Processor) o Winmódems: son módems generalmente internos, en los cuales se han eliminado varias piezas electrónicas (por ejemplo, chips especializados), de manera que el microprocesador del ordenador debe suplir su función mediante un programa. Lo normal es que utilicen como conexión una ranura PCI¹⁶ (o una AMR¹⁷), aunque no todos los módems PCI son de este tipo. El uso de la CPU entorpece el funcionamiento del resto de aplicaciones del usuario. Además, la necesidad de disponer del programa puede imposibilitar su uso con sistemas operativos no soportados por el fabricante, de manera que, por ejemplo, si el fabricante desaparece, el módem quedaría

¹⁶ Peripheral Component Interconnect [26]

¹⁷ Audio/modem raiser [27]

eventualmente inutilizado ante una futura actualización del sistema. A pesar de su bajo coste, resultan poco o nada recomendables.

- Módems completos: los módems clásicos no HSP, bien sean internos o externos. En ellos, el rendimiento depende casi exclusivamente de la velocidad del módem y de la UART del ordenador, no del microprocesador.

Tras toda esta explicación se puede concluir que la utilización de un equipo estándar en la presente solución, garantiza el envío, recepción y posterior tratamiento de la información utilizada por el sistema prototipo para un óptimo resultado del mismo.

CAPÍTULO 3

DESARROLLO DEL SOFTWARE

3.1 INTRODUCCIÓN

Con la creciente popularidad de los dispositivos informáticos móviles, entre los que se encuentran los PDAs y los teléfonos móviles existen varios conjuntos de herramientas y bibliotecas que los desarrolladores pueden utilizar para crear aplicaciones de gran riqueza para estos dispositivos y obviamente para aplicaciones de escritorio.

3.2 VISUAL STUDIO .NET [29]

Es la nueva versión de la familia de herramientas de desarrollo de software de Microsoft (Ver Figura 3-1), naturalmente orientadas hacia su nuevo entorno de programación .NET Framework.

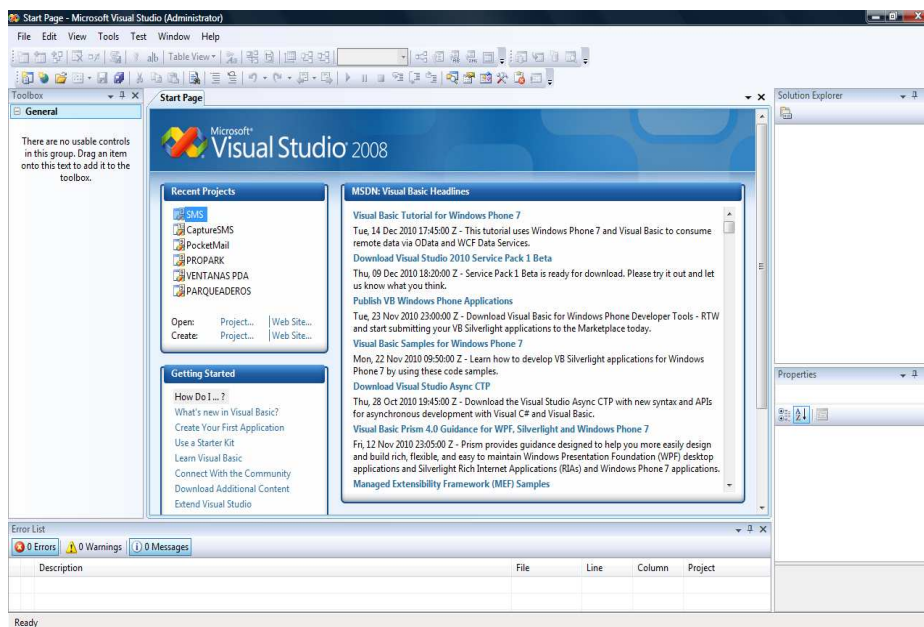


Figura 3-1. Microsoft Visual Studio 2008

Visual Studio .NET (de aquí en adelante VS.NET), al tratarse de un entorno de desarrollo integrado (IDE, Integrated Development Environment), une a todas las

herramientas del SDK (Software Development Kit): compiladores, editores, ayuda, etc., facilitando en gran medida la creación de programas.

Esta versión del entorno incorpora la compatibilidad con aplicaciones en desarrollo para dispositivos móviles inteligentes que utilizan .NET Compact Framework.

Con VS.NET, millones de desarrolladores de aplicaciones de escritorio y otros dispositivos de bolsillo existentes obtienen un entorno de desarrollo uniforme. Esto hace que el desarrollo de aplicaciones sea más fácil y menos costoso, permite que se vuelva a emplear el código y como consiguiente, faculta la producción de aplicaciones más robustas. [28]

De igual manera, impulsa el desarrollo de aplicaciones de “informática de conexión ocasional”, las cuales se ajustan dinámicamente a la disponibilidad de una conexión inalámbrica y logran trabajar fuera de línea de forma útil hasta que se establece la conexión de nuevo. Además con las herramientas de desarrollo móvil de .NET Compact Framework y VS .NET se pueden crear aplicaciones en Visual Basic.NET o C#; es probable que una versión posterior admita otros lenguajes .NET de Windows.

Las aplicaciones de .NET Compact Framework se pueden depurar mediante un emulador que se ejecuta en la máquina de desarrollo o bien, en un dispositivo físico conectado a la máquina de desarrollo a través de un puerto serie (USB) o una conexión de red (Ver Figura 3-2) [31].



Figura 3-2. Emulador PDA WM 6.5 Professional

Previo al diseño de la aplicación cliente y servidor se creó una base de datos donde registrar la información de los respectivos parqueaderos.

3.3 DESARROLLO DE LA BASE DE DATOS EN MYSQL [32]

Una vez instalado el servidor MySQL con XAMPP, se simuló una base de datos la cual, luego de haber superado algunas pruebas, fue subida a la web íntegramente desde el PHPmyAdmin.

Dentro de la base de datos, con el objeto de poder organizar de mejor manera la información obtenida en los respectivos parqueaderos, se crearon tres tablas esenciales denominadas:

1. Parqueaderos
2. Registroplazas
3. ServidorPropark

La primera tabla llamada “parqueaderos” la cual contiene 8 campos (ocho columnas) que proveen la información necesaria a la aplicación del servidor.

Dentro de estos campos se incluye la información referente a:

- ID (Identificación asignada a cada parqueadero por la base de datos)
- NOMBRE (campo asignado para la identificación de cada parqueadero)
- LATITUD (campo asignado para el ingreso de la latitud exacta de cada parqueadero)
- LONGITUD (campo asignado para el ingreso de la longitud exacta de cada parqueadero)
- DIRECCIÓN (campo asignado para el ingreso de la dirección exacta del parqueadero)
- DISPONIBLES (campo asignado para el ingreso y actualización del número de plazas libres por parte de la aplicación cliente instalada en el computador de cada parqueadero)
- OCUPADOS (campo asignado para el ingreso y actualización del número de plazas ocupadas por parte de la aplicación cliente instalada en el computador de cada parqueadero)

- CLAVE (campo asignado para el ingreso y actualización de una clave de seguridad de entrada a la aplicación cliente respectiva)

Al ser un sistema prototipo se consideró iniciar la base de datos con 10 registros (filas) correspondientes a diez parqueaderos (entre públicos y privados indistintamente) (Ver Tabla 3-1).

La segunda tabla, denominada “registroplazas”, cuenta con 6 campos (seis columnas) (Ver Tabla 3-2), y fue creada para almacenar las plazas de los parqueaderos registrados. Su objetivo primordial es el proveer espacio donde guardar la información ingresada por la aplicación del parqueadero (Cliente).

- ID (Identificación asignada a cada registro por la base de datos)
- IDPARQ (Identificación asignada a cada aplicación cliente)
- PLACA (campo asignado para el ingreso de la placa de cada automotor que ingrese al respectivo parqueadero)
- PLAZA (campo destinado para el ingreso de la plaza asignada dentro del parqueadero)
- FECHA (campos destinados para el ingreso de la fecha en la que entra el automotor al respectivo parqueadero)
- HORA (campos destinados para el ingreso de la hora en la que entra el automotor al respectivo parqueadero)

Tabla 3-1. “parqueaderos”





















←T→	Id	Nombre	Latitud	Longitud	Direccion	Disponibles	Ocupados	Clave
 	1	Cadisan	-0.218634	-78.512029	CADISAN: Mejía entre García Moreno y Benalcazar	16	84	1111
 	2	Recreo	-0.251258	-78.523922	EL RECREO.- Av. Pedro Vicente Maldonado S11-122, f...	12	82	2222
 	3	Pintado	-0.245247	-78.532784	EL PINTADO.- Jose Ignacio Canelos Oe5-217 y Sgto....	74	26	3333
 	4	Tejar	-0.215815	-78.516165	EL TEJAR.- Calle Lopez entre Hermano Miguel y Meji...	89	11	4444
 	5	Montufar	-0.220456	-78.508381	MONTUFAR.- Calle Montufar entre Olmedo y Mejía (ju...	10	90	5555
 	6	Rio Coca	-0.163502	-78.473995	RIO COCA.- Av. Rio Coca y de las Palmeras (junto a...	81	19	6666
 	7	La Ronda	-0.224779	-78.51447	LA RONDA.-sur de la Calle Morales entre Venezuela ...	33	77	7777
 	8	Ofelia	-0.108887	-78.48805	LA OFELIA.- entre Diego Vasquez de Cepeda y De los...	46	54	8888
 	9	Quicentro Shopping	-0.177068	-78.47826	QUICENTRO: Av. Naciones Unidas entre 6 de Diciembr...	67	33	9999
 	10	EPN	-0.208767	-78.488914	EPN.-Ladron de Guevara E11-253 (esq. de la Isabel ...	86	14	4321

Tabla 3-2. “registroplaza”

			Id	IdParq	Placa	Plaza	Fecha	Hora de ingreso
<input type="checkbox"/>			4	Rio Coca	ASD-1111	34	15/02/2010	12:00:00
<input type="checkbox"/>			3	Tejar	ASD-1234	12	15/02/2010	14:30

Una última tabla denominada “servidorPropark” sirve de lugar de almacenamiento del nombre de usuario y clave para el acceso a la aplicación servidor, así como un registro “N” el cual guarda el número de parqueaderos registrados.

Se debe recalcar que como característica principal, esta tabla es modificable solo por la persona que administra la base de datos directamente desde el panel de control de la base de datos “cpanel” (Ver Tabla 3-3)

Tabla 3-3 “servidorPropark”

			Indice	Usuario	Contraseña	N
<input type="checkbox"/>			1	usuario	1234	0
<input type="checkbox"/>			2	NO_BORRAR		10

Luego, el proceso de exportación se lo hizo obteniendo el comprimido con gzip. Es importante mencionar que para el presente proyecto se contrató un plan web hosting (Ver Figura 3-3) con dominio: www.xavierporras.com.

Con ayuda del panel de control del plan contratado, “cpanel” (Ver Figura 3-4), se procedió a importar la base de datos a través del PhpMyAdmin y de esta manera manipular la base de datos desde la web (Ver Figura 3-5).

Especificaciones	Premium
Espacio en Disco	10000 Mb
24/7 Support	
Dominio Alojados	3
Email Accounts	50
Bandwidth Allowance	100000 Mb
Guaranteed Network Uptime	
CGI, Ruby (RoR),Perl, PHP, MySQL	
Dominio Gratis	
Cpanel / Fantastico	
Bases de Datos SQL	10

Figura 3-3. Especificaciones plan web hosting

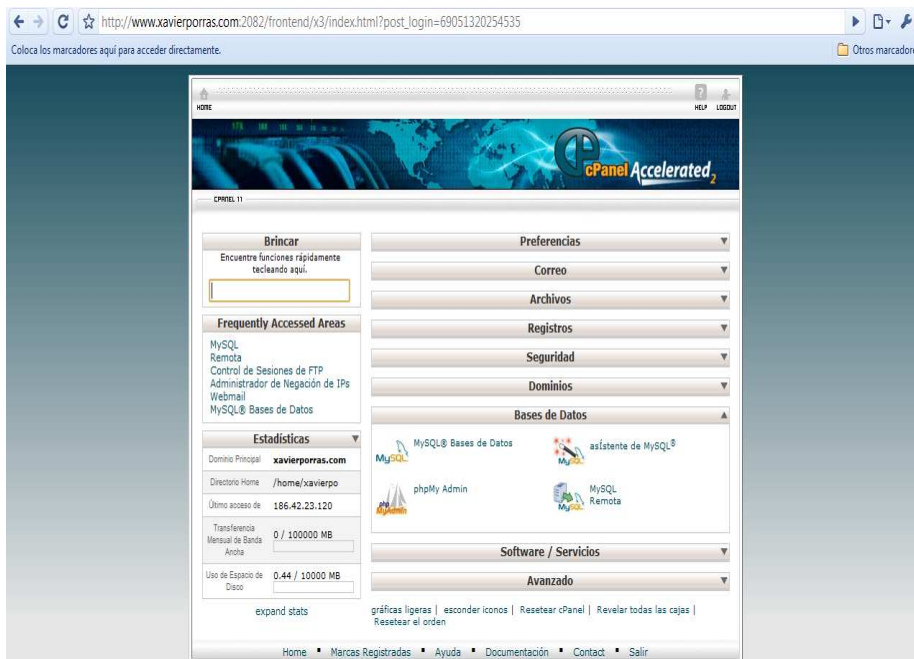


Figura 3-4. Panel de control: www.xavierporras.com

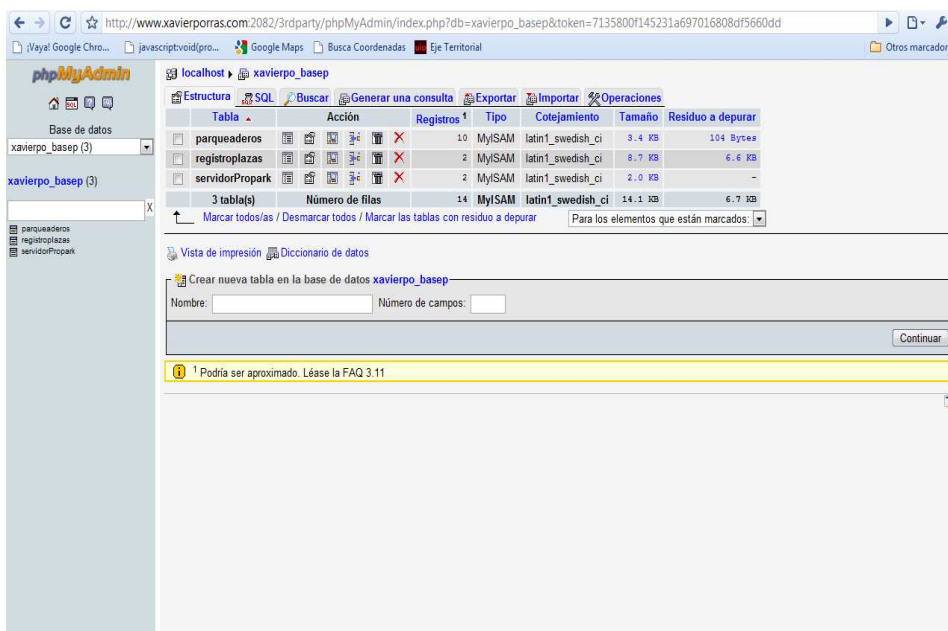


Figura 3-5. Pantalla phpMyAdmin: Base de datos subida en la web

3.4 DESARROLLO DE LA APLICACIÓN CLIENTE

Luego de la creación de la base de datos se desarrolló una aplicación de escritorio para ser instalada en un computador con conexión a internet dentro de cada estacionamiento con el objetivo de poder recopilar y actualizar la información

en la base datos con respecto a las plazas tanto libres y ocupadas de los recintos registrados al sistema prototipo.

3.4.1 ALGORITMO DE LA APLICACIÓN CLIENTE

La aplicación cuenta con la siguiente lógica de funcionamiento (Ver Figura 3-6):

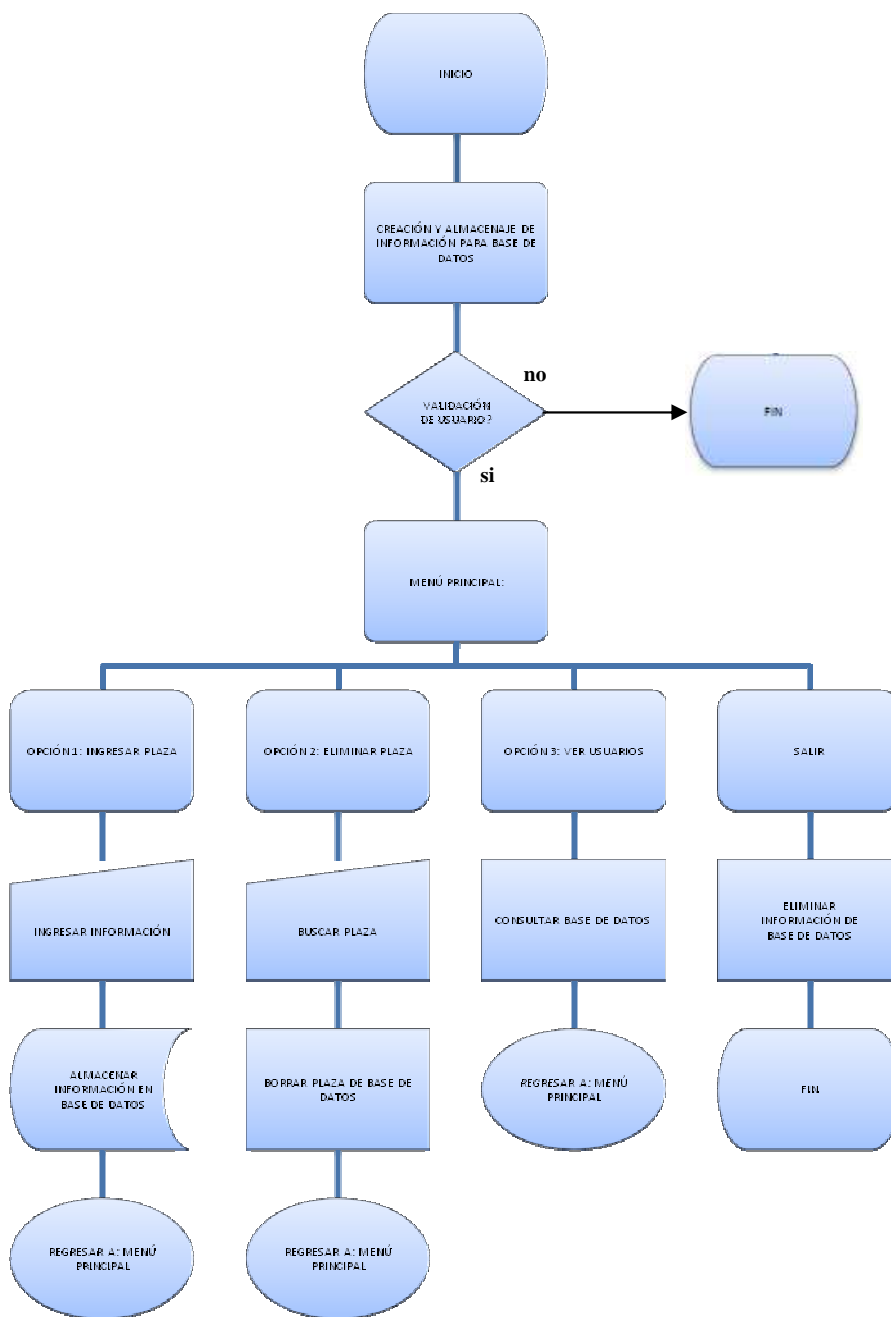


Figura 3-6. Algoritmo de funcionamiento: Aplicación cliente

3.4.2 SIMULACIÓN DE LA INFORMACIÓN PARA LA BASE DE DATOS

Independientemente de cómo se encuentren automatizados los estacionamientos, esta aplicación buscará recopilar información de disponibilidad, la cual permita integrar al parqueadero dentro del sistema prototipo.

Al ser un sistema prototipo se consideró idóneo simular de manera aleatoria, mediante una función 'random', el número de plazas ocupadas sobre un total de cien (100) espacios disponibles dentro del recinto automotor, con el fin de poder observar la actualización del número de espacios disponibles, y por ende ocupados en el campo respectivo de la tabla asignada en la base de datos.

Luego basándonos en este número aleatorio se crearon los datos referentes a las plazas ocupadas los cuales son guardados en la respectiva tabla; para esto se diseñó y programó una interfaz (Ver Figura 3-7) que se muestra mientras se cumple el proceso mencionado.



Figura 3-7. Pantalla de inicio a la aplicación de escritorio para los parqueaderos

3.4.3 SEGURIDAD DE LA APLICACIÓN CLIENTE

El factor seguridad, con respecto a la manipulación de los datos, se lo aseguró mediante una pantalla de identificación, la misma que solicita un nombre de usuario y una contraseña (Ver Figura 3-8). Si el número de intentos erróneos excede las tres veces la aplicación automáticamente se cierra.



Figura 3-8. Pantalla de ingreso a la aplicación cliente

3.4.4 OPCIONES DE LA APLICACIÓN CLIENTE

Si el usuario y la contraseña son correctos la aplicación es redirigida a la ventana de menú principal (Ver Figura 3-9) dentro de la cual se tiene cuatro opciones:

- **INGRESAR PLAZA:** Opción que permite documentar los datos esenciales de un automotor como lo son: placa, plaza asignada, fecha y hora de ingreso. Además se utilizó un ID (identificación) para poder reconocer en que parqueadero se realizó dicho ingreso de datos.
- **ELIMINAR PLAZA:** Opción que permite buscar, a través del número de placa del automotor, la información necesaria para la posterior facturación del servicio prestado.

NOTA: Valga la pena aclarar que el objetivo de esta aplicación no es el brindar un sistema de facturación; sin embargo, esto no sería un

impedimento al momento de realizar una futura implementación integral del sistema ya que es algo solucionable mediante programación.

- VER USUARIOS: Opción destinada a poder visualizar de manera general todos los vehículos registrados en el respectivo parqueadero.
- SALIR: Opción que finaliza la aplicación y borra la información generada dentro de la base de datos dejando espacio en la misma para una futura simulación.

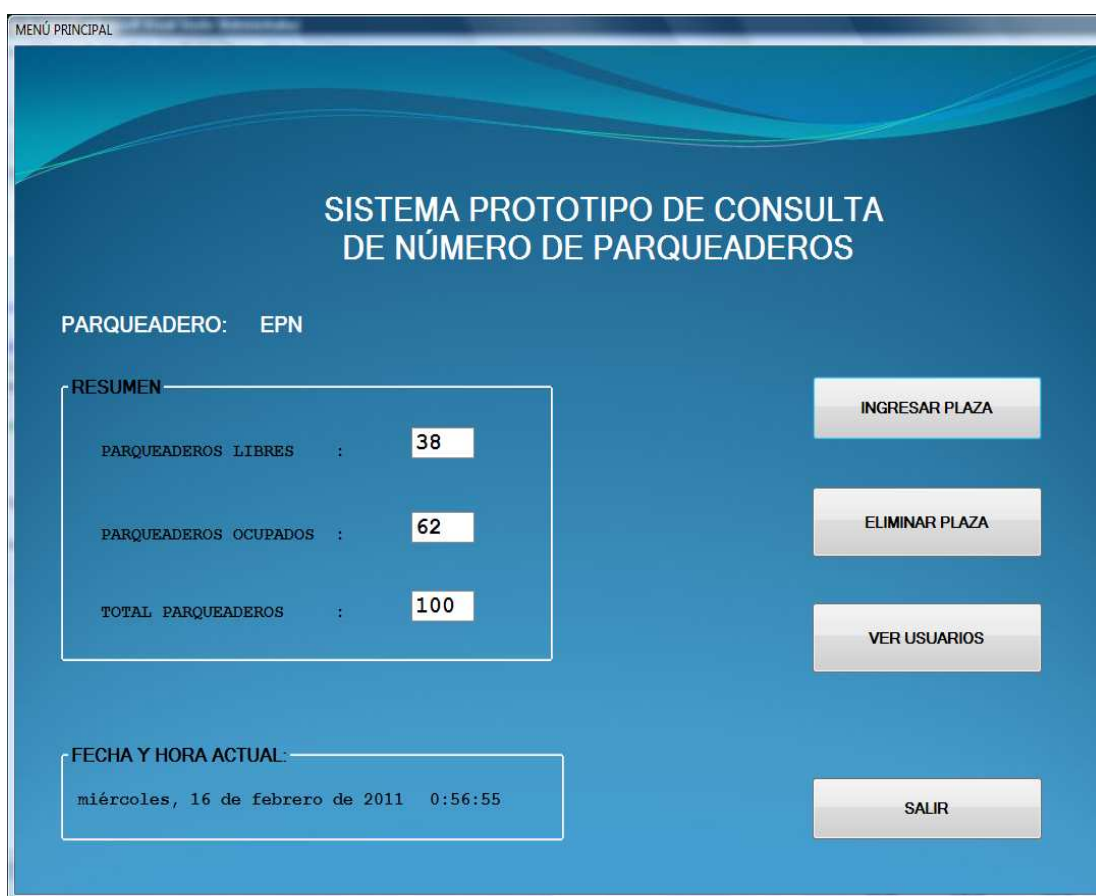


Figura 3-9. Pantalla menú principal

Si se selecciona la opción INGRESAR PLAZA se desplegará una pantalla con la cual se puede registrar los datos del automotor a ser ingresado (Ver Figura 3-10):

INGRESO DE PLAZA

PARQUEADERO: EPN

PARQUEADEROS

DISPONIBLES: 38

OCUPADAS: 62

DATOS A INGRESAR

NÚMERO DE PLACA : ASD-1234

PLAZA ASIGNADA : 8

HORA DE INGRESO : 0:59:06

FECHA : miércoles, 16 de febrero de 2011

<< REGRESAR INGRESAR ID: EPN

Figura 3-10. Pantalla opción: Ingreso de plaza

Si se escoge la opción ELIMINAR PLAZA aparecerá una pantalla (Ver Figura 3-11), con la cual se podrá buscar al automotor mediante su placa dentro del sistema, y luego de encontrarla pasar al proceso de facturación dentro del cual se habilitará la plaza ocupada:

ELIMINAR

FACTURAR TIEMPO

PARQUEADERO: EPN

RESULTADO DE BUSQUEDA

ID : EPN

PLACA : ABC-000

FECHA : 16/02/2011

HORA DE INGRESO : 1:20:42

HORA DE SALIDA : 1:23:33

PLAZA : 50

PLACA: ABC-000_ BUSCAR...

FACTURAR

<< REGRESAR

PARQUEADEROS

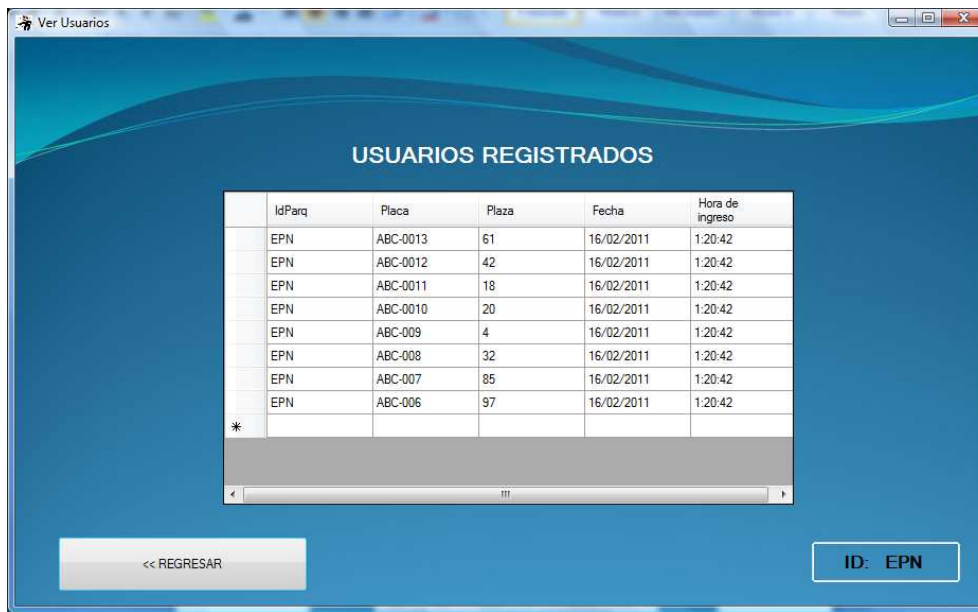
DISPONIBLES: 86 OCUPADAS: 14

FECHA/HORA

miércoles, 16 de febrero de 2011 1:23:42

Figura 3-11. Pantalla opción: Eliminar plaza

Si se selecciona la opción VER USUARIOS se desplegará una pantalla (Ver Figura 3-12), con la cual se podrá observar todos los usuarios registrados dentro del recinto:



The screenshot shows a window titled 'Ver Usuarios' with a blue background. At the top center, it says 'USUARIOS REGISTRADOS'. Below this is a table with the following data:

IdParq	Placa	Plaza	Fecha	Hora de ingreso
EPN	ABC-0013	61	16/02/2011	1:20:42
EPN	ABC-0012	42	16/02/2011	1:20:42
EPN	ABC-0011	18	16/02/2011	1:20:42
EPN	ABC-0010	20	16/02/2011	1:20:42
EPN	ABC-009	4	16/02/2011	1:20:42
EPN	ABC-008	32	16/02/2011	1:20:42
EPN	ABC-007	85	16/02/2011	1:20:42
EPN	ABC-006	97	16/02/2011	1:20:42
*				

At the bottom left, there is a button labeled '<< REGRESAR'. At the bottom right, there is a button labeled 'ID: EPN'.

Figura 3-12. Pantalla opción: Ver usuarios

Si se selecciona la opción SALIR se tendrá una pantalla (Ver Figura 3-13), tras la cual se realiza el proceso de eliminación de los datos generados durante la utilización de la aplicación:

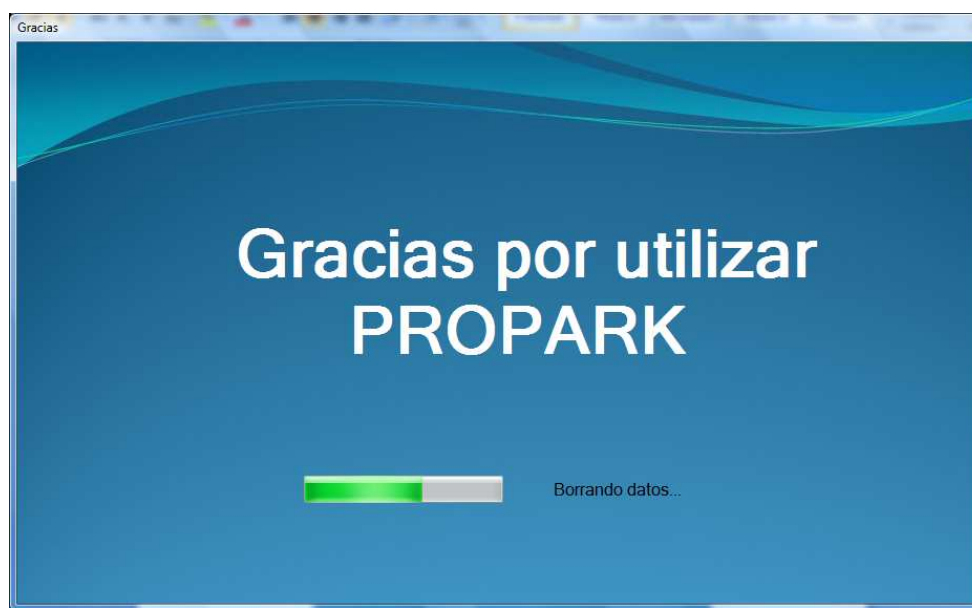


Figura 3-13. Pantalla de salida

3.4.5 ALARMAS DE DISPONIBILIDAD

Además, dentro del menú principal se consideró factible el colocar alarmas de disponibilidad alto y bajo con respecto al número de parqueaderos libres, siendo el número 20 referencia para la alarma de disponibilidad baja (Ver Figura 3-14) y el número 80 para disponibilidad alta (Ver Figura 3-15). De igual manera, se colocaron alarmas si el parqueadero está lleno (Ver Figura 3-16) o está vacío (Ver Figura 3-17) con el fin de alertar al administrador y así poder tomar medidas de comunicación, con respecto a la disponibilidad, a los usuarios que se encuentren en las afueras del recinto a la espera de este servicio.



Figura 3-14. Pantalla: Alarma plazas libres por terminarse

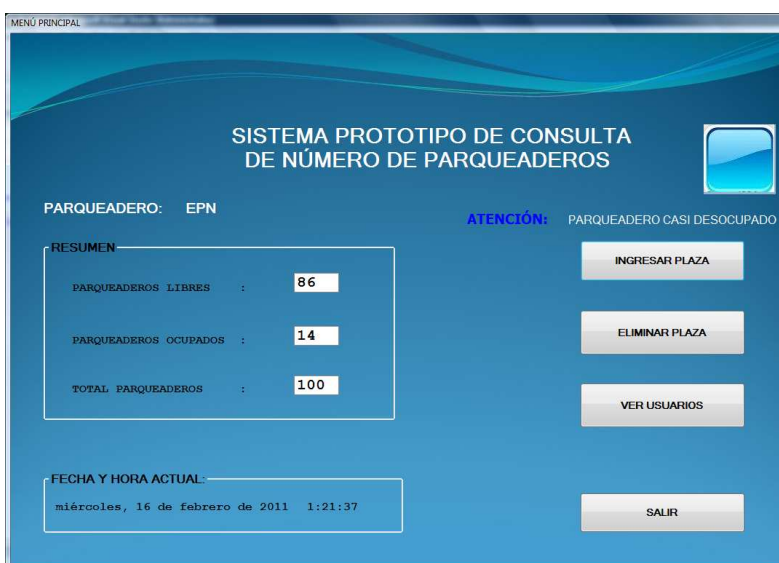


Figura 3-15. Pantalla: Alarma parqueadero casi desocupado

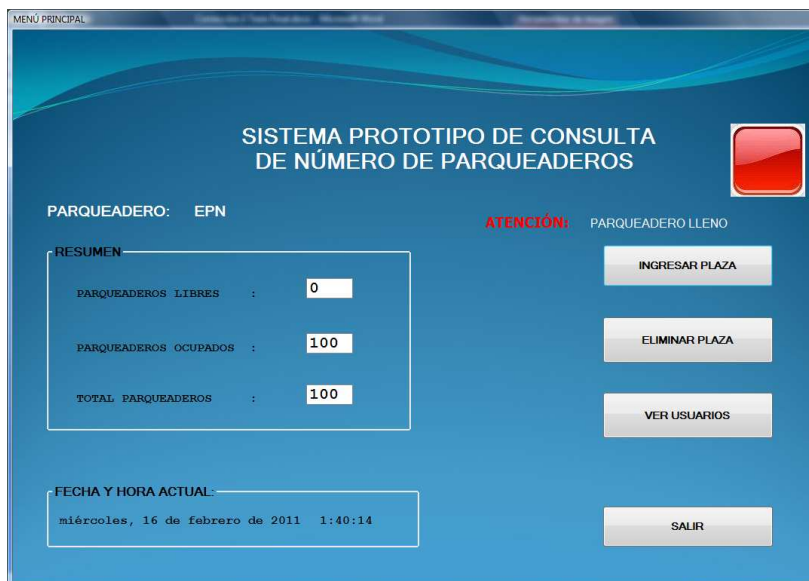


Figura 3-16. Pantalla: Alarma parqueadero lleno



Figura 3-17. Pantalla: Alarma parqueadero vacío

Como medida de seguridad también se implementaron alarmas en el momento del ingreso de dos placas similares en un mismo parqueadero o en otro parqueadero registrado en el sistema, mediante la validación del campo denominado “NÚMERO DE PLACA” (Ver Figura 3-18). Con el fin de salvaguardar la integridad de los automotores registrados.

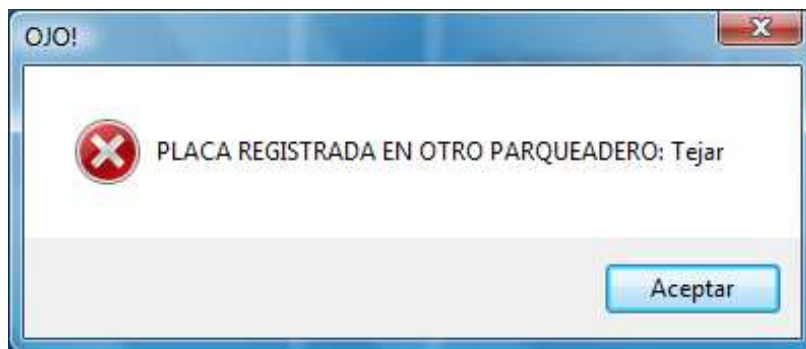


Figura 3-18. Mensaje de validación si la placa ya está registrada

3.5 DESARROLLO Y CONFIGURACIÓN DE LA APLICACIÓN SERVIDOR

El servidor, quien cuenta con un modem GSM para la recepción de mensajes SMS mediante comandos AT¹⁸ (Ver Figura 3-19), cuenta con una aplicación que lleva su mismo nombre, la cual valida la información del mensaje de texto recibido y luego con la ayuda de una algoritmo acondiciona los datos para, en lo posterior, compararlos con las ubicaciones de los parqueaderos registrados en la base de datos y finalmente devolver el número de plazas disponibles del parqueadero más cercano al equipo que realizó la petición en otro mensaje de texto.



¹⁸ Significan: Attention

Figura 3-19. Módem GSM conectado al servidor

3.5.1 INTERFAZ CON MÓDEMS [31]

La comunicación con los módems se realiza a través de una línea serie, y dependiendo del módem, se pueden usar los niveles definidos por la norma RS232 (Módems para PC).

El estándar “de facto” para controlar los módems se basa en los comandos AT HAYES, o comúnmente conocidos como comandos AT.

El módem, antes de realizar una conexión con otro módem, se encuentra en modo comando. En este modo se puede configurar y controlar el módem utilizando los comandos AT.

Para esto se sugiere el tener claro tres puntos:

- 1 el lugar de almacenamiento en el módem de los mensajes de texto entrantes,
- 2 tomar en cuenta que la mayoría de módems no tienen activado el “Eco” de los caracteres que se le envían y;
- 3 para probar la comunicación debe estar seleccionado el modo test.

Una vez establecida la conexión con un módem remoto, se pasa del modo comando al modo conexión, por lo que la información que le llega al módem por la línea serial no es interpretada como comandos AT sino como información a transmitir. Una vez terminada la conexión el módem vuelve al modo comando.

Los comandos AT son cadenas ASCII¹⁹ que comienzan por los caracteres “AT” y terminan con un retorno de carro.

Cada vez que el módem recibe un comando, lo procesa y devuelve un resultado, que normalmente es una cadena ACII salvo que hayamos indicado lo contrario.

Al estar la comunicación en ASCII, podremos utilizar un terminal de comunicaciones desde un ordenador para acceder al módem, bien para configurarlo, hacer pruebas o para establecer una comunicación con otro módem (Ver Figura 3-20).

¹⁹ American Standard Code of Information Interchange

```

+ZEND
+CMTI: "SM",0
at+cmgl="all"
ERROR
at+cmgs="sm"
+CMS ERROR: 304
at+cpms="sm"
+CPMS: 3,8,0,100,3,8

OK
at+cmgf=1
OK
at+cmgr=0
+CMGR: "REC UNREAD", "8000", "11/02/04,13:44:01-20"
00116F2E77053403670D800042006F0072007200610072650D800056006F006C0076006500726413
80005300690067007500690065006E00740065006103C00053C050300003C00B400B403080210
44140B000900090B0D0E0F1011121503000100160300010003080E03054000000060

OK
at+cmgf=0
OK

```

Figura 3-20. Prueba de funcionamiento del módem

3.5.2 INTERFAZ CON MÓDEMS GSM

Los módems GSM no sólo se comportan de forma muy parecida a un modem normal, permitiendo el intercambio de datos con otro módem y utilizándose los comandos AT originales, sino que incluyen muchas más características.

Son como pequeños teléfonos móviles, que incluyen su propia tarjeta SIM (Subscriber Identity Module) para poder funcionar y por tanto permiten, entre lo más importante, gestionar la base de datos de teléfonos, la lista de los mensajes SMS recibidos, enviar mensajes SMS y configurar diversos parámetros.

Para tener acceso a todos esos servicios, y dado que los comandos AT estaban muy extendidos y muy estandarizados, se ha realizado una ampliación, añadiéndose nuevos comandos denominados comandos AT+.

3.5.3 COMANDOS AT

En este apartado se listan algunos de los comandos "AT" implementados en los módems GSM para tener una idea de lo que se puede controlar a través del módem, aunque existen muchos más comandos.

- *Comandos generales:*
 - AT+CGMI: Identificación del fabricante
 - AT+CPAS: Leer estado del módem
- *Comandos del servicio de red:*
 - AT+CSQ: Obtener calidad de la señal
 - AT+WOPN: Leer nombre del operador
- *Comandos para SMS:*
 - AT+CPMS: Seleccionar lugar de almacenamiento de los SMS
 - AT+CMGF: Seleccionar formato de los mensajes SMS
 - Modo texto
 - Modo PDU (Protocol Description Unit)
 - AT+CMGR: Leer un mensaje SMS almacenado
 - AT+CMGL: Listar los mensajes almacenados
 - AT+CMGS: Enviar mensaje SMS

3.5.3.1 Listado de mensajes

Los SMS se dividen en 5 categorías, cada una identificada por una cadena. Por ejemplo, para listar los mensajes se utiliza el comando AT+CMGL=<categoría>, donde <categoría> es una cadena de texto que puede valer lo siguiente:

- “REC UNREAD”: Mensajes recibidos pero no leídos
- “REC READ”: Mensajes recibidos y leídos
- “STO SENT”: Mensajes enviados
- “ALL”: Todos los mensajes

3.5.4 ALGORITMO DE LA APLICACIÓN SERVIDOR

Luego de tener claro los conceptos mencionados se procedió a desarrollar la aplicación de escritorio para el servidor.

Esta aplicación sigue el siguiente patrón lógico (Ver Figura 3-21):

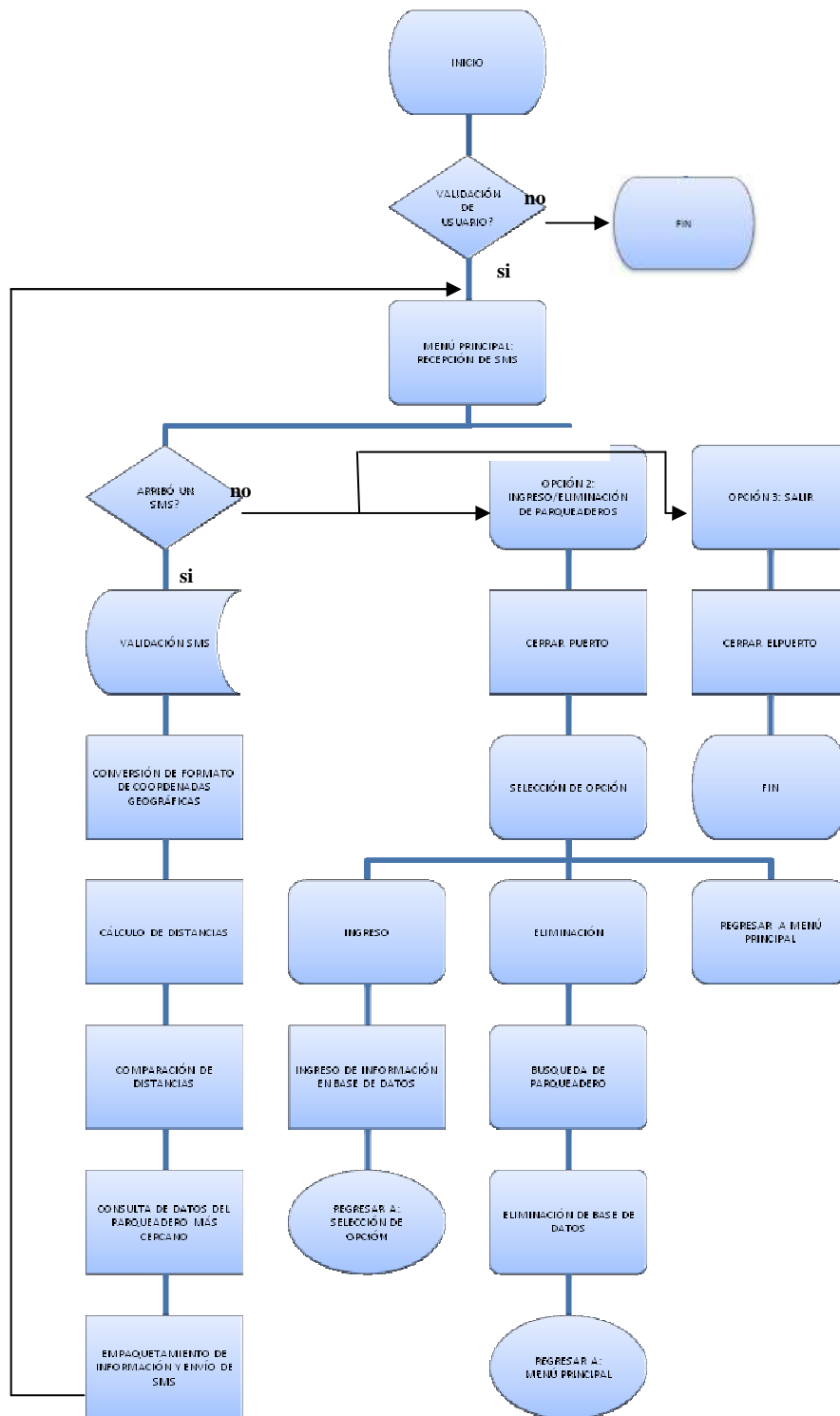


Figura 3-21. Algoritmo de funcionamiento: Aplicación servidor

3.5.5 SEGURIDAD DE LA APLICACIÓN SERVIDOR

Una vez iniciada la aplicación, esta despliega una pantalla de presentación (Ver Figura 3-22) luego de la cual se pasa al proceso de validación del administrador,

ya que la información que maneja esta aplicación es muy delicada y podría causar fallas en el normal desempeño del programa si es manipulado por personal no calificado.

Si el usuario y la clave son los correctos la aplicación se re-direcciona a la pantalla de menú principal la misma que en primera instancia realiza una consulta de información referente a los parqueaderos registrados a la base de datos para luego recibir los mensajes enviados desde el PDA (Ver Figura 3-23).

Para esta última parte mientras se desarrollaba el software se utilizó como herramienta de comprobación, de la recepción correcta de los mensajes, el programa propietario del modem GSM (Ver Figura 3-24).



Figura 3-22. Pantalla de inicio a la aplicación de escritorio para el servidor



Figura 3-23. Pantalla de ingreso a la aplicación servidor: Identificación de usuario

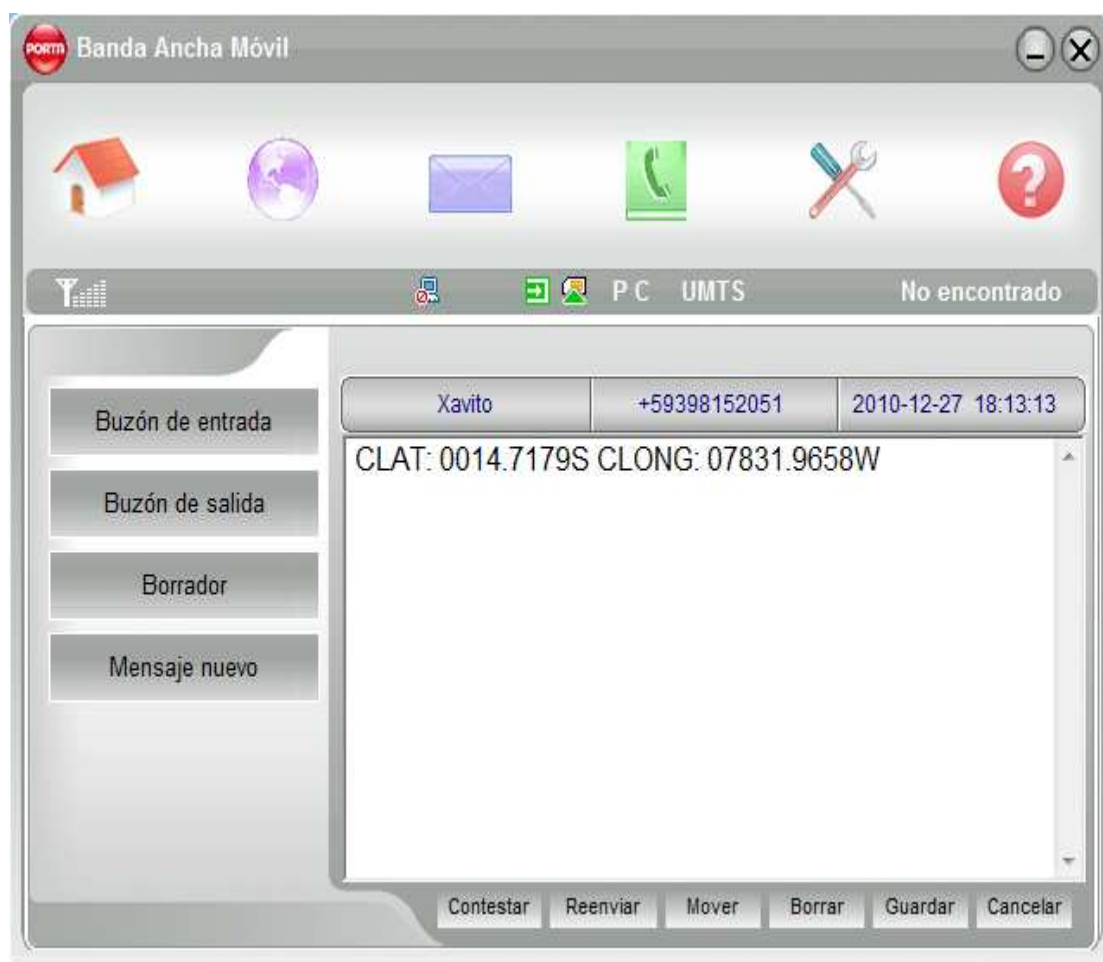


Figura 3-24. Software propietario de comprobación del modem GSM

3.5.6 PROCESAMIENTO DE LA TRAMA SMS

Una vez hecha la consulta se exhiben los datos obtenidos respecto a los parqueaderos registrados para este prototipo (Ver Figura 3-25).

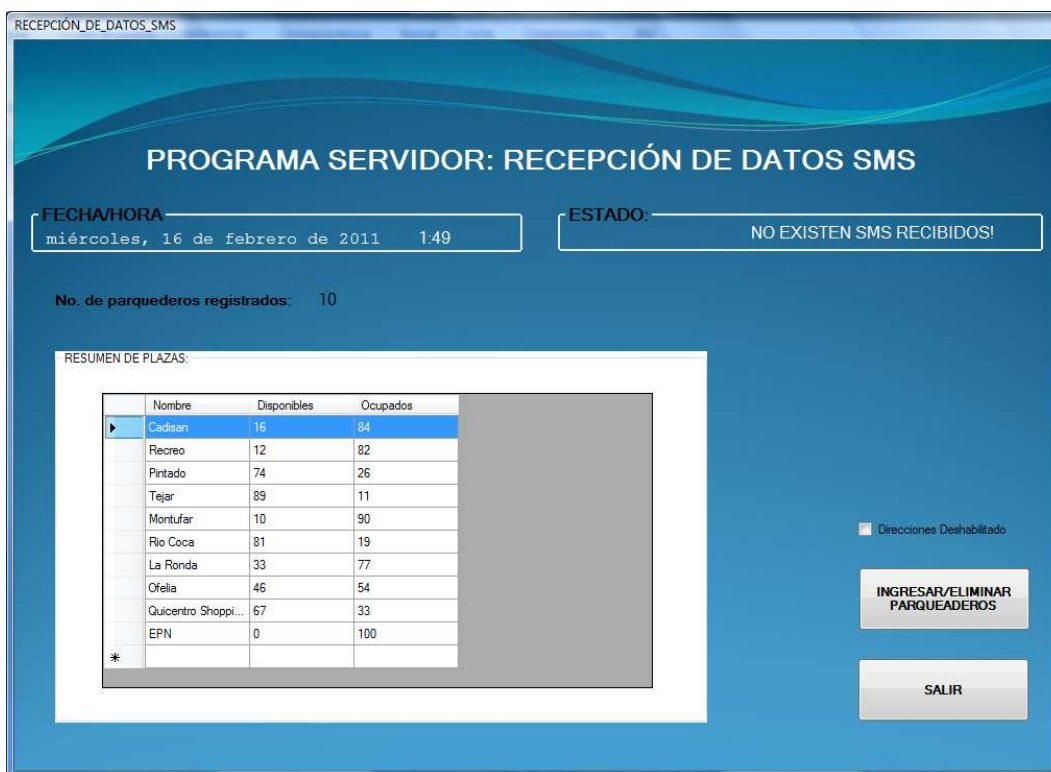


Figura 3-25. Interface para aplicación de escritorio “servidor”

Si un SMS arribó al modem GSM, la aplicación procede a validar los datos recibidos para luego cambiarlos de formato de coordenadas ya que normalmente están en grados, minutos y segundos y se necesita que se encuentren en radianes.

El algoritmo de cálculo de distancias utiliza el principio denominado como “Cálculo de distancias en grandes círculos”, método basado en la fórmula de Haversine²⁰ utilizada por grandes desarrolladores como es el caso de MySql y Google.

En este punto es importante resaltar que la automatización de este proceso busca brindar una buena precisión debido a que se logra una disminución de los errores

²⁰ R.W. Sinnott, “Virtues of the Haversine”, Sky and Telescope, vol. 68, no. 2, 1984, p. 159

manuales o su corrección inmediata al utilizar equipos y tecnología estándar, de donde se determina que prácticamente el error obtenido es el que viene de facto por parte del GPS/Bluetooth.

3.5.6.1 Cálculo de la distancia en grandes círculos [33]

La distancia circular extensa (D) sobre la esfera entre dos puntos A y B se puede calcular usando una fórmula estándar en trigonometría esférica (Ver Ecuación 3-1):

$$\cos D = (\sin a \times \sin b) + [\cos a \times \cos b \times \cos(\delta\lambda)]$$

Ecuación 3-1

Donde:

a y b son las latitudes geográficas de A y B,

($\delta\lambda$) es el valor absoluto de la diferencia en longitud entre A y B.

Aproximadamente 1 grado sobre el gran círculo es igual a 111 mil kilómetros, y se puede multiplicar esa constante por el valor obtenido en grados, para obtener la distancia lineal. Una estimación del elipsoide permite calcular la elevación de cada punto sobre la tierra, incluyendo el nivel del mar.

Es posible formular un modelo matemático de la superficie del nivel del mar de la tierra. Sin embargo, el modelo es muy complejo y no es apropiado para la registración de posiciones geográficas de atributos.

El nivel del mar no coincide con una simple superficie matemática, tal como una esfera o un esferoide. Esto es importante porque en un área pequeña, en un modelo de "tierra plana", el plano horizontal que contiene los ejes N²¹ y E²² se posiciona para que coincida con el nivel del mar.

²¹ Norte

²² Este

Esto no es posible para un 'modelo curvo de la tierra' debido a la irregularidad de la superficie del nivel del mar. En cambio, para una pequeña parte del esferoide, la superficie del nivel del mar coincide con él (Ver Figura 3-26). [34]

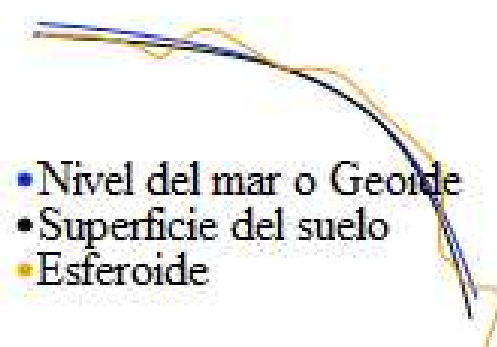


Figura 3-26. Comparación del geoide y el esferoide

Bajo estos preceptos, en la práctica, para obtener la distancia entre dos puntos de la Tierra, se debe partir de los datos de latitud y longitud de los puntos en cuestión. Luego del cambio de forma de las coordenadas se las aplica en la Ecuación 3-2. Para ello se debe multiplicar los grados (entero + decimales) por 0,01745329252 ($\pi/180$)

Se debe recordar que latitud "sur" y longitud "oeste" se consideran valores negativos, en tanto que latitud "norte" y longitud "este" serán valores positivos. El cálculo a realizar es el siguiente:

$$P = \text{Seno}(\text{latitud } 1) * \text{Seno}(\text{latitud } 2) + \text{coseno}(\text{latitud } 1) * \text{coseno}(\text{latitud } 2) * \text{coseno}(\text{longitud } 2 - \text{longitud } 1)$$

Ecuación 3-2

Con el resultado anterior P se procede a calcular la Ecuación 3-3:

$$D = ACOS(P)$$

Ecuación 3-3

3.5.6.2 Simulación del Cálculo de distancias

Como una herramienta de comprobación de cálculo al momento de cambiar el formato de las coordenadas se utilizó la aplicación Microsoft Office Excel, donde primero se hizo la conversión de grados, minutos y segundos a solo grados (Ver Figura 3-27) para luego multiplicar este valor por una constante y finalmente insertarla en la ecuación donde se debió considerar que la función ACOS() devuelve el valor en radianes, y lo que se necesitaba era que se lo haga en grados. Por lo tanto se debió hacer la conversión de grados a radianes multiplicando el resultado por 57,29577951 ($180/\pi$).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		grados	min	seg		60s = 1min	total min	1° = 60min	TOTAL GRAD		grados	erados	grados	grados
2	latitud 1	0	14	42.8886 S		0.71481	14.71481	0.2452468	0.2452468	latitud 1	-0.2452468	0	0	0
3	longitud 1	78	31	58.0224 O		0.96704	31.96704	0.532784	78.532784	longitud 1	0	-78.532784	0	0

Latitud y longitud en solo grados

Figura 3-27. Ejemplo de conversión de formato de coordenadas

Posteriormente a estos valores obtenidos se los multiplica por una constante consiguiendo de esta manera los datos de latitud y longitud en radianes, luego se los reemplaza en la ecuación 3-2 y 3-3, respectivamente, para obtener así la distancia entre los dos puntos seleccionados.

Este último resultado se lo puede expresar en kilómetros mediante la operación de la ecuación 3-4. (Ver Figura 3-28)

$$Km = D * 111,302$$

Ecuación 3-4

	en grados	CONSTANTE	EN RAD	P	D°	POR EL RT
LATITUD1	-0.24524683	0.0174532925199433000	-0.00428036472176	0.99999998253	0.000	1.19204739
LONGITUD1	-78.532784	0.0174532925199433000	-1.37065565155752			
LATITUD2	-0.25125783	0.0174532925199433000	-0.00438527646309			
LONGITUD 2	-78.5239218	0.0174532925199433000	-1.37050097757033			

Distancia obtenida en Km (precisión de 15 dígitos aunque en el gráfico solo aparecen 8)

Figura 3-28. Simulación del cálculo de la distancia realizado por PROPARK

Todos estos cálculos finalmente fueron implementados con ayuda de VS.NET y se realizó una comparación de la precisión del cálculo realizado por la aplicación de escritorio servidor (Ver Figura 3-28) y la que nos provee la página web http://www.tutiempo.net/p/distancias/calcular_distancias.html (Ver Figura 3-29) donde se puede determinar la ventaja del grado de precisión ofrecido por el dispositivo.

EN INTERNET HAY MILLONES DE FOTOGRAFÍAS, ¿PERO SALE

ESPAÑA 05 DE ENERO DEL 2011

Valencia 18°C 7°C

TuTiempo.net

EL TIEMPO CLIMA METEOROLOGÍA ASTRONOMÍA

MAPA TOPOGRÁFICO | CALCULAR DISTANCIAS | CURSOS | COMPRAS INTERNET | MOVIL

Calcula la distancia entre dos puntos de la Tierra

(punto de partida) entre dos puntos de la tierra, introduce las coordenadas de cada uno en los cuadros de texto. Si no tienes coordenadas, introduce el nombre del punto de partida y el punto final.

Datos para el cálculo

Punto de partida Latitud: -0.251258 Longitud: -78.523922 [Buscar coordenadas]

Punto final Latitud: -0.245247 Longitud: -78.532784 [Buscar coordenadas]

[Realizar cálculos]

Resultado del cálculo

Distancia 1.19 Kilómetros

Distancia .74 Millas

Precisión por parte de la página web consultada

Figura 3-29. Cálculo de distancia por www.tutiempo.net

3.5.7 OPCIONES DE LA APLICACIÓN SERVIDOR

Esta aplicación de escritorio, aparte de lo expuesto, permite observar las direcciones de los parqueaderos registrados (Ver Figura 3-30) así como ingresar (Ver Figura 3-31), eliminar (Ver Figura 3-32) los parqueaderos dentro de la base de datos ubicada en la web, ver toda la información de los mismos (Ver Figura 3-33) y cambiar su clave de acceso (Ver Figura 3-34), siempre y cuando se cuente con su respectiva conexión.

RECEPCIÓN_DE_DATOS_SMS

PROGRAMA SERVIDOR: RECEPCIÓN DE DATOS SMS

FECHA/HORA: miércoles, 16 de febrero de 2011 1:46

ESTADO: NO EXISTEN SMS RECIBIDOS!

No. de parqueaderos registrados: 10

RESUMEN DE PLAZAS:

Nombre	Direccion	Disponibles	Ocupados
Cadisan	CADISAN: Meja entre Garcia Moren...	16	84
Recreo	EL RECREO.- Av. Pedro Vicente Mal...	12	82
Pintado	EL PINTADO.- Jose Ignacio Canelos ...	74	26
Tejar	EL TEJAR.- Calle Lopez entre Herman...	9	11
Montufar	MONTUFAR.- Calle Montufar entre Ol...	0	90
Rio Coca	RIO COCA.- Av. Rio Coca y de las Pal...	1	19
La Ronda	LA RONDA.-sur de la Calle Morales e...	33	77
Ofelia	LA OFELIA.- entre Diego Vasquez de	46	54
Quicentro Shoppi...	QUICENTRO: Av. Naciones Unidas ...	67	33
EPN	E. N.-Ladron de Guevara E11-2F. J...	0	100

Direcciones Habilitado

INGRESAR/ELIMINAR PARQUEADEROS

SALIR

Figura 3-30. Pantalla: Programa servidor con direcciones habilitadas

Parqueaderos

ELIMINACIÓN E INGRESO DE PARQUEADEROS A LA BASE DE DATOS

ELIMINAR | INGRESAR | INFO PARQUEADEROS | CAMBIO DE CLAVE

INGRESO DE DATOS

NOMBRE (ID):

LATITUD: (Ejemplo: -0.218634)

LONGITUD: (Ejemplo: -78.512029)

PLAZAS OCUPADAS:

PLAZAS DISPONIBLES:

DIRECCIÓN:
(Se registrarán hasta 100 caracteres,
empezando con el ID en mayúsculas)
Ejemplo: CADISAN: Mejía entre García
Moreno y Benalcazar

CLAVE DE ACCESO:

Ingresar
Parqueadero

FECHA/HORA: miércoles, 16 de febrero de 2011 1:51:42

<< REGRESAR A RECEPCIÓN DE DATOS

Figura 3-31. Pantalla: Ingreso de parqueadero a la base de datos

Parqueaderos

ELIMINACIÓN E INGRESO DE PARQUEADEROS A LA BASE DE DATOS

ELIMINAR | INGRESAR | INFO PARQUEADEROS | CAMBIO DE CLAVE

PARQUEADERO:

BUSCAR PARQUEADERO

NOTA:
Para eliminar un parqueadero
primero realice la búsqueda
del mismo

Eliminar
Parqueadero

FECHA/HORA: miércoles, 16 de febrero de 2011 1:52:19

<< REGRESAR A RECEPCIÓN DE DATOS

Figura 3-32. Pantalla: Eliminación de parqueadero de la base de datos

Parqueaderos

ELIMINACIÓN E INGRESO DE PARQUEADEROS A LA BASE DE DATOS

ELIMINAR | INGRESAR | INFO PARQUEADEROS | CAMBIO DE CLAVE

Mostrar parqueaderos / Actualizar

No. de parqueaderos: 10

Nombre	Latitud	Longitud	Direccion	Disponibles	Ocupados	Clave
Cadisan	-0.218634	-78.512029	CADISAN: Mejia ...	16	84	1111
Recreo	-0.251258	-78.523922	EL RECREO - Av...	12	82	2222
Pintado	-0.245247	-78.532784	EL PINTADO - J...	74	26	3333
Tejar	-0.215815	-78.516165	EL TEJAR - Calle...	89	11	4444
Montufar	-0.220456	-78.508381	MONTUFAR - Ca...	10	90	5555
Rio Coca	-0.163502	-78.473995	RIO COCA - Av. ...	81	19	6666
La Florida	-0.224779	-78.51447	LA RONDA-sur...	33	77	7777
Ofelia	-0.108887	-78.48805	LA OFELIA - entr...	46	54	8888
Quicentro Shoppi...	-0.177068	-78.47826	QUICENTRO: Av...	67	33	9999
EPN	-0.208767	-78.488914	EPN -Ladron de ...	0	100	4321
*						

FECHA/HORA: miércoles, 16 de febrero de 2011 1:53:07

<< REGRESAR A RECEPCIÓN DE DATOS

Figura 3-33. Pantalla: Información total de los parqueaderos

Parqueaderos

ELIMINACIÓN E INGRESO DE PARQUEADEROS A LA BASE DE DATOS

ELIMINAR | INGRESAR | INFO PARQUEADEROS | CAMBIO DE CLAVE

PARQUEADERO:

BUSCAR PARQUEADERO

FECHA/HORA: miércoles, 16 de febrero de 2011 1:55:53

<< REGRESAR A RECEPCIÓN DE DATOS

Figura 3-34. Pantalla: Cambio de clave de acceso a las respectivas aplicaciones “cliente” del sistema.

Es importante mencionar que en el panel de control en la parte del dominio, los permisos de ingreso estén habilitados a través del “MySQL remota” para no tener posibles errores de falla de conexión con la base de datos (Ver Figura 3-35)



Figura 3-35. Mensaje de error del servidor si los permisos de ingreso no están habilitados.

3.6 DESARROLLO Y CONFIGURACIÓN DE LA APLICACIÓN PDA

Una característica importante a la hora de realizar interfaces de usuario en los PDAs es el tamaño muy reducido de su pantalla. Esta limitación obliga a que se tenga que colocar los componentes de las interfaces de una forma adecuada y prescindiendo de los componentes que no sean útiles, para proporcionar accesibilidad y una visibilidad homogénea debido a que el conductor del vehículo tiene poco tiempo para realizar la consulta.

La interacción del usuario con la aplicación se la realizará, en este caso, con toques o deslizamientos sobre una pantalla táctil para simular el movimiento del puntero como el de un ratón en un ordenador.

Los componentes de la interface son los mismos que se utilizan en las aplicaciones para los ordenadores personales, dependiendo de las herramientas utilizadas suelen haber ligeras diferencias, esto es debido a que VS.NET trabaja bajo la misma plataforma que el ambiente Windows.

3.6.1 ALGORITMO DE LA APLICACIÓN PDA

La aplicación desarrollada cuenta con el siguiente algoritmo funcional (Ver Figura 3-36):

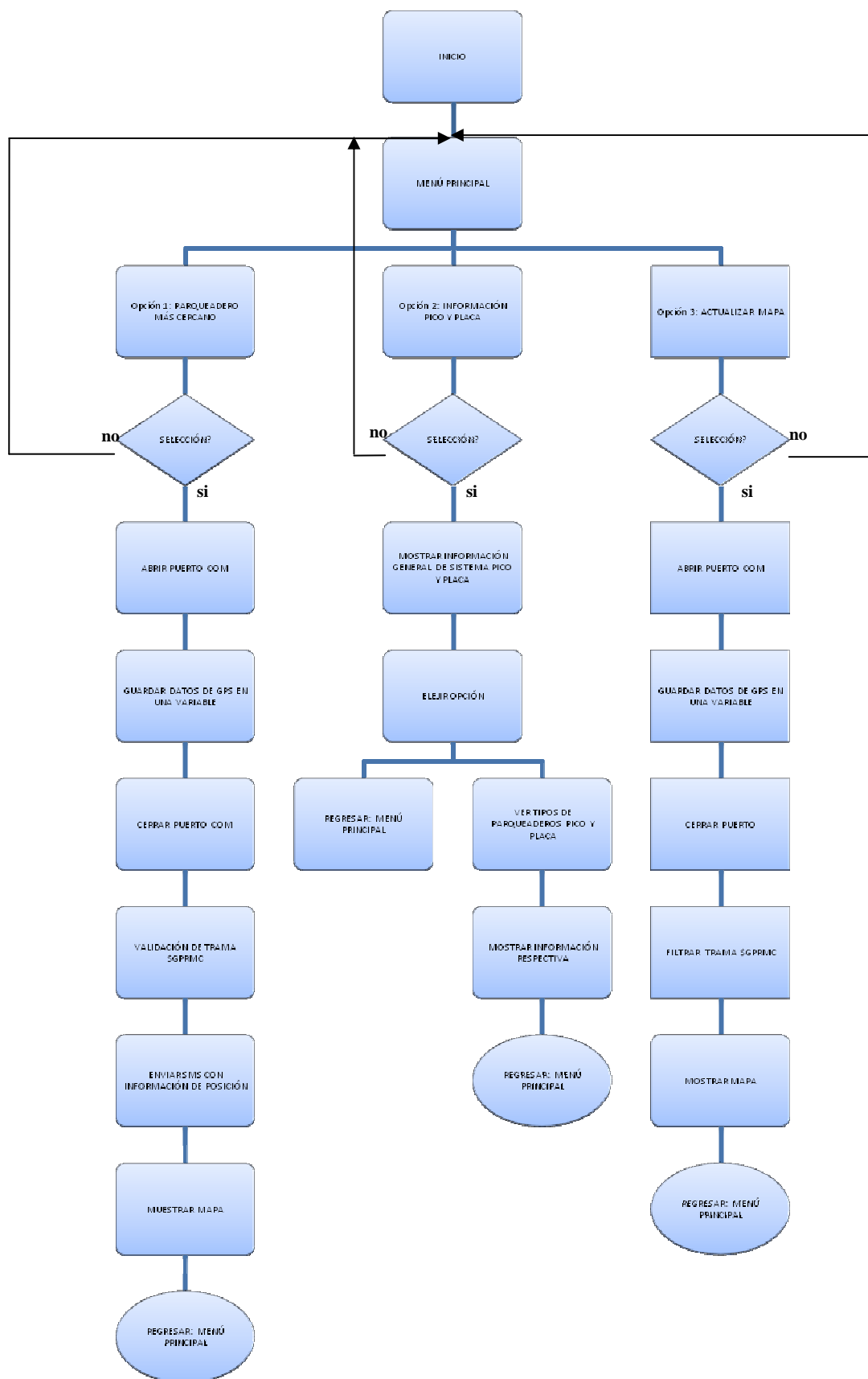


Figura 3-36. Algoritmo: Aplicación PDA

Al momento de ingresar a la aplicación mediante el ícono del menú principal del PDA (Ver Figura 3-37), se muestra al usuario una pantalla de inicio (Ver Figura 3-38) y luego una pantalla con el menú principal (Ver Figura 3-39).



Figura 3-37. Pantalla del menú de Inicio de Windows Mobile



Figura 3-38. Pantalla de Inicio de la aplicación PDA



Figura 3-39. Menú principal aplicación PDA

3.6.2 TRAMA PARA ENCAPSULADO DE DATOS

Al colocarse sobre la opción “PARQUEADERO MÁS CERCANO” se realiza una petición la cual busca en primera instancia establecer conexión Bluetooth con el dispositivo GPS. De ser efectiva esta conexión, la aplicación recibe los datos enviados por el GPS y los filtra, adquiriendo una trama a la que se la procesa para finalmente validar los datos de latitud y longitud.

Estos datos son encapsulados bajo el siguiente formato (Ver Tabla 3-4):

Tabla 3-4. Formato de trama para encapsulado de datos

LAT: ±dd.mmmm LONG: ±dd.mmm	
Donde:	
LAT	= latitud
LONG	= longitud
d	= degree (grados)
m	= minute (minutos)

3.6.3 PROCESO DE CONSULTA (ENVÍO Y RECEPCIÓN DE DATOS)

Luego estos datos son enviados en la trama creada mediante un SMS al servidor (Ver Figura 3-40), como muestra de que el proceso fue realizado con éxito se muestra un mensaje en pantalla (Ver Figura 3-41).

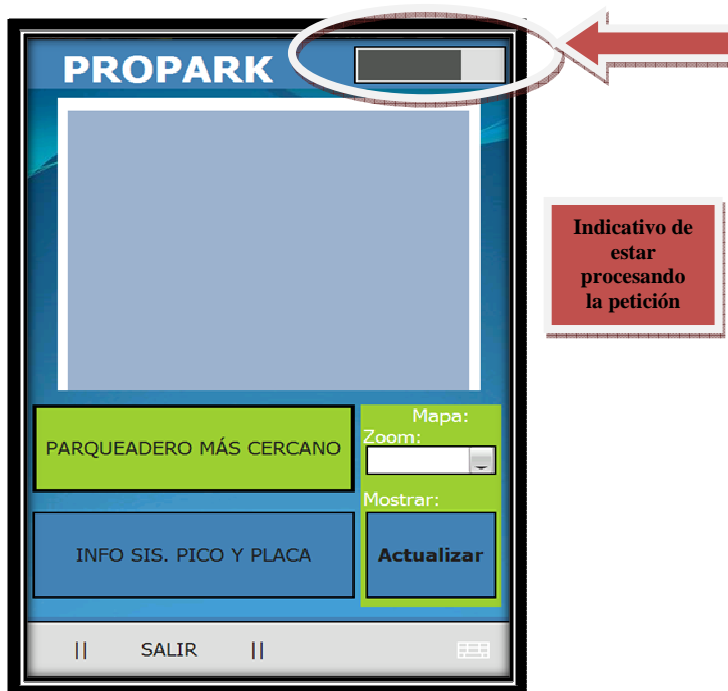


Figura 3-40. Petición de parqueadero más cercano

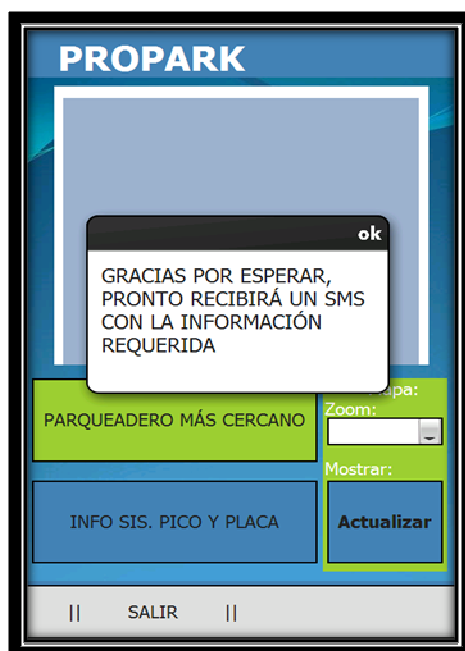


Figura 3-41. Mensaje de confirmación de petición realizada

Luego de haber hecho la petición se observa, mientras se espera la respuesta del servidor (Ver Figura 3-42), la localización de automotor en el momento que se realizó la petición mediante la ayuda del servicio Google Static Maps API²³ (Ver Figura 3-43).



Figura 3.42. SMS enviado por el servidor PROPARK



Figura 3-43. Pantalla de visualización de la ubicación del automotor

²³ Interfaz de programación de aplicaciones

3.6.3.1 Mapas para dispositivos móviles [35]

Los mapas para dispositivos móviles presentan varios problemas debido a que el espacio en pantalla es pequeño, razón por la cual los mapas que se pueden utilizar en un dispositivo móvil no pueden ser tan detallados. Además, muchos de los dispositivos móviles más completos disponen de resoluciones altas para el tamaño de su pantalla, por lo que, por ejemplo, el texto aparece a menudo en tamaños de fuentes más pequeños.

Google ofrece conjuntos de mosaicos especializados para dispositivos móviles para casos prácticos como este.

El API de Google Static Maps permite insertar una imagen de Google Maps en un Web Browser bajo petición. Este servicio envía una imagen de un mapa a partir de los parámetros URL²⁴ modificables mediante una solicitud HTTP²⁵ siempre y cuando se cuente con un plan de datos.

Para cada solicitud se especifica la ubicación del mapa, el tamaño de la imagen, el nivel de zoom, el tipo de mapa y la colocación de marcadores opcionales en lugares determinados del mapa. Además, se puede etiquetar los marcadores con caracteres alfanuméricos, de modo que se puede hacer referencia a ellos en una “clave”.

Cuando se muestra la página web, el navegador solicitará la imagen al API de Google Static Maps y la trazará en la ubicación de la imagen.

Se debe tener en cuenta que los mapas estáticos solo se pueden mostrar en el contenido del navegador; no se permite el uso de mapas estáticos fuera del navegador.

La URL de Google Static Maps API debe tener el siguiente formato:

<http://maps.google.com/maps/api/staticmap?parameters>

²⁴ Uniform Resource Locator, utilizado para especificar un objeto en internet

²⁵ Hypertext Transfer Protocol, protocolo de comunicaciones usado en cada transacción de la web

Algunos parámetros son obligatorios y otros opcionales. Como en las URL estándar, todos los parámetros se separan con el carácter (&).

A continuación, se muestra una lista de los parámetros usados:

- **Parámetros de ubicación:**

center (obligatorio) define el centro del mapa, equidistante de todos los bordes. Este parámetro toma una ubicación como un par separado por comas {latitud,longitud} o como una dirección de cadena identificando una ubicación única en la superficie de la tierra.

zoom (obligatorio) define el nivel de zoom del mapa, que determina su nivel de ampliación. Este parámetro toma un valor numérico correspondiente al nivel de zoom de la región deseada.

- **Parámetros de mapas:**

size (obligatorio) define las dimensiones rectangulares de la imagen del mapa. Este parámetro adopta la forma de una cadena valuexvalue, donde los píxeles horizontales se indican en primer lugar y los verticales en segundo lugar. Si crea un mapa estático con un ancho de 100 píxeles o superior, se reducirá automáticamente el tamaño del logotipo "Con la tecnología de Google".

format (opcional) define el formato de la imagen resultante. De forma predeterminada, el API de Google Static Maps crea imágenes PNG. Hay varios formatos posibles disponibles, incluidos los tipos GIF, JPEG y PNG. El formato a utilizar dependerá de cómo se desee presentar la imagen. JPEG suele proporcionar un mayor grado de compresión, mientras que GIF y PNG proporcionan mayor nivel de detalle.

maptype (opcional) define el tipo de mapa que se va a generar. Hay varios valores de tipo de mapa posibles disponibles, incluidos roadmap, satellite, hybrid y terrain.

mobile (obligatorio) especifica si el mapa se mostrará en un dispositivo móvil. Los valores válidos son true o false. Los mapas que se muestran en

dispositivos móviles pueden utilizar diferentes conjuntos de mosaicos optimizados para estos dispositivos.

- **Parámetros de función:**

markers (opcional) define uno o varios marcadores para adjuntarlos a la imagen en ubicaciones especificadas. Este parámetro adopta una definición de marcador única con parámetros separados por plecas (|). Se pueden colocar varios marcadores en el mismo parámetro **markers** cuando tienen el mismo estilo; se puede añadir marcadores adicionales de diferentes estilos mediante la adición de parámetros **markers** complementarios. Se debe tener en cuenta que si suministra marcadores para un mapa, no se tendrá que especificar los parámetros *center* ni *zoom* (que normalmente son obligatorios).

- **Notificación de parámetros:**

sensor (obligatorio) especifica si la aplicación que solicita el mapa estático va a utilizar un sensor (por ejemplo, un localizador GPS) para determinar la ubicación del usuario. Este parámetro es obligatorio para todas las solicitudes de mapas estáticos.

Además es importante saber a cerca de:

- **Latitudes y longitudes**

Las latitudes y longitudes se definen mediante números dentro de una cadena de texto separado con comas de seis posiciones decimales de precisión. Los niveles de precisión superiores al sexto decimal se ignoran. Los valores de longitud se basan en su distancia a Greenwich (Inglaterra), por donde pasa el meridiano cero. Las latitudes pueden tener cualquier valor entre -90 y 90, mientras que la longitud debe estar comprendida entre -180 y 180. Si especificas un valor de latitud o de longitud no válido, la solicitud será rechazada como una solicitud incorrecta.

3.6.4 INFORMACIÓN REFERENTE AL SISTEMA PICO Y PLACA [36]

En función de las capacidades de almacenamiento del PDA entre otras opciones del software se creó la opción de información respecto al sistema pico y placa, proporcionando en primera instancia información sobre el día y con este dato de forma automática el número de placa restringida durante el mismo; de ser el caso Sábado o Domingo se muestran las letras “NO” las cuales representan la expresión: “No existe restricción”

En la parte central de la ventana se visualiza el perímetro y el horario de restricción.

Finalmente se provee la opción de acceder a mayores datos referentes a los diferentes tipos de parqueaderos dispuestos por el cabildo regente para esta medida y la opción de regresar a realizar una nueva consulta en el menú principal (Ver Figura 3-44) [37].



Figura 3-44. Pantalla: Información general del sistema pico y placa

Si la opción elegida es “VER TIPOS DE PARQ”, se desplegará una pantalla con información de los tipos de parqueaderos que se ofrecen por el sistema pico y placa (Ver Figura 3-45) [38].

- De borde
- Intermodales
- Internos
- Zona Azul



Figura 3-45. Tipos de parqueaderos según el sistema pico y placa

De manera general dentro de cada opción se brinda información con las características de cada tipo de parqueadero y sus respectivas ubicaciones.

En la opción DE BORDE (Ver Figura 3-46) se especifica de forma breve el porqué de su nombre y los cinco sectores donde se encuentran dispuestos los mismos:

1. Carapungo
2. Condado
3. Cuscungo
4. Las cuadas
5. Monte Olivo

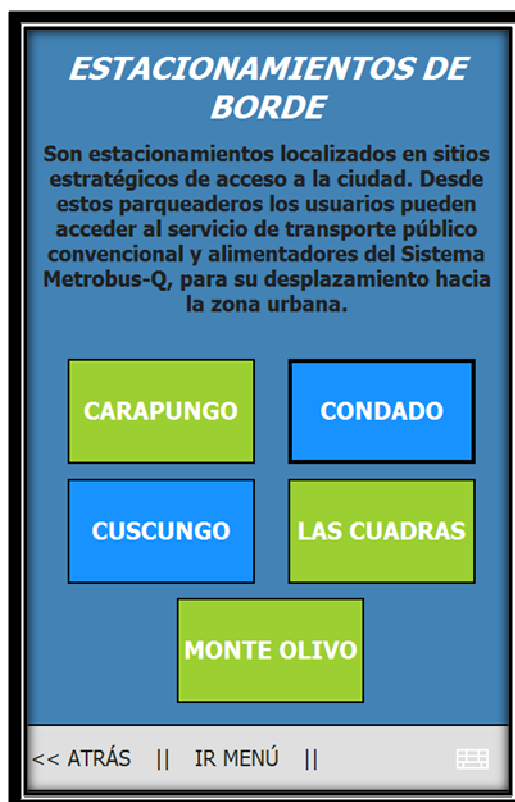


Figura 3-46. Parqueaderos de borde

Al seleccionar cualquiera de las cinco opciones de los parqueaderos de borde se tiene acceso a información complementaria de cada parqueadero referente a (Ver Figura 3-47) [39]:

- Dirección (lugar de ubicación del parqueadero)
- Entrada (por donde se puede ingresar al parqueadero)
- Salida (por donde se puede salir del parqueadero)
- Transporte Público (rutas y operadoras que realizan este servicio)
- Metrobus Q (rutas por donde se realiza este servicio)

Además se cuenta con una opción para volver a ver los tipos de parqueadero o si se desea volver al menú principal.



Figura 3-47. Ejemplo opción parqueadero de borde: Carapungo

Si la opción fue INTERMODALES, se desplegará una ventana con información del tipo de parqueadero y sus cuatro ramificaciones (Ver Figura 3-48):

1. Terminal la Y
2. Terminal Quitumbe
3. Terminal Río Coca
4. Terminal Ofelia

Así como las opciones:

- Atrás.- para volver al menú de tipos de parqueaderos
- Ir Menú.- para volver al menú principal de la aplicación

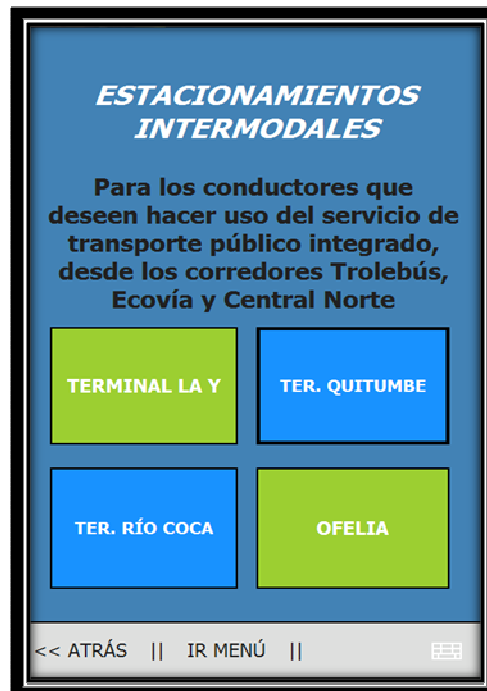


Figura 3-48. Parqueaderos Intermodales

Dentro de cualquiera de estas opciones se tendrá acceso a información complementaria y un mapa de ubicación de los mismos (Ver Figura 3-49).



Figura 3-49. Ejemplo parqueadero intermodal: Trolebús la Y

Si la opción fue INTERNOS, al igual que en los casos anteriores se provee de información referente a estos y sus dos opciones (Ver Figura 3-50):

- Cumandá
- Ññaquito

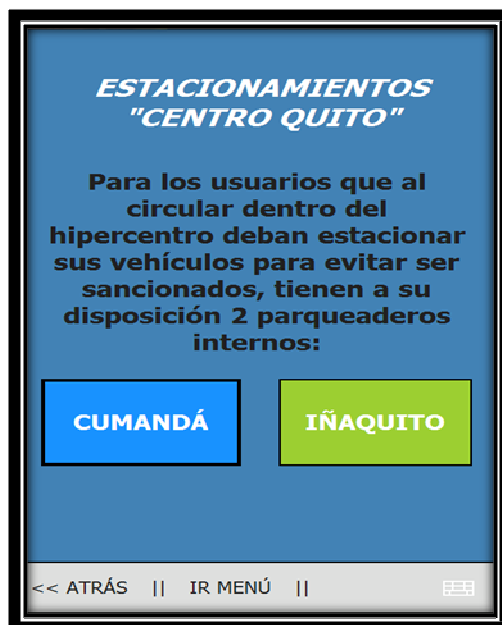


Figura 3-50. Parqueaderos Internos

En estas opciones también se ofrece información extra y un mapa de ubicación (Ver Figura 3-51)



Figura 3-51. Ejemplo parqueadero Interno: Cumandá

Finalmente la opción ZONA AZUL brinda en forma de pestañas información referente a (Ver Figura 3-52):

- Zonas.- Información general de todas las zonas
- La Mariscal.- Información específica sobre la zona.
- La Pradera.- Información específica sobre las dos zonas bajo este nombre.
- La Carolina.- Información específica sobre las dos zonas bajo este nombre.



Figura 3-52. Información Zona Azul

NOTA: Se desea mencionar que para mayor comodidad del usuario se configuró el PDA para que mediante la presión larga de una tecla se pueda acceder de manera directa a la aplicación diseñada (Ver Figura 3-53)



Figura 3-53. Configuración de una tecla del PDA

La opción SALIR nos brinda un cordial mensaje de despedida y cierra la aplicación (Ver Figura 3-54).



Figura 3-54. Mensaje de despedida de la aplicación

3.6.5 FACTORES DE CONSIDERACIÓN [40]

Es importante recalcar que el primer paso para la ejecución del sistema es la adquisición de datos por parte del PDA, para esto es importante que el GPS/BLUETOOTH se encuentre encendido, a lo cual se suma el considerar el tiempo de localización de los satélites por parte del dispositivo GPS el mismo que variará dependiendo de las condiciones a las que se encuentre sometido.

En la Figura 3-55 se muestra la respuesta por parte de la aplicación cuando se desea realizar una petición y el dispositivo GPS no se encuentra aún listo. A esto se le suma la localización del mapa en el punto de coordenadas 0,0.



Figura 3-55. Respuesta de la aplicación si el GPS no está listo.

Si el GPS no se encuentra encendido la aplicación mostrará un mensaje indicando el particular (Ver Figura 3-56) e intentará establecer conexión durante tres oportunidades hasta que el dispositivo sea encendido; caso contrario, se redireccionará la aplicación hacia la pantalla de menú principal para una futura consulta (Ver Figura 3-57).



Figura 3-56. Mensaje de la aplicación si el GPS no está encendido

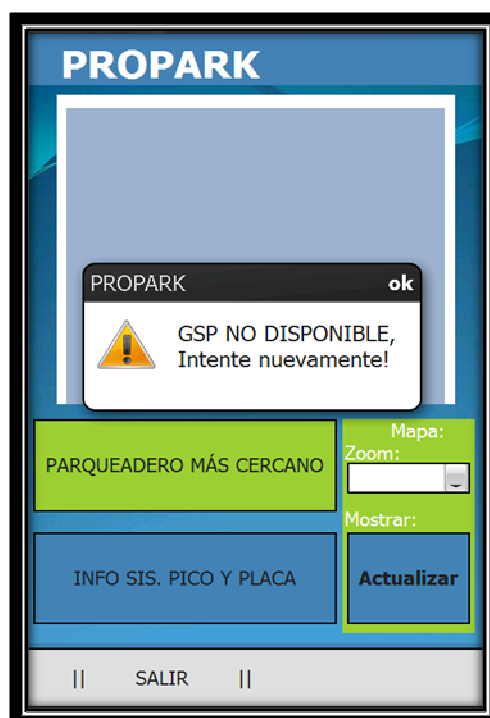


Figura 3-57. Mensaje de no disponibilidad del GPS luego de 3 intentos

Una vez desarrolladas estas aplicaciones el sistema se encuentra completo, permitiendo realizar las respectivas pruebas de funcionamiento y a partir de esto concluir sus resultados.

CAPÍTULO 4

PRUEBAS Y RESULTADOS DEL PROTOTIPO

4.1 INTRODUCCIÓN

Una vez terminada la programación y enlace entre todas las partes constitutivas del sistema prototipo, se realizaron pruebas de funcionamiento con el objetivo de verificar la confiabilidad del mismo.

4.2 PRUEBAS PARA LA APLICACIÓN CLIENTE.

Con respecto a esta aplicación se realizaron pruebas de ingreso y eliminación de registros (placas), con el objeto de constatar si la información es correctamente ingresada dentro de la base de datos, de donde se pudo verificar que el desempeño de la misma se ve afectado por la velocidad y el tráfico de la conexión de internet con la que se cuenta.

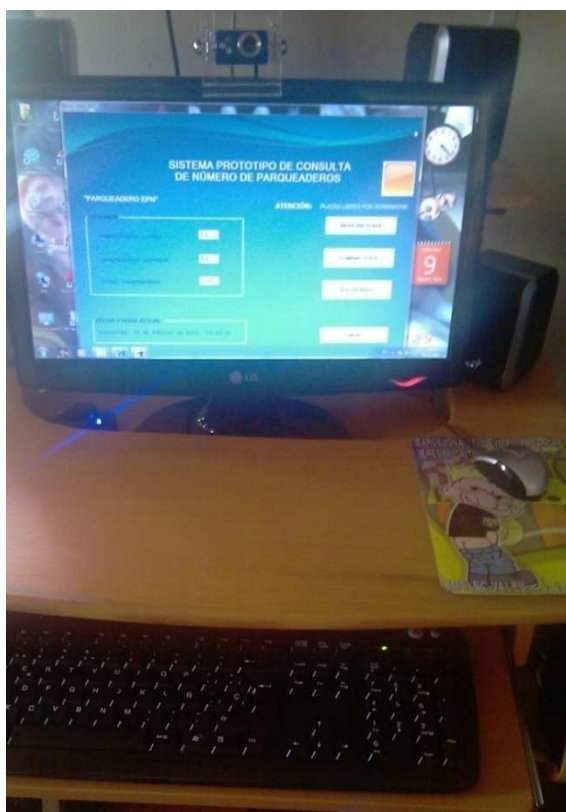


Figura 4-1. Aplicación cliente instalada

4.3 PRUEBAS PARA LA APLICACIÓN SERVIDOR.

En la aplicación “Servidor” se realizaron pruebas para constatar la eficiencia del algoritmo de validación de la información, ya que, en primera instancia, este procedía a utilizar solamente el primer registro de la tarjeta “SIM” con la expectativa de utilizar menos recursos. Bajo esta idea se creía que, al llegar dos mensajes simultáneos el segundo mensaje se almacenaría en el siguiente registro, y en cuanto se desocupara el primero, automáticamente bajaría al puesto libre. Idea que fue descartada debido a que realmente los mensajes son almacenados conforme van llegando al modem y no se mueven de la ubicación asignada.

Ventajosamente este problema se lo solucionó cambiando la lógica del programa instalado en el servidor, la misma que finalmente realiza un barrido de todos los registros permitidos por la capacidad de la tarjeta “SIM”, para el presente caso se utilizó una tarjeta “SIM” de 64k (8 registros).

Si la capacidad de almacenamiento de la SIM es excedida, por características propias de la red GSM los mensajes son almacenados en un SMSC o centro de mensajes cortos (Short Message Service Center) hasta que algún registro sea borrado.



Figura 4-2. Aplicación Servidor instalada

4.4 PRUEBAS PARA LA APLICACIÓN PDA.

Tomando como referencia el perímetro determinado por el sistema pico y placa se realizaron pruebas para cuantificar el tiempo de respuesta luego de una petición realizada.

Dentro de estas pruebas se pudo detectar que la rapidez de respuesta del servidor depende íntegramente de la velocidad del sistema de la red GSM, aproximadamente 30 segundos si la red se encuentra ocupada.

Además, el sistema de consulta por parte del PDA se ve afectado por las condiciones atmosféricas así como por las condiciones locales, es decir si el automotor (y por ende el GPS) se encuentran en lugares cerrados, la adquisición de datos por parte del dispositivo GPS se ve limitada dando como resultado una consulta fallida, la cual valga la pena aclarar, NO ocupa (o gasta) un mensaje SMS mientras no obtengan coordenadas por parte del GPS.

Finalmente esta prueba permite verificar la distancia entre el vehículo y la posición respectiva de los parqueaderos permitiendo dar una idea de cuánto tiempo se tardaría en llegar a dicho recinto y si en dicho tiempo las plazas se mantendrían disponibles



Figura 4-3. Aplicación PDA instalada

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- ❖ Tecnologías inalámbricas como Bluetooth promueven el desarrollo de nuevas soluciones y esquemas de cooperación entre dispositivos que permitan situaciones tales como la creación de sistemas de consulta.
- ❖ Los dispositivos móviles se están diversificando continuamente, por lo que cada vez es más común que un automóvil esté equipado con un dispositivo que permita la ejecución de aplicaciones, conexión a internet, conexión al posicionamiento global (GPS), etc.
- ❖ El presente sistema de consulta, al brindar la disponibilidad inmediata de información con respecto al número de parqueaderos libres, permite la rápida toma de decisión al momento de buscar aparcamiento, por ende una mejor productividad por parte de los dueños del parque automotor ya que, en menor proporción, estarán libres del estrés que conlleva una congestión vehicular. Por otro lado se promueve una mejora en la atención a los usuarios por parte de los centros destinados a dicha actividad ya que al automatizar los procesos de manejo de información habrá una reducción de los recursos operativos los cuales podrán luego ser destinados al mejoramiento del trato al usuario.
- ❖ Al generar una base de datos dedicada se evita la pérdida de tiempo con respecto a la recopilación de información, permitiendo de esta manera generar futuros informes e indicadores, que permitan corregir fallas difíciles de detectar y controlar con un sistema manual.
- ❖ El proveer un sistema de consulta de parqueaderos mejora la eficiencia de las acciones implementadas por el gobierno local y el sector privado, consiguiendo así, disminuir la saturación de las estrechas calles de la ciudad recuperación del espacio público, el impulso del desarrollo económico local y el direccionamiento de la movilidad en la ciudad.

5.2 RECOMENDACIONES

- ❖ Se recomienda evitar utilizar el módem USB ZTE cerca de estructuras metálicas o de dispositivos que emitan ondas electromagnéticas, ya que éstas, afectan en la recepción de las señales.
- ❖ Al ser una parte muy importante el buscar seguridad para todas las personas que ocupen este sistema, se recomienda ser precavidos con la operación de la PC, el PDA y el MÓDEM USB ZTE ya que estos pueden interferir con algunos dispositivos médicos como, equipos auxiliares del oído y marcapasos. Cuando está en uso, según referencias de los fabricantes, por favor mantener estos equipos a más de 20 centímetros de distancia de tales dispositivos médicos.
- ❖ Ser consciente de las restricciones de uso de los dispositivos de este proyecto en lugares donde existan gases explosivos y/o productos explosivos, tales como: gasolineras, depósitos de material inflamable o fábricas de productos químicos.
- ❖ El desempeño del modem puede verse influenciado si la configuración difiere de la recomendada. Además la rapidez de respuesta es mayormente afectada por la velocidad de respuesta de la red GSM.
- ❖ Se recomienda hacer uso de la aplicación con el vehículo detenido, a fin de evitar accidentes de tránsito y posibles sanciones.
- ❖ Es importante tomar en cuenta el número de plazas disponibles totales de cada parqueadero a ser registrado en el sistema, debido a que, este número es considerado como factor limitante en el dimensionamiento de la base de datos.
- ❖ Este sistema puede ser mejorado si se considera una automatización que detecte la presencia de un automotor en la respectiva plaza, lo cual permita conocer efectivamente cuantos espacios se encuentran disponibles y cuantos ocupados.

REFERENCIAS BIBLIOGRÁFICAS

- [1] GONZÁLEZ GÓMEZ Juan, UAM, “El servicio SMS: un enfoque práctico”,
www.learobotics.com/personal/juan/doctorado/sms/sms.pdf
- [2] WIKIPEDIA, “Servicio de mensajes cortos”,
http://es.wikipedia.org/wiki/Servicio_de_mensajes_cortos
- [3] WIKIPEDIA, “Teselado”, <http://es.wikipedia.org/wiki/Teselado>
- [4] WIKIPEDIA, “XAMPP”, <http://es.wikipedia.org/wiki/XAMPP>
- [5] PHPMYADMIN, http://www.phpmyadmin.net/home_page/index.php
- [6] WIKIPEDIA, “Script”, [http://es.wikipedia.org/wiki/Script_\(informática\)](http://es.wikipedia.org/wiki/Script_(informática))
- [7] MAESTROS DEL WEB, “Qué es el PHP”,
<http://www.maestrosdelweb.com/editorial/phpintro/>
- [8] LUCAS.HISPANALINUX, “Tutorial de Perl en Castellano”,
<http://lucas.hispalinux.es/Tutoriales/PERL/tutoperl-print.pdf>
- [9] WIKIPEDIA, “Mysql”, <http://es.wikipedia.org/wiki/MySQL>
- [10] WIKIPEDIA, “Web hosting service”,
http://en.wikipedia.org/wiki/Web_hosting_service
- [11] KOLOTILOV Pavel, Openaces UOC, “Implementación de un procedimiento para la detección y eliminación de entidades duplicadas. Aplicación a los procesos de expropiación”,
http://openaccess.uoc.edu/webapps/o2/bitstream/.../1/pkolotilov_memoria.pdf
- [12] UFM, “Sistemas de coordenadas Geográficas”,
<http://www.tesis.ufm.edu.gt/fisicc/2004/75145/Capítulo%202.htm>
- [13] REINOSO GARCÍA Andy Rafael; TOCAÍN VILLARUEL Christian Andrés, “DISEÑO DE UN PROTOTIPO PARA CONTROLAR UN SEMÁFORO INTELIGENTE USANDO TECNOLOGÍAS GSM/GPRS Y WIRELESS CPU SOBRE UNA PLATAFORMA OPEN-SOFT (LINUX)”, EPN, Quito, junio 2009, págs. 37-48
- [14] MONOGRAFÍAS.COM, “Sistemas Distribuidos”,
<http://www.monografias.com/trabajos16/sistemas-distribuidos/sistemas-distribuidos.shtml#INTRO>

- [15] WIKIPEDIA, "PDA", <http://es.wikipedia.org/wiki/PDA>
- [16] WIKIPEDIA, "Global Positioning System ",
[http://es.wikipedia.org/wiki/Sistema de posicionamiento global](http://es.wikipedia.org/wiki/Sistema_de_posicionamiento_global)
- [17] THE CODE PROJECT, "Writing your own GPS Applications: Part2",
<http://www.codeproject.com/KB/windows/WritingGPSApplications2.aspx>
- [18] SCIENTIFIC COMPONENT, "NMEA 0183 and GPS",
<http://www.scientificcomponent.com/nmea0183.htm>
- [19] KLAUS Betke, "The NMEA 0183 Protocol",
www.tronico.fi/OH6NT/docs/NMEA0183.pdf
- [20] APRS.GIDS, "GPS Nmea Sentence Information", <http://aprs.gids.nl/nmea/>
- [21] WIKIPEDIA, "Bluetooth", <http://es.wikipedia.org/wiki/Bluetooth>
- [22] WIKIPEDIA, "Línea de abonado digital", <http://es.wikipedia.org/wiki/XDSL>
- [23] WIKIPEDIA, "Módem", <http://es.wikipedia.org/wiki/Módem>
- [24] WIKIPEDIA, "Universal Asynchronous Receiver-Transmitter",
<http://es.wikipedia.org/wiki/UART>
- [25] WIKIPEDIA, "PCMCIA", <http://es.wikipedia.org/wiki/PCMCIA>
- [26] WIKIPEDIA, "Peripheral Component Interconnect",
[http://es.wikipedia.org/wiki/Peripheral Component Interconnect](http://es.wikipedia.org/wiki/Peripheral_Component_Interconnect)
- [27] WIKIPEDIA, "AMR", <http://es.wikipedia.org/wiki/AMR>
- [28] GUACHAMÍN CORELLA Pablo Bladimir, Estudio del sistema móvil de comunicaciones GSM y sus futuros servicios de banda ancha, EPN, Quito 2001.
- [29] BLANCO Luis Miguel EIDOS, "Programación en Visual Basic .NET",
<http://www.intercambiosvirtuales.org/libros-manuales/programacin-con-visual-basic-net-luis>
- [30] SOM Guillermo "El Guille". Curso de iniciación a la programación con Visual Basic .NET, www.wiener.edu.pe/.../PROGRAMACION-VISUAL.../Curso-de-iniciacion-a-la-programacion-en-VB-NET.pdf
- [31] MARTINEZ Francisco, "Interfaces en los dispositivos computadores móviles tipo PDA", <http://javimartinez.iespana.es/trabajoingenieriaweb.pdf>

- [32] MYSQL, “Enterprise Data Warehousing with MySQL”,
www.mysql.com/...mysql/.../mysql_wp_for_data_warehousing.pdf
- [33] TUFUNCION, “Distancia entre dos puntos geográficos con MySql”,
<http://www.tufuncion.com/distancia-coordenadas#comment-6224>
- [34] WIKIBOOKS, “Manual del estudiante de ingeniería en sistemas de UTN/Sistemas de información Geográficos/Unidad 2, Análisis espacial”,
[http://es.wikibooks.org/wiki/Manual_del_estudiante_de_Ingeniería_en_Sistemas_de_UTN/Sistemas_de_Información_Geográficos/Unidad_2, Análisis espacial#An.C3.A1lisis_Espacial](http://es.wikibooks.org/wiki/Manual_del_estudiante_de_Ingeniería_en_Sistemas_de_UTN/Sistemas_de_Información_Geográficos/Unidad_2,_Análisis_espacial#An.C3.A1lisis_Espacial)
- [35] GOOGLE, “Guía del usuario de google earth”,
http://earth.google.com/intl/es/userguide/v4/ug_importdata.html
- [36] EMMOP-Q, “Reglamento Pico y placa”,
http://www.emopq.gov.ec/web/guest/servicios-emopq?p_p_id=77&p_p_action=0&p_p_state=maximized&p_p_mode=view&p_p_col_id=&p_p_col_pos=0&p_p_col_count=0&77_struts_action=/journal_content_search/search&77_keywords=pico+y+placa&77_andOperator=true&cur=1
- [37] EMMOP-Q, Boletín de Prensa No. 57, Martes 04 de mayo de 2010, Pico y placa.pdf
- [38] EMDUQ, El estacionamiento vehicular en Quito se ordena,
<http://www.innovario.com/web/index.php>
- [39] NOTICIAS QUITO, “Zona Azul”, <http://www.noticiasquito.gob.ec>
- [40] CHINEA Carlos S, UDLAP, “El estándar Bluetooth IEEE 802.15.1”,
http://catarina.udlap.mx/u_dl_a/tales/documentos/.../capitulo3.pdf

ANEXO A

Especificaciones

HTC modelo HD2

Especificaciones Físicas

Tamaño:



Peso: 157 gramos con batería

Botones:



Velocidad de procesamiento de la CPU

Procesador Snapdragon™ a 1 GHz

Batería¹

Tipo de batería: batería recargable de iones de litio

Capacidad: 1230 mAh

Tiempo de conversación:

- WCDMA: hasta 320 minutos
- GSM: hasta 380 minutos

Tiempo en espera:

- WCDMA: hasta 390 horas
- GSM: hasta 490 horas

Requisitos del sistema Windows recomendado

- Windows XP o Windows Vista®
- Microsoft® ActiveSync® 4.5 para Windows XP
- Microsoft® Windows Mobile Device Center 6.1 para Windows Vista

Pantalla



Tipo: pantalla táctil capacitiva

Resolución: WVGA (480 X 800)

Sistema operativo

Windows Mobile® 6.5 Professional

Almacenamiento

- **ROM:** 512 MB
- **RAM:** 448 MB
- **Ranura de ampliación:** tarjeta de memoria micro SD™ (compatible con SD 2.0)

Bandas de red²



Europa:

- HSPA/WCDMA: 900/2100 MHz
- GSM: 850/900/1800/1900 MHz

Asia-Pacífico:

- HSPA/WCDMA: 900/2100 MHz
- GSM: 850/900/1800/1900 MHz

Internet³

- **Navegador:** Opera Mobile™
- **3G:**
Hasta 7,2 Mb/s de velocidad de descarga
Hasta 2 Mb/s de velocidad de subida
- **GPRS:** hasta 114 kb/s de velocidad de descarga
- **EDGE:** hasta 560 kb/s de velocidad de descarga
- **Wi-Fi®:** IEEE 802.11 b/g

Compartir internet

- Compartir internet a través de USB o Bluetooth
- Router Wi-Fi

Bluetooth

- Bluetooth® 2.1 con EDR (Enhanced Data Rate)

1. La duración de la batería (tiempo de conversación, tiempo de espera y otros) depende de la red y la utilización del teléfono.
2. Las bandas de red en regiones distintas de Europa y Asia-Pacífico pueden ser diferentes en función de la operadora móvil y del lugar. Verifique lo con su operadora móvil.
3. La velocidad de datos real depende de la red o de la potencia de la señal Wi-Fi.

ANEXO B

Dell GPS NAVTEQ Bluetooth Receiver BT-309

Dell GPS Navigation System NAVTEQ Bluetooth Receiver BT-309

This listing includes the following used Navteq-on-Board component:

GPS Bluetooth Receiver (Model BT-309)
12 Volt DC Adapter for easy charging in vehicle
Magnet Mount to attach to metal/car surfaces.

Overview:

This GPS receiver is works with the following PDAs: Dell Axim X50, X51, X51v and X30 series PDAs and HP IPAQ series. This will work with any Pocket PC, SmartPhone or Palm Phone that has Bluetooth Function. Mapping software will work with any Pocket PC as well, not just a Dell Axim PDA.

Features/Specifications:

Provides address-to-address, turn-by-turn directions with visual and spoken guidance.

Moving map display gives you a real-time look at your position.

Dynamic route calculation capability gets you back on track if you miss a turn.

NAVTEQ ON BOARD digital maps comprehensive information on a number of places in the U.S and Canada.

Displays active satellites and signal strengths, as well as ETA, drive time and distance

Compatible for marine, topographical, and airplane navigation

Works up to 30 feet away from bluetooth device. Can be mounted inside/outside of vehicle/bike/boat or airplane.



System Specification

Electrical Characteristics (Receiver)

Frequency L1, 1575.42 MHz

C/A Code 1.023 MHz chip rate

Channels 12

Sensitivity -170 dBW

Accuracy

Position Horizontal 15m 2D RMS (SA off)

WAAS enabled 10m 2D RMS (SA off)

Time 1 micro-second synchronized to GPS time

Velocity 0.1m/sec 95% (SA off)

Datum

Datum WGS-84

Acquisition Rate

Hot start 8 sec., average (with ephemeris and almanac valid)

Warm start 38 sec., average (with almanac but not ephemeris)

Cold start 45 sec., average (neither almanac nor ephemeris)

Reacquisition 0.1 sec. average (interruption recovery time)

Protocol

GPS Output Data NMEA 0183 protocol, and supports command:

GGA, GSA, GSV, RMC, VTG, GLL (VTG and GLL are optional)

GPS transfer rate 38400,N,8,1

Dynamic Condition

Acceleration Limit Less than 4g

Altitude Limit 18,000 meters (60,000 feet) max.

Velocity Limit 515 meters/sec. (1,000 knots) max.

Jerk Limit 20 m/sec**3

External antenna

Power 3.3V

Connector MMCX

Power

Voltage Built-in rechargeable battery(900 mAh) and 5V DC
input charging circuit

Operation Time 8 hr. After fully recharged, in continuous mode
>10 hr in trickle power mode

Physical Characteristics

Dimension 85mm x 47mm x 26mm

Weight 98g

Temperature

Operating -20°~ 70°C

Storage -30°~ 80°C

Humidity Up to 95% non-condensing

Bluetooth Specification

Bluetooth V1.1 Compliant

Supply Voltage : 2.8V ~ 3.3V

Frequency Range : 2.402 ~ 2.480 GHz

Receiver Sensitivity : -80 dBm

Transmit Power : Class 2

Transmitting Range : 3 ~10 m (Depends on environment)

Power Consumption : 45 mA (Typical)

ANEXO C

MODEM ZTE MF100 DATACARD FAQ

Description of MF100 Datacard Specifications

Interface type : USB2.0 HIGH SPEED
 Protocol : GSM/GPRS/EDGE/UMTS/WCDMA/HSDPA
 Dimensions: 69.7mm * 26mm * 11.5mm
 85.3mm* 26mm * 11.5mm (with cap)
 Memory: 32MByte/64MByte (SDRAM/NAND), Up to 32GByte
 Environment: Operating Temperature -10 to 60°C
 Storage Temperature -40 to 80°C
 Humidity 5%~ 95%
 Auto Installation: Support
 Weight: about 21g, about 23g (with cap)
 Message: SMS
 Power supply: 5V
 Processor: ARM 9
 The picture of MF100 as bellow:



Use and Common Setting of Datacard

The Minimum System requirements needed to run software are as follows:
 Windows XP (SP2 and later)\Windows Vista (32bit)\Windows Vista
 (64bit)\Windows 7(NDIS6.2 not supported)\MAC (10.4 later)\Linux (Optional)

Frequency : WCDMA/HSDPA 2100MHz: +24dBm (Power Class 3)
 GSM/GPRS 850MHz/900MHz: +33dBm (Power Class 4)
 GSM/GPRS 1800MHz/1900MHz: +30dBm (Power Class 1)
 EDGE 850MHz/900MHz: +27dBm (Power Class E2)
 EDGE 1800MHz/1900MHz: +26dBm (Power Class E2)

2.1.3. How is the transmission of MF100

Transmission: GSM CS: UL 9.6kbps/DL 9.6kbps
GPRS: Multi-slot Class 10
EDGE: Multi-slot Class 12
WCDMA CS: UL 64kbps/DL 64kbps
WCDMA PS: UL 384kbps/DL 384kbps
HSDPA: DL 3.6Mb/s (Category 6)

The memory of Micro SD card is up to 32GByte.

The quality of the MF110 HSDPA USB Modem supports main prevailing types of laptops or portable computer, such as HP, IBM, Toshiba, Lenovo, Dell, Sony, etc..

Is the modem compatible with Apple Mac

After full installation, the system doesn't install the driver properly

Please install the driver program manually.

ANEXO D

Código Fuente del Sistema PROPARK

D.1 CÓDIGO FUENTE APLICACIÓN CLIENTE

Form 1

```

Public Class Form1
    Public MiStringA As String = valor()
    Public valor1 As String
    Public IDParqueadero As String
    Public fecha As String
    Public hora As String
    Public _Contador As Integer
    Public ocupados As Integer
    Public m_array(ocupados) As String

    Public Sub InicioProceso()

        _Contador = ProgressBar1.Minimum

        lblInfoBD.Text = "Iniciando..."

        Me.Cursor = Cursors.WaitCursor

        BackgroundWorker1.WorkerSupportsCancellation = True

        BackgroundWorker1.WorkerReportsProgress = True

        BackgroundWorker1.RunWorkerAsync()

    End Sub

    Public Sub FinProceso()

        BackgroundWorker1.CancelAsync()

    End Sub

    Function Eliminar(ByVal LB As ListBox) As Int32
        Dim Random1 As New Random()
        Dim asignado0 As String

        Dim i As Int32
        Dim j As Int32
        Dim n As Int32
    verificar:
        For i = 0 To LB.Items.Count - 2
            For j = LB.Items.Count - 1 To i + 1 Step -1
                ' ... si es el mismo
                If LB.Items(i).ToString = LB.Items(j).ToString Then
                    ' elimina el elemento indicando el índice
                    LB.Items.RemoveAt(j)
                    asignado0 = Random1.Next(1, 100)
                    LB.Items.Add(asignado0)
                    GoTo verificar
                    n += 1 'lleva la cuenta de los duplicados
                End If
            Next j
        Next i
    End Function

```



```

        End If
    Next
Next
Return n ' retorna los eliminados
End Function

Public Function valor() As String

    'FUNCIÓN RANDOM PARA NÚMERO DE PARQUEADEROS
    Dim Random As New Random()
    Dim disponibles1 As Integer = Random.Next(1, 100)

    MiStringA = Convert.ToString(disponibles1)
    ocupados = 100 - MiStringA

    Return MiStringA

End Function

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    'NOMBRE DEL PARQUEADERO
    InicioProceso()

    Variables.IDParqueadero = "PEpn"
    IDParqueadero = Variables.IDParqueadero
    fecha = Date.Now.ToShortDateString
    hora = Date.Now.ToLongTimeString

    '*****
    'REALIZO UN RANDOM PARA GENERAR LAS PLAZAS OCUPADAS EN UNA LISTA
    '*****
    Dim Random1 As New Random()
    Dim j As Integer
    Dim asignado0 As String

    For j = 0 To 99 - MiStringA
repetir0:

        asignado0 = Random1.Next(1, 100)

        For Each item As String In lstbxParqueaderos0.Items
            ' Si el elemento existe, repetimos el random
            If (item = asignado0) Then
                GoTo repetir0
            End If
        Next
        ' No existe el elemento; lo añadimos a la lista
        lstbxParqueaderos0.Items.Add(asignado0)
    Next

    '*****
    'muestra el número aleatorio de parqueaderos LIBRES en un combobox
    '*****

```

```

    Dim s As Integer
    For s = 1 To 100
        For Each item In lstbxParqueaderos0.Items
            If s = item Then
                GoTo siguiente
            End If
        Next
        lstbxParqueaderosL.Items.Add(s)
siguiente:
    Next

    Timer1.Enabled = True

End Sub

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
    'DESHABILITO EL TIMER
    Timer1.Enabled = False

    For j = 0 To 99 - MiStringA
'*****INGRESO PARQUEADEROS A LA BASE ACORDE AL NÚMERO RANDÓMICO*****
        'genero la plaza para los parqueadero OCUPADOS
        Dim conexion7 As New MySqlConnection
        Dim query7 As String
        Dim micomando7 As New MySqlCommand(query7, conexion7)
        Try
            conexion7.Open()
        Catch ex As Exception

        End Try
        micomando7.ExecuteNonQuery()
        conexion7.Close()
    Next

    FinProceso()

    'LLAMO A LA SIGUIENTE VENTANA
    Dim z As New Ingreso()
    z.Show()
    Me.Hide()

End Sub

Private Sub BackgroundWorker1_DoWork(ByVal sender As System.Object,
ByVal e As System.ComponentModel.DoWorkEventArgs) Handles
BackgroundWorker1.DoWork
    While Not BackgroundWorker1.CancellationPending

        If Not BackgroundWorker1.CancellationPending Then

            If _Contador = Me.ProgressBar1.Maximum Then _Contador =
ProgressBar1.Minimum Else _Contador += 5

            Threading.Thread.Sleep(100)

```

```

        BackgroundWorker1.ReportProgress(_Contador, "Creando
datos...")

    Else

        e.Cancel = True

        Exit While

    End If

End While

End Sub

Private Sub BackgroundWorker1_ProgressChanged(ByVal sender As Object,
ByVal e As System.ComponentModel.ProgressChangedEventArgs) Handles
BackgroundWorker1.ProgressChanged

    ProgressBar1.Value = e.ProgressPercentage
    lblInfoBD.Text = e.UserState
End Sub

Private Sub BackgroundWorker1_RunWorkerCompleted(ByVal sender As
Object, ByVal e As System.ComponentModel.RunWorkerCompletedEventArgs)
Handles BackgroundWorker1.RunWorkerCompleted

    If Not (e.Error Is Nothing) Then

        Dim msg As String = String.Format("Error: {0}",
e.Error.Message)

        MessageBox.Show(msg)

    Else

        ProgressBar1.Value = ProgressBar1.Minimum

        lblInfoBD.Text = "Creando datos..."

    End If

    Me.Cursor = Cursors.Default

    ProgressBar1.Value = ProgressBar1.Maximum

End Sub

End Class

```

Ingreso

```

Public Class Ingreso
    Public aux As Integer
    Private arreglo() As String = Form1.m_array

    Public Sub iniciar()

        aux = Variables.aux
    
```

```

If usuario.Text = "xxxxxxxxx" Then

    If contraseña.Text = "xxxxxxxxx" Then

        pgbIngreso.Visible = True
        lblCargando.Visible = True

        Dim k As Integer
        For k = 0 To 99
            pgbIngreso.Value = k
            Threading.Thread.Sleep(40)
        Next

        aux = 0
        Variables.aux = aux

        Dim datosms As New PPrincipal()
        datosms.Show()
        Me.Hide()

    Else

        MessageBox.Show("Clave incorrecta... ")
        aux = aux + 1
        Variables.aux = aux

    End If
Else
    MessageBox.Show("Nombre de usuario incorrecto... ")
    aux = aux + 1
    Variables.aux = aux

End If
If aux = 3 Then
    MessageBox.Show("Número de intentos exedidos")
    aux = 0
    Dim salirE As New Gracias
    salirE.Show()
    Me.Hide()

End If
usuario.Clear()
contraseña.Clear()
Refresh()
usuario.Focus()

End Sub

Private Sub TextBox1_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles usuario.KeyDown
    If e.KeyCode = Keys.Enter Then
        contraseña.Focus()
    End If
End Sub

Private Sub TextBox2_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles contraseña.KeyDown
    If e.KeyCode = Keys.Enter Then
        btnIngresar.Focus()
    End If
End Sub

```

```

        End If
    End Sub
    Private Sub TextBox2_KeyPress(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.KeyPressEventArgs) Handles contraseña.KeyPress

        If e.KeyChar = ChrW(13) Then
            iniciar()
        End If
    End Sub

    Private Sub Ingreso_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        Dim d As Integer
        For d = 0 To Form1.lstbxParqueaderosL.Items.Count - 1
            cmbxOrdenados.Items.Add(Form1.lstbxParqueaderosL.Items(d))
        Next
    End Sub

    Private Sub btnSalir_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSalir.Click
        Dim salirElim As New Gracias
        salirElim.Show()
        Me.Hide()

    End Sub

    Private Sub btnIngresar_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnIngresar.Click
        iniciar()
    End Sub
End Class

```

Pprincipal

```

Public Class PPrincipal
    Public MiStringA As String = Form1.MiStringA
    Public valor1 As String, valor2 As String
    Public IDParqueadero As String
    Public Id As String

    Private Sub Form2_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        lblTitulo.BackColor = Color.Transparent
        lblNombre.BackColor = Color.Transparent
        lblLibres.BackColor = Color.Transparent
        lblOcupados.BackColor = Color.Transparent
        lblTotal.BackColor = Color.Transparent
        lblFecha.BackColor = Color.Transparent
        lblHora.BackColor = Color.Transparent
        grbxResumen.BackColor = Color.Transparent
        grbxFechaHora.BackColor = Color.Transparent
        btnIngresarPlaza.Focus()
        pctbxRojo.BackColor = Color.Transparent
        pctbxAmarillo.BackColor = Color.Transparent
        lblEstado.BackColor = Color.Transparent
        lblAtencion.BackColor = Color.Transparent
    End Sub

```

principal:

```

'HABILITO EL TIMER PARA EL RELOJ
Timer2.Enabled = True

'muestra el número aleatorio = PARQUEADEROS LIBRES
lstbLibres.Items.Add(MiStringA)

'muestro PARQUEADEROS OCUPADOS
MiStringA = Convert.ToInt16(MiStringA)
lstbOcupados.Items.Add(100 - MiStringA)

'muestro TOTAL PARQUEADEROS
lstbTotal.Items.Add(100)

'actualizo registros Disponibles y Ocupados en base de datos
valor1 = lstbLibres.Items(0)
valor2 = lstbOcupados.Items(0)

'muestro fecha
LblFecha.Text = Date.Now.ToLongDateString

'Dim conexion1 As New MySqlConnection("Data Source=localhost;
Database=baseparqueaderos; User ID=root;")
Dim conexion1 As New MySqlConnection()

'*++++++*
Dim query1 As String
Dim micomando1 As New MySqlCommand(query1, conexion1)

Try
    conexion1.Open()
Catch ex As Exception

End Try

micomando1.ExecuteNonQuery()
conexion1.Close()

If lstbLibres.Items.Item(0) = 0 Then

    lblEstado.Text = ""
    lblEstado.Text = "PARQUEADERO LLENO"
    lblAtencion.ForeColor = Color.Red

    pctbxAmarillo.Visible = False
    pctbxAzul.Visible = False
    pctbxRojo.Visible = True
    pctbxVerde.Visible = False

    lblEstado.Visible = True
    lblAtencion.Visible = True
    GoTo fin
End If

If lstbLibres.Items.Item(0) <= 20 Then

    lblEstado.Text = ""
    lblEstado.Text = "PLAZAS LIBRES POR TERMINARSE"

```

```
    lblAtencion.ForeColor = Color.Yellow

    pctbxAmarillo.Visible = True
    pctbxAzul.Visible = False
    pctbxRojo.Visible = False
    pctbxVerde.Visible = False

    lblEstado.Visible = True
    lblAtencion.Visible = True
    GoTo fin
End If

If lstbLibres.Items.Item(0) = 100 Then

    lblEstado.Text = ""
    lblEstado.Text = "PARQUEADERO VACÍO"
    lblAtencion.ForeColor = Color.Green

    pctbxAmarillo.Visible = False
    pctbxAzul.Visible = False
    pctbxRojo.Visible = False
    pctbxVerde.Visible = True

    lblEstado.Visible = True
    lblAtencion.Visible = True
    GoTo fin
End If

If lstbLibres.Items.Item(0) >= 80 Then

    lblEstado.Text = ""
    lblEstado.Text = "PARQUEADERO CASI DESOCUPADO"
    lblAtencion.ForeColor = Color.Blue

    pctbxAmarillo.Visible = False
    pctbxAzul.Visible = True
    pctbxRojo.Visible = False
    pctbxVerde.Visible = False

    lblEstado.Visible = True
    lblAtencion.Visible = True
    GoTo fin
End If

    lblEstado.Text = ""
    lblEstado.Text = "PARQUEADERO VACÍO"
    lblAtencion.ForeColor = Color.Green

    pctbxAmarillo.Visible = False
    pctbxAzul.Visible = False
    pctbxRojo.Visible = False
    pctbxVerde.Visible = False

    lblEstado.Visible = False
    lblAtencion.Visible = False
```

fin:

```

End Sub

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer2.Tick
    'MUESTRO CADA SEGUNDO EN EL RELOJ
    lblHora.Text = Now.ToLongTimeString
    Timer2.Interval = 1000
End Sub

Private Sub btnIngresarPlaza_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnIngresarPlaza.Click
    'LLAMO A LA SIGUIENTE VENTANA = INGRESO DE PLAZA
    pctbxRojo.Visible = False
    lblEstado.Visible = False
    lblAtencion.Visible = False

    Dim ingreso As New Form3
    ingreso.Show()
    Me.Hide()
End Sub

Private Sub btnEliminarPlaza_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnEliminarPlaza.Click
    pctbxRojo.Visible = False
    lblEstado.Visible = False
    lblAtencion.Visible = False

    Dim eliminar As New Form4
    eliminar.Show()
    Me.Hide()
End Sub

Private Sub btnVerUsuarios_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btnVerUsuarios.Click
    pctbxRojo.Visible = False
    lblEstado.Visible = False
    lblAtencion.Visible = False

    Dim ver As New Ver_Usuarios
    ver.Show()
    Me.Hide()
End Sub

Private Sub btnSalir_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSalir.Click
    pctbxRojo.Visible = False
    lblEstado.Visible = False
    lblAtencion.Visible = False

    Dim Msg As String, Style As String, Title As String, Response As
String, MyString As String
    Msg = "Seguro que desea SALIR de la aplicación?"
Define botones:
    Style = vbYesNo + MsgBoxStyle.Question
    Title = "ATENCIÓN!" ' Define título.

    Response = MsgBox(Msg, Style, Title) ' Muestra el mensaje
    If Response = vbYes Then ' Si el usuario escoge Yes.
        Dim salir As New Gracias

```



```

        salir.Show()
        Me.Hide()
    Else ' El usuario escoge No.
        Return
    End If
End Sub

```

End Class

Ingreso plaza

```

Public Class Form3
    Public MiStringA As String = PPrincipal.MiStringA
    Public valor1 As String, valor2 As String
    Public asignado As Integer
    Public var As Integer
    Public KeyAscii As Short
    Public fechaparabase As String

    Function SoloNumeros(ByVal Keyascii As Short) As Short
        If InStr("1234567890", Chr(Keyascii)) = 0 Then
            SoloNumeros = 0
        Else
            SoloNumeros = Keyascii
        End If
        Select Case Keyascii
            Case 8
                SoloNumeros = Keyascii
            Case 13
                SoloNumeros = Keyascii
        End Select
    End Function

    Private Sub Form3_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles Me.KeyDown
        If e.KeyCode = Keys.Enter Then
            msktbxPlaca.Focus()
        End If
    End Sub

    Private Sub Timer3_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer3.Tick
        'MUESTRO CADA SEGUNDO EN EL RELOJ
        lblDespliegueHora.Text = Now.ToLongTimeString
        Timer3.Interval = 1000

    End Sub

    Private Sub ComboBox1_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs)
        If e.KeyCode = Keys.Enter Then
            btnIngresarPlaza.Focus()
        End If
    End Sub

    Private Sub cmbbxPlaza_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles cmbbxPlaza.KeyDown
        btnIngresarPlaza.Focus()
    End Sub

```

```

Private Sub cmbbxPlaza_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles cmbbxPlaza.KeyPress
    Dim KeyAscii As Short = CShort(Asc(e.KeyChar))
    KeyAscii = CShort(SoloNumeros(KeyAscii))
    If KeyAscii = 0 Then
        e.Handled = True
    End If
    If KeyAscii = 13 Then
        btnIngresarPlaza.Focus()
    End If
End Sub

```

```

Private Sub msktbxPlaca_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles msktbxPlaca.KeyDown

```

```

    If e.KeyCode = 13 Then
        msktbxPlaca.TextMaskFormat =
MaskFormat.ExcludePromptAndLiterals
        If msktbxPlaca.Text.Length = 0 Then
            MsgBox("INGRESE DATOS DE PLACA", MsgBoxStyle.Critical)
            Return
        Else

            cmbbxPlaza.Focus()
        End If

```

```

    End If
End Sub

```

```

Private Sub msktbxPlaca_TypeValidationCompleted(ByVal sender As
Object, ByVal e As System.Windows.Forms.TypeValidationEventArgs) Handles
msktbxPlaca.TypeValidationCompleted
    msktbxPlaca.TextMaskFormat = MaskFormat.ExcludePromptAndLiterals
    ' si no hay valor en el control, salir
    If msktbxPlaca.Text.Length = 0 Then
        Return
    End If
End Sub

```

```

Private Sub btnIngresarPlaza_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnIngresarPlaza.Click
    '*****FORMATO DE PLACA*****
    msktbxPlaca.TextMaskFormat = MaskFormat.ExcludePromptAndLiterals

    '*****RESTRICCIONES*****
    If String.IsNullOrEmpty(msktbxPlaca.Text) = True Then
        MsgBox("INGRESE DATOS DE PLACA", MsgBoxStyle.Critical)
        msktbxPlaca.Focus()
        msktbxPlaca.TextMaskFormat = MaskFormat.IncludeLiterals

        Return

    Else
        msktbxPlaca.TextMaskFormat = MaskFormat.IncludeLiterals

        'If cmbbxPlaza.SelectedItem.ToString = "".ToString Then

```

```

'End If

If String.IsNullOrEmpty(cmbbxPlaza.SelectedItem) = True Then
    MsgBox("SELECCIONE UNA PLAZA", MsgBoxStyle.Critical)
    cmbbxPlaza.Focus()
    Return
End If

If cmbbxPlaza.SelectedItem.ToString = " ".ToString Then
    MsgBox("SELECCIONE UNA PLAZA", MsgBoxStyle.Critical)
    cmbbxPlaza.Focus()
    Return
End If

valor1 = (Form1.MiStringA - 1).ToString
valor2 = (100 - Form1.MiStringA + 1).ToString

If valor1 = " ".ToString Then
    valor1 = "0"
End If

'*****CONSULTO SI NO ESTÁ REGISTRADO*****

Dim conexion99 As New MySqlConnection()

'Selecciono la tabla a la que quiero consultar
Dim query99 As String
'conexion a utilizar
Dim micomando99 As New MySqlCommand(query99, conexion99)

'abro la conexión
Try
    conexion99.Open()
Catch ex As MySql.Data.MySqlClient.MySqlException
    MsgBox("No se ha podido establecer " & vbCrLf & "la
conexión con la base de datos.", MsgBoxStyle.Critical)
Finally
End Try

'EJECUTO COMANDO
Dim milector99 As MySqlDataReader = micomando99.ExecuteReader

Try
    While milector99.Read()

        If milector99.GetValue(1).ToString = Label9.Text Then

            If milector99.GetValue(2).ToString =
msktbxPlaca.Text Then
                MsgBox("PLACA YA REGISTRADA EN EL PARQUEADERO
: " + milector99.GetValue(1).ToString, MsgBoxStyle.Critical)
                msktbxPlaca.Focus()
                Return
            End If

        Else

```

```

' *+++++*
If milector99.GetValue(1).ToString = "PCadisan" Then
    MsgBox("OJO PLACA REGISTRADA EN OTRO
           PARQUEADERO", MsgBoxStyle.Information)

    Dim Msg As String, Style As String, Title As
        String, Response As String
    Msg = "Desea ingresar la plaza?"
    ' Define botones:
    Style = vbYesNo + MsgBoxStyle.Exclamation
    Title = "ATENCIÓN!" ' Define título.

    Response = MsgBox(Msg, Style, Title) '
    If Response = vbYes Then
        GoTo ingresar
    Else
        Return
    End If

    msktbxPlaca.Focus()

End If

If milector99.GetValue(1).ToString = "PRecreo" Then
    MsgBox("OJO PLACA REGISTRADA EN OTRO
           PARQUEADERO", MsgBoxStyle.Information)

    Dim Msg As String, Style As String, Title As
        String, Response As String
    Msg = "Desea ingresar la plaza?"
    Style = vbYesNo + MsgBoxStyle.Exclamation
    Title = "ATENCIÓN!" ' Define título.

    Response = MsgBox(Msg, Style, Title)
    If Response = vbYes Then
        GoTo ingresar
    Else
        Return
    End If

    msktbxPlaca.Focus()

End If

If milector99.GetValue(1).ToString = "PPintado" Then
    MsgBox("OJO PLACA REGISTRADA EN OTRO
           PARQUEADERO", MsgBoxStyle.Information)

    Dim Msg As String, Style As String, Title As
        String, Response As String
    Msg = "Desea ingresar la plaza?"

    Style = vbYesNo + MsgBoxStyle.Exclamation
    Title = "ATENCIÓN!"

    Response = MsgBox(Msg, Style, Title) '
    If Response = vbYes Then
        GoTo ingresar
    Else
        Return
    End If

```

```

End If

msktbxPlaca.Focus()

End If
If milector99.GetValue(1).ToString = "PTejar" Then
    MsgBox("OJO PLACA REGISTRADA EN OTRO
           PARQUEADERO", MsgBoxStyle.Information)

    Dim Msg As String, Style As String, Title As
        String, Response As String
    Msg = "Desea ingresar la plaza?"
    Style = vbYesNo + MsgBoxStyle.Exclamation
    Title = "ATENCIÓN!"
    Response = MsgBox(Msg, Style, Title)
    If Response = vbYes Then
        GoTo ingresar
    Else
        Return
    End If

    msktbxPlaca.Focus()

End If
If milector99.GetValue(1).ToString = "PMontufar" Then
    MsgBox("OJO PLACA REGISTRADA EN OTRO
           PARQUEADERO", MsgBoxStyle.Information)

    Dim Msg As String, Style As String, Title As
        String, Response As String
    Msg = "Desea ingresar la plaza"
    Style = vbYesNo + MsgBoxStyle.Exclamation
    Title = "ATENCIÓN!"
    Response = MsgBox(Msg, Style, Title)
    If Response = vbYes Then
        GoTo ingresar
    Else
        Return
    End If

    msktbxPlaca.Focus()

End If
If milector99.GetValue(1).ToString = "PLaRonda" Then
    MsgBox("OJO PLACA REGISTRADA EN OTRO
           PARQUEADERO", MsgBoxStyle.Information)

    Dim Msg As String, Style As String, Title As
        String, Response As String
    Msg = "Desea ingresar la plaza"
    Style = vbYesNo + MsgBoxStyle.Exclamation
    Title = "ATENCIÓN!"

    Response = MsgBox(Msg, Style, Title)
    If Response = vbYes Then
        GoTo ingresar
    Else
        Return
    End If
End If

```

```

        msktbxPlaca.Focus()

        End If
        If milector99.GetValue(1).ToString = "Pofelia" Then
            MsgBox("OJO PLACA REGISTRADA EN OTRO
                PARQUEADERO", MsgBoxStyle.Information)

        Dim Msg As String, Style As String, Title As String, Response As String
            Msg = "Desea ingresar la plaza?"
            Style = vbYesNo + MsgBoxStyle.Exclamation
            Title = "ATENCIÓN!"

            Response = MsgBox(Msg, Style, Title) '
            If Response = vbYes Then
                GoTo ingresar
            Else
                Return
            End If

            msktbxPlaca.Focus()

        End If
        If milector99.GetValue(1).ToString = "PQuicentroN" Then
            MsgBox("OJO PLACA REGISTRADA EN OTRO
                PARQUEADERO", MsgBoxStyle.Information)

        Dim Msg As String, Style As String, Title As String, Response As String
            Msg = "Desea ingresar la plaza?"
            Style = vbYesNo + MsgBoxStyle.Exclamation
            Title = "ATENCIÓN!"

            Response = MsgBox(Msg, Style, Title) '
            If Response = vbYes Then
                GoTo ingresar
            Else
                Return
            End If

            msktbxPlaca.Focus()

        End If
        If milector99.GetValue(1).ToString = "PRioCoca"
Then
            MsgBox("OJO PLACA REGISTRADA EN OTRO
                PARQUEADERO", MsgBoxStyle.Information)

            Dim Msg As String, Style As String, Title As
String, Response As String
                Msg = "Desea ingresar la plaza"
                Style = vbYesNo + MsgBoxStyle.Exclamation
                Title = "ATENCIÓN!"
                Response = MsgBox(Msg, Style, Title)
                If Response = vbYes Then
                    GoTo ingresar
                Else
                    Return
                End If

                msktbxPlaca.Focus()

```

```

        End If

        End If
    End While
Catch ex As Exception
End Try

conexion99.Close()

'*****INGRESO DE DATOS*****
ingresar:
    Dim query1 As String
    Dim micomando1 As New MySqlCommand(query1, conexion1)

    Try
        conexion1.Open()
    Catch ex As Exception

    End Try

    micomando1.ExecuteNonQuery()
    MsgBox("INGRESADO: " & msktbxPlaca.Text & " HORA: " &
lblDespliegueHora.Text & " PLAZA: " & cmbbxPlaza.Text,
MsgBoxStyle.Information)
    conexion1.Close()

    '******
    Dim query As String = "UPDATE
`xavierpo_basep`.`parqueaderos` SET `Disponibles` = '" + valor1 +
"', `Ocupados` = '" + valor2 + "' WHERE `parqueaderos`.`Id` =10;"
    Dim micomando As New MySqlCommand(query, conexion)

    Try
        conexion.Open()
    Catch ex As Exception

    End Try

    micomando.ExecuteNonQuery()
    'MsgBox("Datos actualizados")
    conexion.Close()

'*****

If btnIngresarPlaza.CausesValidation = True Then
    Form1.MiStringA = Form1.MiStringA - 1
End If

With cmbbxPlaza
    For i = .SelectedItem.ToString - 1 To 0 Step -1
        lstbx1.Items.Remove(.SelectedItem)
        PPrincipal.ListBox4.Items.Remove(.SelectedItem)
        Form1.lstbxParqueaderosL.Items.Remove(.SelectedItem)
    
```

```

        Next
        Form1.lstbxParqueaderosL.Items.Add(" ")
        Form1.lstbxParqueaderosO.Items.Add(.SelectedItem)
    End With

    Dim main As New PPrincipal
    main.Show()
    Me.Hide()
End If

End Sub

Private Sub btnRegresar_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnRegresar.Click
    Dim menu As New PPrincipal
    menu.Show()
    Me.Hide()
End Sub

Private Sub Form3_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    lblNombreParq.BackColor = Color.Transparent
    lblTituloIngreso.BackColor = Color.Transparent
    grboxParqueaderos.BackColor = Color.Transparent
    grbxDatos.BackColor = Color.Transparent
    lblDespliegueFecha.Text = Date.Now.ToLongDateString
    fechaparabase = Date.Now.ToShortDateString

    msktbxPlaca.Focus()

    'HABILITO EL TIMER PARA EL RELOJ
    Timer3.Enabled = True
    msktbxPlaca.Focus()

    'MUESTRO EL # DE PARQUEADEROS DISPONIBLES Y OCUPADOS

    lblNdisp.Text = Form1.MiStringA
    lblNocup.Text = 100 - Form1.MiStringA

    Dim d As Integer
    For d = 0 To Form1.lstbxParqueaderosL.Items.Count - 1
        cmbbxPlaza.Items.Add(Form1.lstbxParqueaderosL.Items(d))
    Next

    msktbxPlaca.Focus()

End Sub

End Class

```

Facturación

```

Imports MySql.Data.MySqlClient

Public Class Form4
    Public Plaza As Integer

```



```

Public Sub buscar()
    msktbxPlaca.TextMaskFormat = MaskFormat.ExcludePromptAndLiterals

    If String.IsNullOrEmpty(msktbxPlaca.Text) Then
        MsgBox("Ingreso la placa a ser facturada ",
        MsgBoxStyle.Critical)
        msktbxPlaca.Focus()
        msktbxPlaca.TextMaskFormat = MaskFormat.IncludeLiterals

        GoTo fin
    Else
        msktbxPlaca.TextMaskFormat = MaskFormat.IncludeLiterals

        '*****CONSULTA GENERAL*****

        Dim conexion8 As New

        'Selecciono la tabla a la que quiero consultar
        Dim query8 As String
        'conexión a utilizar"
        Dim micomando8 As New MySqlCommand(query8, conexion8)

        'abro la conexión
        Try
            conexion8.Open()
        Catch ex As MySql.Data.MySqlClient.MySqlException
            MsgBox("No se ha podido establecer " & vbCrLf & "la
conexión con la base de datos.", MsgBoxStyle.Critical)
        Finally
            End Try

        'EJECUTO COMANDO
        Dim miledtor8 As MySqlDataReader = micomando8.ExecuteReader

        '*****CONSULTO Y MUESTRO*****
        Try
            While miledtor8.Read()

                If msktbxPlaca.Text = miledtor8.GetValue(2).ToString Then
                    conexion8.Close()
                    GoTo normal

                Else
                    End If
                End While

            Catch ex As Exception
            End Try
            MsgBox("no se encuentra registrado", MsgBoxStyle.Exclamation)
            msktbxPlaca.Clear()
            msktbxPlaca.Focus()

            'cierro la conexión
            conexion8.Close()

```

normal:

```

'*****CONSULTA *****
Dim conexion As New

'Selecciono la tabla a la que quiero consultar
Dim query As String

'conexion a utilizar"
Dim micomando As New MySqlCommand(query, conexion)

'abro la conexión
Try
    conexion.Open()
Catch ex As MySql.Data.MySqlClient.MySqlException
    MsgBox("No se ha podido establecer " & vbCrLf & "la
conexión con la base de datos.", MsgBoxStyle.Critical)
Finally

End Try

'EJECUTO COMANDO
Dim miclector As MySqlDataReader = micomando.ExecuteReader

'*****CONSULTO Y MUESTRO*****
Try
    While miclector.Read()

        If miclector.GetValue(2).ToString = msktbxPlaca.Text Then

            If miclector.GetValue(1).ToString = "PEpn" Then

                txtId.Text = miclector.GetValue(1).ToString()
                'IDPARQ
                txtPlaca.Text = miclector.GetValue(2).ToString()
                txtFecha.Text = miclector.GetValue(4).ToString()
                txtHoraIngreso.Text = miclector.GetValue(5).ToString()
                txtHoraSalida.Text = Label9.Text
                txtPlaza.Text = miclector.GetValue(3).ToString()
                Variables.Plaza = txtPlaza.Text
                TextBox8.Text = miclector.GetValue(0).ToString()
                btnFacturar.Focus()

            End If

        End If

    End While

Catch ex As Exception
End Try
conexion.Close()

'*****CONSULTA PARA VER SI ESTÁ EN OTRO PARQUEADERO*****

```

```

Dim conexion9 As New MySqlConnection()

'Selecciono la tabla a la que quiero consultar
Dim query9 As String
'conexion a utilizar
Dim micomando9 As New MySqlCommand(query9, conexion9)

'abro la conexión
Try
    conexion9.Open()
Catch ex As MySql.Data.MySqlClient.MySqlException
    MsgBox("No se ha podido establecer " & vbCrLf & "la
conexión con la base de datos.", MsgBoxStyle.Critical)
Finally

End Try

'EJECUTO COMANDO
Dim milector9 As MySqlDataReader = micomando9.ExecuteReader

Try
    While milector9.Read()

        If milector9.GetValue(2).ToString = msktbxPlaca.Text
Then

            If milector9.GetValue(1).ToString = "PCadisan" Then
                MsgBox("OJO PLACA REGISTRADA EN OTRO
                PARQUEADERO", MsgBoxStyle.Information)
                msktbxPlaca.Focus()

            End If

            If milector9.GetValue(1).ToString = "PRecreo" Then
                MsgBox("OJO PLACA REGISTRADA EN OTRO
                PARQUEADERO", MsgBoxStyle.Information)
                msktbxPlaca.Focus()

            End If

            If milector9.GetValue(1).ToString = "PPintado" Then
                MsgBox("OJO PLACA REGISTRADA EN OTRO
                PARQUEADERO", MsgBoxStyle.Information)
                msktbxPlaca.Focus()

            End If

            If milector9.GetValue(1).ToString = "PTejar" Then
                MsgBox("OJO PLACA REGISTRADA EN OTRO
                PARQUEADERO", MsgBoxStyle.Information)
                msktbxPlaca.Focus()

            End If

            If milector9.GetValue(1).ToString = "PMontufar" Then
                MsgBox("OJO PLACA REGISTRADA EN OTRO
                PARQUEADERO", MsgBoxStyle.Information)
                msktbxPlaca.Focus()

            End If

            If milector9.GetValue(1).ToString = "PLaRonda" Then
                MsgBox("OJO PLACA REGISTRADA EN OTRO
                PARQUEADERO", MsgBoxStyle.Information)
                msktbxPlaca.Focus()

            End If

            If milector9.GetValue(1).ToString = "POfelia" Then

```

```

        MsgBox("OJO PLACA REGISTRADA EN OTRO
              PARQUEADERO", MsgBoxStyle.Information)
        msktbxPlaca.Focus()
    End If
    If milector9.GetValue(1).ToString = "PQuicentroN" Then
        MsgBox("OJO PLACA REGISTRADA EN OTRO
              PARQUEADERO", MsgBoxStyle.Information)
        msktbxPlaca.Focus()
    End If
    If milector9.GetValue(1).ToString = "PRioCoca" Then
        MsgBox("OJO PLACA REGISTRADA EN OTRO
              PARQUEADERO", MsgBoxStyle.Information)
        msktbxPlaca.Focus()
    End If

    Else

        If milector9.GetValue(2).ToString = msktbxPlaca.Text Then
            MsgBox("PLACA REGISTRADA EN OTRO PARQUEADERO
                  PARQUEADERO", MsgBoxStyle.Information)
        End If
    End If
End While
Catch ex As Exception
End Try

'cierro la conexión
End If
fin:
End Sub
Private Sub Form4_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
    lblTituloFacturar.BackColor = Color.Transparent
    lblTituloParqueadero.BackColor = Color.Transparent
    lblPlaca.BackColor = Color.Transparent
    grbxResultado.BackColor = Color.Transparent
    grbxFechaHora.BackColor = Color.Transparent
    lblFecha.Text = Date.Now.ToLongDateString
    grboxParqueaderos.BackColor = Color.Transparent
    lblEdisponibles.BackColor = Color.Transparent
    lblEocupadas.BackColor = Color.Transparent
    lblNdisp.BackColor = Color.Transparent
    lblNocup.BackColor = Color.Transparent

    'MUESTRO EL # DE PARQUEADEROS DISPONIBLES Y OCUPADOS
    lblNdisp.Text = Form1.MiStringA
    lblNocup.Text = 100 - Form1.MiStringA

    Timer1.Enabled = True
    msktbxPlaca.Focus()

End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBuscar.Click
    buscar()
End Sub

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick

```

```

    'MUESTRO CADA SEGUNDO EN EL RELOJ
    Label9.Text = Now.ToLongTimeString
    Timer1.Interval = 1000
End Sub

Private Sub btnRegresar_Click(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles btnRegresar.Click
    Dim regresoPP As New PPrincipal
    regresoPP.Show()
    Me.Hide()
End Sub

Private Sub btnFacturar_Click(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles btnFacturar.Click
    If String.IsNullOrEmpty(msktbxPlaca.Text) = True Then
        MsgBox("Primero Ingrese la placa a ser facturada ",
            msktbxPlaca.Focus())
        Return
    Else
        If String.IsNullOrEmpty(txtId.Text) = True Then
            MsgBox("PRIMERO REALICE LA CONSULTA",
                Return)
        End If

        If btnBuscar.CausesValidation = True Then
            If Form1.MiStringA = 100 Then
                MsgBox("PARQUEADERO VACÍO", MsgBoxStyle.Critical)
            Else

'*****Elimino registro de base de datos*****
                Dim conexion As New
                Dim query As String
                Dim micomando As New MySqlCommand(query, conexion)

                Try
                    conexion.Open()
                Catch ex As MySql.Data.MySqlClient.MySqlException
                Finally

                End Try

                micomando.ExecuteNonQuery()
                conexion.Close()

                Dim i As Integer
                Plaza = Variables.Plaza

'*****SI SE REPITE LA PLAZA AL MOMENTO DE INSERTARLA NUEVAMENTE*****
                For i = 0 To Form1.MiStringA - 1
                    If Form1.lstbxParqueaderosL.Items.Item(i).ToString = Plaza.ToString Then
                        Dim s As Integer
                        For s = 0 To Form1.MiStringA - 1
                            '*****si son = y hay un espacio " " *****
                            If
                                Form1.lstbxParqueaderosL.Items.Item(s).ToString = " ".ToString Then

                                Form1.lstbxParqueaderosL.Items.Remove(" ")

                                Form1.lstbxParqueaderosL.Items.Add(Plaza + 1)

```

```

Form1.lstbxParqueaderosO.Items.Add(Plaza + 1)
                                MsgBox( "FACTURADO" ,
MsgBoxStyle.MsgBoxRight)
                                Form1.MiStringA = Form1.MiStringA + 1
                                GoTo fuera
                                Else

Form1.lstbxParqueaderosL.Items.Add(Plaza + 1)

Form1.lstbxParqueaderosO.Items.Add(Plaza + 1)

                                MsgBox( "FACTURADO" ,
MsgBoxStyle.Exclamation)
                                Form1.MiStringA = Form1.MiStringA + 1
                                GoTo fuera
                                End If
                                Next

                                End If

                                Next

'*****SI NO SE REPITE LA PLAZA *****
Dim j As Integer
For j = 0 To Form1.MiStringA - 1
    If
Form1.lstbxParqueaderosL.Items.Item(j).ToString = " ".ToString Then
        Form1.lstbxParqueaderosL.Items.Remove(" ")
        Form1.lstbxParqueaderosL.Items.Add(Plaza)

        Form1.lstbxParqueaderosO.Items.Add(Plaza)

        MsgBox( "FACTURADO" , MsgBoxStyle.MsgBoxRight)
        Form1.MiStringA = Form1.MiStringA + 1
        GoTo fuera
    End If

Next

Form1.lstbxParqueaderosL.Items.Add(Plaza)

Form1.lstbxParqueaderosO.Items.Add(Plaza)

MsgBox( "FACTURADO" , MsgBoxStyle.MsgBoxRight)
Form1.MiStringA = Form1.MiStringA + 1
GoTo fuera

fuera:

Dim regresoform2 As New PPrincipal
regresoform2.Show()
Me.Hide()

End If

```

```

        End If

    End If

End Sub

    Private Sub msktbxPlaca_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles msktbxPlaca.KeyDown
        If e.KeyCode = 13 Then
            buscar()
        End If
    End Sub

End Class

```

Usuarios

```

Imports MySql.Data.MySqlClient
Public Class Ver_Usuarios
    Private Sub Ver_Base_de_Datos_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load
        lblTituloBase.BackColor = Color.Transparent
        lblId.BackColor = Color.Transparent
        Label9.BackColor = Color.Transparent
        grbxId.BackColor = Color.Transparent

        Dim conexion As New MySqlConnection
        conexion.Open()

        Dim table As DataTable
        Dim adaptador As MySqlDataAdapter
        Dim builder As MySqlCommandBuilder

        table = New DataTable

        builder = New MySqlCommandBuilder(adaptador)
        adaptador.Fill(table)
        dtgvUsuarios.DataSource = table

    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim regresar As New PPrincipal
        regresar.Show()
        Me.Hide()

    End Sub

Public Class OrdenarColumnaNumerica
    Implements System.Collections.IComparer

    Private sortOrderModifier As Integer = 1
    Private Col As Integer = 0

    Public Sub New(ByVal sortOrder As SortOrder, ByVal columna As
Integer)
        Col = columna
        If sortOrder = sortOrder.Descending Then
            sortOrderModifier = -1

```

```

        ElseIf sortOrder = sortOrder.Ascending Then
            sortOrderModifier = 1
        End If
    End Sub
    Public Function Compare(ByVal x As Object, ByVal y As Object) As
Integer _
        Implements System.Collections.IComparer.Compare

        Dim DataGridViewRow1 As DataGridViewRow = CType(x,
DataGridViewRow)
        Dim DataGridViewRow2 As DataGridViewRow = CType(y,
DataGridViewRow)
        Return (Math.Sign(CLng(DataGridViewRow1.Cells(Col).Value) -
CLng(DataGridViewRow2.Cells(Col).Value))) * sortOrderModifier
    End Function
End Class

    Private Sub dtgvUsuarios_ColumnHeaderMouseClick(ByVal sender As
Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs)
Handles dtgvUsuarios.ColumnHeaderMouseClick
        Static Orden As Integer = 1

        If dtgvUsuarios.Rows.Count > 0 Then
            If dtgvUsuarios.Columns(e.ColumnIndex).Name = "Puntuacion"
Then
                If Orden = 1 Then
                    dtgvUsuarios.Sort(New
OrdenarColumnaNumerica(SortOrder.Ascending, e.ColumnIndex))
                    Orden = 0
                Else
                    dtgvUsuarios.Sort(New
OrdenarColumnaNumerica(SortOrder.Descending, e.ColumnIndex))
                    Orden = 1
                End If
            End If
        End If
    End Sub
End Class

```


Gracias

```

Imports MySql.Data.MySqlClient
Public Class Gracias
    Public _Contador As Integer
    Public MiStringA As String = Form1.MiStringA
    Public IDParqueadero As String
    Public txtID As String = Form4.txtId.Text
    Private Sub Gracias_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        lblGracias.BackColor = Color.Transparent
        lblEtiqueta.BackColor = Color.Transparent
        Timer1.Enabled = True
        InicioProceso()

    End Sub
    Public Sub InicioProceso()

        _Contador = ProgressBar1.Minimum

        lblEtiqueta.Text = "Iniciando..."

        Me.Cursor = Cursors.WaitCursor

        BackgroundWorker1.WorkerSupportsCancellation = True

        BackgroundWorker1.WorkerReportsProgress = True

        BackgroundWorker1.RunWorkerAsync()

    End Sub

    Public Sub FinProceso()

        BackgroundWorker1.CancelAsync()

    End Sub
    Private Sub BackgroundWorker1_DoWork(ByVal sender As System.Object,
ByVal e As System.ComponentModel.DoWorkEventArgs) Handles
BackgroundWorker1.DoWork
        While Not BackgroundWorker1.CancellationPending

            If Not BackgroundWorker1.CancellationPending Then

                If _Contador = Me.ProgressBar1.Maximum Then _Contador =
ProgressBar1.Minimum Else _Contador += 5

                Threading.Thread.Sleep(100)

                BackgroundWorker1.ReportProgress(_Contador, "Borrando
datos...")

            Else

                e.Cancel = True

                Exit While

            End If

        End While
    End Sub

```

```

    End While

End Sub

Private Sub BackgroundWorker1_ProgressChanged(ByVal sender As Object,
ByVal e As System.ComponentModel.ProgressChangedEventArgs) Handles
BackgroundWorker1.ProgressChanged

    ProgressBar1.Value = e.ProgressPercentage

    lblEtiqueta.Text = e.UserState
End Sub

Private Sub BackgroundWorker1_RunWorkerCompleted(ByVal sender As
Object, ByVal e As System.ComponentModel.RunWorkerCompletedEventArgs)
Handles BackgroundWorker1.RunWorkerCompleted

    If Not (e.Error Is Nothing) Then

        ' There was an error during the operation.
        Dim msg As String = String.Format("Error: {0}",
e.Error.Message)
        MessageBox.Show(msg)

    Else

        ' The operation completed normally.
        ProgressBar1.Value = ProgressBar1.Minimum
        lblEtiqueta.Text = "Borrando datos..."

    End If

    Me.Cursor = Cursors.Default
    ProgressBar1.Value = ProgressBar1.Maximum

End Sub

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
    Timer1.Enabled = False

    For j = 0 To 99 - MiStringA Step 1
        '*****CONSULTA *****

        Dim conexion1 As New MySqlConnection()

        Dim query1 As String          'conexion a utilizar"
        Dim micomando1 As New MySqlCommand(query1, conexion1)

        Try
            conexion1.Open()
        Catch ex As MySql.Data.MySqlClient.MySqlException
            MsgBox("No se ha podido establecer " & vbCrLf & "la
conexión con la base de datos.", MsgBoxStyle.Critical)
        Finally

        End Try

        'EJECUTO COMANDO

```

```

Dim miletor1 As MySqlDataReader = micomando1.ExecuteReader

'*****CONSULTO Y MUESTRO*****
Try
    While miletor1.Read()
        IDParqueadero = Variables.IDParqueadero
        If miletor1.GetValue(1).ToString =
IDParqueadero.ToString Then

            txtID = miletor1.GetValue(0).ToString() 'Id

        End If

    End While

Catch ex As Exception
End Try
'cierro la conexión
conexion1.Close()

'*****Elimino PLAZAS de base de datos*****
Dim conexion As New MySqlConnection()
Dim query As String
Dim micomando As New MySqlCommand(query, conexion)

Try
    conexion.Open()
Catch ex As MySql.Data.MySqlClient.MySqlException
    MsgBox("No se ha podido establecer " & vbCrLf & "la
conexión con la base de datos.", MsgBoxStyle.Critical)
Finally

End Try

micomando.ExecuteNonQuery()
Threading.Thread.Sleep(10)
conexion.Close()

Next

FinProceso()

Application.Exit()
End Sub
End Class

```

CÓDIGO FUENTE APLICACIÓN SERVIDOR

Form 1

```
Public Class Form1

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        'HABILITO TIMER
        Timer1.Enabled = True

    End Sub

    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
        'DESHABILITO EL TIMER
        Timer1.Enabled = False

        'LLAMO A LA SIGUIENTE VENTANA
        Dim z As New INGRESO
        z.Show()
        Me.Hide()

    End Sub

End Class
```

Ingreso

```
Public Class INGRESO
    Public aux As Integer

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        iniciar()
    End Sub

    Public Sub iniciar()
        aux = variables.aux

        If TextBox1.Text = "xxxxxxx" Then

            If TextBox2.Text = "xxxxx" Then
                lblCargando.ForeColor = Color.Black
                lblCargando.Text = "Consultando Base de Datos..."
                lblCargando.Visible = True
                pgbIngreso.Visible = True
                lblCargando.Visible = True
                Dim k As Integer
                For k = 0 To 99
                    lblCargando.Visible = True
                    pgbIngreso.Value = k
                    Threading.Thread.Sleep(40)
                Next

                aux = 0
                variables.aux = aux
            End If
        End If
    End Sub
End Class
```

```

        Dim datosms As New RECEPCIÓN_DE_DATOS_SMS()
        datosms.Show()
        Me.Hide()

    Else
        MessageBox.Show("Clave incorrecta... ")
        aux = aux + 1
        variables.aux = aux

    End If
Else
    MessageBox.Show("Nombre de usuario incorrecto... ")
    aux = aux + 1
End If
If aux = 3 Then
    MessageBox.Show("Número de intentos exedidos")
    aux = 0
    Application.Exit()
End If
TextBox1.Clear()
TextBox2.Clear()
Refresh()
TextBox1.Focus()

End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Application.Exit()
End Sub

Private Sub TextBox1_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles TextBox1.KeyDown
    If e.KeyCode = Keys.Enter Then
        TextBox2.Focus()
    End If
End Sub

Private Sub TextBox2_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles TextBox2.KeyDown
    If e.KeyCode = Keys.Enter Then
        Button1.Focus()
    End If
End Sub

Private Sub INGRESO_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    TextBox1.Focus()
    pgbIngreso.Visible = False
    lblCargando.Visible = False

End Sub

Private Sub TextBox2_KeyPress(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.KeyPressEventArgs) Handles TextBox2.KeyPress

    If e.KeyChar = ChrW(13) Then

```

```

        iniciar()
    End If
End Sub

Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles TextBox1.TextChanged

    End Sub
End Class

```

Recepción de datos sms

```

Public Class RECEPCIÓN_DE_DATOS_SMS

    Public Data As String
    Public comando As String
    Public gradlat As String
    Public mindlat As String
    Public segdlat As String
    Public numero As String

    Public gradlon As String
    Public mindlon As String
    Public segdlon As String
    Public distancia As Double

    Public n As Integer
    Public s As Integer
    Public trama As String, tramal As String
    Public indicelat As Integer, indicelong As Integer
    Public LATITUD As String, LONGITUD As String
    Public indicel As String, tramnumero As String, tconf As String,
indconf As String, conf As String

    Private Sub RECEPCIÓN_DE_DATOS_SMS_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        grbxEstado.BackColor = Color.Transparent
        lblParqueadero.BackColor = Color.Transparent
        lblLatitud.BackColor = Color.Transparent
        lblLongitud.BackColor = Color.Transparent
        lblDistancia.BackColor = Color.Transparent
        lblPlazas.BackColor = Color.Transparent
        lblDisponibles.BackColor = Color.Transparent
        lblOcupadas.BackColor = Color.Transparent
        chkDirecciones.BackColor = Color.Transparent
        lblDirecciones.BackColor = Color.Transparent
        lblTitulo.BackColor = Color.Transparent
        lblMsjEstado.BackColor = Color.Transparent
        lblEstado2.BackColor = Color.Transparent
        grbxFechaHora.BackColor = Color.Transparent
        lblFecha.Text = Date.Now.ToLongDateString
        Timer2.Enabled = True

        '*****CONFIGURACIÓN MODEM GSM*****

        SerialPort1.Open()

        'variables
        Dim configsim As String, configeco As String, configudp As String

```

```

'comandos config modem

'comunicación con el modem
SerialPort1.WriteLine(configeco)
Threading.Thread.Sleep(200)
SerialPort1.WriteLine(configsim)
Threading.Thread.Sleep(200)
SerialPort1.WriteLine(configudp)
Threading.Thread.Sleep(200)

'*****CONSULTO EL NÚMERO DE PARQUEADEROS REGISTRADOS (OJO: El
número es solo seteable desde la base de *****
Dim conexion0 As New MySqlConnection()
'Descargo datos de la tabla

Dim query0 As String
Dim micomando0 As New MySqlCommand(query0, conexion0)

Try
    conexion0.Open()
Catch ex As MySql.Data.MySqlClient.MySqlException
    MsgBox("No se ha podido establecer " & vbCrLf & "la conexión
con la base de datos.", MsgBoxStyle.Critical)
Finally

End Try

'ejecuto comando
Dim milector0 As MySqlDataReader = micomando0.ExecuteReader

Try
    While milector0.Read()
        n = milector0.GetValue(7).ToString()
'ref N (número de parqueaderos)
        variables.n = n
        n = variables.n

    End While
Catch ex As Exception

End Try

'cierro la conexión
conexion0.Close()

Dim i As Integer

'*****CONSULTO TODA LA BASE PARA MOSTRAR EN PANTALLA*****
Dim conexion As New MySqlConnection()
For i = 1 To n
    Dim query As String
    Dim micomando As New MySqlCommand(query, conexion)

    Try
        conexion.Open()
        Catch ex As MySql.Data.MySqlClient.MySqlException
            MsgBox("No se ha podido establecer " & vbCrLf & "la
conexión con la base de datos.", MsgBoxStyle.Critical)

```

```

    Finally

    End Try

    Dim milector As MySqlDataReader = micomando.ExecuteReader

    Try
        While milector.Read()

lstdbxParqueaderos.Items.Add(milector.GetValue(1).ToString()) 'NOMBRE
lstdbxLatitud.Items.Add(milector.GetValue(2).ToString()) 'LATITUD
lstdbxLongitud.Items.Add(milector.GetValue(3).ToString()) 'LONGITUD
lstdbxDirecciones.Items.Add(milector.GetValue(4).ToString()) 'DIRECCION
lstdbxDisponibles.Items.Add(milector.GetValue(5).ToString()) 'DISPONIBLES
lstdbxOcupadas.Items.Add(milector.GetValue(6).ToString()) 'OCUPADOS

                End While
        Catch ex As Exception

        End Try

        conexion.Close()

    Next

    '*****HABILITA TIMER DE RECONOCIMIENTO DE SMS*****
    Timer1.Enabled = True

End Sub

Private Sub SerialPort1_DataReceived(ByVal sender As Object, ByVal e As System.IO.Ports.SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived

    Threading.Thread.Sleep(100)
    Data = SerialPort1.ReadExisting()

End Sub

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick

    '*****deshabilito el timer*****
    Timer1.Enabled = False

    '*****limpiar listbox*****
    lstdbxDistancia.Items.Clear()

    lblMsjEstado.Visible = True
    lblEstado2.Visible = False

```



```

Try
  Dim s As Integer
  For s = 0 To 7

    '*****LEO SMS ENTRANTES CADA CIERTO TIEMPO*****
'*****FILTRADO DE INFORMACIÓN CADA 2 SEGUNDOS*****

    trama = Data
    tramal = Data
    tconf = tramal.Substring(indconf, 11)
    conf = tconf.ToString.Substring(9, 2)

    If conf = "OK" Then
      GoTo siguiente
    End If

'*****FILTRO NÚMERO TELEFÓNICO*****

    Try

      Dim tipo As String
      Dim datotipo As String

'*****trama numero*****
      indicel = tramal.IndexOf("CMGR", 0)

      lblMsjEstado.Visible = False
      lblEstado2.ForeColor = Color.Red
      lblEstado2.Visible = True

      tipo = tramnumero.IndexOf("UNREAD", 0)
      If tipo = 11 Then
        datotipo = tramnumero.Substring(tipo, 21)
        Dim numeropp As String

        If numeropp.StartsWith("0") Then
          numero = datotipo.ToString.Substring(9, 9)
          GoTo filtro

        Else
          Dim numerop As String =
            numero = "0" + numerop
          GoTo filtro
          If numero = "0+1234" Then

            Dim bor As String
            Threading.Thread.Sleep(200)

            Dim bor1 As String
            SerialPort1.WriteLine(bor1)
            Threading.Thread.Sleep(200)
            GoTo fin
          End If
        End If
      End If
    Else
      Dim tipol As String

```

```

        tipol = tramnumero.IndexOf("READ", 0)

        If tipol = 11 Then

            Dim del As String
            Threading.Thread.Sleep(200)

            Dim delete As String
            Threading.Thread.Sleep(200)
            GoTo siguiente

        End If
    End If

Catch ex As Exception

End Try

filtro:

Try

    '*****latitud y longitud en grados*****

    indicelat = trama.IndexOf("CLAT:", 0)
    LATITUD = trama.Substring(indicelat, 16)
    indicelong = trama.IndexOf("CLONG:", 0)
    LONGITUD = trama.Substring(indicelong, 18)

    gradlat = LATITUD.Substring(6, 2)
    mindlat = LATITUD.Substring(8, 7)
    segdlat = LATITUD.Substring(15, 1)

    gradlon = LONGITUD.Substring(7, 3)
    mindlon = LONGITUD.Substring(10, 7)
    segdlon = LONGITUD.Substring(17, 1)

    variables.gradlat = gradlat
    variables.mindlat = mindlat
    variables.segdlat = segdlat
    variables.gradlon = gradlon
    variables.mindlon = mindlon
    variables.segdlon = segdlon

    lstbxDisponibles.Items.Clear()
    lstbxOcupadas.Items.Clear()

    '*****CONSULTA DE PARQUEADEROS DISPONIBLES Y OCUPADOS*****

    Dim conexion As New MySqlConnection)

    For i = 1 To n
        Dim query As String

        Dim micomando As New MySqlCommand

        Try
            conexion.Open()
        Catch ex As MySql.Data.MySqlClient.MySqlException
        Finally

```

```

End Try
Dim milector As MySqlConnection
Try
    While milector.Read()

lstdisponibles.Items.Add(milector.GetValue(5).ToString()) 'DISPONIBLES

lstdisponibles.Items.Add(milector.GetValue(6).ToString()) 'OCUPADOS

        End While
    Catch ex As Exception

End Try

'cierro la conexión
conexion.Close()

Next

'*****conversión °' a grados decimales datos pda*****
'+++++++ latitud en grados ++++++

Dim mingradlat As Double
Dim totgradlat As Double

If segdlat = "N" Then 'seglat es el hemisferio
Else
    totgradlat = -totgradlat
End If

'+++++++ longitud en grados ++++++
Dim mingradlong As Double
Dim totgradlong As Double

mingradlong = Convert.ToDouble(mindlon) / 60
mingradlong
totgradlong = Convert.ToDouble(gradlon) +

If segdlon = "E" Then
Else
    totgradlong = -totgradlong
End If

'*****latitud y longitud en
radianes*****
Dim latrad As Double, longrad As Double, latparq As
Double, longparq As Double
Dim tlatparq As Double, tlongparq As Double
Dim pi As Double
pi = 3.14159265358979
latrad = totgradlat * pi / 180
longrad = totgradlong * pi / 180

```

```

'*****CALCULO DE DISTANCIA*****
  Dim j As Integer

  For j = 0 To n - 1
    tlatparq = Convert.ToDouble(lstbxLatitud.Items(j))
    tlongparq = Convert.ToDouble(lstbxLongitud.Items(j))

    latparq = tlatparq * pi / 180
    longparq = tlongparq * pi / 180

'+++++ingreso de datos a fórmula+++++
    Dim P As Double, D As Double

    'FORMULA
    P = Math.Sin(latrad) * Math.Sin(latparq) +
Math.Cos(latrad) * Math.Cos(latparq) * Math.Cos(longrad - longparq)
    D = Math.Acos(P)
    distancia = D * 6378.1366
    lstbxDistancia.Items.Add(distancia.ToString)

  Next

'*****comparación distancia*****
  Dim A As Double, B As Double, k As Integer
  Dim aux As Double

  A = Convert.ToDouble(lstbxDistancia.Items(0))
  aux = A
  B = Convert.ToDouble(lstbxDistancia.Items(1))

  For k = 0 To n - 1
    If k < n - 1 Then
      If A < B Then
        aux = A
        If k = n - 2 Then
          GoTo finalgoritmo
        Else
          B = Convert.ToDouble(lstbxDistancia.Items(k.ToString + 2))

          End If

        Else
          aux = B
          A = aux

          B = Convert.ToDouble(lstbxDistancia.Items(k.ToString + 2))
          End If
        End If
      End If

finalgoritmo:

    If k = n - 1 Then
      If A < B Then
        aux = A

```

```

B = Convert.ToDouble(lstbxDistancia.Items(k.ToString))
    Else
        aux = B
        A = aux
B = Convert.ToDouble(lstbxDistancia.Items(k.ToString))
    End If
End If

Next

Dim r As Integer, cont As Integer
For r = 0 To n - 1
    If aux = lstbxDistancia.Items(r.ToString) Then
        cont = r
    End If
Next

Dim dir As String
dir = LbxDirecciones.Items(cont.ToString)
Dim disp As String
disp = lstbxDisponibles.Items(cont.ToString)
Dim ocup As String
ocup = lstbxOcupadas.Items(cont.ToString)

'*****envío de sms a pda*****
Dim enviodato As String, msj As String

Try

    enviodato = "AT+CMGS=" + ChrW(34) + numero
    SerialPort1.WriteLine(enviodato)
    Threading.Thread.Sleep(200)

    msj = "PARQ_+CERC: " + dir + " DISP: '" + disp
+ "' OCUP: " + ocup + ChrW(26)
    SerialPort1.WriteLine(msj)
    Threading.Thread.Sleep(200)

Try
'*****BORRO SMS*****

    Dim del As String
    del = "AT+CMGD=" + s.ToString + Chr(13)
    SerialPort1.WriteLine(del)
    Threading.Thread.Sleep(200)

    Dim delete As String
    delete = "AT+CMGD=" + s.ToString + ",1"
    SerialPort1.WriteLine(delete)
    Threading.Thread.Sleep(200)

'*****

Catch ex As Exception

```

```

        End Try
    Catch ex As Exception

        End Try

    Catch ex As Exception

    End Try

    Dim delete1 As String
    delete1 = "AT+CMGD=" + s.ToString
    SerialPort1.WriteLine(delete1)
    Threading.Thread.Sleep(200)

    Dim delete2 As String
    delete2 = "AT+CMGD=" + s.ToString + ",0"
    SerialPort1.WriteLine(delete2)
    Threading.Thread.Sleep(200)

siguiente:
    Next

    Catch ex As Exception

    End Try

fin:

'*****HABILITO NUEVAMENTE EL TIMER*****
    Timer1.Enabled = True

'*****

    End Sub

    Private Sub park_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnIEParq.Click

        If MessageBox.Show("Si ingresa un nuevo parqueadero dejará de
leer los sms entrantes, Está seguro?", "Atención",
MessageBoxButtons.YesNo, MessageBoxIcon.Question) = DialogResult.No Then
            Refresh()
        Else
            SerialPort1.Close()
            Dim verpark As New Parqueaderos()
            verpark.Show()
            Me.Hide()
        End If

    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)

    End Sub

```

```

    Private Sub chkDirecciones_CheckedChanged(ByVal sender As Object,
    ByVal e As System.EventArgs) Handles chkDirecciones.CheckedChanged
        If chkDirecciones.Checked = True Then
            lblDirecciones.Visible = True
            LbxDirecciones.Visible = True
            chkDirecciones.Text = "Direcciones Habilitado"
        End If
        If chkDirecciones.Checked = False Then
            lblDirecciones.Visible = False
            LbxDirecciones.Visible = False
            chkDirecciones.Text = "Direcciones Deshabilitado"
        End If
    End Sub

    Private Sub ListBox1_SelectedIndexChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    lstbxParqueaderos.SelectedIndexChanged
        lstbxParqueaderos.SelectedIndex = lstbxLatitud.SelectedIndex

    End Sub

    Private Sub lblLongitud_Click(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles lblLongitud.Click

    End Sub

    Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles Timer2.Tick
        Label9.Text = Now.ToLongTimeString
        Timer1.Interval = 1000
    End Sub

    Private Sub btnSalir_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnSalir.Click
        Dim Msg As String, Style As String, Title As String, Response As
    String, MyString As String
        Msg = "Seguro que desea SALIR de la aplicación?" ' Define el
    mensaje.
        ' Define botones:
        Style = vbYesNo + MsgBoxStyle.Question
        Title = "ATENCIÓN!" ' Define título.

        Response = MsgBox(Msg, Style, Title) ' Muestra el mensaje
        If Response = vbYes Then ' Si el usuario escoge Yes.
            SerialPort1.Close()
            Application.Exit()
        Else ' El usuario escoge No.
            Return
        End If

    End Sub

End Class

Parqueaderos
Imports MySql.Data.MySqlClient
Public Class Parqueaderos
    Public n As Integer

```

```

Dim htparqueaderos As New Hashtable
Public Sub busqueda()
'*****CONSULTA PREVIA BORRADO DE PAQUEADERO*****

    If String.IsNullOrEmpty(txtNombreP.Text) = True Then
        MsgBox("Ingrese el parqueadero a eliminar",
MsgBoxStyle.Information)
    Else

        Dim conexion As New MySqlConnection()

        Dim query As String = "SELECT * FROM `parqueaderos` WHERE
`Nombre` LIKE '" + txtNombreP.Text + "'"

        'conexion a utilizar"
Dim micomando As New MySqlCommand(query, conexion)

        'abro la conexión
Try
            conexion.Open()
        Catch ex As MySql.Data.MySqlClient.MySqlException
            MsgBox("No se ha podido establecer " & vbCrLf & "la
conexión con la base de datos.", MsgBoxStyle.Critical)
        Finally

        End Try

        'EJECUTO COMANDO
Dim miclector As MySqlDataReader = micomando.ExecuteReader

        Try
            While miclector.Read()

                txtNombreN.Text = miclector.GetValue(1).ToString()
                txtLatitudN.Text = miclector.GetValue(2).ToString()
                txtLongitudN.Text = miclector.GetValue(3).ToString()
                txtDireccionN.Text = miclector.GetValue(4).ToString()
                txtID.Text = miclector.GetValue(0).ToString() 'ID

            End While
        Catch ex As Exception

        End Try
        conexion.Close()
    End If
    Refresh()
End Sub

Private Sub Parqueaderos_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load
    lblTitulo.BackColor = Color.Transparent
    grbxFechaHora.BackColor = Color.Transparent
    lblFecha.Text = Date.Now.ToLongDateString
    Timer1.Enabled = True
    txtNombreP.Focus()
End Sub

```



```

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
    lblhora.Text = Now.ToLongTimeString
    Timer1.Interval = 1000
End Sub

Private Sub btnEliminarP_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnEliminarP.Click
    If String.IsNullOrEmpty(txtNombreN.Text) = True Then
        MsgBox("PRIMERO REALICE LA BUSQUEDA", MsgBoxStyle.Critical)
        Return
    End If

'*****Elimino registro de base de datos*****

    If String.IsNullOrEmpty(txtNombreP.Text) = True Then
        MsgBox("Primero busque el parqueadero deseado",
MsgBoxStyle.Information)
    Else
        Dim conexion As New MySqlConnection()
        Dim query As String
        Dim micomando As New MySqlCommand(query, conexion)

        Try
            conexion.Open()
        Catch ex As MySql.Data.MySqlClient.MySqlException
            MsgBox("No se ha podido establecer " & vbCrLf & "la
conexión con la base de datos.", MsgBoxStyle.Critical)
        Finally

        End Try

        micomando.ExecuteNonQuery()
        conexion.Close()

        MessageBox.Show("Parqueadero borrado de base de datos!")
        MsgBox("Parqueadero borrado de base de datos!",
MsgBoxStyle.Information, "Listo!")

        Dim regresoform2 As New Parqueaderos
        regresoform2.Show()

        n = variables.n
        n = n - 1
        variables.n = n

        Dim conexion1 As New MySqlConnection()

        Dim query1 As String
        Dim micomando1 As New MySqlCommand(query1, conexion1)

        Try
            conexion1.Open()
        Catch ex As Exception

        End Try

```

```

micomando1.ExecuteNonQuery()
conexion1.Close()

txtNombreP.Clear()
txtNombreN.Clear()
txtLatitudN.Clear()
txtLongitudN.Clear()
txtDireccionN.Clear()
txtID.Clear()

End If
txtNombreP.Focus()
Refresh()

End Sub

Private Sub btnBuscarP_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnBuscarP.Click
    busqueda()
End Sub

Private Sub btnIngresarP_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnIngresarP.Click
    revisar:
    If String.IsNullOrEmpty(lblNombreP.Text) = True Then
        MsgBox("INGRESE DATO DE IDENTIFICACIÓN",
        MsgBoxStyle.Critical)
        lblNombreP.Focus()

        Return

    End If

    If mskbxPOcup.Text = "- ." Then
        MsgBox("INGRESE UNA LATITUD", MsgBoxStyle.Critical)
        msktbxLatitud.Focus()

        Return

    End If

    If String.IsNullOrEmpty(msktbxLatitud.Text) = True Then
        MsgBox("INGRESE UNA LATITUD", MsgBoxStyle.Critical)
        msktbxLatitud.Focus()

        Return

    End If

    If String.IsNullOrEmpty(msktbxLongitud.Text) = True Then
        MsgBox("INGRESE UNA LONGITUD", MsgBoxStyle.Critical)
        msktbxLongitud.Focus()

        Return

    End If

    If String.IsNullOrEmpty(mskbxPOcup.Text) = True Then
        MsgBox("INGRESE UN NÚMERO", MsgBoxStyle.Critical)
        mskbxPOcup.Focus()

```

```

        Return
    End If

    If String.IsNullOrEmpty(mskbpDisp.Text) = True Then
        MsgBox("INGRESE UNA NÚMERO", MsgBoxStyle.Critical)
        mskbpDisp.Focus()

        Return
    End If

    If String.IsNullOrEmpty(lblDireccionP.Text) = True Then
        MsgBox("INGRESE UNA DIRECCIÓN", MsgBoxStyle.Critical)
        lblDireccionP.Focus()

        Return
    End If

n = variables.n
n = n + 1
variables.n = n

'*****ACTUALIZO N*****
Dim conexion1 As New MySqlConnection()

Dim query1 As String
Dim micomando1 As New MySqlCommand(query1, conexion1)

Try
    conexion1.Open()
Catch ex As Exception

End Try

micomando1.ExecuteNonQuery()
conexion1.Close()

'*****INGRESO PARQUEADERO*****
Dim conexion As New MySqlConnection()
Dim query As String
Dim micomando As New MySqlCommand(query, conexion)

Try
    conexion.Open()
Catch ex As Exception

End Try

'EJECUTO EL COMANDO
micomando.ExecuteNonQuery()

conexion.Close()

```

```

'*****
        MsgBox("INGRESADO: " & lblNombreP.Text & "          LATITUD: " &
msktbxLatitud.Text & "          LONGITUD: " & msktbxLongitud.Text,
MsgBoxStyle.Information, "LISTO!")

'*****

        htparqueaderos = New Hashtable()
        ' añadir valores
        htparqueaderos.Add("ID", lblNombreP.Text)
        htparqueaderos.Add("Latitud", msktbxLatitud.Text)
        htparqueaderos.Add("Longitud", msktbxLongitud.Text)

        Me.lblNombreP.Clear()
        Me.msktbxLatitud.Clear()
        Me.msktbxLongitud.Clear()
        Me.mskbxPOcup.Clear()
        Me.mskbxPDisp.Clear()
        Me.lblDireccionP.Clear()

        Return
fin:
    End Sub

    Private Sub lblDireccionP_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles lblDireccionP.KeyDown
        If e.KeyCode = 13 Then
            btnIngresarP.Focus()
        End If
    End Sub

    Private Sub msktbxLatitud_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles msktbxLatitud.KeyDown
        If e.KeyCode = 13 Then
            msktbxLongitud.Focus()
        End If
    End Sub

    Private Sub msktbxLongitud_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles msktbxLongitud.KeyDown
        If e.KeyCode = 13 Then
            mskbxPOcup.Focus()
        End If
    End Sub

    Private Sub lblNombreP_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles lblNombreP.KeyDown
        If e.KeyCode = 13 Then
            msktbxLatitud.Focus()
        End If
    End Sub

    Private Sub txtNombreN_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtNombreN.TextChanged
        If String.IsNullOrEmpty(txtNombreN.Text) = False Then

```

```
        btnEliminarP.Focus()  
    End If  
End Sub  
  
Private Sub btnRegresar_Click(ByVal sender As System.Object, ByVal e  
As System.EventArgs) Handles btnRegresar.Click  
    Dim REGRESAR As New RECEPCIÓN_DE_DATOS_SMS  
    REGRESAR.Show()  
    Me.Hide()  
End Sub  
  
Private Sub mskbxPOcup_KeyDown(ByVal sender As Object, ByVal e As  
System.Windows.Forms.KeyEventArgs) Handles mskbxPOcup.KeyDown  
    If e.KeyCode = 13 Then  
        mskbxPDisp.Focus()  
    End If  
End Sub  
  
Private Sub mskbxPDisp_KeyDown(ByVal sender As Object, ByVal e As  
System.Windows.Forms.KeyEventArgs) Handles mskbxPDisp.KeyDown  
    If e.KeyCode = 13 Then  
        lblDireccionP.Focus()  
    End If  
End Sub  
  
Private Sub txtNombreP_KeyDown(ByVal sender As Object, ByVal e As  
System.Windows.Forms.KeyEventArgs) Handles txtNombreP.KeyDown  
    If e.KeyCode = 13 Then  
        busqueda()  
    End If  
End Sub  
  
Private Sub tbpgIngreso_Click(ByVal sender As System.Object, ByVal e  
As System.EventArgs) Handles tbpgIngreso.Click  
    lblNombreP.Focus()  
End Sub  
  
Private Sub tbpgEliminación_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles tbpgEliminación.Click  
    txtNombreP.Focus()  
End Sub  
End Class
```

CÓDIGO FUENTE APLICACIÓN PDA

Form 1

```

Public Class Form1
    Public DatoInicial As String
        'DATOS DE FILTRO
    Public gradoslat As String, minutoslat As String, hemisferiolat As
String
    Public gradoslong As String, minutoslong As String, hemisferiolong As
String

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        'habilito el timer
        Timer1.Enabled = True

    End Sub

    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
        Timer1.Enabled = False
        Dim k As Integer

        For k = 0 To 99 Step 4
            pgbIngreso.Value = k
            Threading.Thread.Sleep(40)
        Next

        Dim menu As New MENU
        menu.Show()
        Me.Hide()
    End Sub
End Class

```

Menú

```

Imports System.IO.Ports
Public Class MENU

    Inherits System.Windows.Forms.Form
    Delegate Sub delegado(ByVal data As String)
    Public DatoInicial As String
    Public Uno As String, Dos As String, Tres As String, Data2 As String

    'DATOS DE FILTRO
    Public gradoslat As String, minutoslat As String, hemisferiolat As
String
    Public gradoslong As String, minutoslong As String, hemisferiolong As
String
    Public chk As String
    Public trama As String

    Public Sub Filtradodatos()

```

```

        Dim valid As String, valid1 As String, valid2 As String, valid3
As String
        Dim indice As Integer
        Dim int As Integer, intel As Integer, inte2 As Integer
        int = 0
        intel = 0
        inte2 = 0

        Dim k As Integer

        For k = 0 To 99 Step 4
            prgbrBotonPMC.Visible = True
            prgbrBotonPMC.Value = k
            Threading.Thread.Sleep(40)
        Next
        prgbrBotonPMC.Visible = False

```

conexion:

```

        SrlPDatosMenu.Open()
        Threading.Thread.Sleep(1000)
        Application.DoEvents()

        If SrlPDatosMenu.IsOpen = True Then
            trama = Datoinicial

            Try
                SrlPDatosMenu.Close()
            Catch ex As Exception
                MsgBox("no se cierra el puerto")
            End Try

        End If

        '+++++

        int = int + 1
        If int = 3 Then
            MsgBox("GSP NO DISPONIBLE, Intente nuevamente!",
MsgBoxStyle.Exclamation)
            ' int = 0
            GoTo fin
        End If
        GoTo conexion
    End Try

    '*****FILTRADO DE DATOS*****

    If String.IsNullOrEmpty(trama) = True Then

        MessageBox.Show("Dato no leído, intentando leer")

        intel = intel + 1
        If intel = 3 Then
            MsgBox("GSP NO SE POSICIONA, Revise el equipo!",
MsgBoxStyle.Exclamation)
            intel = 0

```

```

        GoTo fin
    End If

    GoTo conexion

Else
filtro:
    indice = trama.IndexOf("$GPRMC", 0)

    valid = trama.Substring(indice, 44)
    valid1 = valid
    valid2 = valid
    valid3 = valid

    '*****LATITUD*****
    'filtro grados de latitud
    gradoslat = valid1.Substring(20, 2)

    'filtro minutos de latitud
    minutoslat = valid2.Substring(22, 7)

    'filtro hemisferio de latitud
    hemisferiolat = valid3.Substring(30, 1)

    '*****LONGITUD*****
    'filtro grados longitud
    gradoslong = valid.Substring(32, 3)

    'filtro minutos longitud
    minutoslong = valid1.Substring(35, 7)

    'filtro hemisferio longitud
    hemisferiolong = valid2.Substring(43, 1)

    '*****ALMACENANDO LATITUD Y LONGITUD*****
    Variables.gradoslat = gradoslat
    Variables.minutoslat = minutoslat
    Variables.hemisferiolat = hemisferiolat
    Variables.gradoslong = gradoslong
    Variables.minutoslong = minutoslong
    Variables.hemisferiolong = hemisferiolong

    '*****VERIFICANDO LATITUD Y LONGITUD*****
    ListBox1.Items.Add(gradoslat)
    ListBox1.Items.Add(minutoslat)
    ListBox1.Items.Add(hemisferiolat)
    ListBox1.Items.Add(gradoslong)
    ListBox1.Items.Add(minutoslong)
    ListBox1.Items.Add(hemisferiolong)

End If

chk = gradoslat + minutoslat

If chk = "" Then
    GoTo conexion

```



```

        End If

fin:

    End Sub

    Private Sub SrlPDatosMenu_DataReceived(ByVal sender As Object, ByVal
e As System.IO.Ports.SerialDataReceivedEventArgs) Handles
SrlPDatosMenu.DataReceived
        Threading.Thread.Sleep(500)
        Dim As String = SrlPDatosMenu.ReadExisting
        Datoinicial = s

    End Sub

'*****programa principal*****

    Private Sub MENU_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Uno = "1x"
        Dos = "2x"
        Tres = "3x"
        cmbbxZoom.Items.Add(Uno)
        cmbbxZoom.Items.Add(Dos)
        cmbbxZoom.Items.Add(Tres)

        'DATOS DE FILTRO

        gradoslat = Variables.gradoslat
        minutoslat = Variables.minutoslat
        hemisferiolat = Variables.hemisferiolat
        gradoslong = Variables.gradoslong
        minutoslong = Variables.minutoslong
        hemisferiolong = Variables.hemisferiolong

        Me.wbMapaMenu.Url = New
System.Uri("http://maps.google.com/maps/api/staticmap?maptype=roadmap&cen
ter=" + gradoslat + " " + minutoslat + " " + hemisferiolat + "," +
gradoslong + " " + minutoslong + " " + hemisferiolong +
"&zoom=16&size=400x400&markers=color:blue|label:A|" + gradoslat + " " +
minutoslat + " " + hemisferiolat + "," + gradoslong + " " + minutoslong +
" " + hemisferiolong + "&sensor=true&mobile=true")

    End Sub

    Private Sub cmbbxZoom_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cmbbxZoom.SelectedIndexChanged
        gradoslat = Variables.gradoslat
        minutoslat = Variables.minutoslat
        hemisferiolat = Variables.hemisferiolat
        gradoslong = Variables.gradoslong
        minutoslong = Variables.minutoslong
        hemisferiolong = Variables.hemisferiolong
        Dim zoom As String
        If cmbbxZoom.SelectedItem = "1x" Then
            zoom = 14
        End If
        If cmbbxZoom.SelectedItem = "2x" Then

```

```

        zoom = 16
    End If
    If cmbbxZoom.SelectedItem = "3x" Then
        zoom = 18
    End If

    Refresh()

    Me.wbMapaMenu.Url = New
System.Uri("http://maps.google.com/maps/api/staticmap?maptype=roadmap&cen
ter=" + gradoslat + " " + minutoslat + " " + hemisferiolat + "," +
gradoslong + " " + minutoslong + " " + hemisferiolong + "&zoom=" +
zoom.ToString +
"&size=400x400&markers=color:blue&markers=color:blue|label:A|" +
gradoslat + " " + minutoslat + " " + hemisferiolat + "," + gradoslong + "
" + minutoslong + " " + hemisferiolong + "&sensor=true&mobile=true")

    End Sub

    Private Sub btnConsulta_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnConsulta.Click
    petición:

        Filtradodatos()

        chk = gradoslat + minutoslat

        If chk = "0000.0000" Then

            MsgBox("ESPERANDO LOCALIZACIÓN DE SATÉLITES",
MsgBoxStyle.Information)
            Dim recargo As New MENU
            recargo.Show()

        Else
            If chk = "" Then
                'GoTo petición
                MsgBox("REALICE OTRA CONSULTA", MsgBoxStyle.Information)
                GoTo fin
            End If

            '*****ENVIO INFO POR SMS*****

            Try

                MessageBox.Show("GRACIAS POR ESPERAR, PRONTO RECIBIRÁ UN
SMS CON LA INFORMACIÓN REQUERIDA")

                Refresh()

            Catch ex As Exception

            End Try

        End If

        gradoslat = Variables.gradoslat

```

```

minutoslat = Variables.minutoslat
hemisferiolat = Variables.hemisferiolat
gradoslong = Variables.gradoslong
minutoslong = Variables.minutoslong
hemisferiolong = Variables.hemisferiolong

Refresh()

Me.wbMapaMenu.Url = New
System.Uri("http://maps.google.com/maps/api/staticmap?maptype=roadmap&cen
ter=" + gradoslat + " " + minutoslat + " " + hemisferiolat + "," +
gradoslong + " " + minutoslong + " " + hemisferiolong +
"&zoom=16&size=400x400&markers=color:blue&markers=color:blue|label:A|" +
gradoslat + " " + minutoslat + " " + hemisferiolat + "," + gradoslong + "
" + minutoslong + " " + hemisferiolong + "&sensor=true&mobile=true")

fin:

End Sub

Private Sub btnInfoPP_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnInfoPP.Click
    Try
        SrlPDatosMenu.Close()

    Catch ex As Exception
        MsgBox(ex.ToString, "", MessageBoxButtons.OK)
    End Try

    Dim pico As New PICO_Y_PLACA
    pico.Show()
    Me.Hide()
End Sub

Private Sub MenuItem8_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MenuItem8.Click
    Dim fuera As New HASTA_PRONTO_
    fuera.Show()
    Me.Hide()
End Sub

Private Sub MenuItem9_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MenuItem9.Click

    Dim menu As New MENU
    menu.Show()
    Me.Hide()
End Sub

Private Sub btnActualizar_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btnActualizar.Click

    Filtradodatos()

'DATOS DE FILTRO

gradoslat = Variables.gradoslat
minutoslat = Variables.minutoslat
hemisferiolat = Variables.hemisferiolat
gradoslong = Variables.gradoslong

```

```

minutoslong = Variables.minutoslong
hemisferiolong = Variables.hemisferiolong

'MUESTRO EN WEB BROWSER
Me.wbMapaMenu.Url = New
System.Uri("http://maps.google.com/maps/api/staticmap?maptype=roadmap&center=" + gradoslat + " " + minutoslat + " " + hemisferiolat + "," +
gradoslong + " " + minutoslong + " " + hemisferiolong +
"&zoom=16&size=400x400&markers=color:blue|label:A|" + gradoslat + " " +
minutoslat + " " + hemisferiolat + "," + gradoslong + " " + minutoslong +
" " + hemisferiolong + "&sensor=true&mobile=true")

End Sub
Private Sub SrlPDatosMenu_PinChanged(ByVal sender As Object, ByVal e
As System.IO.Ports.SerialPinChangedEventArgs) Handles
SrlPDatosMenu.PinChanged
    If e.EventType = SerialPinChange.CDChanged Then
        If SrlPDatosMenu.CDHolding Then
            MsgBox("Carrier detect on")
        Else
            MsgBox("Carrier detect off")
        End If
    End If
End Sub
End Class

```

Pico y placa

```

Public Class PICO_Y_PLACA

    Private Sub PICO_Y_PLACA_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load

        Dim fecha As String
        Dim datodia As String, datodiam As String
        Dim lunes As String
        Dim martes As String
        Dim miercoles As String
        Dim jueves As String
        Dim viernes As String
        Dim sabado As String, domingo As String

        lblFecha.Text = Date.Now.ToLongDateString
        datodia = lblFecha.Text.Trim.Substring(0, 9)
        datodiam = lblFecha.Text.Trim.Substring(0, 10)
        lunes = datodia.Trim.Substring(0, 5)
        martes = datodia.Trim.Substring(0, 6)
        miercoles = datodiam.Trim.Substring(0, 9)
        jueves = datodia.Trim.Substring(0, 6)
        viernes = datodia.Trim.Substring(0, 7)
        sabado = datodia.Trim.Substring(0, 6)
        domingo = datodia.Trim.Substring(0, 7)

        If lunes = "lunes" Then
            Label2.Text = "1 y 2"
        End If

        If martes = "martes" Then

```

```

        Label2.Text = "3 y 4"
    End If

    If miercoles = "miércoles" Then
        Label2.Text = "5 y 6"
    End If

    If jueves = "jueves" Then
        Label2.Text = "7 y 8"
    End If

    If viernes = "viernes" Then
        Label2.Text = "9 y 0"
    End If

    If sabado = "sábado" Then
        Label2.Text = "NO RESTRICCIÓN"
    End If

    If domingo = "domingo" Then
        Label2.Text = "NO RESTRICCIÓN"
    End If
End Sub

Private Sub MenuItem1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MenuItem1.Click
    Dim rmenu As New MENU
    rmenu.Show()
    Me.Hide()
End Sub

Private Sub MenuItem3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MenuItem3.Click
    Dim todospp As New PARQUEADEROS_PICO_Y_PLACA
    todospp.Show()
    Me.Hide()
End Sub

Private Sub DateTimePicker1_ValueChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim fecha As String
    Dim datodia As String
    Dim lunes As String
    Dim martes As String
    Dim miercoles As String
    Dim jueves As String
    Dim viernes As String

    fecha = lblFecha.Text
    datodia = fecha.Trim.Substring(0, 9)
    lunes = datodia.Trim.Substring(0, 5)
    martes = datodia.Trim.Substring(0, 6)
    miercoles = datodia.Trim.Substring(0, 9)
    jueves = datodia.Trim.Substring(0, 6)
    viernes = datodia.Trim.Substring(0, 7)

    If lunes = "lunes" Then
        Label2.Text = "1 y 2"
    End If

```

```

If martes = "martes" Then
    Label2.Text = "3 y 4"
End If

If miercoles = "miércoles" Then
    Label2.Text = "5 y 6"
End If

If jueves = "jueves" Then
    Label2.Text = "7 y 8"
End If

If viernes = "viernes" Then
    Label2.Text = "9 y 0"
End If

End Sub
End Class

```

Parqueaderos pico y placa

```

Public Class PARQUEADEROS_PICO_Y_PLACA

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        Dim parkdborde As New PARQUEADEROS_DE_BORDE_TODOS
        parkdborde.Show()
        Me.Hide()

    End Sub

    Private Sub MenuItem3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MenuItem3.Click
        Dim menuppp As New MENU
        menuppp.Show()
        Me.Hide()

    End Sub

    Private Sub MenuItem1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MenuItem1.Click
        Dim rpiscop As New PICO_Y_PLACA
        rpiscop.Show()
        Me.Hide()

    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
        Dim interm As New INTERMODALES
        interm.Show()
        Me.Hide()

    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
        Dim todoscentro As New CENTRO_QUIITO

```

```
        todoscentro.Show()  
        Me.Hide()  
  
    End Sub  
  
    Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Button5.Click  
        Dim zona As New ZONA_AZUL  
        zona.Show()  
        Me.Hide()  
  
    End Sub  
  
    Private Sub PARQUEADEROS_PICO_Y_PLACA_Load(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles MyBase.Load  
  
    End Sub  
End Class
```