



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN MANTENIMIENTO AUTOMOTRIZ

TEMA:

**“DISEÑO, CONSTRUCCIÓN E IMPLEMENTACIÓN DE UN TAXÍMETRO
CON ALMACENAMIENTO DE VIAJES MEDIANTE GPS EN EL VEHÍCULO
CHEVROLET CORSA WIND 1.4”**

**PLAN DE TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO
DE INGENIERO EN MANTENIMIENTO AUTOMOTRIZ.**

AUTOR:

MORALES BAYETERO CÉSAR FABRICIO

DIRECTOR:

ING. FREDY ROSERO MSC.

IBARRA, 2016



UNIVERSIDAD TÉCNICA DEL NORTE BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

La Universidad Técnica del Norte dentro del proyecto Repositorio Digital Institucional, determinó la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	100351001-1		
APELLIDOS Y NOMBRES:	MORALES BAYETERO CÉSAR FABRICIO		
DIRECCIÓN:	ATUNTAQUI-CALLE DE LA SUB ESTACIÓN Y GARCÍA MORENO		
EMAIL:	cesarf_mb92@hotmail.com		
TELÉFONO FIJO:		TELÉFONO MÓVIL:	0986619791

DATOS DE LA OBRA	
TÍTULO:	“DISEÑO, CONSTRUCCIÓN E IMPLEMENTACIÓN DE UN TAXÍMETRO CON ALMACENAMIENTO DE VIAJES MEDIANTE GPS EN EL VEHÍCULO CHEVROLET CORSA WIND 1.4”
AUTOR:	MORALES BAYETERO CÉSAR FABRICIO
FECHA: AAAAMMDD	2016/03/22
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TÍTULO POR EL QUE OPTA:	INGENIERÍA EN MANTENIMIENTO AUTOMOTRIZ
ASESOR /DIRECTOR:	ING. FREDY ROSERO MSC.

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, **CÉSAR FABRICIO MORALES BAYETERO**, con cédula de identidad Nro. **1003510011**, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la

disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior Artículo 144.

3. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 22 días del mes de marzo del 2016.

EL AUTOR:



Firma:

Nombre: César Fabricio Morales Bayetero

Cédula: 100351001-1



UNIVERSIDAD TÉCNICA DEL NORTE

CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, **CÉSAR FABRICIO MORALES BAYETERO**, con cédula de identidad Nro. **1003510011**, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículos 4, 5 y 6, en calidad de autor de la obra o trabajo de grado denominado: **“DISEÑO, CONSTRUCCIÓN E IMPLEMENTACIÓN DE UN TAXÍMETRO CON ALMACENAMIENTO DE VIAJES MEDIANTE GPS EN EL VEHÍCULO CHEVROLET CORSA WIND 1.4”**, que ha sido desarrollado para optar por el título de: **INGENIERÍA EN MANTENIMIENTO AUTOMOTRIZ** en la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Técnica del Norte.

Ibarra, a los 22 días del mes de marzo del 2016.

Firma:

Nombre: César Fabricio Morales Bayetero

Cédula: 100351001-1

CERTIFICACIÓN

Certifico que el presente proyecto fue realizado en su totalidad por el señor: Morales Bayetero César Fabricio, como requerimiento para la obtención del título de Ingeniería en Mantenimiento Automotriz.

Atentamente,

A handwritten signature in blue ink, appearing to read 'Fredy Rosero', is centered within a light blue rectangular box. The signature is written in a cursive style.

Ing. Fredy Rosero Msc.

DIRECTOR DEL PROYECTO

AGRADECIMIENTO

Agradezco a Dios, quien me dio la vida, salud y sabiduría para cumplir con éxito esta meta.

A la Universidad Técnica del Norte, por haberme facilitado todas las herramientas necesarias a lo largo de esta carrera.

Al Ing. Fredy Rosero Msc. Director del Proyecto, quien fue la persona que me guio para realizar con éxito el trabajo de grado.

A mi familia, por brindarme en todo momento su cariño y apoyo incondicional, lo que me ha permitido alcanzar todas mis metas y anhelos.

César Morales B.

DEDICATORIA

Dedico el presente trabajo a mi madre, por haberme apoyado en todo momento y ser el pilar fundamental en mi vida, como también por ser mi apoyo incondicional durante el transcurso de mi carrera, que por medio de sus consejos y motivaciones constantes me han permitido día a día ser una mejor persona.

A mi hermano, mi cuñada y mis sobrinas, por brindarme toda su compañía en momentos de alegrías y tristezas, por todo su apoyo moral, espiritual y por todas sus enseñanzas a lo largo de la vida.

A mi gran amigo Andrés Arévalo, a quien Dios ha decidido tenerlo a su lado, por toda su amistad brindada, por tantas risas y alegrías, que más que un amigo fue como un hermano, quien formó parte de mi vida durante largos años, formando parte del desarrollo de la presente tesis. Amigo siempre te recordaré.

César Morales B.

ÍNDICE

	pág.
CARÁTULA.....	i
AUTORIZACIÓN DE USO Y PUBLICACIÓN	ii
CESIÓN DE DERECHOS DE AUTOR	iv
CERTIFICACIÓN	v
AGRADECIMIENTO	vi
DEDICATORIA	vii
ÍNDICE DE CONTENIDO	viii
ÍNDICE DE FIGURAS	xv
ÍNDICE DE TABLAS	xv
ÍNDICE DE ABREVIATURAS.....	xvi
RESUMEN	xviii
ABSTRACT	xix
INTRODUCCIÓN	xx
CAPÍTULO I.....	1
1. CONTEXTUALIZACIÓN DEL PROBLEMA	1
1.1 Antecedentes.	1
1.2 Planteamiento del problema.	1
1.3 Formulación del problema.	2
1.4 Delimitación temporal y espacial.	2
1.5 Objetivos.	2
1.5.1 Objetivo General.	2
1.5.2 Objetivos Específicos.	2
1.6 Justificación.....	3

CAPÍTULO II	4
2. MARCO TEÓRICO.	4
2.1 Movilidad.	4
2.1.1 Movilidad urbana.....	5
2.1.2 Movilidad humana.....	6
2.2 Sistema de posicionamiento global GPS.....	6
2.2.1 Segmento Espacial.....	7
2.2.2 Segmento de control	8
2.2.3 Segmento de usuario.....	9
2.2.4 Funcionamiento.	9
2.2.5 Aplicaciones del GPS	10
2.3 Taxímetro.	10
2.3.1 Características y accesorios	11
2.3.2 Funcionamiento	12
2.3.3 Tarifas.....	13
2.4 Sistemas embebidos.	14
2.4.1 Componentes de un sistema embebido.....	15
2.4.2 Microcontroladores y sistemas embebidos.....	15
2.5 Arduino.	17
2.5.1 Placa hardware libre.	18
2.5.2 Software.....	18
2.5.3 Lenguaje de programación libre de Arduino	19
2.5.4 Programación de la placa Arduino.	19
2.5.5 Ventajas de utilizar Arduino.....	23
2.6 LabVIEW.	24
2.6.1 Entorno.	25
2.6.2 Características.....	27

CAPÍTULO III	31
3. METODOLOGÍA DE LA INVESTIGACIÓN	31
3.1 Tipo de investigación.	31
3.1.1 Investigación bibliográfica.	31
3.1.2 Investigación tecnológica.	31
3.2 Métodos.....	31
3.3 Técnicas e instrumentos.	32
CAPÍTULO IV	33
4. DISEÑO Y CONSTRUCCIÓN DEL TAXÍMETRO.	33
4.1 Diseño de hardware.....	33
4.1.1 Diagrama de bloques.	33
4.1.2 Selección de la placa Arduino.	35
4.1.3 Selección del módulo GPS.	38
4.1.4 Selección de impresora.	41
4.1.5 Selección de LCD.....	43
4.2 Diseño de software.	48
4.2.1 Etapas de diseño.	48
4.2.2 Flujogramas del dispositivo taxímetro.....	48
4.2.3 Diseño del programa de análisis de datos en el software LabVIEW.	57
4.2.4 Comunicación del dispositivo con el software LabVIEW.	67
4.2.5 Sectorización de la ciudad de Ibarra.....	67
4.2.6 Diagramación de circuitos electrónicos.....	70
CAPÍTULO V	73
5. PRUEBAS DE FUNCIONAMIENTO Y ANÁLISIS DE DATOS.	73
5.1 Prueba de comunicación del GPS.	73

5.2 Navegación por el menú de control.	74
5.2.1 Portada.	75
5.2.2 Reloj.	75
5.2.3 Inicio.	76
5.2.4 Fin.	76
5.3 Prueba de funcionamiento del taxímetro.	77
5.4 Prueba de impresión.	80
5.5 Prueba de almacenamiento de datos.	81
5.6 Prueba del software de análisis de datos en LabVIEW.	82
5.6.1 Importación de datos.	82
5.6.2 Estadística origen y destino.	84
5.6.3 Estadística general.	85
5.6.4 Representación gráfica.	85
5.7 Implementación del dispositivo.	87
CAPÍTULO VI	89
6. CONCLUSIONES Y RECOMENDACIONES	89
6.1 Conclusiones.	89
6.2 Recomendaciones.	90
BIBLIOGRAFÍA	91
ANEXOS	93

ÍNDICE DE FIGURAS

	pág.
Figura 1. Satélite Navstar GPS.	7
Figura 2. Las tres partes del GPS.....	7
Figura 3. Constelación de satélites.	8
Figura 4. Segmento de control.....	8
Figura 5. Receptor GPS actual.....	9
Figura 6. Principio de la situación por satélite.	10
Figura 7. Cartel de taxi libre.	11
Figura 8. Taxímetro actual.....	12
Figura 9. Posiciones de funcionamiento del taxímetro.....	13
Figura 10. Sistemas embebidos en todas las áreas.	14
Figura 11. Microcontrolador Atmega328.	16
Figura 12. Arduino Mega 2560.	17
Figura 13. Ejemplo de panel frontal.	26
Figura 14. Ejemplo de diagrama de bloques.	26
Figura 15. Diagrama de bloques.....	33
Figura 16. Diagrama de elementos del prototipo.....	34
Figura 17. Diagrama microcontrolador Atmega2560.....	37
Figura 18. Módulo GPS NEO-6M-0-001.	38
Figura 19. Diagrama GPS NEO-6M.....	40
Figura 20. Impresora térmica.....	41
Figura 21. Alimentación impresora térmica.	42
Figura 22. TFT 3.2” SSD1289.....	43
Figura 23. Diagrama lcd TFT táctil.	45
Figura 24. Pines utilizados por la pantalla táctil TFT en la placa Arduino.	45
Figura 25. Modelo de resistencias de la pantalla táctil.....	47

Figura 26. Etapas de diseño.	48
Figura 27. Flujograma del funcionamiento del dispositivo.	49
Figura 28. Flujograma del programa de inicio.	50
Figura 29. Flujograma de subrutina de menú.	51
Figura 30. Flujograma de subrutina de reloj.	52
Figura 31. Flujograma de subrutina de inicio.	54
Figura 32. Flujograma de subrutina de fin.	55
Figura 33. Flujograma de subrutina de almacenamiento.	56
Figura 34. Flujograma de subrutina home.	56
Figura 35. Diagrama de flujo del programa de análisis de datos.	57
Figura 36. Ventana principal.	58
Figura 37. Ventana de importación de datos.	59
Figura 38. Código para la importación de datos.	59
Figura 39. Código para la creación del archivo Excel estadísticas.	60
Figura 40. Código para eliminar datos.	61
Figura 41. Ventana de estadística origen y destino.	61
Figura 42. Código para importar datos del número de carreras realizadas.	62
Figura 43. Control del rango de fechas.	62
Figura 44. Ventana de estadística general.	63
Figura 45. Código para establecer el rango de fechas.	63
Figura 46. Código para importar datos generales.	64
Figura 47. Ventana de representación gráfica.	64
Figura 48. Ventana del programa Power BI.	65
Figura 49. Código para ejecutar el programa Power BI.	66
Figura 50. Ventana del programa U-center.	66
Figura 51. Código para ejecutar el programa U-center.	67
Figura 52. Comunicación del dispositivo con el Software LabVIEW.	67

Figura 53. Criterios de sectorización.	68
Figura 54. Mapa de las zonas de la ciudad de Ibarra.	69
Figura 55. Circuito para adquisición de datos del GPS.	70
Figura 56. Circuito de almacenamiento de datos.	71
Figura 57. Circuito de conexión entre TFT 3.2” y Arduino.	71
Figura 58. Circuito de conexión entre la impresora térmica y Arduino.	72
Figura 59. Datos del GPS a través del monitor serial.	73
Figura 60. Ventana principal.	75
Figura 61. Ventana Reloj.	75
Figura 62. Ventana de inicio de taxímetro.	76
Figura 63. Ventana de fin de taxímetro.	76
Figura 64. Funcionamiento en la tarifa diurna.	77
Figura 65. Funcionamiento en la tarifa nocturna.	77
Figura 66. Aumento de tarifa por tiempo de espera.	78
Figura 67. Aumento de tarifa por distancia recorrida.	78
Figura 68. Subtotal menor al valor de la carrera mínima.	79
Figura 69. Subtotal mayor al valor de la carrera mínima.	79
Figura 70. Impresión del recibo.	80
Figura 71. Recibo impreso.	80
Figura 72. Datos almacenados en el archivo Excel.	81
Figura 73. Datos almacenados en el archivo de Texto.	81
Figura 74. Portada del programa en LabVIEW.	82
Figura 75. Funcionamiento de la ventana datos (parte 1).	83
Figura 76. Funcionamiento de la ventana datos (parte 2).	83
Figura 77. Ventana estadística origen-destino (parte 1).	84
Figura 78. Ventana estadística origen-destino (parte 2).	84
Figura 79. Ventana estadística general.	85

Figura 80. Ventana representación gráfica.	86
Figura 81. Ventana del programa Power BI.	86
Figura 82. Ventana del programa U-center.	87
Figura 83. Fuente de alimentación.....	88
Figura 84. Dispositivos instalados en el vehículo.	88

ÍNDICE DE TABLAS

	pág.
Tabla 1. Cuadro tarifario de taxis.	13
Tabla 2. Comandos programa Arduino.	20
Tabla 3. Variables programa Arduino.	20
Tabla 4. Operadores Programa Arduino.	21
Tabla 5. Comparadores Programa Arduino.	21
Tabla 6. Estructuras utilizadas en el programa Arduino.....	22
Tabla 7. Librerías para los diferentes elementos.	22
Tabla 8. Evolución de las diferentes versiones de LabVIEW.	24
Tabla 9. Datos técnicos Arduino Mega 2560.	35
Tabla 10. Especificaciones GPS.	39
Tabla 11. Parámetros de funcionamiento GPS.	39
Tabla 12. Pines de conexión a la placa Arduino.....	40
Tabla 13. Especificaciones TFT 3.2".	44
Figura 36. Ventana principal.	58
Figura 37. Ventana de importación de datos.	59
Figura 38. Código para la importación de datos.	59
Tabla 14. Lugares sectorizados en la ciudad de Ibarra.	68
Tabla 15. Función de los iconos del menú.	74

ÍNDICE DE ABREVIATURAS

A	Amperios.
AREF	Referencia de tensión del convertidor analógico a digital.
ATMEL	Compañía de semiconductores y microcontroladores.
AVR	Familia de microcontroladores de fabricante estadounidense.
BPS	Baudios, velocidad para la transmisión de datos.
CAN	Bus de control para los sistemas electrónicos del automóvil.
CPU	Unidad central de proceso.
DC	Corriente continua.
EPROM	Memoria ROM programable borrable.
g	Gramos.
GND	Conexión a tierra.
GPS	Sistema de Posicionamiento Global.
GPRS	Servicio general de paquetes vía radio.
IDE	Entorno de desarrollo integrado.
IOREF	Tensión de entrada y salida de la placa Arduino.
KB	Kilobyte.
LCD	Pantalla de cristal líquido.
mA	Miliamperios.
MHz	Megahercio.
MISO	Recepción de datos al microcontrolador.
mm	Milímetros.
MOSI	Envío de datos desde el microcontrolador.

PWM	Modulación por ancho de pulsos.
RAM	Memoria de acceso aleatorio.
RAW	Formato de archivo digital de imágenes.
RF	Radiofrecuencia.
RSET	Reiniciar.
RX	Línea para la recepción de datos.
SCK	Pulso que marca la sincronización de datos.
SS	Línea para seleccionar un dispositivo electrónico de salida.
SV	Vehículo espacial.
TFT	Transistor de películas finas.
TTL	Lógica transistor a transistor.
TX	Transmisión de datos.
UARTS	Transmisor, receptor asíncrono universal.
USB	Bus universal en serie.
V	Voltios.
VI	Instrumento virtual.
WIRING	Marco de programación de código abierto.

RESUMEN

En la actualidad, se puede evidenciar que la principal problemática del Cantón San Miguel de Ibarra en la gestión del transporte público local, es la falta de información sobre los viajes de las personas dentro de la ciudad y zonas de origen y destino. Esto se debe a que no se posee ningún sistema que proporcione dicha información, lo que representa una falencia dentro del sistema institucional, limitando el desarrollo y ejecución de proyectos que solucionen los problemas de movilidad mediante la reestructuración del sistema de transporte público. En el presente trabajo se realiza la descripción del diseño y construcción de un taxímetro con almacenamiento de viajes utilizando la tecnología del Sistema de Posicionamiento Global (GPS), que permita el estudio de la movilidad urbana en la ciudad, mediante la recolección y análisis de los datos obtenidos a través de su funcionamiento. Para la construcción del dispositivo la programación es un paso fundamental, puesto que inicialmente se establece la comunicación del módulo GPS con la plataforma de hardware libre Arduino y posteriormente se realiza la programación las tarifas acorde a los parámetros establecidos en las leyes de la Agencia Nacional de Tránsito, Transporte Terrestre y Seguridad Vial. La ciudad de Ibarra es dividida en 28 zonas, a través de las cuales se realiza el estudio de origen y destino para determinar que parte necesita de más atención. Para ello, la construcción del dispositivo se efectúa a través de la utilización del sistema embebido Arduino y el análisis de datos se lo realiza a través del Software de Desarrollo de Sistemas NI LabVIEW, con el cuál se obtiene información que permita a la autoridad de Tránsito local, reestructurar las rutas y frecuencias del sistema de transporte público de la ciudad.

ABSTRACT

At present, it is possible to demonstrate that the main problem of the Canton San Miguel of Ibarra in the management of local public transport, is the lack of information on travel of people within the city and areas of origin and destination. This is because there is not possessed any system that provides such information, which represents a flaw in the institutional system, limiting the development and project execution that solve mobility problems by restructuring the public transport system. In this paper the is done the description of the design and construction of a taximeter with storage of travel using the technology of Global Positioning System (GPS), which allows the study of urban mobility in the city by collecting and analyzing data obtained through its operation. For the construction of the device, programming is a critical step, since initially is set the communication of the module GPS with the platform free hardware Arduino and then it is performed the programming of the tariffs according to the parameters established in the laws of the National Agency Transit, Land Transport and Road Safety. The city of Ibarra is divided into 28 zones, across which is realized the study of origin and destination to determine which part needs more attention. For it, the construction of the device is effected through the use of embedded system Arduino and data analysis it is done through the Software Development Systems NI LabVIEW, with which is obtained information allowing the Transit authority local, restructure routes and frequencies of the public transport system of the city.

INTRODUCCIÓN

El presente trabajo tiene como objetivo principal el diseño y construcción de un taxímetro con almacenamiento de viajes mediante GPS, así como también la adquisición de datos a través de su funcionamiento para su posterior análisis, logrando de esta manera obtener información acerca de la movilidad urbana en la ciudad de Ibarra que permita mejorar los problemas relacionados al sistema de transporte público.

En la actualidad la autoridad de tránsito no posee algún sistema que proporcione información sobre los viajes de las personas dentro de la ciudad y zonas de origen y destino, lo que limita el desarrollo de proyectos que solucionen los problemas de movilidad mediante la reestructuración del sistema de transporte local.

El proyecto se desarrolla en la Universidad Técnica del Norte, tomando como área de estudio a la ciudad de Ibarra, de la cual se obtienen los datos proporcionados por el dispositivo y de los cuales se realiza el respectivo análisis. La investigación consta de seis capítulos:

En el primer capítulo se presenta la contextualización del problema, que está conformada por antecedentes, el planteamiento y formulación del problema, objetivo general y específicos, así como también delimitación temporal y espacial, , además de la justificación.

En el capítulo II se aborda el marco teórico de la investigación, en el que se indica conceptos utilizados en la temática, es así que se muestra los diferentes tipos y la importancia de la movilidad en la ciudad, diferenciando la movilidad urbana de la movilidad humana ya que abarcan diferentes campos, además como el GPS es muy importante para la obtención de datos y las zonas de origen y destino se indica su principio de funcionamiento como también de los segmentos que está formado; así también se muestra los modos de funcionamiento del taxímetro y las tarifas para las cuales está regulado. Para finalizar el capítulo se abordan conceptos y forma de programación tanto del programa Arduino como del software LabVIEW, debido a que los dos son de suma importancia para la adquisición, procesamiento y almacenamiento de datos durante el funcionamiento del dispositivo.

En el capítulo III se presenta la metodología de la investigación que contiene el tipo de investigación, métodos y técnicas aplicadas para el desarrollo del proyecto y el cumplimiento de los objetivos planteados.

En el cuarto capítulo se indica el diseño y construcción del dispositivo que está conformado por dos partes principales. En la primera parte se muestra el diseño del hardware, en el que se detalla los elementos electrónicos utilizados para el buen funcionamiento del dispositivo, brindando datos específicos de la TFT táctil, GPS, placa Arduino mega y de la impresora térmica para la generación de los recibos. En la segunda parte se presenta información sobre el desarrollo del software, en el que se indica los flujogramas de los diferentes modos de funcionamiento del taxímetro, así como también la sectorización de la ciudad de Ibarra realizada a través de los datos de longitud y latitud brindados por el GPS. Al final del capítulo se indica los diagramas electrónicos de las conexiones de los diferentes elementos con la placa Arduino mega, que es la que se encarga de dar las instrucciones para el funcionamiento del taxímetro.

En el capítulo V se presentan las pruebas de funcionamiento realizadas por el dispositivo instalado en el vehículo, así como también el análisis de datos de los viajes realizados. En este capítulo se indica la forma de funcionamiento del dispositivo con las diferentes ventanas de las que está conformado y la información que cada una presenta, además de la forma de adquisición de datos a través del software LabVIEW, el cual está conformado por cuatro ventanas que presenta los datos generales de los viajes realizados, estadística origen y destino que se presenta en forma de matriz para conocer el número de viajes realizados desde y hacia los lugares sectorizados, estadística general para presentar información dentro de un determinado rango de fechas y conocer datos sobre los kilómetros recorridos, tiempo de espera y de viaje, como de los costos generados tanto en el día como en la noche, así también en la última ventana se puede representar gráficamente los lugares de inicio y fin de carrera a través del programa Power BI y las rutas seguidas por el vehículo mediante el programa U-center.

Finalmente en el sexto capítulo se presentan las conclusiones y recomendaciones que fueron fruto de la investigación realizada, y en la parte final del trabajo se indica la bibliografía consultada y anexos de gran importancia, en el que se incluye un manual de usuario del funcionamiento del dispositivo como del software de adquisición de datos, además del código de programación.

CAPÍTULO I

1. CONTEXTUALIZACIÓN DEL PROBLEMA

1.1 Antecedentes.

En Ecuador están registrados 42 689 taxis por la Agencia Nacional de Tránsito. De esa cantidad, los propietarios de 32 023 vehículos cumplieron con la disposición de instalar el taxímetro.

En la ciudad de Ibarra existen 940 taxis convencionales y 402 ejecutivos de las diferentes cooperativas y compañías, de los cuales pocos de ellos encienden el taxímetro y la mayoría se exime de prestar este servicio, además las facturas no son emitidas a los clientes en su momento, ya que aducen que el dispositivo se halla en mal estado o no cuentan con el material necesario para la impresión de la factura, por lo cual optan por negociar con el usuario la tarifa, ocasionando que el cliente no perciba una tarifa justa.

En la actualidad existen taxis que ya poseen el sistema GPS, pero estos tienen por finalidad solo calcular la tarifa que debe pagar el cliente, mas no proporciona información sobre la movilidad de las personas, la cual es importante para conocer el estado del sistema de transporte en la ciudad.

1.2 Planteamiento del problema.

En torno a la realidad del cantón San Miguel de Ibarra, se puede evidenciar que el principal problema es la escasa información acerca de la movilidad de las personas dentro de la ciudad, por no poseer ningún sistema que la proporcione, lo que representa una falencia dentro del sistema institucional. De esta manera se imposibilita el desarrollo y ejecución de planes o proyectos que solucionen los problemas en el sistema de transporte.

En la actualidad ningún taxímetro tiene incorporado el sistema GPS que ayude a determinar las coordenadas o posición de inicio y fin de viaje del vehículo, como también

a registrar los viajes realizados por el taxi dentro y fuera de la ciudad, garantizando una tarifa justa al usuario.

Dentro del mismo contexto se puede acotar que este modus operandi no aporta a la seguridad de la movilización pública, debido a que no se cuenta con el sistema de posicionamiento global que permite almacenar y analizar el viaje que se efectúa.

1.3 Formulación del problema.

¿Cómo diseñar, construir e implementar un taxímetro con almacenamiento de viajes mediante GPS en el vehículo Chevrolet Corsa Wind 1.4?

1.4 Delimitación temporal y espacial.

Este proyecto se llevará a cabo desde el mes de Enero hasta el mes de Diciembre en la Universidad Técnica del Norte.

1.5 Objetivos.

1.5.1 Objetivo General.

1. Diseñar, construir e implementar un taxímetro con almacenamiento de viajes mediante GPS en el vehículo Chevrolet Corsa Wind 1.4.

1.5.2 Objetivos Específicos.

1. Diseñar el taxímetro mediante la plataforma de hardware libre “ARDUINO”.
2. Utilizar el sistema GPS para el almacenamiento de viajes realizados.
3. Diseñar el sistema de adquisición de datos del taxímetro mediante el software “LABVIEW”.
4. Realizar pruebas de funcionamiento del dispositivo instalado en el vehículo Chevrolet Corsa Wind 1.4.
5. Desarrollar una herramienta tecnológica que permita a las autoridades locales y nacionales mejorar la movilidad en la ciudad de Ibarra.

1.6 Justificación.

El incorporar el Sistema de Posicionamiento Global (GPS) en el taxímetro, aportará con la información necesaria requerida por parte de las autoridades para planificar nuevas vías o rutas de acceso, así como también organizar horarios y frecuencias de los medios de transporte público.

Así mismo el GPS permitirá controlar el uso del taxímetro por parte de los conductores gracias a la información entregada a las autoridades, a la par el constante análisis de los viajes realizados por el vehículo mejorará la seguridad de los usuarios.

El presente proyecto está enfocado al beneficio de las autoridades que al contar con la información necesaria sobre la movilidad de la ciudadanía podrán brindar mejoras en el sistema de transporte y los ciudadanos, que percibirán una tarifa justa al momento de utilizar el servicio de taxi.

CAPÍTULO II

2. MARCO TEÓRICO.

2.1 Movilidad.

El concepto de movilidad se ha establecido para referirse al conjunto de desplazamientos de personas o de mercancías que se realizan dentro de un entorno físico. “La movilidad como una estrategia intenta mejorar las condiciones de vida, por medio de la distribución igualitaria del espacio público entre las formas de transporte motorizadas y las que no lo son”. (Gordón Salvatierra, 2012)

En la actual sociedad, la movilidad de las personas ha obtenido gran importancia respecto a la de los anteriores periodos de la ciudad contemporánea. Muestra de ello es la importancia que se le ha proporcionado a la movilidad, no solo en el aspecto urbanístico y medioambiental, sino también en los planes de infraestructura de transporte, incluso aunque estos se hayan expresado con otros objetivos más centrados en el desarrollo económico o en la articulación territorial. (Herce Vallejo, 2009)

Esto involucra que el concepto tenga el mismo significado en todos estos discursos, en los que se ha usado con muy diversas intenciones. Pero lo que sí es indiscutible es que la palabra movilidad ha ido desplazando a la palabra transporte, no por motivos circunstanciales, sino porque realmente son conceptos de diferente alcance.

Para poder aclarar mejor la diferencia, se recurre a una acertada reflexión de Peter Hall, que resume las situaciones sociales nuevas que han llevado a este cambio conceptual y que se resume así: “el período de desarrollo de la electrónica y la telemática ha desembocado en la convergencia de complejas redes telemáticas y mecánicas, de modo que se han roto los vínculos de proximidad y han quedado sustituidos por vínculos de conexión a muy diferentes niveles”. (Herce Vallejo, 2009)

Esta sustitución de los niveles de proximidad por relaciones de conexión es consecuente al papel que han alcanzado las redes de infraestructura como reformadoras de territorio, hasta llegar al punto de sustituir a los sistemas de asentamientos urbanos como invariante de la organización espacial.

Las personas se encuentran en un territorio donde se intercalan usos, en un lugar donde se mezclan espacios centrales y periféricos, en el que se realizan actividades diversas y frecuentemente cambiantes, con espacios especializados ya sea en el comercio, servicios, que no solamente atraen usuarios, es decir, generan movilidad, sino que la supervivencia esta sobre su capacidad de incrementar esa atracción. (Herce Vallejo, 2009)

2.1.1 Movilidad urbana.

El fenómeno de la movilidad urbana aparece en la necesidad que tienen las personas para trasladarse de un lugar a otro por varios motivos, siendo los más frecuentes los desplazamientos por trabajo, estudio, aunque también hay otras motivaciones como: compras, servicios, entre otros. Estos se realizan a través de vehículos, transporte público, taxis, que son los que tienen gran trascendencia en la calidad de vida, movilidad y el uso del espacio público.

La movilidad humana se diferencia del transporte humano, debido a que esta no se centraliza al estudio del sistema vial, ni presenta interés sobre los vehículos o los conductores, sino que constituye una relación entre estas características con la acción de movilidad constante, en la que las personas son los actores a través de sus interacciones. (Arrué, 2009)

Los problemas de movilidad que presentan actualmente la mayoría de las ciudades son consecuencia de dos procesos que se han ido solapando en el tiempo. El primero es la utilización del suelo urbano para el transporte, la gran cantidad de espacio urbano que necesita el transporte se adquiere del que necesitan otras actividades, las cuales se ven forzadas a expandirse por el territorio. Con la ampliación de las distancias se necesita cada vez de más desplazamientos motorizados que cada vez reclaman más espacio, generando un círculo vicioso expansivo. El segundo proceso es la especialización del uso del suelo, es decir, a la tendencia de que cada vez más espacio sirve únicamente para un tipo de servicio o actividad, como oficinas, viviendas, centros comerciales, lo que hace que cada vez sea más necesario tener que desplazarse y así menos actividades quedan cerca unas de otras. (Arrué, 2009)

2.1.2 Movilidad humana.

El concepto de movilidad humana se ha determinado para indicar a los desplazamientos que realizan las personas para instalarse en lugares diferentes a aquel en el que vivían ya sea de forma temporal o permanente. Según el primer acuerdo de defensoras/res incluyen a las personas solicitantes de refugio, víctimas de tráfico de personas, migrantes, así como también a las aisladas, desplazadas internas, emigrantes e inmigrantes, por su situación étnico y de género, y otras.

Esta actividad incluye también a los procesos de cruce de los límites establecidos de forma geográfica o política dentro de un determinado país o también en el exterior, es por esta razón que la movilidad humana no se encuentra dentro de los conceptos habituales de migración y se la encierra junto con otros conceptos nombrándola como un derecho humano, ya que la naturaleza humana se ha ido concretando a través de la historia mediante procesos de movilidad. (Benavides & Rodas, 2009)

2.2 Sistema de posicionamiento global GPS.

El Sistema de Posicionamiento Global fue ideado en 1960 y presentó la consolidación para otros proyectos de navegación. Empezó su desarrollo bajo los auspicios de U.S. Air Force, pero fue realmente a partir de 1974 cuando diferentes cuerpos militares de los Estados Unidos unificaron esfuerzos y renombraron al proyecto con el nombre de Navstar Global Positioning System. (Carles Olmedillas, 2013)

Al final el nombre GPS permaneció y el costo del desarrollo de este sistema fue de diez billones de dólares siendo plenamente operacional en 1995. Funciona con 24 satélites circulando alrededor de la Tierra cada 12 horas, proporcionando de esta manera información sobre la posición, hora y velocidad en cualquier lugar del mundo. En la actualidad este sistema es controlado por el Departamento de Defensa de los Estados Unidos.

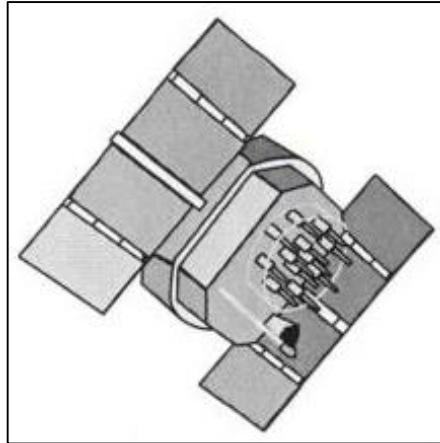


Figura 1. Satélite Navstar GPS.
Fuente: (Carles Olmedillas, 2013)

El sistema GPS está compuesto de tres elementos denominados segmentos, que son:

- Primero. Segmento espacial.
- Segundo. Segmento de control.
- Tercero. Segmento de usuario.

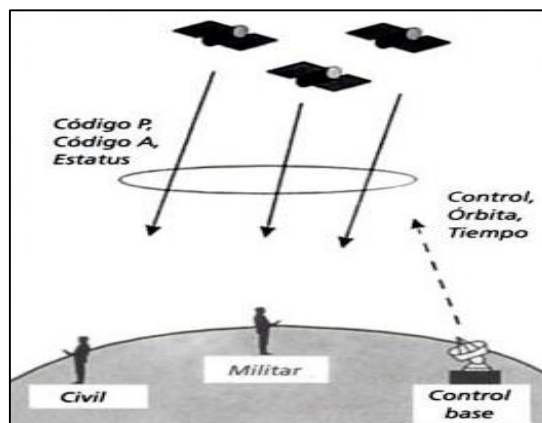


Figura 2. Las tres partes del GPS.
Fuente: (Carles Olmedillas, 2013)

2.2.1 Segmento Espacial.

El segmento espacial está conformado por un conjunto de 24 satélites denominados SV (Space Vehicle). Estos circulan alrededor de la Tierra a 20200 km de altitud formando seis órbitas diferentes con cuatro satélites en cada una.

Cada una de las órbitas tiene una inclinación de 55 grados con respecto al ecuador y separadas 60 grados unas de otras. Cada satélite efectúa una vuelta completa alrededor de la Tierra cada 12 horas, realizando el mismo recorrido todos los días. Al final de 24 horas

menos cuatro minutos debido al desplazamiento de la Tierra alrededor del sol, cada uno de ellos se encuentra en el mismo lugar y la misma configuración respecto a los demás.

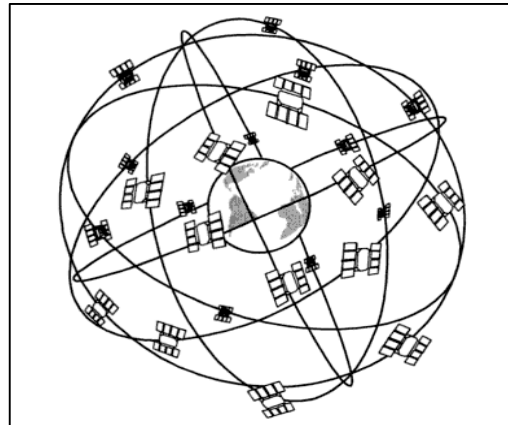


Figura 3. Constelación de satélites.
Fuente: (Carles Olmedillas, 2013)

2.2.2 Segmento de control.

Este segmento es el enlace para el control de los satélites, para rastrearlos y suministrar de esta manera una órbita correcta en tiempo exacto. Las estaciones de monitoreo reciben señales de los satélites continuamente para obtener información y establecer de forma exacta la órbita de los mismos.

Los datos obtenidos por estas estaciones son enviadas a una estación de Control Master, donde se procesa toda esta información para calcular los ajustes de órbita, estado de los relojes, información a transmitir a los satélites utilizando las antenas de la Tierra. Además a través de esta estación se puede enviar correcciones de reloj, comandos de telemetría y otros mensajes.

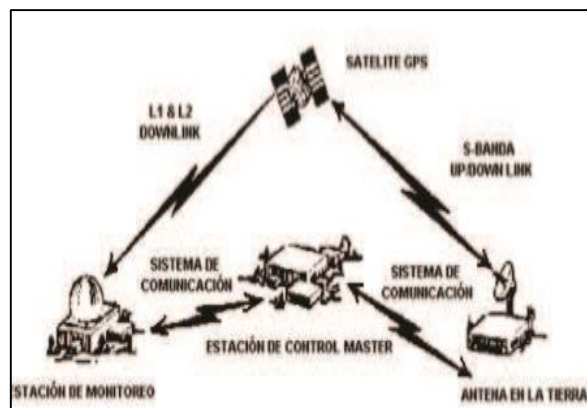


Figura 4. Segmento de control.
Fuente: (Carles Olmedillas, 2013)

2.2.3 Segmento de usuario.

En este segmento se encuentran los receptores GPS civiles y militares diseñados para recibir información emitida por el conjunto de satélites. Estos receptores al recibir información de los satélites brindan las coordenadas exactas, así como también la hora, velocidad en el lugar que se encuentra.

Las ventajas de utilizar estos receptores son las siguientes:

- Sirve de guía para llegar al destino que se elija en un mapa.
- Calcula la posición actual.
- Almacena la posición en una memoria para poder utilizarla en fechas posteriores.



Figura 5. Receptor GPS actual.
Fuente: (Carles Olmedillas, 2013)

2.2.4 Funcionamiento.

El principio utilizado por el GPS para determinar la situación se basa en la medición de la distancia entre el receptor GPS y varios satélites. (Carles Olmedillas, 2013)

Mediante el receptor se logra conocer la posición de cada satélite en el espacio con gran precisión. Cada uno de los satélites transmite constantemente su posición exacta con respecto a la Tierra como también indica la hora exacta de la transmisión del mensaje. Con esta información se calcula el tiempo requerido para que las señales lleguen al receptor y se establezca la distancia al satélite.

Gracias a la distancia y posición del satélite se puede trazar un círculo imaginario en la superficie de la Tierra, en cuyo interior se halla el receptor GPS, es así que con la intersección de varios círculos se puede conocer la posición exacta del receptor.

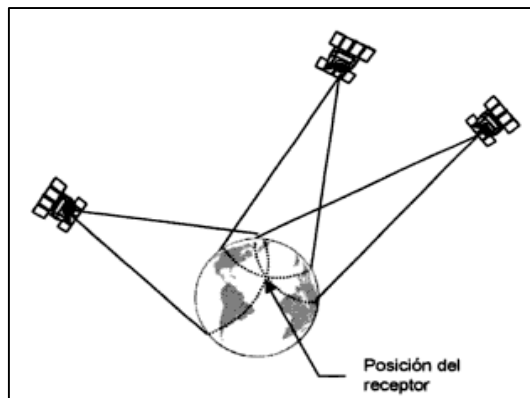


Figura 6. Principio de la situación por satélite.
Fuente: (Carles Olmedillas, 2013)

2.2.5 Aplicaciones del GPS.

El sistema de Posicionamiento Global puede ser utilizado en varias aplicaciones, entre las cuales se tiene:

- En el servicio de transporte para monitorear la flota y las entregas realizadas.
- Se instalan a los buques para su navegación, para de esta manera registrar y controlar los desplazamientos de las embarcaciones.
- Los pilotos utilizan este tipo de sistema para la navegación, fumigación, topografía.
- Se instalan en los vehículos para conocer la ubicación.

2.3 Taxímetro.

Un taxímetro es un instrumento de medida que teniendo en cuenta las tarifas para las que está regulado, se encarga de calcular e indicar la cantidad de dinero que deberá abonar el usuario por el servicio del taxi, esto se determina en función de la distancia recorrida y del tiempo de espera transcurrido desde el inicio de funcionamiento del dispositivo.

El primer vehículo equipado con este dispositivo fue el Daimler Victoria construido por Daimler en 1897. En la antigua Roma, existían taxímetros que operaban a través de

un mecanismo solidario a un eje de una carreta que liberaba pequeñas bolas. Al final del viaje, el pasajero debía pagar en función de las bolas liberadas. En ciertos lugares se utiliza un cartel luminoso para indicar si el taxi está en estado ocupado o libre, como se muestra en la figura 7.



Figura 7. Cartel de taxi libre.
Fuente: (Industrias Saludes, 2015)

Este dispositivo es de uso obligatorio para los conductores de taxis en el Ecuador, y su incumplimiento está establecido como una contravención leve de tercera clase, la cual según la Ley Orgánica de Transporte Terrestre, Tránsito y Seguridad Vial, artículo 141, literal j), establece que: “Incurren en contravención leve de tercera clase y serán sancionados con multa equivalente al quince por ciento de la remuneración básica unificada del trabajador en general, veinte horas de trabajo comunitario y reducción de 4,5 puntos en su licencia de conducir al conductor de un taxi, que no utilice el taxímetro, altere su funcionamiento o no lo exhiba”.

2.3.1 Características y accesorios.

Las características técnicas con las que cuenta un taxímetro, son las siguientes:

- Doble pantalla, una en la que se indica el valor del servicio y otra para indicar el tipo de tarifa aplicada.
- Botones para seleccionar los estados de libre, ocupado y a pagar.
- Impresora para entregar un recibo al pasajero.
- Prestaciones adicionales como indicaciones de fecha, hora, velocidad, distancia recorrida.



Figura 8. Taxímetro actual.
Fuente: (FM & W, 2011)

2.3.2 Funcionamiento.

El taxímetro es un dispositivo que tiene varias posiciones de funcionamiento, las cuales son accionadas a través de sus determinados controles. En el funcionamiento normal este dispositivo tiene las siguientes etapas de comportamiento:

a) Libre.- Esa posición es adoptada por el dispositivo cuando el taxi está vacío. En esta se dispone de varias tarifas las cuales pueden ser cambiadas manualmente por el conductor.

b) Ocupado.- Se entra en esta etapa cuando el taxi comienza el viaje, además se muestra la tarifa actual, así también se puede visualizar información extra como la hora actual, velocidad.

c) A pagar.- Esta etapa empieza al finalizar el viaje. En esta se indica el valor que tiene que cancelar el pasajero, así como también se imprime el recibo en el que se visualiza información como el valor por la distancia recorrida y por el tiempo de espera transcurrido durante el viaje.

En la figura 9 se puede observar un esquema de las posiciones de funcionamiento del taxímetro.

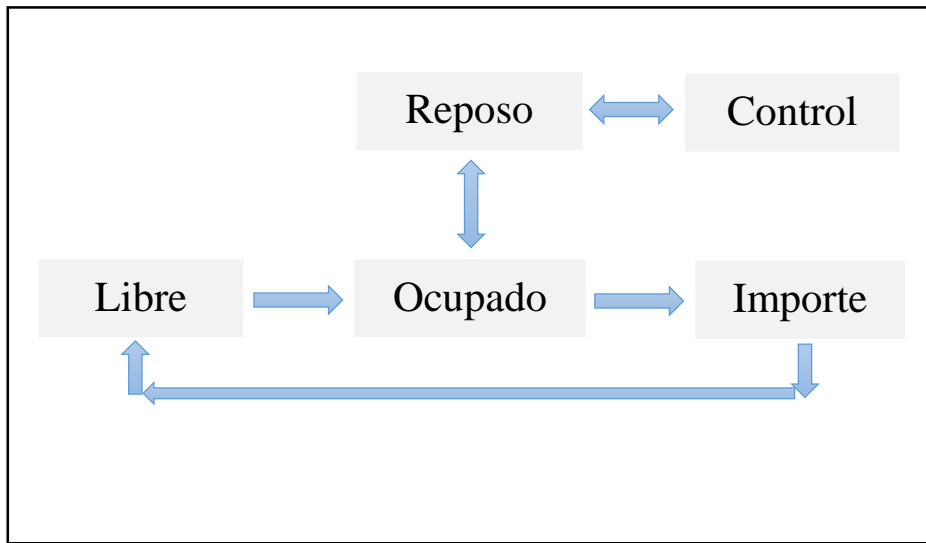


Figura 9. Posiciones de funcionamiento del taxímetro.
Fuente: (INEN, 2013)

2.3.3 Tarifas.

Para tener un buen funcionamiento del taxímetro se debe fijar las tarifas vigentes en las diferentes modalidades, las cuales son establecidas por la Comisión Nacional de Tránsito, Transporte Terrestre y Seguridad Vial. Estas son detalladas en la tabla 1.

Tabla 1. Cuadro tarifario de taxis.

Cuadro tarifario de taxis		
	Diurno (\$)	Nocturno (\$)
Arranque	0.40	0.44
Km. Recorrido	0.40	0.40
Minuto de espera	0.08	0.09
Carrera mínima	1.25	1.50
El horario nocturno comprende desde las 19h00 hasta las 06h00		

Fuente: (Agencia Nacional de Tránsito, 2015)

2.4 Sistemas embebidos.

“Se conoce como sistema embebido a un circuito electrónico computarizado que está diseñado para cumplir una labor específica en un producto”. (Galeano Atehortúa, 2009)

La inteligencia artificial, secuencias y algoritmos de un sistema embebido, se encuentran en la memoria de una pequeña computadora llamada microcontrolador. A diferencia de las computadoras que tiene objetivos generales y están construidos para efectuar gran cantidad de necesidades, los sistemas embebidos se encuentran diseñados para desarrollar objetivos específicos. Por ejemplo, pueden ser dispositivos como un taxímetro, sistemas para el control de acceso, sistemas electrónicos de control del vehículo, microondas, cámaras fotográficas, celulares, reproductor mp3, y varias aplicaciones más.

Básicamente los sistemas embebidos son programados utilizando el lenguaje del microcontrolador interno del mismo, o así también, a través de programas definidos, en los que los lenguajes de programación utilizados pueden ser C o C++. En determinados procesos, en los que las respuestas no tienen al tiempo como elemento determinante, también se puede recurrir a lenguajes de programación interpretados en JAVA. (Cayssials, 2014)



Figura 10. Sistemas embebidos en todas las áreas.
Fuente: (Galeano Atehortúa, 2009)

La cantidad de aplicaciones y de ambientes soportados por los sistemas embebidos aumenta cada vez más, y una de las principales razones fue la introducción de los

procesadores de una sola pastilla, denominada microcontrolador, a través de los cuales gran parte de la electrónica está incorporada, permitiendo reducciones de tamaño, consumo de energía y facilidad de producción.

2.4.1 Componentes de un sistema embebido.

Los elementos importantes que tiene un sistema embebido son los siguientes:

- Microprocesador o microcontrolador, que es la CPU o unidad que aporta la capacidad de control al sistema.
- Comunicación, que es de gran importancia en los sistemas embebidos para la comunicación mediante interfaces estándar. La comunicación puede ser del tipo RS-232, CAN, USB, GPRS.
- Subsistema de presentación, normalmente se utilizan pantallas gráficas, LCD, táctil.
- Actuadores, elementos electrónicos para controlar el sistema, estos pueden ser motores eléctricos, relés.
- Módulo de entradas/salidas analógicas y digitales, son empleadas para digitalizar señales de sensores, para activar diodos.
- Módulo de alimentación de energía, que se encarga de brindar el voltaje y la intensidad que requieren cada uno de los elementos del circuito.
- Conversores de corriente, filtros, circuitos integrados.

2.4.2 Microcontroladores y sistemas embebidos.

Un sistema embebido está basado en un circuito que contiene un microcontrolador en el que su hardware y software se encuentran desarrollados para cumplir con un trabajo específico. El término embebido hace referencia al hecho de que un microcontrolador está instalado dentro de un sistema mayor.

Un microcontrolador es un circuito integrado o chip, es decir, es un dispositivo electrónico que en un solo encapsulado integra un gran número de elementos que poseen las características de ser programables. Este dispositivo es capaz de ejecutar de forma autónoma una serie de instrucciones definidas previamente por el programador. (Artero, 2013)

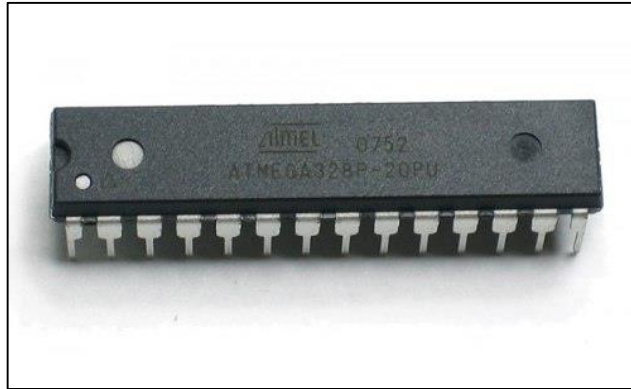


Figura 11. Microcontrolador Atmega328.
Fuente: (Valdés Pérez & Pallàs Areny, 2007)

Un microcontrolador posee en su interior tres elementos básicos: unidad central de proceso, memorias, pastillas de entrada y salida.

Unidad Central de Proceso.- Este elemento es el encargado de ejecutar cada instrucción y de controlar que se la realice correctamente. Normalmente las instrucciones utilizan los datos disponibles previamente, conocidos como datos de entrada, y generan otros datos diferentes como resultado, conocidos como datos de salida, que pueden ser utilizados por la siguiente instrucción.

Diferentes tipos de memorias.- Son las encargadas de almacenar las instrucciones, así como también los diferentes datos que estas necesitan. De esta forma posibilitan que esta información de datos e instrucciones estén siempre disponibles, para que de esta manera el CPU pueda trabajar y acceder con ella en cualquier momento. Generalmente encontramos dos tipos de memoria: las persistentes que se encargan de almacenar información permanentemente incluso tras cortes de alimentación, y las volátiles cuyo contenido se pierde al recibir un corte de alimentación.

Pastillas de entrada y salida.- Son encargadas de comunicar el microcontrolador con el exterior. En las pastillas de entrada del microcontrolador se podrá conectar sensores para que este elemento pueda recibir datos provenientes de su entorno; en las pastillas de salida se podrá conectar actuadores para que el microcontrolador pueda enviar datos u órdenes y así interactuar con el medio físico. Muchas pastillas de la gran mayoría de microcontroladores no son exclusiva para entrada o salida de datos, sino que pueden ser utilizadas para los dos propósitos.

Es decir, un microcontrolador es un computador completo con prestaciones limitadas en un solo chip, que está especializado para ejecutar constantemente un conjunto de instrucciones predefinidas. Las instrucciones irán teniendo en cuenta los datos obtenidos y enviados por las pastillas de entrada o salida y reaccionarán en consecuencia. Los microcontroladores son diseñados para reducir costos económicos y el consumo de energía en un sistema particular, es por esta razón que la unidad de procesamiento, la memoria y los periféricos dependerán de la aplicación.

2.5 Arduino.

La plataforma de instrumentación Arduino es un sistema que permite diseñar y programar un dispositivo electrónico de manera óptima al sistema que se quiere automatizar o controlar, proporcionando alta eficiencia, costo reducido y programación amigable. Por medio de este dispositivo se puede crear elementos interactivos, obtención de datos, envío de comandos. Una mejora de los proyectos Arduino es que el programa principal se ejecuta de forma autónoma en la tarjeta electrónica y al mismo tiempo conserva comunicación con la computadora para presentar la información. (Artero, 2013)

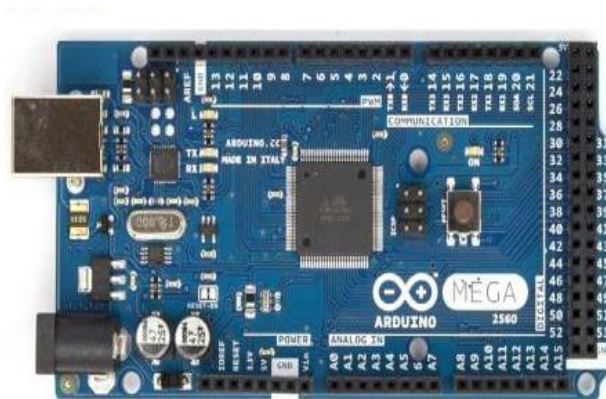


Figura 12. Arduino Mega 2560.
Fuente: (Perea, 2015)

Básicamente Arduino es tres cosas: una placa de hardware libre, un software, un lenguaje de programación libre.

2.5.1 Placa hardware libre.

Incorpora un microcontrolador reprogramable con una serie de pines hembra que están unidos internamente a las pastillas de entrada o salida que permiten conectar de forma sencilla y cómoda diferentes sensores y actuadores.

Al hablar de placa hardware, se refiere en concreto a una placa de circuito impreso, que son superficies fabricadas de un material no conductor, sobre las cuales aparecen pistas laminadas de un material conductor que normalmente es el cobre. Estas placas se utilizan para conectar eléctricamente componentes electrónicos a través de los caminos conductores, además, son la forma más estable y compacta de construir circuitos electrónicos, pero una vez fabricadas su diseño, son muy difíciles de modificarlas.

Cuando se utiliza una placa Arduino, se debe especificar el modelo, ya que existen varias placas de Arduino oficiales y cada una tiene características diferentes, como el tamaño físico, número de pines, modelo del microcontrolador y como consecuencia la cantidad de memoria utilizable. De todas formas aunque sean modelos específicos, los microcontroladores incorporados en las diferentes placas pertenecen a la misma familia tecnológica, por lo tanto su funcionamiento es muy parecido entre sí. En concreto todos los microcontroladores son del tipo AVR, una arquitectura de los microcontroladores desarrollados y fabricados por la marca Atmel. El diseño hardware de la placa está basado originalmente en el de otra placa de hardware libre preexistente, conocida con el nombre de placa Wiring.

2.5.2 Software Arduino.

En palabras más concretas, un entorno de desarrollo libre, gratis y multiplataforma ya que funciona con Linux, MacOS y Windows que se debe instalar en el ordenador y que permite escribir, verificar y cargar en la memoria del microcontrolador las instrucciones que queremos que este empiece a ejecutar. La manera de conectar la placa al computador es mediante un cable USB lo que permite enviar y grabar el conjunto de instrucciones.

Los proyectos en Arduino pueden ser o no autónomos; en el caso de ser autónomos, una vez programado el microcontrolador, la placa no necesita estar conectada al computador y puede funcionar autónomamente si se dispone de alguna fuente de alimentación. En el caso de que la placa Arduino no sea autónoma debe estar conectada

de forma permanente al computador ejecutando un software específico que permita la comunicación y el intercambio de datos entre este y la placa. (Artero, 2013)

2.5.3 Lenguaje de programación libre de Arduino.

Se entiende por lenguaje de programación a cualquier idioma artificial diseñado para expresar instrucciones que pueden ser realizadas por alguna máquina o elemento electrónico. Dentro del lenguaje de Arduino se encuentran elementos parecidos a otros lenguajes de programación existentes, así como también diferentes comandos que permiten especificar de forma coherente y sin errores el conjunto de instrucciones exactas que deseamos programar en el microcontrolador de la placa. Estos comandos son escritos a través del entorno de desarrollo Arduino.

Tanto el entorno de desarrollo como el lenguaje de programación están inspirados en otro entorno y lenguaje libre conocido con el nombre de Processing, que fue desarrollado inicialmente por Ben Fry y Casey Reas. Que el software Arduino sea parecido a Processing no es una cualidad, debido a que está diseñado para facilitar la generación de imágenes en tiempo real, animaciones, interacciones visuales. No obstante, es necesario aclarar que el lenguaje Processing está construido con código escrito en lenguaje Java, en cambio el lenguaje Arduino se basa en código C/C++.

2.5.4 Programación de la placa Arduino.

Para la programación de la placa Arduino se utiliza el software de Arduino IDE que tiene un lenguaje de programación amigable, además, permite cargar las instrucciones a la placa de manera rápida con comunicación USB.

El programa tiene dos partes fundamentales que son la función setup y la función loop. La setup se ejecuta una sola vez, mientras que la función loop se ejecuta un número infinito de veces. Además para la programación se utilizan comandos, variables para el tipo de datos, operadores aritméticos, operadores lógicos, comparadores, estructuras de control y librerías.

Comandos.- Los comandos más utilizados son los que se indican en la tabla 2.

Tabla 2. Comandos programa Arduino.

Comando	Descripción
pinMode (pin, mode);	Se establece el comportamiento de los pines digitales, ya sea de entrada (INPUT), o de salida (OUPUT).
digitalWrite (pin, estado);	Se establece el estado de un pin determinado.
Serial.begin (baudRate);	Inicia la comunicación serial y hace posible la comunicación entre el microcontrolador y el computador.
Serial.println (String);	Con esta se puede enviar textos por medio del puerto serial. En String se escribe el mensaje que se quiere enviar.
delay (tiempo);	Provoca un retraso de la ejecución del código en milisegundos.
millis ();	Devuelve la cantidad de milisegundos que han pasado desde que se inició el programa.
digitalRead (pin);	Lee el estado de un pin digital.
analogWrite (pin, nivel);	Permite usar PWM, es decir modulación por ancho de pulsos. Establece un nivel análogo de voltaje.
analogRead (pin);	Se utiliza con los pines análogos entre A0 y A7. Lee el nivel de voltaje en el pin seleccionado.

Fuente: (Artero, 2013)

Variables para el uso de datos.- Las variables para el uso de datos más utilizados son los que se indican en la tabla 3.

Tabla 3. Variables programa Arduino.

Dato	Descripción
boolean	Dato booleano y puede ser TRUE o FALSE.
Char	Dato tipo carácter.
Int	Almacena números enteros de 2 bits que están entre -32768 hasta 32767
long	Son variables de tamaño extendido que almacenan números desde -2147483648 hasta 2147483648

Continúa →

float	Guarda números decimales que van desde -3.4028235E+38 hasta 3.4028235E+38.
String	Se representan como arrays de carácter.
Array	Un array es una colección de variables que son accedidas mediante un número de índice.
Void	Se usa para la declaración de funciones.

Fuente: (Artero, 2013)

Operadores aritméticos y lógicos.- Los operadores aritméticos y lógicos más utilizados se detallan en la tabla 4.

Tabla 4. Operadores Programa Arduino.

Operador	Función
+	Suma
-	Resta
*	Multiplicación
/	División parte entera
^	Potenciación
%	División parte decimal
&&	Y
!	Negación

Fuente: (Artero, 2013)

Comparadores.- Los comparadores más importantes se presentan en la siguiente tabla:

Tabla 5. Comparadores Programa Arduino.

Signo	Función
==	Igual a
!=	Diferente de
<	Menor que
>	Mayor que
<=	Menor o igual a
>=	Mayor o igual a

Fuente: (Artero, 2013)

Estructuras de control.- Las estructuras de control más utilizadas son las que se indican en la tabla 6.

Tabla 6. Estructuras utilizadas en el programa Arduino.

Estructura	Función
If (condición) { Instrucciones }	Estructura comparativa simple. Si se cumple la condición, se ejecutarán las instrucciones que se coloquen entre las llaves.
If (condición) { Instrucciones A } else { Instrucciones B }	Estructura comparativa doble. Si se cumple la primera condición, entonces ejecutará el bloque de instrucciones A. Si la condición inicial no se cumple, entonces se ejecutará el bloque de instrucciones B.
For (int i=número 1;i<=número 2; i ++)	Se utiliza una variable contadora. Se repetirá un mismo proceso un número finito de veces.
While (condición){ Instrucciones }	Estructura de condiciones. Mientras se esté cumpliendo una condición se ejecutarán las instrucciones entre las llaves un número infinito de veces.

Fuente: (Artero, 2013)

Librerías.- Para la programación de este dispositivo se utilizan las librerías que se indican en la siguiente tabla:

Tabla 7. Librerías para los diferentes elementos.

Librería	Función
TimerOne.h	Librería del timer
TinyGPS++.h	Librería del GPS
SPI.h	Librería para la comunicación con el módulo SD
SD.h	Librería del módulo SD

Continúa
→

Adafruit_Thermal.h	Librería de la impresora térmica
UTouch.h	Librería del control táctil
UTFT.h	Librería de la TFT 3.2”
UTFT_tinyFAT.h	Librería comunicación entre la TFT y SD
tinyFAT.h	Librería para lectura de la SD

Fuente: (Artero, 2013)

2.5.5 Ventajas de utilizar Arduino.

Existen muchas placas de fabricantes diferentes, pero aunque incorporan modelos de microcontroladores distintos, son comparables y ofrecen un funcionamiento similar a las de Arduino. Todas ellas de igual manera vienen de un entorno de desarrollo cómodo y con un lenguaje de programación completo y sencillo. No obstante, la plataforma Arduino ofrece las siguientes ventajas:

- Arduino es libre y extensible. Se puede mejorar el diseño del hardware de las placas, así como también el entorno de desarrollo software y el lenguaje de programación.
- Arduino tiene una gran comunidad. Las personas que lo utilizan comparten continuamente sus ideas y enriquecen la documentación.
- Su entorno de programación es multiplataforma. Se puede ejecutar e instalar en sistemas Windows, Linux y Mac, lo que no ocurre con el software de otras placas.
- Su entorno y lenguaje de programación son claros y simples. Fáciles de utilizar y aprender, a la vez son completos y flexibles para que los programadores avanzados puedan aprovechar de mejor manera todas las posibilidades del hardware. También están documentados con ejemplos detallados con proyectos publicados en diferentes formatos.
- Las placas Arduino son versátiles y reutilizables. Versátiles porque proveen varios tipos de entrada y salida de datos, permitiendo obtener información de sensores enviar señales de múltiples formas a los actuadores; y reutilizables porque se puede utilizar la misma placa para diferentes proyectos.

2.6 LabVIEW.

“LabVIEW es el acrónimo de Laboratory Virtual Instrument Engineering Workbench. Es un lenguaje y a la vez un entorno de programación gráfica en el que se pueden crear aplicaciones de una forma rápida y sencilla” (Lajara y José Rafael, 2008, p. 3).

En 1976, Austin Texas nació la organización National Instrument quien empezó el desarrollo del programa LabVIEW en abril de 1983, logrando sacar al mercado la primera versión en 1986. Hasta la actualidad han ido evolucionando las versiones de este programa, las cuales se observan en la tabla 8.

Tabla 8. Evolución de las diferentes versiones de LabVIEW.

Fecha	Hito
Abril de 1983	Comienza el desarrollo de LabVIEW
Octubre de 1986	LabVIEW 1.0 para Macintosh
Enero de 1990	LabVIEW 2.0
Septiembre de 1992	LabVIEW para Windows
Octubre de 1992	LabVIEW para Sun
Octubre de 1993	LabVIEW 3.0 multiplataforma
Abril de 1994	LabVIEW para Windows NT
Octubre de 1994	LabVIEW para Power Macintosh
Octubre de 1995	LabVIEW para Windows 95
Mayo de 1997	LabVIEW 4.0
Marzo de 1998	LabVIEW 5.0
Febrero de 1999	LabVIEW 5.1 LV para Linux.
Agosto de 2000	LabVIEW 6i
Enero de 2002	LabVIEW 6.1
Mayo de 2003	LabVIEW 7 Express

Continúa



Mayo de 2004	LabVIEW 7.1
Mayo 2005	LabVIEW DSP
Junio de 2005	LabVIEW Embedded
Octubre de 2005	LabVIEW 8
Agosto de 2006	LabVIEW 8.20

Fuente: (Reyes Cortés, Cid Monjaraz, & Vargas Soto, 2013)

Esta plataforma y entorno de desarrollo es muy recomendado para sistemas de hardware y software de pruebas de control y diseño, simulado o real, pues acelera la productividad. Este utiliza un lenguaje de programación denominado G, simbolizando que es un lenguaje gráfico.

Los programas desarrollados se llaman Instrumentos Virtuales debido a que su origen provenía del control de instrumentos, aunque en la actualidad se ha expandido no solo al control electrónico, sino también a la programación embebida, matemáticas, comunicaciones. Entre sus objetivos están el de disminuir el tiempo de desarrollo de aplicaciones y el de permitir la entrada a la informática a los profesionales de cualquier campo.

LabVIEW 2013 es la penúltima versión, con la demostración de poder utilizarse simultáneamente para el diseño del firmware de un instrumento RF. Luego surgió la versión 2014 y actualmente existe la versión 2015 disponible para estudiantes y profesional.

2.6.1 Entorno.

El programa LabVIEW es un instrumento de programación gráfica que inicialmente fue orientado a las aplicaciones de control de elementos electrónicos de instrumentación virtual. Por tal razón todos los programas que son realizados en esta herramienta se guardan en tipos de ficheros VI, cuyo significado es instrumento virtual (Virtual Instrument).

En el programa se dispone de dos ventanas principales que toman el nombre de Panel Frontal y Diagrama de Bloques.

Panel Frontal.- Es la interfaz con el usuario y se utiliza para interactuar con él, al ejecutar el programa. En esta parte los usuarios pueden visualizar los datos del programa en tiempo real.

En el panel frontal, se dispone de varios tipos de controles, donde cada uno tiene asignado un terminal en el diagrama de bloques, es decir, se puede diseñar un programa en el panel frontal utilizando controles e indicadores, que serán los elementos de entrada y salida que interactuarán con la terminal del Vi.

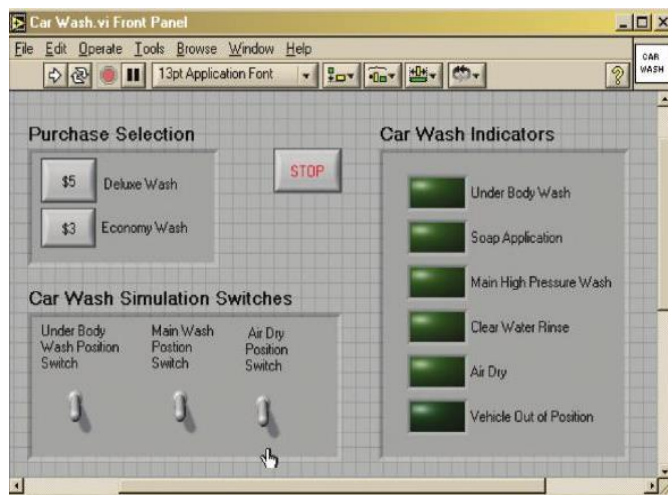


Figura 13. Ejemplo de panel frontal.
Fuente: (Bress, 2013)

Diagrama de Bloques.- Es la parte en la que se efectúa la programación, donde se define la funcionalidad y se colocan iconos que realizan una determinada función interconectados entre sí, ésta ventana se encuentra de fondo blanco.

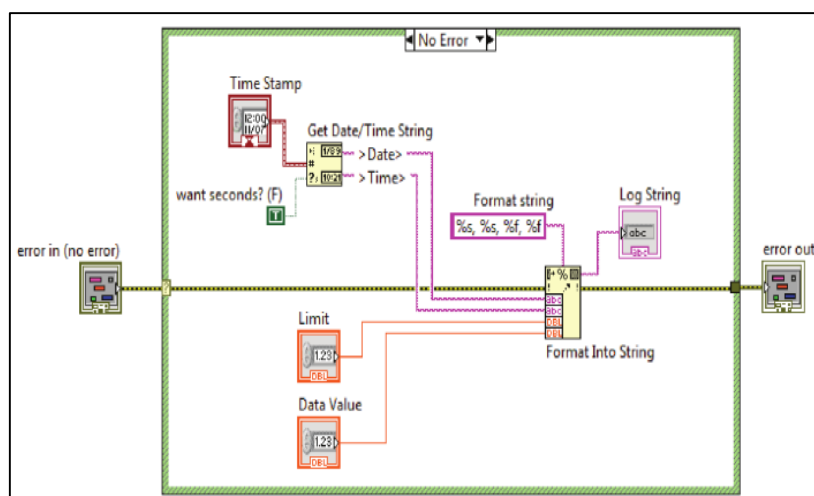


Figura 14. Ejemplo de diagrama de bloques.
Fuente: (Bress, 2013)

El Diagrama de Bloques y el Panel Frontal se encuentran conectados entre sí a través de elementos que se utilizan como entrada y salida de datos conocidos con el nombre de terminales.

2.6.2 Características.

La característica principal de este programa es la facilidad de uso, ya sea para programadores profesionales y para personas que no tienen muchos conocimientos en programación. Los programas en LabView tienen el nombre de instrumentos virtuales (VIs). Para las personas expertas en LabVIEW pueden crear programas con muchos VIs para aplicaciones complejas.

En LabView 7.0 existe un nuevo tipo de subVI conocido como VIs Expreso, que son VIs interactivos con una configuración de caja de diálogo que permite al programador a personalizar la funcionalidad del VI Expreso.

La utilización de este software facilita el manejo de:

a) Interfaces de comunicaciones como:

- Puerto paralelo.
- Puerto serie.
- GPIB.
- VXI.
- PXI.
- TCP/IP, UDP, Data Socket.
- USB.
- Bluetooth.
- OPC.

b) Capacidad de interactuar con otros lenguajes y aplicaciones como:

- DLL: librerías de funciones.
- ActiveX.
- .NET.
- Matlab/Simulink.
- AutoCAD, SolidWorks.

- Multisim.
- Adquisición y tratamiento de imágenes.
- Sincronización de dispositivos.
- Manejo y visualización de graficas con datos dinámicos.
- Trabajo en tiempo real.
- Programación para validación o control.
- Herramientas textuales y gráficas para el tratamiento de señales digitales.

GLOSARIO DE TÉRMINOS

Acrónimo. Suma de los significados de las palabras que lo conforman.

Actuadores. Son dispositivos capaces de transformar la energía neumática, hidráulica o eléctrica, en activación de un proceso, con el objetivo de generar un efecto sobre un proceso automatizado.

Autónomo. No necesita de un computador para funcionar.

Algoritmos. En matemática, ciencias de la computación, lógica, y disciplinas relacionadas, un algoritmo es un conjunto de reglas o instrucciones bien definidas, ordenadas y finitas que permiten realizar alguna actividad a través de pasos sucesivos.

Capacidad eléctrica. Propiedad que tienen los cuerpos para mantener la carga eléctrica.

Circuito integrado. Combinación de diversos parámetros de programación en un estado permanente, es decir que es un sistema que posee funciones ya determinadas y que son incommutables.

Electrónica. Rama de la física y especialización de la ingeniería que estudia y emplea sistemas cuyo funcionamiento se basa en la conducción y el flujo de electrones cargados eléctricamente.

Firmware. Bloque de instrucciones de máquina para propósitos específicos, grabado en un chip, normalmente de lectura/escritura.

Hardware. Es el conjunto de elementos físicos que forman parte de un sistema operativo.

Instrucciones. Acciones que se dan a un ordenador con el objetivo de realizar una determinada operación.

Instrumentación virtual. Se encarga de medir señales, registrar datos y decidir las acciones de control.

Interfaces de comunicaciones. Medios electrónicos en que se comunica o entabla una conexión con otro dispositivo.

Inteligencia artificial. Está diseñado para realizar operaciones que son consideradas propias de la inteligencia humana.

Monitorear. Tener constante información sobre un cuerpo, su posición, velocidad y demás factores que inciden en la información que se requiere.

Movilidad. Desplazamiento de las personas dentro de una sociedad, contemplándose los medios de transporte y a la facilidad de hacerlo.

Plataforma de programación. Sistema que se utiliza como base para hacer funcionar elementos hardware con los que es compatible.

Resistencia eléctrica. Igualdad de oposición que tienen los electrones al moverse a través de un conductor.

Satélites. Dispositivos en órbita alrededor de la Tierra interconectados entre sí, que emiten información sobre las condiciones en el planeta.

Sensor. Dispositivo capaz de detectar señales físicas o químicas, conocidas como variables de instrumentación para transformarlas en variables eléctricas.

Sistema AVL. Sistema de rastreo vehicular automatizado.

Software. Es el conjunto de programas que hacen posible el desarrollo de tareas específicas.

Subsistema. Sistema que se ejecuta sobre un sistema operativo.

Transporte motorizado. Medio de traslado de personas o bienes desde un lugar hasta otro, cuyo medio de propulsión es un motor.

Telemática. Es una materia científica que fue creada por la correlación que existía entre la telecomunicación y la informática.

Tensión eléctrica. Magnitud física que cuantifica la diferencia de potencial eléctrico entre 2 puntos.

Unidad central de proceso. Componente principal de un ordenador, que interpretan las instrucciones almacenadas en los programas.

CAPÍTULO III

3. METODOLOGÍA DE LA INVESTIGACIÓN

3.1 Tipo de investigación.

El presente proyecto es una investigación tecnológica y bibliográfica.

3.1.1 Investigación bibliográfica.

La investigación bibliográfica está inmersa en este proyecto debido a que se recurre a manuales para conocer el correcto uso de la plataforma Arduino y el software LabVIEW, así mismo se emplean libros que proporcionen información sobre las características y forma de conexión del taxímetro y GPS, además, el internet es una ayuda en temas de conceptos y definiciones sobre el proyecto.

3.1.2 Investigación tecnológica.

Este proyecto es de investigación tecnológica debido a que se utiliza la programación y electrónica para crear un nuevo sistema de taxímetro con GPS, que permita generar, transmitir, receptar y almacenar información.

3.2 Métodos.

Para este tipo de investigación, se cuenta con los métodos prácticos de: programación, diseño, mediciones, optimización y método analítico sintético.

Programación.- Se programa la plataforma de hardware libre de Arduino y GPS para su correcto funcionamiento tomando en cuenta la normativa vigente de las tarifas del taxímetro.

Diseño.- Diseño y construcción del taxímetro con GPS para implementarlo en el vehículo Chevrolet Corsa Wind 1.4 mediante la utilización de elementos electrónicos.

Mediciones.- Obtención de datos del dispositivo empleado a manera de simulación para verificar su utilidad.

Optimización.- En base a las diferentes pruebas realizadas se optimizará el taxímetro con el fin de garantizar que abarcará la solución a la necesidad existente.

Método analítico sintético.- Se aplica este método con el cual se obtiene información de fuentes bibliográficas como son: libros, internet, manuales; los cuales servirán de guía para la elaboración del proyecto.

3.3 Técnicas e instrumentos.

Las técnicas utilizadas en este proyecto son:

Adaptación.- Adaptación del GPS en el taxímetro construido para que trabajen conjuntamente y así obtener los datos requeridos.

Análisis de datos.- Obtención y análisis de los datos almacenados por el dispositivo empleado para así certificar su funcionamiento y utilidad.

Pruebas de funcionamiento.- Se realiza pruebas de funcionamiento del dispositivo construido para diagnosticar su funcionamiento y corregir fallas de ser necesario.

CAPÍTULO IV

4. DISEÑO Y CONSTRUCCIÓN DEL TAXÍMETRO.

4.1 Diseño de hardware.

4.1.1 Diagrama de bloques.

Para realizar el diseño de hardware se elige los elementos que cumplan con las características necesarias para el buen funcionamiento del dispositivo, de manera que, con sus respectivas conexiones existan interacciones entre ellas y el microcontrolador.

Una vez conocidos los parámetros de funcionamiento del taxímetro, así como el reglamento de las tarifas en los diferentes modos de operación y las características de sus principales elementos, se diseña el taxímetro con almacenamiento de viajes mediante GPS con el objetivo de adquirir datos y realizar un cálculo estadístico que proporcione la información sobre la movilidad urbana en la ciudad de Ibarra. En la figura 15 se presenta el diagrama de bloques del diseño del dispositivo.

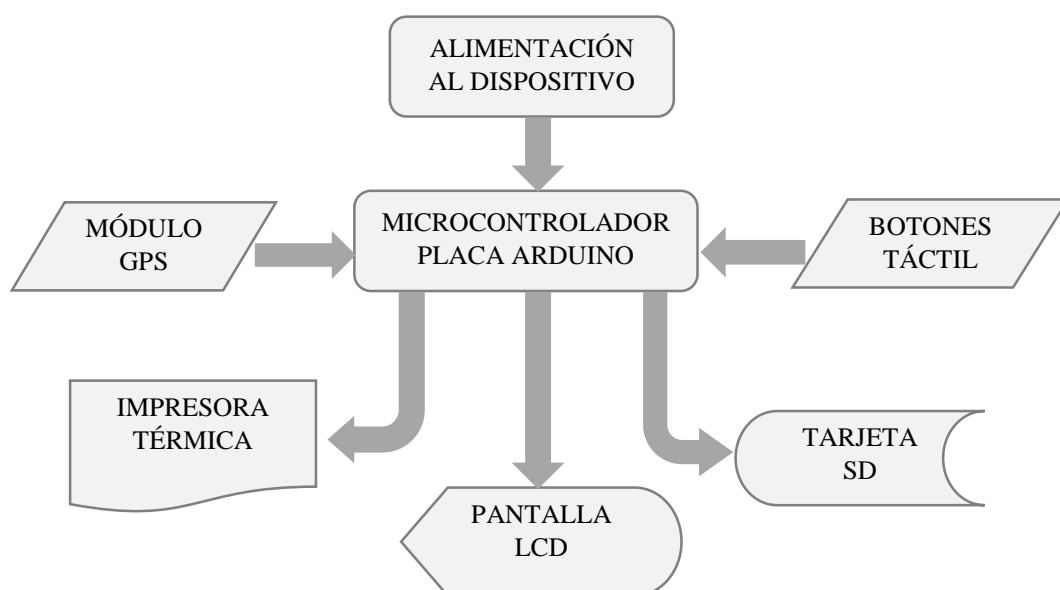


Figura 15. Diagrama de bloques.
Fuente: (Morales)

Para el diseño y construcción del presente proyecto es necesario:

- a) Una placa de Arduino Mega 2560 que realiza el procesamiento de datos del módulo GPS y el almacenamiento de los mismos en la unidad SD, así como también el cálculo del costo a cancelar por el usuario tomando en cuenta todos los parámetros de distancia, velocidad, tiempo de espera.
- b) Un módulo GPS para proporcionar información de distancia recorrida, posición, velocidad, hora, fecha.
- c) Un módulo SD para el almacenamiento de datos de los viajes realizados para su posterior análisis.
- d) Una impresora térmica con comunicación serial que proporcione un recibo con detalles del viaje realizado como: tipo de tarifa, lugar de inicio y fin, hora, fecha, subtotal, total.
- e) Una pantalla LCD para indicar de forma clara y en todo momento el estado de funcionamiento del dispositivo.
- f) Una fuente de alimentación que abastece las necesidades de todos los elementos electrónicos a instalarse.

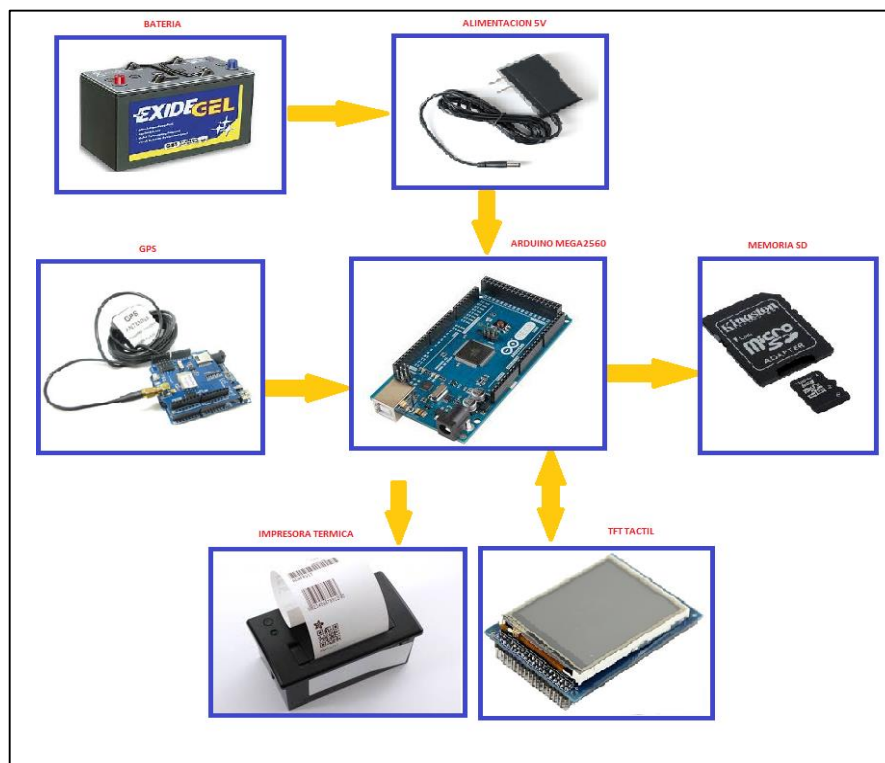


Figura 16. Diagrama de elementos del prototipo.
Fuente: (Morales)

4.1.2 Selección de la placa Arduino.

El dispositivo cuenta con una placa Arduino Mega basada en el Atmega 2560 como elemento principal para el procesamiento de datos y por la cantidad de pines para la instalación de los diferentes elementos electrónicos a utilizar, además, cuenta con 54 pines digitales de entrada o salida, de los cuales 15 de ellos pueden utilizarse como salidas PWM, 16 pines de entrada analógicos, 4 UARTs o puertos series, un oscilador de cristal 16 MHz, conector de alimentación, conexión USB, botón de reinicio.

Datos técnicos.- Los datos técnicos de la placa de Arduino Mega 2560 se detallan en la tabla 9.

Tabla 9. Datos técnicos Arduino Mega 2560.

Características	Descripción
Microcontrolador	Atmega 2560
Tensión de funcionamiento	5V
Voltaje de entrada (recomendado)	De 7voltios a 12voltios
Entrada de voltaje	De 6voltios a 20voltios
Pines digitales	54 (15 pueden ser PWM)
Pines analógicos	16 pines
Corriente CC	20 miliamperios
Corriente CC n pin de 3.3V	50 miliamperios
Memoria disponible	256 KB, donde 8 de ellos son para la gestión del arranque
Memoria SRAM	8 KB utilizados
Memoria EEPROM	4 KB utilizados
Rapidez del reloj	16 MHz
Largo	101.52 mm
Ancho	53,3 mm
Peso	37 g

Fuente: (Arduino, 2015)

Programación.- El Arduino Mega 2560 se programa con el software de Arduino IDE que viene pre programada con el gestor de arranque que le permite cargar a este sin la utilización de un programador de hardware externo, además la placa Arduino se comunica mediante el protocolo STK500.

Alimentación.- Puede ser alimentado mediante la conexión USB o a través de una fuente de alimentación externa. La placa puede funcionar con suministro externo de 7V a 12V, pues, si se lo hace con menos de 7V, el pin de 5V del Arduino puede suministrar menos de cinco voltios y la junta puede volverse inestable, en cambio si la alimentación es de más de 12V el regulador de voltaje puede sobrecalentarse dañando la placa. Los pines de alimentación son como se indican a continuación:

- a) VIN. Se utiliza cuando el voltaje de entrada a la placa es a través de una fuente de alimentación externa.
- b) 5V. Es el pin de salida de 5V regulado a través del regulador de la placa. Esta puede ser alimentada a través de la toma de alimentación de CC (7V-12V), conector USB (5V), o del pin VIN del Arduino (7V-12V).
- c) 3.3V. Proporciona un suministro de 3.3 voltios generados por el regulador de la placa, con una corriente máxima de 50Ma.
- d) GND. Pines de tierra.
- e) IOREF. Este pin de la placa proporciona la referencia de tensión con la que trabaja el microcontrolador. Es un escudo configurado que puede leer el voltaje de instrucción IOREF y seleccione la alimentación adecuada de voltaje en las salidas para trabajar con 5V o 3.3V.

Memoria.- El microcontrolador Atmega 2560 posee 256 KB de memoria flash para almacenar los códigos, de los cuales 8KB son utilizados para el cargador de arranque, 8KB de SRAM y 4KB de EEPROM.

Diagrama de pines de entrada y salida.- Cada uno de los 54 pines digitales de la placa pueden ser utilizados como entrada o salida a través de las funciones pinMode (), digitalWrite () y digitalWrite (). Cada uno de ellos puede recibir o proporcionar 20 mA como funcionamiento recomendado y tiene una resistencia de 20 a 50 k ohmios en configuración pull-up. Se debe evitar valores máximos a 40 mA para conservar el microcontrolador. Algunos pines tienen funciones especializadas como:

- Serial: 0 (RX) y 1 (TX); Serial 1: 19 (RX) y 18 (TX); Serial 2: 17 (RX) y 16 (TX); Serial 3: 15 (RX) y 14 (TX). Se utilizan los pines RX para recibir datos en serie y los TX para transmitir (TTL).
- Interrupciones externas: 2 (interrupción 0); 3 (interrupción 1); 18 (interrupción 5); 19 (interrupción 4); 20 (interrupción 3); 21 (interrupción 2). Estos pines pueden ser configurados para activar interrupciones a nivel bajo, en modo ascendente, descendente o cambio de nivel.
- PWM. De los pines 2 al 13 y 44 al 46, proporcionan salidas PWM de 8 bits con la función analogWrite ().
- SPI. Pines 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Estos soportan la comunicación SPI con la utilización de su respectiva biblioteca.
- LED. Existe un led incorporado al pin 13. Al momento de que el pasador es de bajo nivel, el led está apagado, caso contrario si es de alto nivel, es encendido.

Además tiene 16 entradas analógicas y cada una proporciona 10 bits de resolución, es decir, 1024 valores diferentes. Se mide desde 5V, aunque se puede cambiar al rango superior usando el pin AREF y la función analogReference (). Tenemos dos pines extras que son:

- AREF. Se utiliza para la tensión de referencia para las entradas analógicas.
- Reiniciar. Este pin se utiliza para reiniciar el microcontrolador a través de un botón.

A continuación se muestra el diagrama de pines del microcontrolador Atmega 2560.

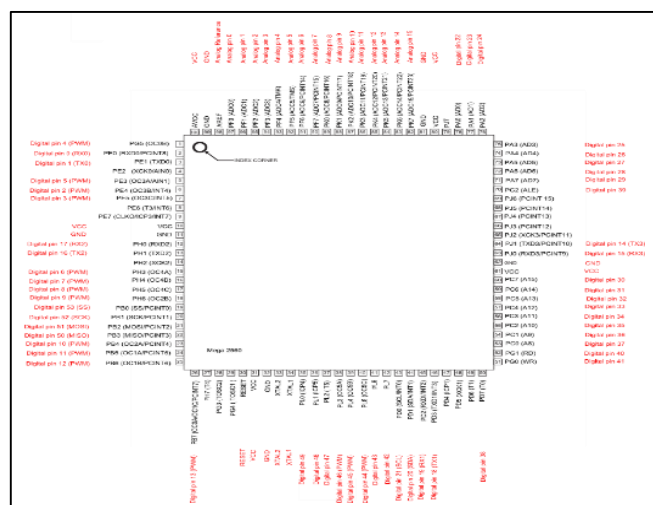


Figura 17. Diagrama microcontrolador Atmega2560.
Fuente: (Arduino, 2015)

Comunicación.- El Arduino Mega 2560 tiene una serie de instalaciones para la comunicación con el computador u otros microcontroladores. Este ofrece cuatro UARTs para TTL de comunicación serie. El software Arduino incluye un monitor serial que permite observar los datos de texto simples para enviarlos hacia y desde la placa.

Los leds RX y TX de la placa parpadean cuando hay transmisión de datos a través del microcontrolador y la conexión USB al computador. También es compatible con la comunicación TWI y SPI.

4.1.3 Selección del módulo GPS.

El módulo GPS utilizado es el shield GPS Arduino puesto que trabaja con hasta veinte satélites a la vez lo que permite obtener gran precisión, además consta de un módulo SD necesario para el almacenamiento de datos.

Este GPS funciona con un nivel de voltaje de 5V o 3.3V compatible con la placa Arduino y está basado en el módulo RoyalTek REB-4216, en el que los pines regulares RX, TX pueden ser conectados a los puertos D0-D7 de la placa Arduino Mega. Además, es diseñado con interfaz SD, permitiendo el almacenamiento de gran cantidad de información.

También brinda tiempo y fecha exacta siendo muy utilizable en aplicaciones como:

- Navegación automotriz.
- Posicionamiento personal.
- Navegación marítima.
- Gestión de flotas.



Figura 18. Módulo GPS NEO-6M-0-001.
Fuente: (ITead Studio, 2011)

Características.- El módulo GPS tiene las siguientes características:

- a) Interfaz de micro SD.
- b) Antena activa con alta sensibilidad de recepción.
- c) Interfaz UART.
- d) Rápido para fijar tiempo bajo nivel de señal.
- e) Temperatura de funcionamiento de $-40^{\circ}\text{C} \sim +85^{\circ}\text{C}$.
- f) Tamaño estándar de escudo Arduino.
- g) Botón de reinicio de Arduino.
- h) ON/OFF que controla la alimentación del módulo GPS.

Especificaciones y características eléctricas.- Las especificaciones de este módulo GPS se presentan en la siguiente tabla:

Tabla 10. Especificaciones GPS.

Tamaño de PCB	55.88mm x 54.1mm x 1.6mm
Indicadores	PWR
Fuente de alimentación	Compatible con Arduino
Protocolo de comunicación	UART
RoSH	Sí

Fuente: (ITEad Studio, 2011)

Y las características eléctricas de funcionamiento de este elemento electrónico son las siguientes:

Tabla 11. Parámetros de funcionamiento GPS.

Parámetro	Min.	Típico	Max.
Tensión de alimentación	4.5	5	5.5
Voltaje de entrada VH	3		5.5
Voltaje de entrada VL	-0,3	0	0.5
Velocidad de transmisión		9600	

Fuente: (ITEad Studio, 2011)

Diagrama.- El diagrama de la placa del módulo receptor así como la de distribución de pines se puede observar en la figura 19.

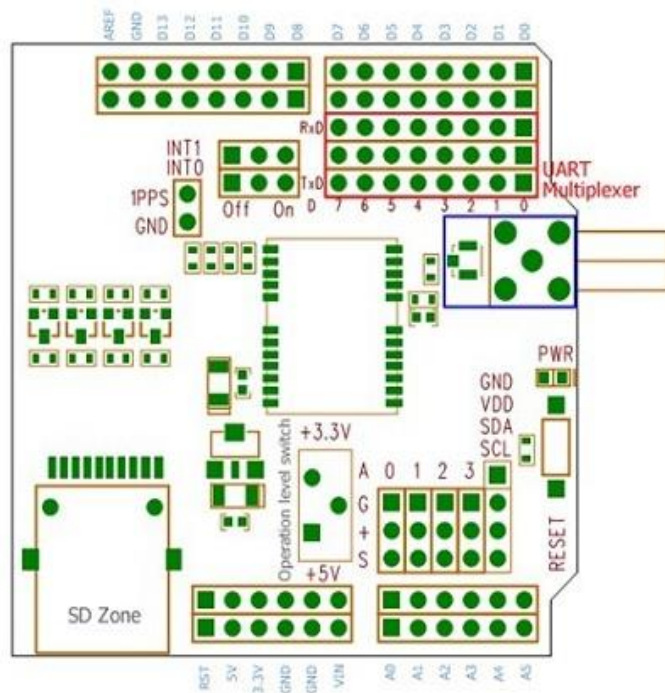


Figura 19. Diagrama GPS NEO-6M.
Fuente: (ITead Studio, 2011)

Los pines que se utilizan para la transmisión de datos con la placa de Arduino son como se indican en la tabla 12.

Tabla 12. Pines de conexión a la placa Arduino.

Arduino PIN	Descripción
D0	Datos
D1	Desde
D10	CSN
D11	MOSI
D12	MISO
D13	SCK
A0	Fugarse
A1	Fugarse
A2	Fugarse

Continúa
→

A3	Fugarse
A4	IIC_SDA
A5	IIC_SCL

Fuente: (ITead Studio, 2011)

4.1.4 Selección de impresora.

La impresora es térmica, pues no necesita de tinta para las impresiones de los recibos por lo que disminuye costos en la operación del dispositivo, es ideal para la interfaz con un microcontrolador que tiene salida de 3.3V-5V con comunicación serie TTL para imprimir texto, códigos de barra, gráficos de mapa bits. Utiliza papel térmico de 57 mm de ancho con un diámetro de rollo de 39 mm.

El panel trasero consta de dos conectores de tres pines: uno para la alimentación y el otro para la comunicación serial con velocidad de transmisión de 19200 bps. Esta no puede funcionar con la alimentación que proporciona el Arduino por lo que requiere de una corriente externa de 5V – 9V con 1.5 amperios para generar el calor necesario para realizar la impresión.



Figura 20. Impresora térmica.

Fuente: (Adafruit, 2015)

Características.- Las características técnicas de la mini impresora térmica son las que se detallan a continuación:

- Requiere de papel térmico de 57 mm de ancho.
- De 5-9V DC y 1.5 A para fuente de alimentación.
- Velocidad de impresión de 50-80 mm/seg.
- Resolución de: 8 puntos/mm, 384 puntos/línea.
- Impresión eficaz con ancho de 48 mm.
- Ancho de papel de 57.5 ± 0.5 mm.
- Diámetro de rollo máximo de 39 mm.
- Comunicación: serial TTL a 19200 baudios.
- Dimensiones: 111*65*57 mm.
- Temperatura de funcionamiento de $5^{\circ}\text{C} \sim 50^{\circ}\text{C}$.
- Temperatura de almacenamiento: $-20 \sim 60^{\circ}\text{C}$.
- Humedad de almacenamiento: 10 ~ 90%.

Alimentación.- En esta impresora no hay movimiento de tinta pero requiere una gran cantidad de corriente puesto que utiliza un cabezal térmico para calentar el papel e imprimir el recibo. Este dispositivo en particular requiere de 5V a 9V y 1.5 amperios de corriente para lo cual se utiliza un adaptador externo.



Figura 21. Alimentación impresora térmica.
Fuente: (Adafruit, 2015)

Comunicación.- En este dispositivo se envía datos a través de una conexión serie de 5V TTL que no es el mismo de 10V del puerto serie del ordenador por lo que no se puede conectar a este para su funcionamiento. Para la conexión con la placa Arduino se utiliza uno de los puertos que se encuentra en su parte posterior.

4.1.5 Selección de LCD.

Se utiliza una pantalla TFT táctil de 3.2 pulgadas con alta resolución lo que brinda gran visualización de los datos y además ya no es necesario realizar conexiones de botones para su control. Se presenta en formato shield para conectar directamente con la placa de Arduino Mega y utiliza el protocolo SPI por lo que se ahorra pines para utilizar con otros elementos.

Esta LCD se basa en un potente controlador SSD1289 y además dispone de una entrada para la tarjeta de memoria que puede ser utilizada para almacenar imágenes y ser mostradas en la pantalla, o también para guardar todo tipo de datos.

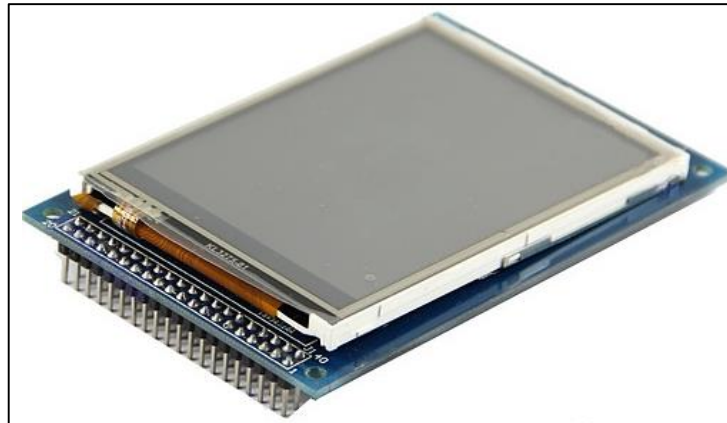


Figura 22. TFT 3.2" SSD1289.
Fuente: (Bust a Tech, 2014)

Características.- Esta LCD táctil tiene las siguientes características:

- Pantalla TFT táctil de 3.2 pulgadas.
- Resolución de 320*240.
- Potente controlador SSD1289.
- Construido en una memoria intermedia RAM de video.
- Pantalla táctil resistiva integrada.

- Controlador táctil TSC2046.
- Display de alta velocidad.
- Ranura para tarjeta micro SD.
- Retroiluminación LED.
- 65.536 colores (16 bits).
- 8 bits de interfaz digital y cuatro líneas de control.
- Utiliza alimentación de 5V.

Especificaciones.- Las especificaciones técnicas de este elemento se muestran en la siguiente tabla:

Tabla 13. Especificaciones TFT 3.2”.

Característica	Valores			
	Min	Típico	Max	Unidad
Voltaje	4.5	5	5.5	VDC
Actual	/	/	250	mA
Tamaño del panel LCD	3.2			pulgada
Ángulo de visión	60 ~ 120			Deg
Resolución	320 x 240			/
LCD a color	65k			/
Tipo de luz de fondo	LED			/
LCD IC controlador	SSD1289			/
Tipo de interfaz SD	SPI			/
Pantalla táctil	Pantalla táctil resistiva de 4 hilos			/
Dimensión	86.6 x 57.7 x 1.8			mm

Fuente: (Bust a Tech, 2014)

Diagrama y definición de pines.- La pantalla táctil utiliza un grupo de pines para el control, interfaz SPI de la SD y para la utilización del touch los cuales se indican en la figura 23.

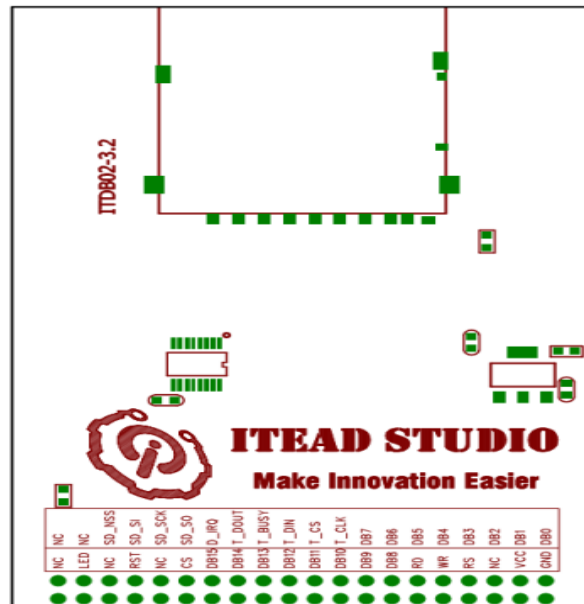


Figura 23. Diagrama lcd TFT táctil.
Fuente: (Bust a Tech, 2014)

En la figura 24 se indica los grupos de pines que se utilizan para su conexión.

LCD CS	1	LCD1	LCD_CS	RS	2	LCD RS
LCD WR	3		WR/CLK	RD	4	LCD RD
LCD RST	5		RST	DB1	6	DB1
DB2	7		DB2	DB3	8	DB3
DB4	9		DB4	DB5	10	DB5
DB6	11		DB6	DB7	12	DB7
DB8	13		DB8	DB10	14	DB10
DB11	15		DB11	DB12	16	DB12
DB13	17		DB13	DB14	18	DB14
DB15	19		DB15	DB16	20	DB16
DB17	21		DB17	GND	22	SD CS
BL_CTR	23		BL	VDD3.3	24	VCC
VCC	25		VDD3.3	GND	26	GND
GND	27		GND	BL_VDD	28	BL_VDD
SPI MISO	29		MISO	MOSI	30	SPI MOSI
T PEN	31		T_PEN	MO	32	FLASH CS
T CS	33		T_CS	CLK	34	SPI CLK

TFT_LCD

Figura 24. Pines utilizados por la pantalla táctil TFT en la placa Arduino.
Fuente: (Bust a Tech, 2014)

Pines de control del monitor.- Para el control de la pantalla de la TFT se requiere la instalación de la librería UTFT en el IDE de Arduino. Esta librería es diseñada especialmente para este tipo de pantalla utilizando el modo de 16 bits. La conexión de los pines de control del monitor a la placa Arduino son las siguientes:

- BL CTR (pin 23) → 3.3 V.
- VCC → 3.3 V.
- RD (pin 4) → 3.3 V.
- GND → GND.
- DB1 ~ DB8 → pin D37 ~ D30.
- DB10 ~ DB17 → pin D22 ~ D29.
- RS (pin 2) → D38.
- WR (pin 3) → D39.
- LCD_CS (pin 1) → D40.
- RSET (pin 5) → D41.

Pines de tarjeta SD.- La TFT consta de una ranura para la tarjeta SD por lo que puede ser usada para subir imágenes. Estas imágenes deben ser convertidas en formato RAW para ser leídas por la pantalla, para lo cual son necesarias las librerías SD tinyFAT y tinyFAT_16 para el uso y visualización de las imágenes almacenadas en la tarjeta de memoria. La conexión de los pines para la interfaz con la placa Arduino son los siguientes:

- SCK (pin 34) → D52.
- SD_CS (pin 22) → D53.
- MISO (pin 29) → D50.
- MOSI (pin 30) → D51.

Los pines MOSI utiliza el Arduino para enviar datos a la tarjeta SD, mientras que el MISO utiliza el Arduino para recibir datos desde la tarjeta SD. La pantalla táctil y la tarjeta SD comparten el MISO y MOSI, y son controlados a través de T_CS y SD_CS por

separado. Mientras T_CS está activa, la pantalla táctil funciona, y mientras SD_CS está activa, la tarjeta SD está en utilización.

Pines de control táctil.- La TFT tiene una pantalla táctil resistiva de 4 hilos y para su utilización es necesaria la instalación de la librería Touch. La conexión de los pines para el control táctil son los siguientes:

- DCLK (pin 34) → D6.
- T_CS (pin 33) → D5.
- MOSI (pin 30) → D4.
- MISO (pin 29) → D3.
- T_PEN (pin 31) → D2.

En la siguiente figura se pueden observar un modelo de las cuatro resistencias que forman parte de la pantalla táctil.

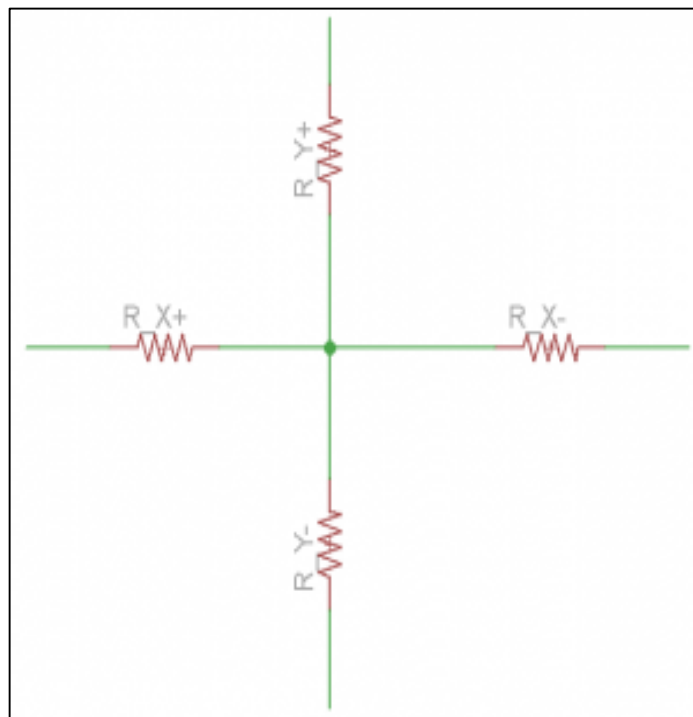


Figura 25. Modelo de resistencias de la pantalla táctil.
Fuente: (Bust a Tech, 2014)

4.2 Diseño de software.

4.2.1 Etapas de diseño.

La construcción del dispositivo se realiza en dos etapas, la primera está constituida por el diseño del taxímetro, la cual está conformado por algunas subrutinas de trabajo como son el inicio y fin de funcionamiento, menú de control, ventana de reloj, almacenamiento de datos y la impresión de recibo; mientras que la segunda es el desarrollo del software LabVIEW para realizar el respectivo análisis de los datos almacenados, los cuales se observan en las ventanas de datos, estadística origen y destino, estadística general y representación gráfica. Estas etapas se las puede observar en la figura 26.

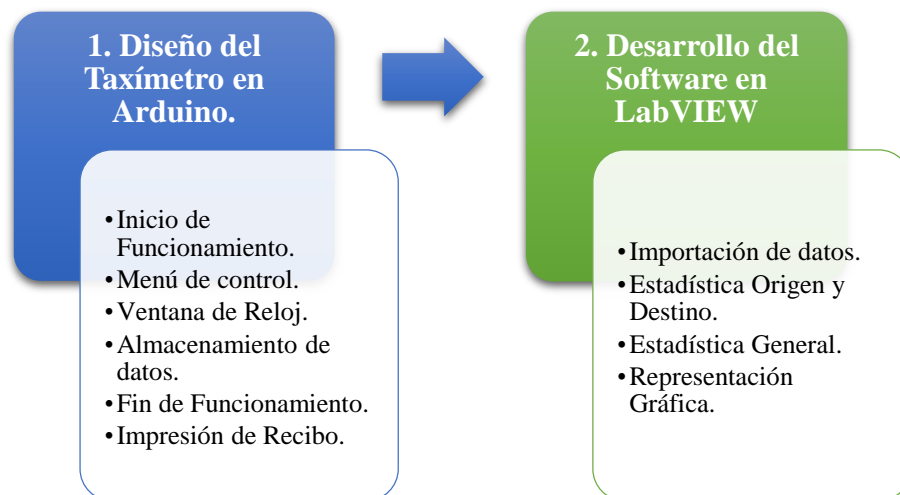


Figura 26. Etapas de diseño.
Fuente: (Morales)

4.2.2 Flujogramas del dispositivo taxímetro.

El taxímetro está diseñado con el objetivo de adquirir datos durante su funcionamiento, de manera que con su respectivo análisis a través del software creado se obtenga información sobre la movilidad de las personas en la ciudad y de esta manera realizar mejoras en el sistema de transporte local. El funcionamiento general del dispositivo se indica en el flujograma de la figura 27.

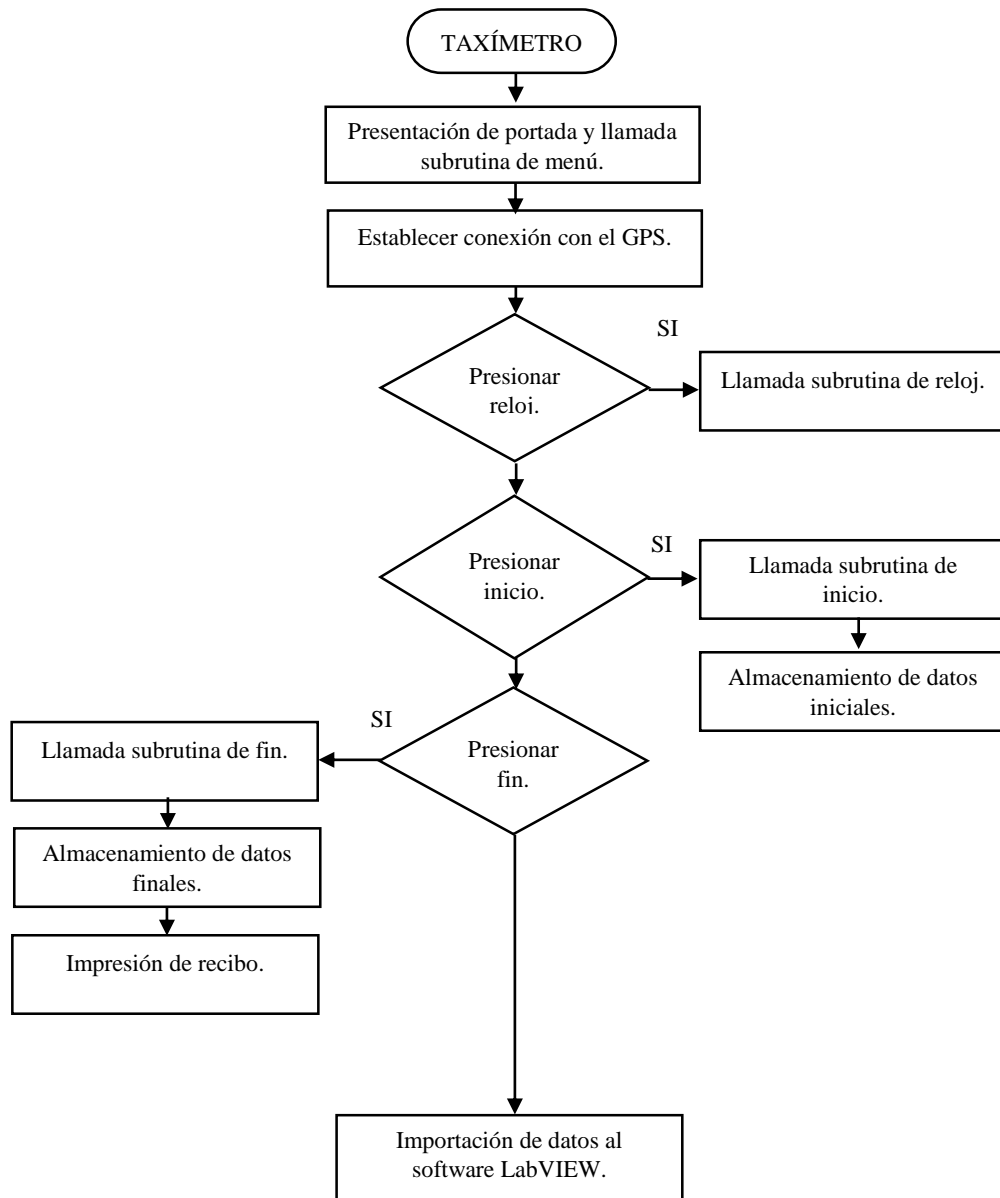


Figura 27. Flujograma del funcionamiento del dispositivo.
Fuente: (Morales)

El programa diseñado para el funcionamiento del taxímetro con almacenamiento de viajes mediante GPS consta de subrutinas que se ejecutan cuando sea necesario. A cada una de ellas se las puede entender como un conjunto de instrucciones que cumplen con una determinada función. El funcionamiento del dispositivo se lo puede observar en la figura 27.

Al programa se lo puede dividir en las siguientes subrutinas:

- Programa de inicio.
- Subrutina de menú.

- Subrutina de reloj.
- Subrutina de inicio.
- Subrutina de fin.
- Subrutina de almacenamiento.
- Subrutina home.

Flujograma del programa de inicio.- Al iniciar el programa cargado en la placa Arduino Mega se definen las variables y sus respectivos valores, configuran puertos, así como también los pines obtienen valores iniciales. Este flujograma indica una subrutina antes de ejecutar el programa principal.

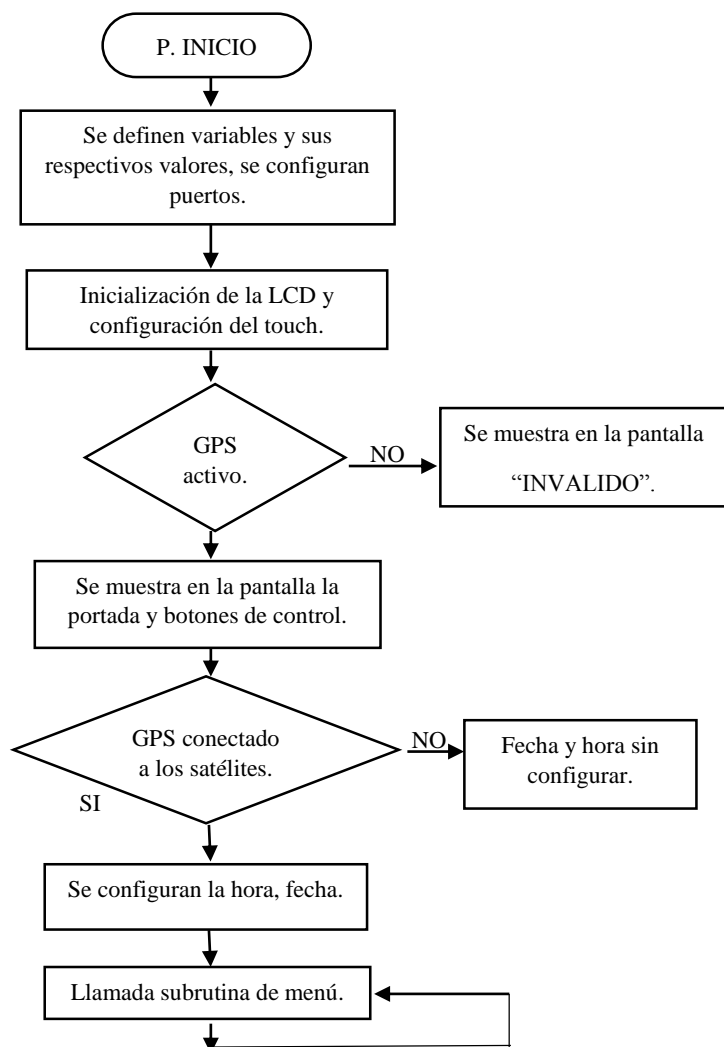


Figura 28. Flujograma del programa de inicio.
Fuente: (Morales)

Flujograma de subrutina menú.- La subrutina de menú del dispositivo es la que permite visualizar los tres botones de control táctil que permiten dar inicio, fin, indicar la hora y fecha tomadas desde el GPS. Al presionar cualquiera de ellos se puede acceder a las funciones respectivas.

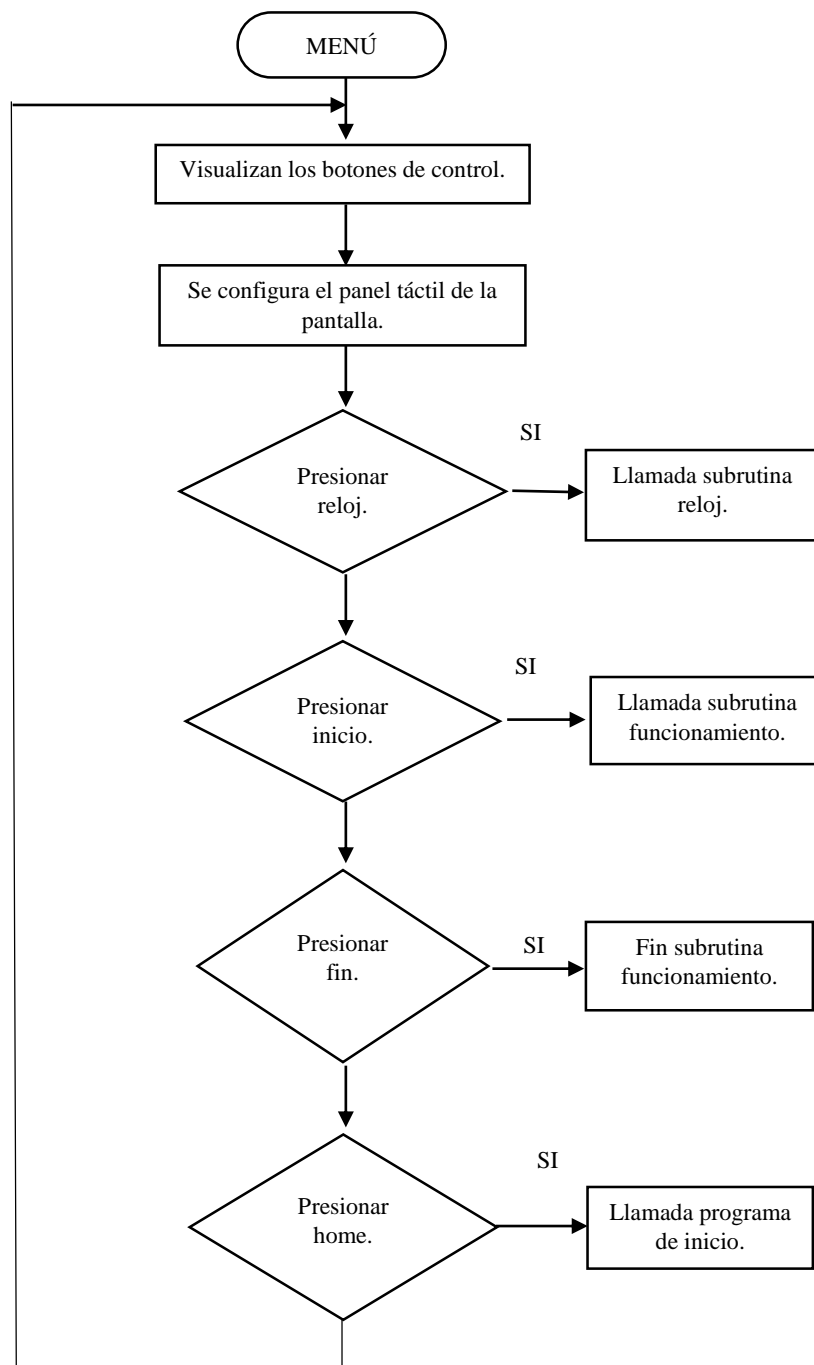


Figura 29. Flujograma de subrutina de menú.
Fuente: (Morales)

Flujograma de subrutina de reloj.- Esta subrutina se ejecuta para visualizar en una nueva ventana un mensaje de bienvenida, así como también la fecha y hora actual, tomada del GPS. En esta se puede observar el menú y un nuevo botón para la subrutina home.

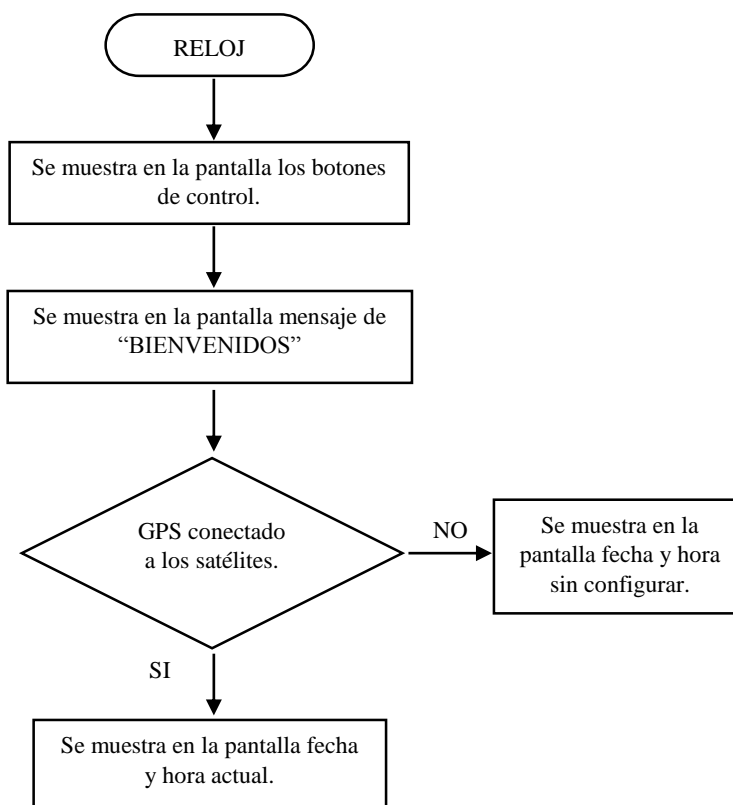


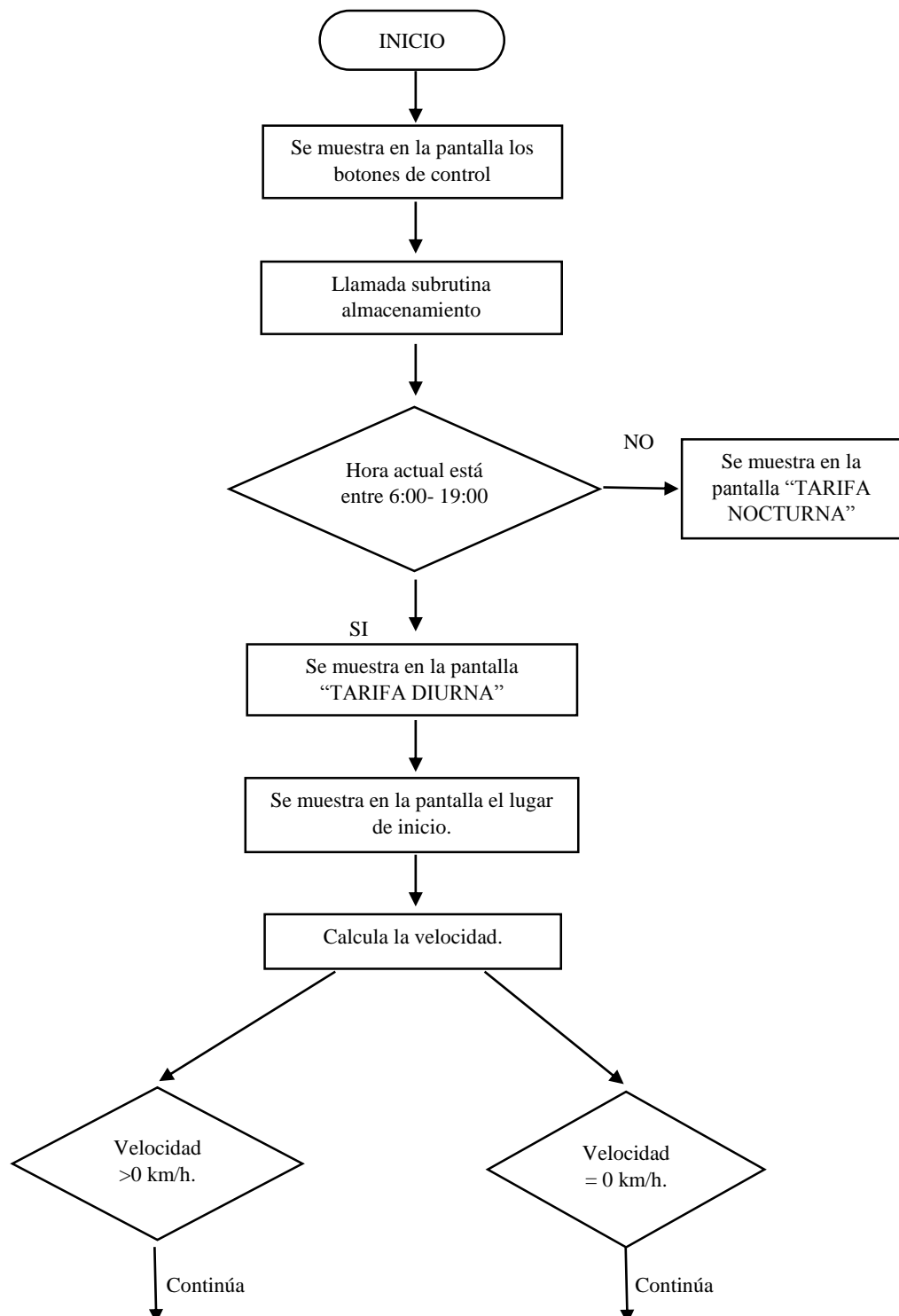
Figura 30. Flujograma de subrutina de reloj.
Fuente: (Morales)

Flujograma de subrutina de inicio.- Esta subrutina se ejecuta para poner en funcionamiento el taxímetro y calcular el costo de la carrera realizada; para el correcto funcionamiento se aplica el reglamento de la tarifa de taxis.

Al presionar el botón de inicio se llama a la subrutina de almacenamiento; luego se verifica la hora para mostrar en la pantalla el tipo de tarifa; con la ayuda del GPS se establece el lugar de partida para mostrarlo en la pantalla y posteriormente se calcula la velocidad del vehículo, si ésta es mayor a 0 Km/h se realiza el cobro por kilómetro

recorrido, pero si es igual 0 Km/h se realiza el cobro por tiempo de espera según los parámetros del reglamento.

En esta subrutina se cuenta con los botones de fin y home; el primero finaliza la subrutina mostrando una nueva ventana en la que se observa el lugar de llegada, el subtotal y total de la carrera y da la orden a la impresora de generar el recibo, mientras que el segundo llama al programa de inicio.



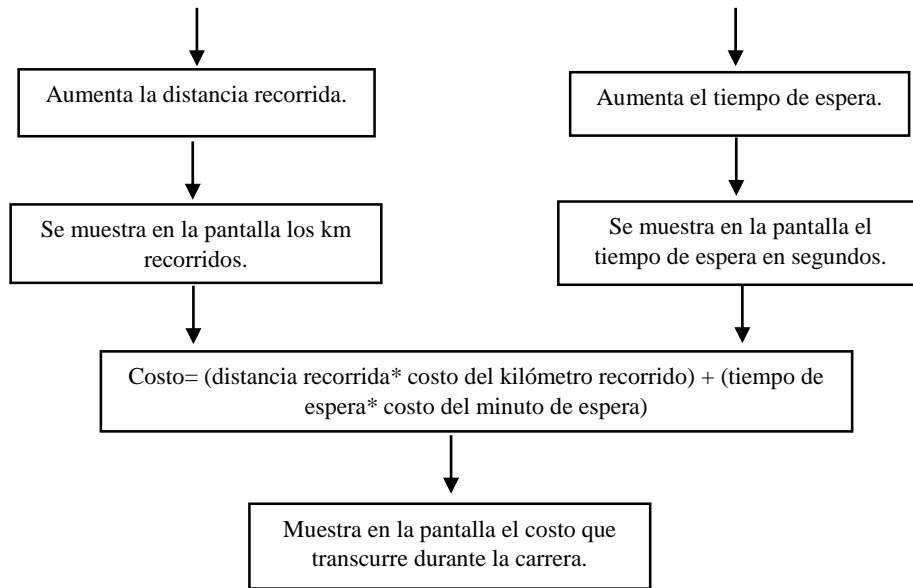
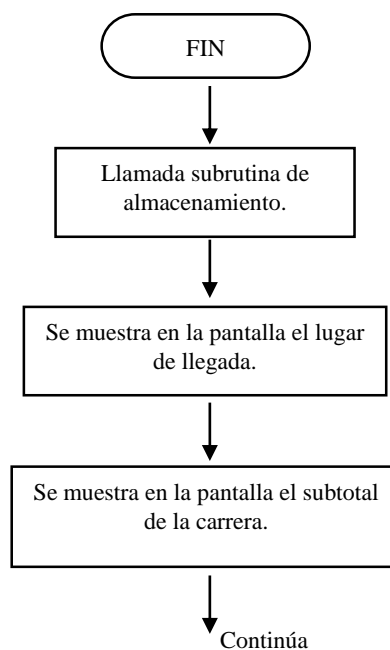


Figura 31. Flujograma de subrutina de inicio.
Fuente: (Morales)

Flujograma de subrutina de fin.- En primera instancia se llama a la subrutina de almacenamiento para guardar la información del lugar de llegada, luego en una nueva ventana de la pantalla se presenta el lugar de llegada, el subtotal y el total de la carrera; posteriormente se muestra un texto de IMPRIMIENDO, para finalmente dar la orden a la impresora de generar el recibo. En esta subrutina también se visualizan los botones de control.



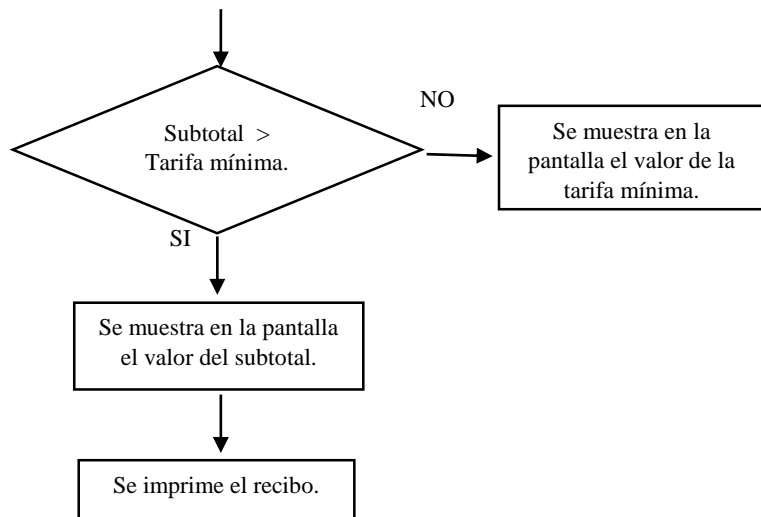
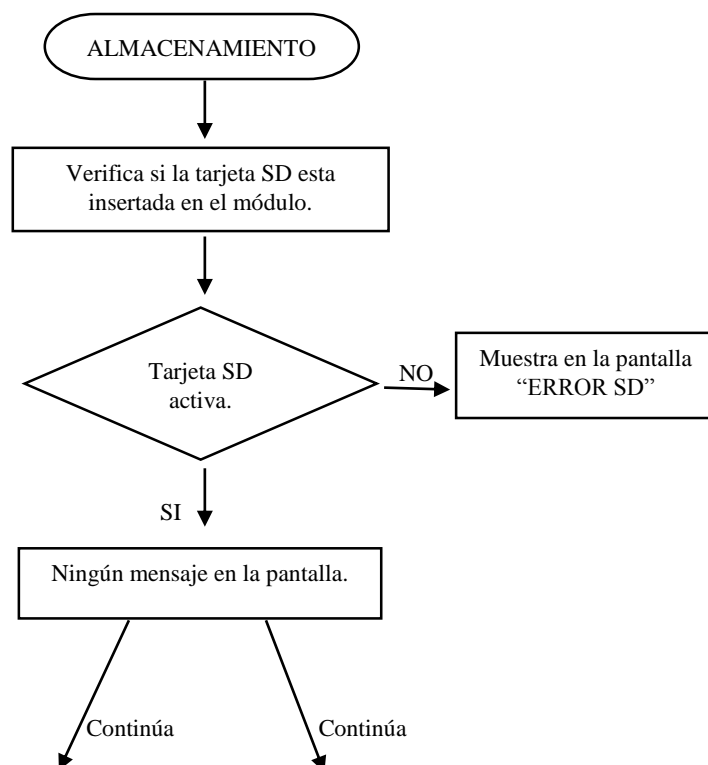


Figura 32. Flujograma de subrutina de fin.
Fuente: (Morales)

Flujograma de subrutina de almacenamiento.- En esta subrutina se realiza el almacenamiento de datos de los viajes realizados, los cuales son guardados en un archivo Excel, mientras que en un archivo de texto se almacena la ruta seguida por el vehículo. Los datos son obtenidos a través de la acción de los botones de inicio y fin.



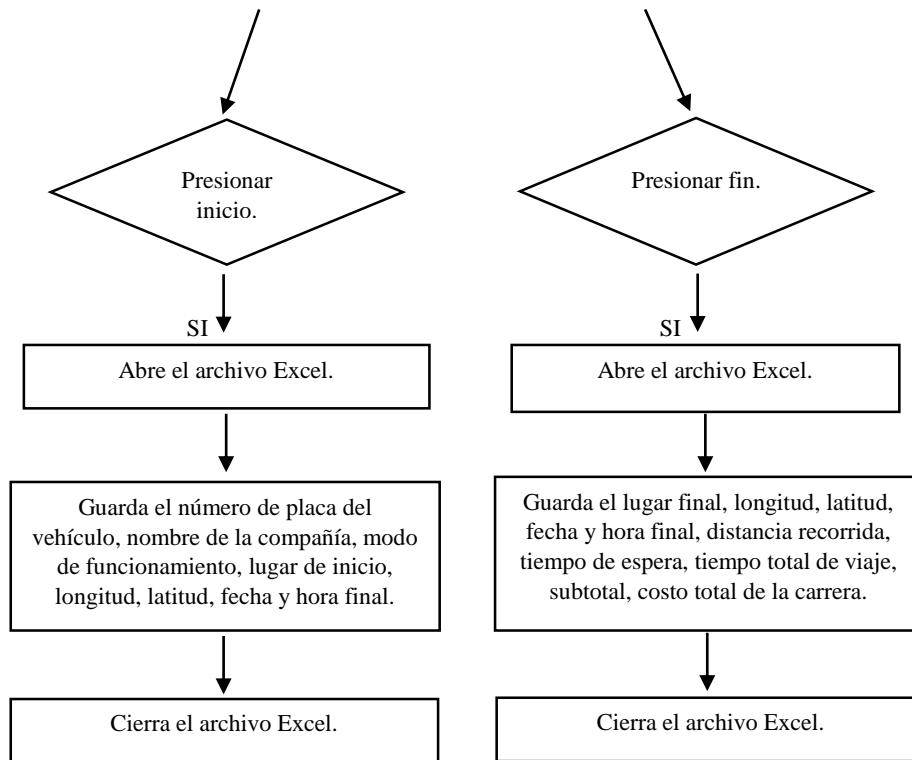


Figura 33. Flujograma de subrutina de almacenamiento.
Fuente: (Morales)

Flujograma de subrutina home.- En esta subrutina se regresa a la portada, así como también se realiza la llamada al programa inicio y las diferentes variables utilizadas regresan a los valores iniciales. Al fin de esta subrutina se observa la portada con los botones de control.

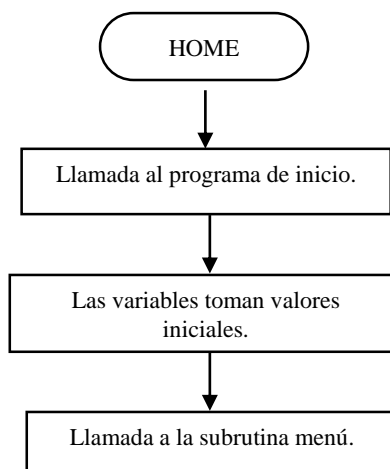


Figura 34. Flujograma de subrutina home.
Fuente: (Morales)

4.2.3 Diseño del programa de análisis de datos en el software LabVIEW.

LabVIEW es un instrumento de programación gráfica que inicialmente fue orientado a las aplicaciones de control de elementos electrónicos de instrumentación virtual. Por tal razón todos los programas que son realizados en esta herramienta se guardan en tipos de ficheros VI, cuyo significado es instrumento virtual.

Utiliza un lenguaje y a la vez un entorno de programación gráfica en el que se pueden crear aplicaciones de una forma rápida y sencilla. El programa creado realiza un conjunto de actividades para el análisis de datos, las cuales se observa en la siguiente figura.

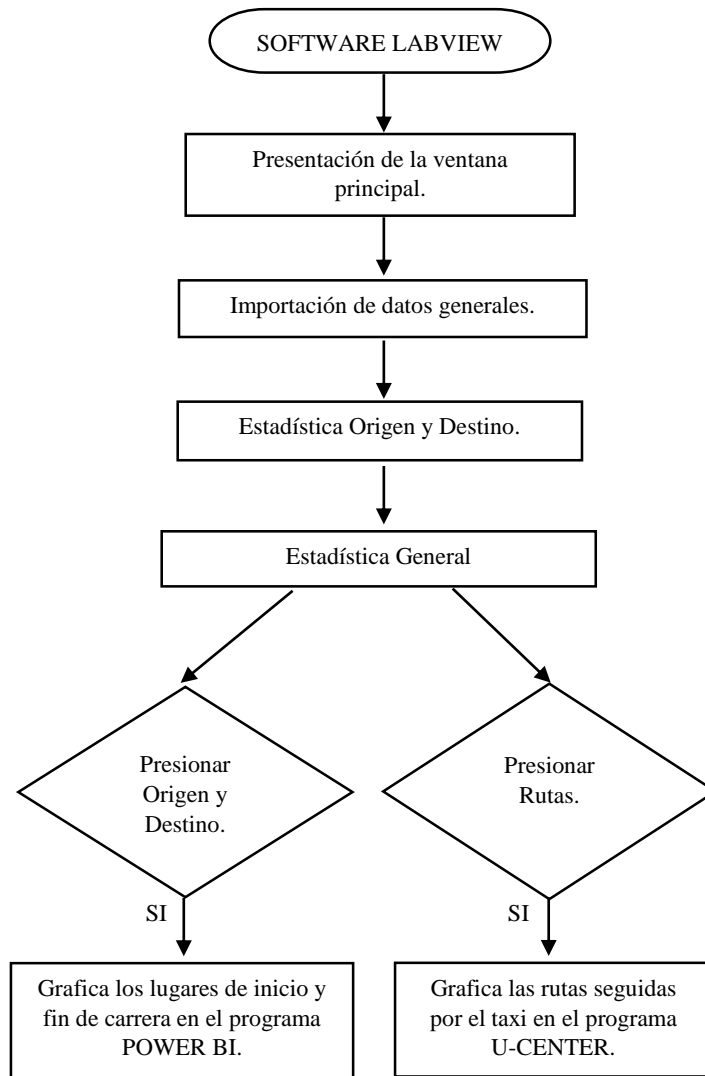


Figura 35. Diagrama de flujo del programa de análisis de datos.

Fuente: (Morales)

Ventana principal.- La pantalla principal del programa consta de un menú realizado a través de un Tab control con cinco páginas, las cuales son denominadas: “HOME”, “DATOS”, “ESTADÍSTICA ORIGEN-DESTINO”, “ESTADÍSTICA GENERAL”, “REPRESENTACIÓN GRÁFICA”.



Figura 36. Ventana principal.
Fuente: (Morales)

Ventanas del programa.- La primera ventana del programa denominada “DATOS”, indica los datos almacenados de la tarjeta SD. Esta consta de dos botones llamados importar datos y borrar datos.

El primero se utiliza para importar datos de los viajes realizados desde el archivo Excel de la tarjeta de memoria a la tabla que se muestra en la ventana. La tabla consta de 17 columnas en las cuales se observa los siguientes datos: placa, nombre de la cooperativa/compañía, tipo de taxi (convencional/ejecutivo), modo de operación (día, noche), lugar de partida, posición inicial, fecha inicial, hora inicial, lugar de llegada, posición final, fecha final, hora final, distancia recorrida en Km, tiempo de espera en segundos, tiempo total de viaje en minutos, subtotal y costo total de la carrera realizada.

El segundo se utiliza para borrar los datos del archivo Excel de la tarjeta de memoria para empezar a guardar los datos de los nuevos viajes a realizarse. La primera ventana se la puede observar en la figura 36.

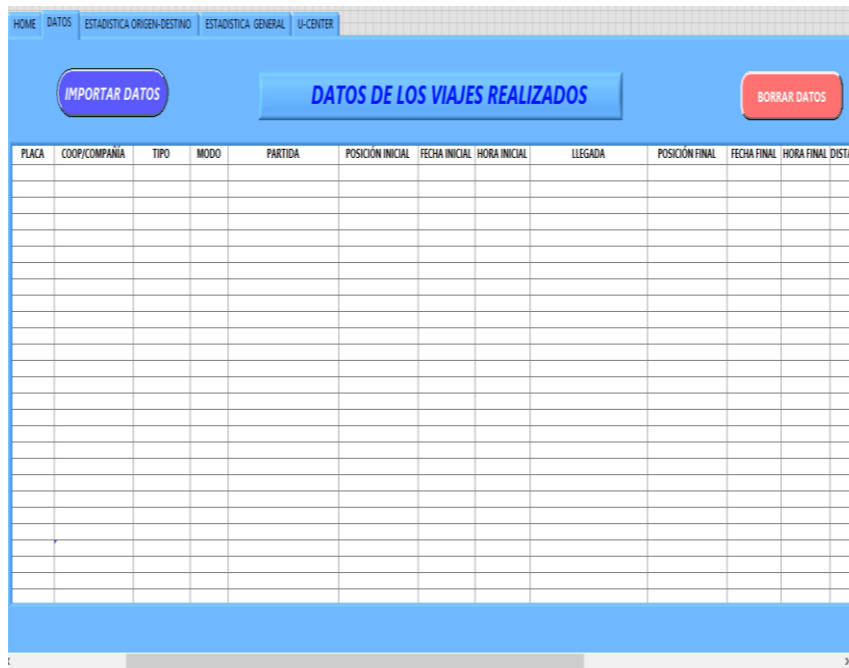


Figura 37. Ventana de importación de datos.
Fuente: (Morales)

A continuación, en la figura 37 se observa el diagrama de bloques de la ventana “DATOS”, para la cual se utiliza un case structure puesto que puede contener uno o más sub diagramas que se ejecutan cuando se llama a la estructura principal.

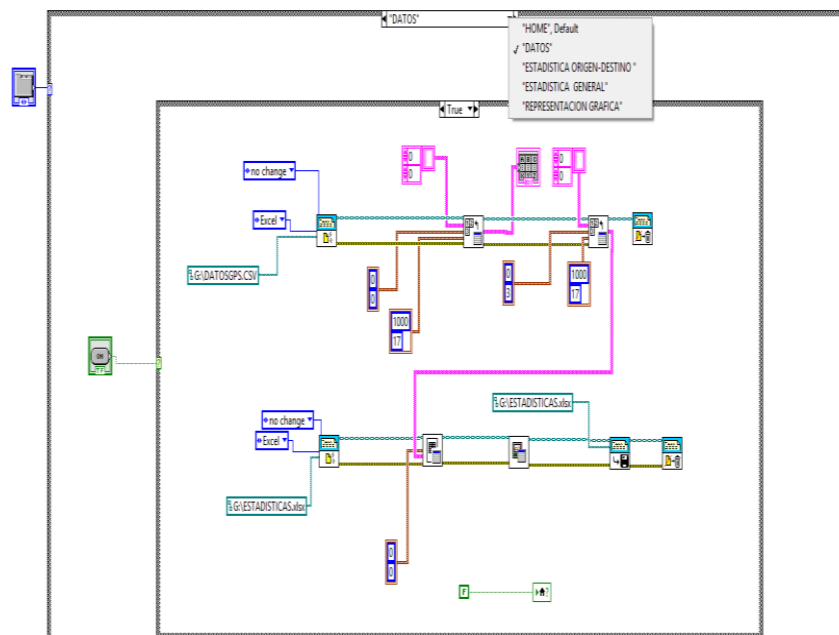


Figura 38. Código para la importación de datos.
Fuente: (Morales)

Para la importación de datos desde el archivo Excel se utiliza una biblioteca llamada LabVIEW Report Generation Toolkit para Microsoft Office que contiene Vis flexibles para crear y editar de manera programática reportes de Microsoft Excel y Word desde NI LabVIEW.

Para la obtención de datos desde el archivo guardado se presiona el botón importar datos que se visualiza en el panel frontal, el cual procede a ejecutar el código estructurado. Para este código es necesario utilizar el New Report VI que procede a abrir el archivo Excel llamado DATOSGPS almacenado en la tarjeta de memoria; también se utiliza la herramienta Excel Get Data VI con la cual se obtiene la información del rango de celdas que se especifica en la misma, que luego se podrá visualizar en la tabla creada. Al finalizar este procedimiento el archivo Excel abierto se vuelve a cerrar con la utilización de la herramienta Dispose Report VI.

Antes de cerrar el archivo mencionado, se copia la información importada a otro archivo Excel llamado ESTADÍSTICAS, el cual es manipulado con la utilización de las herramientas mencionadas anteriormente. En este archivo se guardan los datos adquiridos que serán utilizados para realizar un cálculo estadístico de origen y destino con la ejecución de las fórmulas almacenadas anteriormente en el archivo en ejecución. Todo este procedimiento se ejecuta con el código estructurado que se indica en la figura 38.

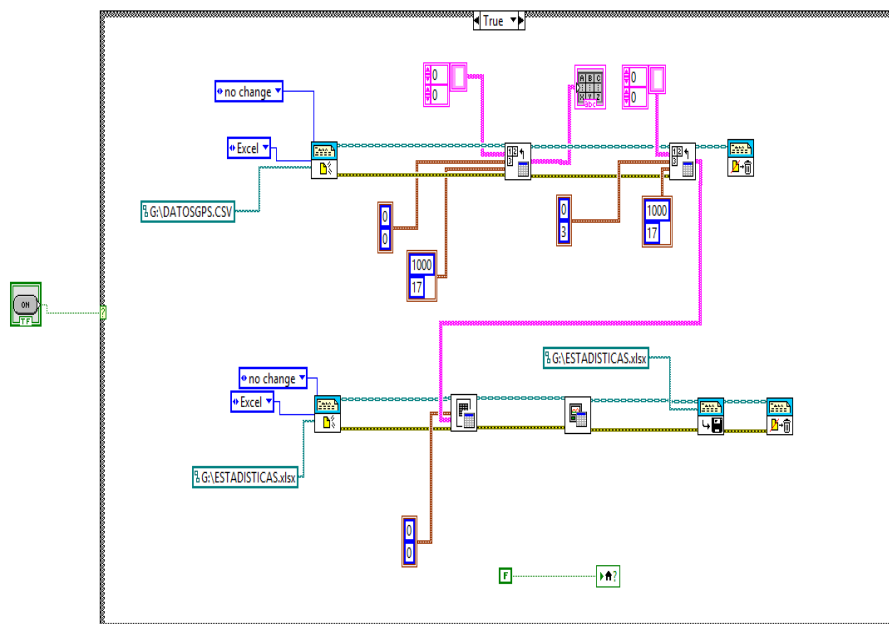


Figura 39. Código para la creación del archivo Excel estadísticas.
Fuente: (Morales)

En la figura 39 se indica el código con el cual se elimina la información almacenada en el archivo principal DATOSGPS, para que de esta manera con la utilización del dispositivo se empiecen a guardar los nuevos datos adquiridos. Este código se ejecuta al presionar el botón borrar datos.

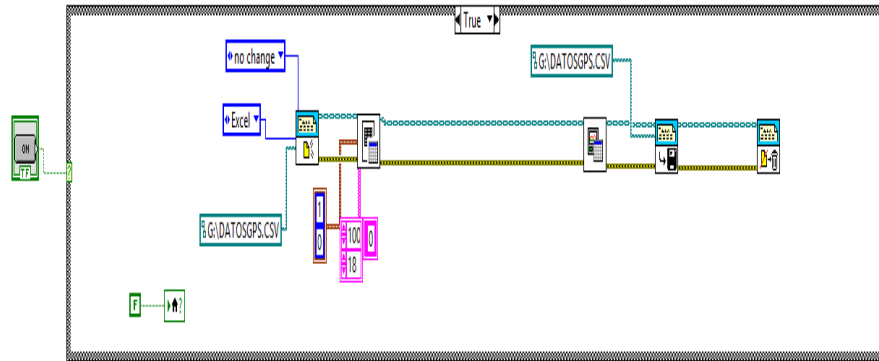


Figura 40. Código para eliminar datos.
Fuente: (Morales)

En la siguiente figura se indica el panel frontal de la segunda ventana llamada “ESTADÍSTICA ORIGEN-DESTINO”, en la cual se indica el número de viajes realizados desde el lugar en el que se inicia el dispositivo hasta el lugar que se da fin. Para obtener los datos se dispone de un botón DATOS, con el cual se importa la información procesada en el archivo Excel ESTADÍSTICAS y así lograr determinar desde donde y hacia qué lugar hay mayor número de viajes.

NÚMERO DE VIAJES EN LA CIUDAD DE IBARRA	
LUGAR DE DESTINO	
LUGAR DE PARTIDA	LUGAR DE DESTINO
AZAYA	AZAYA
HUERTOS FAMILIARES	HUERTOS FAMILIARES
YAHUARCOCHA	YAHUARCOCHA
LAS PALMAS	LAS PALMAS
HOSPITAL DEL SEGURO	HOSPITAL DEL SEGURO
ESTADIO	ESTADIO
EL OLIVO	EL OLIVO
JARDIN DE PAZ	JARDIN DE PAZ
HOSPITAL SAN VICENTE DE PAUL	HOSPITAL SAN VICENTE DE PAUL
LA MERCED	LA MERCED
PUGACHO	PUGACHO
REDONDEL DE LA MADRE	REDONDEL DE LA MADRE
CENTRO MERCADO	CENTRO MERCADO
LA VICTORIA	LA VICTORIA
LA FLORIDA	LA FLORIDA
EL EIIDO	EL EIIDO
PARQUE CENTRICA BULEVAR	PARQUE CENTRICA BULEVAR
TERMINAL TERRESTRE	TERMINAL TERRESTRE
YACUCALLE, PARQUE DE LA FAMILIA	YACUCALLE, PARQUE DE LA FAMILIA
LOS CEIBOS	LOS CEIBOS
LA CAMPINA Y PRIMAVERA	LA CAMPINA Y PRIMAVERA
10 DE AGOSTO	10 DE AGOSTO
CARANQUI	CARANQUI
EIIDO DE CARANQUI	EIIDO DE CARANQUI
LA ESPERANZA	LA ESPERANZA
PRIORATO	PRIORATO
MILAGRO	MILAGRO
SAN ANTONIO	SAN ANTONIO

Figura 41. Ventana de estadística origen y destino.
Fuente: (Morales)

El diagrama de bloques de esta ventana se presenta en la figura 41, en la cual se abre el archivo Excel ESTADÍSTICA y se importan los datos a la tabla que se indica en panel frontal para poder visualizar el número de carreras desde y hacia los diferentes lugares indicados en el mapa de la sectorización de la ciudad de Ibarra. En este código se utiliza la herramienta Index Array que nos ayuda a obtener el dato del total de las carreras realizadas, que fue calculado en el archivo Excel abierto. El código estructurado se ejecuta al presionar el botón llamado datos.

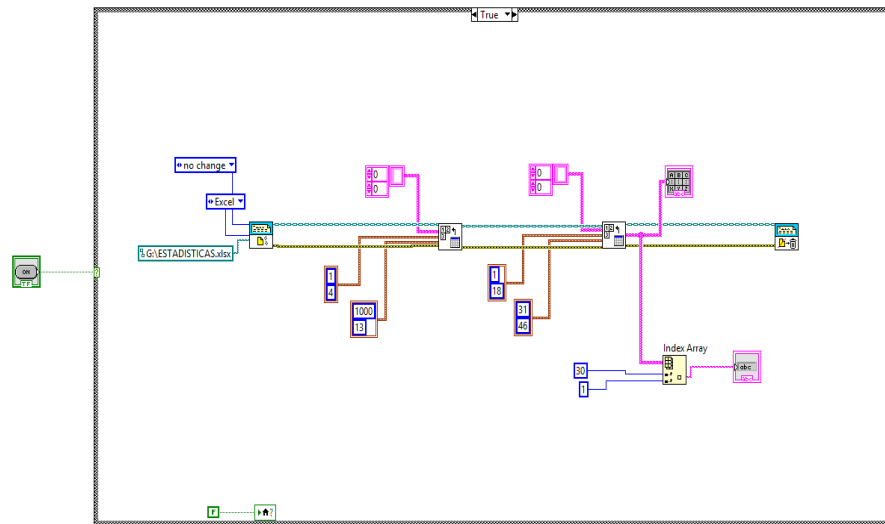


Figura 42. Código para importar datos del número de carreras realizadas.
Fuente: (Morales)

En la tercera ventana denominada “ESTADÍSTICA GENERAL”, se realiza un cálculo general de las carreras realizadas. En el panel frontal se puede observar dos tablas; en una de ellas se puede visualizar las fechas disponibles para seleccionar un rango y realizar el cálculo de la información requerida. Estas fechas son seleccionadas a través de dos controladores con los que se puede definir el inicio y el fin del rango de fechas a utilizar.



Figura 43. Control del rango de fechas.
Fuente: (Morales)

En la tabla principal se observa tres columnas llamadas DÍA, NOCHE, TOTAL, en las cuales se podrán visualizar datos específicos del rango de fechas seleccionadas, estas columnas contienen los siguientes datos: número de viajes, kilómetros recorridos, tiempo de espera, tiempo de viaje, costos generados durante las fechas seleccionadas. Estos datos son obtenidos al pulsar el botón calcular. El panel frontal de esta ventana se puede observar en la figura 43.

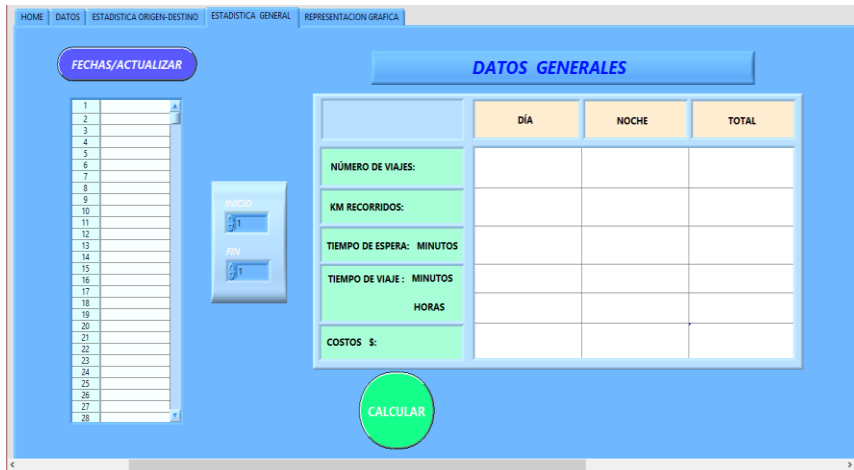


Figura 44. Ventana de estadística general.
Fuente: (Morales)

Esta ventana consta de dos sub diagramas; en el primero se abre el archivo Excel DATOSGPS del cual se obtienen las fechas disponibles de los datos almacenados en este archivo. Además a través de este código con la utilización de los controladores del panel frontal se establece el rango de las fechas para realizar el cálculo estadístico. Estos datos seleccionados son almacenados en otro documento Excel llamado ESTADÍSTICA GENERAL, en el cual se realiza el cálculo de la información mostrada en la tabla principal del panel frontal.

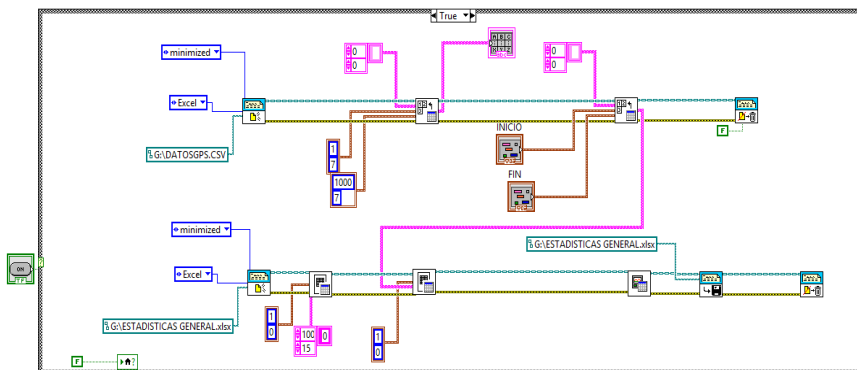


Figura 45. Código para establecer el rango de fechas.
Fuente: (Morales)

Con el segundo sub diagrama se importan los datos calculados con las fórmulas establecidas en el archivo Excel ESTADÍSTICAS GENERAL, que son mostrados detalladamente en la tabla principal.

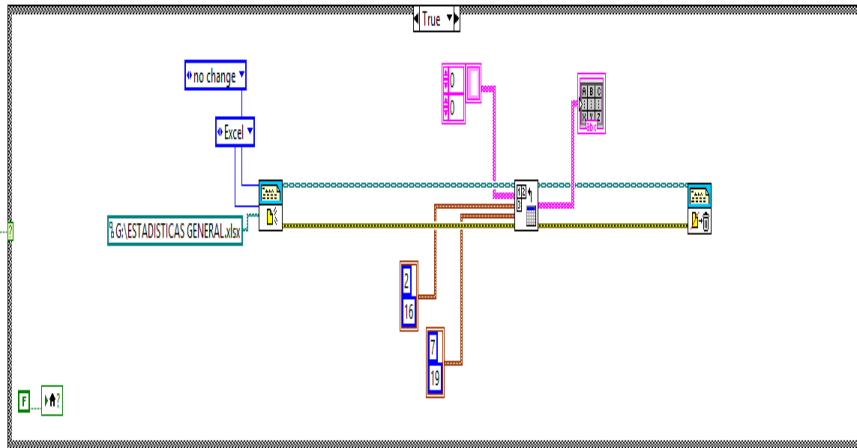


Figura 46. Código para importar datos generales.
Fuente: (Morales)

En la figura 46 se observa la última ventana del programa en la que se muestran imágenes de la herramienta de Power BI y U-center, que se utiliza para graficar los lugares de inicio y fin de las carreras realizadas, así como también las rutas seguidas por el vehículo en los diferentes viajes. Estos son visualizados al pulsar el botón ORIGEN Y DESTINO Y el botón RUTAS que se indican en el panel frontal.



Figura 47. Ventana de representación gráfica.
Fuente: (Morales)

El programa Power BI proporciona una experiencia de inmersión para hacer descubrimientos de datos que no pueden verse en tablas y gráficos 2D tradicionales. Esta herramienta utiliza una versión similar al google map que permite trazar datos geográficos para analizarlos en 3D y crear rutas cinematográficas. De esta manera se grafican los lugares de inicio de la carrera con puntos de color azul y los lugares de llegada con color rojo, siendo de esta manera muy fácil la visualización de los lugares con mayor salida de viajes, así como también los lugares con mayor llegada.

A través de esta herramienta se visualiza dos partes; en la primera los lugares de inicio de cada carrera y en la segunda los lugares finales, así también debajo de cada una de ellas se presentan gráficos de barras que indican el número de carreras realizadas desde y hacia cada uno de los lugares definidos en la sectorización.

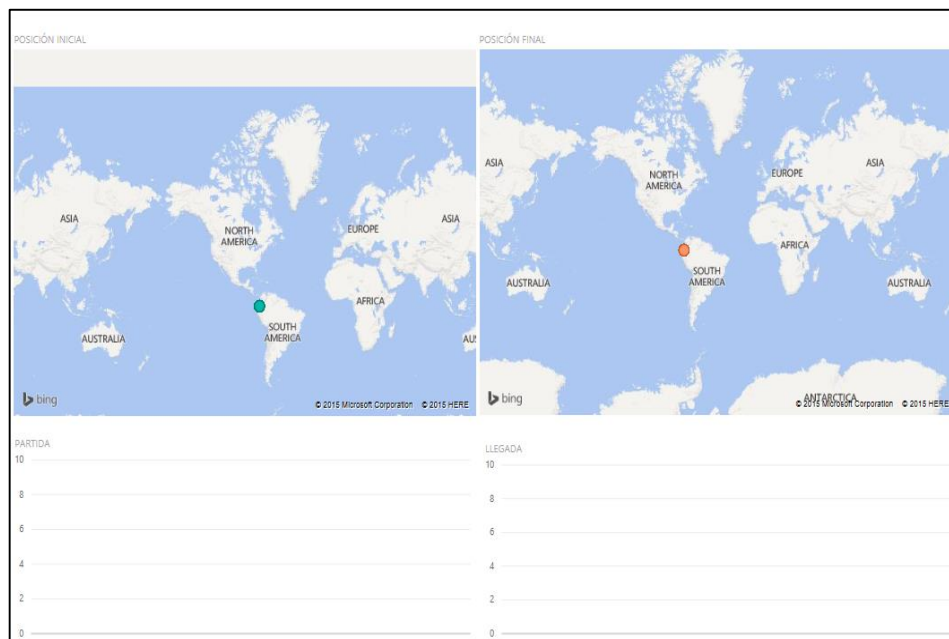


Figura 48. Ventana del programa Power BI.
Fuente: (Morales)

En la figura 48 se observa el código estructurado que permite visualizar esta ventana, el cual está conformado principalmente por un System Exec VI que permite ejecutar la aplicación Power BI.

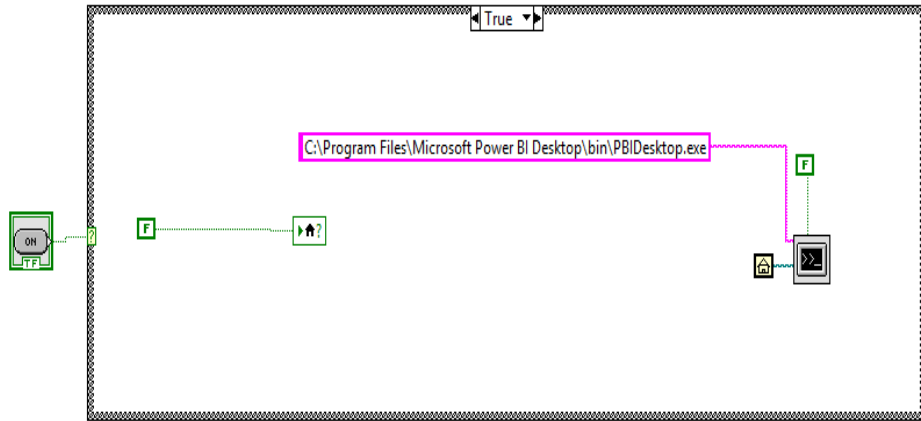


Figura 49. Código para ejecutar el programa Power BI.
Fuente: (Morales)

Adicionalmente tenemos el botón U-CENTER que permite ejecutar la aplicación Ublox U-center, la cual permite graficar las rutas seguidas por el vehículo utilizando el archivo de texto almacenado en la tarjeta de memoria.

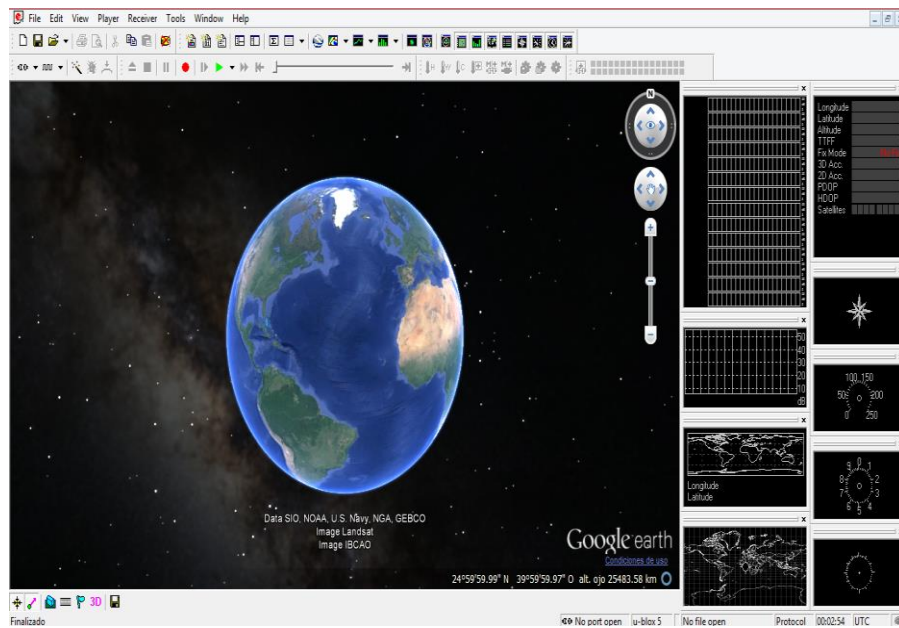


Figura 50. Ventana del programa U-center.
Fuente: (Morales)

En la figura 50 se observa el código estructurado que permite visualizar esta ventana, el cual está conformado principalmente por un System Exec VI que permite ejecutar la aplicación Ublox U-center.

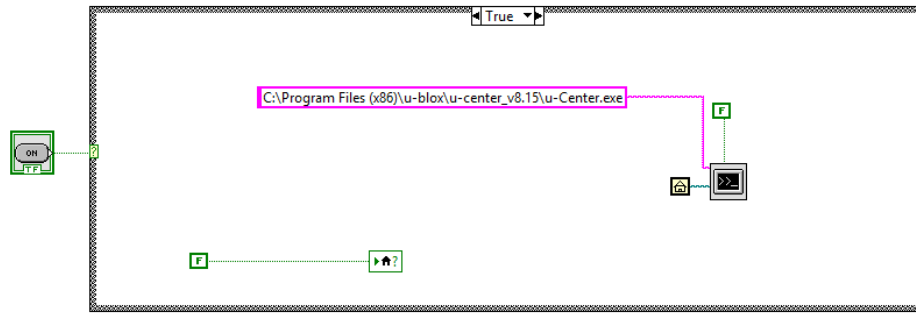


Figura 51. Código para ejecutar el programa U-center.
Fuente: (Morales)

4.2.4 Comunicación del dispositivo con el software LabVIEW.

Para la comunicación y la importación de los datos almacenados por el dispositivo al Software LabVIEW, es necesario retirar la tarjeta SD para insertarla en una PC. De esta manera a través del puerto USB y la ejecución del programa creado se obtiene fácilmente los datos requeridos para su respectivo análisis. Los pasos a seguir para la importación de datos se observan en la siguiente figura.

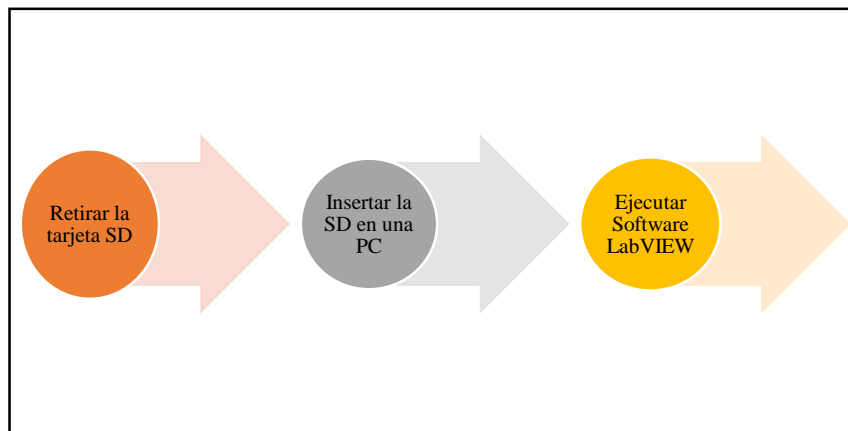


Figura 52. Comunicación del dispositivo con el Software LabVIEW.
Fuente: (Morales)

4.2.5 Sectorización de la ciudad de Ibarra.

Criterios de Sectorización.- Para la sectorización se consideran los valores de longitud y latitud, que son utilizados para la división de la ciudad en forma de cuadrantes. Con la utilización del módulo GPS se puede obtener dichos los valores, logrando identificar fácilmente la posición de inicio y fin del viaje realizado por el taxi. En base al modelo de cuatro etapas, los criterios de sectorización se los puede clasificar en tres aspectos, los cuales se indican en la figura 52.

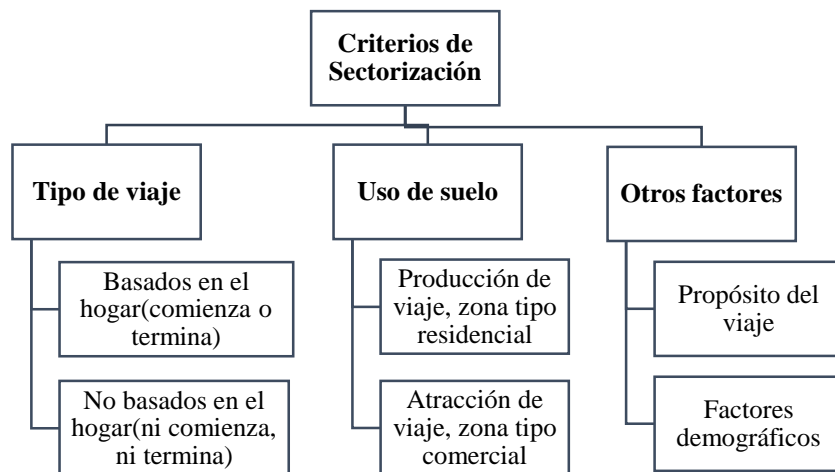


Figura 53. Criterios de sectorización.
Fuente: (Morales)

Zonas de la ciudad de Ibarra.- En base a los criterios de sectorización, se realiza la división de posibles zonas de atracción o producción de viajes, para lo cual, se sugiere que el área de cada cuadrante en zonas urbanas no sea menor de 0.2 km², ni mayor a 0.8km², mientras que en zonas perimetrales, el área del cuadrante puede ser máximo de 10km². De esta manera se realiza la división de la ciudad de Ibarra en 28 cuadrantes, los cuales se indican en la tabla 14.

Tabla 14. Lugares sectorizados en la ciudad de Ibarra.

Sectorización de la ciudad de Ibarra					
1	Azaya	11	Pugacho	21	La Campiña y Primavera
2	Huertos Familiares	12	Redondel de la Madre	22	10 de Agosto
3	Yahuarcocha	13	Centro, Mercado	23	Caranqui
4	Alpachaca	14	La Victoria	24	Ejido de Caranqui
5	Hospital del Seguro	15	La Florida	25	La Esperanza
6	Estadio	16	El Ejido	26	Priorato
7	El Olivo	17	Parque Céntrica Bulevar	27	Milagro

Continúa →

8	Jardín de Paz	18	Terminal Terrestre	28	San Antonio
9	Hospital San Vicente de Paul	19	Yacucalle		
10	La Merced	20	Los Ceibos		

Fuente: (Morales)

Los cuadrantes que se trazaron para cada uno de los lugares se indican en el mapa de la figura 53.

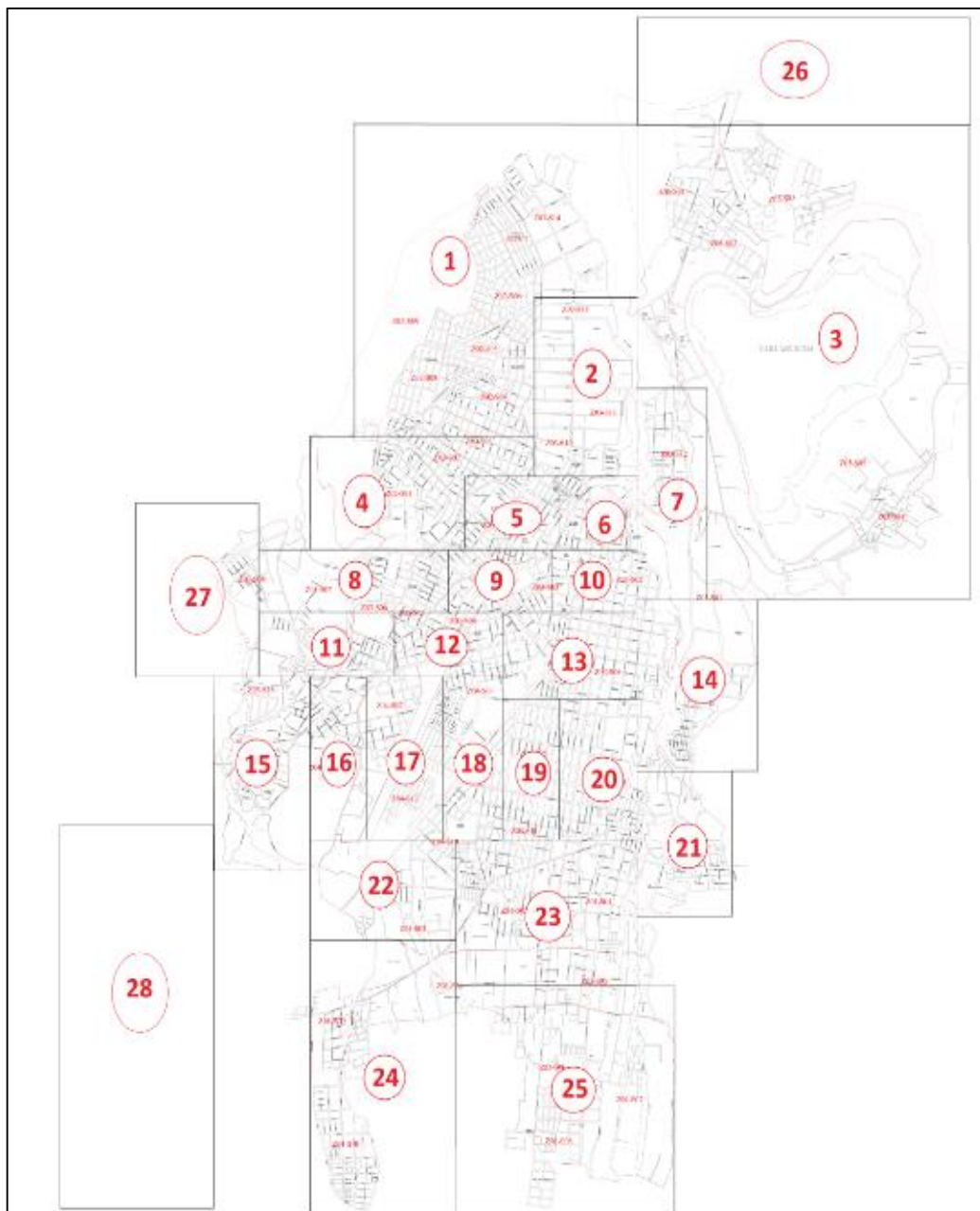


Figura 54. Mapa de las zonas de la ciudad de Ibarra.
Fuente: (Morales)

4.2.6 Diagramación de circuitos electrónicos.

A continuación se indican los circuitos eléctricos de los diferentes elementos electrónicos del dispositivo de acuerdo a la función que desempeñan.

Adquisición de datos del GPS.- A través de la conexión del módulo GPS con la placa Arduino Mega, se obtiene los datos necesarios para el funcionamiento del Taxímetro. Esto se logra con la utilización de la comunicación serial (Rx, Tx) entre estos dos dispositivos.

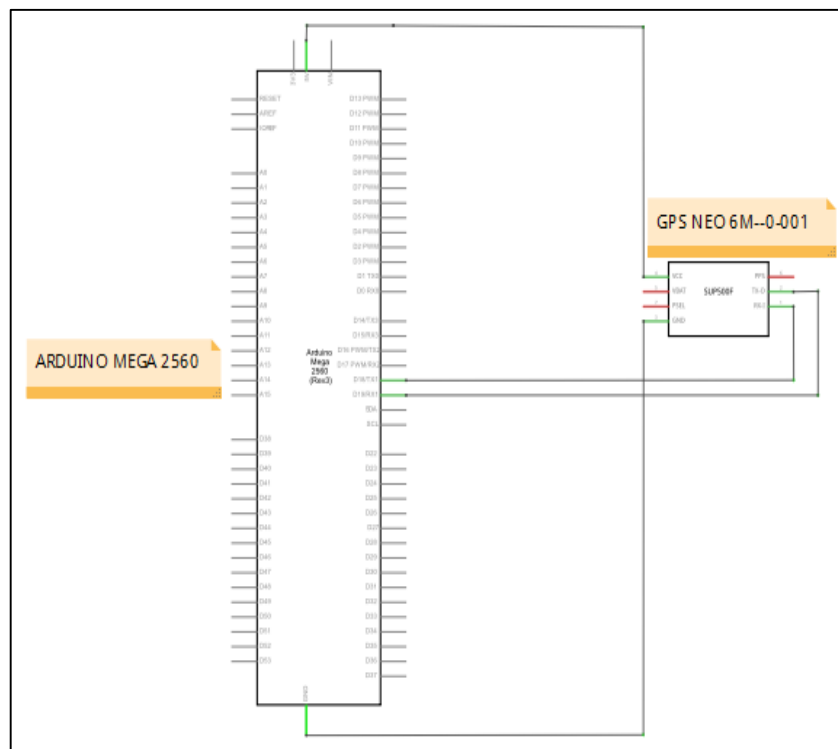


Figura 55. Circuito para adquisición de datos del GPS.
Fuente: (Morales)

Almacenamiento de datos.- Los datos registrados a través del funcionamiento del dispositivo son almacenados en la tarjeta SD. Para ejecutar esta actividad, se realiza la conexión de los pines MISO, MOSI, SCK y CS del módulo SD que viene integrado en la TFT 3.2" con los pines del mismo nombre de la placa Arduino Mega.

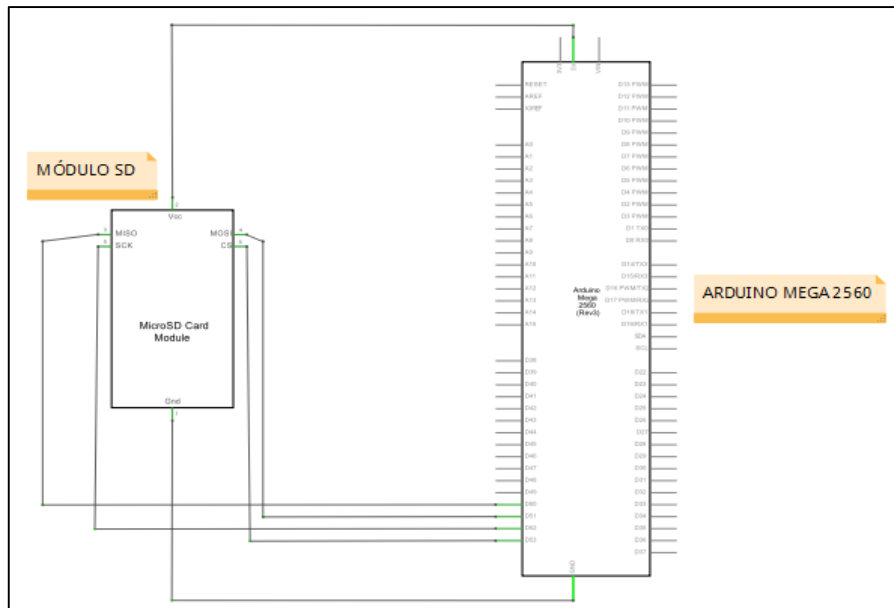


Figura 56. Circuito de almacenamiento de datos.
Fuente: (Morales)

TFT 3.2" y Arduino.- Para el funcionamiento del monitor de la TFT se realiza la conexión de los pines BL CTR, VCC, RD, GND, DB1 ~ DB8, DB10 ~ DB17, RS, WR, LCD_CS, RSET, mientras que para el control táctil, se deben conectar los siguientes pines: DCLK, T_CS, MOSI, MISO, T_PEN.

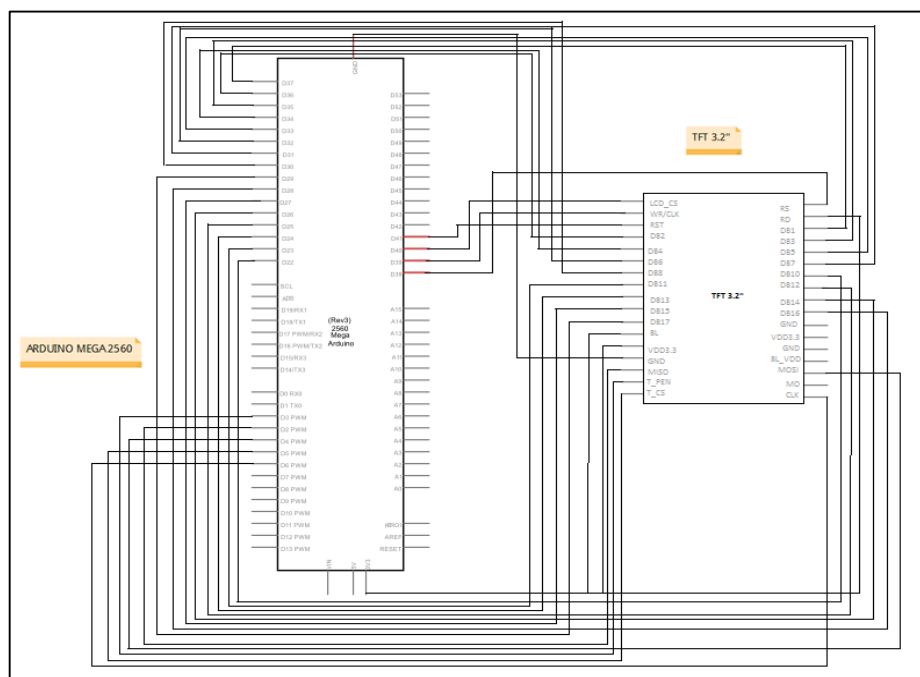


Figura 57. Circuito de conexión entre TFT 3.2" y Arduino.
Fuente: (Morales)

Impresora térmica y Arduino.- La impresora térmica funciona a través de la comunicación serial con la placa Arduino Mega, para lo cual es necesario la conexión de los pines Rx y Tx.

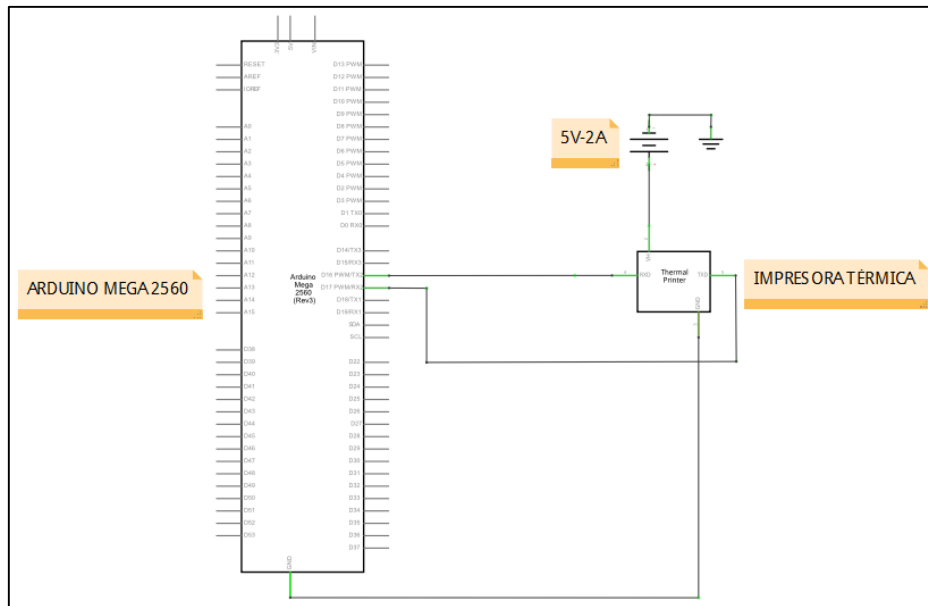


Figura 58. Circuito de conexión entre la impresora térmica y Arduino.
Fuente: (Morales)

El diagrama electrónico completo del dispositivo se indica en el anexo 1.

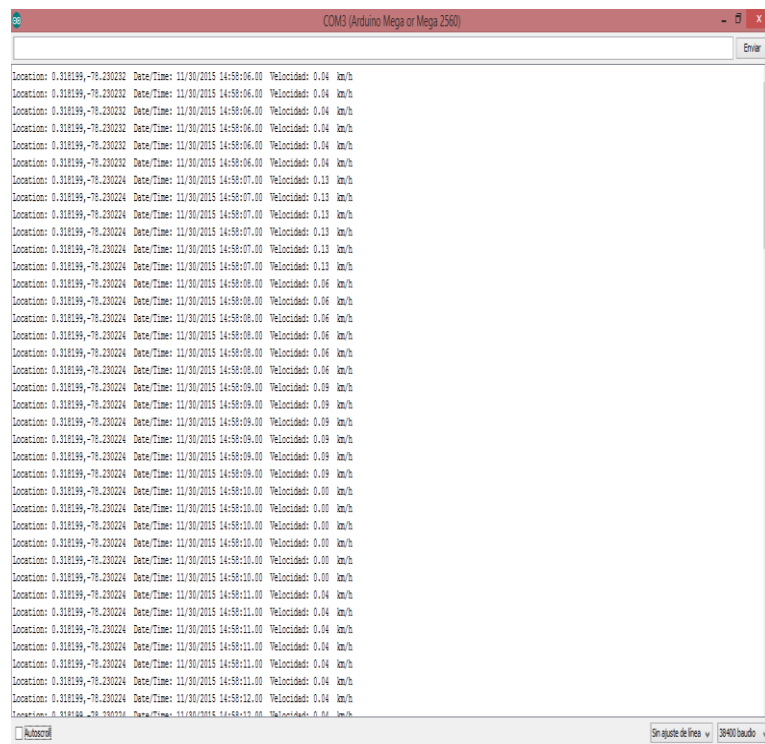
CAPÍTULO V

5. PRUEBAS DE FUNCIONAMIENTO Y ANÁLISIS DE DATOS.

5.1 Prueba de comunicación del GPS.

Realizadas las conexiones entre el módulo GPS y el Arduino Mega se realiza las pruebas de recepción de los datos necesarios para el correcto funcionamiento del taxímetro, puesto que, a través del GPS se obtiene el tiempo, hora, velocidad, distancia recorrida, que son parámetros importantes para establecer las tarifas de funcionamiento del dispositivo. Estos valores son visualizados a través de la comunicación serial entre el módulo GPS y el Arduino Mega 2560.

En la figura 58 se observan los datos recibidos por el GPS.





```
COM3 (Arduino Mega or Mega 2560)
Location: 0.318189,-78.230232 Date/Time: 11/30/2015 14:58:04.00 Velocidad: 0.04 km/h
Location: 0.318189,-78.230232 Date/Time: 11/30/2015 14:58:04.00 Velocidad: 0.04 km/h
Location: 0.318189,-78.230232 Date/Time: 11/30/2015 14:58:04.00 Velocidad: 0.04 km/h
Location: 0.318189,-78.230232 Date/Time: 11/30/2015 14:58:04.00 Velocidad: 0.04 km/h
Location: 0.318189,-78.230232 Date/Time: 11/30/2015 14:58:04.00 Velocidad: 0.04 km/h
Location: 0.318189,-78.230232 Date/Time: 11/30/2015 14:58:04.00 Velocidad: 0.04 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:07.00 Velocidad: 0.13 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:07.00 Velocidad: 0.13 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:07.00 Velocidad: 0.13 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:07.00 Velocidad: 0.13 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:07.00 Velocidad: 0.13 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:07.00 Velocidad: 0.13 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:08.00 Velocidad: 0.06 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:08.00 Velocidad: 0.06 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:08.00 Velocidad: 0.06 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:08.00 Velocidad: 0.06 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:08.00 Velocidad: 0.06 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:08.00 Velocidad: 0.06 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:08.00 Velocidad: 0.06 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:08.00 Velocidad: 0.06 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:08.00 Velocidad: 0.06 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:08.00 Velocidad: 0.06 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:09.00 Velocidad: 0.09 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:09.00 Velocidad: 0.09 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:09.00 Velocidad: 0.09 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:09.00 Velocidad: 0.09 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:09.00 Velocidad: 0.09 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:09.00 Velocidad: 0.09 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:10.00 Velocidad: 0.00 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:10.00 Velocidad: 0.00 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:10.00 Velocidad: 0.00 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:10.00 Velocidad: 0.00 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:10.00 Velocidad: 0.00 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:10.00 Velocidad: 0.00 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:10.00 Velocidad: 0.00 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:10.00 Velocidad: 0.00 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:10.00 Velocidad: 0.00 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:10.00 Velocidad: 0.00 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:10.00 Velocidad: 0.00 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:11.00 Velocidad: 0.04 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:11.00 Velocidad: 0.04 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:11.00 Velocidad: 0.04 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:11.00 Velocidad: 0.04 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:11.00 Velocidad: 0.04 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:11.00 Velocidad: 0.04 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:11.00 Velocidad: 0.04 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:11.00 Velocidad: 0.04 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:12.00 Velocidad: 0.04 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:12.00 Velocidad: 0.04 km/h
Location: 0.318189,-78.230224 Date/Time: 11/30/2015 14:58:12.00 Velocidad: 0.04 km/h
```

Figura 59. Datos del GPS a través del monitor serial.
Fuente: (Morales)

5.2 Navegación por el menú de control.

El taxímetro cuenta con una TFT táctil de 3.2 pulgadas, a través de la cual se observa un menú de control representado con cuatro iconos, que son los encargados de mostrar cada una de las ventanas de funcionamiento del dispositivo. La función de cada uno de ellos se indica en la tabla 15.

Tabla 15. Función de los iconos del menú.

Icono	Función
	Muestra la segunda pantalla, en la cual se visualiza la fecha y hora actual.
	Muestra la ventana con el funcionamiento del taxímetro, se observa el tipo de tarifa, lugar inicio, tiempo de espera y kilómetros recorridos.
	Muestra la ventana al dar fin al taxímetro y se observan datos como el lugar final, subtotal y total a cancelar de la carrera.
	A través de este botón se muestra la pantalla principal y todas las variables toman sus valores iniciales.

Fuente: (Morales)

Durante el funcionamiento del dispositivo se distinguen cuatro ventanas que indican las siguientes actividades:

- Portada.
- Reloj.
- Inicio del funcionamiento del taxímetro.
- Fin del funcionamiento del taxímetro.

5.2.1 Portada.

En esta ventana se indica una imagen de fondo y el título del trabajo de grado: “Taxímetro con almacenamiento de viajes mediante GPS”, mientras que en la parte inferior se encuentra el menú de control con los iconos de Inicio, Reloj y Fin, como se indica en la figura 59.



Figura 60. Ventana principal.
Fuente: (Morales)

5.2.2 Reloj.

Al presionar el icono de Reloj se muestra la segunda ventana, en la cual se observa un mensaje de bienvenida, la fecha y hora actual, que son datos obtenidos directamente del GPS. Además se indica en el menú el icono Home, el cual permite mostrar la ventana principal.



Figura 61. Ventana Reloj.
Fuente: (Morales)

5.2.3 Inicio.

Al presionar el icono Inicio se muestra la ventana con los datos de funcionamiento del taxímetro. En ella se observa el tipo de tarifa, el lugar de inicio de la carrera, el valor de arranque inicial, el tiempo de espera y los kilómetros recorridos.



Figura 62. Ventana de inicio de taxímetro.
Fuente: (Morales)

5.2.4 Fin.

Al presionar el icono Fin se muestra la última ventana, en la cual se indica el lugar de llegada, el subtotal y el total de la carrera, para finalmente proceder a imprimir automáticamente el recibo.



Figura 63. Ventana de fin de taxímetro.
Fuente: (Morales)

5.3 Prueba de funcionamiento del taxímetro.

El taxímetro tiene dos tarifas de funcionamiento: Tarifa Diurna y Tarifa Nocturna. En la pantalla se indica la respectiva tarifa en la que se encuentra en funcionamiento el dispositivo.

En la tarifa diurna se indica el lugar de inicio, el arranque de la tarifa que es de 0,40 centavos, el tiempo de espera en segundos y los kilómetros recorridos, mientras que en la tarifa nocturna el único dato que cambia es el valor del arranque de la tarifa que es de 0,44 centavos.



Figura 64. Funcionamiento en la tarifa diurna.
Fuente: (Morales)



Figura 65. Funcionamiento en la tarifa nocturna.
Fuente: (Morales)

Además, a través del funcionamiento en las dos tarifas se observa la simulación de un led rojo que se prende y se apaga al momento que el taxímetro comienza a cobrar por tiempo de espera, como se indica en la siguiente figura.



Figura 66. Aumento de tarifa por tiempo de espera.
Fuente: (Morales)

De igual manera cuando el taxímetro cobra por distancia recorrida se observa la simulación de un led, en este caso verde, que se prende y se apaga como se indica en la figura 66.



Figura 67. Aumento de tarifa por distancia recorrida.
Fuente: (Morales)

En la figura 67 se muestra la última ventana que se despliega al presionar el botón fin, en la cual se indican dos valores, un subtotal y un total. Cuando el subtotal es menor al valor mínimo de la tarifa correspondiente, se indica como total el valor de la carrera mínima.



Figura 68. Subtotal menor al valor de la carrera mínima.
Fuente: (Morales)

Pero cuando el subtotal sobrepasa el valor de la carrera mínima en la tarifa correspondiente, el valor del total de la carrera será el mismo que el subtotal, como se observa en la figura 68.



Figura 69. Subtotal mayor al valor de la carrera mínima.
Fuente: (Morales)

5.4 Prueba de impresión.

En la ventana de fin de taxímetro se observa un mensaje de IMPRIMIENDO, y seguidamente se genera el recibo de la carrera a través de la impresora térmica, tal como se indica en la figura 69.



Figura 70. Impresión del recibo.
Fuente: (Morales)

En el recibo impreso se indica información importante como la compañía a la que pertenece el taxi, datos del conductor, lugar de inicio y llegada, fecha y hora de emisión, así como también el desglose del valor a cancelar en la tarifa correspondiente, tal como se indica en la siguiente figura.



Figura 71. Recibo impreso.
Fuente: (Morales)

5.5 Prueba de almacenamiento de datos.

Durante el funcionamiento del taxímetro se guardan dos archivos en la tarjeta de memoria incorporada en el dispositivo, uno en formato Excel y el otro en formato de Texto. En archivo Excel se almacenan los datos de los diferentes viajes realizados por el vehículo, entre ellos tenemos: Placa, Compañía, Tipo, Modo, Partida, Posición Inicial, Fecha Inicial, Hora Inicial, Llegada, Posición Final, Fecha Final, Hora Final, Distancia (Km), Tiempo de Espera (seg), Tiempo de Viaje (min), Subtotal (\$) y Total (\$), como se indica en la figura 71.

PLACA	COMPAÑIA	TIPO	MODOS	PARTIDA	POSICION INICIAL	FECHA INICIAL	HORA INICIAL	LLEGADA	POSICION FINAL	FECHA FINAL	HORA FINAL	DISTANCIA (km)	T ESPERA (seg)	T VIAJE (min)	T VALOR (\$)	SUBTOTAL (\$)	TOTAL (\$)
BA-9757	BRELAGUIRESA	EJECUTIVO	Da	LAFI OROA	0.32945-78.10430	27/11/2015	14:54:49	RECINDEL DE LA MANA	0.34839-78.12373	27/11/2015	14:58:22	1.55	93.25	4.54	1.13	1.25	1.25
BA-9757	BRELAGUIRESA	EJECUTIVO	Da	CENTRO MERCADO	0.34839-78.10430	27/11/2015	14:45:09	LA ESPERANZA	0.34443-78.12162	27/11/2015	14:53:43	5.95	79.50	14.95	2.73	2.73	2.73
BA-9757	BRELAGUIRESA	EJECUTIVO	Da	LA ESPERANZA	0.34443-78.12162	27/11/2015	15:03:25	LA ESPERANZA	0.34443-78.12162	27/11/2015	15:03:25	6.95	79.50	14.95	2.73	2.73	2.73
BA-9757	BRELAGUIRESA	EJECUTIVO	Da	CARAMQUI	0.32278-78.18786	27/11/2015	15:08:32	LA VICTORIA	0.34257-78.19503	27/11/2015	15:14:38	2.33	69.00	6.08	1.42	1.42	1.42
BA-9757	BRELAGUIRESA	EJECUTIVO	Da	LA VICTORIA	0.34837-78.10776	27/11/2015	15:10:00	CENTRO MERCADO	0.35049-78.11626	27/11/2015	15:23:06	1.55	83.25	5.07	1.13	1.25	1.25
BA-9757	BRELAGUIRESA	EJECUTIVO	Da	CENTRO MERCADO	0.34839-78.10782	27/11/2015	15:21:44	CENTRO MERCADO	0.34262-78.11692	27/11/2015	15:24:23	1.15	85.00	7.61	1.60	1.60	1.60
BA-9757	BRELAGUIRESA	EJECUTIVO	Da	LOS CEBOS	0.34245-78.12093	27/11/2015	15:38:44	TERMINAL TERRESTRE	0.34336-78.12583	27/11/2015	15:38:43	0.50	44.25	2.08	0.66	1.25	1.25
BA-9757	BRELAGUIRESA	EJECUTIVO	Da	TERMINAL TERRESTRE	0.34336-78.12093	27/11/2015	15:43:12	RECINDEL DE LA MANA	0.34762-78.13173	27/11/2015	15:43:18	1.15	49.50	3.05	0.91	1.35	1.35
BA-9757	BRELAGUIRESA	EJECUTIVO	Da	HOSPITAL SAN VICENTE DE	0.35242-78.12161	27/11/2015	15:47:51	CENTRO MERCADO	0.34833-78.12162	27/11/2015	15:52:21	1.02	110.00	4.48	0.96	1.25	1.25
BA-9757	BRELAGUIRESA	EJECUTIVO	Da	CENTRO MERCADO	0.34730-78.12217	27/11/2015	15:53:00	TERMINAL TERRESTRE	0.34493-78.12330	27/11/2015	15:56:50	0.98	81.50	3.80	0.63	1.25	1.25

Figura 72. Datos almacenados en el archivo Excel.
Fuente: (Morales)

En cambio en el archivo en formato de Texto se almacenan las coordenadas que son tomadas cada 100 metros de desplazamiento del vehículo, las cuales son utilizadas para graficar las rutas seguidas en las diferentes carreras realizadas por el taxi. Estos datos se observan en la figura 72.

```

SGPRNC,192558.00,A,0020,39745,N,07888,86397,W,34,979,85,07,271115SGPRNC,192552.00,A,0020,39902,N,07888,86466,W,34,552,86,03,271115,,D*76
SGPVTG,86,03,T,,M,34,552,N,63,901,K,D*04
SGRGG,192552.00,0020,39902,N,07888,86466,W,2,12,0,77,2249,4,M,13,5,M,,0000*76
SGRPSA,4,3,46,02,24,25,51,20,18,21,29,12,14,15,1,46,0,77,1,24*02
SGRPSV,4,1,15,02,13,133,40,85,04,074,,10,15,327,44,12,18,150,43*76
SGRPSV,4,2,15,14,27,242,34,15,27,014,44,18,17,327,45,20,25,031,46*71
SGRPSV,4,3,15,21,30,337,41,24,62,092,50,25,20,189,39,29,55,227,44*77
SGRPSV,4,4,15,23,19,090,39,46,66,200,47,51,56,269,50*74
SGRGLL,0020,39902,N,07888,86466,W,192552.00,A,D*78

*76G554.00,A,0020,40041,N,07888,84552,W,34,404,85,99,271115,,D*77
SGPVTG,85,99,T,,M,34,404,N,63,883,K,D*00
SGRGG,192554.00,0020,40041,N,07888,84552,W,2,12,0,77,2250,6,M,13,5,M,,0000*78
SGRPSA,4,3,46,02,24,25,51,20,18,21,29,12,14,15,1,46,0,77,1,24*02
SGRPSV,4,1,16,02,13,133,40,85,04,074,,09,,31,10,15,327,44*44
SGRPSV,4,2,16,12,18,150,44,14,27,242,33,15,27,014,43,18,17,327,45*79
SGRPSV,4,3,15,20,25,031,47,21,30,337,40,24,62,092,50,25,20,189,39*78
SGRPSV,4,4,16,29,55,227,43,33,19,090,40,46,66,260,47,51,56,269,50*7A
SGRGLL,0020,40041,N,07888,84552,W,192554.00,A,D*7A

SGPRNC,192556.00,A,0020,40183,N,07888,82723,W,31,691,85,45,271115,,D*73
SGPVTG,85,45,T,,M,31,691,N,58,691,K,D*08
SGRGG,192556.00,0020,40183,N,07888,82723,M,2,12,0,77,2250,6,M,13,5,M,,0000*71
SGRPSA,4,3,46,02,24,25,51,20,18,21,29,12,14,15,1,46,0,77,1,24*02
SGRPSV,4,1,15,02,13,133,40,85,04,074,,10,15,327,44,12,18,150,42*74
SGRPSV,4,2,15,14,27,242,34,15,27,014,44,18,17,327,44,20,25,031,47*71
SGRPSV,4,3,15,21,30,337,39,46,66,200,46,51,56,269,50*74
SGRPSV,4,4,15,23,19,090,39,46,66,200,46,51,56,269,50*74
SGRGLL,0020,40183,N,07888,82723,W,31,691,85,45,271115,,D*73
SGPRNC,192558.00,A,0020,40312,N,07888,81214,W,24,272,85,02,271115,,D*78
SGPVTG,85,02,T,,M,24,272,N,44,952,K,D*08
SGRGG,192558.00,0020,40312,N,07888,81214,W,2,12,0,77,2251,2,M,13,5,M,,0000*72
SGRPSA,4,3,46,02,24,25,51,20,18,21,29,12,14,15,1,46,0,77,1,24*02
SGRPSV,4,1,15,02,13,133,40,85,04,074,,10,15,327,44,12,18,150,43*76
SGRPSV,4,2,15,14,27,242,35,15,27,014,43,18,17,327,42,20,25,031,48*7E
SGRPSV,4,3,15,21,30,337,39,24,62,092,49,25,20,189,38,29,55,227,41*74
SGRPSV,4,4,15,23,19,090,39,46,66,200,46,51,56,269,49*45
    
```

Figura 73. Datos almacenados en el archivo de Texto.
Fuente: (Morales)

5.6 Prueba del software de análisis de datos en LabVIEW.

Para comprobar el correcto funcionamiento del software se retira la tarjeta de memoria del dispositivo para insertarla en la PC, de esta manera se procede a la descarga de los datos de los dos archivos almacenados al programa LabVIEW. Para continuar con la prueba se procede a abrir el programa y ponerlo en modo de ejecución, en el que inicialmente se visualiza la portada como se indica en la siguiente figura.



Figura 74. Portada del programa en LabVIEW.
Fuente: (Morales)

En la portada se observa cinco opciones en la parte superior que son: HOME, DATOS, ESTADÍSTICA ORIGEN Y DESTINO, ESTADÍSTICA GENERAL, REPRESENTACIÓN GRÁFICA, las cuales tienen funciones específicas para el análisis de datos.

5.6.1 Importación de datos.

Al dirigirse a la ventana de DATOS del programa se visualizan dos botones denominados IMPORTAR DATOS y BORRAR DATOS. Al presionar el primero se observa que los datos de los diferentes viajes realizados se encuentran separados y organizados correctamente en la respectiva tabla que se visualiza en las figuras 74 y 75.

HOME DATOS ESTADISTICA ORIGEN-DESTINO ESTADISTICA GENERAL REPRESENTACION GRAFICA

IMPORTAR DATOS DATOS DE LOS VIAJES REALIZADOS BORRAR DATOS

PLACA	COOP/COMPANIA	TIPO	MODOS	PARTIDA	POSICION INICIAL	FECHA INICIAL	HORA INICIAL	LLEGADA	POSICION FINAL	FECHA FINAL	HORA FINAL	DIST
IBA-9757	BRISLAGTURIS S.A.	EJECUTIVO	Dia	LA FLORIDA	0.339842;-78.148223	27/11/2015	14:25:49	REDONDEL DE LA MADRE	0.346909;-78.132873	27/11/2015	14:30:22	
IBA-9757	BRISLAGTURIS S.A.	EJECUTIVO	Dia	CENTRO, MERCADO	0.349883;-78.119430	27/11/2015	14:45:09	LA ESPERANZA	0.304425;-78.121627	27/11/2015	14:59:43	
IBA-9757	BRISLAGTURIS S.A.	EJECUTIVO	Dia	LA ESPERANZA	0.304425;-78.121566	27/11/2015	15:00:25	LA ESPERANZA	0.304425;-78.121566	27/11/2015	15:00:25	
IBA-9757	BRISLAGTURIS S.A.	EJECUTIVO	Dia	CARANQUI	0.322782;-78.118766	27/11/2015	15:08:32	LA VICTORIA	0.342570;-78.111503	27/11/2015	15:14:38	
IBA-9757	BRISLAGTURIS S.A.	EJECUTIVO	Dia	LA VICTORIA	0.346537;-78.107376	27/11/2015	15:18:00	CENTRO, MERCADO	0.352489;-78.116325	27/11/2015	15:23:06	
IBA-9757	BRISLAGTURIS S.A.	EJECUTIVO	Dia	CENTRO, MERCADO	0.348969;-78.117729	27/11/2015	15:27:44	CENTRO, MERCADO	0.342592;-78.119552	27/11/2015	15:35:23	
IBA-9757	BRISLAGTURIS S.A.	EJECUTIVO	Dia	LOS CEBOS	0.342645;-78.120983	27/11/2015	15:36:44	TERMINAL TERRESTRE	0.343356;-78.123631	27/11/2015	15:38:49	
IBA-9757	BRISLAGTURIS S.A.	EJECUTIVO	Dia	TERMINAL TERRESTRE	0.343056;-78.129508	27/11/2015	15:40:13	REDONDEL DE LA MADRE	0.347628;-78.131973	27/11/2015	15:43:18	
IBA-9757	BRISLAGTURIS S.A.	EJECUTIVO	Dia	HOSPITAL SAN VICENTE DE PAULI	0.352342;-78.128761	27/11/2015	15:47:51	CENTRO, MERCADO	0.348233;-78.121612	27/11/2015	15:52:21	
IBA-9757	BRISLAGTURIS S.A.	EJECUTIVO	Dia	CENTRO, MERCADO	0.347301;-78.121777	27/11/2015	15:53:00	TERMINAL TERRESTRE	0.344970;-78.125938	27/11/2015	15:56:50	

Figura 75. Funcionamiento de la ventana datos (parte 1).
Fuente: (Morales)

ESTADISTICA GENERAL REPRESENTACION GRAFICA

DATOS DE LOS VIAJES REALIZADOS BORRAR DATOS

PARTIDA	POSICION INICIAL	FECHA INICIAL	HORA INICIAL	LLEGADA	POSICION FINAL	FECHA FINAL	HORA FINAL	DISTANCIA(Km)	T. ESPERA(seg)	T. VIAJE(min)	SUBTOTAL(S)	TOTAL(S)
A FLORIDA	0.339842;-78.148223	27/11/2015	14:25:49	REDONDEL DE LA MADRE	0.346909;-78.132873	27/11/2015	14:30:22	1.55	59.25	4.54	1.10	1.25
RO, MERCADO	0.349883;-78.119430	27/11/2015	14:45:09	LA ESPERANZA	0.304425;-78.121627	27/11/2015	14:59:43	5.55	79.50	14.55	2.73	2.73
ESPERANZA	0.304425;-78.121566	27/11/2015	15:00:25	LA ESPERANZA	0.304425;-78.121566	27/11/2015	15:00:25	5.55	79.50	14.57	2.73	2.73
CARANQUI	0.322782;-78.118766	27/11/2015	15:08:32	LA VICTORIA	0.342570;-78.111503	27/11/2015	15:14:38	2.33	69.00	6.09	1.42	1.42
A VICTORIA	0.346537;-78.107376	27/11/2015	15:18:00	CENTRO, MERCADO	0.352489;-78.116325	27/11/2015	15:23:06	1.55	83.25	5.07	1.13	1.25
RO, MERCADO	0.348969;-78.117729	27/11/2015	15:27:44	CENTRO, MERCADO	0.342592;-78.119552	27/11/2015	15:35:23	1.15	162.00	7.61	1.08	1.25
OS CEBOS	0.342645;-78.120983	27/11/2015	15:36:44	TERMINAL TERRESTRE	0.343356;-78.123631	27/11/2015	15:38:49	0.50	44.25	2.08	0.66	1.25
INAL TERRESTRE	0.343056;-78.129508	27/11/2015	15:40:13	REDONDEL DE LA MADRE	0.347628;-78.131973	27/11/2015	15:43:18	1.15	40.50	3.05	0.91	1.25
HAN VICENTE DE PAULI	0.352342;-78.128761	27/11/2015	15:47:51	CENTRO, MERCADO	0.348233;-78.121612	27/11/2015	15:52:21	1.02	111.00	4.48	0.96	1.25
RO, MERCADO	0.347301;-78.121777	27/11/2015	15:53:00	TERMINAL TERRESTRE	0.344970;-78.125938	27/11/2015	15:56:50	0.88	61.50	3.80	0.83	1.25

Figura 76. Funcionamiento de la ventana datos (parte 2).
Fuente: (Morales)

El segundo botón (BORRAR DATOS), permite limpiar todos los datos almacenados en los archivos de la tarjeta de memoria, para así empezar a almacenar los datos de los nuevos viajes realizados por el vehículo.

5.6.2 Estadística origen y destino.

En la ventana ESTADÍSTICA ORIGEN-DESTINO se presenta el botón DATOS, el cual al ser presionado se indica el número de viajes realizados desde y hacia los lugares definidos en la sectorización. Estos datos son visualizados en una matriz que indica en su parte derecha los lugares de inicio de la carrera, y en la parte superior los lugares de llegada, siendo de esta manera muy fácil la visualización desde y hacia qué lugar se realizaron la mayor cantidad de viajes. Así mismo en la parte inferior de la matriz se puede observar el total de las carreras realizadas y registradas por el taxímetro. Esto se indica en las figuras 76 y 77.

Figura 77. Ventana estadística origen-destino (parte 1).

Fuente: (Morales)

Figura 78. Ventana estadística origen-destino (parte 2).

Fuente: (Morales)

5.6.3 Estadística general.

En la página ESTADÍSTICA GENERAL se muestran datos generales de las carreras realizadas durante un determinado rango de fechas que son limitadas a través de un control de inicio y fin. Estos datos son registrados en una tabla dividida en tres columnas de DÍA, NOCHE Y TOTAL, como se indica en la figura 78.

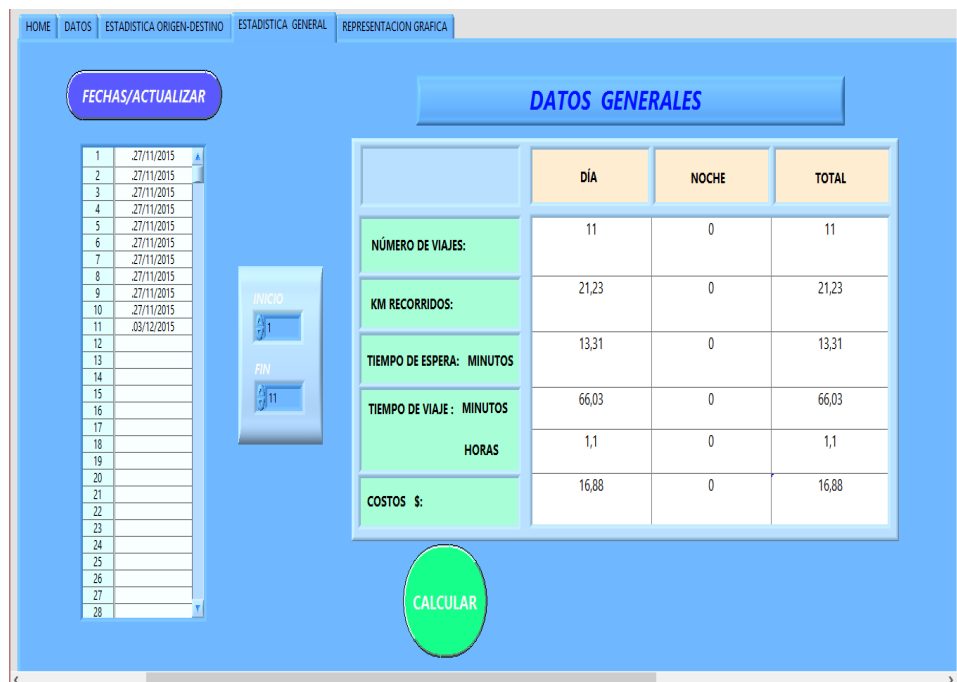


Figura 79. Ventana estadística general.
Fuente: (Morales)

Los datos visualizados en la tabla de la figura 78 son del número de viajes, kilómetros recorridos, tiempo de espera, tiempo de viaje y costos generados durante el rango establecido en las diferentes tarifas de día y noche, así también se indica un total de cada uno de estos parámetros.

5.6.4 Representación gráfica.

A través de esta ventana se ejecutan dos programas que sirven para graficar los lugares de inicio y fin de carrera, así como también las rutas seguidas en los diferentes viajes. Estos programas son abiertos al presionar los respectivos botones que son visualizados en la figura 79.



Figura 80. Ventana representación gráfica.
Fuente: (Morales)

Visualización de lugares de inicio y llegada en POWER BI.- Al presionar el botón POWER BI de la venta de REPRESENTACIÓN GRÁFICA se ejecuta este programa, a través del cual se grafican los puntos de inicio y fin de carrera, que son visualizados en una sola ventana, pero en mapas diferentes. Los lugares de inicio son representados con puntos de color verde, mientras que los lugares de llegada con puntos de color rojo, como se observa en la figura 80.

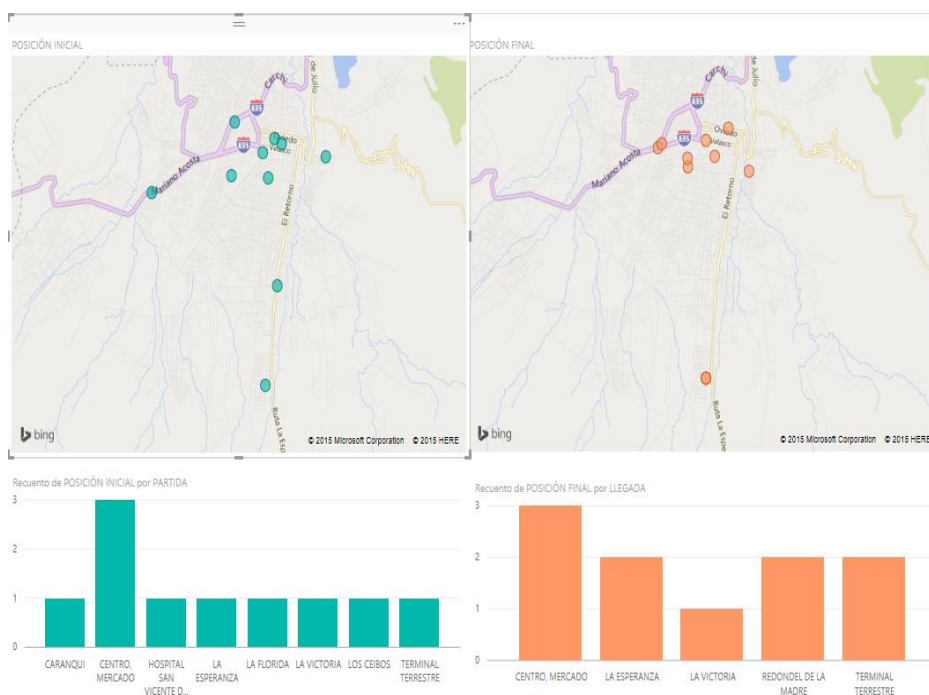


Figura 81. Ventana del programa Power BI.
Fuente: (Morales)

Además, debajo de cada uno de los mapas se genera un gráfico de barras, a través del cual se identifica los lugares con mayor número de viajes al inicio y fin de las carreras realizadas.

Visualización de rutas seguidas en U-CENTER.- Con la ejecución del programa U-center se grafica las rutas seguidas por el vehículo en cada una de las carreras realizadas, que se logra gracias al archivo de texto almacenado en la tarjeta de memoria, que al ser abierto por este programa se grafica todos los puntos almacenados a través del funcionamiento del taxímetro durante el desplazamiento del vehículo. Esto se indica en la figura 81.

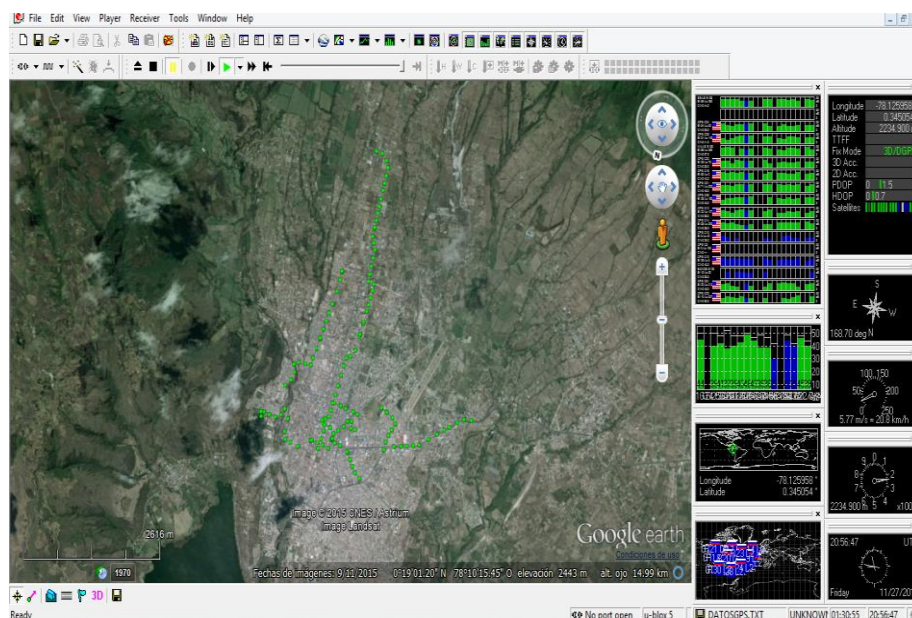


Figura 82. Ventana del programa U-center.
Fuente: (Morales)

5.7 Implementación del dispositivo.

Para la implementación del dispositivo en el vehículo se toma el voltaje de 12 voltios de la batería a través del encendedor. Este voltaje es enviado a una fuente de alimentación del taxímetro, la cual está construida a través de un inversor de corriente y dos reguladores de voltaje, de 5 voltios y 2 amperios para la impresora térmica, mientras que para el dispositivo taxímetro se utiliza un regulador de 5 voltios y 0.5 amperios, necesarios para su correcto funcionamiento.

La fuente de alimentación consta de una entrada de voltaje, dos salidas y un porta fusibles para proteger los elementos electrónicos, tal como se indica en la figura 82.



Figura 83. Fuente de alimentación.
Fuente: (Morales)

En la figura 83 se observa la impresora térmica y el taxímetro instalados en el vehículo en lugares visibles para el usuario del taxi como también para el conductor.



Figura 84. Dispositivos instalados en el vehículo.
Fuente: (Morales)

En el anexo 2 se presenta el diagrama de conexión de los diferentes elementos del dispositivo.

CAPÍTULO VI

6. CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones.

- El taxímetro diseñado en Arduino Mega realiza todas las funciones de un taxímetro convencional, permite facturar cada viaje e imprimir sus respectivos comprobantes, respetando las tarifas vigentes en el respectivo horario de funcionamiento de acuerdo a lo establecido por la autoridad competente.
- Las opciones adicionales que brinda el prototipo desarrollado, son almacenaje de datos de kilometraje, distancia, tiempos de espera, ubicación de inicio y fin de carrera, para su posterior adquisición hacia una PC y procesamiento.
- La información almacenada con el uso del dispositivo, puede ser importada a través de una MicroSD hacia una PC fácilmente, que a través de la utilización del software desarrollado aportarán información valiosa para las instituciones encargadas de planificar y realizar proyectos en lo que se refiere al sistema de transporte en la ciudad.
- El Sistema de Posicionamiento Global acoplado al taxímetro permite guardar una posición cada 150 metros durante el desplazamiento del vehículo, datos que al ser subidos al programa U-center grafican las rutas seguidas en los diferentes viajes realizados.
- El dispositivo GPS trabaja con hasta 20 satélites a la vez, por lo que proporciona una precisión de hasta 5 metros, obteniendo de esta manera gran exactitud al momento de definir los lugares de inicio y llegada de carrera.

- El costo final del prototipo de taxímetro es de \$250 dólares americanos, siendo un precio muy competitivo en el mercado, respecto al valor de otros taxímetros que actualmente se comercializan en el país.

6.2 Recomendaciones.

- Una vez realizadas las pruebas de funcionamiento del dispositivo con resultados satisfactorios, se recomienda su uso para obtener información que permita a la autoridad de Tránsito local, reestructurar las rutas y frecuencias del sistema de transporte público en la ciudad.
- Los datos recibidos por el dispositivo GPS no se ven afectados por las condiciones meteorológicas, pero si es recomendable realizar pruebas de funcionamiento en túneles y lugares con edificios altos, debido a que estos pueden bloquear la señal enviada por los satélites.
- Se recomienda el uso del software de programación de Arduino IDE, ya que tiene un lenguaje de programación amigable, además, incluye los drivers de todas las tarjetas disponibles, lo que facilita la carga de códigos desde el computador.
- La placa Arduino Mega provee de varios tipos de entrada y salida de datos, permitiendo obtener y enviar información de múltiples formas, proporcionando alta eficiencia y costos reducidos, siendo recomendable su utilización para diferentes proyectos.
- Es recomendable el uso de la impresora térmica, ya que no necesita de tinta para la impresión de los recibos por lo que disminuye costos en la operación del dispositivo.
- Estudiar la posibilidad de incorporar al taxímetro un elemento electrónico que permita la adquisición de datos en tiempo real, para conocer la ubicación del vehículo en la ciudad y de esta manera brindar seguridad al conductor y a los usuarios del taxi.

BIBLIOGRAFÍA

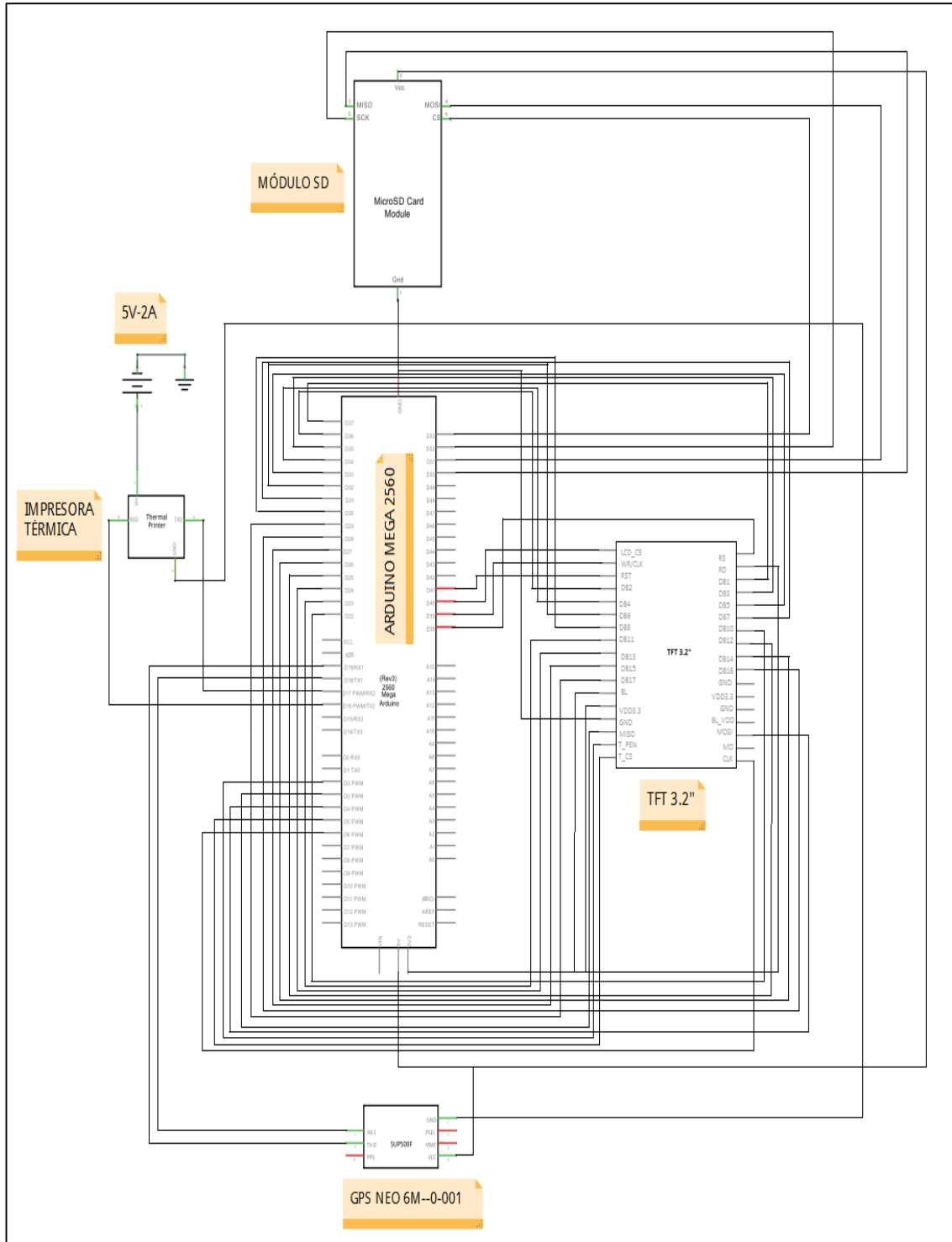
- *Adafruit*. (18 de Diciembre de 2015). Obtenido de <https://learn.adafruit.com>
- *Arduino*. (2015). Obtenido de <https://www.arduino.cc>
- Arrué, J. J. (2009). *Sobre la movilidad urbana*. Obtenido de http://www.ciudadnuestra.org/index.php?fp_cont=1505
- Artero, Ó. T. (2013). *Arduino : curso práctico de formación*. España: RC Libros.
- Benavides, G., & Rodas, S. (2009). *Protocolo de Asistencia Integral a Personas en Movilidad*. Quito, Ecuador: Inédito.
- Bress, T. (2013). *Effective LabVIEW Programming*. NTS Press.
- *Bust a Tech*. (25 de Diciembre de 2014). Obtenido de <http://www.bustatech.com>
- Carles Olmedillas, J. (2013). *Introducción a los sistemas de navegación por satélite*. Barcelona: Editorial UOC.
- Cayssials, R. (2014). *Sistemas embebidos FPGA*. España: Marcombo, S.A.
- *FM & W*. (2011). Obtenido de <http://www.fmwservicios.com>
- Galeano Atehortúa, G. A. (2009). *Programación de sistemas embebidos en C, teoría y prácticas aplicadas a cualquier microcontrolador*. México: Alfaomega Grupo Editor.
- Gordón Salvatierra, M. S. (2012). *La movilidad sustentable en Quito: una vision desde los mas vulnerables*. Quito: Ediciones AbyaYala.
- Herce Vallejo, M. (2009). *Sobre la MOVILIDAD en la ciudad*. Barcelona: Reverté.
- *Industrias Saludes*. (2015). Obtenido de <http://www.industriassaludes.es>
- INEN. (2013). *Taxímetros, requisitos metrológicos y técnicos*. Obtenido de www.inen.gob.ec
- *ITead Studio*. (25 de Octubre de 2011). Obtenido de <http://www.iteadstudio.com>

- Perea, F. (2015). *Arduino Essentials*. Packt Publishing Ltd.
- Reyes Cortés, F., Cid Monjaraz, J., & Vargas Soto, E. (2013). *Mecatrónica: control y automatización*. México: Alfaomega Grupo Editor.
- Tránsito, A. N. (25 de Mayo de 2015). *Agencia Nacional de Tránsito*. Obtenido de Agencia Nacional de Tránsito.
- Valdés Pérez, F. E., & Pallàs Areny, R. (2007). *Microcontroladores: fundamentos y aplicaciones con PIC*. España: Marcombo.

ANEXOS

ANEXO 1.

DIAGRAMA ELECTRÓNICO COMPLETO DEL DISPOSITIVO.



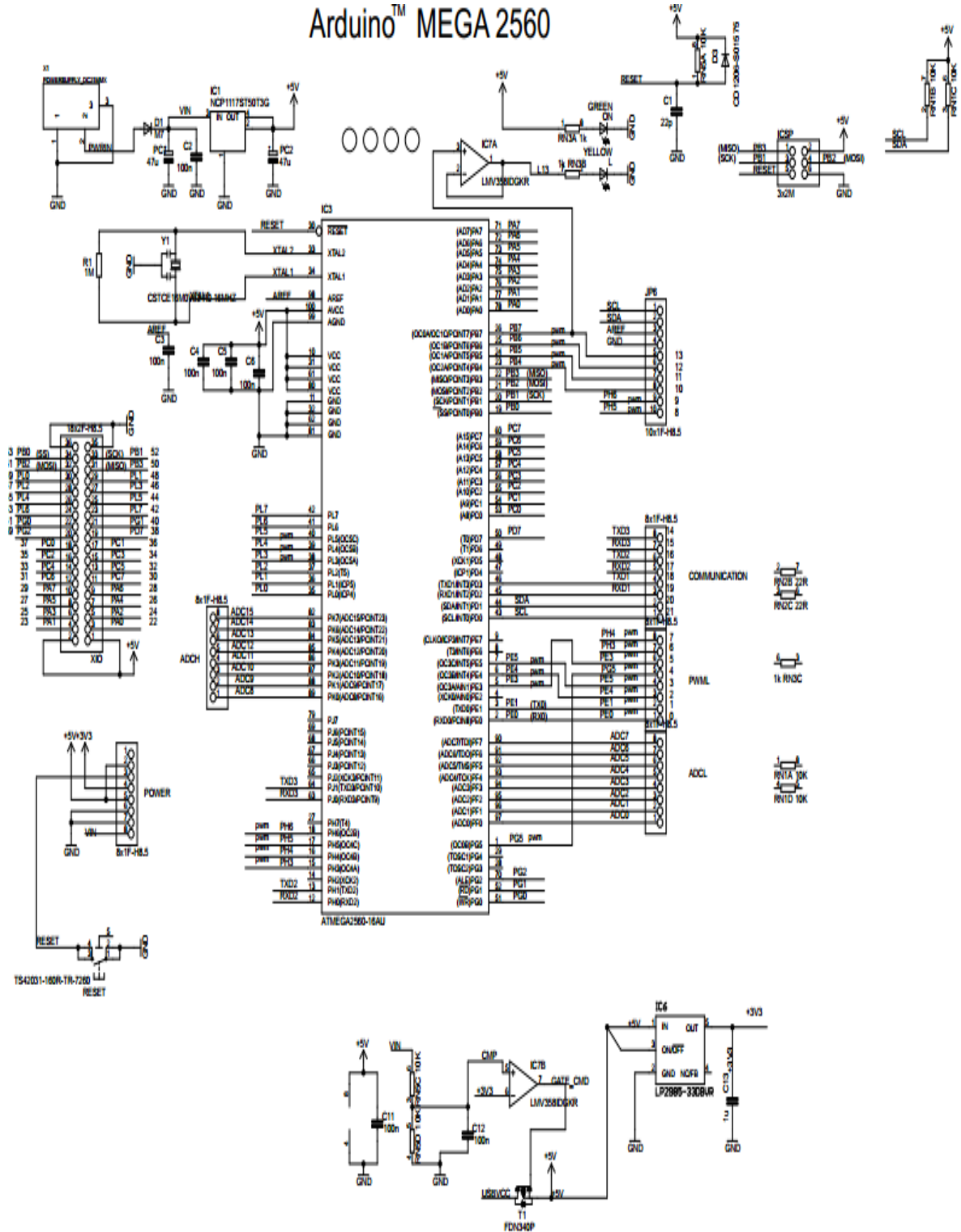
ANEXO 2.

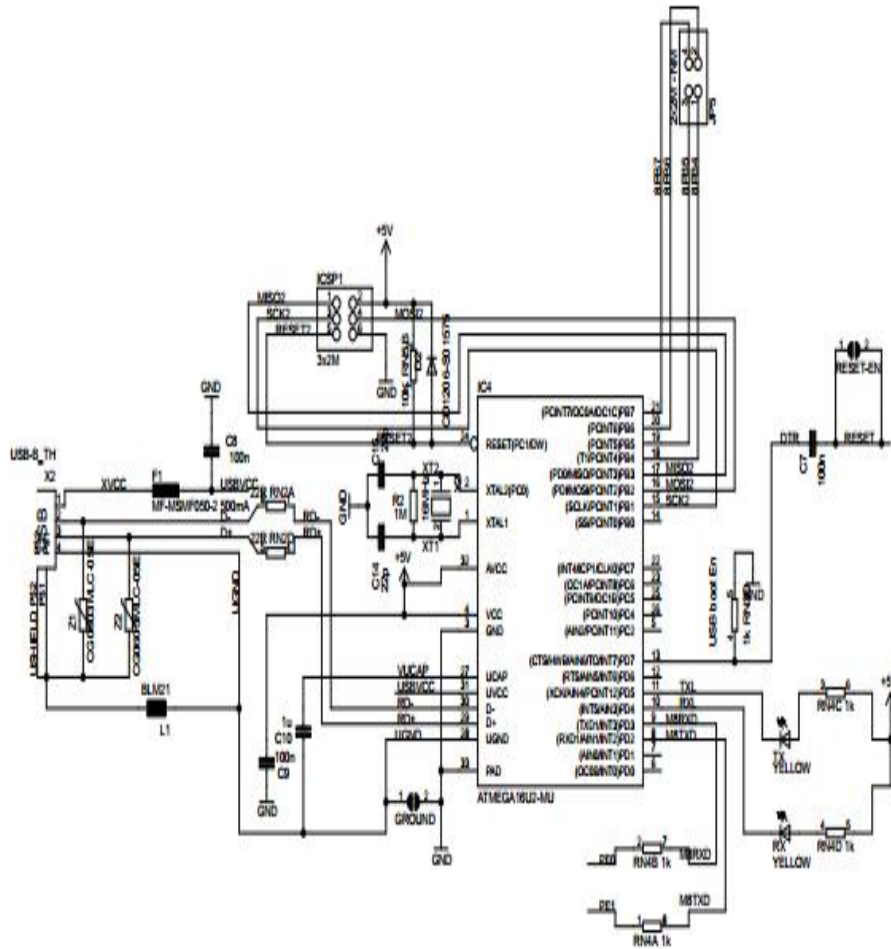
DIAGRAMA DE CONEXIÓN DEL TAXÍMETRO.



ANEXO 3.

DATASHEET ARDUINO MEGA 2560





Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

ARDUINO is a registered trademark.

ANEXO 4.

DATASHEET MÓDULO GPS



NEO-6 • Data Sheet

Operating conditions

All specifications are at an ambient temperature of 25°C.

Parameter	Symbol	Module	Min	Typ	Max	Units	Condition
Power supply voltage	VCC	NEO-6G	1.75	1.8	1.95	V	
		NEO-6QM	2.7	3.0	3.6	V	
		NEO-6RW					
Supply voltage USB	VDDUSB	All	3.0	3.3	3.6	V	
Backup battery voltage	V_BCKP	All	1.4		3.6	V	
Backup battery current	I_BCKP	All		22		µA	V_BCKP = 1.8 V, VCC = 0V
Input pin voltage range	Vin	All	0		VCC	V	
Digital IO Pin Low level input voltage	Vil	All	0		0.2*VCC	V	
Digital IO Pin High level input voltage	Vih	All	0.7*VCC		VCC	V	
Digital IO Pin Low level output voltage	Vol	All			0.4	V	IOL=4mA
Digital IO Pin High level output voltage	Voh	All	VCC-0.4			V	IOL=4mA
USB_DM, USB_DP	VinU	All	Compatible with USB with 22 Ohms series resistance				
VCC_RF voltage	VCC_RF	All		VCC-0.1		V	
VCC_RF output current	ICC_RF	All			50	mA	
Antenna gain	Gant	All			50	dB	
Receiver Chain Noise Figure	NFtot	All		3.0		dB	
Operating temperature	Topr	All	-40		85	°C	

Table 10: Operating conditions

Operation beyond the specified operating conditions can affect device reliability.

Indicative power requirements

Table 11 lists examples of the total system supply current for a possible application.

Parameter	Symbol	Module	Min	Typ	Max	Units	Condition
Max. supply current**	Iccp	All			67	mA	VCC = 3.6 V** / 1.95 V**
		Icc Acquisition	All	47**		mA	
		Icc Tracking (Max Performance mode)	NEO-6G/Q/T NEO-6M/R/W	40** 39**		mA	
Average supply current**	Icc	Icc Tracking (Stc mode)	NEO-6G/Q/T	38**		mA	VCC = 3.0 V** / 1.8 V**
			NEO-6M/R/W	37**		mA	
		Icc Tracking (Power Save mode / 1 Hz)	NEO-6G/Q	12**		mA	
			NEO-6M	11**		mA	

Table 11: Indicative power requirements

Values in Table 11 are provided for customer information only as an example of typical power requirements. Values are characterized on samples, actual power requirements can vary depending on FW version used, external circuitry, number of SVs tracked, signal strength, type of start as well as time, duration and conditions of test.

Pin Definition

Pin assignment

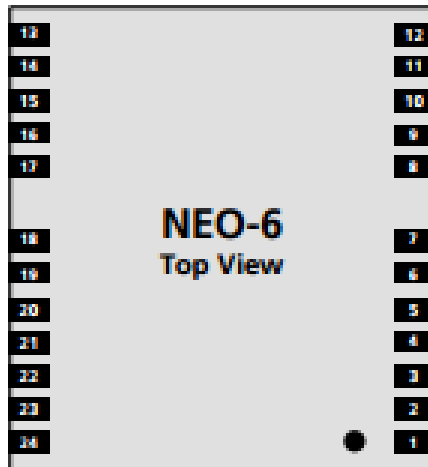


Figure 2 Pin Assignment

No	Module	Name	I/O	Description
1	All	Reserved	I	Reserved
2	All	SS_N	I	SP Slave Select
3	All	TIMEPULSE	O	Timepulse (1PPS)
4	All	EXTINT0	I	External Interrupt Pin
5	All	USB_DM	I/O	USB Data
6	All	USB_DP	I/O	USB Data
7	All	VDDUSB	I	USB Supply
8	All	Reserved		See Hardware Integration Manual Pin 8 and 9 must be connected together.
9	All	VCC_RF	O	Output Voltage RF section Pin 8 and 9 must be connected together.
10	All	GND	I	Ground
11	All	RF_IN	I	GPS signal input
12	All	GND	I	Ground
13	All	GND	I	Ground
14	All	MOSMCFG_COM0	O/I	SP MOSI / Configuration Pin. Leave open if not used.
15	All	MISOCFG_COM1	I	SP MISO / Configuration Pin. Leave open if not used.
16	All	CFG_GPSCLK	I	Power Mode Configuration Pin / SP Clock. Leave open if not used.
17	All	Reserved	I	Reserved
18	All	SDA2	I/O	I2C Data
19	All	SCL2	I/O	I2C Clock
20	All	TxD1	O	Serial Port 1
21	All	RxD1	I	Serial Port 1

No	Module	Name	I/O	Description
22	All	V_BCKP	I	Backup voltage supply
23	All	VCC	I	Supply voltage
24	All	GND	I	Ground

Table 8: Pinout


Pins designated Reserved should not be used. For more information about Pinouts see the LEA-6/NEO-6/MAX-6 Hardware Integration Manual [1].

Block diagram

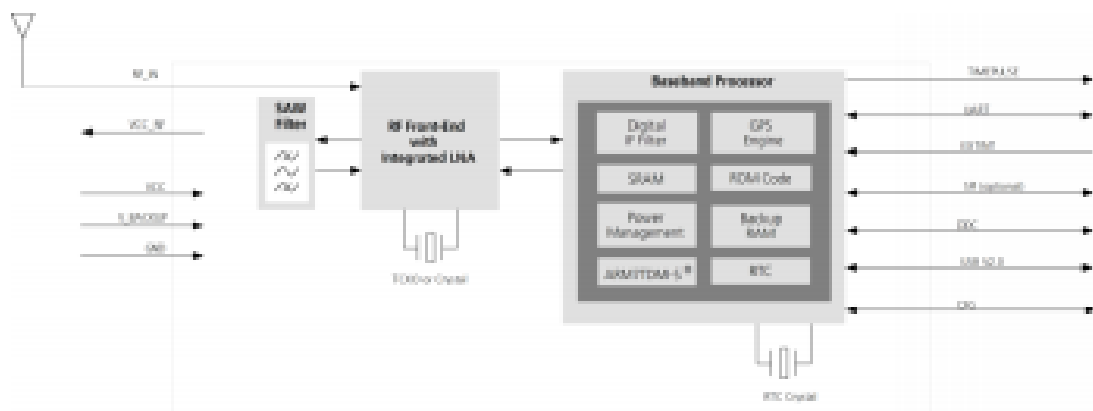


Figure 1: Block diagram (For available options refer to the product features table in section 1.2).

ANEXO 5.

DATASHEET TFT TOUCH 3.2 PULGADAS SSD1289

1 General Description

SSD1289 TFT Driver is an all in one driver that integrated the RAM, power circuits, gate driver and source driver into single chip. It can drive a 262k color a-TFT panel with resolution of 240 RGB x 320.

It also integrated the controller function and consists of up to 172,800 bytes (240 x 320 x 18 / 8) Graphic Display Data RAM (GDDRAM) such that it interfaced with common MCU through 8/9/16/18-bits 6800-series / 8080-series compatible Parallel Interface or Serial Interface and stored the data in the GDDRAM. Auxiliary 18-bits video interface (VSYNC, HSYNC, DOTCLK, ENABLE) are integrated into SSD1289 for displays animated image.

SSD1289 embeds DC-DC Converter and Voltage generator to provide all necessary voltage required by the driver with minimum external components. A Common Voltage Generation Circuit is included to drive the TFT-display counter electrode. An Integrated Gamma Control Circuit is also included that can be adjusted by software commands to provide maximum flexibility and optimal display quality.

SSD1289 can be operated down to 1.16V and provide different power save modes. It is suitable for any portable battery-driven applications requiring long operation period with compact size.

2 FEATURES

- Power Supply: $V_{DD} = 1.65V - 1.95V$ (non-regulated input for logic)
 $V_{DDIO} = 1.4V - 3.6V$ (regulated input for logic)
 $V_{CI} = 2.5V - 3.6V$ (power supply for internal analog circuit)
- Maximum Gate Driving Output Voltage: 30V p-p
- Source Driving Output Voltage: 0-5V
- Low Current Sleep Mode and 8-color display mode for power saving
- Display Size: 240 RGB x 320
- Display Color Support: 262k colors a-TFT displays
- 8/9/16/18-bits 6800-series / 8080-series Parallel Interface, Serial Peripheral Interface
- 18-bit RGB-interface for animated displays (VSYNC, HSYNC, DOTCLK, DEN, and D0-17)
- On-Chip 172,800 bytes (240x320x18/8) Graphic Display Data RAM
- RAM write synchronization function
- Support Line and Frame Inversion
- Software selection on Center Screen Scrolling, Top Screen Scrolling, Bottom Screen Scrolling and Whole Screen Scrolling.
- Source and Gate scan direction control
- On-Chip Voltage Generator
- On-Chip DC-DC Converter up to 6x / -6x
- Programmable Gamma Correction Curve
- Non-Volatile Memory (OTP) for VCOM calibration
- Programmable Common Electrode Voltage amplitude and level for Cs on common structure
- Support Cs on gate structure
- Available in COG package

3 ORDERING INFORMATION

Table 1 – Ordering Information

Ordering Part Number	Source (x RGB)	Gate	Package Form	Reference
SSD1289Z	240	320	Gold Bump Die	

4 BLOCK DIAGRAM

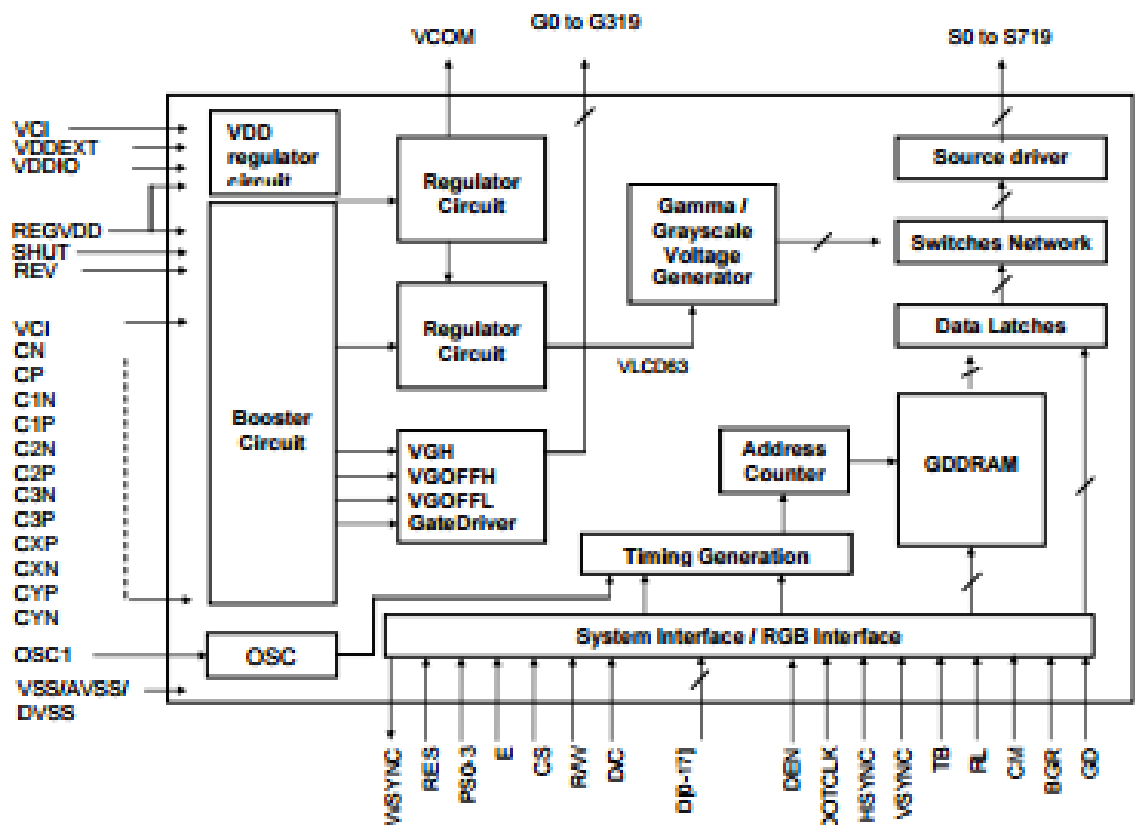


Figure 1. SSD1289 Block Diagram Description

ANEXO 6.

ESPECIFICACIONES IMPRESORA TÉRMICA

Feature:

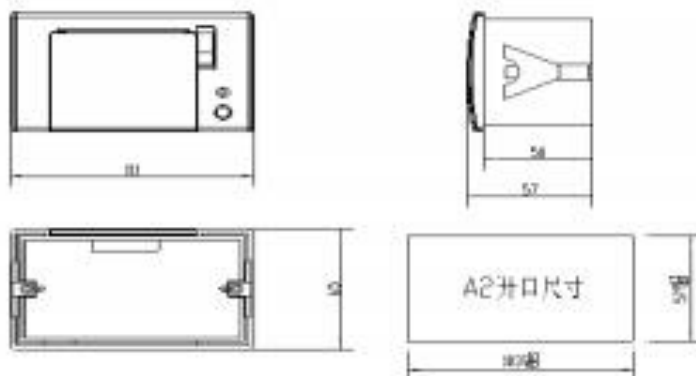
- ※ Smart appearance
- ※ Easy paper loading
- ※ Low noise thermal printing
- ※ Different interfaces optional
- ※ Front panel make paper replacement easily
- ※ Easily embedded to any kinds of instruments and meters

Application:

- ◆ Oiling Machine print proposal
- ◆ Queue machine Print proposal
- ◆ Recording Meter print proposal
- ◆ Self-service Print proposal
- ◆ Ticket Machine print proposal
- ◆ Medical instrument print proposal
- ◆ Weight Machine Print proposal
- ◆ Electric Instrument Print proposal
- ◆ Test Instrument Print proposal

Specification:

Print	Printing Method	Thermal Dot Line
	Printing Speed	50-80mm/s
	Resolution	8 dots/mm, 384 dots/line
Character	Effective Printing Width	48mm
	Character Set	ASCII,GB2312-80(Chinese)
	Print Font	ANK:5×7, Chinese: 12x24,24×24
Paper Spec	Paper Type	Thermal paper
	Paper Width	57.5± 0.5mm
	Paper Roll Diameter	Max: 39mm
Reliability	MCBF	5 million lines
Interface		Serial(RS-232,TTL), Parallel
Insert Depth		50mm
Power Supply (Adapter)		DC5V-9V
Physical	Outline Dimension (WxDxH)	111x65x57mm
	Installation Port Size	103 x 57mm
	Color	Beige/Black
Environment	Operating Temp	5°C ~ 50°C
	Operating Humidity	10% ~ 80%
	Storage Temp	-20°C ~ 60°C
	Storage Humidity	10% ~ 90%



Model No.: A2 Micro Pannel Thermal Printer

ANEXO 7.

MANUAL DE USUARIO

**TAXÍMETRO CON ALMACENAMIENTO
DE VIAJES MEDIANTE GPS**

MANUAL DE USUARIO



CONTENIDO

1. OPERACIÓN

1.1 Dispositivo taxímetro.

1.1.1. Menú de control.

1.1.2. Ventana de reloj.

1.1.3. Inicio de funcionamiento.

1.1.4. Fin de funcionamiento.

1.1.5. Impresión del recibo.

1.1.6. Paso a libre.

1.2 Software análisis de datos.

1.2.1 Importación de datos.

1.2.2 Estadística origen y destino.

1.2.3 Estadística general.

1.2.4 Representación gráfica.

1.2.4.1 Origen y destino.

1.2.4.2 Rutas.

2. MANTENIMIENTO

1. OPERACIÓN

1.1 Dispositivo taxímetro.

1.1.1 Menú de control.

El dispositivo cuenta con una pantalla táctil de 3.2", en la que se observa una portada con tres botones principales para el control de su funcionamiento. Estos se indican en la siguiente figura.



Figura 1. Menú de control.

1.1.2 Ventana de reloj.


Al presionar el botón  se indica la ventana de reloj, en la que se observa un mensaje de bienvenida, la hora y fecha actual tomadas del GPS.



Figura 2. Ventana de reloj.

1.1.3 Inicio de funcionamiento.


Al presionar el botón  se inicia el funcionamiento del taxímetro, en esta ventana se observa información como: tipo de tarifa, lugar de inicio, valor de arranque, tiempo de espera y kilómetros recorridos.



Figura 3. Ventana de funcionamiento de taxímetro.

1.1.4 Fin de funcionamiento.


Al presionar el botón  se da fin al funcionamiento del taxímetro, en esta ventana se indica el lugar de llegada, el subtotal y total que debe cancelar el usuario por el servicio de taxi.



Figura 4. Ventana de fin de funcionamiento.


1.1.5 Impresión de recibo.

La impresión del recibo se lo realiza a través de una impresora térmica y se efectúa automáticamente al momento de dar fin al funcionamiento del taxímetro.



Figura 5. Impresión de recibo.

1.1.6 Paso a libre.

Al presionar el botón  se pasa a la posición de libre y se indica la portada con su respectivo menú de control.


1.2 Software análisis de datos.

El software de análisis de datos cuenta con cinco ventanas en las que se observa la información de los viajes realizados, estadística origen y destino, estadística general y la representación gráfica de los lugares de inicio y llegada, como de las rutas seguidas por el taxi en sus diferentes viajes.




Figura 6. Portada software análisis de datos.

1.2.1 Importación de datos.

Al presionar el botón  se realiza la importación de los datos de los viajes realizados por el vehículo. Estos se visualizan en una tabla de 17 columnas.

PLAZA	COMPañIA	TIPO	MODDO	PARRIDA	POSICION INICIAL	FECHA INICIAL	HORA INICIAL	LLEGADA	POSICION FINAL	FECHA FINAL	HORA FINAL	DISTANCIA(km)	T. ESPERA(seg)	T. VIAJE(min)	SUBTOTALS	TOTALS
BA-4757	BROLAGTURIS S.A.	EXECUTIVO	Diá	LA FLORIDA	0.039642, -78.142253	27/11/2015	14:25:48	REGIONAL DE LA MADRE	0.348898, -78.120773	27/11/2015	15:40:22	1.55	39:25	4:54	1.50	1.25
BA-4757	BROLAGTURIS S.A.	EXECUTIVO	Diá	CENTRO, MERCADO	0.348898, -78.119430	27/11/2015	14:45:09	LA ESPERANZA	0.304433, -78.121827	27/11/2015	15:49:40	5.55	79:50	14:57	2:73	2:73
BA-4757	BROLAGTURIS S.A.	EXECUTIVO	Diá	LA ESPERANZA	0.304433, -78.121966	27/11/2015	15:00:15	LA ESPERANZA	0.304433, -78.121966	27/11/2015	15:00:25	5.55	79:50	14:57	2:73	2:73
BA-4757	BROLAGTURIS S.A.	EXECUTIVO	Diá	CARANQUI	0.022782, -78.107896	27/11/2015	15:08:52	LA VICTORIA	0.342376, -78.117063	27/11/2015	15:14:29	2.33	49:00	6:08	1:42	1:42
BA-4757	BROLAGTURIS S.A.	EXECUTIVO	Diá	LA VICTORIA	0.348898, -78.107378	27/11/2015	15:18:00	CENTRO, MERCADO	0.309469, -78.116263	27/11/2015	15:33:06	1.55	33:25	5:07	1:53	1:25
BA-4757	BROLAGTURIS S.A.	EXECUTIVO	Diá	CENTRO, MERCADO	0.348898, -78.117729	27/11/2015	15:27:44	CENTRO, MERCADO	0.345392, -78.119932	27/11/2015	15:33:23	1.15	162:00	7:41	1:08	1:25
BA-4757	BROLAGTURIS S.A.	EXECUTIVO	Diá	LOS CERROS	0.342846, -78.120983	27/11/2015	15:36:44	TERMINAL TERRESTRE	0.342836, -78.120831	27/11/2015	15:38:49	0.50	44:25	2:08	0:46	1:25
BA-4757	BROLAGTURIS S.A.	EXECUTIVO	Diá	TERMINAL TERRESTRE	0.342836, -78.120988	27/11/2015	15:46:13	REGIONAL DE LA MADRE	0.342836, -78.120989	27/11/2015	15:48:18	1.15	40:20	3:59	0:51	1:25
BA-4757	BROLAGTURIS S.A.	EXECUTIVO	Diá	HOSPITAL SAN VICENTE DE PAULI	0.323242, -78.120781	27/11/2015	15:47:51	CENTRO, MERCADO	0.342333, -78.121812	27/11/2015	15:52:21	1.02	111:00	4:48	0:56	1:25
BA-4757	BROLAGTURIS S.A.	EXECUTIVO	Diá	CENTRO, MERCADO	0.347301, -78.122177	27/11/2015	15:53:00	TERMINAL TERRESTRE	0.344879, -78.120638	27/11/2015	15:56:50	0.88	61:50	3:80	0:53	1:25

Figura 7. Ventana de datos.

Al presionar el botón  se borran los datos almacenados en la tarjeta de memoria.

1.2.2 Estadística origen y destino.

Al presionar el botón **DATOS** se observa la cantidad de viajes realizados desde y hacia los diferentes lugares de la ciudad de Ibarra. Estos datos son presentados en una matriz, en la que en su lado izquierdo se encuentran los lugares de inicio y en su parte superior los lugares de llegada.

	AZAYA	MIRAFLORES FAMILIARES	YAHUARCOCHA	LAS PALMAS	HOSPITAL DEL SEGURO	ESTADIO	EL OLIVO	JARDIN DE PAZ	HOSPITAL SAN VICENTE DE PAUL	LA MERCEZ	FRIGIFRIO	REDONDEL DE LA MADRE	CENTRO, MERCADO	LA VICTORIA	LA FLORIDA	EL ERIDO	PARKING CENTRICA BULEVAR	TERMINAL TERRESTRE	YACUICALLE PARQUE DE LA FAMILIA	LOS CERROS	LA CAMPANA Y PRIMAVERA	10 DE AGOSTO	CARANQUI	ERIDO DE CARANQUI	LA ESPERANZA	PROGRESO	MILAGRO	SAN ANTONIO
AZAYA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
MIRAFLORES FAMILIARES	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
YAHUARCOCHA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
LAS PALMAS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
HOSPITAL DEL SEGURO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ESTADIO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
EL OLIVO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
JARDIN DE PAZ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
HOSPITAL SAN VICENTE DE PAUL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
LA MERCEZ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
FRIGIFRIO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
REDONDEL DE LA MADRE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
CENTRO, MERCADO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
LA VICTORIA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
LA FLORIDA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
EL ERIDO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
PARKING CENTRICA BULEVAR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
TERMINAL TERRESTRE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
YACUICALLE PARQUE DE LA FAMILIA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
LOS CERROS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
LA CAMPANA Y PRIMAVERA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
10 DE AGOSTO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
CARANQUI	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ERIDO DE CARANQUI	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
LA ESPERANZA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
PROGRESO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
MILAGRO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
SAN ANTONIO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Figura 8. Ventana de origen y destino.

1.2.3 Estadística general.

Al presionar el botón **FECHAS/ACTUALIZAR** se visualiza las fechas disponibles para el análisis de datos generales. Estas fechas son presentadas en una tabla como se indica en la siguiente figura.

1	-27/11/2015
2	-27/11/2015
3	-27/11/2015
4	-27/11/2015
5	-27/11/2015
6	-27/11/2015
7	-27/11/2015
8	-27/11/2015
9	-27/11/2015
10	-27/11/2015
11	-03/12/2015
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	

Figura 9. Tabla de control de fechas.


Al presionar el botón **CALCULAR** se procede a realizar el cálculo de la información de las carreras realizadas durante el periodo de fechas seleccionadas.

	DÍA	NOCHE	TOTAL
NÚMERO DE VIAJES:	11	0	11
KM RECORRIDOS:	21,23	0	21,23
TIEMPO DE ESPERA: MINUTOS	13,31	0	13,31
TIEMPO DE VIAJE: MINUTOS	66,03	0	66,03
HORAS	1,1	0	1,1
COSTOS \$:	16,88	0	16,88

Figura 10. Tabla de estadística general.

1.2.4 Representación gráfica.

1.2.4.1 Origen y destino.

Al presionar el botón  se ejecuta en programa Power BI, en el que se visualiza en un mapa los lugares de inicio con color verde y los lugares de llegada con color rojo. Además se presenta en la parte inferior una gráfica de barras con la cantidad de carreras realizadas desde y hacia los lugares indicados en el mapa.

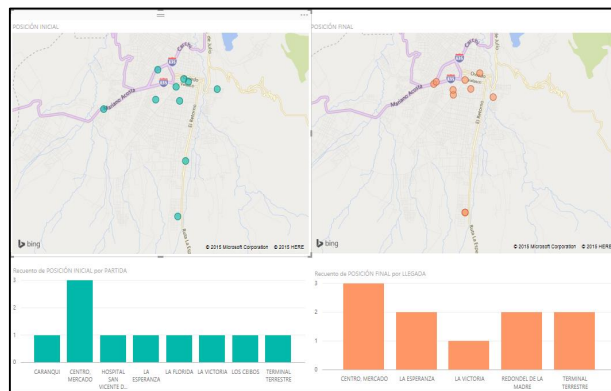



Figura 11. Representación gráfica de origen y destino.

1.2.4.2 Rutas.

Al presionar el botón  se ejecuta el programa U-center, en el que al carga el archivo de texto almacenado en la tarjeta de memoria se procede a graficar las rutas seguidas por el vehículo en los diferentes viajes realizados.

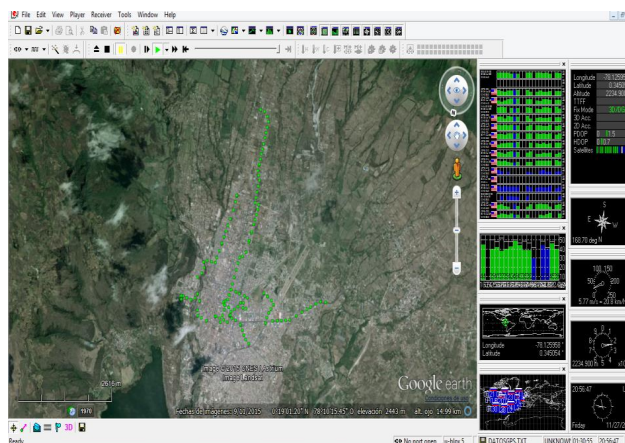


Figura 12. Representación gráfica de rutas.

2. MANTENIMIENTO

Un mantenimiento preventivo apropiado reduce los riesgos de reparaciones de las partes del dispositivo, así como también aumenta la vida útil del mismo, para ello se realiza las siguientes recomendaciones:

- Proteger del polvo el mismo que afecta al buen funcionamiento del dispositivo electrónico.
- Realizar una limpieza trimestral de cada uno de los componentes.
- Revisar semanalmente, las conexiones de los cables del dispositivo para no perder comunicación entre ellos.
- Utilizar paños especiales para la limpieza de la pantalla, evitar rayones.
- Revisar la conexión de la antena del GPS, semanalmente, para lograr obtener una buena recepción de señal.
- Revisar que el dispositivo se mantenga firme para evitar la vibración de sus componentes electrónicos.
- Revisar la conexión de la alimentación semanalmente, para evitar fallas en su funcionamiento.

ANEXO 8.

CÓDIGO DE PROGRAMACIÓN

```
#include <TimerOne.h> //LIBRERIA DEL TIMER
#include <TinyGPS++.h> //LIBRERIA DEL GPS
#include <SPI.h> //LIBRERIA DE LA COMUNICACION SPI
#include <SD.h> //LIBRERIA DE LA SD
#include "Adafruit_Thermal.h" //LIBRERIA DE LA IMPRESORA TERMICA
#include <UTouch.h> //LIBRERIA DEL CONTROL TACTIL
#include <UTFT.h> //LIBRERIA DE LA TFT 3.2"
#include <UTFT_tinyFAT.h> //LIBRERIA COMUNICACION ENTRE LA TFT Y SD
#include <tinyFAT.h> //LIBRERIA PARA LA LECTURA DE LA SD
#include <avr/pgmspace.h>

//Variables para el control
unsigned long tiempo2;
unsigned long tiempo_max= 4000;
boolean estado= false;
int almacena_ruta=0;
int centesimo=0;
int unidad=0;
int decena=4;
int dolar=0;
int dolarAnt=0;
int decenaAnt=0;
int unidadAnt=0;
int unidad_noche=4;
int decena_noche=4;
int diaAnt=0;
int mesAnt=0;
int a_oAnt=0;
int horaAnt=0;
int minutoAnt=0;
int unidad_nocheAnt=0;
int decena_nocheAnt=0;
double tiempo_total=0;//long
double recorrido_total=0;//float
double recorrido2=0;//float
double tiempo_viaje=0;//long
String lugar_inicio;
String lugar_fin;
long tiempo_totalAnt=0;
float recorrido_totalAnt=0;
float velocidad;
TinyGPSPlus gps; //Creamos la funcion del GPS
float recorrido=0;
float Lnganterior ;
float Latanterior ;
float Lngactual;
float Latactual;
long distanciasegundo;
int var=0;
int cont=0;
const int tiempo=0.5; //Tiempo en segundos
//Instanciamos la TFT
UTFT myGLCD(ITDB32S,38,39,40,41);
UTFT_tinyFAT myFiles(&myGLCD);
char* files320[]={ "PORTADA.RAW", "PORTADA2.RAW" };
char* files2[];
// Declaramos las fuentes para la TFT
extern uint8_t SmallFont[];
extern uint8_t BigFont[];
extern uint8_t SevenSegNumFont[];
extern uint8_t SevenSegNumFontPlusPlus[];

//Instanciamos el Touch
UTouch ts( 6, 5, 4, 3, 2);
File tarjetaSD;
#define SD_CS 53
Adafruit_Thermal printer(&Serial2); // Comunicacion serial con la impresora termica

void setup(){
  Timer1.initialize(750000);
  Serial.begin(38400);
  Serial1.begin(38400);
  Serial2.begin(19200);
  //Configuracion de la TFT
  myGLCD.InitLCD();
  myGLCD.fillScr(VGA_WHITE);
  myGLCD.setBackColor(VGA_TRANSPARENT);
  file.initFAT();
  //Configuracion del Touch
```

```

ts.InitTouch();
ts.setPrecision(PREC_MEDIUM);
//Configuracion de la SD
pinMode(53,OUTPUT);
if (!SD.begin(SD_CS))
{
myGLCD.setColor(VGA_RED);
myGLCD.setFont(BigFont);
myGLCD.print("ERROR SD", CENTER, 120);
return;
}
myGLCD.fillScr(VGA_WHITE);
myFiles.loadBitmap(0, 0, 320, 240, "PORTADA.RAW");
myGLCD.setColor(VGA_BLUE);
myGLCD.setFont(SmallFont);
myGLCD.print(" TAXIMETRO CON ALMACENAMIENTO ", CENTER, 5);
myGLCD.print("DE VIAJES MEDIANTE GPS.", CENTER, 18);
myGLCD.print("I.M.A.", 3, 225);
}

void loop()
{
while (Serial1.available() > 0)
if (gps.encode(Serial1.read()))
if (millis() > 30000 && gps.charsProcessed() < 10)
{
Serial.println(F("No hay COM GPS"));
while(true);
}
if(estado==true)
{
fecha();
hora();
}
long x, y;

if(ts.dataAvailable())
{
ts.read();
x = ts.getX();
y = ts.getY();
if((x>=320) && (x<=370))
{
if((y>=150) && (y<=175))
{
myGLCD.fillScr(VGA_BLUE);
myFiles.loadBitmap(0, 0, 320, 240, "PORTADA2.RAW");
myGLCD.setColor(VGA_BLUE);
myGLCD.fillRoundRect(0, 0, 319, 195);
myGLCD.setColor(VGA_WHITE);
myGLCD.drawRoundRect(0, 0, 319, 195);
myGLCD.drawRoundRect(1, 1, 318, 195);
myGLCD.drawRoundRect(2, 2, 317, 195);
myGLCD.setColor(VGA_RED);
myGLCD.setFont(BigFont);
myGLCD.print("BIENVENIDOS", CENTER, 4);
myGLCD.setColor(VGA_WHITE);
myGLCD.print("FECHA: dd/mm/aa", 25, 35);
myGLCD.print("HORA: hh:mm", 25, 110);
myGLCD.setColor(VGA_YELLOW);
myGLCD.drawRoundRect(15, 50, 300, 105);
myGLCD.drawRoundRect(16, 51, 299, 105);
myGLCD.drawRoundRect(17, 52, 298, 105);
myGLCD.setColor(VGA_LIME);
myGLCD.drawRoundRect(25, 125, 290, 180);
myGLCD.drawRoundRect(26, 126, 289, 180);
myGLCD.drawRoundRect(27, 127, 288, 180);
myGLCD.setColor(VGA_AQUA);
myGLCD.fillRect(25, 21, 275, 22);
myGLCD.fillRect(30, 24, 270, 25);
estado=true;
}
}
}
if((x>=330) && (x<=370))
{
if((y>=0) && (y<=25))
{
estado=false;
centicentavo=0;
unidad=0;
decena=4;
dolar=0;
dolarAnt=0;
decenaAnt=0;
unidadAnt=0;
unidad_noche=4;
}
}
}
}

```

```

decena_noche=4;
unidad_nocheAnt=0;
decena_nocheAnt=0;
tiempo_total=0;
recorrido_total=0;
recorrido2=0;
tiempo_viaje=0;
tiempo_totalAnt=0;
recorrido_totalAnt=0;
recorrido=0;
var=0;
cont=0;
myGLCD.fillRect(VGA_WHITE);
myFiles.loadBitmap(0, 0, 320, 240, "PORTADA.RAW");
myGLCD.setBackgroundColor(VGA_TRANSPARENT);
myGLCD.setColor(VGA_BLUE);
myGLCD.setFont(SmallFont);
myGLCD.print(" TAXIMETRO CON ALMACENAMIENTO ", CENTER, 5);
myGLCD.print("DE VIAJES MEDIANTE GPS.", CENTER, 18);
myGLCD.print("I.M.A.", 3, 225);
}
}

if((x>=320) && (x<=370))
{
if((y>=100) && (y<=125))
{
estado=false;
myGLCD.fillRect(VGA_BLACK);
myFiles.loadBitmap(0, 0, 320, 240, "PORTADA2.RAW");
myGLCD.setColor(VGA_WHITE);
myGLCD.fillRect(0, 0, 319, 196);
myGLCD.setColor(VGA_BLACK);
myGLCD.drawRoundRect(2, 2, 317, 194);
myGLCD.setColor(VGA_WHITE);
myGLCD.drawRoundRect(0, 196, 319, 239);
myGLCD.setColor(VGA_AQUA);
myGLCD.fillRect(6, 50, 310, 80);
myGLCD.setColor(VGA_YELLOW);
myGLCD.fillRect(6, 90, 155, 185);
myGLCD.setColor(VGA_RED);
myGLCD.fillRect(170, 100, 265, 130);
myGLCD.setColor(VGA_LIME);
myGLCD.fillRect(170, 155, 265, 185);
myGLCD.setColor(VGA_BLACK);
myGLCD.drawRoundRect(6, 50, 310, 80);
myGLCD.setColor(VGA_BLACK);
myGLCD.drawRoundRect(6, 90, 155, 185);
myGLCD.drawRoundRect(170, 100, 265, 130);
myGLCD.drawRoundRect(170, 155, 265, 185);
myGLCD.drawRoundRect(9, 53, 307, 80);
myGLCD.drawRoundRect(9, 93, 152, 185);
myGLCD.drawRoundRect(8, 52, 308, 80);
myGLCD.drawRoundRect(8, 92, 153, 185);
myGLCD.drawRoundRect(7, 51, 309, 80);
myGLCD.drawRoundRect(7, 91, 154, 185);
myGLCD.drawRoundRect(170, 100, 265, 130);
myGLCD.drawRoundRect(170, 155, 265, 185);
myGLCD.drawRoundRect(171, 101, 264, 130);
myGLCD.drawRoundRect(171, 156, 264, 185);
myGLCD.drawRoundRect(172, 102, 263, 130);
myGLCD.drawRoundRect(172, 157, 263, 185);
myGLCD.drawRoundRect(173, 103, 262, 130);
myGLCD.drawRoundRect(173, 158, 262, 185);
myGLCD.setColor(VGA_BLACK);
myGLCD.setBackgroundColor(VGA_TRANSPARENT);
myGLCD.setFont(SmallFont);
myGLCD.print("LUGAR DE INICIO:", 15, 35);
myGLCD.print("T. ESPERA (s):", 175, 85);
myGLCD.print("DISTANCIA (Km):", 175, 140);
almacena_ruta=1;

if(gps.time.hour()>=11 && gps.time.hour()<24)
{
myGLCD.setColor(VGA_BLUE);
myGLCD.setFont(BigFont);
myGLCD.print("TARIFA DIURNA", CENTER, 7);
myGLCD.fillRect(20, 25, 290, 27);
myGLCD.fillRect(20, 29, 290, 30);
almacena_inicio_dia();
myGLCD.setColor(VGA_BLACK);
myGLCD.print(lugar_inicio, 18, 60);
Timer1.attachInterrupt(Dia);
}
else
{

```



```

myGLCD.setColor(VGA_BLUE);
myGLCD.setFont(BigFont);
myGLCD.print("TARIFA NOCTURNA", CENTER, 7);
myGLCD.fillRect(20, 25, 290, 27);
myGLCD.fillRect(20, 29, 290, 30);
almacena_inicio_noche();
myGLCD.setColor(VGA_BLACK);
myGLCD.print(lugar_inicio, 18, 60);
Timer1.attachInterrupt(Noche);
}
}
}

if((x>=320) && (x<=370))
{
if((y>=50) && (y<=75))
{
estado=false;
if(gps.time.hour()>=11 && gps.time.hour(<24)
{
almacena_fin_dia();
Timer1.stop();
almacena_ruta=1;
myGLCD.fillRect(VGA_BLACK);
myFiles.loadBitmap(0, 0, 320, 240, "PORTADA2.RAW");
myGLCD.setColor(VGA_BLUE);
myGLCD.fillRect(0, 0, 319, 196);
myGLCD.setColor(VGA_BLACK);
myGLCD.drawRoundRect(2, 2, 317, 194);
myGLCD.setColor(VGA_BLUE);
myGLCD.drawRoundRect(0, 196, 319, 239);
myGLCD.setColor(VGA_LIME);
myGLCD.fillRect(6, 50, 308, 80);
myGLCD.setColor(VGA_BLACK);
myGLCD.drawRoundRect(6, 50, 308, 80);
myGLCD.drawRoundRect(9, 53, 305, 80);
myGLCD.drawRoundRect(8, 52, 306, 80);
myGLCD.drawRoundRect(7, 51, 307, 80);
myGLCD.setColor(VGA_RED);
myGLCD.fillRect(6, 85, 308, 115);
myGLCD.setColor(VGA_BLACK);
myGLCD.drawRoundRect(6, 85, 308, 115);
myGLCD.drawRoundRect(9, 88, 305, 115);
myGLCD.drawRoundRect(8, 87, 306, 115);
myGLCD.drawRoundRect(7, 86, 307, 115);
myGLCD.setColor(VGA_WHITE);
myGLCD.fillRect(6, 120, 308, 190);
myGLCD.setColor(VGA_BLACK);
myGLCD.drawRoundRect(6, 120, 308, 190);
myGLCD.drawRoundRect(9, 123, 305, 190);
myGLCD.drawRoundRect(8, 122, 306, 190);
myGLCD.drawRoundRect(7, 121, 307, 190);
myGLCD.setColor(VGA_WHITE);
myGLCD.fillRect(20, 25, 300, 27);
myGLCD.fillRect(20, 29, 300, 30);
myGLCD.setBackgroundColor(VGA_TRANSPARENT);
myGLCD.setFont(SmallFont);
myGLCD.print("LUGAR FIN:", 15, 35);
myGLCD.setColor(VGA_BLACK);
myGLCD.setFont(BigFont);
myGLCD.print(lugar_fin, 18, 60);
myGLCD.setFont(BigFont);
myGLCD.print("SUBTOTAL: ", 40, 95);
myGLCD.printNumF((0.40+(recorrido_total/1000)*0.40)+((tiempo_total/60000)*0.08),2, 200, 95);
myGLCD.setColor(VGA_RED);
myGLCD.print("TOTAL: ", 20, 140);
if((0.40+(recorrido_total/1000)*0.40)+((tiempo_total/60000)*0.08)<1.25)
{
myGLCD.setFont(SevenSegNumFontPlusPlus);
myGLCD.print("1.25", 110, 130);
myGLCD.setFont(BigFont);
myGLCD.print(" USD", 220, 170);
}
else
{
myGLCD.setFont(SevenSegNumFontPlusPlus);
myGLCD.printNumF((0.40+(recorrido_total/1000)*0.40)+((tiempo_total/60000)*0.08),2, 110, 130);
myGLCD.setFont(BigFont);
myGLCD.print(" USD", 220, 170);
}
}
delay(2000);
IMPRIMIR_DIA();
}
else
{
almacena_fin_noche();
}
}
}

```

```

Timer1.stop();
myGLCD.fillScr(VGA_BLACK);
myFiles.loadBitmap(0, 0, 320, 240, "PORTADA2.RAW");
myGLCD.setColor(VGA_BLUE);
myGLCD.fillRoundRect(0, 0, 319, 196);
myGLCD.setColor(VGA_BLACK);
myGLCD.drawRoundRect(2, 2, 317, 194);
myGLCD.setColor(VGA_BLUE);
myGLCD.drawRoundRect(0, 196, 319, 239);
myGLCD.setColor(VGA_LIME);
myGLCD.fillRoundRect(6, 50, 308, 80);
myGLCD.setColor(VGA_BLACK);
myGLCD.drawRoundRect(6, 50, 308, 80);
myGLCD.drawRoundRect(9, 53, 305, 80);
myGLCD.drawRoundRect(8, 52, 306, 80);
myGLCD.drawRoundRect(7, 51, 307, 80);
myGLCD.setColor(VGA_RED);
myGLCD.fillRoundRect(6, 85, 308, 115);
myGLCD.setColor(VGA_BLACK);
myGLCD.drawRoundRect(6, 85, 308, 115);
myGLCD.drawRoundRect(9, 88, 305, 115);
myGLCD.drawRoundRect(8, 87, 306, 115);
myGLCD.drawRoundRect(7, 86, 307, 115);
myGLCD.setColor(VGA_WHITE);
myGLCD.fillRoundRect(6, 120, 308, 190);
myGLCD.setColor(VGA_BLACK);
myGLCD.drawRoundRect(6, 120, 308, 190);
myGLCD.drawRoundRect(9, 123, 305, 190);
myGLCD.drawRoundRect(8, 122, 306, 190);
myGLCD.drawRoundRect(7, 121, 307, 190);
myGLCD.setColor(VGA_WHITE);
myGLCD.fillRect(20, 25, 300, 27);
myGLCD.fillRect(20, 29, 300, 30);
myGLCD.setBackgroundColor(VGA_TRANSPARENT);
myGLCD.setFont(SmallFont);
myGLCD.print("LUGAR FIN:", 15, 35);
myGLCD.setColor(VGA_BLACK);
myGLCD.setFont(BigFont);
myGLCD.print(lugar_fin, 18, 60);
myGLCD.setFont(BigFont);
myGLCD.print("SUBTOTAL: ", 40, 95);
myGLCD.printNumF(0.44+(recorrido_total/1000)*0.40)+((tiempo_total/60000)*0.09), 2, 200, 95);
myGLCD.setColor(VGA_RED);
myGLCD.print("TOTAL: ", 20, 140);
if((0.44+(recorrido_total/1000)*0.40)+((tiempo_total/60000)*0.09)<1.50, 20, 140)
{
myGLCD.setFont(SevenSegNumFontPlusPlus);
myGLCD.print("1.50", 110, 130);
myGLCD.setFont(BigFont);
myGLCD.print(" USD", 220, 170);
}
else
{
myGLCD.setFont(SevenSegNumFontPlusPlus);
myGLCD.printNumF(0.44+(recorrido_total/1000)*0.40)+((tiempo_total/60000)*0.09), 2, 110, 130);
myGLCD.setFont(BigFont);
myGLCD.print(" USD", 220, 170);
}
delay(2000);
IMPRIMIR_NOCHE();
}
}
}

if(almacena_ruta==1)
{
ruta();
almacena_ruta=0;
}
}

void fecha()
{
if (gps.date.isValid())
{
myGLCD.setBackgroundColor(VGA_BLUE);
myGLCD.setFont(SevenSegNumFont);
myGLCD.setColor(VGA_YELLOW);
myGLCD.printNumI(gps.date.day(), 18, 53);
myGLCD.fillRect(82, 75, 89, 79);
myGLCD.printNumI(gps.date.month(), 90, 53);
myGLCD.fillRect(155, 75, 162, 79);
myGLCD.printNumI(gps.date.year(), 170, 53);
}
}

```

```

}

void hora()
{
  if (gps.time.isValid())
  {
    myGLCD.setBackgroundColor(VGA_BLUE);
    myGLCD.setColor(VGA_LIME);
    int hora=(gps.time.hour()-5);
    if(hora==5)
    {myGLCD.print("19", 70, 128);
    }
    else if(hora==4)
    {myGLCD.print("20", 70, 128);
    }
    else if(hora==3)
    {myGLCD.print("21", 70, 128);
    }
    else if(hora==2)
    {myGLCD.print("22", 70, 128);
    }
    else if(hora==1)
    {myGLCD.print("23", 70, 128);
    }
    else
    {myGLCD.printNumI(hora, 70, 128);
    }
    myGLCD.fillRect(143, 145, 149, 150);
    myGLCD.fillRect(143, 160, 149, 165);
    myGLCD.printNumI(gps.time.minute(), 160, 128);
  }
}

void Dia()
{
  myGLCD.setBackgroundColor(VGA_YELLOW);
  myGLCD.setFont(SevenSegNumFont);
  myGLCD.setColor(VGA_RED);
  myGLCD.printNumI(dolar, 25, 115);
  myGLCD.fillRect(62, 157, 67, 163);
  myGLCD.printNumI(decena, 70, 115);
  myGLCD.printNumI(unidad, 105, 115);

  velocidad=(gps.speed.kmph());
  if(velocidad<2)
  {
    velocidad=0;
  }
  Serial.print(F(" Velocidad: "));
  Serial.print(velocidad);
  Serial.print(F(" km/h "));
  tiempo_viaje=tiempo_viaje+750;

  if(velocidad >0 )
  {
    cont=cont+1;
    if(cont>=tiempo)
    {
      Lngactual =gps.location.lng();
      Latactual =gps.location.lat();
      if(var==0)
      {
        Lnganterior =gps.location.lng();
        Latanterior =gps.location.lat();
        var=155;
      }
      distanciasegundo = (long)TinyGPSPlus::distanceBetween(Latactual,Lngactual,Latanterior,Lnganterior) ;
      recorrido= recorrido+distanciasegundo;
      Serial.print("RECORRIDO: ");Serial.println(recorrido);
      Lnganterior =gps.location.lng();
      Latanterior =gps.location.lat();
      cont=0;
    }
  }

  if (recorrido >=20)
  {
    recorrido2++;
    recorrido=0;
    unidad++;
    almacena_ruta=1;
  }
  if(unidad >9)
  {
    unidad=0;
    decena++;
  }
}

```

```

    if(decena > 9)
    {
        decena=0;
        dolar++;
    }
    recorrido_total=recorrido2*25;
    Serial.print("DISTANCIA: "); Serial.println(recorrido_total);
    myGLCD.setBackColor(VGA_LIME);
    myGLCD.setFont(BigFont);
    myGLCD.setColor(VGA_BLACK);
    myGLCD.printNumF(recorrido_total/1000, 2, 185, 164);
    int radius = 10;
    myGLCD.setColor(VGA_WHITE);
    myGLCD.fillCircle(298, 170, radius);
    myGLCD.fillCircle(298, 170, radius);
    myGLCD.fillCircle(298, 170, radius);
    myGLCD.fillCircle(298, 170, radius);
    myGLCD.fillCircle(298, 170, radius);
    myGLCD.fillCircle(298, 170, radius);
    myGLCD.fillCircle(298, 170, radius);
    myGLCD.fillCircle(298, 170, radius);
    myGLCD.fillCircle(298, 170, radius);
    myGLCD.fillCircle(298, 170, radius);
    myGLCD.setColor(VGA_LIME);
    myGLCD.fillCircle(298, 170, radius);

    }
    else
    {
        delay(750);
        cententavo++;
        Serial.println(cententavo);
        if(cententavo >9)
        {
            cententavo=0;
            unidad++;
        }
        if(unidad >9)
        {
            unidad=0;
            decena++;
        }
        if(decena > 9)
        {
            decena=0;
            dolar++;
        }

        tiempo_total=tiempo_total+750;
        myGLCD.setBackColor(VGA_RED);
        myGLCD.setFont(BigFont);
        myGLCD.setColor(VGA_BLACK);
        myGLCD.printNumI(tiempo_total/1000, 190, 110);
        int radius = 10;
        myGLCD.setColor(VGA_WHITE);
        myGLCD.fillCircle(298, 115, radius);
        myGLCD.fillCircle(298, 115, radius);
        myGLCD.fillCircle(298, 115, radius);
        myGLCD.fillCircle(298, 115, radius);
        myGLCD.fillCircle(298, 115, radius);
        myGLCD.fillCircle(298, 115, radius);
        myGLCD.fillCircle(298, 115, radius);
        myGLCD.fillCircle(298, 115, radius);
        myGLCD.fillCircle(298, 115, radius);
        myGLCD.fillCircle(298, 115, radius);
        myGLCD.setColor(VGA_RED);
        myGLCD.fillCircle(298, 115, radius);
    }
}

void Noche()
{
    myGLCD.setBackColor(VGA_YELLOW);
    myGLCD.setFont(SevenSegNumFont);
    myGLCD.setColor(VGA_RED);
    myGLCD.printNumI(dolar, 25, 115);
    myGLCD.fillRect(62, 157, 67, 163);
    myGLCD.printNumI(decena_noche, 70, 115);
    myGLCD.printNumI(unidad_noche, 105, 115);
    velocidad=(gps.speed.kmph());
    if(velocidad<2)
    {
        velocidad=0;
    }
    Serial.print(F(" Velocidad: "));
    Serial.print(velocidad);
    Serial.print(F(" km/h "));
    tiempo_viaje=tiempo_viaje+750;
}

```

```

if(velocidad >0 )
{
cont=cont+1;
if(cont>=tiempo)
{
Lngactual =gps.location.lng();
Latactual =gps.location.lat();
if(var==0)
{
Lnganterior =gps.location.lng();
Latanterior =gps.location.lat();
var=155;
}
distanciasegundo = (long)TinyGPSPlus::distanceBetween(Latactual,Lngactual,Lat anterior,Lnganterior) ;
recorrido= recorrido+distanciasegundo;
Serial.print("RECORRIDO: ");Serial.println(recorrido);
Lnganterior =gps.location.lng();
Lat anterior =gps.location.lat();
cont=0;
}

if (recorrido >=20)
{
recorrido2++;
recorrido=0;
unidad_noche++;
almacena_ruta=1;
}
if(unidad_noche >9)
{
unidad_noche=0;
decena_noche++;
}
if(decena_noche > 9)
{
decena=0;
dolar++;
}
recorrido_total=recorrido2*25;
Serial.print("DISTANCIA: "); Serial.println(recorrido_total);
myGLCD.setBackgroundColor(VGA_LIME);
myGLCD.setFont(BigFont);
myGLCD.setColor(VGA_BLACK);
myGLCD.printNumF(recorrido_total/1000, 2, 185, 164);
int radius = 10;
myGLCD.setColor(VGA_WHITE);
myGLCD.fillCircle(298, 170, radius);
myGLCD.fillCircle(298, 170, radius);
myGLCD.fillCircle(298, 170, radius);
myGLCD.fillCircle(298, 170, radius);
myGLCD.fillCircle(298, 170, radius);
myGLCD.fillCircle(298, 170, radius);
myGLCD.fillCircle(298, 170, radius);
myGLCD.fillCircle(298, 170, radius);
myGLCD.fillCircle(298, 170, radius);
myGLCD.fillCircle(298, 170, radius);
myGLCD.setColor(VGA_LIME);
myGLCD.fillCircle(298, 170, radius);
}

else
{
delay(750);
centicentavo++;
Serial.println(centicentavo);
if(centicentavo >9)
{
centicentavo=0;
unidad_noche++;
}
if(unidad_noche >9)
{
unidad_noche=0;
decena_noche++;
}
if(decena_noche > 9)
{
decena_noche=0;
dolar++;
}
tiempo_total=tiempo_total+750;
myGLCD.setBackgroundColor(VGA_RED);
myGLCD.setFont(BigFont);
myGLCD.setColor(VGA_BLACK);
myGLCD.printNumI(tiempo_total/1000, 190, 110);
int radius = 10;
myGLCD.setColor(VGA_WHITE);
}

```

```

        myGLCD.fillCircle(298, 115, radius);
        myGLCD.fillCircle(298, 115, radius);
        myGLCD.fillCircle(298, 115, radius);
        myGLCD.fillCircle(298, 115, radius);
        myGLCD.fillCircle(298, 115, radius);
        myGLCD.fillCircle(298, 115, radius);
        myGLCD.fillCircle(298, 115, radius);
        myGLCD.fillCircle(298, 115, radius);
        myGLCD.fillCircle(298, 115, radius);
        myGLCD.setColor(VGA_RED);
        myGLCD.fillCircle(298, 115, radius);
    }
}

void IMPRIMIR_DIA()
{
    myGLCD.setColor(VGA_YELLOW);
    myGLCD.setFont(BigFont);
    myGLCD.print("IMPRIMIENDO", CENTER, 7);
    // Justificacion del texto(right, center, left) -- 'L', 'C', 'R'
    printer.justify('C');
    printer.setSize('S');
    printer.println(F("***COMPANIA***"));
    printer.setSize('M');
    printer.println(F("BRISLAGTURIS S.A."));
    printer.setSize('S');
    printer.println(F("IBARRA-ECUADOR"));
    printer.println(F("RUC:-----"));
    printer.println(F("Aut. SRI:-----"));
    printer.println(F("-----"));
    printer.println(F("PROPIETARIO"));
    printer.println(F("CELULAR"));
    printer.println(F("Placa:IBA-9757"));
    printer.println();
    printer.justify('C');
    printer.print(F("Concepto: TRANSPORTE"));
    printer.println();
    printer.justify('L');
    printer.println();
    printer.print(F("Fecha de emision: "));
    printer.print(gps.date.day());
    printer.print(F("/"));
    printer.print(gps.date.month());
    printer.print(F("/"));
    printer.print(gps.date.year());
    printer.println();
    printer.print(F("Hora de emision: "));
    int hora=(gps.time.hour()-5);
    printer.print(hora);
    printer.print(F(":"));
    printer.print(gps.time.minute());
    printer.print(F(":"));
    printer.print(gps.time.second());
    printer.justify('L');
    printer.println();
    printer.print(F("Inicio: "));
    printer.print(lugar_inicio);
    printer.println();
    printer.print(F("Llegada: "));
    printer.print(lugar_fin);
    printer.println();
    printer.print(F("Tiempo de viaje: "));
    printer.print(tiempo_viaje/60000);
    printer.print(F(" min"));
    printer.println();
    printer.println();
    printer.justify('C');
    printer.println(F(".....Tarifa Diurna....."));
    printer.justify('L');
    printer.println();
    printer.print(F("Arranque:          $ 0.40"));
    printer.justify('L');
    printer.println();
    printer.print(F("Distancia: "));
    printer.print(recorrido_total/1000);
    printer.print(F("Km"));
    printer.justify('L');
    printer.println();
    printer.print(F("Tarifa $ 0.40/Km      $ "));
    printer.print((recorrido_total/1000)*0.40);
    printer.justify('L');
    printer.println();
    printer.print(F("Espera: "));
    printer.print(tiempo_total/1000);
    printer.print("seg");
    printer.justify('L');
}

```

```

printer.println();
printer.print(F("Tarifa: $ 0.08/Min   $ "));
printer.print((tiempo_total/60000)*0.08);
printer.println();
printer.justify('L');
printer.setSize('M');
printer.print(F("   SUBTOTAL--- $ "));
printer.print(0.40+((recorrido_total/1000)*0.40)+((tiempo_total/60000)*0.08));
printer.setSize('S');
printer.justify('L');
printer.println();
printer.print(F("Minima:           $ 1.25"));
printer.justify('L');
printer.println();
printer.print(F("IVA  0%"));
printer.println();
printer.justify('L');
printer.setSize('M');
printer.print(F("   TOTAL--- $ "));
if((0.40+((recorrido_total/1000)*0.40)+((tiempo_total/60000)*0.08))<1.25)
{
  printer.print(F("1.25"));
}
else
{
  printer.print(0.40+((recorrido_total/1000)*0.40)+((tiempo_total/60000)*0.08));
}
printer.setSize('S');
printer.justify('C');
printer.println();
printer.print(F("-----"));
printer.println(F("Taximetro con almacenamiento"));
printer.println(F("de viajes mediante GPS"));
printer.println(F("UNIVERSIDAD TECNICA DEL NORTE"));
printer.println(F("INGENIERIA AUTOMOTRIZ"));
printer.sleep(); // Parar la impresora
delay(3000L); // Para la impresora por 3 segundos
printer.wake(); // Activar la impresora
printer.setDefault(); // Restaurar la impresora a los valores predeterminados
}

```

```

void IMPRIMIR_NOCHE()
{
  myGLCD.setColor(VGA_YELLOW);
  myGLCD.setFont(BigFont);
  myGLCD.print("IMPRIMIENDO", CENTER, 7);
  printer.justify('C');
  printer.setSize('S');
  printer.println(F("***COMPANIA***"));
  printer.setSize('M');
  printer.println(F("BRISLAGTURIS S.A."));
  printer.setSize('S');
  printer.println(F("IBARRA-ECUADOR"));
  printer.println(F("RUC:-----"));
  printer.println(F("Aut. SRI:-----"));
  printer.println(F("-----"));
  printer.println(F("PROPIETARIO"));
  printer.println(F("CELULAR"));
  printer.println(F("Placa:IBA-9757"));
  printer.println();
  printer.justify('C');
  printer.print(F("Concepto: TRANSPORTE"));
  printer.println();
  printer.justify('L');
  printer.println();
  printer.print(F("Fecha de emision:  "));
  printer.print(gps.date.day());
  printer.print(F(" "));
  printer.print(gps.date.month());
  printer.print(F(" "));
  printer.print(gps.date.year());
  printer.println();
  printer.print(F("Hora de emision:  "));
  int hora=(gps.time.hour()-5);
  if(hora==5)
  {printer.print("19");
  }
  else if(hora==4)
  {printer.print("20");
  }
  else if(hora==3)
  {printer.print("21");
  }
  else if(hora==2)
  {printer.print("22");
  }
}

```

```

    }
    else if(hora==1)
    {printer.print("23");
    }
    else
    {printer.print(hora);
    }
printer.print(F(":"));
printer.print(gps.time.minute());
printer.print(F(":"));
printer.print(gps.time.second());
printer.justify('L');
printer.println();
printer.print(F("Inicio: "));
printer.print(lugar_inicio);
printer.println();
printer.print(F("Legada: "));
printer.print(lugar_fin);
printer.println();
printer.print(F("Tiempo de viaje: "));
printer.print(tiempo_viaje/60000);
printer.print(F(" min"));
printer.println();
printer.println();
printer.justify('C');
printer.println(F(".....Tarifa Nocturna....."));
printer.justify('L');
printer.println();
printer.print(F("Arranque:          $ 0.44"));
printer.justify('L');
printer.println();
printer.print(F("Distancia: "));
printer.print(recorrido_total/1000);
printer.print(F("Km"));
printer.justify('L');
printer.println();
printer.print(F("Tarifa $ 0.40/Km      $ "));
printer.print((recorrido_total/1000)*0.40);
printer.justify('L');
printer.println();
printer.print(F("Espera: "));
printer.print(tiempo_total/1000);
printer.print("seg");
printer.justify('L');
printer.println();
printer.print(F("Tarifa: $ 0.09/Min    $ "));
printer.print((tiempo_total/60000)*0.09);
printer.println();
printer.justify('L');
printer.setSize('M');
printer.print(F("      SUBTOTAL--- $ "));
printer.print(0.44+((recorrido_total/1000)*0.40)+((tiempo_total/60000)*0.09));
printer.setSize('S');
printer.justify('L');
printer.println();
printer.print(F("Minima:          $ 1.50"));
printer.justify('L');
printer.println();
printer.print(F("IVA 0%"));
printer.println();
printer.justify('L');
printer.setSize('M');
printer.print(F("      TOTAL--- $ "));
if((0.44+((recorrido_total/1000)*0.40)+((tiempo_total/60000)*0.09))<1.50)
{
    printer.print(F("1.50"));
}
else
{
    printer.print(0.44+((recorrido_total/1000)*0.40)+((tiempo_total/60)*0.09));
}
printer.setSize('S');
printer.justify('C');
printer.println();
printer.print(F("-----"));
printer.println(F("Taximetro con almacenamiento"));
printer.println(F("de viajes mediante GPS"));
printer.println(F("UNIVERSIDAD TECNICA DEL NORTE"));
printer.println(F("INGENIERIA AUTOMOTRIZ"));
printer.sleep();
delay(3000L);
printer.wake();
printer.setDefault();
}
}

void almacena_inicio_dia()

```



```

{
tarjetaSD=SD.open("DatosGPS.csv",FILE_WRITE);
if(tarjetaSD)
{
tarjetaSD.print(F("Dia"));
tarjetaSD.print(",");
tarjetaSD.print(F("IBA-9757"));
tarjetaSD.print(",");
tarjetaSD.print(F("BRISLAGTURIS S.A."));
tarjetaSD.print(",");
tarjetaSD.print(F("EJECUTIVO"));
tarjetaSD.print(",");
tarjetaSD.print(F("Dia"));
tarjetaSD.print(",");
lugar_partida();
tarjetaSD.print(",");
if (gps.location.isValid()
{
tarjetaSD.print(gps.location.lat(), 6);
tarjetaSD.print(F(","));
tarjetaSD.print(gps.location.lng(), 6);
tarjetaSD.print(",");
}
)
if (gps.date.isValid()
{
tarjetaSD.print(".");
if (gps.date.day() < 10)
{
tarjetaSD.print("0");
}
}
tarjetaSD.print(gps.date.day());
tarjetaSD.print(F("/"));
if (gps.date.month() < 10)
{
tarjetaSD.print("0");
}
}
tarjetaSD.print(gps.date.month());
tarjetaSD.print(F("/"));
tarjetaSD.print(gps.date.year());
tarjetaSD.print(",");
}
if (gps.time.isValid()
{
tarjetaSD.print(".");
if (gps.time.hour() < 10)
{
tarjetaSD.print("0");
}
}
int hora=(gps.time.hour())-5;
tarjetaSD.print(hora);
tarjetaSD.print(F(":"));
if (gps.time.minute() < 10)
{
tarjetaSD.print("0");
}
tarjetaSD.print(gps.time.minute());
tarjetaSD.print(F(":"));
if (gps.time.second() < 10)
{
tarjetaSD.print("0");
}
tarjetaSD.print(gps.time.second());
tarjetaSD.print(",");
}
tarjetaSD.close();
}
}

void almacena_inicio_noche()
{
tarjetaSD=SD.open("DatosGPS.csv",FILE_WRITE);
if(tarjetaSD)
{
tarjetaSD.print(F("Noche"));
tarjetaSD.print(",");
tarjetaSD.print(F("IBA-9757"));
tarjetaSD.print(",");
tarjetaSD.print(F("BRISLAGTURIS S.A."));
tarjetaSD.print(",");
tarjetaSD.print(F("EJECUTIVO"));
tarjetaSD.print(",");
tarjetaSD.print(F("Noche"));
tarjetaSD.print(",");
lugar_partida();
tarjetaSD.print(",");
if (gps.location.isValid()

```

```

{
tarjetaSD.print(gps.location.lat(), 6);
tarjetaSD.print(F(",");
tarjetaSD.print(gps.location.lng(), 6);
tarjetaSD.print(";");
}
if (gps.date.isValid())
{
tarjetaSD.print(".");
if (gps.date.day() < 10)
{
tarjetaSD.print("0");
}
tarjetaSD.print(gps.date.day());
tarjetaSD.print(F(",");
if (gps.date.month() < 10)
{
tarjetaSD.print("0");
}
tarjetaSD.print(gps.date.month());
tarjetaSD.print(F(",");
tarjetaSD.print(gps.date.year());
tarjetaSD.print(";");
}
if (gps.time.isValid())
{
tarjetaSD.print(".");
int hora=(gps.time.hour()-5);
if(hora==5)
{tarjetaSD.print("19");
}
else if(hora==4)
{tarjetaSD.print("20");
}
else if(hora==3)
{tarjetaSD.print("21");
}
else if(hora==2)
{tarjetaSD.print("22");
}
else if(hora==1)
{tarjetaSD.print("23");
}
else
{tarjetaSD.print(hora);
}
tarjetaSD.print(F(",");
if (gps.time.minute() < 10)
{
tarjetaSD.print("0");
}
tarjetaSD.print(gps.time.minute());
tarjetaSD.print(F(",");
if (gps.time.second() < 10)
{
tarjetaSD.print("0");
}
tarjetaSD.print(gps.time.second());
tarjetaSD.print(";");
}
tarjetaSD.close();
}
}

void almacena_fin_dia()
{
tarjetaSD=SD.open("DatosGPS.csv",FILE_WRITE);
if(tarjetaSD)
{
lugar_llegada();
tarjetaSD.print(",");
if (gps.location.isValid())
{
tarjetaSD.print(gps.location.lat(), 6);
tarjetaSD.print(F(",");
tarjetaSD.print(gps.location.lng(), 6);
tarjetaSD.print(";");
}
if (gps.date.isValid())
{
tarjetaSD.print(".");
if (gps.date.day() < 10)
{
tarjetaSD.print("0");
}
tarjetaSD.print(gps.date.day());
}
}
}

```

```

tarjetaSD.print(F(""));
if (gps.date.month() < 10)
{
tarjetaSD.print("0");
}
tarjetaSD.print(gps.date.month());
tarjetaSD.print(F(""));
tarjetaSD.print(gps.date.year());
tarjetaSD.print(";");
}
if (gps.time.isValid())
{
tarjetaSD.print(".");
if (gps.time.hour() < 10)
{
tarjetaSD.print("0");
}
int hora=(gps.time.hour())-5;
tarjetaSD.print(hora);
tarjetaSD.print(F(":"));
if (gps.time.minute() < 10)
{
tarjetaSD.print("0");
}
tarjetaSD.print(gps.time.minute());
tarjetaSD.print(F(":"));
if (gps.time.second() < 10)
{
tarjetaSD.print("0");
}
tarjetaSD.print(gps.time.second());
tarjetaSD.print(";");
}
tarjetaSD.print(recorrido_total/1000);
tarjetaSD.print(";");
tarjetaSD.print(tiempo_total/1000);
tarjetaSD.print(";");
tarjetaSD.print(tiempo_viaje/60000);
tarjetaSD.print(";");
tarjetaSD.print(0.40+((recorrido_total/1000)*0.40)+((tiempo_total/60000)*0.08));
tarjetaSD.print(";");
if((0.40+(recorrido_total/1000)*0.40)+((tiempo_total/60000)*0.08)<1.25)
{
tarjetaSD.print(F("1.25"));
}
else
{
tarjetaSD.print(0.40+((recorrido_total/1000)*0.40)+((tiempo_total/60000)*0.08));
}
tarjetaSD.println();
tarjetaSD.close();
}
}

void almacena_fin_noche()
{
tarjetaSD=SD.open("DatosGPS.csv",FILE_WRITE);
if(tarjetaSD)
{
lugar_llegada();
tarjetaSD.print(";");
if (gps.location.isValid())
{
tarjetaSD.print(gps.location.lat(), 6);
tarjetaSD.print(F(","));
tarjetaSD.print(gps.location.lng(), 6);
tarjetaSD.print(";");
}
if (gps.date.isValid())
{
tarjetaSD.print(".");
if (gps.date.day() < 10)
{
tarjetaSD.print("0");
}
tarjetaSD.print(gps.date.day());
tarjetaSD.print(F(""));
if (gps.date.month() < 10)
{
tarjetaSD.print("0");
}
tarjetaSD.print(gps.date.month());
tarjetaSD.print(F(""));
tarjetaSD.print(gps.date.year());
tarjetaSD.print(";");
}
}
}

```

```

if (gps.time.isValid())
{
tarjetaSD.print(".");
int hora=(gps.time.hour()-5);
if(hora==5)
{tarjetaSD.print("19");
}
else if(hora==4)
{tarjetaSD.print("20");
}
else if(hora==3)
{tarjetaSD.print("21");
}
else if(hora==2)
{tarjetaSD.print("22");
}
else if(hora==1)
{tarjetaSD.print("23");
}
else
{tarjetaSD.print(hora);
}
tarjetaSD.print(F(":"));
if (gps.time.minute() < 10)
{
tarjetaSD.print("0");
}
tarjetaSD.print(gps.time.minute());
tarjetaSD.print(F(":"));
if (gps.time.second() < 10)
{
tarjetaSD.print("0");
}
tarjetaSD.print(gps.time.second());
tarjetaSD.print(";");
}
tarjetaSD.print(recorrido_total/1000);
tarjetaSD.print(";");
tarjetaSD.print(tiempo_total/1000);
tarjetaSD.print(";");
tarjetaSD.print(tiempo_viaje/60000);
tarjetaSD.print(";");
tarjetaSD.print(0.44+((recorrido_total/1000)*0.40)+((tiempo_total/60000)*0.09));
tarjetaSD.print(";");
if((0.44+((recorrido_total/1000)*0.40)+((tiempo_total/60000)*0.09))<1.50)
{
tarjetaSD.print(F("1.50"));
}
else
{
tarjetaSD.print(0.44+((recorrido_total/1000)*0.40)+((tiempo_total/60000)*0.09));
}
tarjetaSD.println();
tarjetaSD.close();
}
}

void lugar_partida()
{
if (gps.location.isValid())
{
long latitud=(gps.location.lat()*1000000);
long longitud=abs(gps.location.lng()*1000000);
Serial.print("latitud=");
Serial.print(latitud);
Serial.println();
Serial.print("longitud=");
Serial.print(longitud);
Serial.println();

if(390194>=latitud && latitud>=365508 && (78142340)>=longitud && longitud>=(78124423))
{
tarjetaSD.print(F("AZAYA"));
Serial.print(F("AZAYA"));
lugar_inicio=String("Azaya");
}
else if(390194>=latitud && latitud>=377321 && (78124445)>=longitud && longitud>=(78113565))
{
tarjetaSD.print(F("AZAYA"));
Serial.print(F("AZAYA"));
lugar_inicio=String("Azaya");
}
else if(377321>=latitud && latitud>=362311 && (78124445)>=longitud && longitud>=(78114343))
{
tarjetaSD.print(F("HUERTOS FAMILIARES"));
}
}
}

```

```

Serial.print(F("HUERTOS FAMILIARES"));
lugar_inicio=String("Huertos Familiares");
}
else if(390194>=latitud && latitud>=369693 && (78114392)>=longitud && longitud>=(78085592))
{
tarjetaSD.print(F("YAHUARCOCHA"));
Serial.print(F("YAHUARCOCHA"));
lugar_inicio=String("Yahuarcocha");
}
else if(369693>=latitud && latitud>=350236 && (78108460)>=longitud && longitud>=(78085592))
{
tarjetaSD.print(F("YAHUARCOCHA"));
Serial.print(F("YAHUARCOCHA"));
lugar_inicio=String("Yahuarcocha");
}
else if(365551>=latitud && latitud>=355520 && (78146106)>=longitud && longitud>=(78130699))
{
tarjetaSD.print(F("ALPACHACA"));
Serial.print(F("ALPACHACA"));
lugar_inicio=String("Alpachaca");
}
else if(365551>=latitud && latitud>=362311 && (78130699)>=longitud && longitud>=(78124412))
{
tarjetaSD.print(F("ALPACHACA"));
Serial.print(F("ALPACHACA"));
lugar_inicio=String("Alpachaca");
}
else if(362289>=latitud && latitud>=355520 && (78130699)>=longitud && longitud>=(78121991))
{
tarjetaSD.print(F("HOSPITAL DEL SEGURO"));
Serial.print(F("HOSPITAL DEL SEGURO"));
lugar_inicio=String("Hosp. del Seguro");
}
else if(362257>=latitud && latitud>=355520 && (78122170)>=longitud && longitud>=(78114354))
{
tarjetaSD.print(F("ESTADIO"));
Serial.print(F("ESTADIO"));
lugar_inicio=String("Estadio");
}
else if(369693>=latitud && latitud>=350236 && (78114354)>=longitud && longitud>=(78108460))
{
tarjetaSD.print(F("EL OLIVO"));
Serial.print(F("EL OLIVO"));
lugar_inicio=String("El Olivo");
}
else if(355595>=latitud && latitud>=350890 && (78148826)>=longitud && longitud>=(78132225))
{
tarjetaSD.print(F("JARDIN DE PAZ"));
Serial.print(F("JARDIN DE PAZ"));
lugar_inicio=String("Jardin de Paz");
}
else if(355595>=latitud && latitud>=350890 && (78132225)>=longitud && longitud>=(78122826))
{
tarjetaSD.print(F("HOSPITAL SAN VICENTE DE PAUL"));
Serial.print(F("HOSPITAL SAN VICENTE DE PAUL"));
lugar_inicio=String("Hosp. San Vicente de P.");
}
else if(355595>=latitud && latitud>=350890 && (78122826)>=longitud && longitud>=(78114750))
{
tarjetaSD.print(F("LA MERCED"));
Serial.print(F("LA MERCED"));
lugar_inicio=String("La Merced");
}
else if(350890>=latitud && latitud>=346335 && (78150081)>=longitud && longitud>=(78137037))
{
tarjetaSD.print(F("PUGACHO"));
Serial.print(F("PUGACHO"));
lugar_inicio=String("Pugacho");
}
else if(350890>=latitud && latitud>=346335 && (78137037)>=longitud && longitud>=(78124887))
{
tarjetaSD.print(F("REDONDEL DE LA MADRE"));
Serial.print(F("REDONDEL DE LA MADRE"));
lugar_inicio=String("Redondel de la Madre");
}
else if(350890>=latitud && latitud>=343536 && (78124887)>=longitud && longitud>=(78115018))
{
tarjetaSD.print(F("CENTRO, MERCADO"));
Serial.print(F("CENTRO, MERCADO"));
lugar_inicio=String("Centro, Mercado");
}
else if(350236>=latitud && latitud>=337440 && (78115055)>=longitud && longitud>=(78104469))
{
tarjetaSD.print(F("LA VICTORIA"));
Serial.print(F("LA VICTORIA"));
lugar_inicio=String("La Victoria");
}

```

```

}
else if(346335>=latitud && latitud>=325640 && (78158825)>=longitud && longitud>=(78144588))
{
tarjetaSD.print(F("LA FLORIDA"));
Serial.print(F("LA FLORIDA"));
lugar_inicio=String("La Florida");
}
else if(346335>=latitud && latitud>=331637 && (78144588)>=longitud && longitud>=(78139387))
{
tarjetaSD.print(F("EL EJIDO"));
Serial.print(F("EL EJIDO"));
lugar_inicio=String("El Ejido");
}
else if(346335>=latitud && latitud>=331637 && (78139387)>=longitud && longitud>=(78132182))
{
tarjetaSD.print(F("PARQUE CENTRICA BULEVAR"));
Serial.print(F("PARQUE CENTRICA BULEVAR"));
lugar_inicio=String("P. Centrica Bulevar");
}
else if(346335>=latitud && latitud>=331637 && (78132182)>=longitud && longitud>=(78124887))
{
tarjetaSD.print(F("TERMINAL TERRESTRE"));
Serial.print(F("TERMINAL TERRESTRE"));
lugar_inicio=String("Terminal Terrestre");
}
else if(343536>=latitud && latitud>=331637 && (78124887)>=longitud && longitud>=(78122188))
{
tarjetaSD.print(F("YACUCALLE-PARQUE DE LA FAMILIA"));
Serial.print(F("YACUCALLE-PARQUE DE LA FAMILIA"));
lugar_inicio=String("Yacucalle");
}
else if(343536>=latitud && latitud>=331637 && (78122188)>=longitud && longitud>=(78115055))
{
tarjetaSD.print(F("LOS CEIBOS"));
Serial.print(F("LOS CEIBOS"));
lugar_inicio=String("Los Ceibos");
}
else if(337440>=latitud && latitud>=325342 && (78115052)>=longitud && longitud>=(78104469))
{
tarjetaSD.print(F("LA CAMPIÑA Y PRIMAVERA"));
Serial.print(F("LA CAMPIÑA Y PRIMAVERA"));
lugar_inicio=String("La Campiña, Primavera");
}
else if(331637>=latitud && latitud>=323215 && (78144588)>=longitud && longitud>=(78131269))
{
tarjetaSD.print(F("10 DE AGOSTO"));
Serial.print(F("10 DE AGOSTO"));
lugar_inicio=String("10 de Agosto");
}
else if(331637>=latitud && latitud>=320616 && (78131269)>=longitud && longitud>=(78115052))
{
tarjetaSD.print(F("CARANQUI"));
Serial.print(F("CARANQUI"));
lugar_inicio=String("Caranqui");
}
else if(323215>=latitud && latitud>=285107 && (78144588)>=longitud && longitud>=(78131269))
{
tarjetaSD.print(F("EJIDO DE CARANQUI"));
Serial.print(F("EJIDO DE CARANQUI"));
lugar_inicio=String("Ejido de Caranqui");
}
else if(320616>=latitud && latitud>=285107 && (78131269)>=longitud && longitud>=(78112202))
{
tarjetaSD.print(F("LA ESPERANZA"));
Serial.print(F("LA ESPERANZA"));
lugar_inicio=String("La Esperanza");
}
else if(402714>=latitud && latitud>=390194 && (78121280)>=longitud && longitud>=(78085592))
{
tarjetaSD.print(F("PRIORATO"));
Serial.print(F("PRIORATO"));
lugar_inicio=String("Priorato");
}
else if(356871>=latitud && latitud>=348197 && (78153938)>=longitud && longitud>=(78146508))
{
tarjetaSD.print(F("MILAGRO"));
Serial.print(F("MILAGRO"));
lugar_inicio=String("Milagro");
}
else if(344630>=latitud && latitud>=318537 && (78180878)>=longitud && longitud>=(78159813))
{
tarjetaSD.print(F("SAN ANTONIO"));
Serial.print(F("SAN ANTONIO"));
lugar_inicio=String("San Antonio");
}
else

```

```

    {
        tarjetaSD.print(F("NO DEFINIDO"));
        Serial.print(F("NO DEFINIDO"));
        lugar_inicio=String("No Definido");
    }
}

void lugar_llegada()
{
    if (gps.location.isValid())
    {
        long latitud=(gps.location.lat()*1000000;
        long longitud=abs(gps.location.lng()*1000000;
        Serial.print("latitud=");
        Serial.print(latitud);
        Serial.println();
        Serial.print("longitud=");
        Serial.print(longitud);
        Serial.println();
        if(390194>=latitud && latitud>=365508 && (78142340)>=longitud && longitud<=(78124423))
        {
            tarjetaSD.print(F("AZAYA"));
            Serial.print(F("AZAYA"));
            lugar_fin=String("Azaya");
        }
        else if(390194>=latitud && latitud>=377321 && (78124445)>=longitud && longitud<=(78113565))
        {
            tarjetaSD.print(F("AZAYA"));
            Serial.print(F("AZAYA"));
            lugar_fin=String("Azaya");
        }
        else if(377321>=latitud && latitud>=362311 && (78124445)>=longitud && longitud<=(78114343))
        {
            tarjetaSD.print(F("HUERTOS FAMILIARES"));
            Serial.print(F("HUERTOS FAMILIARES"));
            lugar_fin=String("Huertos Familiares");
        }
        else if(390194>=latitud && latitud>=369693 && (78114392)>=longitud && longitud<=(78085592))
        {
            tarjetaSD.print(F("YAHUARCOCHA"));
            Serial.print(F("YAHUARCOCHA"));
            lugar_fin=String("Yahuarcocha");
        }
        else if(369693>=latitud && latitud>=350236 && (78108460)>=longitud && longitud<=(78085592))
        {
            tarjetaSD.print(F("YAHUARCOCHA"));
            Serial.print(F("YAHUARCOCHA"));
            lugar_fin=String("Yahuarcocha");
        }
        else if(365551>=latitud && latitud>=355520 && (78146106)>=longitud && longitud<=(78130699))
        {
            tarjetaSD.print(F("ALPACHACA"));
            Serial.print(F("ALPACHACA"));
            lugar_fin=String("Alpachaca");
        }
        else if(365551>=latitud && latitud>=362311 && (78130699)>=longitud && longitud<=(78124412))
        {
            tarjetaSD.print(F("ALPACHACA"));
            Serial.print(F("ALPACHACA"));
            lugar_fin=String("Alpachaca");
        }
        else if(362289>=latitud && latitud>=355520 && (78130699)>=longitud && longitud<=(78121991))
        {
            tarjetaSD.print(F("HOSPITAL DEL SEGURO"));
            Serial.print(F("HOSPITAL DEL SEGURO"));
            lugar_fin=String("Hosp. del Seguro");
        }
        else if(362257>=latitud && latitud>=355520 && (78122170)>=longitud && longitud<=(78114354))
        {
            tarjetaSD.print(F("ESTADIO"));
            Serial.print(F("ESTADIO"));
            lugar_fin=String("Estadio");
        }
        else if(369693>=latitud && latitud>=350236 && (78114354)>=longitud && longitud>=(78108460))
        {
            tarjetaSD.print(F("EL OLIVO"));
            Serial.print(F("EL OLIVO"));
            lugar_fin=String("El Olivo");
        }
        else if(355595>=latitud && latitud>=350890 && (78148826)>=longitud && longitud>=(78132225))
        {
            tarjetaSD.print(F("JARDIN DE PAZ"));
            Serial.print(F("JARDIN DE PAZ"));
            lugar_fin=String("Jardin de Paz");
        }
    }
}

```

```

else if(355595>=latitud && latitud>=350890 && (78132225)>=longitud && longitud>=(78122826))
{
tarjetaSD.print(F("HOSPITAL SAN VICENTE DE PAUL"));
Serial.print(F("HOSPITAL SAN VICENTE DE PAUL"));
lugar_fin=String("Hosp. San Vicente de P.");
}
else if(355595>=latitud && latitud>=350890 && (78122826)>=longitud && longitud>=(78114750))
{
tarjetaSD.print(F("LA MERCED"));
Serial.print(F("LA MERCED"));
lugar_fin=String("La Merced");
}
else if(350890>=latitud && latitud>=346335 && (78150081)>=longitud && longitud>=(78137037))
{
tarjetaSD.print(F("PUGACHO"));
Serial.print(F("PUGACHO"));
lugar_fin=String("Pugacho");
}
else if(350890>=latitud && latitud>=346335 && (78137037)>=longitud && longitud>=(78124887))
{
tarjetaSD.print(F("REDONDEL DE LA MADRE"));
Serial.print(F("REDONDEL DE LA MADRE"));
lugar_fin=String("Redondel de la Madre");
}
else if(350890>=latitud && latitud>=343536 && (78124887)>=longitud && longitud>=(78115018))
{
tarjetaSD.print(F("CENTRO, MERCADO"));
Serial.print(F("CENTRO, MERCADO"));
lugar_fin=String("Centro, Mercado");
}
else if(350236>=latitud && latitud>=337440 && (78115055)>=longitud && longitud>=(78104469))
{
tarjetaSD.print(F("LA VICTORIA"));
Serial.print(F("LA VICTORIA"));
lugar_fin=String("La Victoria");
}
else if(346335>=latitud && latitud>=325640 && (78158825)>=longitud && longitud>=(78144588))
{
tarjetaSD.print(F("LA FLORIDA"));
Serial.print(F("LA FLORIDA"));
lugar_fin=String("La Florida");
}
else if(346335>=latitud && latitud>=331637 && (78144588)>=longitud && longitud>=(78139387))
{
tarjetaSD.print(F("EL EJIDO"));
Serial.print(F("EL EJIDO"));
lugar_fin=String("El Ejido");
}
else if(346335>=latitud && latitud>=331637 && (78139387)>=longitud && longitud>=(78132182))
{
tarjetaSD.print(F("PARQUE CENTRICA BULEVAR"));
Serial.print(F("PARQUE CENTRICA BULEVAR"));
lugar_fin=String("P. Centrica Bulevar");
}
else if(346335>=latitud && latitud>=331637 && (78132182)>=longitud && longitud>=(78124887))
{
tarjetaSD.print(F("TERMINAL TERRESTRE"));
Serial.print(F("TERMINAL TERRESTRE"));
lugar_fin=String("Terminal Terrestre");
}
else if(343536>=latitud && latitud>=331637 && (78124887)>=longitud && longitud>=(78122188))
{
tarjetaSD.print(F("YACUCALLE-PARQUE DE LA FAMILIA"));
Serial.print(F("YACUCALLE-PARQUE DE LA FAMILIA"));
lugar_fin=String("Yacucalle");
}
else if(343536>=latitud && latitud>=331637 && (78122188)>=longitud && longitud>=(78115055))
{
tarjetaSD.print(F("LOS CEIBOS"));
Serial.print(F("LOS CEIBOS"));
lugar_fin=String("Los Ceibos");
}
else if(337440>=latitud && latitud>=325342 && (78115052)>=longitud && longitud>=(78104469))
{
tarjetaSD.print(F("LA CAMPIÑA Y PRIMAVERA"));
Serial.print(F("LA CAMPIÑA Y PRIMAVERA"));
lugar_fin=String("La Campiña, Primavera");
}
else if(331637>=latitud && latitud>=323215 && (78144588)>=longitud && longitud>=(78131269))
{
tarjetaSD.print(F("10 DE AGOSTO"));
Serial.print(F("10 DE AGOSTO"));
lugar_fin=String("10 de Agosto");
}
else if(331637>=latitud && latitud>=320616 && (78131269)>=longitud && longitud>=(78115052))
{

```



```

tarjetaSD.print(F("CARANQUI"));
Serial.print(F("CARANQUI"));
lugar_fin=String("Caranqui");
}
else if(323215>=latitud && latitud>=285107 && (78144588)>=longitud && longitud>=(78131269))
{
tarjetaSD.print(F("EJIDO DE CARANQUI"));
Serial.print(F("EJIDO DE CARANQUI"));
lugar_fin=String("Ejido de Caranqui");
}
else if(320616>=latitud && latitud>=285107 && (78131269)>=longitud && longitud>=(78112202))
{
tarjetaSD.print(F("LA ESPERANZA"));
Serial.print(F("LA ESPERANZA"));
lugar_fin=String("La Esperanza");
}
else if(402714>=latitud && latitud>=390194 && (78121280)>=longitud && longitud>=(78085592))
{
tarjetaSD.print(F("PRIORATO"));
Serial.print(F("PRIORATO"));
lugar_fin=String("Priorato");
}
else if(356871>=latitud && latitud>=348197 && (78153938)>=longitud && longitud>=(78146508))
{
tarjetaSD.print(F("MILAGRO"));
Serial.print(F("MILAGRO"));
lugar_fin=String("Milagro");
}
else if(344630>=latitud && latitud>=318537 && (78180878)>=longitud && longitud>=(78159813))
{
tarjetaSD.print(F("SAN ANTONIO"));
Serial.print(F("SAN ANTONIO"));
lugar_fin=String("San Antonio");
}
else
{
tarjetaSD.print(F("NO DEFINIDO"));
Serial.print(F("NO DEFINIDO"));
lugar_fin=String("No Definido");
}
}
}

void ruta()
{
tarjetaSD=SD.open("DatosGPS.txt",FILE_WRITE);
if(tarjetaSD)
{
unsigned long timerLed1 = 0;
timerLed1 = millis();
while(millis() - timerLed1 < 1100)
{
if(Serial.available())
{
char a=Serial.read();
Serial1.print(a);
tarjetaSD.print(a);
}
if(Serial1.available())
{
char b=Serial1.read();
Serial.print(b);
tarjetaSD.print(b);
}
}
tarjetaSD.println();
tarjetaSD.close();
}
}

```