



# **UNIVERSIDAD TÉCNICA DEL NORTE**

## **FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**

### **CARRERA DE INGENIERÍA EN MANTENIMIENTO AUTOMOTRIZ**

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

#### **TEMA:**

**“IMPLEMENTACIÓN DE UN SISTEMA GPS PARA LA VERIFICACIÓN DE  
TIEMPOS DE VIAJE DE UN AUTOBÚS”**

#### **AUTOR:**

**PORTILLA YÉPEZ JEFERSON RICARDO**

#### **DIRECTOR:**

**ING. FREDY ROSERO MSC.**

**IBARRA, 2016**

## **ACEPTACIÓN DEL DIRECTOR**

En mi calidad de Director del plan de trabajo de grado, previo a la obtención del título de Ingeniería en Mantenimiento Automotriz, nombrado por el Honorable Consejo Directivo de la Facultad de Ingeniería en Ciencias Aplicadas.

### **CERTIFICO:**

Que una vez analizado el plan de grado cuyo título es “IMPLEMENTACIÓN DE UN SISTEMA GPS PARA LA VERIFICACIÓN DE TIEMPOS DE VIAJE DE UN AUTOBÚS” presentado por el señor: PORTILLA YÉPEZ JEFERSON RICARDO con número de cédula 100349595-7, doy fe que dicho trabajo reúne los requisitos y méritos suficientes para ser sometido a presentación pública y evaluación por parte de los señores integrantes del jurado examinador que se designe.

Atentamente:



---

ING. FREDY ROSERO MSC.  
DIRECTOR DEL TRABAJO DE GRADO



**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:

| <b>DATOS DE CONTACTO</b>    |           |  |            |
|-----------------------------|-----------|--|------------|
| <b>CÉDULA DE IDENTIDAD:</b> | <b>DE</b> | 100349595-7  |            |
| <b>APELLIDOS Y NOMBRES:</b> | <b>Y</b>  | PORTILLA YÉPEZ JEFERSON RICARDO                        |            |
| <b>DIRECCIÓN:</b>           |           | IBARRA, AV. MARIANO ACOSTA Y CALLE LUCIO TARQUINO PAEZ |            |
| <b>EMAIL:</b>               |           | jrportillay_@hotmail.com                               |            |
| <b>TELÉFONO FIJO:</b>       | 2630765   | <b>TELÉFONO MÓVIL:</b>                                 | 0985570135 |

| <b>DATOS DE LA OBRA</b>        |   |
|--------------------------------|---|
| <b>TÍTULO:</b>                 | IMPLEMENTACIÓN DE UN SISTEMA GPS PARA LA VERIFICACIÓN DE TIEMPOS DE VIAJE DE UN AUTOBÚS |
| <b>AUTOR (ES):</b>             | PORTILLA YÉPEZ JEFERSON RICARDO   |
| <b>FECHA: AAAAMMDD</b>         | 2016/10/05  |
| SOLO PARA TRABAJOS DE GRADO    |   |
| <b>PROGRAMA:</b>               | <input type="checkbox"/> <b>PREGRADO</b> <input type="checkbox"/> <b>POSGRADO</b>       |
| <b>TITULO POR EL QUE OPTA:</b> | INGENIERÍA EN MANTENIMIENTO AUTOMOTRIZ  |
| <b>ASESOR /DIRECTOR:</b>       | ING. FREDY ROSERO MSC.  |

**2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD**

Yo, **PORTILLA YÉPEZ JEFERSON RICARDO**, con cédula de identidad Nro. **100349595-7**, 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 5 días del mes de octubre del 2016.

Firma: .....

Nombre: PORTILLA YÉPEZ JEFERSON RICARDO

Cédula: 100349595-7



## UNIVERSIDAD TÉCNICA DEL NORTE

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

Yo, **PORTILLA YÉPEZ JEFERSON RICARDO**, con cédula de identidad Nro. **100349595-7**, 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: **“IMPLEMENTACIÓN DE UN SISTEMA GPS PARA LA VERIFICACIÓN DE TIEMPOS DE VIAJE DE UN AUTOBÚS”**, 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 5 días del mes de octubre de 2016.

Firma: .....

Nombre: PORTILLA YÉPEZ JEFERSON RICARDO

Cédula: 100349595-7

## DEDICATORIA

A mis Padres Manuel Portilla y Narcisa Yépez, en especial a mi madre por haberme apoyado en todo momento, por sus palabras de ánimo, sus valores, por estar pendiente de todo lo que me sucede, por ser mi mejor amiga, por su esfuerzo y dedicación como madre, pero más que nada, por su amor.

A mis abuelitos, por los ejemplos de perseverancia por permitirme compartir con ellos parte de mi vida.

A mis hermanos Bryan y Nasly, por permitirme ser para ellos un ejemplo de superación.

A mi tía Mercedes Yépez por tanto apoyo confianza y cariño por desear y querer siempre lo mejor para mis hermanos y para mí.

También se la dedico a una mujer especial Joselyn Vásquez por su cariño comprensión y confianza depositada en mí, por ser la persona que ha estado en los momentos de alegría y tristeza, brindándome su apoyo y motivación para ser mejor persona.

## **AGRADECIMIENTO**

A Dios por brindarme la vida protegerme darme fortaleza sabiduría y perseverancia para ser cada día mejor persona, guiándome siempre por un camino de bien, por permitirme cumplir uno de mis mayores objetivos en mi vida universitaria.

A mis padres Narcisa y Manuel por brindarme su apoyo incondicional en todo momento y depositar su confianza total en mí, por enseñarme y demostrarme que todo lo que son y hacen es por el bien y superación de mis hermanos y mío.

A mis abuelitos María Luz y Víctor por sus consejos, apoyo moral y económico, siendo un pilar fundamental para culminar con esta propuesta en mi vida. De igual manera a mi abuelita Anita por brindarme sus mejores deseos oraciones y bendiciones siempre.

A toda mi familia que estuvo pendiente dándome ánimos y apoyo incondicional, de manera especial a mi tía Mercedes y mi prima Fernanda por entregarme su amor infinito, por estar siempre a mi lado porque admiro su fortaleza para ayudar siempre y brindar lo mejor de ellas sin esperar nada a cambio.

Mis más altos sentimientos de gratitud y estima para el Ing. Fredy Rosero Msc. Docente de la Facultad de Ingeniería en Ciencias Aplicadas, director de la investigación quien durante la realización de este presente trabajo supo ofrecerme su mano amiga con paciencia confianza y dedicación.

## ÍNDICE

|   |          |
|---|----------|
| PORTADA.....                              | i        |
| CERTIFICACIÓN .....                       | ii       |
| AUTORIZACIÓN DE USO Y PUBLICACIÓN.....    | iii      |
| CESIÓN DE DERECHOS DEL AUTOR.....         | v        |
| DEDICATORIA.....                          | vi       |
| AGRADECIMIENTO .....                      | vii      |
| ÍNDICE .....                              | viii     |
| ÍNDICE DE FIGURAS .....                   | xiii     |
| ÍNDICE DE TABLAS .....                    | xvi      |
| ÍNDICE DE ABREVIATURAS .....              | xvii     |
| GLOSARIO DE TÉRMINOS.....                 | xviii    |
| RESUMEN .....                             | xx       |
| ABSTRACT .....                            | xxi      |
| INTRODUCCIÓN .....                        | xxii     |
| <br>                                      |          |
| <b>CAPÍTULO I.....</b>                    | <b>1</b> |
| <b>1. PROBLEMA DE INVESTIGACIÓN. ....</b> | <b>1</b> |
| 1.1 Antecedentes. ....                    | 1        |
| 1.2 Planteamiento del problema.....       | 1        |
| 1.3 Formulación del problema.....         | 2        |
| 1.4 Delimitación.....                     | 2        |
| 1.5 Objetivos. ....                       | 2        |
| 1.5.1 Objetivo general. ....              | 2        |
| 1.5.2 Objetivos específicos. ....         | 2        |
| 1.6 Justificación.....                    | 3        |



|   |    |
|---|----|
| <b>CAPÍTULO II</b> .....  | 4  |
| <b>2. MARCO TEÓRICO</b> .....                                   | 4  |
| 2.1 Servicio de bus interprovincial.....                        | 4  |
| 2.1.1 Características. ....                                     | 5  |
| 2.1.2 Diferencia entre transporte interprovincial y urbano..... | 5  |
| 2.2 Sistema de Posicionamiento Global (GPS).....                | 6  |
| 2.2.1 Composición del GPS .....                                 | 7  |
| 2.2.2 Cobertura terrestre de los satélites .....                | 7  |
| 2.2.3 Funcionamiento.....                                       | 8  |
| 2.2.4 Medición de la distancia de un satélite.....              | 9  |
| 2.2.5 Precisión .....   | 9  |
| 2.2.6 Precisión horizontal.....                                 | 10 |
| 2.2.7 Precisión vertical .....                                  | 12 |
| 2.2.8 Fuentes de error.....                                     | 13 |
| 2.2.9 Receptor GPS .....  | 14 |
| 2.2.10 Formato de presentación de datos.....                    | 14 |
| 2.3 Arduino.....  | 14 |
| 2.3.1 Origen de arduino .....                                   | 16 |
| 2.3.2 Arduino software libre.....                               | 16 |
| 2.3.3 Arduino hardware libre .....                              | 16 |
| 2.3.4 Ventajas de elegir Arduino .....                          | 17 |
| 2.3.5 Entorno de programación.....                              | 17 |
| 2.4 LabVIEW.....  | 19 |
| 2.4.1 Principales características.....                          | 19 |
| 2.4.2 Programación.....   | 20 |
| 2.4.3 Macros de Excel.....                                      | 21 |
| 2.4.3.1 Lenguaje Visual Basic.....                              | 22 |

|  |           |
|--|-----------|
| 2.4.3.2 VBA y Visual Basic (VB).....           | 23        |
| <b>CAPÍTULO III.....</b>                       | <b>24</b> |
| <b>3. METODOLOGÍA DE LA INVESTIGACIÓN.....</b> | <b>24</b> |
| 3.1 Tipo de investigación.....                 | 24        |
| 3.1.1 Investigación Bibliográfica.....         | 24        |
| 3.1.2 Investigación Tecnológica.....           | 24        |
| 3.2 Métodos.....                               | 24        |
| 3.3 Técnicas e instrumentos.....               | 25        |
| <b>CAPÍTULO IV.....</b>                        | <b>26</b> |
| <b>4. PROPUESTA: PROCESO Y RESULTADOS.....</b> | <b>26</b> |
| 4.1 Etapas de diseño.....                      | 26        |
| 4.2 Selección de elementos electrónicos.....   | 27        |
| 4.2.1 Placa Arduino.....                       | 27        |
| 4.2.1.1 Especificaciones.....                  | 28        |
| 4.2.1.2 Programación.....                      | 28        |
| 4.2.1.3 Alimentación.....                      | 28        |
| 4.2.1.4 Memoria.....                           | 29        |
| 4.2.1.5 Comunicación.....                      | 29        |
| 4.2.1.6 Características físicas.....           | 30        |
| 4.2.2 Módulo GPS.....                          | 30        |
| 4.2.2.1 Diagrama de bloques GPS.....           | 31        |
| 4.2.2.2 Tiempo de precisión.....               | 31        |
| 4.2.2.3 Protocolos disponibles.....            | 31        |
| 4.2.2.4 Asignación de pines.....               | 32        |
| 4.2.2.5 Especificaciones mecánicas.....        | 33        |
| 4.2.3 Pantalla.....                            | 34        |

|   |    |
|---|----|
| 4.2.3.1 Especificaciones.....                                 | 35 |
| 4.2.3.2 Pines. ....   | 36 |
| 4.2.3.5 Shield TFT 3.2”.....                                  | 36 |
| 4.3 Implementación de hardware y software de Arduino. ....    | 38 |
| 4.3.1 Diagrama de conexión del módulo GPS. ....               | 38 |
| 4.3.2 Diagrama de conexión de la pantalla. ....               | 41 |
| 4.3.3 Diagrama electrónico completo.....                      | 44 |
| 4.4 Definición de los lugares de control. ....                | 46 |
| 4.5 Flujogramas del dispositivo. ....                         | 47 |
| 4.5.1 Flujograma subrutina menú.....                          | 48 |
| 4.5.2 Flujograma subrutina viajes. ....                       | 49 |
| 4.5.3 Flujograma subrutina funcionamiento. ....               | 50 |
| 4.5.4 Flujograma subrutina almacenamiento de datos.....       | 51 |
| 4.5.5 Flujograma subrutina home.....                          | 52 |
| 4.6 Diseño del software de análisis de datos en LabVIEW. .... | 52 |
| 4.6.1 Portada.....  | 53 |
| 4.6.2 Atrasos/Adelantos. ....                                 | 53 |
| 4.6.3 Estadísticas.....                                       | 56 |
| 4.6.4 Velocidades.....  | 58 |
| 4.7 Pruebas de funcionamiento.....                            | 63 |
| 4.7.1 Pruebas del Hardware.....                               | 63 |
| 4.7.1.1 Pruebas de tiempos de Adelanto y Atraso. ....         | 63 |
| 4.7.1.2 Pruebas de velocidad.....                             | 66 |
| 4.7.2 Pruebas del Software.....                               | 67 |
| 4.8 Manual de funcionamiento. ....                            | 69 |
| 4.8.1 Manual del dispositivo.....                             | 69 |
| 4.8.1.1 Identificación de botones.....                        | 69 |

|   |           |
|---|-----------|
| 4.8.1.2 Selección de viajes.....              | 70        |
| 4.8.1.3 Funcionamiento en viaje. ....         | 72        |
| 4.8.2 Manual del Software LabVIEW. ....       | 74        |
| <b>CAPÍTULO V. ....</b>                       | <b>78</b> |
| <b>5. CONCLUSIONES Y RECOMENDACIONES.....</b> | <b>78</b> |
| 5.1 Conclusiones.....                         | 78        |
| 5.2 Recomendaciones. ....                     | 79        |
| <b>BIBLIOGRAFÍA .....</b>                     | <b>81</b> |
| <b>ANEXOS.....</b>                            | <b>82</b> |

## ÍNDICE DE FIGURAS

|  |    |
|--|----|
| Figura 1. Transporte interprovincial.....                          | 4  |
| Figura 2. Satélite.....  | 6  |
| Figura 3. Constelación de satélites.....                           | 7  |
| Figura 4. Visibilidad de los satélites.....                        | 8  |
| Figura 5. Principio de la situación por satélite.....              | 9  |
| Figura 6. Precisión horizontal en función de la latitud.....       | 10 |
| Figura 7. Distribución del error este-oeste y norte-sur.....       | 11 |
| Figura 8. Probabilidad del error horizontal.....                   | 11 |
| Figura 9. Observación del error vertical durante 2h30 minutos..... | 12 |
| Figura 10. Precisión vertical según la latitud.....                | 13 |
| Figura 11. Entorno de programación.....                            | 18 |
| Figura 12. Panel frontal.....                                      | 21 |
| Figura 13. Diagrama de bloques.....                                | 21 |
| Figura 14. Conjunto de instrucciones escritas en VBA.....          | 22 |
| Figura 15. Módulo de VBA.....                                      | 23 |
| Figura 16. Etapas de diseño.....                                   | 26 |
| Figura 17. Arduino Mega 2560.....                                  | 27 |
| Figura 18. Módulo GPS.....   | 30 |
| Figura 19. Diagrama de bloques módulo GPS.....                     | 31 |
| Figura 20. Distribución de pines GPS.....                          | 33 |
| Figura 21. Especificaciones mecánicas GPS.....                     | 33 |
| Figura 22. Pantalla TFT 3.2".....                                  | 35 |
| Figura 23. Diagrama de pines TFT 3.2".....                         | 36 |
| Figura 24. Distribución de pines Shield TFT 3.2".....              | 37 |
| Figura 25. Diagrama de conexión módulo GPS.....                    | 38 |

|  |    |
|--|----|
| Figura 26. Código limitación de áreas de control. ....           | 39 |
| Figura 27. Código de velocidad, hora y fecha. ....               | 39 |
| Figura 28. Código velocidad máxima. ....                         | 40 |
| Figura 29. Código velocidad mínima. ....                         | 40 |
| Figura 30. Diagrama de conexión de la pantalla. ....             | 42 |
| Figura 31. Código control touch. ....                            | 43 |
| Figura 32. Código almacenamiento de datos. ....                  | 43 |
| Figura 33. Diagrama electrónico completo. ....                   | 44 |
| Figura 34. Código declaración de elementos electrónicos. ....    | 45 |
| Figura 35. Código void setup. ....                               | 45 |
| Figura 36. Delimitación de los lugares de control. ....          | 46 |
| Figura 37. Flujograma general. ....                              | 48 |
| Figura 38. Flujograma subrutina menú. ....                       | 49 |
| Figura 39. Flujograma viajes. ....                               | 49 |
| Figura 40. Flujograma subrutina funcionamiento. ....             | 51 |
| Figura 41. Flujograma subrutina almacenamiento de datos. ....    | 51 |
| Figura 42. Flujograma subrutina home. ....                       | 52 |
| Figura 43. Flujograma software análisis de datos. ....           | 52 |
| Figura 44. Portada programa análisis de datos. ....              | 53 |
| Figura 45. Botones para la conexión con la SD. ....              | 54 |
| Figura 46. Código para selección de cooperativa y vehículo. .... | 54 |
| Figura 47. Ventana de datos generales. ....                      | 55 |
| Figura 48. Código para la importación de datos. ....             | 55 |
| Figura 49. Selección de viajes. ....                             | 56 |
| Figura 50. Selección de número de viaje. ....                    | 56 |
| Figura 51. Código para la selección de viajes. ....              | 57 |
| Figura 52. Tabla de adelantos y atrasos. ....                    | 57 |

|  |    |
|--|----|
| Figura 53. Código para la importación de viajes. ....                    | 58 |
| Figura 54. Ventana de velocidades. ....                                  | 59 |
| Figura 55. Tabla de velocidad máxima. ....                               | 59 |
| Figura 56. Código para importar la velocidad máxima. ....                | 60 |
| Figura 57. Cuadro de velocidad mínima. ....                              | 60 |
| Figura 58. Código para importar la velocidad máxima. ....                | 61 |
| Figura 59. Botón de representación gráfica. ....                         | 61 |
| Figura 60. Código para la ejecución de Power BI. ....                    | 62 |
| Figura 61. Representación gráfica de velocidades máximas y mínimas. .... | 62 |
| Figura 62. Tiempos de adelanto y atraso Ibarra-Quito. ....               | 67 |
| Figura 63. Velocidades máximas y mínimas Ibarra-Quito. ....              | 68 |
| Figura 64. Presentación gráfica de velocidades Ibarra-Quito. ....        | 68 |
| Figura 65. Instalación del dispositivo. ....                             | 69 |
| Figura 66. Botones de control. ....                                      | 70 |
| Figura 67. Ventana de selección de viajes. ....                          | 71 |
| Figura 68. Ventana de funcionamiento del dispositivo. ....               | 71 |
| Figura 69. Funcionamiento del dispositivo. ....                          | 72 |
| Figura 70. Archivo viajes.csv. ....                                      | 73 |
| Figura 71. Archivos vmax.csv y vmin.csv. ....                            | 74 |
| Figura 72. Programa de análisis de datos. ....                           | 74 |
| Figura 73. Ventana de datos generales. ....                              | 75 |
| Figura 74. Ventana de adelantos y atrasos. ....                          | 75 |
| Figura 75. Ventana de velocidades. ....                                  | 76 |
| Figura 76. Ventana de representación gráfica. ....                       | 77 |

## ÍNDICE DE TABLAS

|  |    |
|--|----|
| Tabla 1. Valores Corrientes de las fuentes de error..... | 13 |
| Tabla 2. Especificaciones de los microcontroladores..... | 15 |
| Tabla 3. Opciones de la barra de herramientas. ....      | 18 |
| Tabla 4. Especificaciones Arduino Mega. ....             | 28 |
| Tabla 5.Pines GPS.....                                   | 32 |
| Tabla 6. Simbología especificaciones mecánicas. ....     | 34 |
| Tabla 7. Pines de conexión Shield TFT 3.2". ....         | 37 |
| Tabla 8. Lugares de control.....                         | 46 |
| Tabla 9. Tiempos de viaje Ibarra-Quito. ....             | 63 |
| Tabla 10. Tiempos de viaje Ibarra-Quito. ....            | 64 |
| Tabla 11. Prueba de velocidad.....                       | 66 |
| Tabla 12. Presentación de tiempos de control. ....       | 67 |



## ÍNDICE DE ABREVIATURAS

|       |  |
|-------|--|
| A     | Amperios.  |
| AREF  | Referencia de tensión del convertidor analógico a digital.   |
| AVR   | Microcontroladores de fabricante estadounidense.             |
| DC    | Corriente continua.  |
| EPROM | Memoria ROM programable borrable.                            |
| g     | Gramos.  |
| GND   | Conexión a tierra.   |
| GPS   | Sistema de Posicionamiento Global.                           |
| IDE   | Entorno de desarrollo integrado.                             |
| IOREF | Tensión de entrada y salida de la placa Arduino.             |
| 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.                              |
| 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. |
| TTL   | Lógica transistor a transistor.                              |
| TX    | Transmisión de datos.  |
| UART  | Transmisor, receptor asíncrono universal.                    |
| UPS   | Universal Polar Stereographic.                               |
| USB   | Bus universal en serie.                                      |
| UTM   | Sistema de coordenadas universal transversal de Mercator.    |
| V     | Voltios.   |
| Vi    | Instrumento virtual.   |

## GLOSARIO DE TÉRMINOS.

**Actuadores.** Son dispositivos que transforman la energía eléctrica para la ejecución de un proceso.

**Adquisición.** Obtención de datos de muestra para generar información que pueda ser analizada por algún ordenador.

**Señales analógicas.** Señales producidas por algún fenómeno electromagnético que puede ser representada por una función matemática.

**Señales digitales.** Señales cuyo contenido puede ser codificado según los signos que contiene y que representan valores discretos.

**Electromagnética.** Cantidad de energía almacenada en un espacio con campo electromagnético que se expresa en función de la intensidad del campo.

**Electrónica.** Rama de la ingeniería que se encarga de estudiar la conducciones del flujo de electrones.

**Sistema Embebido.** Sistema diseñado para realizar algunas funciones referentes a sistemas de computación.

**Fiabilidad.** Es la probabilidad que tiene algún sistema para funcionar correctamente en un determinado proceso.

**Hardware.** Es un conjunto de elementos que forman parte de un sistema de computadora.

**Instrucciones.** Conjunto de acciones que se definen a un computador con la finalidad de efectuar una operación.

**Interacción.** Acción recíproca entre varios objetos.

**Interfaz.** Elementos capaces de transformar señales realizadas por algún dispositivo en señales perceptibles por otros.

**Latitud.** Distancia angular que existe desde algún lugar de la superficie terrestre hasta el paralelo del Ecuador.

**Longitud.** Distancia angular que existe desde algún lugar de la superficie terrestre hasta el meridiano cero.

**Microcontroladores.** Circuitos integrados que pueden realizar instrucciones grabadas en su memoria.

**Multiplataforma.** Programas informáticos que tiene la característica de ser ejecutados en varios sistemas computacionales.

**Robótica.** Técnica utilizada para el diseño y construcción de robots que realizan trabajos generalmente industriales.

**Satélite.** Vehículo espacial que se mantiene en una órbita alrededor del planeta.

**Sensores.** Dispositivos capaces de detectar magnitudes físicas para transformarlas en variables eléctricas.

**Sistema operativo.** Conjunto de programas que tienen la capacidad de controlar procesos de un computador para el funcionamiento de otros programas.

**Tecnología.** Conjunto de conocimientos técnicos que son utilizados para el diseño de servicios adaptables al medio ambiente.

## RESUMEN

Desde años atrás y hasta la actualidad se puede observar que las diferentes cooperativas de transporte interprovincial controlan el tiempo de viaje de cada una de sus unidades únicamente definiendo la hora de salida y llegada a su destino, provocando que los usuarios no obtengan un buen servicio debido a que no conocen la hora específica a la que debe pasar el vehículo a través de los diferentes lugares ubicados entre la ruta a seguir, esto se debe a que no se dispone de algún dispositivo que permita controlar los diferentes tiempos. Además, al no contar con alguno de estos sistemas no es posible registrar los viajes realizados por el autobús que nos permita obtener información estadística para conocer el tiempo y los lugares de espera que realizan en los diferentes viajes. El dispositivo es desarrollado a través de la plataforma de hardware libre Arduino, en el que mediante la programación es posible recolectar información brindada durante su funcionamiento, permitiendo que las autoridades pertinentes tengan conocimiento de los lugares en los que el autobús realiza esperas y adelantos, y de esta manera imponer la respectiva sanción al responsable del vehículo. Además, con el registro de la velocidad máxima y mínima de viaje del autobús se podrá brindar mayor seguridad a los usuarios y al chofer de la respectiva unidad. La hora de cruce por los diferentes lugares de control son registrados a través del funcionamiento del módulo GPS instalado en el dispositivo, tomando en cuenta valores de longitud y latitud. Para realizar un análisis estadístico se desarrolla un programa de análisis de datos en el que se observa información de tiempos totales de adelantos y atrasos, como también información de cada viaje realizado en un determinado tiempo, además de registrar las velocidades máximas y mínimas y de su respectiva representación gráfica a través del programa Power BI.

## **ABSTRACT**

Years ago and even today it can be seen that the different interprovincial transport cooperatives control the travel time of each of its units only by defining the time of departure and arrival at destination, causing that users do not get good service because they do not know the specific time at which the vehicle must pass through different places located between the route to follow, this is because they not have any control device that allows them determine those different times. Also, by not having any of those systems it is not possible to record the trips of each bus to allow us to obtain statistical information to determine the time and waiting places made in the different trips. The device is developed through the free hardware platform Arduino in which by programming it is possible to collect information provided during its operation, allowing that the relevant authorities be aware of the places where the bus takes pauses and advances. In this way impose the respective sanction to the responsible of the vehicle. In addition, by registering the maximum and minimum travel speed of bus it may provide greater security to users and the driver of the respective unit. The crossing different control sites are registered through the operation of the GPS module installed on the device, taking into account latitude and longitude values. To perform the statistician analysis it is developed a data analysis program in which it is observed the Total information related to advances and delays, as well as information of each trip in a certain time apart from recording the maximum and minimum speeds and its respective graphical representation through the Power BI program.

## INTRODUCCIÓN

El presente trabajo de grado tiene como principal objetivo el diseño de un dispositivo para el control de tiempo de viaje de un autobús, que está desarrollado a través de la plataforma de hardware libre Arduino y de un módulo GPS con el que se registra la hora de cruce del vehículo por los diferentes lugares de control.

De esta manera es posible almacenar en la tarjeta de memoria incorporada la información de los diferentes viajes realizados para su posterior análisis en el programa creado a través del software LabVIEW. El trabajo de grado se realiza en la Universidad Técnica del Norte y el área de aplicación es en las unidades de transporte interprovincial de la ciudad de Ibarra.

En el capítulo primero se analiza el problema de investigación conformado por los antecedentes, planteamiento y formulación del problema, así como de los objetivos generales y específicos, finalizando con la justificación.

En el segundo capítulo se presenta el marco teórico, en el que se abordan conceptos utilizados en la investigación, de esta manera se indica la conformación del sistema GPS y de su principio de funcionamiento, así como también de la precisión que los receptores poseen y de las causas de los posibles errores al momento de definir una posición. Además, como el dispositivo es desarrollado a través de Arduino se presentan conceptos y forma de programación mediante esta plataforma; finalizando con definiciones del software LabVIEW debido a que a través de este se desarrolla el programa de análisis de datos.

En el capítulo tercero se indica la metodología aplicada en el desarrollo del trabajo de grado, indicando el tipo de investigación utilizado, así como también de los métodos y técnicas aplicadas en su realización.

En el cuarto capítulo se presenta el desarrollo del dispositivo, indicando su diseño físico, así como también las conexiones realizadas para su buen

funcionamiento. Se muestra también los códigos realizados en el software LabVIEW para el funcionamiento del programa de análisis de datos y de las respectivas ventanas de presentación de la información estadística. Además, se presenta las pruebas de funcionamiento realizadas del dispositivo y del software a través de datos de muestra realizados en diferentes viajes.

En el quinto y último capítulo se muestra las conclusiones y recomendaciones obtenidas durante el desarrollo del dispositivo, finalizando con la bibliografía y anexos de importancia como el código de programación.

# **CAPÍTULO I.**

## **1. PROBLEMA DE INVESTIGACIÓN.**

### **1.1 Antecedentes.**

Desde hace años atrás las diferentes cooperativas de transporte interprovincial controlan el tiempo de viaje de cada una de sus unidades únicamente definiendo la hora de salida y llegada a su destino, lo que hace que los usuarios no obtengan un buen servicio debido a que no conocen la hora específica a la que debe pasar el vehículo a través de los diferentes lugares ubicados entre la ruta a seguir.

En la actualidad no se encuentran dispositivos que se hayan desarrollado para mejorar este tipo de control y así brindar un mejor servicio a los usuarios de las diferentes cooperativas. Además, al no contar con alguno de estos sistemas no es posible registrar los viajes realizados por el vehículo que nos permita obtener información estadística para conocer el tiempo y los lugares de espera que realizan los autobuses en los diferentes viajes.

### **1.2 Planteamiento del problema.**

Al no incorporar un dispositivo para el control de tiempo en los autobuses no se podrá realizar mejoras en el servicio brindado a los usuarios, debido a que no conocerán el tiempo de viaje a su destino ocasionando de esta manera molestias entre ellos y el conductor.



La información recolectada y brindada a través del funcionamiento del dispositivo permitirá a autoridad responsable de la cooperativa conocer los tiempos y los lugares en los que el autobús realiza esperas y adelantos, y de esta manera imponer la respectiva sanción al responsable del vehículo.

Además, al registrar la velocidad máxima y mínima de viaje del autobús se podrá brindar mayor seguridad a los usuarios y al chofer de la respectiva unidad.

### **1.3 Formulación del problema.**

¿Cómo implementar un sistema GPS para la verificación de tiempos de viaje de un autobús?

### **1.4 Delimitación.**

El presente proyecto se llevará a cabo desde el mes de septiembre hasta el mes de junio, en la Universidad Técnica del Norte en la ciudad de Ibarra, provincia de Imbabura.

### **1.5 Objetivos.**

#### **1.5.1 Objetivo general.**

1. Implementar un sistema GPS para la verificación de tiempos de viaje de un autobús.

#### **1.5.2 Objetivos específicos.**

1. Construir el dispositivo a través de la plataforma de hardware libre Arduino.
2. Utilizar el módulo GPS para la identificación de lugares de control.

3. Desarrollar un programa en el software LabVIEW para el análisis de datos.

### **1.6 Justificación.**

Con la incorporación de un dispositivo que permita controlar el tiempo de viaje de los autobuses de las diferentes cooperativas de transporte interprovincial, los choferes podrán conocer en tiempo real la velocidad y tiempo de atraso o adelanto en los diferentes lugares situados entre la ruta de cada viaje y así brindar un mejor servicio a sus usuarios debido a que podrán conocer el tiempo de llegada a su destino.

Además, con el dispositivo empleado se podrá almacenar datos específicos que permitan obtener información estadística sobre los atrasos o adelantos realizados durante cada viaje en un periodo determinado, así como también registrar los aumentos y disminución de velocidad, datos que al ser subidos al programa de análisis de datos creado permitirán conocer a través de un mapa los lugares en los que el vehículo realiza este tipo de maniobras.

## CAPÍTULO II.

### 2. MARCO TEÓRICO.

#### 2.1 Servicio de bus interprovincial.

El servicio de transporte interprovincial esta aplicado al transporte colectivo de pasajeros. A diferencia de un transporte privado, los usuarios están obligados a respetar las frecuencias y rutas que ofrecen las diferentes cooperativas existentes; habitualmente los viajeros comparten el medio de transporte y las diferentes unidades están disponibles para el público en general. (Morales Costa & Flores Fernandez, 2009)

Este tipo de transporte esta proporcionado por una o varias cooperativas que prestan el servicio a través del cobro directo a los pasajeros, que normalmente es regulado por las autoridades de la ANT (Agencia Nacional de Tránsito) a nivel nacional. Por eventos históricos, económicos, políticos y sociales, el transporte público de ciertos países es diferente al de otros; por ejemplo, en las ciudades de Europa tiene varios y frecuentes servicios a diferentes ciudades con población densa, mientras que en zonas como América disponen de sistemas de transporte menos complejos.



Figura 1. Transporte interprovincial.

### **2.1.1 Características.**

Dentro de las características que posee un sistema de transporte interprovincial, la demanda está condicionada por los pasajeros y la oferta por la disponibilidad de vehículos, infraestructura, servicios prestados y los conductores. Lo que se diferencia de un transporte privado en la que la demanda es la persona en el vehículo, mientras que la oferta viene a ser las vías de circulación.

El transporte de pasajeros esta evaluado de diferente manera por parte de los usuarios o trabajadores, además, las frecuencias y recorridos que realiza cada autobús puede ser diferente para los ciudadanos que se encuentran al inicio de la ciudad y para los que se encuentran al final. Esto hace que la comprensión de tránsito puede ser más pertinente al momento que se apele gran variedad de perspectivas.

### **2.1.2 Diferencia entre transporte interprovincial y urbano.**

La diferencia que existe entre el transporte interprovincial y el transporte urbano está en que este último permite el desplazamiento de pasajeros únicamente de un lugar a otro dentro de la ciudad por zonas específicas. Es una opción de disminución de contaminación debido a que permite el desplazamiento de varias personas en un solo viaje y no necesitan recorrer distancias largas, movilizandó además a personas que no poseen vehículos.

El transporte urbano además tiene un costo por pasajero fijo lo que se diferencia del transporte interprovincial cuyo costo por varía según la distancia que tiene que recorrer para llegar a su destino. Los horarios de recorridos también son muy diferentes debido a que las unidades de transporte interprovincial únicamente se guían por hora de salida y llegada a las diferentes ciudades, además, no disponen de viajes frecuentes a diferencia del transporte urbano.

## 2.2 Sistema de Posicionamiento Global (GPS).

Este sistema aparece en 1978 con el objetivo de reemplazar a los sistemas GPS con demasiado bajo rendimiento de esa época, siendo ideado por el Departamento de Defensa de los Estados Unidos. No obstante, solo fue accesible a la mayoría de la población a mediados del año 80 y en menos de 10 años abarcó por completo hacia los demás sistemas existentes. (Correia, 2010)

Está compuesto por 24 satélites distribuidos alrededor de la Tierra a 20000 kilómetros de altitud. Estos satélites envían de forma permanente señal a los receptores GPS para determinar su posición ya sea en mar, aire o tierra.



Figura 2. Satélite.

Fuente: (Correia, 2010)

Para el uso civil, el GPS permite determinar una ubicación con un error inferior a 100 metros, pero también hay que tomar en cuenta que en ciertos casos la posición puede ser muy inferior a lo establecido. Así también, en algunas horas del día, en un cinco por ciento de los casos, esta precisión puede tener una degradación de hasta 300 metros cuando las condiciones son adversas.

El GPS presenta un gran avance tecnológico para las personas que necesitan saber su ubicación con mucha precisión, convirtiéndose de esta manera en un instrumento insustituible para muchos profesionales como también para algunas actividades. (Correia, 2010)

### 2.2.1 Composición del GPS

Este sistema está compuesto por tres segmentos distintos pero indispensables para su correcto funcionamiento, estos segmentos son:

- a) Primer segmento. Formado por el conjunto de satélites, también es conocido como segmento espacial.
- b) Segundo segmento. Constituida por los segmentos de control.
- c) Tercer segmento. Está conformado por los receptores GPS.

### 2.2.2 Cobertura terrestre de los satélites

La distribución de los satélites hace que sea posible obtener en el 99.9% de las situaciones un mínimo de cuatro satélites con visibilidad a 5° o más por alto del horizonte, con una disminución de la precisión igual o inferior a seis, independiente del lugar en el que se encuentre sobre la Tierra. Así también, esto significa que en un minuto por día, el sistema de Posicionamiento Global puede tener una posición incorrecta. (Correia, 2010)

Además, si el lugar no se encuentra despejado, la cobertura disponible de los satélites puede ser insuficiente en su totalidad. En algunos lugares del planeta, la cobertura brindada por los satélites no es muy segura en más del 3% en unos 45 minutos por día.

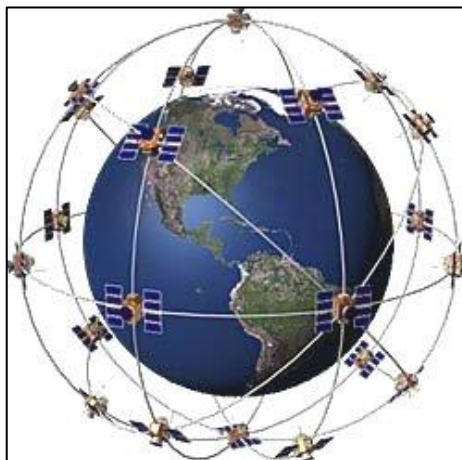


Figura 3. Constelación de satélites.

Fuente: (Correia, 2010)

En algunos de los casos también sucede debido a que uno o varios satélites se encuentran por cierto tiempo fuera de servicio, que por lo general es provocado por un mantenimiento del satélite o un problema técnico. En ningún caso, algunos de los satélites se encuentran fuera de servicio por más de 24 horas, no existen cuatros satélites fuera de servicio por un mes, ni tampoco tres al mismo tiempo. El número de satélites visibles varía durante un día, teniendo 8 como promedio y el porcentaje de ellos visibles durante las 24 horas se puede observar en la figura 4.

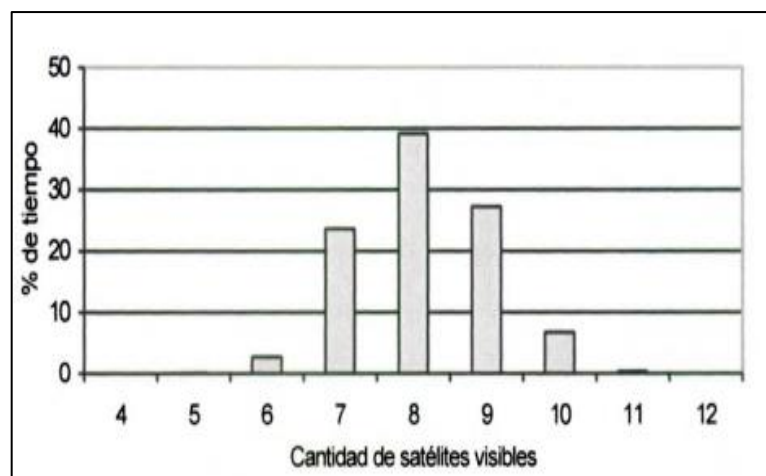


Figura 4. Visibilidad de los satélites.

Fuente: (Correia, 2010)

Los satélites no siempre se encuentran visibles debido a que no se encuentran en la órbita indicada o que ésta se encuentre oscilando debido a fenómenos imprevistos.

### 2.2.3 Funcionamiento

El funcionamiento del Sistema GPS para determinar una ubicación en el planeta se basa en la distancia entre el receptor y los satélites. Cada uno de ellos transmiten su posición en cada instante en las orbitas alrededor de la Tierra, así también, a la misma vez la hora y fecha exacta.

De esta manera se calcula el tiempo que necesitan las señales para llegar a la Tierra y así establecer la distancia entre ellos. Con la distancia y posición del

satélite se puede trazar varios círculos imaginarios en cuya intersección se encuentra el receptor, esto se puede observar en la siguiente figura.

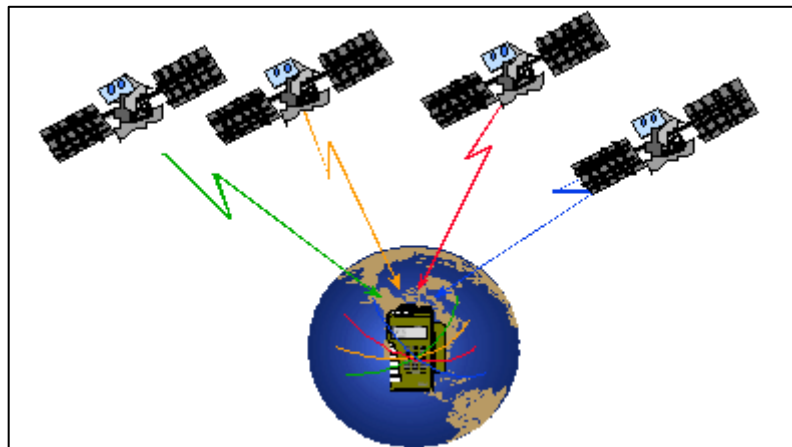


Figura 5. Principio de la situación por satélite.

Fuente: (Correia, 2010)

#### 2.2.4 Medición de la distancia de un satélite

La medición de la distancia entre un satélite y el receptor se basa en la propagación de ondas electromagnéticas. El tiempo que tarda en llegar una señal es directamente proporcional a su distancia recorrida a una velocidad de 300000 kilómetros por cada segundo.

Cuando nos encontramos en cualquier parte del planeta, el tiempo que se tarda en llegar la señal al receptor es de aproximadamente 67 milisegundos cuando el satélite se encuentra en el zenit, pero si se encuentra en el horizonte el tiempo es de 86 milisegundos.

El receptor GPS mide el tiempo que tarda en llegar la señal a través de la hora emitida y la hora de llegada, calculando de esta manera la distancia al satélite.

#### 2.2.5 Precisión

La precisión que proporciona cualquier dispositivo es un factor muy importante, pero cuando de este depende la seguridad de las personas, la



precisión se convierte en un factor vital. Es por esta razón que es de mucha importancia saber la fiabilidad proporcionada por el GPS durante su funcionamiento.

La precisión proporcionada por el dispositivo GPS puede ser de dos clases:

- Precisión horizontal.
- Precisión vertical.

### 2.2.6 Precisión horizontal

Los dispositivos GPS se desarrollaron para proporcionar una precisión horizontal inferior a cien metros en el 95% de las situaciones. En un 4.99% la precisión puede estar entre los 100 y 300 metros, mientras que en un 0.01% superior a los 300 metros.

Estos son valores máximos, lo que significa que el dispositivo GPS puede tener un error de más de 100 metros en un tiempo de 1 hora con 12 minutos y mayor a 300 metros durante 9 segundos. (Letham, 2009)

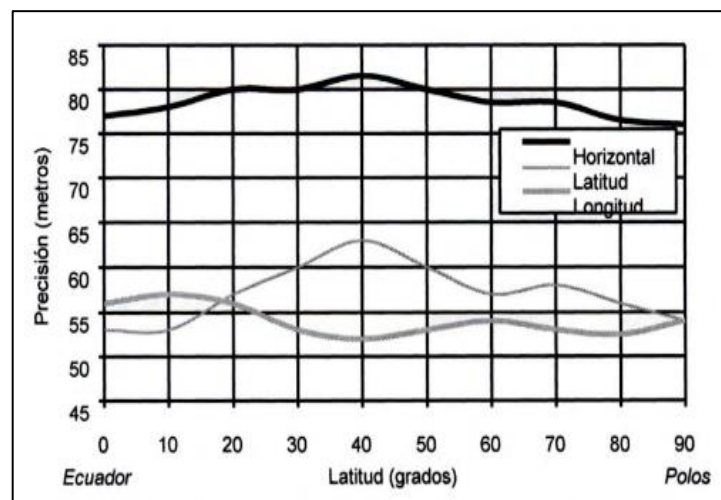


Figura 6. Precisión horizontal en función de la latitud.

Fuente: (Letham, 2009)

No obstante, a través de las pruebas realizadas se ha llegado a concluir que el GPS puede tener un error menor a 83 metros en el 95% de las situaciones, teniendo también un error máximo de 261 metros.

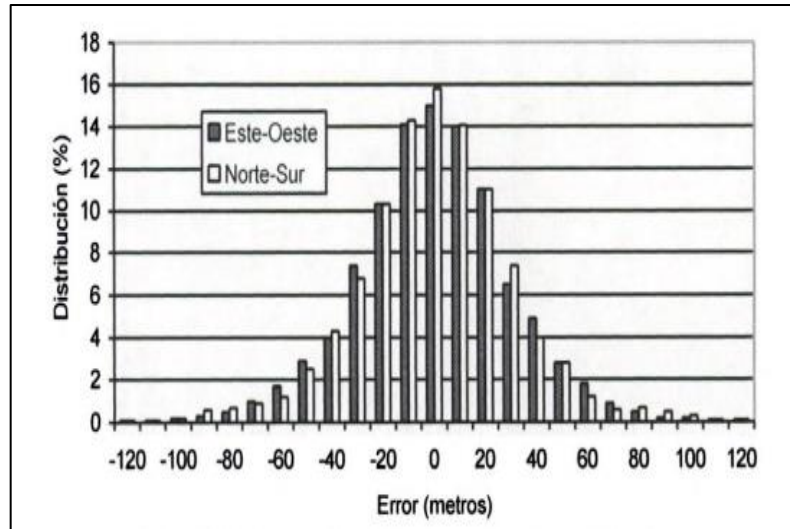


Figura 7. Distribución del error este-oeste y norte-sur.  
Fuente: (Letham, 2009)

En la siguiente figura se puede observar la probabilidad de error horizontal proporcionada por el GPS, en la que en el 50% de las observaciones realizadas existe un error inferior a 30 metros.

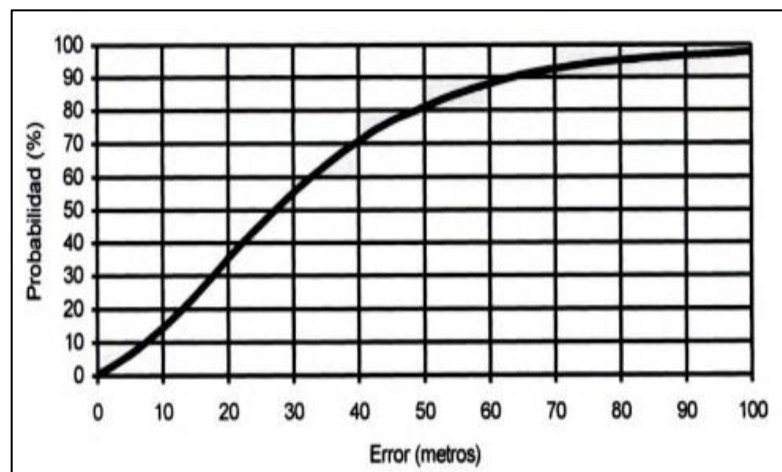


Figura 8. Probabilidad del error horizontal.  
Fuente: (Letham, 2009)

### 2.2.7 Precisión vertical

La precisión vertical tiene un error inferior o superior a 156 metros en el 95% de las situaciones e inferior o superior a 500 metros en el 99.99% de los casos. La precisión vertical es peor que la horizontal debido a que cuando más elevada es la posición del receptor mejor será la precisión vertical obtenida. (Letham, 2009)

No obstante, hay menos posibilidades de que el satélite este por encima de los 45° del horizonte que, por debajo, además no existe ninguna posibilidad que el receptor GPS se encuentre por encima del satélite. En la siguiente figura se observa el error vertical durante un tiempo de dos horas.

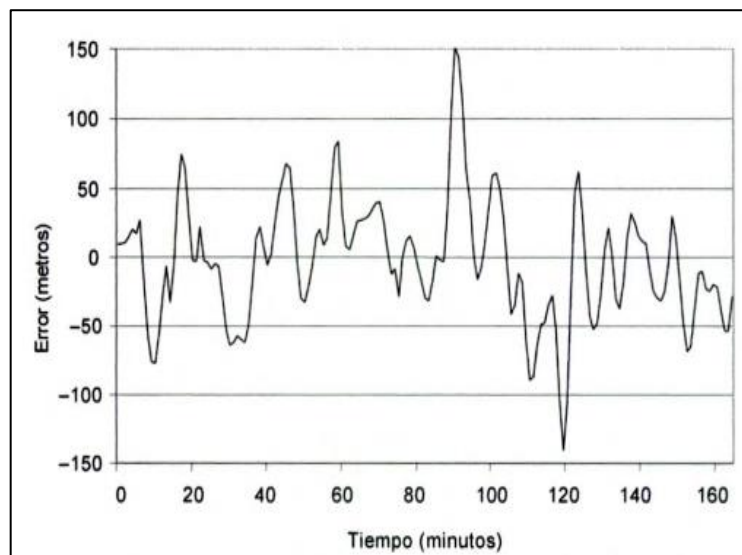


Figura 9. Observación del error vertical durante 2h30 minutos.

Fuente: (Letham, 2009)

La precisión vertical proporcionada por un GPS varía en función de la latitud, obteniéndose una mayor precisión a los 55° norte y sur, obteniéndose un error inferior a los 105 metros en el 95% de las ocasiones. La peor precisión se halla en los polos debido a que no existe ningún satélite que la sobrepase.

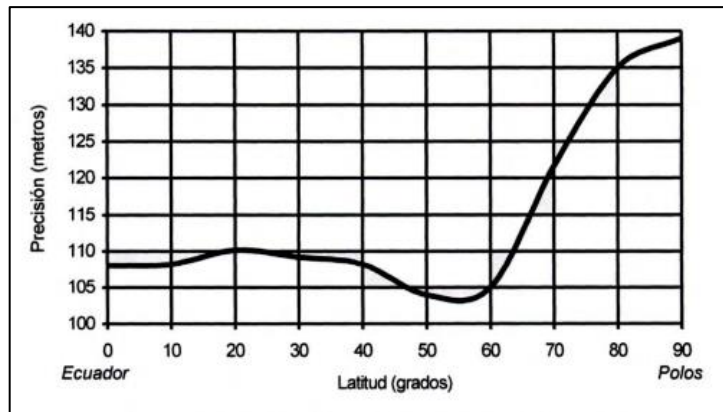


Figura 10. Precisión vertical según la latitud.  
Fuente: (Letham, 2009)

La precisión vertical es únicamente un valor teórico ya que puede variar de la manera en que el receptor seleccione los satélites para determinar una posición.

### 2.2.8 Fuentes de error

La precisión del GPS puede ser afectada por varias fuentes de error que algunas de ellas son de origen natural, otras de origen técnico o voluntario. En la siguiente tabla se observa un error específico provocado por alguno de estos elementos. ( Florez Lopez & Isaza Posse, 2015)

Tabla 1. Valores Corrientes de las fuentes de error.

| Fuentes de error                   | Error habitual (metros) |
|------------------------------------|-------------------------|
| SA                                 | 24                      |
| Ionósfera no modelizada            | 7                       |
| Tropósfera                         | 0.7                     |
| Reflexiones parásitas              | 1.2                     |
| Reloj y precisión de los satélites | 3.6                     |
| Ruido del receptor                 | 1.5                     |
| Error debido al satélite           | En torno a 25 metros    |

Fuente: ( Florez Lopez & Isaza Posse, 2015)

### **2.2.9 Receptor GPS**

Un receptor GPS tiene la función específica de señalar una posición, de todas maneras es necesario tener en cuenta que la presentación de datos corresponda a nuestras necesidades y que además, tenga todos los sistemas geodésicos de las cartas a utilizar. ( Florez Lopez & Isaza Posse, 2015)

Debido a que existe gran cantidad de modelos de receptores, hay que elegir el que mejor se adapte a nuestras necesidades en función de sus características técnicas, lo que permitirá aprovechar de mejor manera el receptor GPS. La precisión de la posición brindada por el receptor dependerá del tipo de filtrado a utilizar , como también de los algoritmos de posicionamiento logrando que no exista gran variación en la precisión.

### **2.2.10 Formato de presentación de datos**

La posición brindada por un receptor puede presentarse en algunos formatos, según el equipo y los países. A continuación se presentan los formatos de presentación de longitud y latitud:

- Horas y fracciones de hora: dddd.ddddd°
- Horas, minutos y fracciones de minuto: hddd°mm.mmm’
- Horas, minutos, segundos y fracciones de segundos: hddd°mm’ss.s”

Para la presentación en tierra es necesario el formato UTM, en cambio en las zonas polares es indispensable la presentación UPS.

## **2.3 Arduino.**

Arduino es una placa de hardware libre que posee un microcontrolador que puede ser reprogramable con pines hembra para fácil conexión de elementos electrónicos externos de sensores y actuadores. Al hablar de placa hardware se refiere a un circuito impreso con superficies fabricadas en material no conductor con pistas de material conductor (cobre). Cuando se habla de placa Arduino es necesario especificar su modelo debido a que existen muchas placas oficiales con características diferentes, como el tamaño, número de pines, memoria. De cualquier forma, aunque sean modelos diferentes, estas placas poseen

microcontroladores que pertenecen a la misma familia de tipo AVR, siendo su funcionamiento muy parecido entre sí. (Artero, 2013)

Arduino es una plataforma de computación física con sistemas digitales en los que es posible la conexión de sensores y actuadores, basadas en una placa con entradas y salidas digitales y analógicas. Este software y hardware es desarrollado a través de una biblioteca que simplifica la escritura de la programación C/C++. (Pedrera, 2014)

Es un software libre, multiplataforma y gratis que debe ser instalado en el ordenador para escribir, verificar y cargar el código configurado a la placa Arduino a utilizar a través de la utilización de un cable USB. Los proyectos realizados en Arduino pueden ser y no autónomos; en el caso de serlo, la placa no debe estar necesariamente conectada a un computador, pero en caso contrario debe estar permanentemente conectada al ordenador.

En su lenguaje de programación se utilizan diferentes comandos, variables para el almacenamiento de datos. Las placas Arduino se basan en los microcontroladores de la familia Atmega y cada uno de ellos tiene diferentes características como se indica en la siguiente tabla.

Tabla 2. Especificaciones de los microcontroladores.

|                   | <b>Atmega168</b> | <b>Atmega328</b> | <b>Atmega1280</b> |
|-------------------|------------------|------------------|-------------------|
| Voltaje operativo | 5voltios         | 5voltios         | 5voltios          |
| Voltaje entrada   | 7-12voltios      | 7-12voltios      | 7-12voltios       |
| Voltaje límite    | 6-20voltios      | 6-20voltios      | 6-20voltios       |
| Pines de entrada  | 14               | 14               | 54                |
| Pines análogos    | 6                | 6                | 16                |
| Corriente         | 40mA             | 40mA             | 40mA              |
| Memoria           | 16kb             | 32kb             | 128kb             |
| SRAM              | 1kb              | 2kb              | 8kb               |
| EEPROM            | 512bytes         | 1kb              | 4kb               |

Fuente: (Arduino, 2016)

### **2.3.1 Origen de arduino**

El software y placa arduino fue creada en el año 2005 a través del Instituto de Diseño Interactivo de Ivrea, que es un centro donde los estudiantes realizan experimentos de dispositivos electrónicos para la interacción con las personas. Esta placa surgió en la necesidad que tienen los estudiantes de contar con un elemento para ser utilizado en las aulas y de bajo costo, y que pueda funcionar con cualquier sistema operativo.

No obstante, este Instituto decidió liberar Arduino para que las personas de todo el planeta tenga la posibilidad de utilizarlo para el desarrollo de proyectos y de esta manera poder realizar mejoras y mantenerlo en funcionamiento. Massimo Banzi fue la persona que encabezaba al grupo Arduino Team y el responsable del diseño de la placa. (Artero, 2013)

### **2.3.2 Arduino software libre**

Se ha dicho que Arduino es un hardware y software libre, que según la organización encargada de promover el desarrollo y uso del software libre, consideran que este puede ser considerado como tal siempre y cuando cumpla con las siguientes características:

- Libertad 0. Libertad de utilizar el programa para cualquier propósito y en cualquier sistema.
- Libertad 1. Libertad de analizar el funcionamiento interno del programa para así adaptarlo a necesidades particulares.
- Libertad 2. Libertad de comercializar copias.
- Libertad 3. Libertad de realizar mejoras en el programa para hacerlas públicas a los demás usuarios de este sistema.

### **2.3.3 Arduino hardware libre**

Para que Arduino sea un hardware libre debe compartir los principios y metodologías de sus software. El hardware libre permite que las personas

puedan estudiarlo para modificarlo, mejorarlo, reutilizarlo y compartir los cambios realizados con los demás. Para esto, es necesario que las personas tengan acceso a los ficheros esquemáticos que detallan la información necesaria para poder realizar cualquier cambio por cuenta propia y sin problemas.

El objetivo principal del hardware libre, es facilitar y acercar a la electrónica, robótica y tecnología actual a las personas en general de manera activa y no pasiva. Por estas razones es necesario conocer el concepto de hardware y software libre, debido a que no solo es importante para la electrónica e informática, sino para la vida en general. (Artero, 2013)

#### **2.3.4 Ventajas de elegir Arduino**

Existen diferentes placas para el control de elementos electrónicos que ofrecen un funcionamiento similar al de Arduino, pero a diferencia, esta plataforma tiene las siguientes ventajas:

- Libre y extensible. Cualquier persona puede mejorar el diseño del hardware y el entorno del software, como también el lenguaje de programación.
- Posee una gran comunidad. Utilizado por muchas personas que comparten sus ideas continuamente.
- Multiplataforma. Puede ser instalado en cualquier sistema operativo.

#### **2.3.5 Entorno de programación**

Su entorno de desarrollo esta constituido por un editor de texto en el que se escribe el código de programación y una barra de herramientas con botones que tienen funciones específicas, además de una conexión USB para la comunicación con el ordenador.



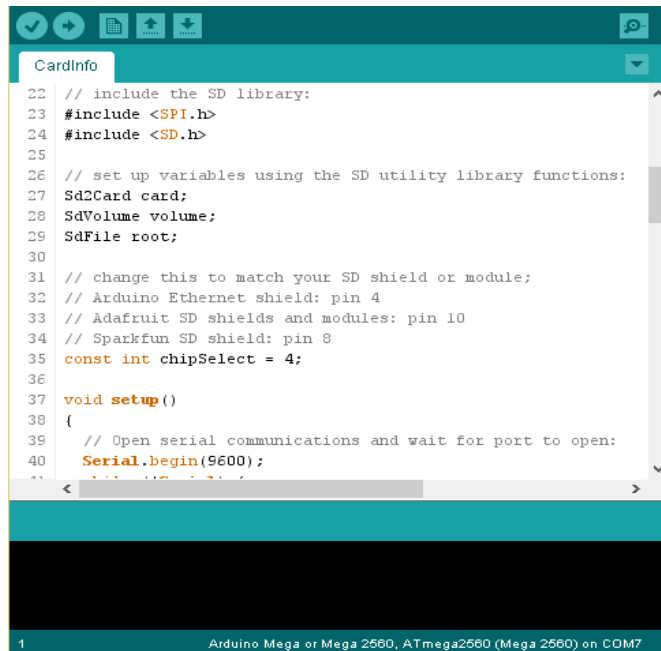


Figura 11. Entorno de programación.

Para escribir un código, Arduino utiliza una aplicación denominada sketch, que son escritos en el editor de texto. A través de este hay la posibilidad de cortar y pegar, buscar o reemplazar las instrucciones a cargar en la placa. La barra de herramientas permite verificar el código escrito, guardado y apertura, como también el monitor serial, tal como se muestra en la siguiente tabla.

Tabla 3. Opciones de la barra de herramientas.

| Herramienta    | Descripción                             |
|----------------|---|
| Verify/Compile | Busca errores en el código              |
| Stop           | Finaliza el monitor serie               |
| New            | Crea un nuevo sketch                    |
| Open           | Presenta un menú de todos los programas |
| Save           | Guarda el programa                      |
| Serial monitor | Inicia el monitor serial                |

Fuente: (Artero, 2013)

## **2.4 LabVIEW.**

El programa LabVIEW es una herramienta que permite resolver los problemas actuales de manera más rápida y eficiente, con la finalidad de resolver y evolucionar futuros objetivos. Además, ofrece integración con la gran variedad de hardware de medidas existentes, así como también con el software de las últimas tecnologías.

La palabra LabVIEW es un acrónimo de Laboratory Virtual Instrumentation Engineering Workbench, que describe a una plataforma y entorno de desarrollo que se utiliza para diseñar sistemas a través de la aplicación de un lenguaje de programación gráfica. Este software es muy recomendado para sistemas con hardware para el control y diseño, ya sea simulado o real y embebido, debido a que acelera la productividad. Utiliza un lenguaje de programación conocido con el nombre de lenguaje G, donde simboliza que es un lenguaje de tipo gráfico visual. (Lajara Vizcaíno & Pelegrí Sebastián, 2012)

LabVIEW fue ideado por la organización National Instruments en el año de 1976, pero fue operable únicamente para máquinas MAC. Este programa salió al mercado en el año de 1986 y en la actualidad es plenamente operacional para las distintas plataformas existentes.

Los programas que se desarrollan en LabVIEW tienen el nombre de Vis o Instrumentos Virtuales, en el que su origen proviene del control de instrumentos electrónicos, aunque no solo se utiliza para estos objetivos sino también para la programación embebida, matemáticas, comunicaciones, entre otros. (Lajara Vizcaíno & Pelegrí Sebastián, 2012)

### **2.4.1 Principales características**

La principal característica de este programa es su facilidad de uso para programadores profesionales como también para personas con conocimientos mínimos en programación, debido a que tienen la habilidad de realizar

programas complejos que son difíciles de hacer con otros lenguajes tradicionales. Este programa dispone de facilidades para la utilización de:

- Interfaz de comunicación.
- Interacción de lenguajes y aplicaciones.
- Aplicación en Matlab.
- Utilización en AutoCAD, SolidWorks.
- Herramienta gráfica para el proceso de señales digitales.
- Manejo de datos dinámicos.
- Adquisición de imágenes.
- Control de movimientos.
- Adquisición de datos en tiempo real.
- Sincronización con diferentes dispositivos externos.

#### **2.4.2 Programación.**

Es una herramienta gráfica de programación, lo que significa que en la creación de programas el código se dibuje y no se escriba, facilitando de esta manera su comprensión. Debido a que ya se dispone de gran cantidad de bloques pre diseñados, el usuario tiene la facilidad de crear proyectos de mayor dificultad, invirtiendo menos tiempo en la programación para dedicarse con mayor intensidad al diseño de la interfaz gráfica y la interacción con el usuario.

Cada instrumento virtual consta de dos partes que son:

- a) *Panel frontal.*** Es la parte de la interfaz con el usuario para la visualización de datos en tiempo real cuando el programa se está ejecutando. Es el lugar en el que se utilizan controles para la entrada y salida de datos, así como también indicadores para su presentación.



Figura 12. Panel frontal.

**b) Diagrama de bloques.** Es el lugar en el que se realiza la programación y se define su funcionamiento.

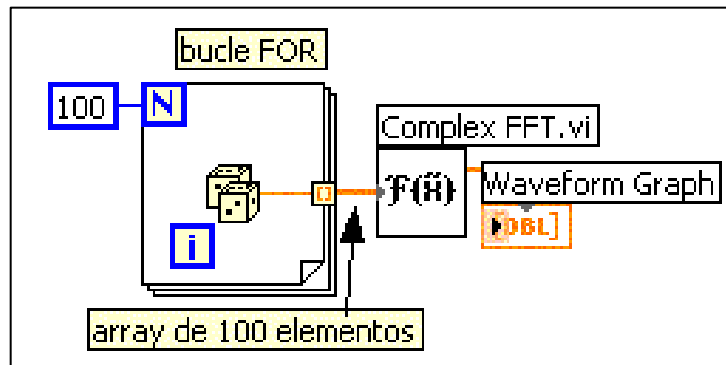


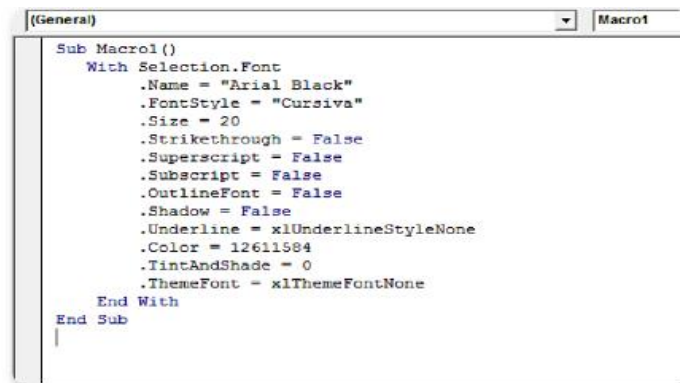
Figura 13. Diagrama de bloques.

En la parte del panel frontal todos los controles e indicadores tienen asignado un terminal en el diagrama de bloques, es decir, con estos elementos el usuario definirá las entradas y salidas que interactúen con los terminales del instrumento virtual.

### 2.4.3 Macros de Excel.

Una macro de Excel son secuencias de instrucciones que están escritas en lenguaje VBA que son almacenadas en un módulo. Cuando se llama o activa a una macro, ya sea a través de un botón o combinación de teclas, se desencadenan las instrucciones guardadas en ella. Desde el punto de vista general, las macros ayudan a automatizar tareas, es decir, son partes de códigos de programación interpretados por Excel que desarrollan distintas

tareas o sucesión de acciones. De esta manera con la utilización de una macro se puede realizar acciones repetitivas fácilmente. (Zanini, 2013)



```
Sub Macrol1()  
  With Selection.Font  
    .Name = "Arial Black"  
    .FontStyle = "Cursiva"  
    .Size = 20  
    .Strikethrough = False  
    .Superscript = False  
    .Subscript = False  
    .OutlineFont = False  
    .Shadow = False  
    .Underline = xlUnderlineStyleNone  
    .Color = 12611584  
    .TintAndShade = 0  
    .ThemeFont = xlThemeFontNone  
  End With  
End Sub
```

Figura 14. Conjunto de instrucciones escritas en VBA.

Fuente: (Zanini, 2013)

Sin embargo, las macros no pueden ser utilizadas únicamente para este uso, sino también para desarrollar aplicaciones complejas con la utilización del lenguaje de programación en el que están interpretadas las macros. El punto fuerte que tienen las macros es su facilidad de creación y ejecución, y se puede disminuir mucho tiempo al reemplazar las tareas repetitivas por las macros.

#### 2.4.3.1 Lenguaje Visual Basic.

Excel es una potente aplicación que viene incluida en el paquete de Microsoft Office y que proporciona funciones y herramientas distintas para analizar, administrar y compartir datos. Esta aplicación brinda muchas posibilidades para automatizar tareas que se realiza de manera cotidiana, así como también desarrollar aplicaciones que están basadas en el análisis de datos numéricos a través de la creación de macros.

Visual Basic es un lenguaje de programación que tiene mucho en común con las aplicaciones incluidas en el paquete de Microsoft Office, así como también con el de otros programas como son el Corel Draw o Autocad. En la imagen de la figura 15 se puede observar el módulo de VBA con las respectivas herramientas de programación.

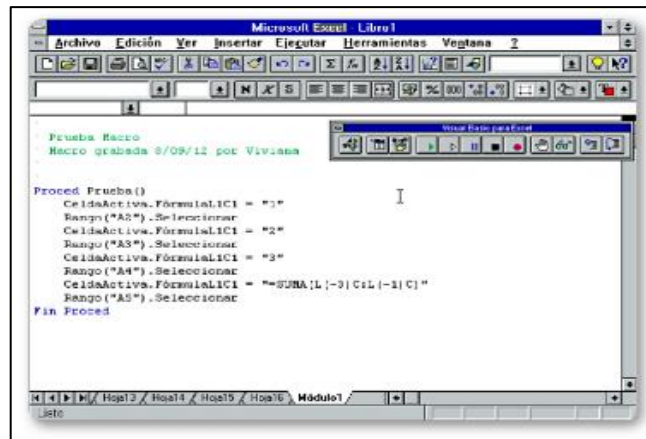


Figura 15. Módulo de VBA.

Fuente: (Zanini, 2013)

### 2.4.3.2 VBA y Visual Basic (VB).

Visual Basic para aplicaciones es una versión de Visual Basic que está embebido en las demás aplicaciones de Microsoft Office que permite escribir un grupo de instrucciones para programar varios objetos en Excel, como una hoja de cálculo, gráficos o celdas. La diferencia que existe entre Visual Basic y Visual Basic para aplicaciones está en que el primero permite desarrollar ejecutables que pueden instalarse en cualquier ordenador que trabaje bajo el entorno de Windows. (Zanini, 2013)

En cambio, VBA únicamente permite escribir instrucciones que se ejecutan en el entorno de Excel. Tiene un lenguaje de aprendizaje fácil y además el código en VB y VBA son muy parecidos por lo que pueden ser utilizados para comprender el lenguaje utilizado en macros.

## **CAPÍTULO III.**

### **3. METODOLOGÍA DE LA INVESTIGACIÓN.**

#### **3.1 Tipo de investigación.**

El presente trabajo de grado es de investigación bibliográfica y tecnológica, las cuales se describen a continuación:

##### **3.1.1 Investigación Bibliográfica.**

Es de investigación bibliográfica debido a que se realiza una extensa búsqueda de información de forma sistemática sobre el tema a tratar en el presente trabajo de grado, a través de la revisión de libros, manuales que sirvan de guía para para el diseño y construcción del dispositivo.

##### **3.1.2 Investigación Tecnológica.**

La investigación tecnológica está sumergida en el presente trabajo de grado porque con el diseño del dispositivo para el control de tiempo de viajes de un autobús se vincula de forma natural con la innovación tecnológica.

#### **3.2 Métodos.**

La metodología aplicada es la siguiente:

- **Diseño electrónico.** Elaboración de diagramas electrónicos para conocer la forma de conexión de los diferentes elementos utilizados para la elaboración del dispositivo.

- **Optimización.** Mejoras en el funcionamiento del dispositivo para lograr con éxito los objetivos planteados para el trabajo de grado.
- **Recolección de datos.** Realizar viajes a manera de simulación para el almacenamiento de datos y su respectivo análisis en el programa LabVIEW.
- **Adaptación.** Adaptación del GPS, pantalla y módulo Arduino para lograr un trabajo en conjunto y de esta manera mostrar la información respectiva y realizar un correcto almacenamiento de datos.

### **3.3 Técnicas e instrumentos.**

Las técnicas e instrumentos aplicados son:

- **Análisis de datos.** Realizar el análisis de datos a través de la información almacenada en la tarjeta de memoria del dispositivo durante los viajes realizados por el autobús, para de esta manera poder realizar mejoras en su servicio.
- **Pruebas de funcionamiento.** Realizar pruebas de funcionamiento a través de las diferentes rutas asignadas en la programación del dispositivo para así verificar su utilidad y forma de funcionamiento.



## CAPÍTULO IV.

### 4. PROPUESTA: PROCESO Y RESULTADOS.

#### 4.1 Etapas de diseño.

Para el diseño del dispositivo se utilizan seis etapas necesarias para conseguir los objetivos planteados que son: diseño del dispositivo, selección de elementos electrónicos, diagramas, conexiones, programación de Arduino, desarrollo del software y pruebas de funcionamiento; cada una de ellas contienen actividades específicas como se indica en el diagrama de la figura 16.

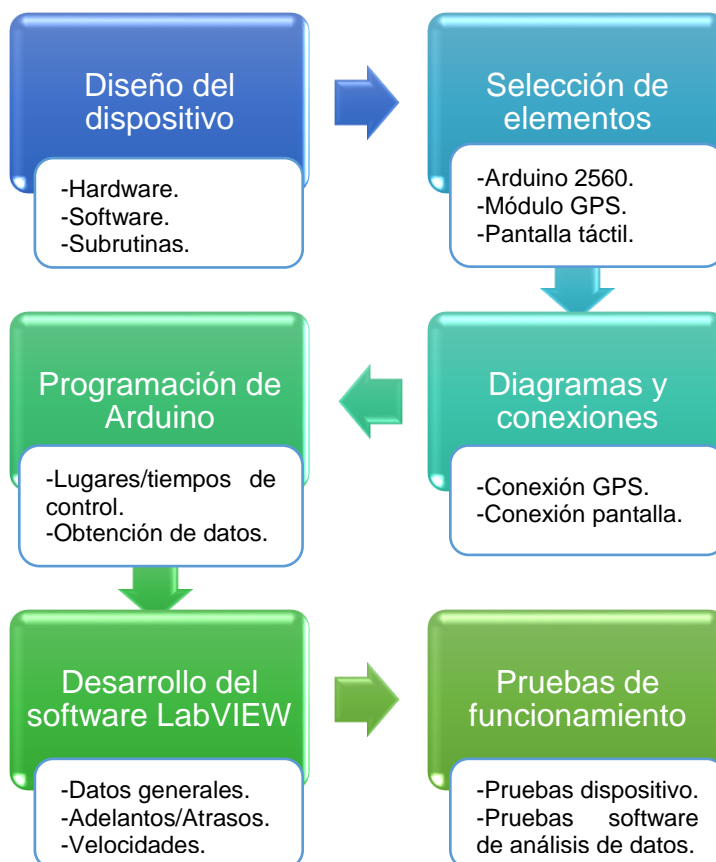


Figura 16. Etapas de diseño.

## 4.2 Selección de elementos electrónicos.

Para el diseño del dispositivo se utilizan elementos electrónicos como: módulo Arduino, módulo GPS, pantalla. Cada uno de los elementos desempeña una función específica para el buen funcionamiento del dispositivo y el almacenamiento de datos para su posterior análisis. Los elementos electrónicos a utilizar para el diseño del dispositivo son los que se detallan a continuación.

### 4.2.1 Placa Arduino.

La placa Arduino utilizada es la Mega 2560 que posee un microcontrolador Atmega 2560. Esta cuenta con 54 pines digitales y 16 analógicos para la conexión de sensores, además, posee una conexión USB y un botón de reinicio. Este elemento electrónico brinda tres puertos para la comunicación serial, lo que permite conectar hasta tres elementos externos al dispositivo a desarrollarse.

La comunicación entre el computador y Arduino se produce a través del puerto serie, sin embargo posee un convertidor USB-serie, por lo que sólo se necesita conectar el dispositivo a la computadora utilizando un cable USB como el que se utiliza en las impresoras.

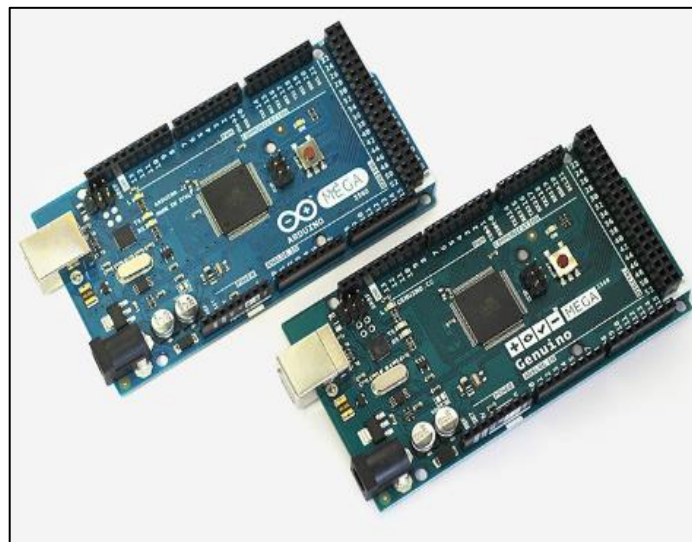


Figura 17. Arduino Mega 2560.

Fuente: (Arduino, 2016)

#### 4.2.1.1 Especificaciones.

Las especificaciones de la placa Arduino mega 2560 son las que se indican en la tabla 4.

Tabla 4. Especificaciones Arduino Mega.

| <b>Microcontrolador</b>       | <b>Atmega 2560</b>                                  |
|-------------------------------|---|
| Tensión para funcionamiento   | 5V  |
| Voltaje en la entrada         | 7V-12V  |
| Voltaje límite de entrada     | 6V-20V  |
| Entradas y salidas digitales  | 54 pines  |
| Pines analógicos              | 16 pines  |
| Corriente en cada pin         | 20 mA   |
| Corriente para Pin 3.3Voltios | 50 mA   |
| Memoria                       | 256 kB, 8 kB es utilizado por el gestor de arranque |
| SRAM                          | 8 kB de memoria                                     |
| EEPROM                        | 4 kB de memoria                                     |
| Velocidad de reloj            | 16 MHz  |
| Longitud                      | 101.52 mm   |
| Anchura                       | 53,3 mm   |
| Peso                          | 37 g  |

Fuente: (Arduino, 2016)

#### 4.2.1.2 Programación.

Para la programación de la placa Arduino Mega se utiliza el software Arduino IDE. Esta placa viene pre programada a través de un cargador de arranque que permite cargar un código nuevo a ella sin la necesidad de utilizar un programador de hardware externo. Además, esta placa se comunica a través de la utilización de un protocolo conocido con el nombre de STK500.

#### 4.2.1.3 Alimentación

El Arduino 2560 puede ser alimentado mediante la conexión del cable USB o a través de una fuente de alimentación externa. La alimentación externa puede conectarse en el enchufe de 2.1 milímetros de ancho que viene instalada en la placa o mediante los pines GND y Vin a través de una batería.

La placa puede funcionar correctamente a través de una alimentación de 6 voltios a 20 voltios, pero si se realiza con menos de 7 voltios, esta puede volverse inestable y si se utiliza más de 12 voltios su regulador de voltaje puede sobrecalentarse dañando de esta manera la placa. El valor recomendado de alimentación es de 7 voltios a 12 voltios.

Los pines utilizados para la alimentación son los siguientes:

- *Vin*. Se utiliza este pin cuando se alimenta la placa a través de una fuente externa.
- *5V*. Este pin suministra una alimentación de 5 voltios que pueden ser utilizados por los dispositivos conectados a la placa Arduino.
- *3V3*. A través de este se suministra una alimentación de 3.3 voltios con un consumo máximo de corriente de 50 miliamperios.
- *GND*. Pines de tierra.
- *IOREF*. Proporciona un valor de voltaje con la que trabaja el microcontrolador.

#### **4.2.1.4 Memoria**

La placa Arduino Mega 2560 contiene 256 KB de memoria para el almacenamiento de códigos, de los cuales 8kb son utilizados por el gestor de arranque, mientras que 8kb más se utiliza por la memoria SRAM y 4kb por la memoria EEPROM.

#### **4.2.1.5 Comunicación**

La placa contiene un conjunto de instalaciones para la comunicación con un computador o con otros microcontroladores. Esta dispone de 4 puertos UART para la comunicación serial TTL de 5 voltios. El software de Arduino IDE dispone además de un monitor serial en el que se puede observar datos de texto enviados hacia y desde la placa.

#### 4.2.1.6 Características físicas.

La longitud de la placa es de 4 pulgadas mientras que el ancho de 2.1 pulgadas, pero con el conector USB y el Jack de conexión estas dimensiones se extienden mucho más. La distancia que existe entre cada uno de los pines digitales es de 7 a 8 milésimas de pulgadas.

La Mega 2560 ha sido diseñada para ser compatible con la mayoría de los shield disponibles para el Arduino Uno y las diferentes placas anteriores.

#### 4.2.2 Módulo GPS.

El módulo GPS NEO-6M pertenece a una familia de receptores con alto rendimiento en su funcionamiento y fijación de posiciones. Estos dispositivos son muy flexibles y rentables debido a que ofrecen numerosas opciones de conectividad, a través de su arquitectura compacta, energía y memoria hacen que su funcionamiento sea el ideal para elementos móviles con estrictas limitaciones.

Este módulo cuenta con un TTFF de hasta cincuenta canales que en menos de 1 segundo es capaz de realizar la búsqueda del espacio de tiempo/frecuencia para encontrar los satélites al instante. Su diseño y tecnología innovadora permite que trabaje hasta con veinte satélites a la vez, proporcionando una exactitud de hasta cinco metros.

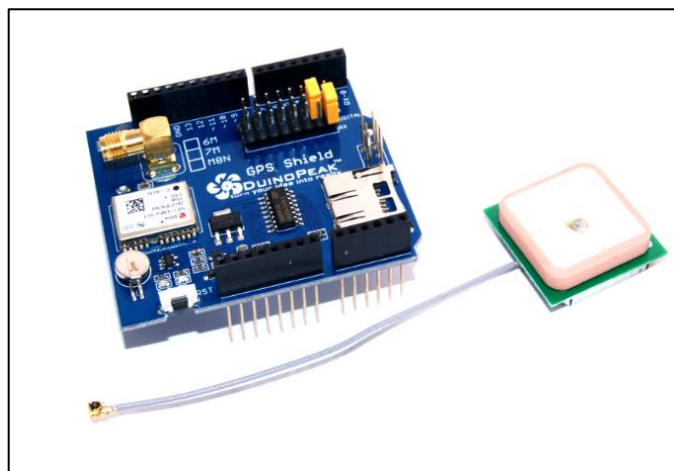


Figura 18. Módulo GPS.  
Fuente: (Arduino, 2016)

#### 4.2.2.1 Diagrama de bloques GPS.

El diagrama de bloques del módulo GPS NEO-6M-001 es como el que se indica en la figura 19.

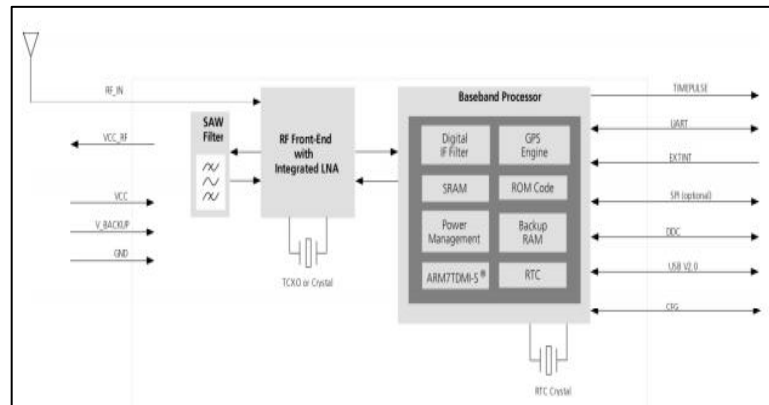


Figura 19. Diagrama de bloques módulo GPS.

Fuente: (Arduino, 2016)

#### 4.2.2.2 Tiempo de precisión.

El módulo GPS proporciona una hora especial para brindar una mayor precisión de temporización, debido a que está diseñado para su utilización con configuraciones de antenas fijas. Proporciona niveles de precisión muy altos en posiciones estáticas y en movimiento, lo que hace que sea ideal para aplicaciones tales como topografía, cartografía, entre otras.

#### 4.2.2.3 Protocolos disponibles.

Los protocolos para los cuales está disponible el módulo GPS son los que se indican a continuación:

- *UART*. Disponen de una interfaz UART configurable para la comunicación serial con otros dispositivos electrónicos.
- *USB*. Proporciona una versión USB 2.0 certificado para los diferentes sistemas operativos.
- *Interfaz SPI*. Permite la conexión con dispositivos externos para ahorrar pines de conexión, además puede ser operado como elemento maestro o esclavo.

#### 4.2.2.4 Asignación de pines.

Los pines que contiene el módulo GPS utilizado están asignados tal como se indican en la tabla 5.

Tabla 5. Pines GPS.

| N° | Nombre       | I/O | Descripción           |
|----|--------------|-----|-----------------------|
| 1  | Reservado    | I   | Reservado             |
| 2  | SS_N         | I   | Chip select           |
| 3  | Timepulse    | O   | Timepulse             |
| 4  | Interrupción | I   | Interrupción externa  |
| 5  | USB_DM       | I/O | Datos USB             |
| 6  | USB_DP       | I/O | Datos USB             |
| 7  | VDDUSB       | I   | Suministro USB        |
| 8  | Reservado    | I   | Reservado             |
| 9  | VCC_RF       | O   | Salida de voltaje     |
| 10 | GND          | I   | Tierra                |
| 11 | RF_IN        | I   | Señal GPS             |
| 12 | GND          | I   | Tierra                |
| 13 | GND          | I   | Tierra                |
| 14 | MOSI         | I   | SPI MOSI              |
| 15 | MISO         | I   | SPI MISO              |
| 16 | SCK          | I   | SPI reloj             |
| 17 | Reservado    | I   | Reservado             |
| 18 | SDA2         | I/O | Datos DDC             |
| 19 | SCL2         | I/O | Reloj DDC             |
| 20 | TXD1         | O   | Puerto serie 1        |
| 21 | RXD1         | I   | Puerto serie 1        |
| 22 | V_BCKP       | I   | Suministro de tensión |
| 23 | VCC          | I   | Alimentación          |
| 24 | GND          | I   | Tierra                |

Fuente: (Arduino, 2016)

La distribución de cada pin en la placa del módulo se puede observar en la figura 20, que indica la cantidad de pines a cada lado del módulo necesarios para la comunicación con la placa Arduino Mega.

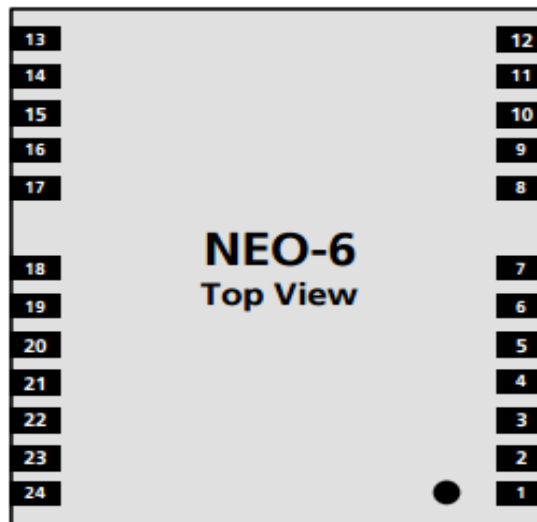


Figura 20. Distribución de pines GPS.  
Fuente: (Arduino, 2016)

#### 4.2.2.5 Especificaciones mecánicas

Las especificaciones mecánicas se indican en la figura 21.

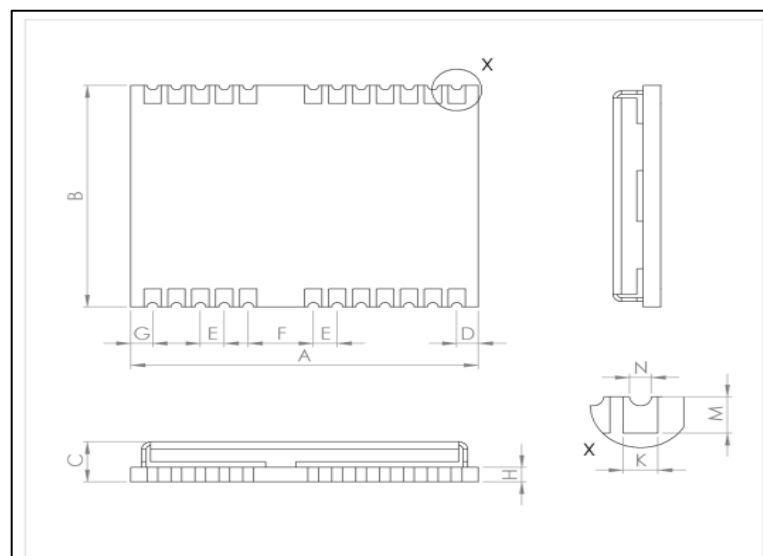


Figura 21. Especificaciones mecánicas GPS.  
Fuente: (Arduino, 2016)



La simbología utilizada en el gráfico de la figura 19 se indica en la siguiente tabla, con las medidas de cada una de las partes del módulo y el diseño con el cual está desarrollado.

Tabla 6. Simbología especificaciones mecánicas.

| <b>Símbolo</b> | <b>Mínimo(mm)</b> | <b>Normal(mm)</b> | <b>Máximo(mm)</b> |
|----------------|-------------------|-------------------|-------------------|
| A              | 15.9              | 16.0              | 16.6              |
| B              | 12.1              | 12.2              | 12.3              |
| C              | 2.2               | 2.4               | 2.6               |
| D              | 0.9               | 1.0               | 1.3               |
| E              | 1.0               | 1.1               | 1.2               |
| F              | 2.9               | 3.0               | 3.1               |
| G              | 0.9               | 1.0               | 1.3               |
| H              |                   | 0.82              |                   |
| K              | 0.7               | 0.8               | 0.9               |
| M              | 0.8               | 0.9               | 1.0               |
| N              | 0.4               | 0.5               | 0.6               |
| Peso           |                   | 1.6g              |                   |

Fuente: (Banzi, 2012)

#### 4.2.3 Pantalla.

La pantalla utilizada es una LCD TFT de 3.2 pulgadas con resolución de 320\*240 y colores de 65K, además, posee 34 pines de interfaz no solo para la utilización de la LCD, sino también para el control táctil y la tarjeta SD que viene integrada en el módulo de la pantalla.

Está conformada por un controlador SSD1289 con 16 bits para la interfaz de datos y un controlador táctil HY-TFT 320. Utiliza la comunicación SPI para trabajar con la tarjeta SD y sus pines incluyen en el módulo.



Figura 22. Pantalla TFT 3.2”.

Fuente: (Arduino, 2016)

#### 4.2.3.1 Especificaciones.

Las especificaciones técnicas de la pantalla se indican a continuación:

- TFT LCD a color de 3.2”.
- Resolución de 320\*240.
- 65536 colores de 16 bits.
- Microcontrolador potente SSD1289.
- Fabricado en memoria RAM de video.
- Pantalla táctil resistiva.
- Ranura para tarjeta SD integrada.
- Alimentación de 5 voltios.
- Luz LED de fondo.



Tabla 7. Pines de conexión Shield TFT 3.2”.

| Pin Shield  | Pin Arduino                 | Descripción              |
|---|-----------------------------|--------------------------|
| + 3V3   | 3.3V                        | Alimentación             |
| GND   | GND                         | Tierra                   |
| LED   | 5V                          | Alimentación LCD         |
| D80-D81   | D37-D30, D22-D29            | Interfaz paralela        |
| LCD_RS, LCD_WR,<br>LCD_RD, LCD_CS,<br>LCD_RST         | D38, D39, 3.3V,<br>D40, D41 | Control de datos<br>LCD. |
| TP_CLK, TP_CS,<br>TP_DIN, TP_BUSY,<br>TP_DOUT, TP_IRQ | D6, D5, D4, D7, D3,<br>D2   | Control táctil           |
| SD_CS, MISO,<br>CLK, MOSI                             | D53, D50, D52, D51          | Control tarjeta SD       |

Fuente: (Lequerica, 2013)

La distribución de pines se indica en la figura 24.

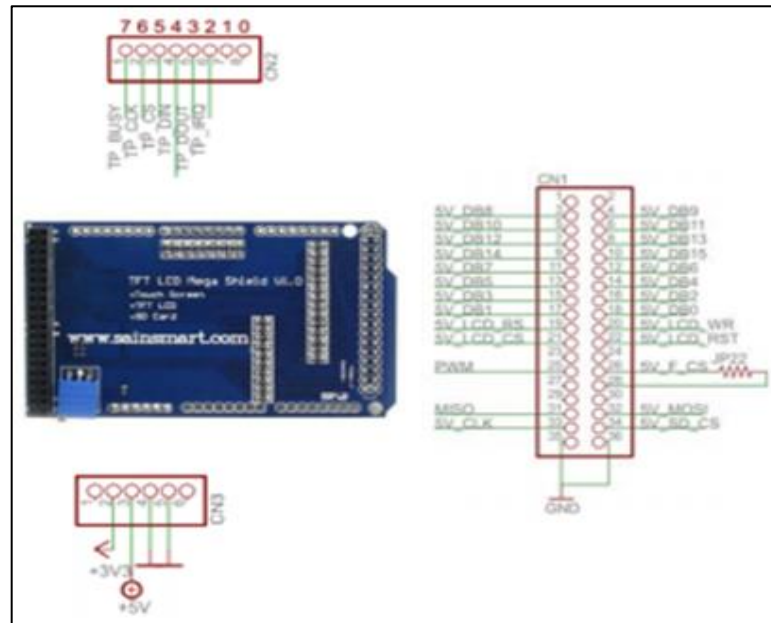


Figura 24. Distribución de pines Shield TFT 3.2”.

Fuente: (Arduino, 2016)

## 4.3 Implementación de hardware y software de Arduino.

### 4.3.1 Diagrama de conexión del módulo GPS.

La conexión del módulo GPS en la placa Arduino Mega se realiza a través de un shield como se indica en la figura 25. Este módulo es conectado a la comunicación serial1 (RX1, TX1) de Arduino para la obtención de los datos proporcionados por el módulo GPS. La conexión de pines se realiza como se detalla a continuación:

- Rx GPS → Tx1 Arduino.
- Tx GPS → Rx1 Arduino.
- GND GPS → GND Arduino.
- 5V GPS → 5V Arduino.

El módulo GPS trabaja a 9600 baudios y para ello es necesario instalar la librería <TinyGPS++.h> que permite obtener de forma rápida los valores de hora, fecha, longitud, latitud y velocidad necesarias para el correcto funcionamiento del dispositivo.

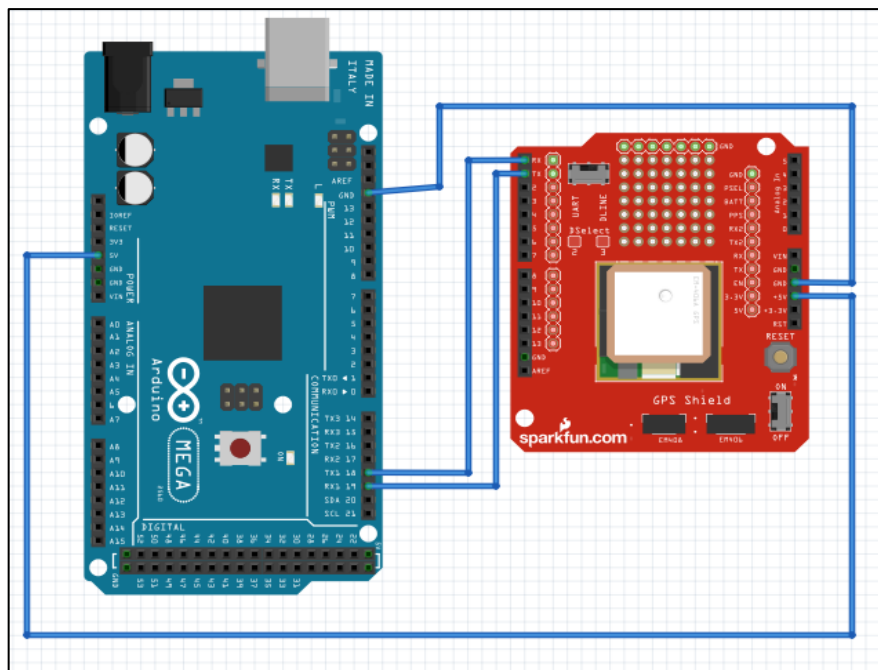


Figura 25. Diagrama de conexión módulo GPS.

El funcionamiento en conjunto del módulo GPS y de la placa Arduino permite establecer las áreas para los lugares de control del tiempo real a la que pasa el autobús. Esto se logra gracias a la biblioteca instalada y a las respectivas instrucciones del código programado de la figura 26, que con los valores de longitud y latitud permiten limitar las áreas para el control de tiempo.

```

codigo2 $
512 //TIEMPO REAL 1
513 if (3615>=LAT && LAT>=3602) //IBARRA TERMINAL
514 {
515     if (781124>=LONG && LONG>=781114)
516     {
517         posicion1++;
518         if (posicion1==1)
519         {
520             horall=h;
521             minutoll=Bus.time.minute();
522             tiempo2=tiempo;
523             diferencial=(3-tiempo2)/60;
524             myGLCD.setFont(BigFont);
525             myGLCD.setBackgroundColor(VGA_LIME);
526             myGLCD.setColor(VGA_BLUE);
527             myGLCD.printNumI(diferencial, 235, 145);
528             if (diferencial<0)
529             {
530                 horal=horall;
531                 minutol=minutoll-abs(diferencial);
532             }
533             else if (diferencial>=0)
534             {
535                 horal=horall;
536                 minutol=minutoll+diferencial;
537             }
538         }
539     }

```

Figura 26. Código limitación de áreas de control.

La velocidad, hora y fecha actual que se indica en la pantalla del dispositivo durante su funcionamiento son adquiridas a través de los comandos de la librería <TinyGPS++.h>. Para obtener la hora en Ecuador es necesario restar cinco horas a la proporcionada por el GPS debido a que este posee la hora del meridiano cero, tal como se indica en el código de la figura 27.

```

codigo2 $
452 int h= Bus.time.hour();
453 h= h-5; //restamos cinco horas para la obtener la hora del ecuador
454 if (h<0)
455 {
456     h= h+24;
457 }
458 long LONG=abs(Bus.location.lng())*10000;
459 long LAT=abs(Bus.location.lat())*10000;
460 tiempo++; //para tiempo de viaje
461 Serial.print(tiempo);
462 myGLCD.setFont(SevenSegNumFont);
463 myGLCD.setBackgroundColor(VGA_BLUE);
464 myGLCD.setColor(VGA_WHITE);
465 myGLCD.printNumI(Bus.speed.kmph(), 140, 30);
466 int velocidad=Bus.speed.kmph();
467 if (Bus.speed.kmph()<2)
468 {
469     velocidad=0;
470 }
471 myGLCD.printNumI(velocidad, 140, 30);
472 if (velocidad<10)
473 {
474     myGLCD.setColor(VGA_BLUE);
475     myGLCD.fillRect(170, 30, 230, 80);
476 }

```

Figura 27. Código de velocidad, hora y fecha.

La velocidad máxima y mínima registradas por el módulo GPS durante el funcionamiento del dispositivo son almacenadas en archivos diferentes de la tarjeta de memoria. La velocidad máxima se guarda cuando el autobús sobrepasa el límite de velocidad de 90km/h establecidos por la ANT durante un tiempo de 10 segundos estimados para que el conductor disminuya la velocidad; estos datos se almacenan en el archivo VMAX.CSV a través del código que se indica en la figura 28.

```

codigo2 $
1320 //VELOCIDAD MAXIMA
1321 if (velocidad>=100)
1322 {
1323     .
1324     maxima++;
1325     if(maxima==10) //10 segundos
1326     {
1327         Serial.println("velocidad maxima");
1328         int vmax=velocidad;
1329         datos=SD.open("VMAX.csv", FILE_WRITE);
1330         if (datos)
1331         {
1332             datos.println();
1333             datos.print(titulo);
1334             datos.print(",");
1335             datos.print(",");
1336             datos.print(dia);
1337             datos.print("/");
1338             datos.print(mes);
1339             datos.print("/");
1340             datos.print(a_o);
1341             datos.print(",");
1342             datos.print(",");
1343             datos.print(Bus.time.hour()-5); // RESTAR 5 HORAS XQ DA LA HORA DEL MERIDIANO CERO
1344             datos.print(",");
1345             datos.print(Bus.time.minute());
1346             datos.print(",");
1347             datos.print(Bus.location.lat(), 6);

```

Figura 28. Código velocidad máxima.

La velocidad mínima se almacena cuando el autobús se mantiene a una velocidad menor a 30km/h durante un tiempo de 50 segundos en el archivo VMIN.CSV, con la finalidad de conocer los lugares en los que hay pérdidas de tiempo de viaje. El código utilizado para el almacenamiento de esta información se presenta en la figura 29.

```

codigo2 $
1362 //VELOCIDAD MINIMA
1363 if (velocidad<30)
1364 {
1365     minima++;
1366     if(minima==50) //50 segundos
1367     {
1368         Serial.println("velocidad minima");
1369         int vmin=velocidad;
1370         datos=SD.open("VMIN.csv", FILE_WRITE);
1371         if (datos)
1372         {
1373             datos.println();
1374             datos.print(titulo);
1375             datos.print(",");
1376             datos.print(",");
1377             datos.print(dia);
1378             datos.print("/");
1379             datos.print(mes);
1380             datos.print("/");
1381             datos.print(a_o);
1382             datos.print(",");
1383             datos.print(",");
1384             datos.print(Bus.time.hour()-5);
1385             datos.print(",");
1386             datos.print(Bus.time.minute());
1387             datos.print(",");
1388             datos.print(Bus.location.lat(), 6);
1389             datos.print(",");

```

Figura 29. Código velocidad mínima.

### 4.3.2 Diagrama de conexión de la pantalla.

En la pantalla táctil de 3.2" se indican datos en tiempo real según el funcionamiento del dispositivo, además, dispone de una ranura de tarjeta SD para el respectivo almacenamiento de datos. Para su correcto funcionamiento se realiza la conexión de pines para el control de la pantalla, control táctil, tarjeta SD, tal como se indica en la figura 30; para el control de datos de la pantalla se realiza la conexión de los siguientes pines:

- LCD\_RS → D38
- LCD\_WR → D39
- LCD\_RD → 3.3V
- LCD\_CS → D40
- LCD\_RST → D41

La TFT tiene incorporada una pantalla táctil de cuatro hilos y la interfaz necesaria para su funcionamiento es como se indica a continuación:

- Pin 34 (DCLK) al pin D6 de Arduino.
- Pin 33 (T\_CS) al pin D5 de Arduino.
- Pin 30 (MOSI) al pin D4 de Arduino.
- Pin 29 (MISO) al pin D3 de Arduino.
- Pin 31 (T\_PEN) al pin D2 de Arduino.

El módulo SD de la pantalla puede ser utilizada para mostrar imágenes en la pantalla almacenadas en su interior, como también para almacenar datos importantes registrados durante el funcionamiento del dispositivo. Para mostrar las imágenes es necesario transformarlas en formato RAW. La interfaz necesaria para su funcionamiento es como se indica a continuación:

- Pin 34 (SCK) al pin D52 de Arduino.
- Pin 22 (SD\_CS) al pin D53 de Arduino.
- Pin 29 (MISO) al pin D50 de Arduino.
- Pin 30 (MOSI) al pin D51 de Arduino.



La pantalla táctil comparte los pines MOSI y MISO con la tarjeta SD, que son controlados por separado con los pines T\_CS y SD\_CS.

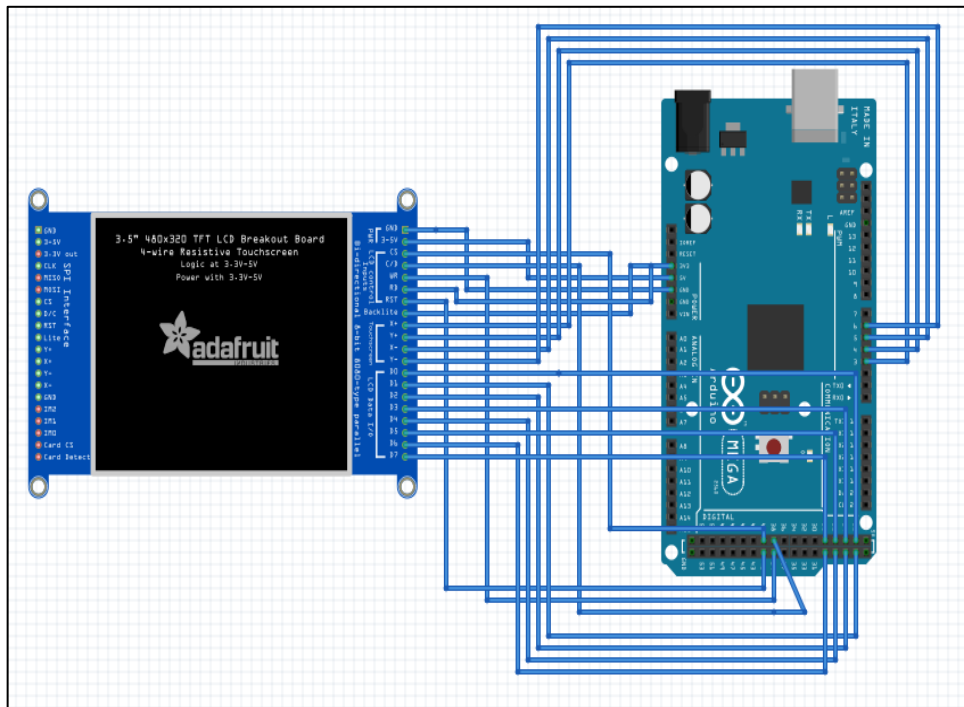


Figura 30. Diagrama de conexión de la pantalla.

Además, para la utilización de este elemento electrónico es necesario instalar las librerías que se indican a continuación.

- <SD.h>
- <UTouch.h>
- <UTFT.h>
- <UTFT\_tinyFAT.h>
- <tinyFAT.h>

Para el control táctil se definen las coordenadas touch de la pantalla de acuerdo al área en la que se limitan los botones del menú de control, de esta manera solo estas quedarán activas y el resto de la pantalla desactivada, lo que permitirá pulsar cualquier parte de la pantalla sin que haya cambios en el funcionamiento, el código para esta acción se indica en la figura 31.

```

codigo2$
141     long x, y; //Touch
142     if(ts.dataAvailable())
143     {
144         ts.read();
145         x = ts.getX();
146         y = ts.getY();
147
148         if((x>=310) && (x<=360))//para boton inicio
149         {
150             if((y>=160) && (y<=220))
151             {
152                 myGLCD.fillRect(VGA_BLUE);
153                 myFiles.loadBitmap(0, 0, 320, 240, "INICIO.RAW");
154                 myGLCD.setColor(VGA_YELLOW);
155                 myGLCD.fillRect(40, 30, 290, 60);
156                 myGLCD.fillRect(40, 70, 290, 100);
157                 myGLCD.fillRect(40, 110, 290, 140);
158                 myGLCD.fillRect(40, 150, 290, 180);
159                 myGLCD.setColor(VGA_RED);
160                 myGLCD.fillRect(5, 30, 30, 60);
161                 myGLCD.fillRect(5, 70, 30, 100);
162                 myGLCD.fillRect(5, 110, 30, 140);
163                 myGLCD.fillRect(5, 150, 30, 180);
164                 myGLCD.setColor(VGA_BLACK);
165                 myGLCD.setFont(BigFont);
166                 myGLCD.print("VIAJES", CENTER, 4);
167                 myGLCD.print("IBARRA - QUITO", CENTER, 40);
168                 myGLCD.print("QUITO - IBARRA", CENTER, 80);

```

Figura 31. Código control touch.

El almacenamiento de datos se realiza con la utilización del módulo SD que viene integrado en la pantalla, para ello se utiliza la librería <SD.h> que permite almacenar los datos requeridos para el análisis estadístico de forma rápida y ordenada. El almacenamiento se realiza al momento de dar fin al dispositivo y los datos se guardan en el archivo VIAJES.CSV de la tarjeta de memoria a través de la ejecución del código que se indica en la figura 32.

```

codigo2$
2231     datos=SD.open("VIAJE.csv",FILE_WRITE);
2232     if(datos)
2233     {
2234         datos.println();
2235         datos.print(titulo);
2236         datos.print(";");
2237         datos.print(".");
2238         datos.print(dia);
2239         datos.print("/");
2240         datos.print(mes);
2241         datos.print("/");
2242         datos.print(a_o);
2243         datos.print(";");
2244         datos.print(".");
2245         if(hora<10){datos.print("0");}
2246         datos.print(hora);
2247         datos.print(";");
2248         if(minuto<10){datos.print("0");}
2249         datos.print(minuto);
2250         datos.print(";");
2251         datos.print(".");
2252         if(horal<10){datos.print("0");}
2253         datos.print(horal);
2254         datos.print(";");
2255         if(minutol<10){datos.print("0");}
2256         datos.print(minutol);
2257         datos.print(";");
2258         datos.print(".");

```

Figura 32. Código almacenamiento de datos.

### 4.3.3 Diagrama electrónico completo.

En la figura 33 se indica el diagrama electrónico final con la conexión del módulo GPS y de la pantalla táctil a través de los cuales se realiza la presentación de datos en tiempo real y el almacenamiento de los mismos en la tarjeta de memoria para su posterior análisis en el software LabVIEW.

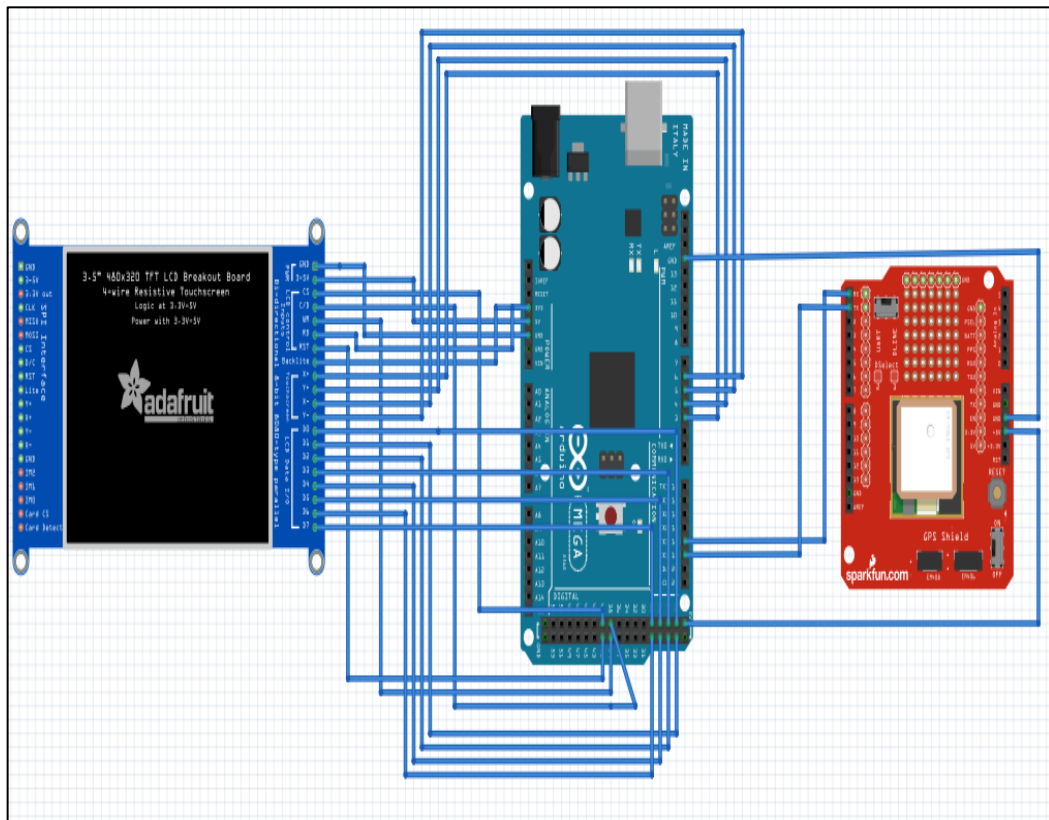


Figura 33. Diagrama electrónico completo.

Para un buen funcionamiento en conjunto de estos elementos electrónicos se define antes del void setup del código el tipo de controlador que utilizan y los pines a los cuales están conectados para así no tener inconvenientes al momento del funcionamiento del dispositivo, estos se declaran de la forma que se indica en el código de la figura 34.

```

codigo2$
82
83 TinyGPSPlus Bus;
84 //Instanciamos la TFT
85 UTFT myGLCD(ITDB32S,38,39,40,41);//definir el tipo de pantalla
86 UTFT_tinyFAT myFiles(&myGLCD); //definir a la tarjeta de memoria
87 char* files320[]={"INICIO.PAW"};
88 char* files[1];
89
90 // Declaramos las fuentes para la TFT
91 extern uint8_t SmallFont[];
92 extern uint8_t BigFont[];
93 extern uint8_t SevenSegNumFont[];
94 extern uint8_t SevenSegNumFontPlusPlus[];
95
96 //Instanciamos el Touch
97 UTouch ts( 6, 5, 4, 3, 2); //declaracion de pines tactil
98 File datos;//dar nombre a la memoria sd
99 #define SD_CS 53 //pin para la sd

```

Figura 34. Código declaración de elementos electrónicos.

Además, el código cargado a la placa Arduino Mega se ejecuta cada segundo gracias a la librería <TimerOne.h> incorporada a las instrucciones de programación. De esta manera en el void setup del código se especifica la activación del timer en valores de microsegundos, como también la velocidad de comunicación del módulo GPS y los pines a los cuales está asignado el módulo de la tarjeta SD, tal como se indica en la figura 35.

```

codigo2$
100
101 void setup() {
102
103 Serial1.begin(9600); //Gps con arduino se comunica con Arduino a 9600 en el puerto Tx Rx (1)
104 Serial.begin(38400); // Arduino con la Pc en el puerto Tx Rx (0)
105 Timer1.initialize(1000000);//cada 1 segundo pero el timer se define en microsegundos
106 //Configuracion de la TFT
107 myGLCD.InitLCD();
108 myGLCD.fillScr(VGA_BLACK);//color de fondo de pantalla
109 myGLCD.setBackColor(VGA_TRANSPARENT);
110 file.initFAT();//comunicacion con la sd
111 //Configuracion del Touch
112 ts.InitTouch();//inicializar el touch
113 ts.setPrecision(PREC_MEDIUM);//defines la presicion del tactil
114 //Configuracion de la SD
115 pinMode(53,OUTPUT);//pin de salida de la memoria
116 if (!SD.begin(SD_CS))
117 {
118 // Serial1.print("ERROR SD");
119 return;
120 }
121
122 myGLCD.fillScr(VGA_BLACK);
123 myFiles.loadBitmap(0, 0, 320, 240, "INICIO.PAW");//cargar la imagen
124
125
126 }
127

```

Figura 35. Código void setup.

#### 4.4 Definición de los lugares de control.

Para la definición de los lugares de control en cada uno de los viajes es necesario delimitar a través de cuadrantes de aproximadamente 15400 metros cuadrados para asegurar el almacenamiento de datos y la toma de posición del autobús, a través de los valores de longitud y latitud proporcionados por el GPS, tal como se indica en la figura 36.

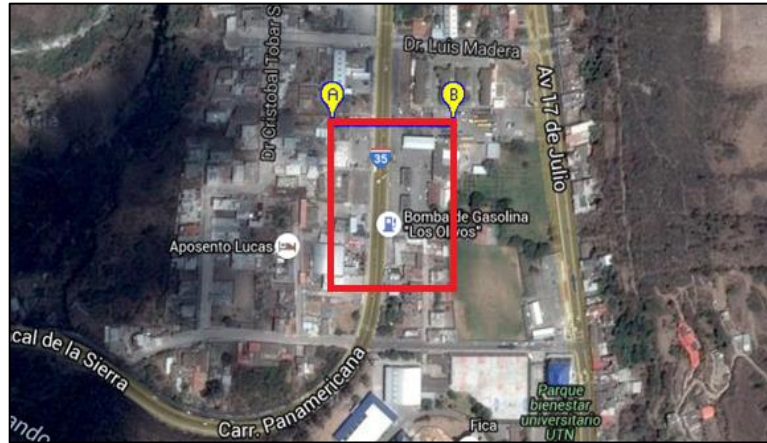


Figura 36. Delimitación de los lugares de control.

En la tabla 8 se indica los diez puntos de control definidos en los cuatro viajes programados en el dispositivo, cada uno de ellos están ubicados en la ruta a seguir por el autobús.

Tabla 8. Lugares de control.

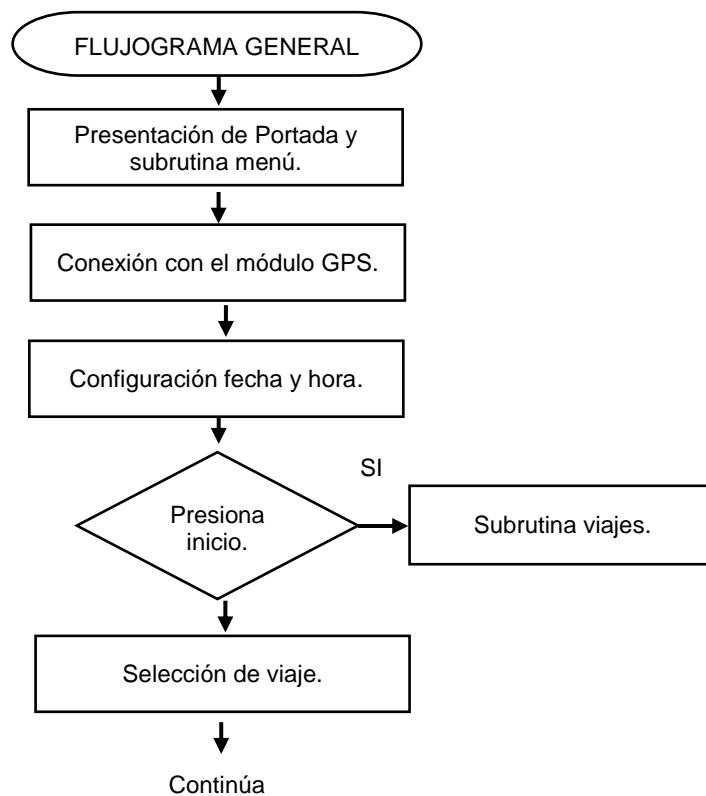
| Lugares de control | Viajes          |                 |                 |                 |
|--------------------|-----------------|-----------------|-----------------|-----------------|
|                    | Ibarra-Quito    | Quito-Ibarra    | Ibarra-Tulcán   | Tulcán-Ibarra   |
| 1                  | Terminal Ibarra | Terminal Quito  | Terminal Ibarra | Terminal Tulcán |
| 2                  | La Florida      | Calderón        | Olivo           | Control Policía |
| 3                  | San Antonio     | Guallabamba     | Salinas         | Guagua Negro    |
| 4                  | Atuntaqui       | Tabacundo       | Juncal          | Julio Andrade   |
| 5                  | Otavalo         | Cajas           | Bolívar         | San Gabriel     |
| 6                  | Cajas           | Otavalo         | San Gabriel     | Bolívar         |
| 7                  | Tabacundo       | Atuntaqui       | Julio Andrade   | Juncal          |
| 8                  | Guallabamba     | San Antonio     | Guagua Negro    | Salinas         |
| 9                  | Calderón        | La Florida      | Control Policía | Olivo           |
| 10                 | Terminal Quito  | Terminal Ibarra | Terminal Tulcán | Terminal Ibarra |

#### 4.5 Flujogramas del dispositivo.

El funcionamiento del software está conformado por 5 subrutinas para cumplir con los objetivos planteados, cada una de ellas realiza una función específica para recolectar y almacenar datos para su posterior análisis. Las subrutinas utilizadas son las siguientes:

- Subrutina menú.
- Subrutina viajes.
- Subrutina funcionamiento.
- Subrutina almacenamiento.
- Subrutina home.

El dispositivo está desarrollado con la finalidad de presentar datos de tiempos de atrasos y adelantos en tiempo real como también de su almacenamiento para el respectivo análisis estadístico en el software desarrollado. El funcionamiento general del dispositivo desarrollado se presenta en el flujograma de la figura 37.



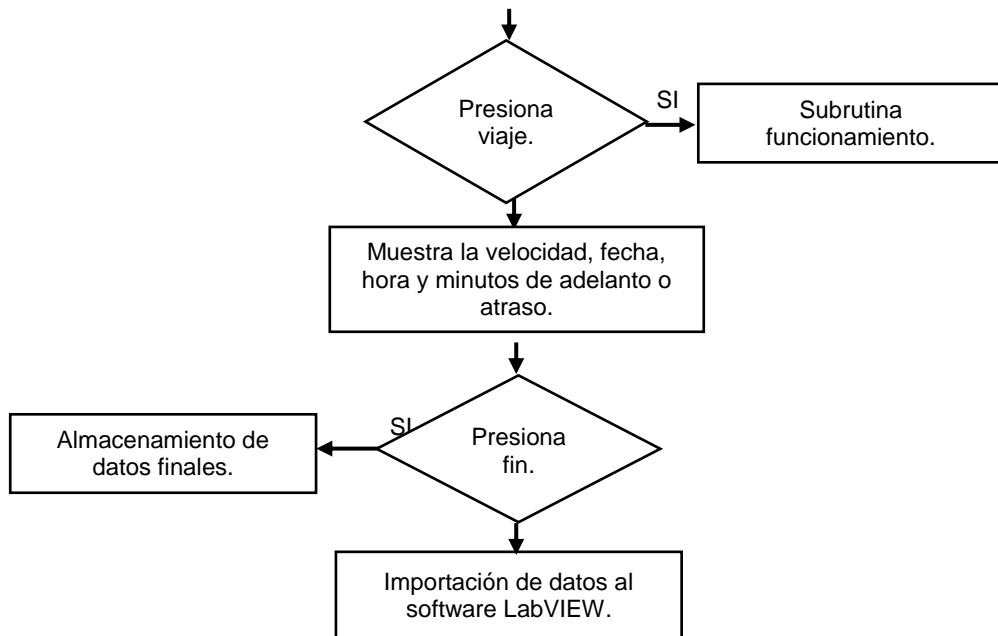
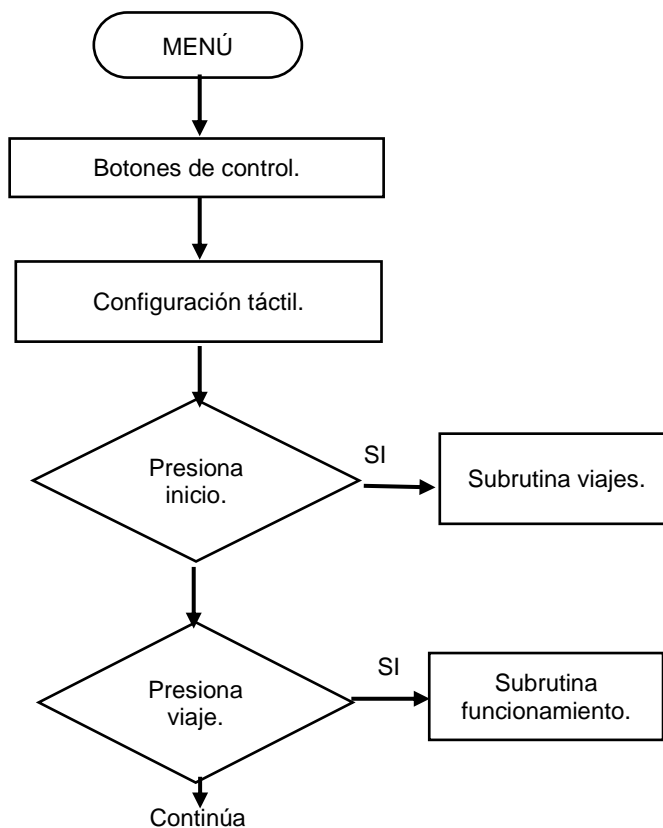


Figura 37. Flujograma general.

#### 4.5.1 Flujograma subrutina menú.

Esta subrutina permite la identificación de los botones de control que ayudan a iniciar el dispositivo, visualizar los viajes y finalizar su funcionamiento. El proceso de esta subrutina se observa en el flujograma de la figura 38.



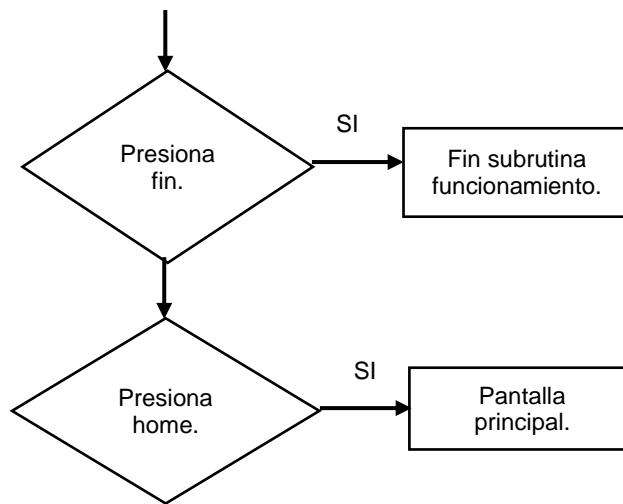


Figura 38. Flujograma subrutina menú.

#### 4.5.2 Flujograma subrutina viajes.

Al presionar el botón inicio que se encuentra en la pantalla principal se presenta la segunda ventana en la que se observa la lista de viajes a seleccionar. Al seleccionar el viaje indicado se presenta una tercera ventana en la que se observa información brindada por el dispositivo hacia el conductor en tiempo real con datos tomados directamente del GPS. El funcionamiento de la ventana viajes se indica en el flujograma de la figura 39.

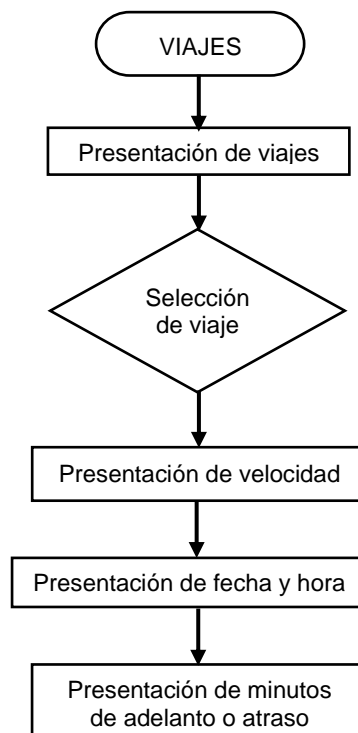


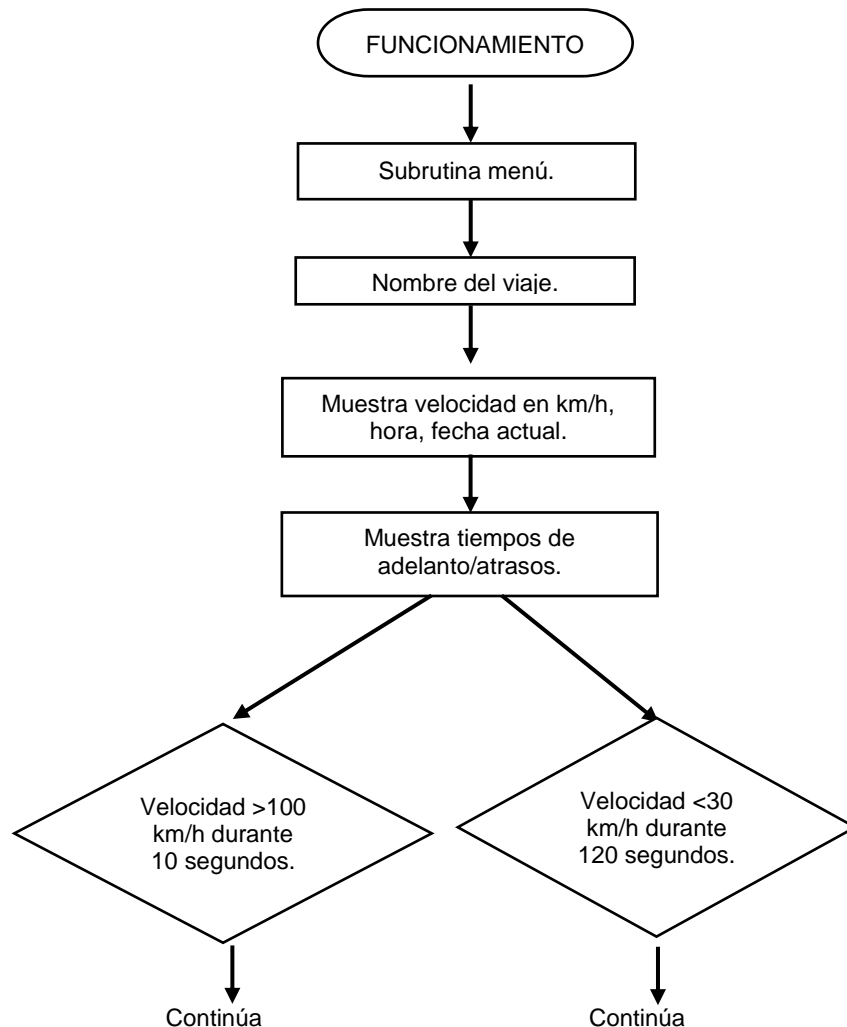
Figura 39. Flujograma viajes.



### 4.5.3 Flujograma subrutina funcionamiento.

Esta subrutina inicia al momento de seleccionar el viaje a realizarse, con la presentación de una tercera pantalla en el dispositivo con datos proporcionados por el módulo GPS. Durante el funcionamiento se indica la velocidad en km/h, hora y fecha actual, y mientras se pasa por los lugares de control los tiempos de adelanto o atraso con diferentes colores para diferenciar del anterior.

Además, calcula la velocidad a la que va el autobús y si pasa de 90km/h durante un tiempo de 10 segundos abre el archivo VMAX.CSV y almacena información de latitud, longitud, hora y fecha; así también si el autobús permanece durante 50 segundos a una velocidad menor a 30 km/h abre el archivo VMIN.CSV de la tarjeta de memoria y almacena la misma información que en la velocidad máxima para su posterior análisis.



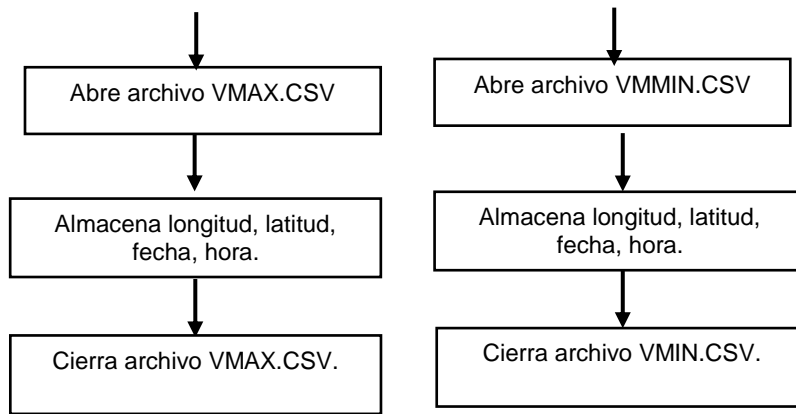


Figura 40. Flujograma subrutina funcionamiento.

#### 4.5.4 Flujograma subrutina almacenamiento de datos.

En esta subrutina se realiza el almacenamiento de datos recolectados a través del funcionamiento del dispositivo, que serán almacenados en el archivo VIAJESBUS.CSV al momento de finalizar su funcionamiento. Los datos almacenados son: el nombre del viaje realizado, la hora y fecha de salida, la hora de tiempos de control y tiempos reales.

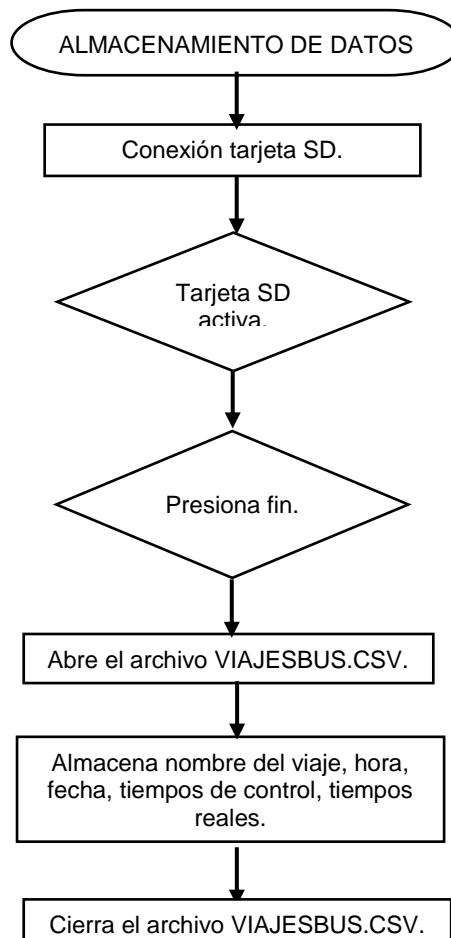


Figura 41. Flujograma subrutina almacenamiento de datos.

#### 4.5.5 Flujograma subrutina home.

Este proceso permite regresar a la pantalla principal del dispositivo, así como también establecer las variables a sus valores iniciales.

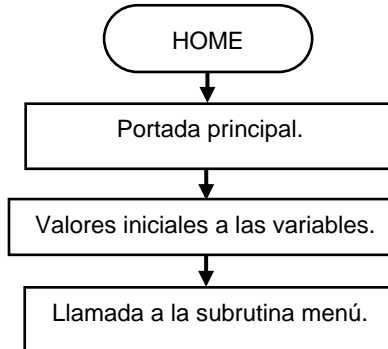


Figura 42. Flujograma subrutina home.

#### 4.6 Diseño del software de análisis de datos en LabVIEW.

El programa de análisis de datos es construido a través del software LabVIEW con la utilización de la herramienta Report Generation Toolkit para Microsoft Excel que permite la importación de datos de los archivos almacenados en la tarjeta de memoria del dispositivo.

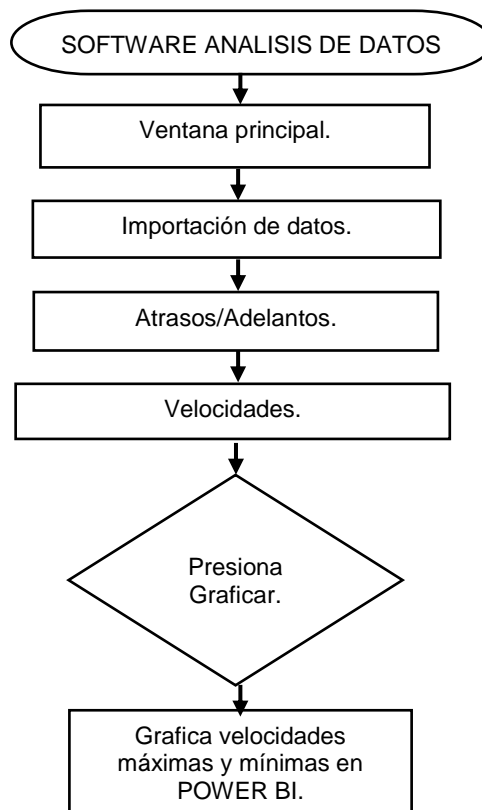


Figura 43. Flujograma software análisis de datos.

#### 4.6.1 Portada.

El programa de análisis de datos está diseñado a través de un Tab control con cuatro páginas para mostrar información específica como es: ATRASOS/ADELANTOS, ESTADÍSTICAS Y VELOCIDADES. En la portada se indica el nombre de la institución con el respectivo tema de trabajo de grado, como se indica en la figura 44.



Figura 44. Portada programa análisis de datos.

#### 4.6.2 Atrasos/Adelantos.

En la segunda página de atrasos/adelantos se presenta información de los minutos de atrasos y adelantos realizados por el autobús en los diferentes viajes durante un tiempo determinado. La información presentada son tiempos totales realizados por el autobús, así como también tiempos promedio según la cantidad de viajes registrados.

La primera parte está formada por un selector del nombre de cooperativa y número de vehículo que al presionar el botón de conectar se establece la conexión con la tarjeta de memoria y de forma inmediata se prende un led rojo,

caso contrario si la tarjeta de memoria no se encuentra insertada en la Pc, este led no se enciende. Los botones se encuentran definidos como se indica en la figura 45.

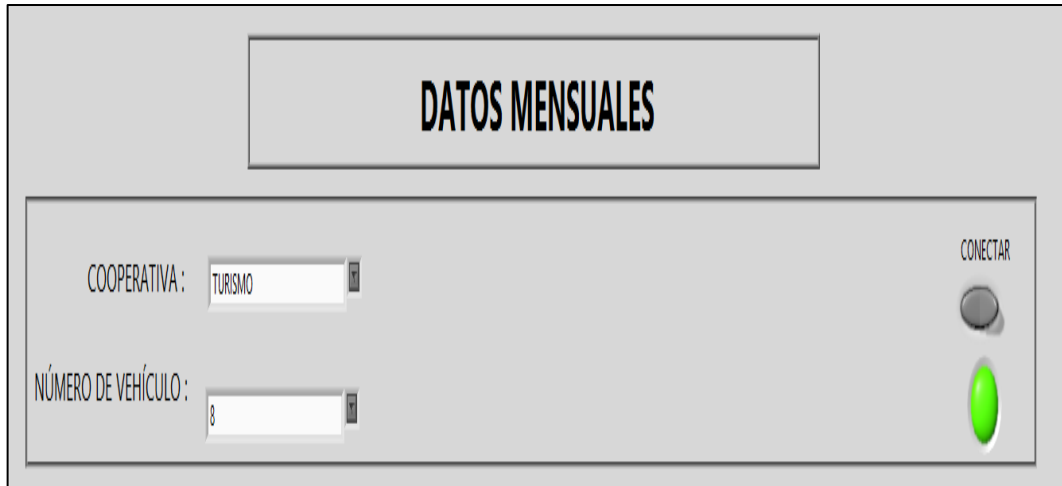


Figura 45. Botones para la conexión con la SD.

El código utilizado es el que se indica en la figura 46, que está conformado por dos combo box para la selección de la cooperativa y número del vehículo, que al seleccionar los correctos se ejecuta un código que permite la importación de los datos almacenados en el archivo Viaje.CSV, para pegarlos en el archivo ViajesBus.xlsm que es en el que se realiza todos los cálculos necesarios para la presentación de la respectiva información.

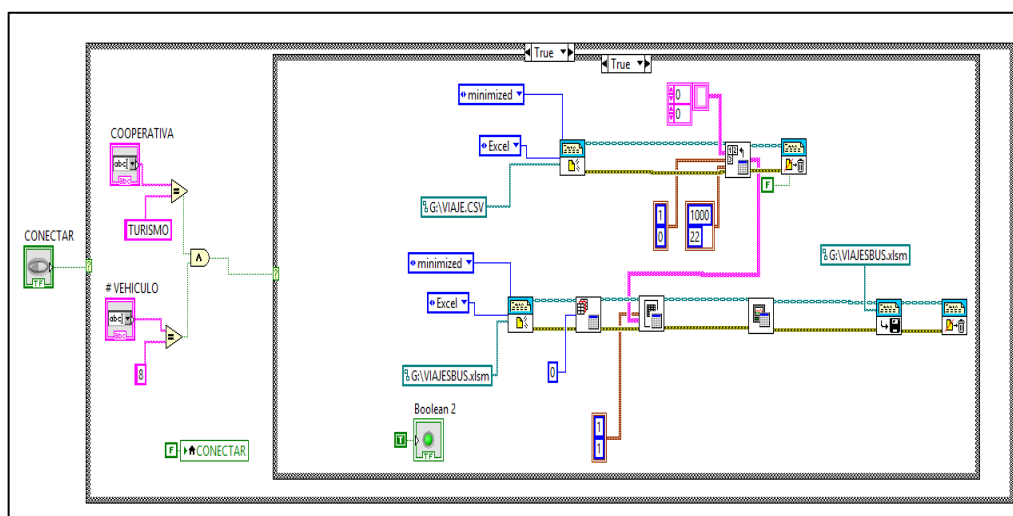


Figura 46. Código para selección de cooperativa y vehículo.

En la segunda parte se presentan datos específicos de: total de viajes realizados, total de minutos de atraso, total de minutos de adelanto y frente a cada uno de estos un promedio según los viajes realizados por el autobús, como se indica en la figura 47.

| DATOS MENSUALES      |                             |                          |
|----------------------|-----------------------------|--------------------------|
| COOPERATIVA :        | TURISMO                     | CONECTAR                 |
| NÚMERO DE VEHÍCULO : | 8                           |                          |
| TOTAL DE VIAJES      | TOTAL MINUTOS DE ATRASO :   | PROMEDIO MINUTOS/VIAJE : |
| 26                   | -35                         | -1.35                    |
|                      | TOTAL MINUTOS DE ADELANTO : | PROMEDIO MINUTOS/VIAJE : |
|                      | 11                          | 0.42                     |
| CALCULAR             |                             |                          |

Figura 47. Ventana de datos generales.

Estos datos se muestran a través del código de la figura 48 que al ser ejecutado abre el archivo ViajesBus.xslm para importar los datos ya procesados a los respectivos cuadros que se muestran en la primera página. Cada uno de ellos contiene tiempos específicos de total de viajes, tiempos de adelantos y atrasos, como también un promedio de cada uno.

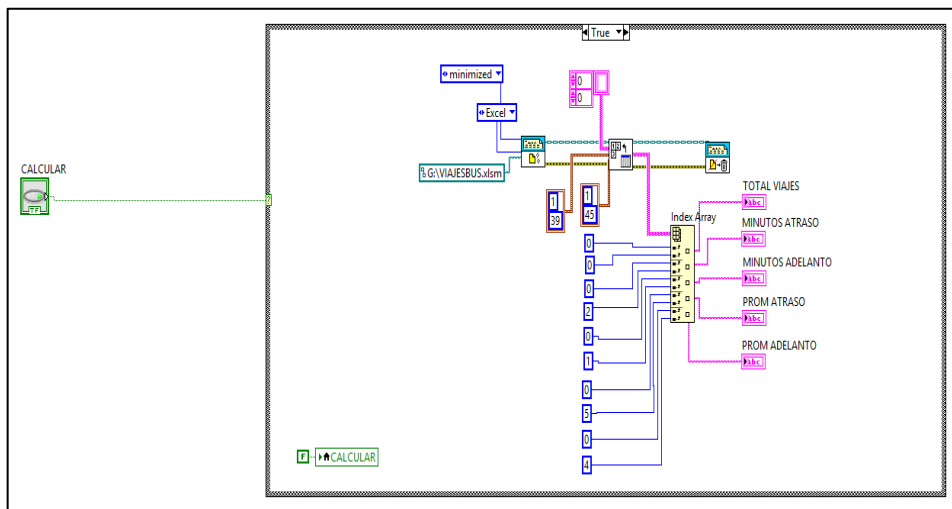


Figura 48. Código para la importación de datos.

### 4.6.3 Estadísticas.

La tercera página de estadísticas muestra información de los viajes realizados por el autobús, así como también viajes específicos para conocer los minutos de adelantos o atrasos en los diferentes puntos de control definidos en la programación.

Los viajes se seleccionan a través de un combo box como se presenta en la figura 49, y la selección de viajes específicos se los realiza mediante e control que se indica en la figura 50; luego de seleccionar el rango de viajes es necesario presionar el botón que se encuentran junto para de esta manera refrescar los datos seleccionados.

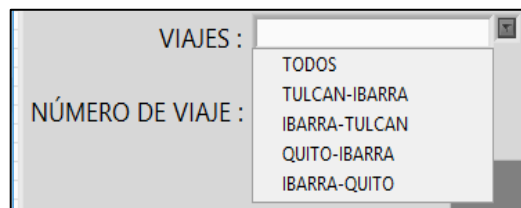


Figura 49. Selección de viajes.

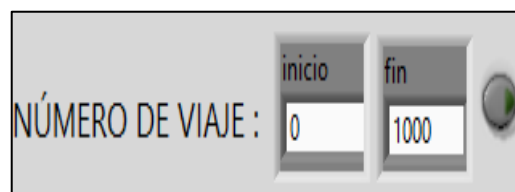


Figura 50. Selección de número de viaje.

Esto se obtiene a través del código de la figura 51 que importa los datos de los viajes del archivo ViajesBus.xlsm, que es en el que se realizan todos los cálculos pertinentes para presentar el análisis estadístico en la respectiva ventana del software desarrollado. Los cálculos se realizan en el archivo Excel a través de la utilización de macros grabadas en el mismo.

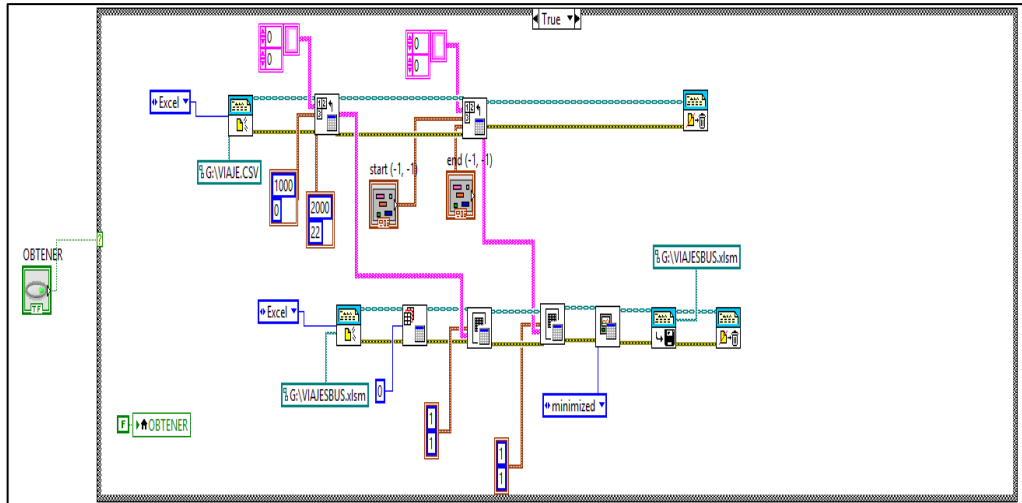


Figura 51. Código para la selección de viajes.

La información de cada viaje se presenta en la tabla de la figura 52, en la cual se indica cada viaje con su número respectivo, fecha y hora de salida, así también se indica los atrasos de cada viaje en los puntos de control establecidos inicialmente. En la parte inferior de la tabla se presenta un promedio de los minutos de atrasos y adelantos en cada uno de los puntos de control de los viajes seleccionados.

VIAJES : IBARRA-QUITO

NÚMERO DE VIAJE : inicio 0 fin 1000

| LUGARES DE CONTROL |       |              |            |        |            |             |           |         |       |           |             |          |          |
|--------------------|-------|--------------|------------|--------|------------|-------------|-----------|---------|-------|-----------|-------------|----------|----------|
| #                  | VIAJE | FECHA        | HH SALIDA  | IBARRA | LA FLORIDA | SAN ANTONIO | ATUNTAQUI | OTAVALO | CAJAS | TABACUNDO | GUALLABAMBA | CARCELEN | TERMINAL |
| V                  | 1     | IBARRA-QUITO | .29/6/2016 | .17:01 | 0          | 0           | 0         | 0       | 0     | 0         | 0           | 0        | 0        |
| I                  | 3     | IBARRA-QUITO | .30/6/2016 | .19:45 | 0          | 0           | 0         | 0       | 0     | 0         | 0           | 0        | 0        |
| A                  | 4     | IBARRA-QUITO | .30/6/2016 | .19:46 | 0          | 0           | 0         | 0       | 0     | 0         | 0           | 0        | 0        |
| J                  | 5     | IBARRA-QUITO | .30/6/2016 | .19:46 | 0          | 0           | 1         | 0       | 0     | 0         | 1           | 2        | 2        |
| E                  | 8     | IBARRA-QUITO | .0/36/2016 | .22:29 | 0          | 0           | 0         | 0       | 0     | 0         | 0           | 0        | 0        |
| S                  | 9     | IBARRA-QUITO | .30/6/2016 | .22:30 | 0          | 0           | 0         | 0       | 0     | 0         | 0           | 0        | 0        |
|                    | 12    | IBARRA-QUITO | .30/6/2016 | .13:00 | 0          | 0           | -1        | -1      | -3    | -3        | -3          | -1       | -1       |
|                    | 15    | IBARRA-QUITO | .0/0/2000  | .19:00 | 0          | 0           | 0         | 0       | 0     | 0         | 0           | 0        | 0        |
|                    | 16    | IBARRA-QUITO | .0/0/2000  | .19:00 | 0          | 0           | 0         | 0       | 0     | 0         | 0           | 0        | 0        |
|                    | 17    | IBARRA-QUITO | .0/0/2000  | .09:40 | 0          | 0           | 0         | 0       | 0     | 0         | 0           | 0        | 0        |
|                    | 18    | IBARRA-QUITO | .8/7/2016  | .09:43 | 0          | 0           | 0         | 0       | 0     | 0         | 0           | 0        | 0        |
|                    | 19    | IBARRA-QUITO | .22/7/2016 | .22:18 | 0          | 0           | 0         | 0       | 0     | 0         | 0           | 0        | 0        |
|                    | 20    | IBARRA-QUITO | .22/7/2016 | .22:34 | 0          | 0           | 0         | 0       | 0     | 0         | 0           | 0        | 0        |
|                    | 21    | IBARRA-QUITO | .22/7/2016 | .22:38 | 0          | 0           | 0         | 0       | 0     | 0         | 0           | 0        | 0        |
| <b>ATRASOS</b>     |       |              |            | 0      | 0          | -1          | -1        | -3      | -3    | -3        | -3          | -1       | -1       |
| <b>ADELANTOS</b>   |       |              |            | 0      | 0          | 1           | 0         | 0       | 0     | 0         | 1           | 2        | 2        |

CALCULAR

Figura 52. Tabla de adelantos y atrasos.



Estos datos son obtenidos a través del botón calcular que ejecuta el código diseñado y que se presenta en la figura 53. Este consta por cinco códigos diferentes que se ejecutan según el viaje seleccionado en el panel frontal, y cada uno de ellos activa una macro distinta en el archivo VIAJESBUS.xlsm para realizar el cálculo de los viajes y tiempos de adelantos/atrasos.

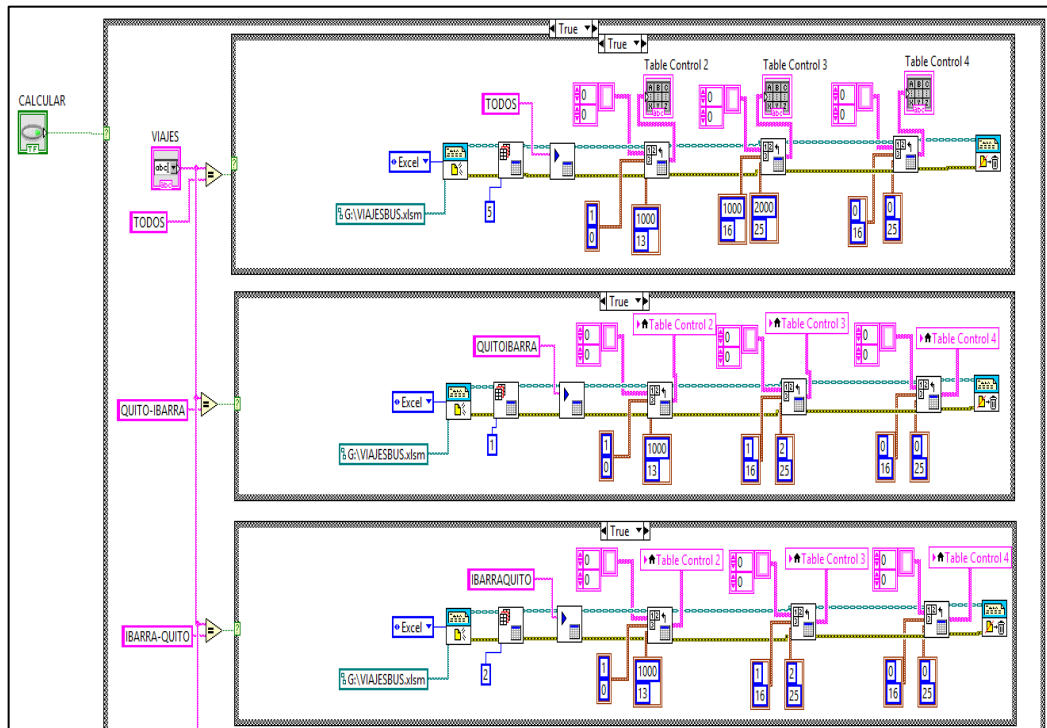


Figura 53. Código para la importación de viajes.

#### 4.6.4 Velocidades.

En la cuarta página de velocidades se indican dos tablas en las cuales se presentan los lugares en los que el autobús tiene valores mínimos o máximos de velocidad. Estos datos son almacenados en la tarjeta de memoria al momento en que el autobús pasa de 90km/h durante 10 segundos y cuando permanece con velocidad menor a 30km/h durante 50 segundos.

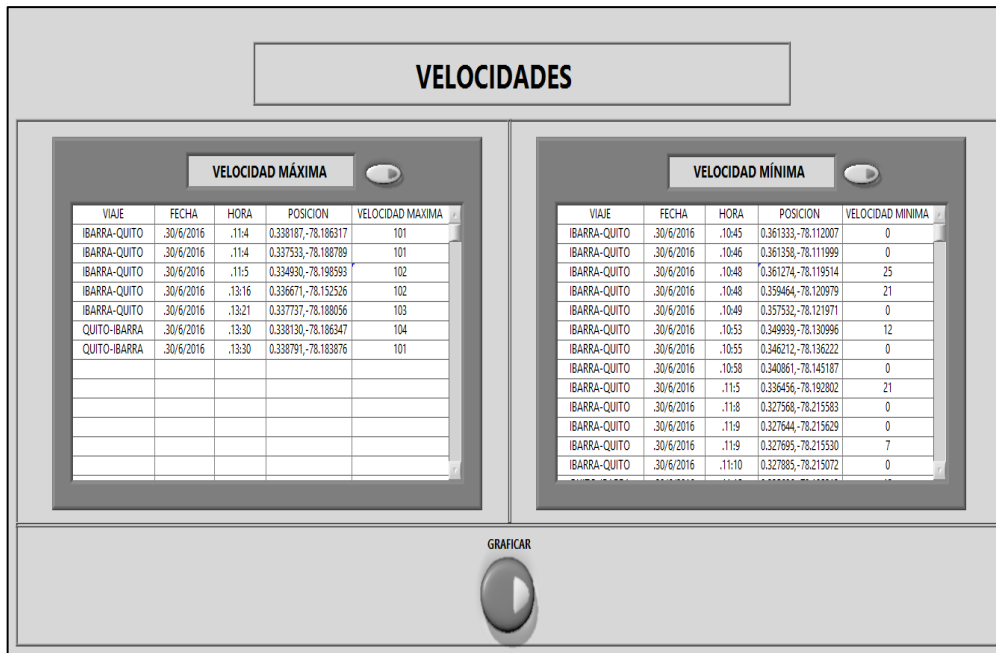


Figura 54. Ventana de velocidades.

En la primera tabla como en la figura 55 se indica las velocidades máximas que realizó el vehículo en los diferentes viajes. Estos datos son indicados con la información de nombre del viaje realizado, la posición, hora, fecha y la velocidad que se obtuvo durante su funcionamiento.

| <b>VELOCIDAD MÁXIMA</b> |            |        |                     |                  |
|-------------------------|------------|--------|---------------------|------------------|
| VIAJE                   | FECHA      | HORA   | POSICION            | VELOCIDAD MAXIMA |
| IBARRA-QUITO            | .30/6/2016 | .11:4  | 0.338187,-78.186317 | 101              |
| IBARRA-QUITO            | .30/6/2016 | .11:4  | 0.337533,-78.188789 | 101              |
| IBARRA-QUITO            | .30/6/2016 | .11:5  | 0.334930,-78.198593 | 102              |
| IBARRA-QUITO            | .30/6/2016 | .13:16 | 0.336671,-78.152526 | 102              |
| IBARRA-QUITO            | .30/6/2016 | .13:21 | 0.337737,-78.188056 | 103              |
| QUITO-IBARRA            | .30/6/2016 | .13:30 | 0.338130,-78.186347 | 104              |
| QUITO-IBARRA            | .30/6/2016 | .13:30 | 0.338791,-78.183876 | 101              |
|                         |            |        |                     |                  |
|                         |            |        |                     |                  |
|                         |            |        |                     |                  |
|                         |            |        |                     |                  |
|                         |            |        |                     |                  |
|                         |            |        |                     |                  |

Figura 55. Tabla de velocidad máxima.

Estos datos se obtienen al presionar el botón que se encuentra junto al cuadro de velocidad máxima y que ejecuta el código de la figura 56. Este código abre un archivo de nombre Vmax. CSV en el que se almacenan todas las velocidades máximas, y las copia al archivo xlsm en las que se realiza todas las operaciones necesarias para su presentación en la tabla de la página velocidades.

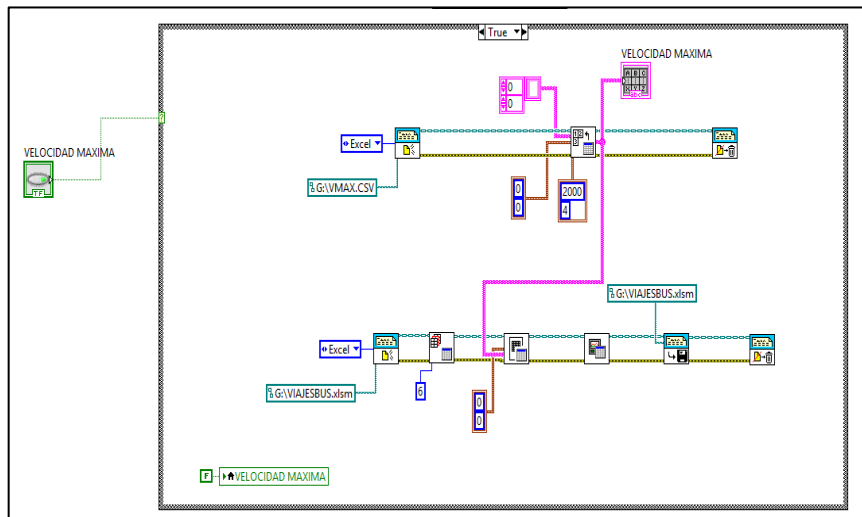


Figura 56. Código para importar la velocidad máxima.

En la segunda tabla de velocidad mínima se indica la información de las velocidades mínimas alcanzadas durante los diferentes viajes. Estas velocidades se obtienen al durar un tiempo de 50 segundos bajo los 30km/h y se indican en la tabla de la figura 57.

| VELOCIDAD MÍNIMA |            |        |                     |                  |
|------------------|------------|--------|---------------------|------------------|
| VIAJE            | FECHA      | HORA   | POSICION            | VELOCIDAD MINIMA |
| IBARRA-QUITO     | .30/6/2016 | .10:45 | 0.361333,-78.112007 | 0                |
| IBARRA-QUITO     | .30/6/2016 | .10:46 | 0.361358,-78.111999 | 0                |
| IBARRA-QUITO     | .30/6/2016 | .10:48 | 0.361274,-78.119514 | 25               |
| IBARRA-QUITO     | .30/6/2016 | .10:48 | 0.359464,-78.120979 | 21               |
| IBARRA-QUITO     | .30/6/2016 | .10:49 | 0.357532,-78.121971 | 0                |
| IBARRA-QUITO     | .30/6/2016 | .10:53 | 0.349939,-78.130996 | 12               |
| IBARRA-QUITO     | .30/6/2016 | .10:55 | 0.346212,-78.136222 | 0                |
| IBARRA-QUITO     | .30/6/2016 | .10:58 | 0.340861,-78.145187 | 0                |
| IBARRA-QUITO     | .30/6/2016 | .11:5  | 0.336456,-78.192802 | 21               |
| IBARRA-QUITO     | .30/6/2016 | .11:8  | 0.327568,-78.215583 | 0                |
| IBARRA-QUITO     | .30/6/2016 | .11:9  | 0.327644,-78.215629 | 0                |
| IBARRA-QUITO     | .30/6/2016 | .11:9  | 0.327695,-78.215530 | 7                |
| IBARRA-QUITO     | .30/6/2016 | .11:10 | 0.327885,-78.215072 | 0                |

Figura 57. Cuadro de velocidad mínima.

Este código al igual que en la velocidad máxima se ejecuta al presionar el botón junto al cuadro velocidad mínima y de esta manera se ejecuta el programa de la figura 57, que tiene el mismo funcionamiento del código de la figura 58, pero en este caso el archivo seleccionado es de nombre Vmin.CSV.

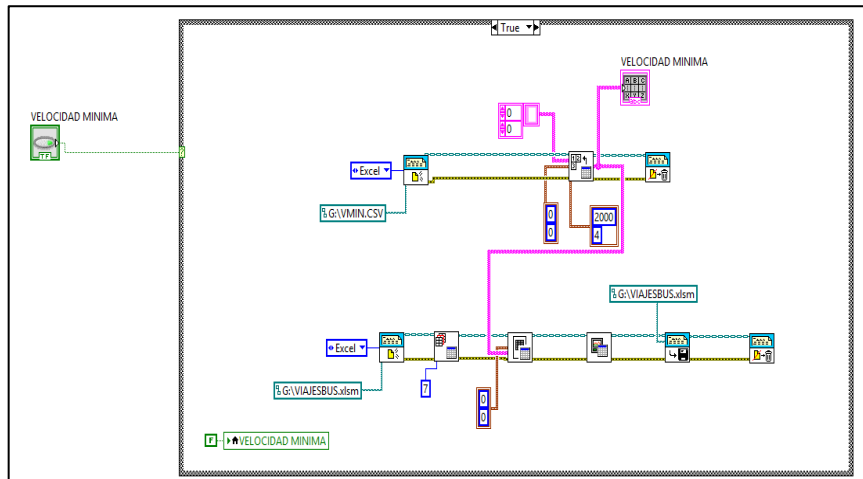


Figura 58. Código para importar la velocidad máxima.

Además, en esta página se muestra un botón graficar, que ejecuta el programa PowerBi que ofrece una vista de 360 grados con la información más importante en un solo lugar; además, los datos se actualizan en tiempo real permitiendo observar datos estadísticos en 2D y 3D, de esta manera en su entorno se grafican en un mapa los lugares en los que el autobús obtuvo las velocidades mínimas y máximas. El botón para ejecutar el programa es como el que se indica en la figura 59.



Figura 59. Botón de representación gráfica.

El código realizado para la ejecución del programa de representación gráfica es como se indica en la figura 60, que está conformado por un System Exec VI que permite ejecutar la aplicación Power BI con solo colocar la dirección en la que se encuentra en el ordenador.

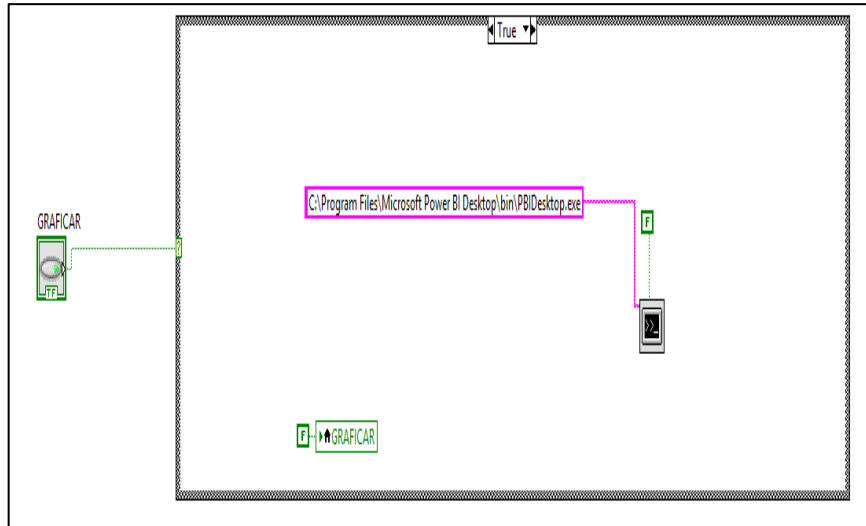


Figura 60. Código para la ejecución de Power BI.

Al ejecutar este programa y actualizar la ventana inicial del mismo se indica de forma gráfica y con puntos de color verde los lugares con velocidad máxima, mientras que con color rojo se indican los lugares con velocidad mínima, tal como se indica en la figura 61.

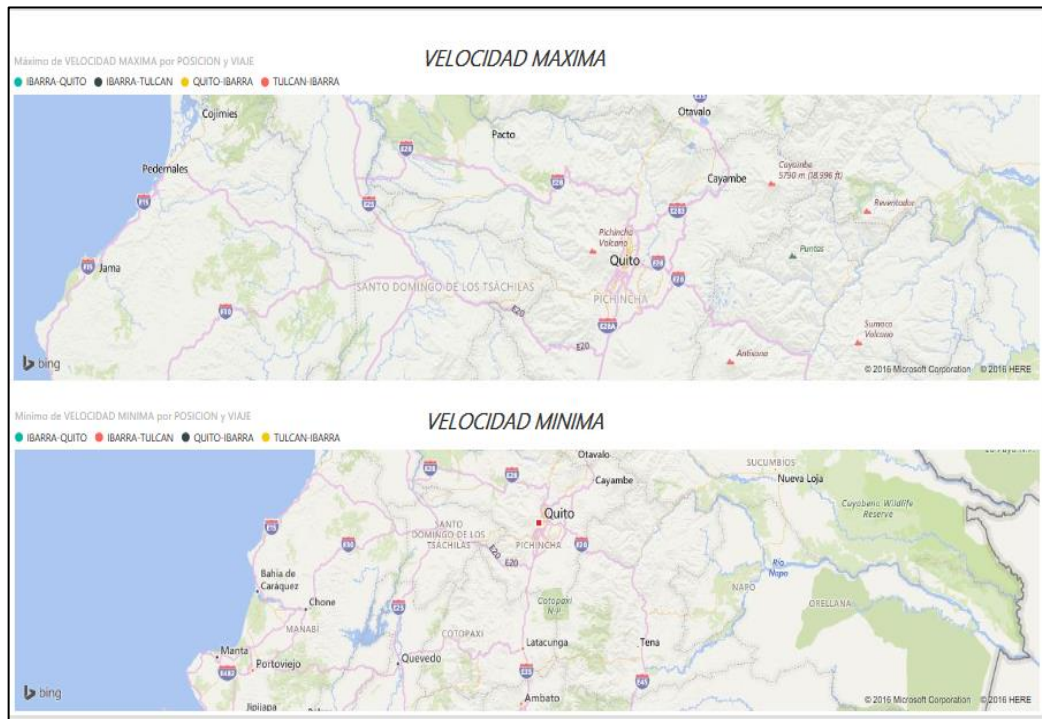


Figura 61. Representación gráfica de velocidades máximas y mínimas.

#### 4.7 Pruebas de funcionamiento.

##### 4.7.1 Pruebas del Hardware.

Para verificar la funcionalidad del dispositivo se realiza una prueba de viaje Ibarra-Quito tomando en cuenta los tiempos de viaje a cada uno de los lugares de control establecidos en la programación. Estos tiempos se indica en la tabla 9.











Tabla 9. Tiempos de viaje Ibarra-Quito.









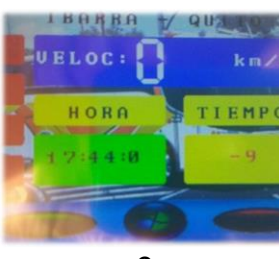

| <b>IBARRA-QUITO</b> |                           |                                 |
|---------------------|---------------------------|---------------------------------|
| <b>Nº</b>           | <b>LUGARES DE CONTROL</b> | <b>TIEMPO DE VIAJE(minutos)</b> |
| 1                   | Terminal Ibarra           | 0                               |
| 2                   | La Florida                | 5                               |
| 3                   | San Antonio               | 3                               |
| 4                   | Atuntaqui                 | 10                              |
| 5                   | Otavalo                   | 15                              |
| 6                   | Cajas                     | 15                              |
| 7                   | Tabacundo                 | 10                              |
| 8                   | Guallabamba               | 30                              |
| 9                   | Calderón                  | 20                              |
| 10                  | Terminal Quito            | 10                              |

##### 4.7.1.1 Pruebas de tiempos de Adelanto y Atraso.

Estos datos se obtienen a través del funcionamiento del dispositivo indicando en la pantalla y con colores diferentes los minutos de adelanto o atraso en cada uno de los lugares de control establecidos en la ruta a seguir por el autobús. En la tabla 10 se indica los lugares, la hora de control, la hora actual a la que pasa el autobús por dichos lugares, el tiempo de adelanto o atraso tomados del dispositivo y una foto del lugar exacto de la adquisición de información.

Tabla 10. Tiempos de viaje Ibarra-Quito.

| HORA CONTROL | HORA ACTUAL | TIEMPO DE ADELANTO O ATRASO   | LUGAR   | OBS.     |
|--------------|-------------|---|---|----------|
| 15:32:26     | 15:32:26    | <br>0    | <br>Terminal Ibarra | Correcto |
| 15:37:26     | 15:40:11    | <br>-2   | <br>La Florida      | Correcto |
| 15:40:26     | 15:43:21    | <br>-2 | <br>San Antonio   | Correcto |
| 15:50:26     | 15:50:20    | <br>0  | <br>Atuntaqui     | Correcto |
| 16:05:26     | 16:01:18    | <br>4  | <br>Otavalo       | Correcto |

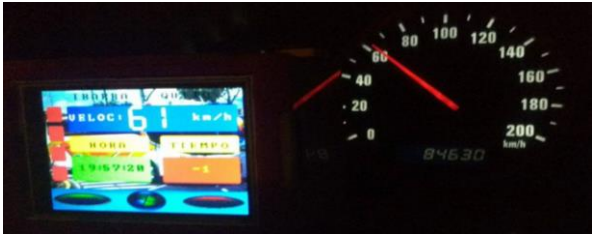


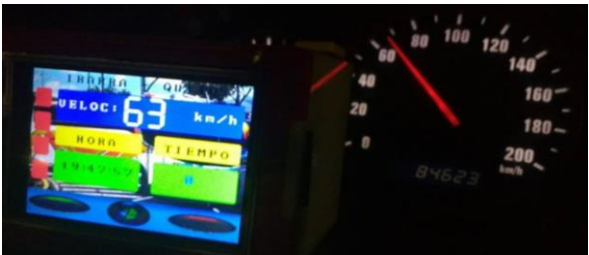
|          |          |   |  |          |
|----------|----------|---|--|----------|
| 16:20:26 | 16:19:57 |  <p>0</p>    |  <p>Cajas</p>            | Correcto |
| 16:30:26 | 16:31:09 |  <p>0</p>    |  <p>Tabacundo</p>        | Correcto |
| 17:00:26 | 17:01:10 |  <p>0</p>   |  <p>Guallabamba</p>     | Correcto |
| 17:20:26 | 17:23:34 |  <p>-3</p> |  <p>Calderón</p>       | Correcto |
| 17:30:26 | 17:39:40 |  <p>-9</p> |  <p>Terminal Quito</p> | Correcto |



#### 4.7.1.2 Pruebas de velocidad.

Para verificar la velocidad proporcionada por el módulo GPS instalado en el dispositivo se realizan cuatro pruebas de funcionamiento comparando los valores con los obtenidos en el velocímetro del autobús. Estos datos y su diferencia se indica en la tabla 11.

Tabla 11. Prueba de velocidad.

| PRUEBAS DE VELOCIDAD  |  |
|---|--|
| VELOCIDADES   | DIFERENCIA(km/h)   |
|    | <p>Velocidad de 61 km/h.<br/>No existe diferencia de velocidad entre los dos valores.</p>                      |
|  | <p>Velocidad de 30km/h en el vehículo.<br/>Velocidad de 28km/h en el dispositivo.<br/>Diferencia de 2km/h.</p> |
|  | <p>Velocidad de 45 km/h.<br/>No existe diferencia de velocidad entre los dos valores.</p>                      |
|  | <p>Velocidad de 66km/h en el vehículo.<br/>Velocidad de 63km/h en el dispositivo.<br/>Diferencia de 3km/h.</p> |

Con los datos obtenidos a través de las pruebas realizadas se llega a la conclusión de que la velocidad proporcionada por el módulo GPS es muy exacta a comparación con la velocidad adquirida por el velocímetro del autobús obteniendo resultados satisfactorios.

#### 4.7.2 Pruebas del Software.

Para verificar que los datos almacenados en la tarjeta de memoria son iguales a los presentados en la pantalla del dispositivo se indica la figura 62, en la que se observa los tiempos de adelanto y atraso obtenidos en la prueba de viaje Ibarra-Quito. Se puede observar que los datos obtenidos son iguales a los presentados en la tabla 10, llegando a la conclusión de que el almacenamiento se realiza de forma correcta y exacta con los valores indicados por el dispositivo.

| LUGARES DE CONTROL |              |           |           |        |            |             |           |         |       |           |             |          |          |
|--------------------|--------------|-----------|-----------|--------|------------|-------------|-----------|---------|-------|-----------|-------------|----------|----------|
| #                  | VIAJE        | FECHA     | HH SALIDA | IBARRA | LA FLORIDA | SAN ANTONIO | ATUNTAQUI | OTAVALO | CAJAS | TABACUNDO | GUALLABAMBA | CALDERON | TERMINAL |
| 1                  | IBARRA-QUITO | 27/7/2016 | .15:32    | 0      | -2         | -2          | 0         | 4       | 0     | 0         | 0           | -3       | -9       |
| <b>ATRASOS</b>     |              |           |           | 0      | -2         | -2          | 0         | 0       | 0     | 0         | 0           | -3       | -9       |
| <b>ADELANTOS</b>   |              |           |           | 0      | 0          | 0           | 0         | 4       | 0     | 0         | 0           | 0        | 0        |

Figura 62. Tiempos de adelanto y atraso Ibarra-Quito.

Las velocidades máximas y mínimas obtenidas durante el viaje realizado se indican en la figura 63, en la que se observa el lugar con los valores de longitud y latitud, la fecha, hora y el valor de velocidad obtenido al momento en el que el autobús realiza dicha maniobra.

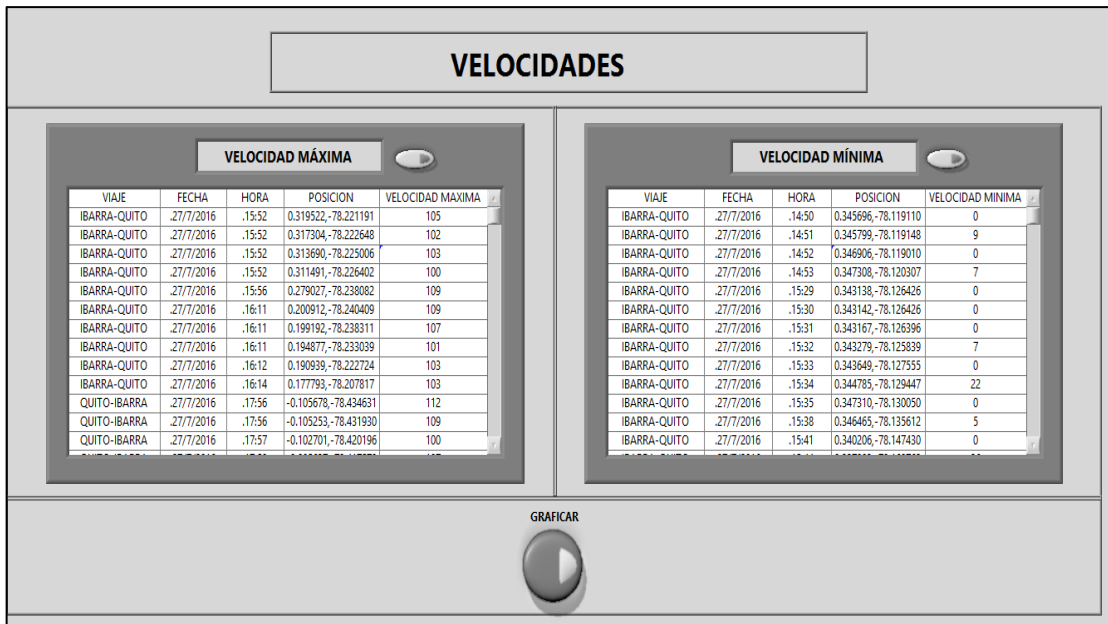


Figura 63. Velocidades máximas y mínimas Ibarra-Quito.

Además, con la ejecución del programa Power Bi se puede observar a través de un mapa los lugares en los que el autobús obtiene las velocidades máximas y mínimas durante su desplazamiento. Las velocidades obtenidas en el viaje Ibarra- Quito se puede observar en la figura 64.

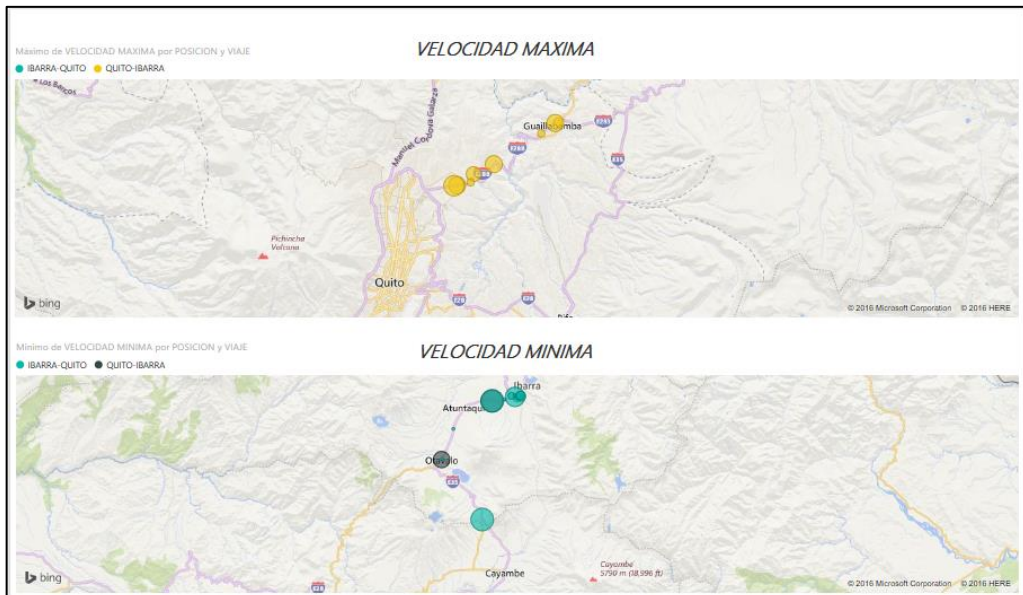


Figura 64. Presentación gráfica de velocidades Ibarra-Quito.

## 4.8 Manual de funcionamiento.

### 4.8.1 Manual del dispositivo.

El dispositivo diseñado funciona con una alimentación de cinco voltios proporcionados por un regulador de voltaje que tiene una entrada de corriente de 12 voltios procedentes de la batería del autobús. Este se instala tal como se indica en la figura 65.



Figura 65. Instalación del dispositivo.

#### 4.8.1.1 Identificación de botones.

El dispositivo diseñado cuenta con tres botones para su correcto funcionamiento. Cada uno de ellos desempeña una función específica y se los ha llamado: Inicio, Fin y Home, como se observa en la figura 66. El botón inicio permite visualizar una segunda pantalla con la lista de viajes disponibles; el botón fin permite finalizar el funcionamiento al dispositivo y el botón home regresa a la pantalla principal y de esta manera todas las variables definidas regresan a sus valores iniciales.

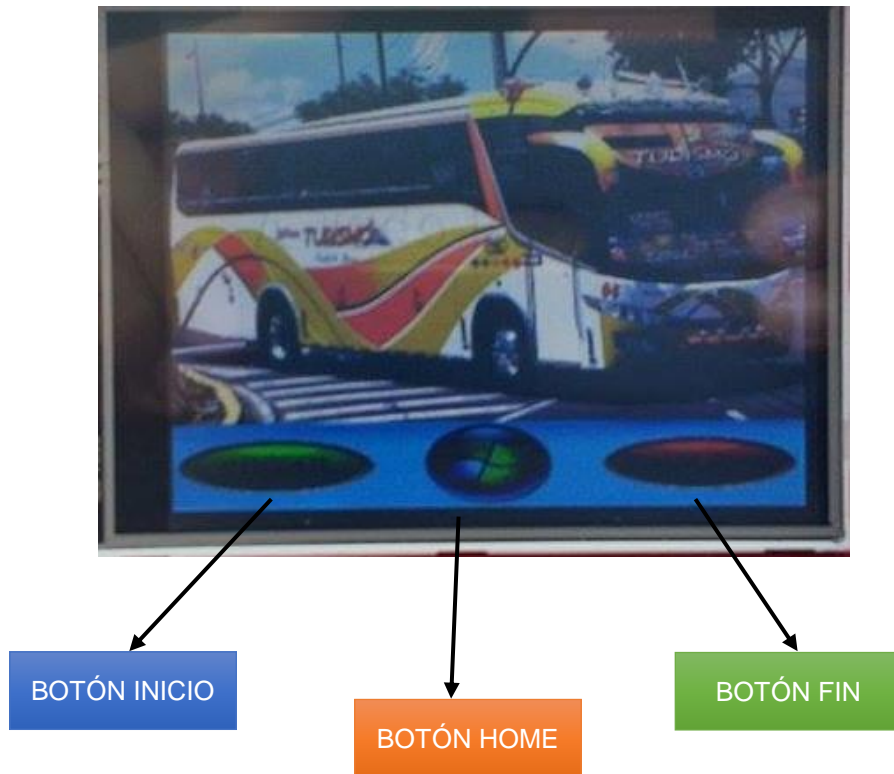


Figura 66. Botones de control.

El botón Inicio se utiliza para mostrar la pantalla de selección de viajes que son cuatro: Ibarra-Quito, Quito-Ibarra, Ibarra- Tulcán, Tulcán- Ibarra; el botón Fin sirve para finalizar el funcionamiento del dispositivo; mientras que el botón Home se utiliza para volver a la pantalla principal.

#### **4.8.1.2 Selección de viajes.**

Para la selección de viajes se dispone de un cuadro rojo junto al nombre del viaje a elegir, como se observa en la figura 67. Para seleccionar el viaje se presiona el cuadro rojo junto al nombre de cada uno de ellos. Se dispone de cuatro viajes que son: Ibarra - Quito, Quito - Ibarra, Ibarra -Tulcán, Tulcán – Ibarra.



Figura 67. Ventana de selección de viajes.

Al presionar el cuadro se muestra una tercera pantalla que indica datos importantes y en tiempo real durante el funcionamiento del dispositivo. Los datos presentados contienen la siguiente información: nombre del viaje, velocidad en km/h, la hora y los minutos de atraso o adelanto al pasar por cada uno de los puntos de control establecidos, tal como se indica en la figura 68.



Figura 68. Ventana de funcionamiento del dispositivo.

#### 4.8.1.3 Funcionamiento en viaje.

Durante el funcionamiento del dispositivo se muestra la información en la pantalla distribuida en cuadros específicos como se indica en la figura 69, indicando la velocidad en km/h, la hora y fecha actual, así como también los minutos de adelanto y atraso a medida que el autobús pasa por los lugares de control establecidos en la ruta a seguir.

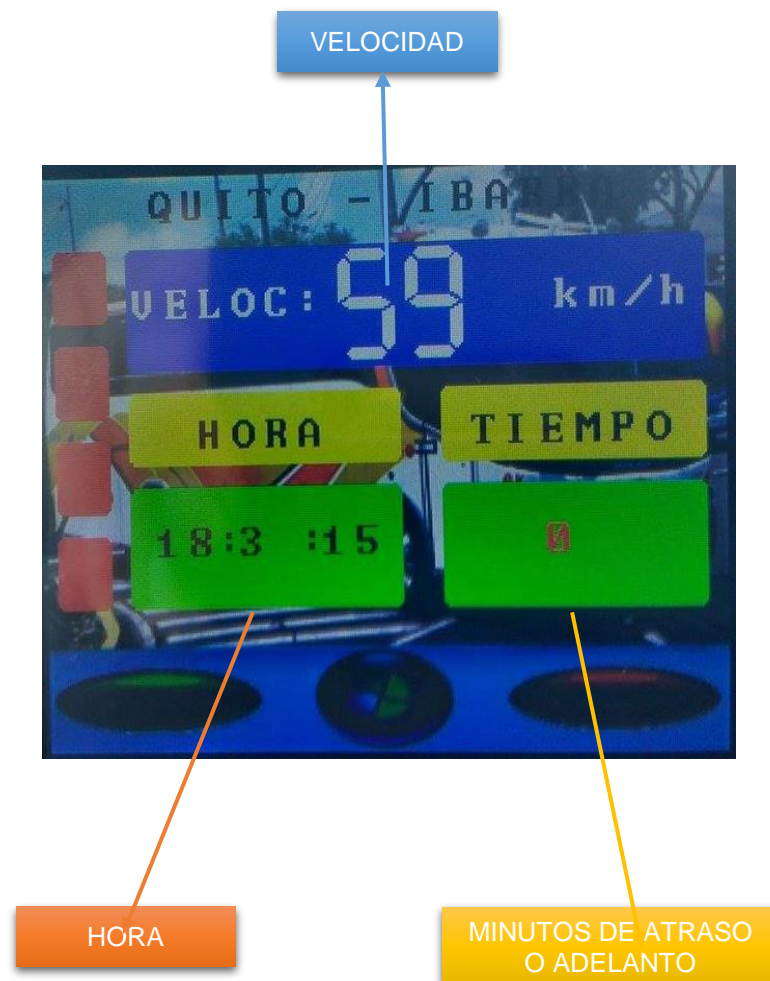


Figura 69. Funcionamiento del dispositivo.

Al momento de que el vehículo pasa por cada punto de control el dispositivo indica en la pantalla el tiempo de adelanto o atraso en minutos. Este tiempo se presenta en cuadros de diferentes colores siendo: rojo para atraso, azul para adelanto y verde para tiempo igual a cero, tal como se observa en la siguiente tabla.

Tabla 12. Presentación de tiempos de control.

| Adelanto | Igual | Atraso |
|----------|-------|--------|
|          |       |        |

Además, a través de su funcionamiento se escucha mediante un parlante un sonido cuando el vehículo sobrepasa los límites de velocidad, de esta manera se produce un sonido agudo cuando la velocidad es mayor a 80km/h, un sonido medio cuando es mayor a 85km/h y un sonido fuerte al sobrepasar los 90km/h. Así también, se almacena datos específicos del viaje que son importantes para el análisis estadístico en el programa de análisis de datos creado. Esta información se almacena en archivos Excel con extensión CSV como se indica en la figura 70.

| A  | B            | C         | D             | E      | F      | G      | H      | I      | J      | K      | L      | M      | N      | O      | P      | Q      |        |
|----|--------------|-----------|---------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 1  | VIAJE        | FECHA     | HORA SALID/P1 | P2     | P3     | P4     | P5     | P6     | P7     | P8     | P9     | P10    | P11    | P12    | P13    | P14    |        |
| 2  | IBARRA-QUIT. | 22/6/2016 | .17:16        | .17:16 | .17:19 | .17:22 | .17:25 | .17:28 | .17:30 | .17:32 | .17:34 | .17:36 | .17:38 | .17:16 | .17:19 | .17:20 | .17:25 |
| 3  | IBARRA-QUIT. | 22/6/2016 | .18:35        | .18:35 | .18:38 | .18:41 | .18:44 | .18:47 | .18:49 | .18:51 | .18:53 | .18:55 | .18:57 | .18:35 | .18:39 | .18:41 | .18:43 |
| 4  | QUITO-IBARRA | 23/6/2016 | .14:18        | .14:18 | .14:20 | .14:22 | .14:24 | .14:26 | .14:28 | .14:31 | .14:34 | .14:37 | .14:40 | .14:18 | .14:21 | .14:22 | .14:24 |
| 5  |              |           |               |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| 6  |              |           |               |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| 7  |              |           |               |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| 8  |              |           |               |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| 9  |              |           |               |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| 10 |              |           |               |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| 11 |              |           |               |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| 12 |              |           |               |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| 13 |              |           |               |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| 14 |              |           |               |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| 15 |              |           |               |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| 16 |              |           |               |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| 17 |              |           |               |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| 18 |              |           |               |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| 19 |              |           |               |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| 20 |              |           |               |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| 21 |              |           |               |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| 22 |              |           |               |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| 23 |              |           |               |        |        |        |        |        |        |        |        |        |        |        |        |        |        |

Figura 70. Archivo viajes.csv.



También se almacenan en dos archivos diferentes la velocidad máxima y velocidad mínima en los archivos Vmax.CSV y Vmin.CSV respectivamente, como se observa en la figura 71. Los datos almacenados son el nombre del viaje, fecha, hora, posición en valores de longitud y latitud, y la velocidad obtenida.

|    | A            | B          | C      | D            | E                | F |    | A            | B          | C      | D            | E                | F |
|----|--------------|------------|--------|--------------|------------------|---|----|--------------|------------|--------|--------------|------------------|---|
| 1  | VIAJE        | FECHA      | HORA   | POSICION     | VELOCIDAD MAXIMA |   | 1  | VIAJE        | FECHA      | HORA   | POSICION     | VELOCIDAD MINIMA |   |
| 2  | IBARRA-QUITO | .22/6/2016 | .17:30 | 0.338945,-78 | 90               |   | 2  | IBARRA-QUITO | .22/6/2016 | .17:16 | 0.361219,-78 | 0                |   |
| 3  | IBARRA-QUITO | .22/6/2016 | .17:30 | 0.338776,-78 | 93               |   | 3  | IBARRA-QUITO | .22/6/2016 | .17:21 | 0.355254,-78 | 0                |   |
| 4  | IBARRA-QUITO | .22/6/2016 | .17:30 | 0.338601,-78 | 95               |   | 4  | IBARRA-QUITO | .22/6/2016 | .17:22 | 0.353620,-78 | 0                |   |
| 5  | IBARRA-QUITO | .22/6/2016 | .17:30 | 0.338421,-78 | 99               |   | 5  | IBARRA-QUITO | .22/6/2016 | .17:24 | 0.349910,-78 | 0                |   |
| 6  | IBARRA-QUITO | .22/6/2016 | .17:30 | 0.338233,-78 | 101              |   | 6  | IBARRA-QUITO | .22/6/2016 | .17:26 | 0.346172,-78 | 0                |   |
| 7  | IBARRA-QUITO | .22/6/2016 | .17:30 | 0.338041,-78 | 103              |   | 7  | IBARRA-QUITO | .22/6/2016 | .17:32 | 0.337763,-78 | 0                |   |
| 8  | IBARRA-QUITO | .22/6/2016 | .17:30 | 0.337845,-78 | 105              |   | 8  | IBARRA-QUITO | .22/6/2016 | .17:40 | 0.326873,-78 | 6                |   |
| 9  | IBARRA-QUITO | .22/6/2016 | .17:30 | 0.337648,-78 | 106              |   | 9  | QUITO-IBARRA | .22/6/2016 | .18:6  | 0.340832,-78 | 10               |   |
| 10 | IBARRA-QUITO | .22/6/2016 | .17:30 | 0.337456,-78 | 103              |   | 10 | IBARRA-QUITO | .22/6/2016 | .18:28 | 0.362167,-78 | 0                |   |
| 11 | IBARRA-QUITO | .22/6/2016 | .17:30 | 0.337270,-78 | 99               |   | 11 | IBARRA-QUITO | .22/6/2016 | .18:30 | 0.359775,-78 | 21               |   |
| 12 | IBARRA-QUITO | .22/6/2016 | .17:30 | 0.337089,-78 | 97               |   | 12 | IBARRA-QUITO | .22/6/2016 | .18:36 | 0.359777,-78 | 7                |   |
| 13 | IBARRA-QUITO | .22/6/2016 | .17:30 | 0.336912,-78 | 94               |   | 13 | IBARRA-QUITO | .22/6/2016 | .18:37 | 0.359776,-78 | 0                |   |
| 14 | IBARRA-QUITO | .22/6/2016 | .17:35 | 0.339292,-78 | 90               |   | 14 | IBARRA-QUITO | .22/6/2016 | .18:38 | 0.361327,-78 | 0                |   |
| 15 | IBARRA-QUITO | .22/6/2016 | .17:35 | 0.339236,-78 | 91               |   | 15 | IBARRA-QUITO | .22/6/2016 | .18:40 | 0.356811,-78 | 0                |   |
| 16 | IBARRA-QUITO | .22/6/2016 | .17:35 | 0.339181,-78 | 94               |   | 16 | IBARRA-QUITO | .22/6/2016 | .18:45 | 0.346002,-78 | 0                |   |
| 17 | IBARRA-QUITO | .22/6/2016 | .17:35 | 0.339123,-78 | 94               |   | 17 | IBARRA-QUITO | .22/6/2016 | .18:47 | 0.340767,-78 | 6                |   |
| 18 | IBARRA-QUITO | .22/6/2016 | .17:35 | 0.339062,-78 | 96               |   | 18 | IBARRA-QUITO | .22/6/2016 | .18:56 | 0.327862,-78 | 8                |   |
| 19 | IBARRA-QUITO | .22/6/2016 | .17:35 | 0.339000,-78 | 97               |   | 19 | QUITO-IBARRA | .23/6/2016 | -.5:0  | 0.327476,-78 | 0                |   |
| 20 | IBARRA-QUITO | .22/6/2016 | .17:35 | 0.338936,-78 | 98               |   | 20 | QUITO-IBARRA | .23/6/2016 | -.5:10 | 0.345748,-78 | 0                |   |
| 21 | IBARRA-QUITO | .22/6/2016 | .17:35 | 0.338871,-78 | 98               |   | 21 | QUITO-IBARRA | .23/6/2016 | -.5:17 | 0.361434,-78 | 0                |   |
| 22 | IBARRA-QUITO | .22/6/2016 | .17:35 | 0.338807,-78 | 97               |   | 22 | QUITO-IBARRA | .23/6/2016 | -.5:19 | 0.362147,-78 | 9                |   |
| 23 | IBARRA-QUITO | .22/6/2016 | .17:35 | 0.338745,-78 | 96               |   | 23 | QUITO-IBARRA | .23/6/2016 | -.5:19 | 0.362147,-78 | 0                |   |

Figura 71. Archivos vmax.csv y vmin.csv.

#### 4.8.2 Manual del Software LabVIEW.

El programa de análisis de datos contiene cuatro ventanas como se indica en la figura 72, que muestra información de atrasos/adelantos, estadística y velocidades, cada una de ellas con tablas para mostrar los datos importados desde la tarjeta de memoria del dispositivo luego de realizar los cálculos correspondientes.



Figura 72. Programa de análisis de datos.

En la segunda ventana se indica el total de viajes, así como también los minutos de atrasos y adelantos que realiza el autobús en los diferentes viajes, además, frente a cada uno de estos se indica el promedio realizado en cada viaje, como se indica en la figura 73.

The screenshot shows a software interface titled 'DATOS MENSUALES'. At the top, there are tabs for 'PORTADA', 'ATRASOS/ADELANTOS', 'ESTADISTICAS', and 'VELOCIDADES'. Below the tabs, there are input fields for 'COOPERATIVA' (set to 'TURISMO') and 'NÚMERO DE VEHÍCULO' (set to '8'). A 'CONECTAR' button with a green indicator light is on the right. The main area contains six summary boxes: 'TOTAL DE VIAJES' (26), 'TOTAL MINUTOS DE ATRASO' (-35), 'PROMEDIO MINUTOS/VIAJE' (-1.35), 'TOTAL MINUTOS DE ADELANTO' (11), and 'PROMEDIO MINUTOS/VIAJE' (0.42). A 'CALCULAR' button is at the bottom center.

Figura 73. Ventana de datos generales.

En la tercera ventana se indica la información de cada viaje, luego de seleccionar cada uno de ellos en el filtro establecido al principio de la ventana, como se puede observar en la figura 74. Al final de la tabla se observa un promedio de atrasos o adelantos en cada uno de los lugares de control establecidos en cada viaje.

The screenshot shows a software interface titled 'LUGARES DE CONTROL'. It has tabs for 'PORTADA', 'ATRASOS/ADELANTOS', 'ESTADISTICAS', and 'VELOCIDADES'. There are filters for 'VIAJES' (set to 'IBARRA-QUITO') and 'NÚMERO DE VIAJE' (range 0 to 1000). The main area is a table with columns for '#', 'VIAJE', 'FECHA', 'HH SALIDA', and 12 control points: IBARRA, LA FLORIDA, SAN ANTONIO, ATUNTAQUI, OTAVALO, CAJAS, TABACUNDO, GUALLABAMBA, CARCELEN, and TERMINAL. Below the table are summary rows for 'ATRASOS' and 'ADELANTOS'. A 'CALCULAR' button is at the bottom center.

| #                | VIAJE        | FECHA     | HH SALIDA | IBARRA | LA FLORIDA | SAN ANTONIO | ATUNTAQUI | OTAVALO | CAJAS | TABACUNDO | GUALLABAMBA | CARCELEN | TERMINAL |
|------------------|--------------|-----------|-----------|--------|------------|-------------|-----------|---------|-------|-----------|-------------|----------|----------|
| 1                | IBARRA-QUITO | 29/6/2016 | .17:01    | 0      | 0          | 0           | 0         | 0       | 0     | 0         | 0           | 0        | 0        |
| 3                | IBARRA-QUITO | 30/6/2016 | .19:45    | 0      | 0          | 0           | 0         | 0       | 0     | 0         | 0           | 0        | 0        |
| 4                | IBARRA-QUITO | 30/6/2016 | .19:46    | 0      | 0          | 0           | 0         | 0       | 0     | 0         | 0           | 0        | 0        |
| 5                | IBARRA-QUITO | 30/6/2016 | .19:46    | 0      | 0          | 1           | 0         | 0       | 0     | 0         | 1           | 2        | 2        |
| 8                | IBARRA-QUITO | 30/6/2016 | .22:29    | 0      | 0          | 0           | 0         | 0       | 0     | 0         | 0           | 0        | 0        |
| 9                | IBARRA-QUITO | 30/6/2016 | .22:30    | 0      | 0          | 0           | 0         | 0       | 0     | 0         | 0           | 0        | 0        |
| 12               | IBARRA-QUITO | 30/6/2016 | .19:00    | 0      | 0          | -1          | -1        | -3      | -3    | -3        | -3          | -1       | -1       |
| 15               | IBARRA-QUITO | 30/6/2016 | .19:00    | 0      | 0          | 0           | 0         | 0       | 0     | 0         | 0           | 0        | 0        |
| 16               | IBARRA-QUITO | 30/6/2016 | .19:00    | 0      | 0          | 0           | 0         | 0       | 0     | 0         | 0           | 0        | 0        |
| 17               | IBARRA-QUITO | 30/6/2016 | .09:40    | 0      | 0          | 0           | 0         | 0       | 0     | 0         | 0           | 0        | 0        |
| 18               | IBARRA-QUITO | 30/6/2016 | .09:43    | 0      | 0          | 0           | 0         | 0       | 0     | 0         | 0           | 0        | 0        |
| 19               | IBARRA-QUITO | 30/6/2016 | .22:18    | 0      | 0          | 0           | 0         | 0       | 0     | 0         | 0           | 0        | 0        |
| 20               | IBARRA-QUITO | 30/6/2016 | .22:34    | 0      | 0          | 0           | 0         | 0       | 0     | 0         | 0           | 0        | 0        |
| 21               | IBARRA-QUITO | 30/6/2016 | .22:38    | 0      | 0          | 0           | 0         | 0       | 0     | 0         | 0           | 0        | 0        |
|                  |              |           |           | 0      | 0          | -1          | -1        | -3      | -3    | -3        | -3          | -1       | -1       |
| <b>ATRASOS</b>   |              |           |           | 0      | 0          | -1          | -1        | -3      | -3    | -3        | -3          | -1       | -1       |
| <b>ADELANTOS</b> |              |           |           | 0      | 0          | 1           | 0         | 0       | 0     | 0         | 1           | 2        | 2        |

Figura 74. Ventana de adelantos y atrasos.

Además, es posible seleccionar viajes en específico para conocer sus adelantos o atrasos en cada punto de control, que al final de cada columna indica un promedio de adelanto o atraso en cada uno de ellos. En la cuarta y última ventana se presenta la información de las velocidades máximas en la primera tabla, mientras que en la segunda tabla las velocidades mínimas, como se observa en la figura 75.

| VELOCIDAD MÁXIMA |           |        |                     |                  |
|------------------|-----------|--------|---------------------|------------------|
| VIAJE            | FECHA     | HORA   | POSICION            | VELOCIDAD MAXIMA |
| IBARRA-QUITO     | 30/6/2016 | .11:4  | 0.338187,-78.186317 | 101              |
| IBARRA-QUITO     | 30/6/2016 | .11:4  | 0.337533,-78.188789 | 101              |
| IBARRA-QUITO     | 30/6/2016 | .11:5  | 0.334930,-78.198593 | 102              |
| IBARRA-QUITO     | 30/6/2016 | .13:16 | 0.338671,-78.152526 | 102              |
| IBARRA-QUITO     | 30/6/2016 | .13:21 | 0.337737,-78.188056 | 103              |
| QUITO-IBARRA     | 30/6/2016 | .13:30 | 0.338130,-78.186347 | 104              |
| QUITO-IBARRA     | 30/6/2016 | .13:30 | 0.338791,-78.183876 | 101              |

| VELOCIDAD MÍNIMA |           |        |                     |                  |
|------------------|-----------|--------|---------------------|------------------|
| VIAJE            | FECHA     | HORA   | POSICION            | VELOCIDAD MINIMA |
| IBARRA-QUITO     | 30/6/2016 | .10:45 | 0.361333,-78.112007 | 0                |
| IBARRA-QUITO     | 30/6/2016 | .10:46 | 0.361358,-78.111999 | 0                |
| IBARRA-QUITO     | 30/6/2016 | .10:48 | 0.361274,-78.119514 | 25               |
| IBARRA-QUITO     | 30/6/2016 | .10:48 | 0.359464,-78.120979 | 21               |
| IBARRA-QUITO     | 30/6/2016 | .10:49 | 0.357532,-78.121971 | 0                |
| IBARRA-QUITO     | 30/6/2016 | .10:53 | 0.349939,-78.130996 | 12               |
| IBARRA-QUITO     | 30/6/2016 | .10:55 | 0.346212,-78.136222 | 0                |
| IBARRA-QUITO     | 30/6/2016 | .10:58 | 0.340861,-78.145187 | 0                |
| IBARRA-QUITO     | 30/6/2016 | .11:5  | 0.336456,-78.192802 | 21               |
| IBARRA-QUITO     | 30/6/2016 | .11:8  | 0.327568,-78.215583 | 0                |
| IBARRA-QUITO     | 30/6/2016 | .11:9  | 0.327644,-78.215629 | 0                |
| IBARRA-QUITO     | 30/6/2016 | .11:9  | 0.327695,-78.215530 | 7                |
| IBARRA-QUITO     | 30/6/2016 | .11:10 | 0.327885,-78.215072 | 0                |

Figura 75. Ventana de velocidades.

La velocidad máxima o mínima se registra en cada viaje realizado y es almacenada en el respectivo archivo en la tarjeta de memoria del dispositivo. La información almacenada es el nombre del viaje realizado, la fecha, hora, posición y la velocidad registrada en ese instante.

En la misma ventana se dispone de un botón de Graficar que se utiliza para ejecutar el programa PowerBi, en el cual se grafican los lugares en los que el autobús sobrepasa los 90km/h o se mantiene durante 50 segundos en una velocidad menor a 30km/h; estos puntos son graficados como se observa en la figura 76.

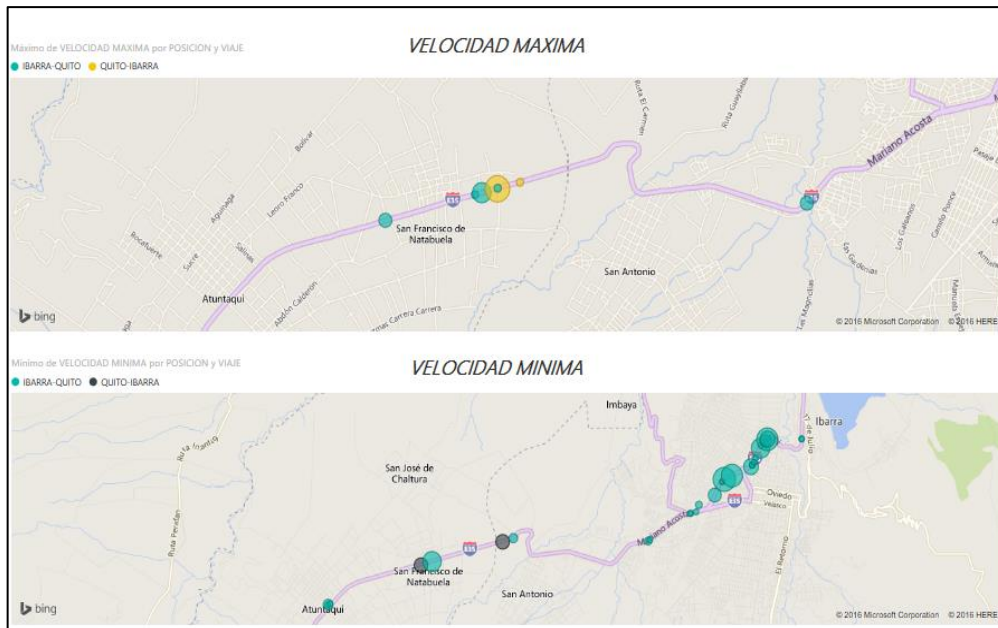


Figura 76. Ventana de representación gráfica.

## **CAPÍTULO V.**

### **5. CONCLUSIONES Y RECOMENDACIONES.**

#### **5.1 Conclusiones.**

- El dispositivo desarrollado a través de la plataforma de hardware libre Arduino permite conocer los minutos de adelanto y atraso que tiene un autobús durante su viaje, a través de los diferentes lugares de control establecidos en la programación.
- El módulo GPS permite identificar los lugares de control de tiempo de viaje mediante las áreas determinadas a través de los valores de longitud y latitud, para así determinar los adelantos y atrasos.
- El programa de análisis de datos desarrollado en el software LabVIEW brinda información estadística del total de minutos de atraso y adelanto, como también la información de cada uno de los viajes realizados durante un tiempo determinado.
- El dispositivo almacena datos del viaje realizado como: nombre del viaje, fecha y hora de inicio, hora real y de control, tiempo de adelanto o atraso, velocidades máximas y mínimas, longitud y latitud, que permitirán obtener información estadística a través del programa desarrollado en el software LabVIEW.
- La alimentación del dispositivo se realiza a través de una conexión de 12 voltios de la batería a un regulador que proporciona 5 voltios de corriente continua necesarios para su funcionamiento.

- La característica adicional que brinda el dispositivo desarrollado es el reporte en tiempo real de los minutos de adelanto o atraso en cada uno de los lugares de control establecidos en la ruta a seguir por el autobús, información que permitirá al conductor un mejor control en el tiempo de viaje.

## **5.2 Recomendaciones.**

- Se recomienda el uso de este dispositivo para obtener información que permita a las autoridades responsables de las diferentes cooperativas de autobús conocer los minutos de adelanto y atraso de las unidades en los diferentes viajes realizados y así puedan establecer la sanción correspondiente.
- Es recomendable realizar pruebas de funcionamiento en lugares con poca señal de GPS, debido a que puede afectar a funcionamiento del dispositivo, proporcionando datos inválidos para su análisis correspondiente.
- Se recomienda la utilización del dispositivo como un Tacógrafo debido a que permite registrar diversos sucesos originados durante el desplazamiento del autobús como: velocidad (mínima y máxima), tiempo de adelanto o atraso, que pueden ser almacenados para su posterior análisis.
- Se recomienda el uso de la plataforma de hardware libre Arduino puesto que proporciona herramientas y librerías compatibles con diferentes elementos electrónicos, facilitando de esta manera su programación.
- El software LabVIEW permite la importación de datos desde tarjetas de almacenamiento, como también la visualización de información en tiempo

real, por lo que se recomienda su utilización para el desarrollo de diferentes proyectos.

- Estudiar la posibilidad de incorporar al dispositivo una alarma que alerte al conductor cuando intente rebasar en lugares prohibidos, y de la misma manera controlar el tiempo límite para permanecer en una parada.
- Incorporar una forma de control para el conductor responsable del autobús en cada viaje realizado, ya sea a través del número de cédula, nombre del conductor, licencia de conducir o huella digital.
- Se recomienda su utilización como herramienta de reporte de velocidad máxima en tiempo real para control de los buses interprovinciales en carretera por parte de la Policía Nacional, y a su vez con su respectivo almacenamiento procesar un reporte acumulado para la respectiva sanción en la matriculación del vehículo.

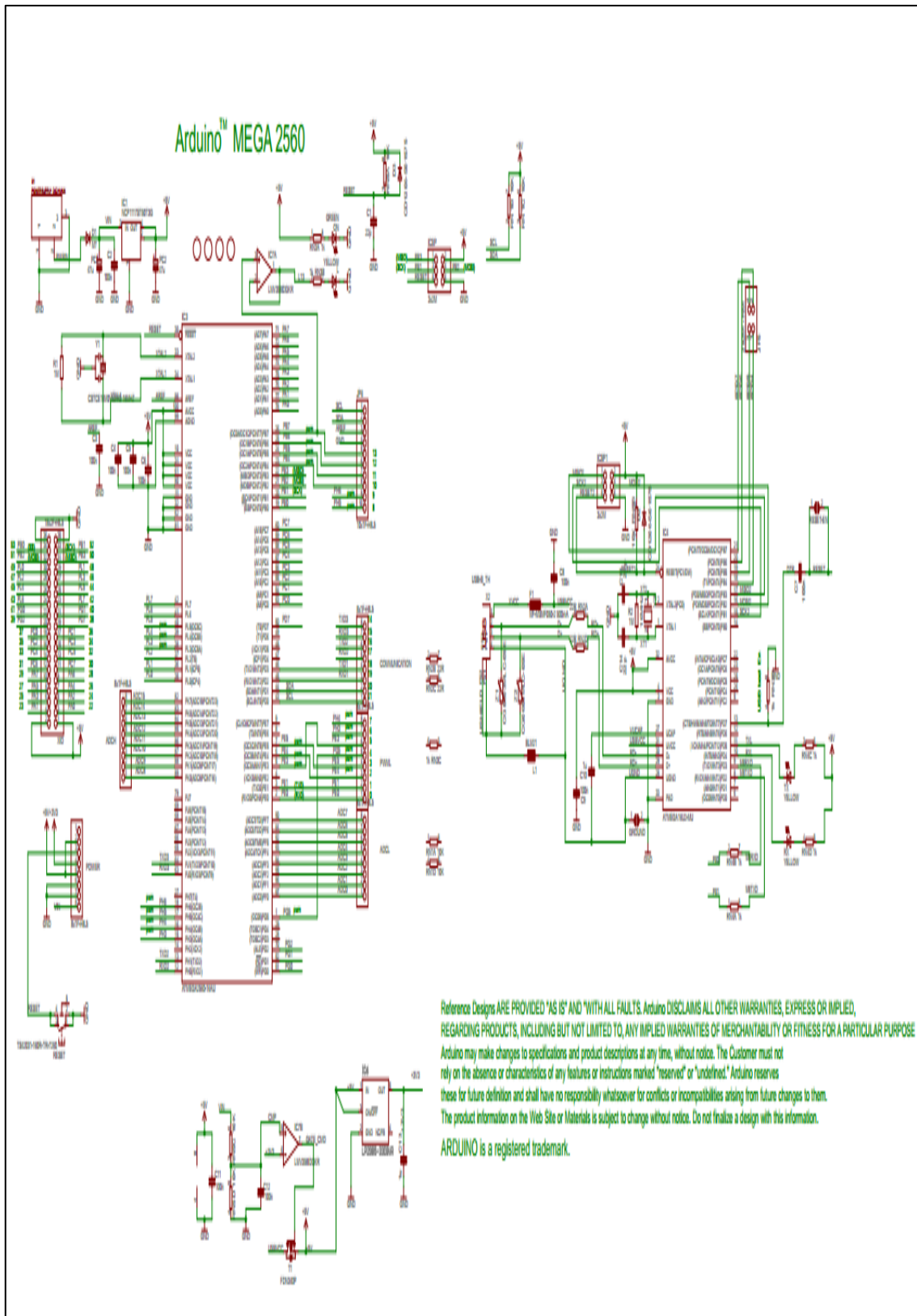
## BIBLIOGRAFÍA

- Florez Lopez, M. V., & Isaza Posse, G. M. (2015). *Descubre tu GPS*. Grupo Editorial Colombia.
- Arduino. (2016). *Arduino*. Obtenido de Arduino: <https://www.arduino.cc/>
- Arnalich, S., & Urruela, J. (2012). *GPS, Google Earth y Cooperación*. Arnalich.
- Artero, Ó. T. (2013). *Arduino: Curso práctico de formación*. RC Libros.
- Banzi, M. (2012). *Introducción a Arduino*. Grupo Anaya Comercial.
- Correia, P. (2010). *Guía práctica del GPS*. Marcombo.
- Fesser, G. (2013). *Ruedas y el secreto del GPS*. Grupo Editorial España.
- GOILAV, N., & LOI, G. (2016). *Arduino: Aprender a desarrollar para crear objetos inteligentes*.
- Lajara Vizcaíno, J. R., & Pelegrí Sebastián, J. (2012). *LabVIEW: Entorno gráfico de programación*. Marcombo.
- Lázaro, A. M. (2010). *LabVIEW 6i: programación gráfica para el control de instrumentación*. Paraninfo.
- Lequerica, J. R. (2013). *Arduino práctico*. Anaya Multimedia-Anaya Interactiva.
- Letham, L. (2009). *GPS facil. Uso del sistema de posicionamiento global*. Paidotribo.
- Morales Costa, L., & Flores Fernandez, L. E. (2009). *La problemática del transporte interprovincial de pasajeros*. SIC.
- Pedrera, A. C. (2014). *Arduino para principiantes*. IT Campus Academy.
- Zanini, V. (2013). *Macros en excel 2013*. Buenos Aires: USERSHOP.



# **ANEXOS.**

# ANEXO 1. DATASHEET ARDUINO MEGA 2560.



## ANEXO 2. 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<br>NEO-6MVT | 2.7  | 3.0     | 3.6     | V     |                             |
| 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,<br>VCC = 0V |
| Input pin voltage range                  | V <sub>in</sub>    | All                 | 0  |         | VCC     | V     |                             |
| Digital IO Pin Low level input voltage   | V <sub>l</sub>     | All                 | 0  |         | 0.2*VCC | V     |                             |
| Digital IO Pin High level input voltage  | V <sub>h</sub>     | All                 | 0.7*VCC  |         | VCC     | V     |                             |
| Digital IO Pin Low level output voltage  | V <sub>ol</sub>    | All                 |  |         | 0.4     | V     | I <sub>oh</sub> =4mA        |
| Digital IO Pin High level output voltage | V <sub>oh</sub>    | All                 | VCC -0.4   |         |         | V     | I <sub>oh</sub> =4mA        |
| USB_DM, USB_DP                           | V <sub>inU</sub>   | All                 | Compatible with USB with 22 Ohms series resistance |         |         |       |                             |
| VCC_RF voltage                           | VCC_RF             | All                 |  | VCC-0.1 |         | V     |                             |
| VCC_RF output current                    | I <sub>CC_RF</sub> | All                 |  |         | 50      | mA    |                             |
| Antenna gain                             | G <sub>ant</sub>   | All                 |  |         | 50      | dB    |                             |
| Receiver Chain Noise Figure              | NF <sub>tot</sub>  | All                 |  | 3.0     |         | dB    |                             |
| Operating temperature                    | T <sub>opr</sub>   | 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**    | I <sub>ccp</sub>                                     | All                   |     |                                    | 67  | mA    | VCC = 3.6 V <sup>†</sup> /<br>1.95 V <sup>†</sup> |
|                          | I <sub>cc</sub> Acquisition                          | All                   |     | 47 <sup>†</sup>                    |     | mA    |   |
|                          | I <sub>cc</sub> Tracking<br>(Max Performance mode)   | NEO-6MVT<br>NEO-6MPPV |     | 40 <sup>†</sup><br>39 <sup>†</sup> |     | mA    |   |
| Average supply current** | I <sub>cc</sub> Tracking<br>(Fix mode)               | NEO-6MVT<br>NEO-6MPPV |     | 38 <sup>†</sup><br>37 <sup>†</sup> |     | mA    | VCC = 3.0 V <sup>†</sup> /<br>1.8 V <sup>†</sup>  |
|                          | I <sub>cc</sub> Tracking<br>(Power save mode / 1 Hz) | NEO-6MVT<br>NEO-6M    |     | 12 <sup>†</sup><br>11 <sup>†</sup> |     | 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.

| No | Module | Name   | I/O | Description           |
|----|--------|--------|-----|-----------------------|
| 22 | AI     | V_BCKP | I   | Backup voltage supply |
| 23 | AI     | VCC    | I   | Supply voltage        |
| 24 | AI     | 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

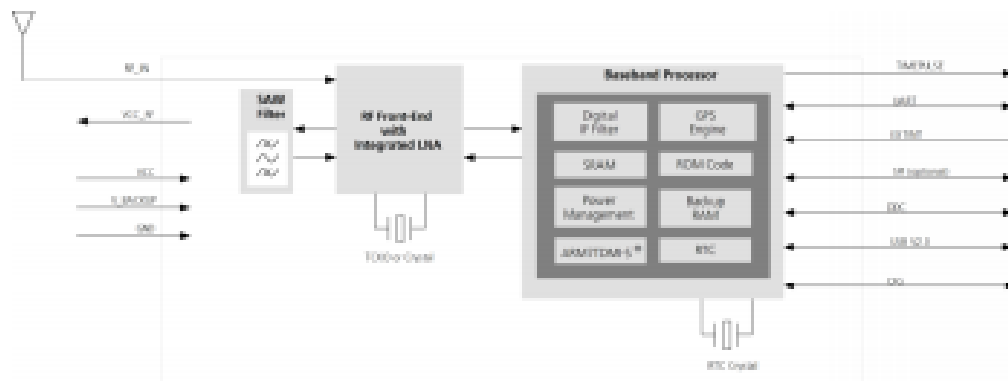


Figure 1: Block diagram (For available options refer to the product features table in section 1.2).

**ANEXO 3.**  
**TABLA DE COSTOS.**

| <b>ELEMENTO</b>                  | <b>COSTO(\$)</b> |
|----------------------------------|------------------|
| Arduino Mega 2560                | 90               |
| Módulo GPS                       | 120              |
| Pantalla táctil 3.2"             | 90               |
| Shield para pantalla táctil      | 30               |
| Regulador de voltaje de 12V a 5V | 20               |
| Cable de alimentación            | 20               |
| Caja                             | 30               |
| <b>TOTAL</b>                     | <b>400</b>       |

## ANEXO 4.

### CÓDIGO DE PROGRAMACIÓN.

```
#include <SPI.h> //comunicacion pantalla
#include <SD.h> //tarjeta de memoria
#include <TinyGPS++.h> //GPS
#include <UTouch.h> //tactil de la pantalla
#include <UTFT.h> //pantalla
#include <UTFT_tinyFAT.h> //utilizar el módulo de la memoria de la pantalla
#include <tinyFAT.h> //conexion con la memoria
#include <TimerOne.h> //timer
```

```
long tiempo=0;
int guarda=0;
String título; //texto
int dia=0;
int mes=0;
int a_o=0;
int hora=0;
int minuto=0;
int hora1=0;
int minuto1=0;
int hora2=0;
int minuto2=0;
int hora3=0;
int minuto3=0;
int hora4=0;
int minuto4=0;
int hora5=0;
int minuto5=0;
int hora6=0;
int minuto6=0;
int hora7=0;
int minuto7=0;
int hora8=0;
int minuto8=0;
int hora9=0;
int minuto9=0;
int hora10=0;
int minuto10=0;
int hora11=0;
int minuto11=0;
int hora12=0;
int minuto12=0;
int hora13=0;
int minuto13=0;
int hora14=0;
int minuto14=0;
int hora15=0;
int minuto15=0;
int hora16=0;
int minuto16=0;
int hora17=0;
int minuto17=0;
int hora18=0;
int minuto18=0;
int hora19=0;
int minuto19=0;
int hora20=0;
int minuto20=0;
int posicion1=0;
int posicion2=0;
int posicion3=0;
int posicion4=0;
int posicion5=0;
int posicion6=0;
int posicion7=0;
int posicion8=0;
int posicion9=0;
int posicion10=0;
int almacenamiento=0;
int minima=0;
int maxima=0;
int diferencia1=0;
int diferencia2=0;
int diferencia3=0;
int diferencia4=0;
int diferencia5=0;
int diferencia6=0;
int diferencia7=0;
int diferencia8=0;
int diferencia9=0;
int diferencia10=0;
int tiempo2=0;
```

```

int altavoz=13;
int frecuencia1=600;
int frecuencia2=698;
int frecuencia3=987;

TinyGPSPPlus Bus;
//Instanciamos la TFT
UTFT myGLCD(ITDB32S,38,39,40,41);//definir el tipo de pantalla
UTFT_tinyFAT myFiles(&myGLCD); //definir a la tarjeta de memoria
char* files320[]={"INICIO.RAW"};
char* files[1];

// 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); //declaracion de pines tactil
File datos;//dar nombre a la memoria sd
#define SD_CS 53 //pin para la sd

void setup() {

Serial1.begin(9600); //Gps con arduino se comunica con Arduino a 9600 en el puerto Tx Rx (1)
Serial.begin(38400); // Arduino con la Pc en el puerto Tx Rx (0)
Timer1.initialize(1000000);//cada 1 segundo pero el timer se define en microsegundos
//Configuracion de la TFT
myGLCD.InitLCD();
myGLCD.fillScr(VGA_BLACK);//color de fondo de pantalla
myGLCD.setBackColor(VGA_TRANSPARENT);
file.initFAT();//comunicacion con la sd
//Configuracion del Touch
ts.InitTouch();//inicializar el touch
ts.setPrecision(PREC_MEDIUM);//defines la precision del tactil
//Configuracion de la SD
pinMode(53,OUTPUT);//pin de salida de la memoria
if (!SD.begin(SD_CS))
{
// Serial1.print("ERROR SD");
return;
}
myGLCD.fillScr(VGA_BLACK);
myFiles.loadBitmap(0, 0, 320, 240, "INICIO.RAW");//cargar la imagen
}

void loop() {
while (Serial1.available()>0) // verificar si hay comunicacion entre el Gps y el Ard
{
if (Bus.encode (Serial1.read())) //verificar mientras se encuentre enviando datos por el serial 1
{
}
}
if (millis ()>3000&&Bus.charsProcessed()<10) //comprobar si en 3 segundos me debe enviar por lo menos 10 caracteres
{
Serial.println (F("Sin comunicacon"));
while (true); //ciclo infinito
}
long x, y; //Touch
if(ts.dataAvailable())
{
ts.read();
x = ts.getX();
y = ts.getY();

if((x>=310) && (x<=360))//para boton inicio
{
if((y>=160) && (y<=220))
{
Timer1.stop();
myGLCD.fillScr(VGA_BLUE);
myFiles.loadBitmap(0, 0, 320, 240, "INICIO.RAW");
myGLCD.setColor(VGA_YELLOW);
myGLCD.fillRect(40, 30, 290, 60);
myGLCD.fillRect(40, 70, 290, 100);
myGLCD.fillRect(40, 110, 290, 140);
myGLCD.fillRect(40, 150, 290, 180);
myGLCD.setColor(VGA_RED);
myGLCD.fillRect(5, 30, 30, 60);
myGLCD.fillRect(5, 70, 30, 100);
myGLCD.fillRect(5, 110, 30, 140);
myGLCD.fillRect(5, 150, 30, 180);
myGLCD.setColor(VGA_BLACK);
myGLCD.setFont(BigFont);
myGLCD.print("VIAJES", 40, 4);
myGLCD.print("IBARRA - QUITO", CENTER, 40);
}
}
}
}

```

```

myGLCD.print("QUITO - IBARRA", CENTER, 80);
myGLCD.print("IBARRA - TULCAN", CENTER, 120);
myGLCD.print("TULCAN - IBARRA", CENTER, 160);
Timer1.attachInterrupt(mostrarlInfo);
}
}
if((x>=60) && (x<=85))//activar el void ibarra-quito
{
if((y>=205) && (y<=225))
{
Timer1.stop();
myGLCD.fillScr(VGA_WHITE);
myFiles.loadBitmap(0, 0, 320, 240, "INICIO.RAW");
myGLCD.setColor(VGA_WHITE);
myGLCD.fillRect(0, 0, 319, 196);
myGLCD.setColor(VGA_BLACK);
myGLCD.setFont(BigFont);
myGLCD.print("IBARRA - QUITO", CENTER, 4);
myGLCD.setColor(VGA_BLUE);
myGLCD.fillRect(40, 30, 310, 80);
myGLCD.setColor(VGA_YELLOW);
myGLCD.fillRect(40, 90, 170, 120);
myGLCD.setColor(VGA_YELLOW);
myGLCD.fillRect(190, 90, 310, 120);
myGLCD.setColor(VGA_LIME);
myGLCD.fillRect(40, 130, 170, 180);
myGLCD.setColor(VGA_LIME);
myGLCD.fillRect(190, 130, 310, 180);
myGLCD.setColor(VGA_BLACK);
myGLCD.print("HORA", 70, 100);
myGLCD.print("TIEMPO", 200, 100);
myGLCD.setColor(VGA_RED);
myGLCD.fillRect(5, 30, 30, 60);
myGLCD.fillRect(5, 70, 30, 100);
myGLCD.fillRect(5, 110, 30, 140);
myGLCD.fillRect(5, 150, 30, 180);
Timer1.attachInterrupt(ibarraquito);
}
}
if((x>=130) && (x<=155)) //quito-ibarra
{
if((y>=205) && (y<=225))
{
Timer1.stop();
myGLCD.fillScr(VGA_WHITE);
myFiles.loadBitmap(0, 0, 320, 240, "INICIO.RAW");
myGLCD.setColor(VGA_WHITE);
myGLCD.fillRect(0, 0, 319, 196);
myGLCD.setColor(VGA_BLACK);
myGLCD.setFont(BigFont);
myGLCD.print("QUITO - IBARRA", CENTER, 4);
myGLCD.setColor(VGA_BLUE);
myGLCD.fillRect(40, 30, 310, 80);
myGLCD.setColor(VGA_YELLOW);
myGLCD.fillRect(40, 90, 170, 120);
myGLCD.setColor(VGA_YELLOW);
myGLCD.fillRect(190, 90, 310, 120);
myGLCD.setColor(VGA_LIME);
myGLCD.fillRect(40, 130, 170, 180);
myGLCD.setColor(VGA_LIME);
myGLCD.fillRect(190, 130, 310, 180);
myGLCD.setColor(VGA_BLACK);
myGLCD.print("HORA", 70, 100);
myGLCD.print("TIEMPO", 200, 100);
myGLCD.setColor(VGA_RED);
myGLCD.fillRect(5, 30, 30, 60);
myGLCD.fillRect(5, 70, 30, 100);
myGLCD.fillRect(5, 110, 30, 140);
myGLCD.fillRect(5, 150, 30, 180);
Timer1.attachInterrupt(quitoibarra);
}
}
if((x>=175) && (x<=200))//ibarra-tulcan
{
if((y>=205) && (y<=225))
{
Timer1.stop();
myGLCD.fillScr(VGA_WHITE);
myFiles.loadBitmap(0, 0, 320, 240, "INICIO.RAW");
myGLCD.setColor(VGA_WHITE);
myGLCD.fillRect(0, 0, 319, 196);
myGLCD.setColor(VGA_BLACK);
myGLCD.setFont(BigFont);
myGLCD.print("IBARRA - TULCAN", CENTER, 4);
myGLCD.setColor(VGA_BLUE);
myGLCD.fillRect(40, 30, 310, 80);
}
}

```



```

myGLCD.setColor(VGA_YELLOW);
myGLCD.fillRect(40, 90, 170, 120);
myGLCD.setColor(VGA_YELLOW);
myGLCD.fillRect(190, 90, 310, 120);
myGLCD.setColor(VGA_LIME);
myGLCD.fillRect(40, 130, 170, 180);
myGLCD.setColor(VGA_LIME);
myGLCD.fillRect(190, 130, 310, 180);
myGLCD.setColor(VGA_BLACK);
myGLCD.print("HORA", 70, 100);
myGLCD.print("TIEMPO", 200, 100);
myGLCD.setColor(VGA_RED);
myGLCD.fillRect(5, 30, 30, 60);
myGLCD.fillRect(5, 70, 30, 100);
myGLCD.fillRect(5, 110, 30, 140);
myGLCD.fillRect(5, 150, 30, 180);
Timer1.attachInterrupt(ibarratulcan);
}
}

if((x>=245) && (x<=270))//tulcan-ibarra
{
  if((y>=205) && (y<=225))
  {
    Timer1.stop();
    myGLCD.fillRect(VGA_WHITE);
    myFiles.loadBitmap(0, 0, 320, 240, "INICIO.RAW");
    myGLCD.setColor(VGA_WHITE);
    myGLCD.fillRect(0, 0, 319, 196);
    myGLCD.setColor(VGA_BLACK);
    myGLCD.setFont(BigFont);
    myGLCD.print("TULCAN - IBARRA", CENTER, 4);
    myGLCD.setColor(VGA_BLUE);
    myGLCD.fillRect(40, 30, 310, 80);
    myGLCD.setColor(VGA_YELLOW);
    myGLCD.fillRect(40, 90, 170, 120);
    myGLCD.setColor(VGA_YELLOW);
    myGLCD.fillRect(190, 90, 310, 120);
    myGLCD.setColor(VGA_LIME);
    myGLCD.fillRect(40, 130, 170, 180);
    myGLCD.setColor(VGA_LIME);
    myGLCD.fillRect(190, 130, 310, 180);
    myGLCD.setColor(VGA_BLACK);
    myGLCD.print("HORA", 70, 100);
    myGLCD.print("TIEMPO", 200, 100);
    myGLCD.setColor(VGA_RED);
    myGLCD.fillRect(5, 30, 30, 60);
    myGLCD.fillRect(5, 70, 30, 100);
    myGLCD.fillRect(5, 110, 30, 140);
    myGLCD.fillRect(5, 150, 30, 180);
    Timer1.attachInterrupt(tulcanibarra);
  }
}

if((x>=310) && (x<=360))//boton home
{
  if((y>=100) && (y<=125))
  {
    Timer1.stop();
    myGLCD.fillRect(VGA_BLACK);
    myFiles.loadBitmap(0, 0, 320, 240, "INICIO.RAW");
    myGLCD.setBackgroundColor(VGA_TRANSPARENT);
  }
}

tiempo=0;
guarda=0;
dia=0;
mes=0;
a_o=0;
hora=0;
minuto=0;
hora1=0;
minuto1=0;
hora2=0;
minuto2=0;
hora3=0;
minuto3=0;
hora4=0;
minuto4=0;
hora5=0;
minuto5=0;
hora6=0;
minuto6=0;
hora7=0;
minuto7=0;
hora8=0;
minuto8=0;
hora9=0;
minuto9=0;

```

```

hora10=0;
minuto10=0;
hora11=0;
minuto11=0;
hora12=0;
minuto12=0;
hora13=0;
minuto13=0;
hora14=0;
minuto14=0;
hora15=0;
minuto15=0;
hora16=0;
minuto16=0;
hora17=0;
minuto17=0;
hora18=0;
minuto18=0;
hora19=0;
minuto19=0;
hora20=0;
minuto20=0;
posicion1=0;
posicion2=0;
posicion3=0;
posicion4=0;
posicion5=0;
posicion6=0;
posicion7=0;
posicion8=0;
posicion9=0;
posicion10=0;
almacenamiento=0;
minima=0;
maxima=0;
diferencia1=0;
diferencia2=0;
diferencia3=0;
diferencia4=0;
diferencia5=0;
diferencia6=0;
diferencia7=0;
diferencia8=0;
diferencia9=0;
diferencia10=0;
tiempo2=0;
Timer1.attachInterrupt(informacionhora);
    }
    }
    if((x>=310) && (x<=360))//boton fin
    {
        if((y>=5) && (y<=80))
        {
            Timer1.stop();
            almacenar(); //codigo almacenar
            noTone(altavoz);
        }
    }
}

void mostrarInfo()
{
    //Datos de latitud y longitud
    if (Bus.location.isValid())
    {
        //Serial.print("Lat: ");
        Serial.print(Bus.location.lat(),4 );
        Serial.print(" ");
        Serial.print(Bus.location.lng(),4 );
        Serial.print(" ");
    }
    if (Bus.altitude.isValid()) // Datos de altitud en metros
    {
        Serial.print(Bus.altitude.meters());
        Serial.print(" ");
    }
    if (Bus.date.isValid()) // Datos de fecha día, mes, año
    {
        Serial.print(Bus.date.day());
        Serial.print("/");
        Serial.print(Bus.date.month());
        Serial.print("/");
        Serial.print(Bus.date.year());
        Serial.print(" ");
    }
}

```

```

    if (Bus.time.isValid()) // Datos de tiempo hora, minuto, segundo
    {
        int h= Bus.time.hour();
        h= h-5;
        if (h<0)
            h= h+24;
        Serial.print(h);
        Serial.print(":");
        Serial.print(Bus.time.minute());
        Serial.print(":");
        Serial.print(Bus.time.second());
        Serial.print(" ");
    }
    if (Bus.speed.isValid()) // Datos de velocidad en km/h
    {
        Serial.print(Bus.speed.kmph());
        Serial.println ( );
    }
}
else
{
    Serial.print(F("Datos invalidos")); // De lo contrario anuncie datos invalidos
}
    int h= Bus.time.hour();
    h= h-5;//restamos cinco horas para la obtener la hora del ecuador
    if (h<0)
    {
        h= h+24;
    }
    myGLCD.setColor(VGA_WHITE);
    myGLCD.fillRoundRect(160, 0, 290, 25);
    myGLCD.setColor(VGA_BLACK);
    myGLCD.drawRoundRect(160, 0, 290, 25);
    myGLCD.setFont(BigFont);
    myGLCD.setColor(VGA_BLACK);
    myGLCD.printNumI(h, 170, 5);
    myGLCD.print(":", 200, 5);
    myGLCD.printNumI(Bus.time.minute(), 210, 5);
    myGLCD.print(":", 240, 5);
    int seg=Bus.time.second();
    myGLCD.printNumI(seg, 250, 5);
}

void informacionhora()
{
    myGLCD.setColor(VGA_BLUE);
    myGLCD.fillRoundRect(90, 0, 220, 25);
    myGLCD.setColor(VGA_WHITE);
    myGLCD.drawRoundRect(90, 0, 220, 25);
    int h= Bus.time.hour();
    h= h-5;//restamos cinco horas para la obtener la hora del ecuador
    if (h<0)
    {
        h= h+24;
    }
    myGLCD.setFont(BigFont);
    myGLCD.setColor(VGA_WHITE);
    myGLCD.printNumI(h, 100, 5);
    myGLCD.print(":", 130, 5);
    myGLCD.printNumI(Bus.time.minute(), 140, 5);
    myGLCD.print(":", 170, 5);
    int seg=Bus.time.second();
    myGLCD.printNumI(seg, 180, 5);
}

void ibarraquito()
{
    int h= Bus.time.hour();
    h= h-5;//restamos cinco horas para la obtener la hora del ecuador
    if (h<0)
    {
        h= h+24;
    }
    long LONG=abs(Bus.location.lng())*10000;
    long LAT=abs(Bus.location.lat())*10000;
    tiempo++; //para tiempo de viaje
    Serial.print(tiempo);
    myGLCD.setFont(SevenSegNumFont);
    myGLCD.setBackColor(VGA_BLUE);
    myGLCD.setColor(VGA_WHITE);
    myGLCD.printNumI(Bus.speed.kmph(), 140, 30);
    int velocidad=Bus.speed.kmph();
    if(Bus.speed.kmph(<2)
    {
        velocidad=0;
    }
}

```

```

myGLCD.printNumI(velocidad, 140, 30);
if(velocidad<10)
{
myGLCD.setColor(VGA_BLUE);
myGLCD.fillRect(170, 30, 230, 80);
}
if(velocidad<100 && 90<velocidad)
{
myGLCD.setColor(VGA_BLUE);
myGLCD.fillRect(205, 30, 238, 80);
}
myGLCD.setColor(VGA_WHITE);
myGLCD.setFont(BigFont);
myGLCD.print("VELOC:", 40, 45);
myGLCD.print("km/h", 240, 45);
myGLCD.setFont(BigFont);
myGLCD.setColor(VGA_BLACK);
myGLCD.setBackgroundColor(VGA_LIME);
myGLCD.printNumI(h, 50, 145);
myGLCD.print(":", 80, 145);
myGLCD.printNumI(Bus.time.minute(), 90, 145);
myGLCD.print(":", 120, 145);
int seg=Bus.time.second();
myGLCD.printNumI(seg, 130, 145);
if(seg<10)
{
myGLCD.setColor(VGA_LIME);
myGLCD.fillRect(147, 140, 165, 170);
}
if(Bus.time.minute()==0)
{
myGLCD.setColor(VGA_LIME);
myGLCD.fillRect(106, 140, 118, 170);
}
//TIEMPO CONTROL INICIO
if(tiempo==1)
{
titulo=String("IBARRA-QUITO");
dia= Bus.date.day();
mes=Bus.date.month();
a_o=Bus.date.year();
hora=h;
minuto=Bus.time.minute();
}

//TIEMPO REAL 1
if(3615>=LAT && LAT>=3602 //IBARRA TERMINAL 3459>=LAT && LAT>=3432
{
if(781124>=LONG && LONG>=781114) //781291>=LONG && LONG>=781245
{
posicion1++;
if(posicion1==1)
{
hora1=h;
minuto1=Bus.time.minute();
tiempo2=tiempo;
diferencia1=(3-tiempo2)/60;
myGLCD.setFont(BigFont);
myGLCD.setBackgroundColor(VGA_LIME);
myGLCD.setColor(VGA_BLUE);
myGLCD.printNumI(diferencia1, 235, 145);
if(diferencia1<0)
{
hora1=hora11;
minuto1=minuto11-abs(diferencia1);
}
else if(diferencia1>=0)
{
hora1=hora11;
minuto1=minuto11+diferencia1;
}
}
}
}

//TIEMPO REAL 2
if(3595>=LAT && LAT>=3584) //LA FLORIDA 3405>=LAT && LAT>=3394
{
if(781218>=LONG && LONG>=781205) //781491>=LONG && LONG>=781471
{
posicion2++;
if(posicion2==1)
{
tiempo2=0;
hora12=h;
minuto12=Bus.time.minute();
}
}
}

```

```

tiempo2=tiempo;
diferencia2=(180-tiempo2)/60;
if(diferencia2>0)
{
myGLCD.setColor(VGA_BLUE);
myGLCD.fillRect(190, 130, 310, 180);
myGLCD.setBackgroundColor(VGA_BLUE);
}
if(diferencia2<0)
{
myGLCD.setColor(VGA_RED);
myGLCD.fillRect(190, 130, 310, 180);
myGLCD.setBackgroundColor(VGA_RED);
}
if(diferencia2==0)
{
myGLCD.setColor(VGA_LIME);
myGLCD.fillRect(190, 130, 310, 180);
myGLCD.setBackgroundColor(VGA_LIME);
}
myGLCD.setFont(BigFont);
myGLCD.setColor(VGA_YELLOW);
myGLCD.printNumI(diferencia2, 235, 145);
if(diferencia2<0)
{
hora2=hora12;
minuto2=minuto12-abs(diferencia2);
}
else if(diferencia2>=0)
{
hora2=hora12;
minuto2=minuto12+diferencia2;
}
}
}
}

//TIEMPO REAL 3
if(3556>=LAT && LAT>=3543) //SAN ANTONIO 3386>=LAT && LAT>=3370
{
if(781254>=LONG && LONG>=781237) //781697>=LONG && LONG>=781673
{
posicion3++;
if(posicion3==1)
{
tiempo2=0;
hora13=h;
minuto13=Bus.time.minute();
tiempo2=tiempo;
diferencia3=(360-tiempo2)/60;
if(diferencia3>0)
{
myGLCD.setColor(VGA_BLUE);
myGLCD.fillRect(190, 130, 310, 180);
myGLCD.setBackgroundColor(VGA_BLUE);
}
if(diferencia3<0)
{
myGLCD.setColor(VGA_RED);
myGLCD.fillRect(190, 130, 310, 180);
myGLCD.setBackgroundColor(VGA_RED);
}
if(diferencia3==0)
{
myGLCD.setColor(VGA_LIME);
myGLCD.fillRect(190, 130, 310, 180);
myGLCD.setBackgroundColor(VGA_LIME);
}
myGLCD.setFont(BigFont);
myGLCD.setColor(VGA_BLACK);
myGLCD.printNumI(diferencia3, 235, 145);
if(diferencia3<0)
{
hora3=hora13;
minuto3=minuto13-abs(diferencia3);
}
else if(diferencia3>=0)
{
hora3=hora13;
minuto3=minuto13+diferencia3;
}
}
}
}
}

//TIEMPO REAL 4

```

```

if(3475>=LAT && LAT>=3466) //ATUNTAQUI 3283>=LAT && LAT>=3265
{
  if(781333>=LONG && LONG>=781315) //782163>=LONG && LONG>=782139
  {
    posicion4++;
    if(posicion4==1)
    {
      tiempo2=0;
      hora14=h;
      minuto14=Bus.time.minute();
      tiempo2=tiempo;
      diferencia4=(540-tiempo2)/60;
      if(diferencia4>0)
      {
        myGLCD.setColor(VGA_BLUE);
        myGLCD.fillRect(190, 130, 310, 180);
        myGLCD.setBackgroundColor(VGA_BLUE);
      }
      if(diferencia4<0)
      {
        myGLCD.setColor(VGA_RED);
        myGLCD.fillRect(190, 130, 310, 180);
        myGLCD.setBackgroundColor(VGA_RED);
      }
      if(diferencia4==0)
      {
        myGLCD.setColor(VGA_LIME);
        myGLCD.fillRect(190, 130, 310, 180);
        myGLCD.setBackgroundColor(VGA_LIME);
      }
      myGLCD.setFont(BigFont);
      myGLCD.setColor(VGA_WHITE);
      myGLCD.printNumI(diferencia4, 235, 145);
      if(diferencia4<0)
      {
        hora4=hora14;
        minuto4=minuto14-abs(diferencia4);
      }
      else if(diferencia4>=0)
      {
        hora4=hora14;
        minuto4=minuto14+diferencia4;
      }
    }
  }
}

//TIEMPO REAL 5
if(3404>=LAT && LAT>=3394) //OTAVALO 2374>=LAT && LAT>=2357
{
  if(781486>=LONG && LONG>=781478) //782573>=LONG && LONG>=782555
  {
    posicion5++;
    if(posicion5==1)
    {
      tiempo2=0;
      hora15=h;
      minuto15=Bus.time.minute();
      tiempo2=tiempo;
      diferencia5=(720-tiempo2)/60;
      if(diferencia5>0)
      {
        myGLCD.setColor(VGA_BLUE);
        myGLCD.fillRect(190, 130, 310, 180);
        myGLCD.setBackgroundColor(VGA_BLUE);
      }
      if(diferencia5<0)
      {
        myGLCD.setColor(VGA_RED);
        myGLCD.fillRect(190, 130, 310, 180);
        myGLCD.setBackgroundColor(VGA_RED);
      }
      if(diferencia5==0)
      {
        myGLCD.setColor(VGA_LIME);
        myGLCD.fillRect(190, 130, 310, 180);
        myGLCD.setBackgroundColor(VGA_LIME);
      }
      myGLCD.setFont(BigFont);
      myGLCD.setColor(VGA_YELLOW);
      myGLCD.printNumI(diferencia5, 235, 145);
      if(diferencia5<0)
      {
        hora5=hora15;
        minuto5=minuto15-abs(diferencia5);
      }
    }
  }
}

```

```

else if(diferencia5>=0)
{
    hora5=hora15;
    minuto5=minuto15+diferencia5;
}
}
}

//TIEMPO REAL 6
if(3380>=LAT && LAT>=3374) //TABACUNDO 466>=LAT && LAT>=457
{
    if(781687>=LONG && LONG>=781675) //782020>=LONG && LONG>=782007
    {
        posicion6++;
        if(posicion6==1)
        {
            tiempo2=0;
            hora16=h;
            minuto16=Bus.time.minute();
            tiempo2=tiempo;
            diferencia6=(840-tiempo2)/60;
            if(diferencia6>0)
            {
                myGLCD.setColor(VGA_BLUE);
                myGLCD.fillRoundRect(190, 130, 310, 180);
                myGLCD.setBackgroundColor(VGA_BLUE);
            }
            if(diferencia6<0)
            {
                myGLCD.setColor(VGA_RED);
                myGLCD.fillRoundRect(190, 130, 310, 180);
                myGLCD.setBackgroundColor(VGA_RED);
            }
            if(diferencia6==0)
            {
                myGLCD.setColor(VGA_LIME);
                myGLCD.fillRoundRect(190, 130, 310, 180);
                myGLCD.setBackgroundColor(VGA_LIME);
            }
            myGLCD.setFont(BigFont);
            myGLCD.setColor(VGA_BLACK);
            myGLCD.printNumI(diferencia6, 235, 145);
            if(diferencia6<0)
            {
                hora6=hora16;
                minuto6=minuto16-abs(diferencia6);
            }
            else if(diferencia6>=0)
            {
                hora6=hora16;
                minuto6=minuto16+diferencia6;
            }
        }
    }
}

//TIEMPO REAL 7
if(3412>=LAT && LAT>=3403) //GUALLABAMBA 495>=LAT && LAT>=470
{
    if(781766>=LONG && LONG>=781754) //783442>=LONG && LONG>=783398
    {
        posicion7++;
        if(posicion7==1)
        {
            tiempo2=0;
            hora17=h;
            minuto17=Bus.time.minute();
            tiempo2=tiempo;
            diferencia7=(960-tiempo2)/60;
            if(diferencia7>0)
            {
                myGLCD.setColor(VGA_BLUE);
                myGLCD.fillRoundRect(190, 130, 310, 180);
                myGLCD.setBackgroundColor(VGA_BLUE);
            }
            if(diferencia7<0)
            {
                myGLCD.setColor(VGA_RED);
                myGLCD.fillRoundRect(190, 130, 310, 180);
                myGLCD.setBackgroundColor(VGA_RED);
            }
            if(diferencia7==0)
            {
                myGLCD.setColor(VGA_LIME);
            }
        }
    }
}

```

```

myGLCD.fillRect(190, 130, 310, 180);
myGLCD.setBackgroundColor(VGA_LIME);
}
myGLCD.setFont(BigFont);
myGLCD.setColor(VGA_WHITE);
myGLCD.printNumI(diferencia7, 235, 145);
if(diferencia7<0)
{
  hora7=hora17;
  minuto7=minuto17-abs(diferencia7);
}
else if(diferencia7>=0)
{
  hora7=hora17;
  minuto7=minuto17+diferencia7;
}
}
}

//TIEMPO REAL 8
if(3367>=LAT && LAT>=3360) // NATABUELA SEMAFORO
{
  if(781937>=LONG && LONG>=781924)
  {
    posicion8++;
    if(posicion8==1)
    {
      tiempo2=0;
      hora18=h;
      minuto18=Bus.time.minute();
      tiempo2=tiempo;
      diferencia8=(1080-tiempo2)/60;
      if(diferencia8>0)
      {
        myGLCD.setColor(VGA_BLUE);
        myGLCD.fillRect(190, 130, 310, 180);
        myGLCD.setBackgroundColor(VGA_BLUE);
      }
      if(diferencia8<0)
      {
        myGLCD.setColor(VGA_RED);
        myGLCD.fillRect(190, 130, 310, 180);
        myGLCD.setBackgroundColor(VGA_RED);
      }
      if(diferencia8==0)
      {
        myGLCD.setColor(VGA_LIME);
        myGLCD.fillRect(190, 130, 310, 180);
        myGLCD.setBackgroundColor(VGA_LIME);
      }
      myGLCD.setFont(BigFont);
      myGLCD.setColor(VGA_WHITE);
      myGLCD.printNumI(diferencia8, 235, 145);
      if(diferencia8<0)
      {
        hora8=hora18;
        minuto8=minuto18-abs(diferencia8);
      }
      else if(diferencia8>=0)
      {
        hora8=hora18;
        minuto8=minuto18+diferencia8;
      }
    }
  }
}

//TIEMPO REAL 9
if(3342>=LAT && LAT>=3331) //CARCELEN 978>=LAT && LAT>=897
{
  if(782044>=LONG && LONG>=782029) //784741>=LONG && LONG>=784651
  {
    posicion9++;
    if(posicion9==1)
    {
      tiempo2=0;
      hora19=h;
      minuto19=Bus.time.minute();
      tiempo2=tiempo;
      diferencia9=(1200-tiempo2)/60;
      if(diferencia9>0)
      {
        myGLCD.setColor(VGA_BLUE);
        myGLCD.fillRect(190, 130, 310, 180);

```



```

    myGLCD.setBackgroundColor(VGA_BLUE);
  }
  if(diferencia9<0)
  {
    myGLCD.setColor(VGA_RED);
    myGLCD.fillRect(190, 130, 310, 180);
    myGLCD.setBackgroundColor(VGA_RED);
  }
  if(diferencia9==0)
  {
    myGLCD.setColor(VGA_LIME);
    myGLCD.fillRect(190, 130, 310, 180);
    myGLCD.setBackgroundColor(VGA_LIME);
  }
  myGLCD.setFont(BigFont);
  myGLCD.setColor(VGA_YELLOW);
  myGLCD.printNuml(diferencia9, 235, 145);
  if(diferencia9<0)
  {
    hora9=hora19;
    minuto9=minuto19-abs(diferencia9);
  }
  else if(diferencia9>=0)
  {
    hora9=hora19;
    minuto9=minuto19+diferencia9;
  }
}
}
}
//TIEMPO REAL 10
if(3278>=LAT && LAT>=3269) //TERMINAL QUITO 1015>=LAT && LAT>=986
{
  if(782163>=LONG && LONG>=782146) //784720>=LONG && LONG>=784688
  {
    posicion10++;
    if(posicion10==1)
    {
      tiempo2=0;
      hora20=h;
      minuto20=Bus.time.minute();
      tiempo2=tiempo;
      diferencia10=(1320-tiempo2)/60;
      if(diferencia10>0)
      {
        myGLCD.setColor(VGA_BLUE);
        myGLCD.fillRect(190, 130, 310, 180);
        myGLCD.setBackgroundColor(VGA_BLUE);
      }
      if(diferencia10<0)
      {
        myGLCD.setColor(VGA_RED);
        myGLCD.fillRect(190, 130, 310, 180);
        myGLCD.setBackgroundColor(VGA_RED);
      }
      if(diferencia10==0)
      {
        myGLCD.setColor(VGA_LIME);
        myGLCD.fillRect(190, 130, 310, 180);
        myGLCD.setBackgroundColor(VGA_LIME);
      }
      myGLCD.setFont(BigFont);
      myGLCD.setColor(VGA_BLACK);
      myGLCD.printNuml(diferencia10, 235, 145);
      if(diferencia10<0)
      {
        hora10=hora20;
        minuto10=minuto20-abs(diferencia10);
      }
      else if(diferencia10>=0)
      {
        hora10=hora20;
        minuto10=minuto20+diferencia10;
      }
    }
  }
}
//VELOCIDAD MAXIMA
if(velocidad>=90)
{
  maxima++;
  if(maxima==10) //10 segundos
  {
    Serial.println("velocidad maxima");
    int vmax=velocidad;
    datos=SD.open("VMAX.csv",FILE_WRITE);
  }
}

```

```

if(datos)
{
  datos.println();
  datos.print(titulo);
  datos.print(",");
  datos.print(".");
  datos.print(dia);
  datos.print("/");
  datos.print(mes);
  datos.print("/");
  datos.print(a_o);
  datos.print(",");
  datos.print(".");
  datos.print(Bus.time.hour()-5);// RESTAR 5 HORAS XQ DA LA HORA DEL MERIDIANO CERO
  datos.print(".");
  datos.print(Bus.time.minute());
  datos.print(",");
  datos.print(Bus.location.lat(), 6);
  datos.print(",");
  datos.print(Bus.location.lng(), 6);
  datos.print(",");
  datos.print(vmax);
}
datos.close();
maxima=0;
}
else //CASO CONTRARIO MAXIMA VUELVE A CERO
{
  maxima=0;
}
if(velocidad>80 && 85>velocidad) //velocidad mayor a 80
{
  tone(altavoz,frecuencia3);
}
if(velocidad>85 && 90>velocidad) //velocidad mayor a 85
{
  tone(altavoz,frecuencia2);
}
if(velocidad>90) //velocidad mayor a 90
{
  tone(altavoz,frecuencia1);
}
if(velocidad<80) //velocidad menor a 80 no sonido
{
  noTone(altavoz);
}
//VELOCIDAD MINIMA
if(velocidad<30)// VELOCIDAD MENOR A 30
{
  minima++;
  if(minima==40) //10 segundos
  {
    Serial.println("velocidad minima");
    int vmin=velocidad;
    datos=SD.open("VMIN.csv",FILE_WRITE);
    if(datos)
    {
      datos.println();
      datos.print(titulo);
      datos.print(",");
      datos.print(".");
      datos.print(dia);
      datos.print("/");
      datos.print(mes);
      datos.print("/");
      datos.print(a_o);
      datos.print(",");
      datos.print(".");
      datos.print(Bus.time.hour()-5);
      datos.print(".");
      datos.print(Bus.time.minute());
      datos.print(",");
      datos.print(Bus.location.lat(), 6);
      datos.print(",");
      datos.print(Bus.location.lng(), 6);
      datos.print(",");
      datos.print(vmin);
    }
    datos.close();
    minima=0;
  }
}
else //CASO CONTRARIO MINIMA O SEGUNDOS VUELVE A CERO
{
  minima=0;
}

```

```

    }
}
void almacenar()
{
    almacenamiento++;
    if(almacenamiento==1)
    {
        datos=SD.open("VIAJE.csv",FILE_WRITE);
        if(datos)
        {
            datos.println();
            datos.print(titulo);
            datos.print(",");
            datos.print(".");
            datos.print(dia);
            datos.print("/");
            datos.print(mes);
            datos.print("/");
            datos.print(a_o);
            datos.print(",");
            datos.print(".");
            if(hora<10){datos.print("0");}
            datos.print(hora);
            datos.print(":");
            if(minuto<10){datos.print("0");}
            datos.print(minuto);
            datos.print(",");
            datos.print(".");
            if(hora1<10){datos.print("0");}
            datos.print(hora1);
            datos.print(":");
            if(minuto1<10){datos.print("0");}
            datos.print(minuto1);
            datos.print(",");
            datos.print(".");
            if(hora2<10){datos.print("0");}
            datos.print(hora2);
            datos.print(":");
            if(minuto2<10){datos.print("0");}
            datos.print(minuto2);
            datos.print(",");
            datos.print(".");
            if(hora3<10){datos.print("0");}
            datos.print(hora3);
            datos.print(":");
            if(minuto3<10){datos.print("0");}
            datos.print(minuto3);
            datos.print(",");
            datos.print(".");
            if(hora4<10){datos.print("0");}
            datos.print(hora4);
            datos.print(":");
            if(minuto4<10){datos.print("0");}
            datos.print(minuto4);
            datos.print(",");
            datos.print(".");
            if(hora5<10){datos.print("0");}
            datos.print(hora5);
            datos.print(":");
            if(minuto5<10){datos.print("0");}
            datos.print(minuto5);
            datos.print(",");
            datos.print(".");
            if(hora6<10){datos.print("0");}
            datos.print(hora6);
            datos.print(":");
            if(minuto6<10){datos.print("0");}
            datos.print(minuto6);
            datos.print(",");
            datos.print(".");
            if(hora7<10){datos.print("0");}
            datos.print(hora7);
            datos.print(":");
            if(minuto7<10){datos.print("0");}
            datos.print(minuto7);
            datos.print(",");
            datos.print(".");
            if(hora8<10){datos.print("0");}
            datos.print(hora8);
            datos.print(":");
            if(minuto8<10){datos.print("0");}
            datos.print(minuto8);
            datos.print(",");
            datos.print(".");
            if(hora9<10){datos.print("0");}
            datos.print(hora9);

```

```

datos.print(".");
if(minuto9<10){datos.print("0");}
datos.print(minuto9);
datos.print(",");
datos.print(".");
if(hora10<10){datos.print("0");}
datos.print(hora10);
datos.print(".");
if(minuto10<10){datos.print("0");}
datos.print(minuto10);
datos.print(",");
datos.print(".");
if(hora11<10){datos.print("0");}
datos.print(hora11);
datos.print(".");
if(minuto11<10){datos.print("0");}
datos.print(minuto11);
datos.print(",");
datos.print(".");
if(hora12<10){datos.print("0");}
datos.print(hora12);
datos.print(".");
if(minuto12<10){datos.print("0");}
datos.print(minuto12);
datos.print(",");
datos.print(".");
if(hora13<10){datos.print("0");}
datos.print(hora13);
datos.print(".");
if(minuto13<10){datos.print("0");}
datos.print(minuto13);
datos.print(",");
datos.print(".");
if(hora14<10){datos.print("0");}
datos.print(hora14);
datos.print(".");
if(minuto14<10){datos.print("0");}
datos.print(minuto14);
datos.print(",");
datos.print(".");
if(hora15<10){datos.print("0");}
datos.print(hora15);
datos.print(".");
if(minuto15<10){datos.print("0");}
datos.print(minuto15);
datos.print(",");
datos.print(".");
if(hora16<10){datos.print("0");}
datos.print(hora16);
datos.print(".");
if(minuto16<10){datos.print("0");}
datos.print(minuto16);
datos.print(",");
datos.print(".");
if(hora17<10){datos.print("0");}
datos.print(hora17);
datos.print(".");
if(minuto17<10){datos.print("0");}
datos.print(minuto17);
datos.print(",");
datos.print(".");
if(hora18<10){datos.print("0");}
datos.print(hora18);
datos.print(".");
if(minuto18<10){datos.print("0");}
datos.print(minuto18);
datos.print(",");
datos.print(".");
if(hora19<10){datos.print("0");}
datos.print(hora19);
datos.print(".");
if(minuto19<10){datos.print("0");}
datos.print(minuto19);
datos.print(",");
datos.print(".");
if(hora20<10){datos.print("0");}
datos.print(hora20);
datos.print(".");
if(minuto20<10){datos.print("0");}
datos.print(minuto20);
datos.print(",");
}
datos.close();
}
}

```