



*“Responsabilidad con pensamiento positivo”*

**UNIVERSIDAD TECNOLÓGICA ISRAEL**

**TRABAJO DE TITULACIÓN EN OPCIÓN AL GRADO DE:  
INGENIERO EN ELECTRÓNICA DIGITAL Y  
TELECOMUNICACIONES**

**TEMA:**

DESARROLLO DE UN PROTOTIPO PARA EL CONTROL DE PLAZAS  
VEHICULARES PARA LA ZONA AZUL DEL SECTOR DE LA MARISCAL EN LA  
CIUDAD DE QUITO

**AUTOR:**

DARWIN PAUL RUIZ GUTIÉRREZ

**TUTOR:**

PhD. ALFONSO JOSE ZOZAYA

**QUITO, ECUADOR**

**2019**

## **DECLARACIÓN**

Yo, Darwin Paul Ruiz Gutiérrez, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; que he consultado las referencias bibliográficas que se incluyen en este documento, y que no he plagiado dicha información.

---

Darwin Paul Ruiz Gutiérrez

# CERTIFICACIÓN TUTOR

UNIVERSIDAD TECNOLÓGICA ISRAEL

## APROBACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de titulación certifico:

Que el trabajo de titulación “**DESARROLLO DE UN PROTOTIPO PARA EL CONTROL DE PLAZAS VEHICULARES PARA LA ZONA AZUL DEL SECTOR DE LA MARISCAL EN LA CIUDAD DE QUITO.**”, presentado por el **Sr. Darwin Paul Ruiz Gutiérrez**, estudiante de la carrera de Electrónica Digital y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito D.M. Enero del 2019

TUTOR

.....

PhD. Alfonso José Zozaya

## **AGRADECIMIENTO**

Primero quiero agradecer a Dios por darme la vida, la fortaleza, su bendición y el don del conocimiento para así poder concluir una logro propuesto.

A mis padres, mis hermanos y toda mi familia por la confianza y el apoyo incondicional para que yo pueda superarme y poder terminar este objetivo en mi vida

Un agradecimiento especial al Ingeniero Alfonso Zozaya quien me guio con sus conocimientos durante el desarrollo de la presente trabajo de titulación. Igualmente a mis amigos y compañeros de estudio que compartimos conocimientos y tiempo, para poder culminar de este sueño.

**Darwin**

## **DEDICATORIA**

A mis Padres, Segundo Ruiz y Esthela Gutiérrez por ser un ejemplo a seguir como personas fuertes, trabajadoras y que no se rinden ante las adversidades y brindarme la mejor herencia que es el estudio, este trabajo es en reconocimiento a su apoyo incondicional y todo su esfuerzo.

A mi esposa Vanessa Torres y mi hijo Paul que son mi motor de vida y me inspiran día a día para ser una mejor persona gracias por su comprensión y todo el cariño brindado.

**Darwin**

# ÍNDICE

DECLARACIÓN.....	ii
CERTIFICACIÓN TUTOR.....	iii
AGRADECIMIENTO .....	iv
DEDICATORIA .....	v
ÍNDICE.....	vi
ÍNDICE DE FIGURAS .....	xii
ÍNDICE DE TABLAS .....	xv
RESUMEN .....	xvi
SUMMARY.....	xvii
INTRODUCCIÓN .....	xviii
Antecedentes de la situación de objeto de estudio .....	xviii
Planteamiento y justificación del problema .....	xix
Objetivo general: .....	xx
Objetivos específicos:.....	xx
Alcance .....	xxi
Descripción de los capítulos.....	xxii
CAPÍTULO I.....	1
1.    FUNDAMENTACIÓN TEÓRICA .....	1
1.1.    Introducción .....	1
1.2.    Introducción a Smart City .....	2
1.2.1. <i>Smart Mobility</i> .....	2

1.3.	Sistema de gestión para estacionamientos .....	3
1.4.	Sistema Tarifado de estacionamiento en Quito.....	5
1.5.	Internet de las Cosas (IoT) .....	5
1.5.1.	Aplicaciones y Beneficios del Internet de las Cosas .....	6
1.5.2.	IoT en sistemas de Parqueo Inteligente .....	6
1.6.	Sistemas de Parqueo Inteligente.....	6
1.6.1.	Los beneficios de un Parquímetro Inteligente .....	7
1.7.	<i>Arduino Board</i> .....	7
1.8.	Microcontrolador.....	8
1.9.	Servicios de GSM.....	10
1.9.1.	Servicio SMS.....	11
1.9.2.	Comandos AT.....	11
1.10.	<i>Microsoft Visual Studio y Visual Basic .NET</i> .....	12
1.11.	HTML.....	13
1.12.	Web.....	14
1.12.1.	Aplicación Web .....	14
1.13.	Sensor .....	15
1.14.	Actuadores .....	15
CAPÍTULO II .....		16
2.	MARCO METODOLÓGICO .....	16

2.1.	Modalidad de la investigación .....	16
2.2.	Tipos e instrumentos de la investigación .....	16
2.2.1.	Investigación bibliográfica .....	16
2.2.2.	Investigación de Campo.....	17
2.2.3.	Investigación Explicativa.....	17
2.3.	Variable de la investigación .....	17
2.3.1.	Cuantitativo.....	17
2.3.2.	Deductivo.....	18
2.3.3.	Inductivo .....	18
2.4.	Método Analítico Sintético .....	19
CAPÍTULO III.....		20
3.	PROPUESTA.....	20
3.1.	Arquitectura General .....	20
3.2.	Módulo de Control .....	23
3.3.	Descripción del Hardware utilizado.....	23
3.3.1.	Arduino MEGA ADK.....	23
3.3.2.	Sensor ultrasónico HC-SR04 .....	24
3.3.3.	<i>Display</i> GLCD .....	25
3.3.4.	Reloj DS3231 .....	27
3.3.5.	Lector SD o micro SD .....	29



3.3.6.	SIM900 GSM GPRS.....	30
3.3.7.	Selector de monedas CH-926.....	31
3.3.8.	Diodos LED de alta luminosidad.....	33
3.3.9.	Pulsadores de Selección.....	33
3.4.	Descripción del <i>Software</i> utilizado .....	34
3.4.1.	Software Libre .....	34
3.4.2.	Arduino IDE .....	34
3.4.3.	Software de Diseño Electrónico.....	35
3.5	Diseño Electrónico .....	36
3.5.1	Diseño de <i>Hardware</i> en <i>Proteus</i> .....	36
3.5.2	Diagrama Esquemático .....	36
3.6	Diseño de Software .....	39
3.6.1	Código para envío de información del Parquímetro.....	39
3.6.2	Código para envío de información del Servidor.....	42
3.7	Ventajas del Producto.....	43
3.8	Análisis de Costos .....	43
CAPÍTULO IV .....		44
4.	IMPLEMENTACIÓN .....	44
4.1	Desarrollo.....	44
4.1.1	Elaboración del PCB.....	44

4.2	Ensamblaje del prototipo parquímetro .....	48
4.2.1	Tótem Metálico.....	48
4.2.2	Montaje de equipos .....	49
4.3	Pruebas de Funcionamiento .....	51
4.3.1	Encendido del prototipo.....	52
4.3.2	Ejecución del censado del vehículo .....	53
4.3.3	Funcionamiento de los actuadores y luces.....	54
4.3.4	Funcionamiento de pantalla GLCD .....	56
4.3.5	Funcionamiento del monedero electrónico.....	57
4.3.6	Envío información hacia el servidor.....	58
4.3.7	Envío de la información a la página web.....	59
4.4	Análisis de resultados.....	60
	CONCLUSIONES .....	64
	RECOMENDACIONES.....	65
	BIBLIOGRAFÍA .....	66
	<b>ANEXOS</b> .....	68
	ANEXO A (DATASHEET ARDUINO MEGA).....	69
	ANEXO B (DATASHEET SENSOR HC-SR04).....	77
	ANEXO C (MANUAL MONEDERO CH-926).....	81
	ANEXO D (PLACAS ELECTRÓNICAS PARQUÍMETRO) .....	84

ANEXO E (TABLA DE COSTOS DEL PROTOTIPO) .....	86
ANEXO F (DIAGRAMAS ELECTRÓNICOS) .....	87
ANEXO F (MANUAL USUARIO PARQUÍMETRO) .....	89
1. INTRODUCCIÓN .....	3
2. PARTES DEL TÓTEM .....	4
2.1 Pantalla Visualización Principal.....	4
2.2 Selectores de Plazas .....	4
2.3 Luces Led de alarmas a eventos .....	5
2.4 Monedero Electrónico .....	6
3. MODOS DE USO DEL PARQUÍMETRO ELECTRÓNICO .....	6
3.1 Plaza Libre.....	6
3.2 Plaza ocupada y sin pago .....	7
3.3 Plaza ocupada y pagada.....	8
3.4 Monedas a ingresar.....	9

## ÍNDICE DE FIGURAS

<b>Figura 1-1:</b> Señalización de estacionamientos .....	4
<b>Figura 1-2:</b> Parquímetros Mecánicos.....	4
<b>Figura 1-3:</b> Tarjeta Arduino UNO .....	8
<b>Figura 1-4:</b> Estructura interna de un microcontrolador .....	9
<b>Figura 3-1:</b> Arquitectura de la Solución de Parqueo .....	20
<b>Figura 3-2:</b> Diagrama en bloques del parquímetro electrónico .....	21
<b>Figura 3-3:</b> Arquitectura del Parquímetro Electrónico .....	22
<b>Figura 3-4:</b> Parquímetro Electrónico .....	23
<b>Figura 3-5:</b> Tarjeta Arduino MEGA ADK .....	24
<b>Figura 3-6:</b> Diagrama Arduino – HC-SR04 .....	25
<b>Figura 3-7:</b> Diagrama de flujo del sensor HC-SR04 .....	25
<b>Figura 3-8:</b> Display GLCD 128x64.....	26
<b>Figura 3-9:</b> Diagrama Arduino – Pantalla GLCD .....	26
<b>Figura 3-10:</b> Módulo de reloj DS3231 .....	27
<b>Figura 3-11:</b> Diagrama Arduino – Módulo DS3231 .....	28
<b>Figura 3-12:</b> Diagrama de flujo del Módulo DS3231 .....	28
<b>Figura 3-13:</b> Lector Micro SD.....	29
<b>Figura 3-14:</b> Diagrama Arduino – Módulo SD .....	29

<b>Figura 3-15:</b> Diagrama de flujo del Módulo SD .....	30
<b>Figura 3-16:</b> Shield SIM900 GPRS .....	31
<b>Figura 3-17:</b> Selector de Monedas CH-926.....	32
<b>Figura 3-18:</b> Led RGB.....	33
<b>Figura 3-19:</b> Led RGB.....	33
<b>Figura 3-20:</b> Entorno Desarrollo Arduino IDE .....	34
<b>Figura 3-21:</b> Pantallas de Trabajo ISIS .....	35
<b>Figura 3-22:</b> Diagrama esquemático del Tarjeta Principal.....	37
<b>Figura 3-23:</b> Diagrama esquemático control de luces .....	38
<b>Figura 3-24:</b> Diagrama esquemático control Sensores.....	39
<b>Figura 4-1:</b> Circuito impreso .....	45
<b>Figura 4-2:</b> Impresión del diseño y colocación en la Fibra. ....	46
<b>Figura 4-3:</b> Transferencia de la impresión en la baquelita .....	46
<b>Figura 4-4:</b> Retiro de la impresión de las pistas. ....	47
<b>Figura 4-5:</b> Montaje de elementos electrónicos en placa .....	47
<b>Figura 4-6:</b> Módulo de control completo.....	48
<b>Figura 4-7:</b> Dimensiones de la Caja Contenedora.....	49
<b>Figura 4-8:</b> Caja Contenedora .....	49
<b>Figura 4-9:</b> Montaje de PCB y actuadores .....	50

<b>Figura 4-10:</b> Montaje de los dispositivos en la parte delantera .....	50
<b>Figura 4-11:</b> Instalación de sensores .....	51
<b>Figura 4-12:</b> Pantalla GLCD de inicialización .....	52
<b>Figura 4-13:</b> Verificación sensor plaza1.....	54
<b>Figura 4-14:</b> Luz roja plaza1 ocupada.....	54
<b>Figura 4-15:</b> Luz roja plaza1 ocupada y pagada.....	55
<b>Figura 4-16:</b> Luz azul plaza 2 libre.....	55
<b>Figura 4-17:</b> Pantalla GLCD principal .....	56
<b>Figura 4-18:</b> Pantalla GLCD de plaza ocupada.....	56
<b>Figura 4-19:</b> Pantalla GLCD de plaza libre .....	57
<b>Figura 4-20:</b> Validación de 10 cent en el monedero.....	58
<b>Figura 4-21:</b> Envío comandos AT hacia el servidor.....	58
<b>Figura 4-22:</b> Recepción trama en la Web .....	59

## ÍNDICE DE TABLAS

<b>Tabla 1.1</b> Teleservicios GSM .....	11
<b>Tabla 3.1</b> Cabecera del <i>String</i> de envió .....	40
<b>Tabla 3.2</b> Código selección de plazas .....	41
<b>Tabla 3.3</b> Códigos envió desde el Servidor .....	42
<b>Tabla 4.1</b> Lista de validación global del parquímetro.....	53
<b>Tabla 4.2</b> Lista de validación sensores .....	60
<b>Tabla 4.3</b> Lista de validación luces RGB.....	61
<b>Tabla 4.4</b> Lista de validación pantalla GLCD .....	61
<b>Tabla 4.5</b> Lista de validación de Monedero Electrónico .....	62
<b>Tabla 4.6</b> Lista de validación del SIM900 .....	63

## **RESUMEN**

En este documento se detalla el diseño y construcción de un prototipo para el control de plazas disponibles en la Zona Azul de la Mariscal. El prototipo consiste principalmente en un tótem de control con sistema microprocesado, El módulo de control está constituido por una tarjeta Arduino el que se encargará de realizar la comunicación entre el parquímetro y el servidor, a este también se conecta con varios dispositivos de entrada, salida.

Como dispositivos de entrada se tiene sensores de proximidad ultrasónicos HC-SR04 que validarán el estado del parqueo, además se dispone de un botones de selección y un monedero electrónico. Los dispositivos de salida son una pantalla GLCD y luces Led RGB, mientras que en la parte de comunicaciones se usa un módulo SIM900 compatible con Arduino.

El prototipo es capaz de interactuar y controlar 4 plazas de parqueo, para realizar el cobro por el uso de los estacionamientos. Por cada evento de uso de un estacionamiento, el prototipo interactúa con el cliente mostrando diferentes acciones, en la pantalla nos indican por ejemplo la hora de llegada y el tiempo que tiene el vehículo para permanecer estacionado y las luces Led RGB cambia de tonalidad de acuerdo a dichos eventos.

El módulo SIM900 permite la comunicación del prototipo con el servidor de para realizar el monitoreo remoto de los estacionamientos, así como para llevar un registro que permita ejercer un control más sólido en el uso de los parquímetros.

### **Palabras Claves**

Arduino

Servidor

Plazas

Sensores

Parquímetro



## SUMMARY

This document details the design and construction of a prototype for the control of available spaces in the Zona Azul of La Mariscal. The prototype consists mainly of a control totem with microprocessed system, The control module is constituted by an Arduino card that is in charge of making the communication between the parking meter and the server to this also connects with several input devices, output.

As input devices we have HC-SR04 ultrasonic proximity sensors that will validate the parking status, in addition there is a selection button and an electronic purse. The output devices are a GLCD screen and RGB Led lights, while in the communications part a SIM900 module compatible with Arduino is used.

The prototype is able to interact and control 4 parking spaces, to charge for the use of parking lots. For each event of use of a parking lot, the prototype interacts with the client showing different actions as on the screen indicate for example the time of arrival and the time the vehicle has to remain parked and the RGB Led lights change of tonality according to those events.

The SIM900 module allows the communication of the prototype with the server to perform the remote monitoring of the parking lots, as well as to keep a record that allows to exercise a more solid control in the use of the parking meters.

Keywords

Arduino

Server

Plazas

Sensors

Parquímetro

# INTRODUCCIÓN

## **Antecedentes de la situación de objeto de estudio**

Un estudio llevado a cabo por IBM en más de 50 ciudades de todo el mundo revela que cada ciudad tiene una propia problemática de movilidad, plazas de parqueo y recaudación por el uso del parqueo, los responsables de movilidad defieren retos comunes y buscan que sus ciudades sean más limpias y reduzcan la congestión mejorando tráfico, sobre todo incrementando y mejorando el uso de espacios públicos de parqueo especialmente para vehículos privados. La congestión vehicular generada por el mal estacionamiento es uno de los principales retos a los que se enfrentan mayoría de ciudades, la implementación de nuevos sistemas de parqueo rotativo tarifado inteligente conlleva beneficios sorprendentes. Sin embargo, la mayoría de estos proyectos en Latinoamérica aún se encuentran en una fase muy inicial, por lo cual los líderes gubernamentales coinciden en que es necesario invertir en infraestructuras que ayuden eficazmente la movilidad y el sistema de parqueo público.

Actualmente en países desarrollados se encuentra muchos sistemas electrónicos que ayudan a dar soluciones a problemas existentes en la sociedad, una de ellas es la movilidad y el sistema de parqueo público, ya permiten mantener el orden, el monitoreo de eventos y la correcta recolección por los valores de parqueo.

## **Planteamiento y justificación del problema**

La movilidad implica en el desarrollo urbano de un país, por ende acarrea un sin número de problemas socio-económicos que afectan la funcionalidad y en general la felicidad de los pobladores. La solución está lejos de ser encontrada como una fórmula de aplicación, por lo que son motivo de estudios y experimentaciones en todo el mundo. Para los gobiernos y de manera especial para las municipalidades, la movilidad se ha constituido en uno de los ejes de mayor preocupación por resolver, pues su situación es notoria y se siente en la ciudadanía y los diferentes sectores sociales, profesionales y productivos, quienes exigen soluciones siempre inmediatas.

En la actualidad el sistema de parqueo rotativo tarifado en la ciudad de Quito se maneja en ciertas zonas más conflictivas con el programa zona azul, este se opera de manera informal o con recursos humanos que se encargan de administrar una cantidad específica de espacios o de calles y que generan un ticket de pago específico, por lo que no hay un correcto control de espacios de uso, de tiempos de estacionamiento, de recaudación y a su vez genera más tráfico en estas zonas.

El proyecto permitirá la construcción de un tótem de interrelación a un costo asequible. Su manejo será sencillo e intuitivo, podrá controlar 4 plazas vehiculares además tendrá pantallas de visualización de mensajes informativo y preventivo, se contará con luces de alerta, botones de selección y un monedero electrónico, también tendrá sistemas de seguridad para apertura y cambio de alcancías de los valores, adicionalmente constará con un sistema de visualización de eventos en tiempo real que nos ayudara a prevenir cualquier incidente.

**Objetivo general:**

- Desarrollar un prototipo para el control de plazas vehiculares en la zona azul del sector de la Mariscal en la ciudad de Quito

**Objetivos específicos:**

- Diseñar un sistema de control automático de plazas en la zona azul mediante un controlador Arduino Mega ADK y su tarjeta de control SIM 900 para la comunicación inalámbrica hacia un Servidor Central.
- Construir tótem de servicios que permita la interrelación con el cliente para el pago del parqueadero específico y verificación de su tiempo disponible.
- Desarrollar un sistema de monitoreo de los eventos sucedidos en los tótems de servicios en tiempo real.

## **Alcance**

En el desarrollo de este proyecto se ha considerado cubrir los siguientes aspectos:

- Construcción de una cabina de interrelación a un costo accesible que cuente con una pantalla GLCD, 4 botones de selección, luces led indicadoras, un monedero electrónico y que controlen 4 estaciones de parqueo.
- Selección de equipos electrónicos de vanguardia que ayuden en la captación, procesamiento y envío de la información.
- Diseño del Hardware de control para procesar información de actuadores y sensores.
- Desarrollo de la programación del controlador Arduino en la estación de servicio y programación de eventos sucedidos.
- Montaje de todos los elementos y tarjetas electrónicas involucradas en el módulo de control y pago.
- Puesta en marcha, pruebas de funcionamiento, calibraciones y ajustes necesarios para el funcionamiento óptimo del proceso.
- Elaboración manual de usuario y operación.

## **Descripción de los capítulos**

En el capítulo I se detalla fundamentos teóricos de las tecnologías empleadas, los principales componentes utilizados, Beneficios de IoT en movilidad, *Software* Libre, Módulo Arduino, Módulo GSM SIM 900 y el protocolo de comunicación GSM que se aplicaran en el desarrollo de este proyecto.

En el capítulo II se explica los mecanismos de investigación: cualitativa y cuantitativa, tipos de investigación: teórica, explicativa y descriptiva. Instrumentos o herramientas de investigación.

El capítulo III contiene la propuesta del trabajo de titulación, detalla el diseño y construcción del *hardware* y la programación del software utilizado, también detallas las características técnicas de los dispositivos que se implementaran, los diagramas de bloques y el flujograma de funcionamiento del sistema. También incluye los diseños y esquemas electrónicos del proyecto.

El capítulo IV se explica la implementación total del proyecto, la construcción de las tarjetas de control, los softwares utilizados para elaboración de los programas tanto del Arduino como del sistema de eventos, también las pruebas de funcionamiento del tótem y del sistema completo.

Finalmente se mencionan conclusiones y recomendaciones recopiladas durante la construcción del prototipo, así como también las fuentes bibliográficas y se adjuntan anexos.

# CAPÍTULO I

## FUNDAMENTACIÓN TEÓRICA

### 1.1. Introducción

Según los últimos datos del Anuario de Transportes 2015, publicado por el Instituto Nacional de Estadística y Censos (INEC) en el Ecuador hay aproximadamente 2 millón de vehículos matriculados y en ciudades grandes como Quito y Guayaquil concentran las mayores cantidades. En la capital existen alrededor de 492.568 automotores matriculados (INEC, 2016) de Quito se concentra un porcentaje equivalente a casi el 30 % del total del parque automotor nacional.

Este proyecto tiene como fin la automatización y construcción de un servicio de parqueo inteligente para las calles de Quito llamado “Parquímetro Electrónico” este proyecto ayudará reducir el tráfico vehicular formado por la falta de áreas de estacionamiento. A la par se plantea conseguir un mayor rendimiento, con el cobro automatizado de los valores de estacionamiento, conociendo en tiempo real las plazas libres, fecha y hora, valores reales de pagos que se generen con los distintos módulos, actuadores, estableciendo métodos acorde a las nuevas tendencias tecnológicas para *Smart City*.

## **1.2. Introducción a Smart City**

Según (Puche Lara, 2014, p.25). Dice que se considera una *Smart City* a aquella ciudad que emplea las tecnologías de la información y las comunicaciones para mejorar su infraestructura y que sus componentes, servicios públicos ofrecidos sean más interactivos, eficientes y la ciudadanía pueda tener más beneficios a través de estas plataformas. De igual manera comenta que para que una ciudad esté más comprometida con su entorno se debe tomar en cuenta el punto de vista medioambiental como en lo relativo a los elementos culturales e históricos

Desde el punto de vista del ciudadano una *Smart City* es aquella ciudad que le hace la vida más fácil y cómoda sin notarlo. Todo está encaminado a optimizar el tiempo de las personas, evitando pérdidas del mismo en tareas habituales. Desde el aspecto tecnológico consiste en el uso de la tecnología en todos sus niveles para alcanzar nuevos productos innovadores que faciliten la forma de hacer que cualquier servicio o aplicación estén interconectadas y saber lo que está ocurriendo en el mundo real.

### **1.2.1. Smart Mobility**

Es común en todas las ciudades (ya sean grandes o pequeñas) el gran problema de encontrar aparcamiento tanto en espacios públicos como privados. La búsqueda de aparcamiento produce en el usuario desatención hacia la tarea de conducción, aumentando la inseguridad al volante. Así mismo se produce un incremento en el vertido a la atmósfera de gases contaminantes, y a su vez, incrementa la congestión de vehículos en las calles.

El proceso en apariencia interminable de dar vueltas hasta encontrar un lugar donde aparcar es una experiencia demasiado común entre los conductores. Los retrasos motivados por el aparcamiento producen frustración a los usuarios, afectan negativamente al medio ambiente, reducen la seguridad y suponen pérdidas económicas en la población.

Los sistemas tarifados de estacionamiento surgen de la necesidad de generar espacios de aparcamiento en sectores de mucha concurrencia de personas como son centros turísticos, plazas comerciales y centros financieros, donde la oferta de lugares de parqueo es limitada, y



el espacio físico no está diseñado para absorber la demanda de la población que visita en coche a estos lugares.

### **1.3. Sistema de gestión para estacionamientos**

Es común en todas las ciudades (ya sean grandes o pequeñas) el gran problema de encontrar aparcamiento tanto en espacios públicos como privados. La búsqueda de aparcamiento produce en el usuario desatención hacia la tarea de conducción, aumentando la inseguridad al volante. Así mismo se produce un incremento en el vertido a la atmósfera de gases contaminantes, y a su vez, incrementa la congestión de vehículos en las calles.

El proceso en apariencia interminable de dar vueltas hasta encontrar un lugar donde aparcar es una experiencia demasiado común entre los conductores. Los retrasos motivados por el aparcamiento producen frustración a los usuarios, afectan negativamente al medio ambiente, reducen la seguridad y suponen pérdidas económicas en la población.

Los sistemas tarifados de estacionamiento surgen de la necesidad de generar espacios de aparcamiento en sectores de mucha concurrencia de personas como son centros turísticos, plazas comerciales y centros financieros, donde la oferta de lugares de parqueo es limitada, y el espacio físico no está diseñado para absorber la demanda de la población que visita en coche a estos lugares.

Los sistemas tarifados de estacionamiento usan rotulación vertical y horizontal muy particular de color azul para indicar su cobertura, costo, horario y el tiempo máximo por ocupar una plaza de estacionamiento. La Figura 1.1 muestra una rotulación en la zona de la mariscal este indica que los espacios de paqueo situados junto a la vereda son tarifados



**Figura 1-1:** Señalización de estacionamientos  
**Fuente:** Diario El Comercio

En Ecuador el Sistema Tarifado de estacionamiento se ha materializado en ciudades de la región sierra como ejemplos a las siguientes ciudades: Ambato, Cuenca, Ibarra, Loja y Quito. Los diferentes sistemas que operan en cada una de las ciudades mencionadas son los manuales mientras que en ciudades de la costa como Salinas, Playas y Samborondón ya se ha optado por sistemas mecánicos de pago denominados parquímetros (El Comercio, 2018). En la Figura 1.2. Se aprecia un sistema de parquímetro mecánico instalado en zonas de la ciudad de Samborondón.



**Figura 1-2:** Parquímetros Mecánicos  
**Fuente:** Diario Expreso

#### **1.4. Sistema Tarifado de estacionamiento en Quito**

El municipio de Quito a través de la empresa pública metropolitana de movilidad y obras públicas en vista de la gran demanda de espacios de parqueo para usuarios con vehículos de transporte liviano y semi-pesados en la zona de la Mariscal, vio la necesidad de crear un sistema de estacionamiento tarifado llamado “Zona Azul” que se sustenta en la Ordenanza Metropolitana No. 0221, la Resolución No. 0555, el Reglamento No. 0455, y la Ordenanza Metropolitana No. 031. (Consejo Metropolitano de Quito, 2012). Su objetivo principal es devolver a los usuarios el derecho a utilizar la vía pública de una manera organizada y ordenada a través de la generación de una oferta de plazas libres de parqueo. Este es un sistema de cobro semiautomático que al estacionar una persona encargada genera un recibo con la hora y fecha de parqueo y cobra por dicho servicio, pero no es muy confiable ya que entra a competir con el error humano que pueda existir por el incorrecto uso del mismo.

Este sistema genera contravenciones como estacionar un vehículo en la zona tarifada sin haber realizado previamente el pago correspondiente al tiempo de estacionamiento o exceder el ciclo de uso del aparcamiento que haya sido cubierto mediante el pago del valor referente. El valor que el infractor deberá cancelar en cualquiera de los casos antes mencionados un valor corresponde al diez por ciento (10%) del salario mínimo unificado (Cuerpo de agentes de control Quito, 2018).

#### **1.5. Internet de las Cosas (IoT)**

El concepto IoT tiene como objetivo principal la conexión a internet tanto seres animados como inertes, dando a conocer el estado de todo lo interconectado para establecer un vínculo con estos de forma inmediata.

Puede parecer un despropósito conectar objetos a la red, ya que estos no son capaces de mantener conversaciones razonables con un usuario, pero tampoco es lo que se pretende. Lo que se busca es adquirir información relevante acerca del estado de un objeto o de su entorno, para con esto realizar acciones determinadas gracias a la sonorización e interconexión de multitud de dispositivos y ambientes a internet (Jayavardhana Gubbi, Octubre 2011).

Objeto físico + controlador + sensor + actuador + Internet = Internet de las cosas

Internet de las cosas es un instrumento de ayuda para la humanidad, que ha demostrado que puede mejorar las condiciones en nuestra vida cotidiana a su vez mejorado la eficiencia de todas las aplicaciones donde se lo usa.

### **1.5.1. Aplicaciones y Beneficios del Internet de las Cosas**

IoT orienta el futuro de cualquier cuerpo en cualquier sitio y en un determinado momento. IoT es conocido por interconectar varios dispositivos físicos con las redes. En IoT, varios dispositivos físicos están integrados con diferentes tipos de sensores y otros dispositivos para intercambiar datos entre ellos. Este sistema es una mezcla entre *software* y hardware donde se programan para ejecutar situaciones específicas de funcionamiento. Se puede conocer el valor y estado de estos datos desde cualquier parte del mundo haciendo uso de la nube. Esto se puede utilizar para crear un mundo digital, hogares inteligentes, sistemas de salud y el intercambio de datos de la vida real como la banca inteligente. Aunque el Internet de las cosas ha surgido hace mucho tiempo, ahora se está volviendo popular.

A la par existen también algunos desafíos al usar IOT. La seguridad es la principal preocupación al usar IOT ya que si no se toma bien consideración en este punto los datos pueden ser fácilmente obtenidos y podrían modificar dichos valores, como en los valores de recaudación, donde se requiere más seguridad. El estándar es otro factor en el que se debe utilizar un equipo IOT bien diseñado y adecuado y de un costo considerablemente bajo.

### **1.5.2. IoT en sistemas de Parqueo Inteligente**

Actualmente existen muchos aplicativos con IoT, este proyecto de titulación se concentrará en la utilización de tecnología IoT en Sistemas de Parqueadero “Parquímetro Electrónico”. Primero se detalla el concepto de Sistemas de Parqueadero Inteligente y surgen beneficios del uso de este tipo de sistemas.

## **1.6. Sistemas de Parqueo Inteligente**

El objetivo en los sistemas de parqueo inteligente es ayudar a los conductores a localizar plazas de aparcamiento de una manera eficiente mediante tecnologías de la información y Apps para dispositivos móviles. Si bien la idea inicial se ha propuesto hace diez años, los

últimos cinco años se ve cada vez más sistemas de parqueo inteligente. Debido al rápido aumento del uso de teléfonos inteligentes, ahora muchos son capaces de conectarse a Internet y buscar información cuando están fuera de casa (Herrador, 2013).

Con el fin de conocer el estado de las plazas de parqueo, se instalan sensores fijos para poder detectar presencia vehicular. Se puede utilizar una variedad de sensores, como son: sensores ultrasónicos, sensores infrarrojos, magnetómetros, sensores magnéticos, sensores de peso en movimiento, o la tecnología (RFID) que es identificación por radio frecuencia, igualmente ordenadores de imagen, de video.

Los sensores envían la información en tiempo real a los dispositivos de almacenamiento de datos y mediante estas aplicaciones los conductores pueden obtener la información actualizada en una pantalla de visualización y podrán intercambiar acciones como pago por el servicio o visualización del tiempo disponible de su espacio.

#### **1.6.1. Los beneficios de un Parquímetro Inteligente**

El plan general de cualquier mecanismo con tecnología IoT es hacer la vida más fácil. Al ser una aplicación IoT tienen varias ventajas; no exclusivamente para los conductores, sino también para la rentabilidad de recaudación y el público en general.

Un beneficio puede ser la capacidad para registrar datos sobre el aparcamiento. Todos los datos son afinadamente y secuencialmente registrados y se pueden utilizar para los análisis correspondientes. Por ejemplo: ¿Cuándo un vehículo se estaciona?, ¿En qué espacio se estaciona? ¿Cuál es tiempo de estacionamiento? ¿Cuál es valor en fines de semana o en horas pico? Sobre una base de estos datos, el administrador puede reconocer patrones y tratar de influir en ellos, por ejemplo, el uso de diferentes tasas por hora. Las tácticas de precios pueden ser aplicados con el fin de mejorar los beneficios económicos para de la Ciudad.

#### **1.7. Arduino Board**

Es una estructura electrónica de diseño de *hardware* libre asentada en una placa con un microcontrolador. Con *software* y *hardware* dúctiles y fáciles de manipular, Arduino ha sido creado para adaptarse a necesidades de toda índole, como son para aficionados de la

electrónica y experimentados desarrolladores de equipos electrónicos y robóticos. Igualmente consta de un simple, pero consumado, entorno de desarrollo, que admite interactuar con la plataforma de manera muy amigable. Se puede especificar como una sencilla herramienta de contribución a la diseño de prototipos, u objetos interactivos destinados a proyectos de toda índole y de multitecnologías. En la Figura 1.3 se muestra un Arduino Uno que es una de las placas más vendidas a nivel mundial.



**Figura 1-3:** Tarjeta Arduino UNO  
**Fuente:** <https://www.arduino.cc/en/Guide/ArduinoUNO>

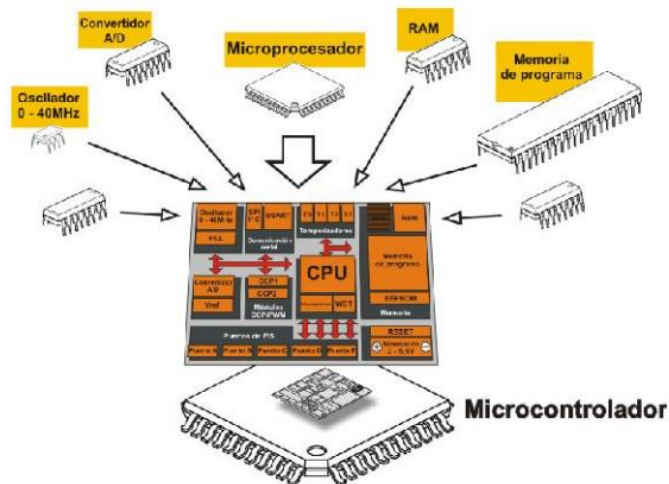
Con la tarjeta de adquisición Arduino se puede recopilar extensa información del entorno sin mucha complejidad. Por sus pines de entrada se puede trabajar con toda gama de sensores de toda marca o presentación como temperatura, flujo, presencia luminosidad, presión y que nos ofrecen la forma de controlar o actuar sobre ciertos principios del entorno que le rodean, un ejemplo claro es el control luces, encendido de motores, activación de alarmas y numerosos actuadores. Gracias a que la plataforma es de hardware libre, “Las placas Arduino pueden ser adquiridas y manipuladas por cualquier aficionado o compradas ya montadas de fábrica ya que Arduino está basado en microcontroladores ATMEGA8 y ATMEGA168 de Atmel” (Arduino, 2018).

### **1.8. Microcontrolador**

Según (Miñarro, 2009). Un microcontrolador representa un dispositivo que alberga un sistema pequeño dentro de un chip, entre estos están CPU, memorias RAM, ROM, *drivers*, reloj en tiempo real, periféricos como conversores análogos / digitales (A/D) y temporizadores.

La designación de Intel para familia de microcontroladores 8051 es nombrar MCS 51. Pero durante este tiempo la mayor parte de empresas fabricantes de semiconductores, han ido proyectando productos que han conquistado el mercado y que hace extremadamente complicado para los ingenieros y desarrolladores seleccionar el microcontrolador más óptimo para cada aplicación, pues se presentan micros con capacidades de trabajo de 8 bits, 16 bits y en la actualidad de 32 bits. Con arquitecturas extremadamente completas, capacidades de cálculo elevadas y consumen menor energía.

La Figura 1.4 se muestra que es un sistema electrónico, inmerso en un circuito integrado en el cual convergen los elementos de un procesador programable, que se caracteriza por poseer un sistema físico completo que se ajusta a las necesidades de su entorno. Su tamaño reducido y su capacidad de configuración hicieron que su utilización tenga un crecimiento vertiginoso, lo que amplía su campo de aplicación.



**Figura 1-4:** Estructura interna de un microcontrolador  
**Fuente:** (Verle, 2009)

Estos sistemas utilizan una tecnología compleja en la que interactúan distintos conceptos correlacionados entre sí, que dificultan establecer normas que garanticen su fiabilidad y modificación de los sistemas constituidos por ellos para mejorar sus prestaciones en función de su diseño.

## 1.9. Servicios de GSM

El servicio que aporta la tecnología GSM es el de telefonía. En GSM, las señales analógicas de voz son decodificadas digitalmente para posteriormente transmitirse por la red como corriente de datos.

Según (Tomasi, 2003). En la prestación de datos por GSM los usuarios pueden enviar y recibir datos hasta con 9600 bps a suscriptores que estén establecidos en las redes POTS (*Plain Old Telephone Service*), servicio telefónico antiguo simple y usuarios que tengan ISDN, para una red pública de datos en paquetes conmutados (PSPDN, de *Packet Switched Public Data Network*), y con una red pública de datos de circuitos conmutados, mediante diversos métodos y protocolos de acceso, como X.25. También, como la GSM es una red digital, no se requiere módem entre el usuario y esa red

En Europa, el sistema GSM manipula las bandas de frecuencia de 850, 900 y 1800 MHz, y en países como Estados Unidos se utiliza la banda de frecuencia de 1900 MHz. En consecuencia algunos dispositivos de comunicación móvil pueden operar tanto en Europa como en Estados Unidos.

La tarjeta SIM es la que se encarga de identificar y validar la red al usuario y al teléfono móvil. Estos aparatos se identifican gracias a un número característico de identificación llamado IMEI (Identificador internacional de equipos móviles), combinado por 15 dígitos. Es decir cada tarjeta contiene una SIM que posee un número de identificación único denominado IMSI (Identificador internacional de abonados móviles).

En correlación a la arquitectura de red GSM todo terminal móvil estar comprendido por una tarjeta SIM (Módulo de identificación de abonado) y el apropiado dispositivo, para regular a un teléfono móvil.

GSM es un sistema de telefonía celular que permite la transmisión digital de voz y datos con una gran eficacia que se ha desarrollado por todo el mundo. Para utilizarlo, debe colocarse un teléfono celular compatible, acreditarse al servicio GSM, utilizar un cable GSM y emplear un software de comunicaciones compatible con GSM.



El GSM propone igualmente un servicio de intercambio de mensajes alfanumérico corto (140 caracteres como máximo) tanto en emisión como en recepción. Puede ser utilizado en emisión por todos aquellos equipos terminales capaces de emitir mensajes alfanuméricos hacia una Terminal GSM.

La tabla 1.1 ofrece la lista de servicios GSM, frente al tipo de información que pueden asociarse a dichos teleservicios que como se puede observar, son diferentes a los servicios portadores.

**Tabla 1.1** Teleservicios GSM

<b>Tipo de Información</b>	<b>Servicios Ofrecidos</b>
Palabras	Telefonía, llamadas de urgencia
Datos	Mensajes Punto a Punto
Textos Cortos	Transmisión de mensajes cortos alfanuméricos
Gráfica	Fax

**Fuente:** <https://goo.gl/XkMufJ>

### **1.9.1. Servicio SMS**

Es una prestación de mensajes cortos (*Short Message Service*), está adecuado en los teléfonos móviles que reconocen el mensaje de envíos cortos. Entre teléfonos fijos, móviles, otros terminales de mano, fue creado originariamente como es estándar de telefonía móvil digital GSM. El servicio final propuesto es una comunicación extremo-extremo entre la estación móvil (MS) y la entidad (SME).

### **1.9.2. Comandos AT**

Los comandos AT, también distinguidos como comandos Hayes (en honor a su desarrollador Dennis Hayes), son una cadena de instrucciones que conforman un interfaz de

comunicación entre usuario y un servidor. Su abreviatura AT por la que son mundialmente conocidos estos comandos procede de la palabra ‘*attention*’.

Aunque la finalidad principal de los comandos AT fue la comunicación con módems, la telefonía móvil GSM/GPRS igualmente amparó este lenguaje como estándar de comunicación.

Actualmente, todos los teléfonos móviles GSM poseen una serie determinada de comandos AT que permiten configurarlos por medio de estas instrucciones y mostrarles una serie de acciones que necesitamos que se ejecuten, como marcar un número de teléfono, enviar o leer un SMS, preguntar el estado de conexión a la red, escribir en la agenda de contactos.

Gracias a los comandos AT ya no se depende del canal de comunicación a través de los cuales sean enviados como infrarrojo, *Bluetooth*, se podrá manipular la tarjeta de adquisición Arduino para la transmisión de los comandos AT a un módulo que pueda interpretarlos y funcionar con el correcto resultado.

Como son muchos los comandos se consultara en el manual de comandos AT oficial de nuestro módulo GSM/GPRS (SIM900), únicamente se mencionará junto con una breve descripción, aquellos que puedan trascender de forma principal en el desarrollo del proyecto.

#### **1.10. Microsoft Visual Studio y Visual Basic .NET**

*Microsoft Visual Studio* es una plataforma de desarrollo integrado (IDE, por sus siglas en inglés) que funcionan en sistemas operativos Windows y tiene en su entorno un editor, un compilador y un depurador de código. En dicho entorno de desarrollo, están soportados varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, y *Visual Basic .NET*, son parecidos a los entornos de desarrollo web tales como ASP.NET (Studio, 2018).

Visual Studio permite a los programadores crear aplicativos, sitios y aplicaciones web, así como servicios web en cualquier entorno que pueda soportar la plataforma .NET. Así

también se pueden crear aplicaciones que se interrelacionen entre servidores, páginas web y dispositivos móviles.

Para este Proyecto Final de Carrera se ha manejado Visual Basic .NET para el desarrollo de la aplicación de la interfaz gráfica que recoge datos de la tarjeta de red del ordenador y muestra por pantalla la situación actual del entorno controlado.

Visual Basic .NET es un lenguaje de programación dirigido por eventos y orientado a objetos, fue desarrollado por Alan Cooper para la empresa Microsoft. La primera versión surgió en el año 1991 con un medio relativamente sencillo para facilitar la creación de programas gráficos y que se puede especificar como un idioma del clásico lenguaje BASIC implementado sobre el Framework .NET. Visual Basic .NET suele presentarse como un sistema RAD (*Rapid Application Development*), ya que permite desarrollar aplicaciones de forma rápida, especialmente para demos. (Studio, 2018).

### **1.11. HTML**

Según (Group CSS, 2018). *HyperText Markup Language* es un lenguaje sobresaliente para la elaboración de páginas web. Fue desarrollado por la Organización Europea de Investigación Nuclear (CERN) en el año 1945 con el fin de crear un sistema de almacenamiento. Es un lenguaje sencillo cuya codificación se lleva a cabo mediante el uso de etiquetas, que permiten interconectar diversos conceptos y formatos

Mediante estas etiquetas se crean diferentes tipos de elementos que tendrán sentido tanto por su contenido como por la especificación de sus atributos. Cabe destacar, que HTML permite la incorporación de ciertos códigos adicionales que se conocen como script, los cuales aportan funcionalidad a la página web. Entre los scripts que pueden agregarse, destacamos JavaScript que ha sido el utilizado en el Proyecto actual.

## **1.12. Web**

La palabra WEB es el vocablo muy usado para referirse a la Red Mundial de páginas o Documentos de texto enlazados. Un documento enlazado solo es un documento que contiene varios enlaces a otros documentos o páginas de texto. También conocidas con el apodo de documento de Hipertexto.

Se puede definir que “WEB” es la red encargada de presentarnos de manera visual el contenido alojado en el internet, se emplea un lenguaje de programación llamado HTML1. Fue creada en 1989 por el británico Tim Berners-Lee quien es considerado el padre de la WEB y es probablemente el servicio más popular de Internet (Kendall, 2005).

### **1.12.1. Aplicación Web**

Se nombra aplicación web a todas los instrumentos que los desarrolladores pueden utilizar ingresando a un servidor web a través de la intranet o Internet con el beneficio de un navegador. También se dice que es una aplicación de software que se cifran en un lenguaje sobrellevado por los navegadores web en la que se aloja al navegador.

“En la Ingeniería de software se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un Servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador” (Ecuared., 2018).

“Una aplicación web es un conjunto de páginas que interactúan unas con otras y con diversos recursos en un servidor web, incluidas bases de datos. Esta unión permite efectuar algunas características en su sitio como cuadros de productos virtuales y administradores de noticias y contenidos. Adicionalmente se podrá ejecutar consultas en bases de datos, registrar e ingresar solicitudes, pedidos y múltiples tipos de información en línea en tiempo real” (SurOnline, 2018).

### **1.13. Sensor**

El sensor es un dispositivo que cuando se expone a un fenómeno físico (temperatura, desplazamiento, fuerza, etc.) genera una señal de salida proporcional tanto eléctrica, mecánica o magnética. El término transductor es a menudo utilizado como sinónimo de sensor, sin embargo, idealmente, un sensor es un dispositivo que responde a un cambio En el fenómeno físico. Por otro lado, un transductor es un dispositivo que convierte una forma de Energía en otra forma de energía.

Los sensores son transductores cuando detectan una forma de entrada de energía y la salida en una forma diferente de energía. Por ejemplo, un termopar responde a un cambio de temperatura (Energía térmica) y produce un cambio proporcional en la fuerza electromotriz (energía eléctrica). Por lo tanto, Un termopar puede llamarse sensor y / o transductor (Anjanappa, 2002).

Los sensores también se pueden clasificar como pasivos o activos. En los sensores pasivos, la potencia necesaria para la salida es proporcionada por el propio fenómeno físico detectado (tal como un termómetro), mientras que la salida los sensores activos requieren una fuente de alimentación externa. Además, los sensores se clasifican como analógicos o digitales basados en el tipo de señal de salida. Analógicamente los sensores producen señales continuas.

### **1.14. Actuadores**

Los actuadores son básicamente el músculo detrás de un sistema electrónico que acepta un comando de control principalmente en forma de una señal eléctrica y produce un cambio en el sistema físico generando fuerza, Movimiento, calor, flujo, etc. Normalmente, los actuadores se utilizan conjuntamente con la fuente de alimentación y mecanismo de acoplamiento, la unidad de alimentación suministra CA o CC a la Tensión y corriente. El mecanismo de acoplamiento actúa como la interfaz entre el actuador y el sistema. Mecanismos típicos incluyen cremallera y piñón, transmisión por engranaje, accionamiento por correa, tornillo sinfín y tuerca, pistón, y los vínculos (Anjanappa, 2002).

## **CAPÍTULO II**

### **MARCO METODOLÓGICO**

#### **2.1. Modalidad de la investigación**

Esta investigación está enmarcado en un enfoque mixto integrando métodos cuantitativo y cualitativo que muestra una investigación aplicada, que en base al proyecto de estudio se adaptan en el avance del trabajo investigativo y se orientan en analizar un modelo de propuesta que nos brindan la mejor solución al problema.

#### **2.2. Tipos e instrumentos de la investigación**

En el presente estudio se aplica la investigación que esta enmarcada en el proceso explicativo, descriptivo y de campo.

##### **2.2.1. Investigación bibliográfica**

El presente proyecto se basó en utilizar investigación bibliográfica, debido a que la definición científica del tema investigado está enfocado a la tecnología siendo esta la que permite dejar ver las relaciones fundamentales del objeto de investigación esta aplicado para realizar una formulación de la hipótesis del plan estudiado y de esta forma permite diagnosticar la siguiente metodología de la investigación.

“Los métodos teóricos están potencializados para realizar del salto cualitativo que permite escalar del acondicionamiento de información empírica, explica determinar los inconvenientes para la representación de las teorías en la investigación”. (Ocaña, 2009)

### **2.2.2. Investigación de Campo**

Se basó en los datos primarios adquiridos frente a la realidad del desorden vehicular al parquear, la misma que fue establecida directamente de la fuente para verificar la condición actual que está atravesando la Zona de la mariscal en la ciudad de Quito. En definitiva permite estudiar una situación para poder diagnosticar el o los problemas existentes en ese sitio de estudio.

### **2.2.3. Investigación Explicativa**

#### **2.3. Variable de la investigación**

Esta investigación está vinculada con las causas, por lo tanto describe el problema. Permite encontrar los fundamentos de porque es necesario instalar un parquímetro electrónico en este sector, para después buscar la solución oportuna y adecuada de esta manera exponer el por qué ocurre y en qué situaciones se da o porque dos o más variables están identificadas.

También indaga sobre la necesidad de tener el control de las plazas de parqueo en la Zona Azul del sector de la mariscal, debido a esto se puede establecer conclusiones para tener una visión de la aplicación del prototipo de movilización, dentro de un espacio determinado.

#### **2.3.1. Cuantitativo**

Se denomina cuantitativo debido a que de manera estadística se presentaron las respuestas con mayor relevancia dentro del proceso de investigación, las mismas que ayudaron a certificar la viabilidad o no del prototipo del sistema de movilización.

Está dirigida a profundizar casos específicos y no ha generalizado, su preocupación no es prioritariamente calcular, sino calificar y describir el fenómeno social desde faces definitivas, como sean descubiertos por los elementos oportunos que están entre la situación analizada o estudiada (Torres, 2006, pág. 45).

Este método permite encontrar los problemas de seguridad existentes en la sociedad, de tal manera establecer cuál es la dirección del trabajo investigado.

### **2.3.2. Deductivo**

En efecto realiza un análisis previo dando a conocer los problemas de recaudación por parte del municipio considerando en lo particular las necesidades y demandas para explicar el procedimiento de sistema evidenciando La problemática actual de pago y cobros por el espacio de parqueo y proponer como solución un nuevo sistema de cobro.

### **2.3.3. Inductivo**

Muestra los inconvenientes que actualmente existen en el estacionamiento en la calle por ende poder conocer de qué manera la implementación de un nuevo método logrará auxiliar a eliminar los problemas existentes. Consiste en basar en hechos y estudios y resultados de observaciones o experiencias donde se analiza, observa y clasifica para solicitar una hipótesis que proponga una solución al problema planteado y ayude a elaborar la justificación y los antecedentes del mismo(Sánchez, 2012).

Este método se utiliza para exteriorizar los problemas que actualmente existen, dando a conocer un nuevo método de visualización y pago por el espacio ocupado, el mismo será amigable y puede ser utilizado por todos los usuarios.



## **2.4. Método Analítico Sintético**

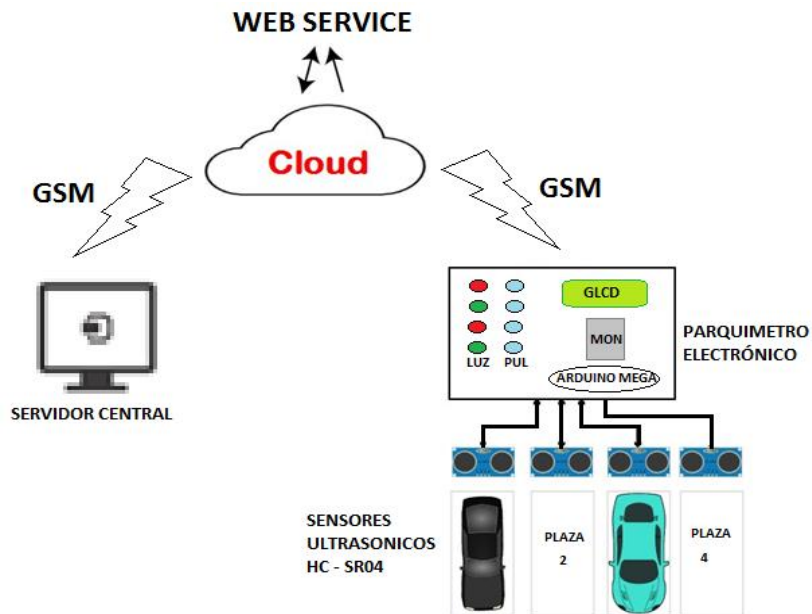
Este método es un procedimiento el cual permite conocer más del objeto de análisis en varios elementos o componentes para adquirir nuevos conocimientos se deduce y aplicar para comprender mejor su actuación para establecer las nuevas teorías. “Facilita para el observación y resolución del problema con la finalidad de ofrecer facilidad, bienestar y calidad a los clientes el tiempo de obtener una plaza para parquear” (Garrido, Ivan Hurtado Leon & Josefina Toro, 20011).

## CAPÍTULO III

### PROPUESTA

#### 3.1. Arquitectura General

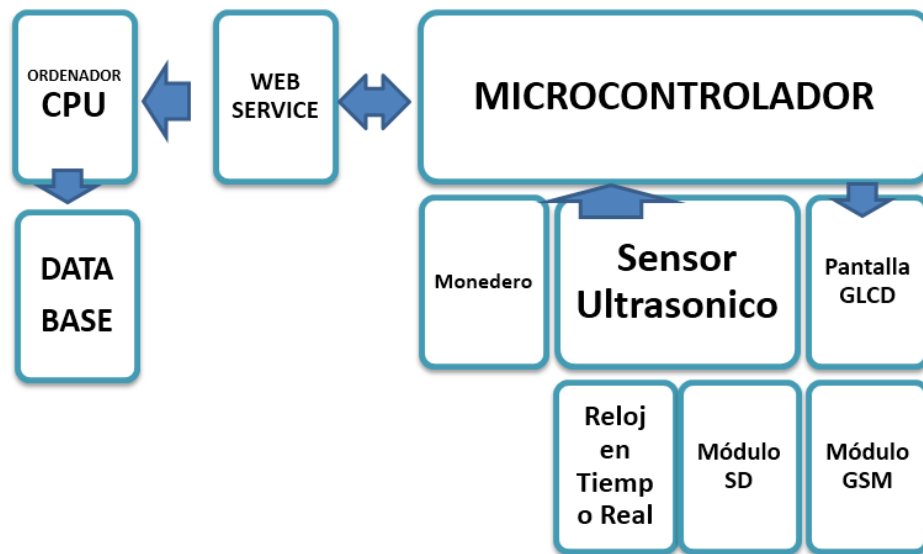
Para diseñar, construir e implementar la estación de pago Parquímetro Electrónico se establece controlar 4 estaciones de pago mediante sensores ultrasónicos HC – SR04 que verificarán la presencia de los vehículos estacionados en las plazas asignadas, estos sensores estarán conectados a través de cable hacia el tótem de control, En la Figura 3.1 se muestra la arquitectura global del sistema a implementar.



**Figura 3-1:** Arquitectura de la Solución de Parqueo  
**Fuente:** Propia

Al tener el control de los espacios de parqueo se puede presentar la fecha y hora de estacionamiento en la pantalla GLCD, a la par las luces led cambiarán su estado ya sean a colores verde, rojo o azul de acuerdo al evento presentado, también se podrá registrar de forma eficaz los ingresos por el pago de las plazas de parqueo mediante el monedero electrónico toda esta información de la estación será enviada a un servidor central para guardar dicha información en una base de datos.

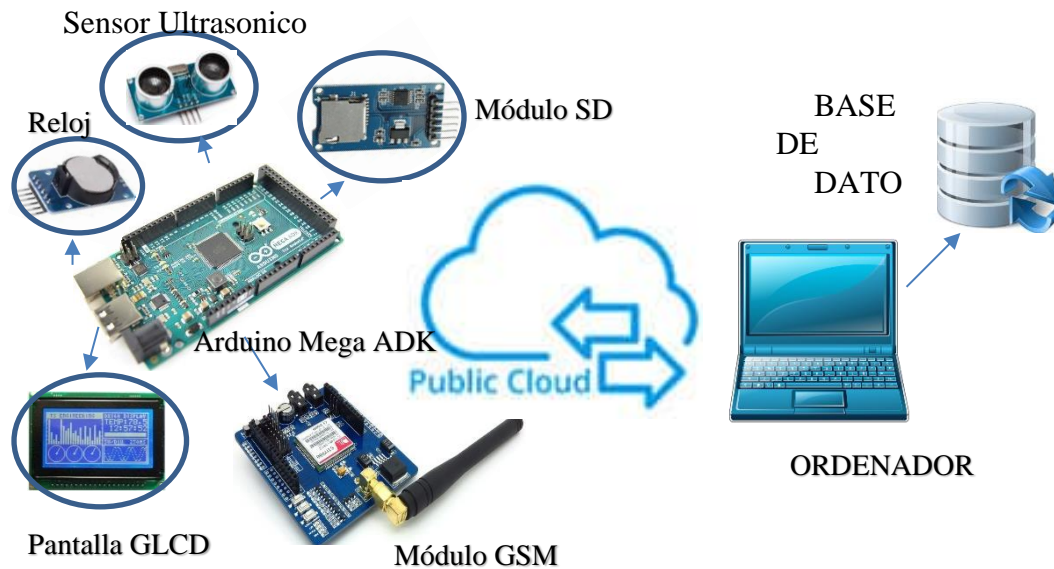
Internamente el parquímetro electrónico propone un sistema distribuido que componen de módulos electrónicos interconectados a un microcontrolador principal (Arduino Mega) este procesará la información adquirida y la publicará mediante comunicación GSM a un receptor que se conectará a un *Web Service* para poder interpretar la información a un ordenador en la Figura 3.2 se muestra el diagrama de bloques de la solución del prototipo a desarrollar.



**Figura 3-2:** Diagrama en bloques del parquímetro electrónico  
**Fuente:** Propia

La estación de pago estará conformada por un controlador principal (Arduino) que interpretará los datos de los sensores y permitirá interactuar con actuadores, botones, pantallas Led y un monedero electrónico. Estos datos se transmitirán mediante tecnología GSM con un

módulo de comunicación inalámbrica SIM 900 a un ordenador el cual procesará la información y la acumulará en un a base de datos, luego se publicara en un *Web Service* para en una pantalla conocer los eventos sucedidos de cada estación. En la Figura 3.3 se muestra un diagrama de operación con los dispositivos a utilizar en este proyecto.



**Figura 3-3:** Arquitectura del Parquímetro Electrónico  
**Fuente:** Propia

En el diagrama de la Figura 3.3 se observa que se utiliza un Arduino Mega ADK como el cerebro de la estación de pago al cual se conecta varios módulos adicionales. A este microprocesador se conecta un Reloj DS3231 que nos dará fecha y hora en tiempo real y constará de un módulo SD que permitirá guardar todos los valores del monedero electrónico cuando se desconecte del servicio eléctrico. También se conecta al microcontrolador una pantalla para que el usuario pueda interactuar y verificar los correctos eventos. Por último el sistema de control constará con un módulo SIM 900 para la comunicación inalámbrica hacia el servidor.

A continuación se detallan los equipos, datos técnicos, eléctricos y electrónicos que conforman el parquímetro Electrónico.

### 3.2. Módulo de Control

Es la unidad que se encargará de la visualización, interrelación entre los dispositivos internos y el cliente, será un tótem metálico en donde se colocará las pantallas, las luces indicadoras, los pulsadores, el monedero electrónico en la Figura 3.4 se muestra un de parquímetro electrónico que distribuye (Insumos *Turnomatic*, 2017) de similares características al que se va a construir.



**Figura 3-4:** Parquímetro Electrónico  
**Fuente:** Insumos *Turnomatic* & Parquímetros

### 3.3. Descripción del Hardware utilizado

Los dispositivos internos del parquímetro electrónico esta compuestos por la placa controladora Arduino Mega ADK, un reloj DS3231, un módulo SD, una pantalla GLCD, luces led RGB, pulsadores, sensores ultrasónicos HC –SR04 y un monedero electrónico.

#### 3.3.1. Arduino MEGA ADK

Se optó por un Arduino Mega ADK por ser la placa con mayores características de la familia Arduino ver Figura 3.5 el Arduino Mega ADK está asentado en el microcontrolador Atmega2560, que contiene 54 pines digitales E/S, 16 pines de entradas analógicas, un

oscilador de cristal de 16 MHz, un *plug* de alimentación que tolera voltajes superiores a los 9 voltios y un botón de *reset*.



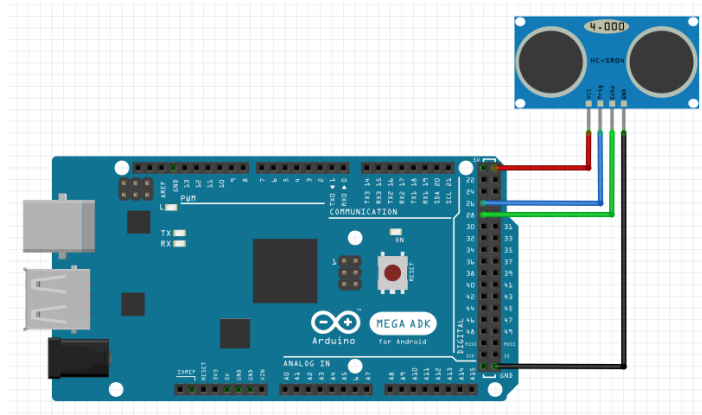
**Figura 3-5:** Tarjeta Arduino MEGA ADK

**Fuente:** <https://www.arduino.cc/en/Guide/Arduino-MEGA-ADK>

Es totalmente compatible con las *shields* que se utilizarán, también cuenta con una memoria que duplica su capacidad en comparación con el resto de placas, por tanto es el más adecuado para nuestro proyecto en los que se requerirá un gran número de entradas y salidas disponibles, además abarcará un código de programación extenso y centralizará todas las funciones en una sola plataforma como control de tiempo, envío SMS, activación de alarmas, luces he interpretación de interrupciones.

### **3.3.2. Sensor ultrasónico HC-SR04**

El módulo ultrasónico nos permitirá establecer la ubicación del vehículo en el espacio asignado ya que ofrecen un excelente rango de detección sin contacto, su trabajo no es afectado por la luz solares o materiales oscuros. El rango máximo es de 4 metros, y el más corto la distancia medible es de 2 cm con una resolución de 3 mm. Trabajan a un voltaje de 5 voltios con un consumo menor a los 2 mA en reposo y 15mA en presencia de algún objeto, en la Figura 3.6 Se establece la conexión entre el Arduino y el sensor HC-SR04 utilizando los pines *Trig* y *Echo* del sensor hacia una entrada y salida digital del Arduino. Para la ejecución de nuestro prototipo y cada vez que el sensor detecta un obstáculo este refleja una alarma de aviso hacia el microcontrolador.



**Figura 3-6:** Diagrama Arduino – HC-SR04  
**Fuente:** Elaboración propia

El módulo HC-SR04 no necesita ningún biblioteca extra El módulo ultrasónico procesa la información como se muestra en el diagrama de flujo de la Figura 3.7



**Figura 3-7:** Diagrama de flujo del sensor HC-SR04  
**Fuente:** Elaboración propia

### 3.3.3. Display GLCD

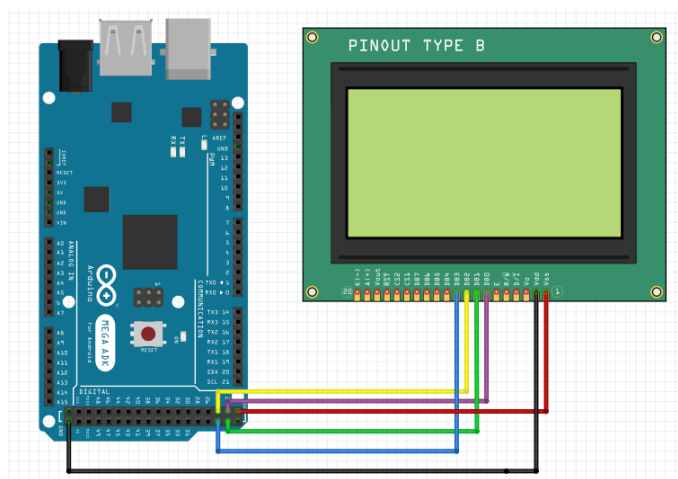
También conocida como pantalla grafica es un módulo que permite exponer información gráfica o alfanumérica al usuario de manera visual a través de 128 X64 píxeles de

resolución he incluye *displays* de cristal líquido o LED para su iluminación. En la Figura 3.8 se aprecian los caracteres que presentan información en correspondencia con la programación que se montará en el Arduino. Se puede alimentar con un voltaje de 5 -7 voltios un consumo de aproximadamente 220mA con la luz de fondo al 100%, la luz de fondo es azul y su contraste es ajustable con un potenciómetro externo.



**Figura 3-8:** Display GLCD 128x64  
**Fuente:** (García González, Navarro, & Montenegro, 2013)

En la Figura 3.9 Se muestra la conexión entre el Arduino y la pantalla GLCD que se utilizará en nuestro módulo de interrelación con el usuario. Para este caso será necesario una biblioteca adicional al programa principal que nos facilitara la correcta manipulación y control de la pantalla.



**Figura 3-9:** Diagrama Arduino – Pantalla GLCD  
**Fuente:** Elaboración propia



Los Pines Vss y Vdd son destinados para conectar los 5 voltios y tierra, que requiere el módulo para su funcionamiento. El Pin Vo se encarga de ajustar el contraste de los datos presentados y visualizados en la pantalla. Para terminar los Pines DB0 al DB7 representan las 8 líneas de transmisión de datos.

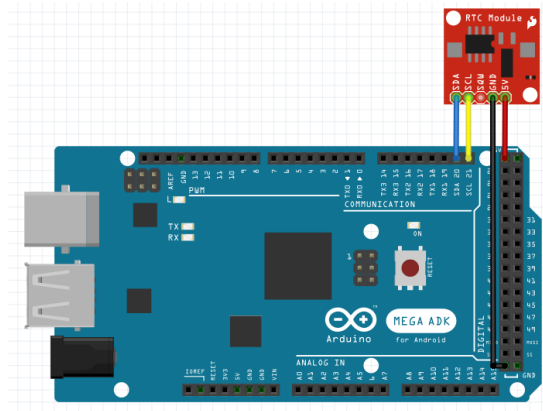
### 3.3.4. Reloj DS3231

El módulo DS3231 que se muestra en la Figura 3.10 es de gran utilidad al trabajar con eventos que requieren exactitud a lo largo del tiempo y ese facilitará realizar el proyecto para registrar todos los eventos en tiempo real hasta aproximadamente la fecha 2100. Es un reloj de alta fidelidad que maneja una comunicación serie tipo I<sup>2</sup>C. Este circuito integrado lleva la cuenta de la hora y también la fecha, es capaz de seguir operando con una tensión de 3Vdc absorbiendo muy poca corriente, lo que le admite usar una batería de litio como alimentación, para poder seguir llevando la hora aun cuando se haya desconectado de la fuente de alimentación.



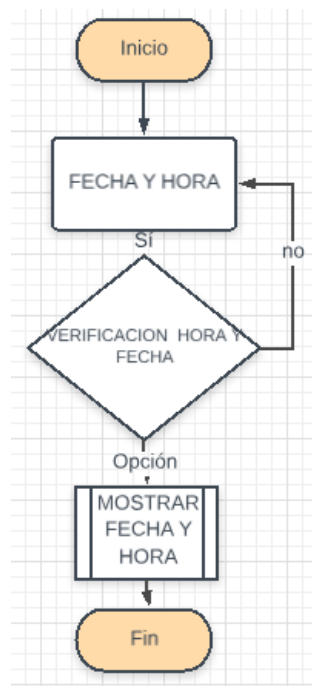
**Figura 3-10:** Módulo de reloj DS3231  
**Fuente:** (García González, Navarro, & Montenegro, 2013)

La Figura 3.11 se muestra el módulo de reloj, necesario para actualizar los registros, se comunica con Arduino utilizando el protocolo I<sup>2</sup>C, por lo que se conectará los pines VCC, GND para la alimentación y los pines SCL y SDA en la comunicación tipo serie hacia el Arduino.



**Figura 3-11:** Diagrama Arduino – Módulo DS3231  
**Fuente:** Elaboración propia

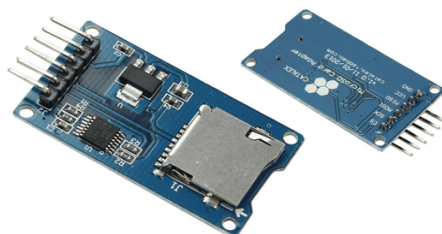
El módulo DS3231 necesita una biblioteca adicional para su correcto funcionamiento el cual se adjunta en el programa principal cargado al Arduino. El módulo RTC procesa la información como se muestra en el diagrama de flujo de la Figura 3.12



**Figura 3-12:** Diagrama de flujo del Módulo DS3231  
**Fuente:** Elaboración propia

### 3.3.5. Lector SD o micro SD

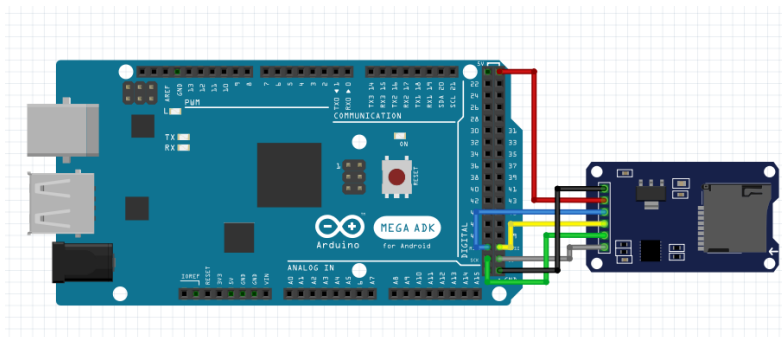
Dentro del desarrollo con Arduino se encuentran lectores SD estos son los módulos que permiten almacenar información mediante el uso de micro SD ver Figura 3.13 la lectura y conexión con el controlador principal se puede realizarse a través de bus SPI que es con la tecnología que se implementara en este proyecto por su alta tasa de transferencia de datos.



**Figura 3-13:** Lector Micro SD

**Fuente:** <https://www.luisllamas.es/tarjeta-micro-sd-arduino/>

Tiene un voltaje de alimentación de 3.3Vdc, pero se conectará de forma sencilla a Arduino, ya que el modulo incluye un regulador de voltaje que permite alimentar directamente a 5V. En la Figura 3.14 se muestra de diagrama de conexión para nuestro prototipo.

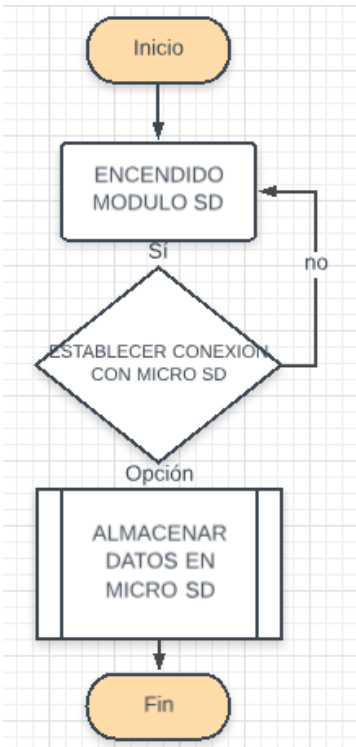


**Figura 3-14:** Diagrama Arduino – Módulo SD

**Fuente:** Propia

Emplear una tarjeta SD o micro SD con Arduino tiene la ventaja de proporcionar una memoria casi ilimitada para nuestros proyectos. Además es no volátil y puede ser extraída y

conectada a un ordenador con facilidad. En la Figura 3.15 se muestra el flujograma correspondiente al módulo SD este no necesita una librería adicional al programa Arduino.

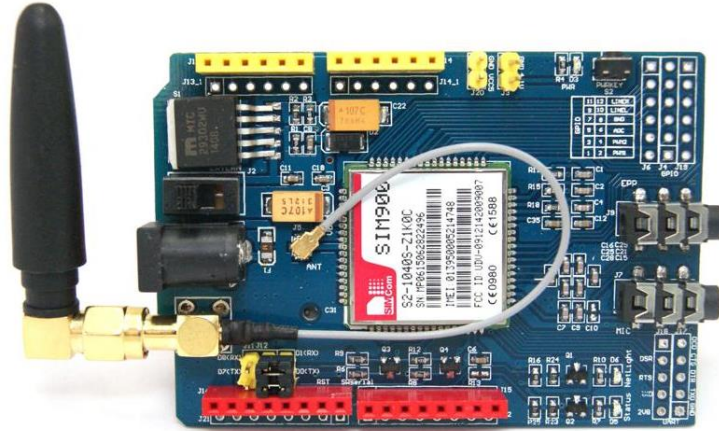


**Figura 3-15:** Diagrama de flujo del Módulo SD

**Fuente:** Propia

### 3.3.6. SIM900 GSM GPRS

El SIM900 GSM/GPRS *shield* de la Figura 3.16 es un módulo de altas prestaciones muy compacto de comunicación inalámbrica que es compatible con todos los módulos de Arduino, esta tarjeta está basada en un controlador SIM900 GSM. El GPRS está dispuesto para controlarse por vía UART usando los comandos AT. Por lo tanto sólo se conecta la tarjeta al Arduino y emprende la comunicación a través de comandos AT es óptimo para sistemas remotos, comunicación directa entre diferentes puntos de control, mandar mensajes de texto a celulares y conectarse a la nube.



**Figura 3-16:** Shield SIM900 GPRS  
**Fuente:** hetpro-store.com

Se seleccionó esta tarjeta ya que es muy compatible con Arduino para controlar los eventos del parquímetro vía GSM, también porque es de fácil adquisición en el mercado local además su costo es bajo.

No cabe duda de que se trata de un módulo barato que se puede adquirir en el mercado local a un costo relativamente bajo y es esencial para el uso de sistemas de comunicación móvil y que ofrece todos los servicios que requerimos para poder llevar a cabo el desarrollo de nuestro proyecto.

La tarjeta se conecta directamente al microcontrolador Arduino y comienza la comunicación mediante comandos AT, permite enviar SMS, MMS, GPRS utiliza una banda cuádruple: 850; 900; 1800 y 1900 MHZ, por lo que funciona perfectamente con cualquier operadora de nuestro país. Además de tiene un voltaje de alimentación en un rango de 3.8 a 5 voltios con un consumo de 22mA en reposo y 2 A al momento de la transmisión.

### **3.3.7. Selector de monedas CH-926**

Es un selector de monedas múltiples, puede aceptar hasta 6 tipos de monedas diferentes para nuestro proyecto se configuró monedas de 5 centavos, 10 centavos, y 25 centavos. Tanto

Nacionales como Americanas al mismo tiempo. Este tipo de selector de monedas es ampliamente utilizado en sistemas de autogestión, se basa en sensores que principalmente verifican en el material, el peso y el tamaño para identificar monedas.

El CH-926 de la Figura 3.17 es muy estable y preciso, incluso cuando el entorno cambia, como la temperatura, la humedad, etc. Para aumentar la precisión, se sugiere que se utilice diferentes versiones de monedas para la configuración del selector.



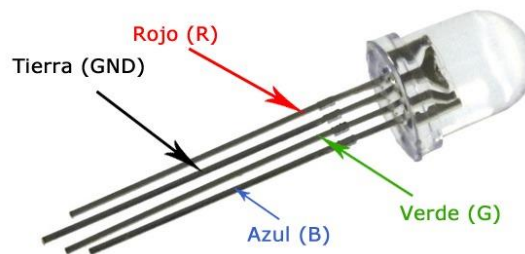
**Figura 3-17:** Selector de Monedas CH-926  
**Fuente:** <http://www.coin-selector.com>

Acepta monedas de un diámetro de: 15mm ~ 32mm y grosor de 1.2mm ~ 3.8mm en nuestro proyecto se usará monedas que se encuentran en ese rango, tiene un tasa de precisión de identificación del 99.5% con una velocidad de transmisión de pulsos de  $\leq 0.6s$ . Trabaja a 12 voltios con un consumo de 65 mA al momento de la selección.

La comunicación entre el Arduino y el selector de monedas se hace en un puerto digital leyendo un tren de pulsos ya que de acuerdo a la configuración de las monedas estas generan un diferente tren de pulsos y estos son interpretados por el puerto digital como interrupciones.

### 3.3.8. Diodos LED de alta luminosidad

Los diodos LED son emisores de luz recubiertos que se usaran para indicar al usuario cual espacio fue el asignado y en qué estado se encuentra ya se libre, ocupado, pagado, o pendiente se utilizará los led RGB para generar estados verdes , rojos o azules de acuerdo al evento suscitado. En la Figura 3.18 se presenta la forma física del Led RGB cátodo común. Para nuestro proyecto se utilizará una matriz de cuatro diodos en un indicador lumínico.



**Figura 3-18:** Led RGB  
**Fuente:** solarbotics.com

### 3.3.9. Pulsadores de Selección

Para manejar el pago de las plazas de estacionamiento es necesario controlarlos a través de pulsadores que están conectados hacia 4 pines digitales del Arduino, cada uno de ellos se configuró como entrada y se deben activar mediante resistencias de *Pull-Up*, llamadas también resistencias de polarización. En la Figura 3.19 se muestra el pulsador a utilizar en nuestro proyecto que incluye una luz interna blanca para una mejor presentación.



**Figura 3-19:** Led RGB  
**Fuente:** dhgate.com

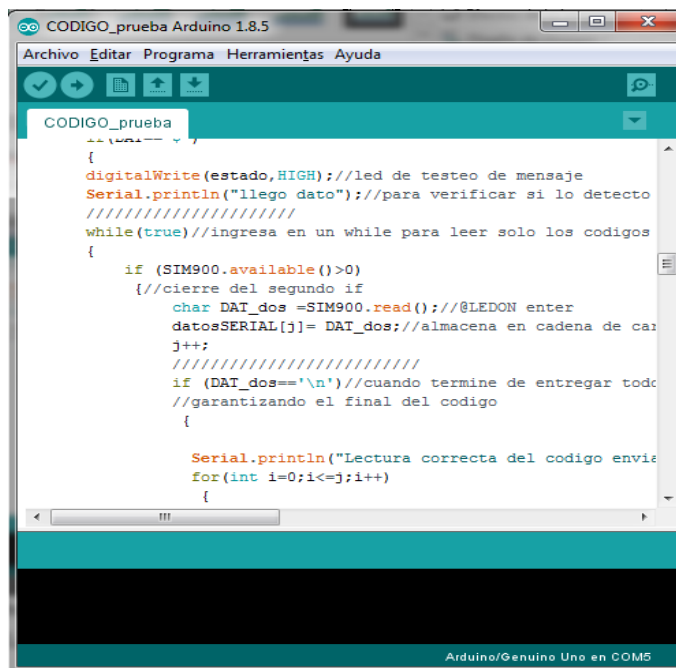
## 3.4. Descripción del *Software* utilizado

### 3.4.1. Software Libre

El software libre entrega al usuario muchas opciones como: libertad de uso, de estudio y modificación, de distribución y redistribución de las versiones modificadas (*Free Software Foundation Europe*, 2016). Adicionalmente existen licencias para software libre que lo garantizan y que le dan una cobertura legal, como por ejemplo la GPL, Licencia Pública General (GNU, 2016).

### 3.4.2. Arduino IDE

Arduino IDE es una interfaz de desarrollo electrónico de código abierto para la fabricación de prototipos asentados en software de código libre. Se basa en lenguaje de programación tipo C++, El microcontrolador de Arduino se programa mediante el lenguaje de programación Arduino basado en *Wiring* y el ambiente de desarrollo se basa en *Processing*. De manera similar, puedes incluir código AVR-C directamente en los programas Arduino. En la Figura 3.20 se muestra la pantalla de desarrollo de programas Arduino.



**Figura 3-20:** Entorno Desarrollo Arduino IDE

**Fuente:** Elaboración propia



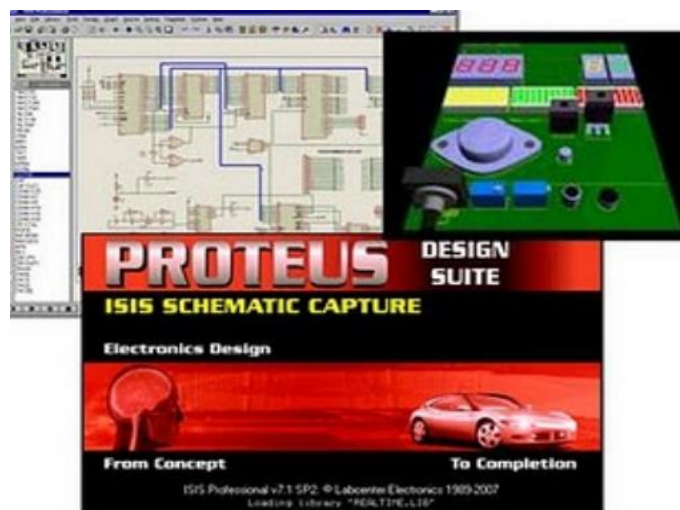
El *software* Arduino IDE se establece en sistemas operativos como Windows, Linux y *Macintosh* OSX. La totalidad de los sistemas para microcontroladores están limitados para uso exclusivo de *Windows*. El ambiente de programación de Arduino es fácil de usar para novatos, pero adecuadamente flexible para que usuarios avanzados puedan aprovecharlo también. (Arduino, 2016).

### 3.4.3. Software de Diseño Electrónico

Entre los programas empleados para el diseño electrónico de circuitos y simulación de los mismos el más usado y de fácil manejo es *Proteus* que establece dos interfaces una para diseños eléctricos y otra para la elaboración de diseños PCB y de bosquejos antes de ser enviados a la manufactura

Se ha elegido *Proteus*, debido a sus varias prestaciones, y a su compatibilidad con el *software* CAM y su uso para la realización de PCBs.

El entorno tanto de ISIS como ARES es amigable con el usuario, contiene las respectivas barras de menú desplegables, barras de herramientas, ventana de componentes, zona de trabajo, barra de control de simulación estas se encuentran distribuidas alrededor de toda la pantalla en la Figura 3.21 se muestra las pantallas de trabajo.



**Figura 3-21:** Pantallas de Trabajo ISIS  
**Fuente:** *Labcenter Electronics*

## **3.5 Diseño Electrónico**

### **3.5.1 Diseño de *Hardware* en *Proteus***

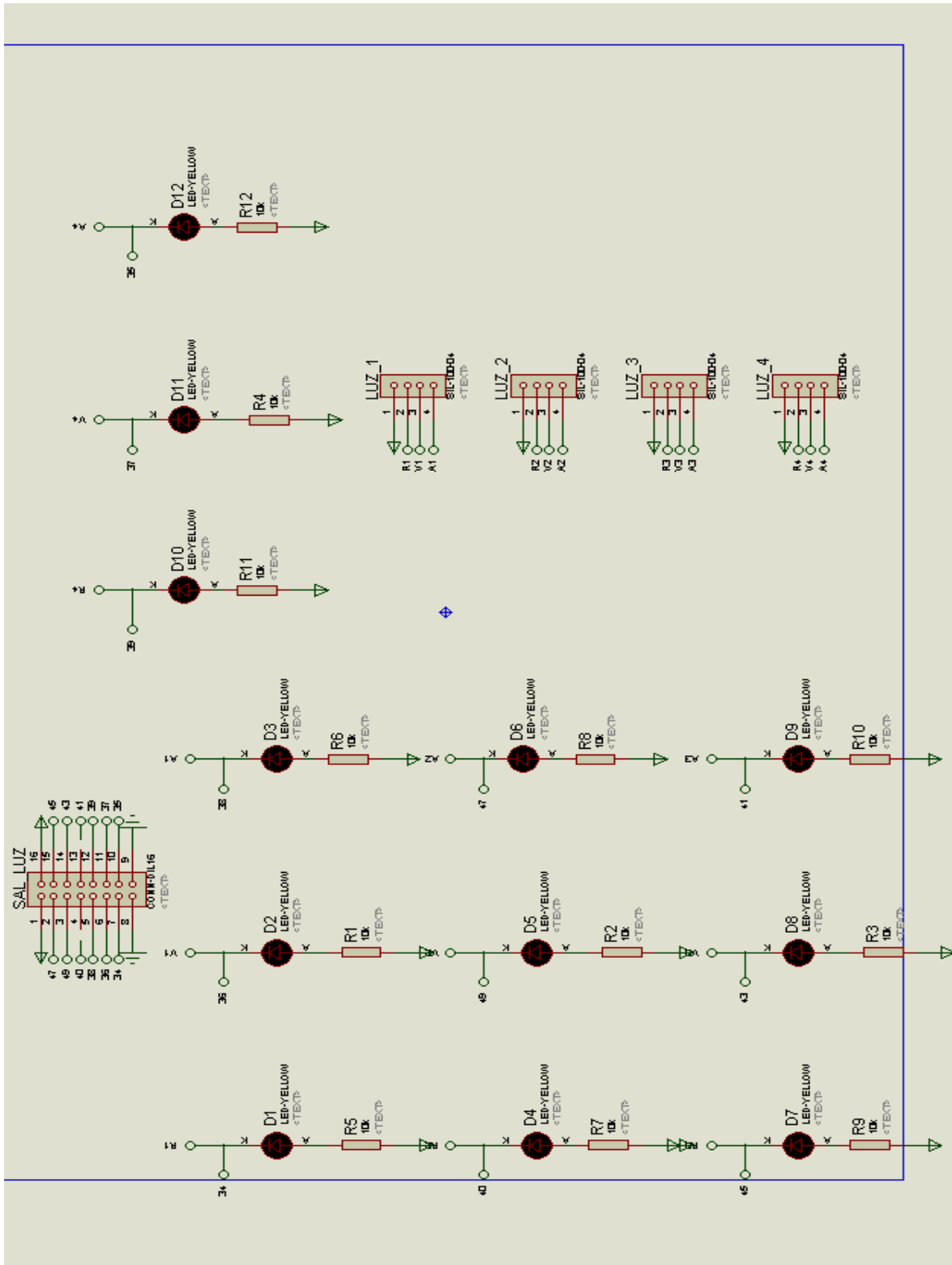
Para la elaboración del parquímetro electrónico y específicamente del módulo de control acoplado al Arduino se realizó una serie ordenada de actividades, en primer lugar se diseñó el diagrama esquemático en el *software Proteus*, el cual también permitió la simulación de eventos en sensores y actuadores, a la par se pudo generar los diagramas electrónicos PCB, luego se realizó la transferencia manual de las pistas *footprint* en una placa de fibra, después se elimina el exceso de cobre de las PCBs y se procedió a realizar las perforaciones en los espacios asignados para finalmente soldar los elementos en la placa de control.

A continuación se indica la elaboración de los diagramas esquemáticos y circuitos electrónicos. Para la generación de dichos esquemas se usaron las herramientas del programa *Proteus*.

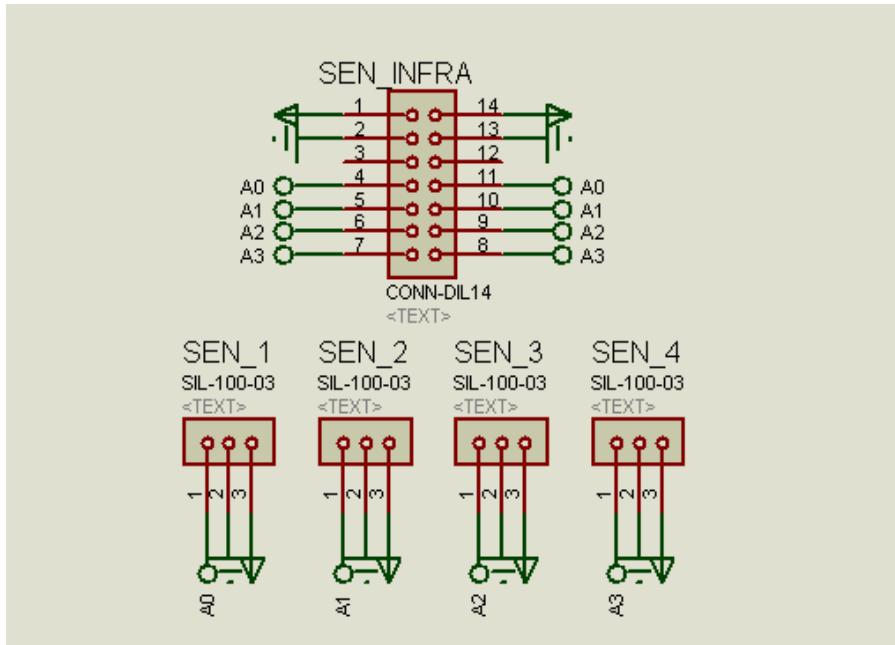
### **3.5.2 Diagrama Esquemático**

Para generar el circuito impreso se elaboró el diagrama esquemático teniendo como referencial principal al controlador Arduino *Mega ADK*, el programa al no disponer de una librería con el *footprint* de dicho controlador se utilizó pines de referencia con similares características y al alrededor del controlador se ubicó elementos que complementan el módulo de control como son el reloj DS3231, el buzzer, los pines correspondientes a los pulsadores, el lector micro SD y conectores auxiliares que realizan la comunicación hacia cada uno de los dispositivos externos como los sensores, las luces visualización y la pantalla GLCD. En la Figura 3.22 se muestra el diagrama esquemático principal del proyecto, en la figura 3.23 y 3.24 se muestra el diagrama de control de luces y de control de sensores respectivamente estas placas esta interconectadas entre sí mediante buses de datos.





**Figura 3-23:** Diagrama esquemático control de luces  
**Fuente:** Elaboración propia



**Figura 3-24:** Diagrama esquemático control Sensores  
**Fuente:** Elaboración propia

Se verificó que cada conexión existente se encuentre correctamente referenciada ya que de ellas dependerá cómo se generen las líneas de conducción en los diagramas. Además se confirmó que existía una sola referencia (tierra), y también que todos los dispositivos que conforman el sistema estén correctamente polarizados.

### 3.6 Diseño de Software

Para la transmisión de datos entre los parquímetros y la PC se estableció un tipo de codificación de la información por cada evento ocurrido en los tótems de cobro, esta se establece mediante comandos AT por el envío de mensajes de texto con una información cifrada en una cadena de datos llamada *String*.

#### 3.6.1 Código para envío de información del Parquímetro

Toda la información que se envía desde los parquímetros hacia el servidor es una cadena denominada *String* que contiene los siguientes datos como son: evento ocurrido, el módulo que genera la acción, en que plaza de estacionamiento se realizó la acción, la hora y fecha en que ocurrió el evento. Y se la representa en bloque que se detalla a continuación.



### Bloque A

Es el primer código de 3 letras que se genera en el *String* sirve para identificar el tipo de mensaje del evento ocurrido para ser enviado desde el parquímetro hacia el servidor central, los códigos pueden ser los descritos en la tabla 3.1.

**Tabla 3.1** Cabecera del *String* de envío

Código	Significado
CSP	Carro sin pago
CPT	Carro se pasó del tiempo
CPP	Carro pago parquímetro
ATM	Abrir tótem puerta externa
ASA	Alerta de sacar alcancía
ACA	Alerta de cambio de alcancía
AAL	Alarma alcancía llena
PPA	Plaza disponible

**Fuente:** Elaboración propia

## Bloque B

Para el siguiente bloque se presenta un código de 3 números que indica la numeración del parquímetro que envió la información al servidor y es válido desde el 0 hasta el 999.

## Bloque C

En el siguiente bloque la información que presenta la cadena del *String* enviada desde el parquímetro se refiere a la plaza de parqueo que genera la alarma. En la tabla 3.2 se muestra los códigos correspondientes a las plazas este proyecto contiene 4 plaza de uso.

**Tabla 3.2** Código selección de plazas

<b>Código</b>	<b>Significado</b>
0	Todo el parquímetro
1	Plaza 1
2	Plaza 2
3	Plaza 3
4	Plaza 4

**Fuente:** Elaboración propia

## Bloque D

En la parte D de la cadena del *String* se presenta el envío de fecha y hora a la que se produjo la alarma esta se presenta en el siguiente formato: año seguido por el mes finalmente el día como se muestra 20180817 estos datos son unidos sin separación de signos o espacios

### 3.6.2 Código para envío de información del Servidor

El servidor al igual que el parquímetro también envía datos necesarios al parquímetro para el correcto funcionamiento. En la tabla 3.3 se muestran los datos necesarios para el envío de una cadena *String* de similares características a lo descrito anteriormente.

**Tabla 3.3** Códigos enviados desde el Servidor

<b>Datos necesarios</b>	<b>Letra clave</b>
Tiempo de gracia	T
Valor establecido por hora	V
Porcentaje o valor de alerta para alcancía llena	P
Horario de funcionamiento del parquímetro	H

**Fuente:** Elaboración propia

Estos datos se envían en una cadena *String* anteponiendo la letra clave. Por ejemplo, si el tiempo de gracia son 5 minutos, valor por hora es de \$0.50 Centavos, el valor de alerta de alcancía llena son \$50 dólares, y la hora de funcionamiento es de 7 A.M a 19 P.M, la cadena *String* envía la información de la siguiente manera.

**T05,V050,P500,H07001900**



### **3.7 Ventajas del Producto**

- Incrementar la disponibilidad espacios para estacionamiento.
- Reducir el tiempo de búsqueda de un lugar de parqueo.
- Reducir el estacionamiento ilegal en ciertos lugares del espacio público, como cruces peatonales, veredas y generar doble vía.
- Disuadir el estacionamiento de larga estancia.
- Reducir el congestionamiento vehicular.
- Ordenar la vía pública.
- Fomentar diferentes formas de movilidad.
- Aligerar la velocidad de vehículos de emergencia.
- Optimizar la seguridad vial.
- Generar recursos que puedan ser reinvertidos en el espacio público, específicamente en la movilidad de la ciudad.
- Permite consultar bases de datos para conocer reportes o anomalías de los vehículos o del módulo en tiempo real.

### **3.8 Análisis de Costos**

Para la construcción de prototipo se utilizó 22 ítems entre componentes y materiales electrónicos necesarios que en total suman la cantidad de 832 dólares que se presenta detallado en la tabla del anexo E.

## **CAPÍTULO IV**

### **IMPLEMENTACIÓN**

#### **4.1 Desarrollo**

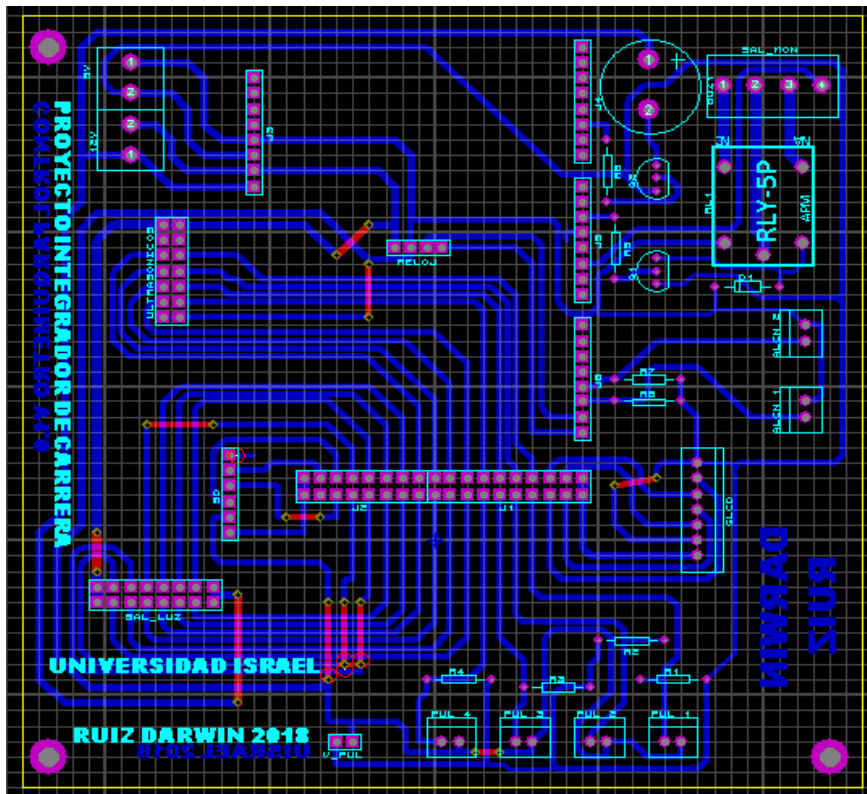
En este capítulo se detalla la implementación del parquímetro electrónico que consiste en la construcción de los PCB, la caja contenedora, la programación completa del sistema electrónico y puesta en marcha del proyecto.

##### **4.1.1 Elaboración del PCB**

Una vez concluido los diagramas esquemáticos se realizó el diseño en el módulo ARES. La primera acción realizada fue posicionar adecuadamente cada uno de los elementos es decir verificar que se encuentren todos los elementos adyacentes entre sí para evitar caminos largo de conexión. Luego se trazó las pistas o rutas de conducción también los ayudamos de la herramienta que dispone ARES con la opción de *Auto-Router*, para que sea el *software* el que automáticamente genere las pistas o rutas de conducción, al final de esta opción siempre queda unos conflictos de conexión que se los solucionó de forma manual generando pequeños puentes de conexión.

Después de que el *software* generó las líneas de conexión, se consideró que las dimensiones de los contactos permitan soldar fácilmente los elementos y se aumentó el tamaño de los *Path* de soldadura. También se recomienda asegurar una distancia apropiada entre una y otra pista, para evitar contactos no deseados que los llamamos cortos. En la Figura 4.1 se presenta el circuito impreso de la placa principal de prototipo.

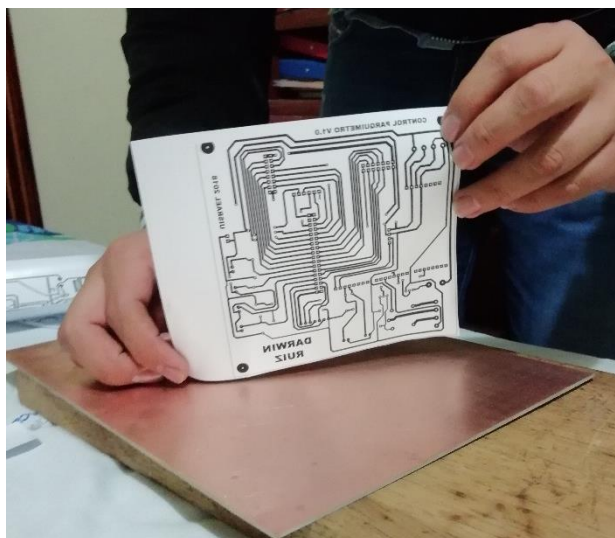
En el anexo D se presenta los 3 diseños del PCB generados individualmente estos PCBs se interconectan entre si físicamente mediante un bus de datos a la placa del controlador Arduino *Mega* ADK.



**Figura 4-1:** Circuito impreso  
**Fuente:** Elaboración propia

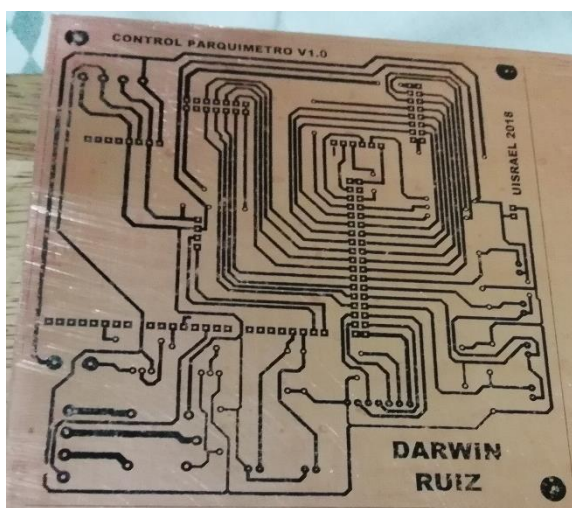
Es conveniente nombrar que para este proyecto se utilizó las librerías de dispositivos que genera *Proteus* pero para ciertos casos puntuales no se obtuvo la información sobre el *footprint*. Por lo tanto, para el desarrollo del proyecto se ha creado librerías adicionales que contienen la información sobre el tamaño y las características de los elementos a usar.

Una vez listos los diagramas electrónicos, se imprimió el circuito en una impresora láser, en una hoja especial de papel transfer y se procedió a realizar la transferencia a una placa de fibra de vidrio con lámina de cobre mediante la técnica del planchado. (Ver Figura 4.2)



**Figura 4-2:** Impresión del diseño y colocación en la Fibra.  
**Fuente:** Elaboración propia

Después de ser transferidos los diagramas electrónicos a la placa como muestra la Figura 4.3 se procede a retirar los excesos de cobre con la técnica del quemado con una solución de Cloruro Férrico (ver Figura 4.4).

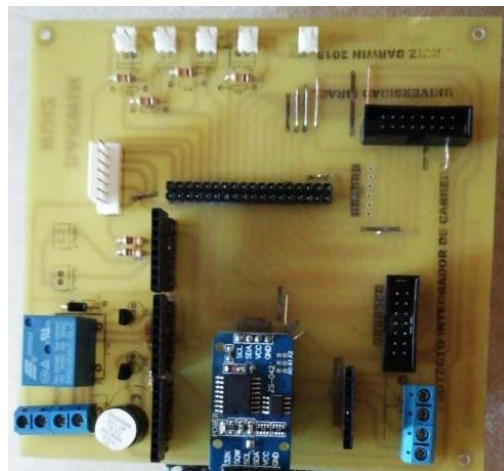


**Figura 4-3:** Transferencia de la impresión en la baquelita  
**Fuente:** Elaboración propia



**Figura 4-4:** Retiro de la impresión de las pistas.  
**Fuente:** Elaboración propia

Luego se procedió a realizar las respectivas perforaciones que permitieron el montaje de cada elemento electrónico para posteriormente soldarlos en la Figura 4.5 se aprecia cómo se montaron los elementos en la placa de fibra.



**Figura 4-5:** Montaje de elementos electrónicos en placa  
**Fuente:** Elaboración propia

Luego del montaje de los elementos que se aprecia en la Figura 4.6, que se conectó del módulo SIM 900 con placa de control y el Arduino Mega ADK con el resto de componentes electrónicos mediante pines de conexión denominados *Headers* hembra.



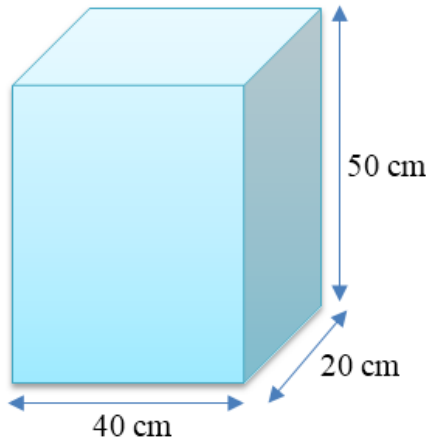
**Figura 4-6:** Módulo de control completo  
**Fuente:** Elaboración propia

## 4.2 Ensamblaje del prototipo parquímetro

El ensamblaje del parquímetro es la primera fase de la construcción previa la realización de las pruebas de funcionamiento.

### 4.2.1 Tótem Metálico

Tomando en cuenta el funcionamiento del equipo, el material de tótem del prototipo es metálico. Y dentro de esta caja se adecuan varios dispositivos. Se dimensiono apropiadamente el tótem del parquímetro, se tomó en cuenta el tamaño de cada uno de los elementos que se incluirá en ella. En la Figura 4.7 se muestran las dimensiones específicas de la caja que se ha elaborado.



**Figura 4-7:** Dimensiones de la Caja Contenedora  
**Fuente:** Elaboración propia

#### 4.2.2 Montaje de equipos

Construido el tótem metálico se realizó los cortes y los agujeros respectivos en la parte frontal del tótem para instalar cada uno de los elementos. La Figura 4.8, muestra la parte delantera de la caja metálica con sus cortes.



**Figura 4-8:** Caja Contenedora  
**Fuente:** Elaboración propia

Luego se procedió a ubicar cada uno de los dispositivos en la caja. En primer lugar se fijó las placas usando postes a la lámina metálica colocada en la parte interior izquierda de la caja contenedora. Del mismo modo se instaló el colector de monedas y los fines de carrera

para la puerta y la alcancía. En la Figura 4.9 se presenta el montaje del PCB y todos los actuadores.



**Figura 4-9:** Montaje de PCB y actuadores

**Fuente:** Elaboración propia

En la parte frontal se instalaron los dispositivos que complementan la solución como son la pantalla GLCD, el monedero electrónico, los selectores, las luces Led indicadoras. Estos dispositivos interactúan con el usuario en el proceso de pago del espacio de parqueo. En la Figura 4.10 se visualiza el montaje de los dispositivos en la parte frontal de la caja metálica.



**Figura 4-10:** Montaje de los dispositivos en la parte delantera

**Fuente:** Elaboración propia



Se instalaron los sensores de presencia en el interior de 4 cajas, uno por caja la cuales se encuentra construido de acrílico y permiten dar la seguridad necesaria al sensor y adecuada colocación en el piso para las pruebas respectivas. En la Figura 4.11 se muestran dichos sensores.



**Figura 4-11:** Instalación de sensores

**Fuente:** Elaboración propia

### **4.3 Pruebas de Funcionamiento**

Una vez realizado el diseño y la implementación de los componentes electrónicos en el dispositivo, se realizaron las pruebas de funcionamiento de todos los módulos y componentes instalados, para comprobar el desempeño del parquímetro.

Se establecieron las siguientes pruebas de valoración del sistema implementado.

- Encendido del prototipo.
- Pruebas de censado de vehículos en estacionamientos.
- Validación en el cambio de coloraciones de las luces
- Verificación de funcionamiento de pantalla GLCD
- Validación funcionamiento del monedero electrónico
- Envío información hacia el servidor.
- Envío información a la página web

Cada una de las pruebas indicadas anteriormente comprende a su vez un cierto número de sub-pruebas. Para la realización de estas se ha previsto la utilización de varias listas de validación.

#### 4.3.1 Encendido del prototipo

Se ha considerado 6 ítems en el lista de validación que se muestra en la tabla 4.1 de esta etapa, esto de acuerdo al número de dispositivos o módulos activos, o a su vez que requieren alimentación son los mismos que se escriben a continuación:

La primera validación que se realizó fue el encendido del módulo SIM900 esto se confirmó ingresando al monitor serial de Arduino y se verificó la cadena *String* de datos de inicialización. Al mismo tiempo se validó el correcto encendido de las luces RGB, se verifico visualizando que se enciendan las luces de forma secuencial y aleatoria en colores verde, azul, rojo, naranja como se programó en el microcontrolador. La siguiente validación que se realizó en el encendido fue que se visualizó en la Pantalla GLCD y se presentó el mensaje de Iniciando.... Como se aprecia en la Figura 4.12 durante unos 20 segundos que se establece la conexión.



**Figura 4-12:** Pantalla GLCD de inicialización

**Fuente:** Elaboración propia

Finalmente la última validación en el encendido de equipos del tótem se observó que las luces integradas en el monedero electrónico se encendieran y evidencio que el *buzzer* emitiera

un sonido de inicialización. Finalmente se comprobó que al ingresar cualquier tipo de moneda el modulo las valido correctamente.

**Tabla 4.1** Lista de validación global del parquímetro

ITEM	PRUEBA DE FUNCIONAMIENTO	FUNCIONAMIENTO	
1	Encendido de pantalla GLCD	√	CORRECTO
2	Encendido Luces Led RGB de forma secuencial	√	CORRECTO
3	Encendido del monedero electrónico	√	CORRECTO
4	Verificación de sensores	√	CORRECTO
5	Verificación de actuadores	√	CORRECTO
6	Pruebas de comunicación	√	CORRECTO

Fuente: Elaboración propia

#### 4.3.2 Ejecución del censado del vehículo

Se comprobó a los sensores de vehículos instalados en los espacios de parqueo, simulando la presencia de vehículos con bancos plásticos de una altura aproximada de 30 cm y de un peso de 1,5 Kg y bancos metálicos de una altura de 50 cm y un peso aproximado de 3 Kg. Se realizaron estas pruebas con estos elementos ya que los vehículos están compuestos principalmente de dichos materiales. En la Figura 4.13 se presenta las pruebas con un banco plástico.



**Figura 4-13:** Verificación sensor plaza1

**Fuente:** Elaboración propia

### **4.3.3 Funcionamiento de los actuadores y luces**

Para las luces RGB se comprobó su cambio de tonalidad de acuerdo al evento ocurrido. En la Figura 4.14 se puede visualizar la activación de la luz roja que mostró que se encuentra un vehículo en la plaza 1 pero no ha pagado por el estacionamiento en dicho caso fue la silla de plástico colocada en la plaza1.



**Figura 4-14:** Luz roja plaza1 ocupada

**Fuente:** Elaboración propia

También se logró visualizar otro evento programado. En la Figura 4.15 se muestra la condición para el evento de plaza ocupada y pagada. Las luces verdes son las que muestran que un vehículo se parqueo y pago por el espacio de estacionamiento.



**Figura 4-15:** Luz roja plaza1 ocupada y pagada

**Fuente:** Elaboración propia

Para la última validación de las luces se comprobó su cambio de tonalidad respectiva. En la Figura 4.16 se muestra la visualización de la plaza 2 disponible es decir no se encuentra ningún vehículo estacionado en dicha lugar y el estado que se mostró es de disponible y las luces se encendieron de color azul como se programó.



**Figura 4-16:** Luz azul plaza 2 libre

**Fuente:** Elaboración propia

#### 4.3.4 Funcionamiento de pantalla GLCD

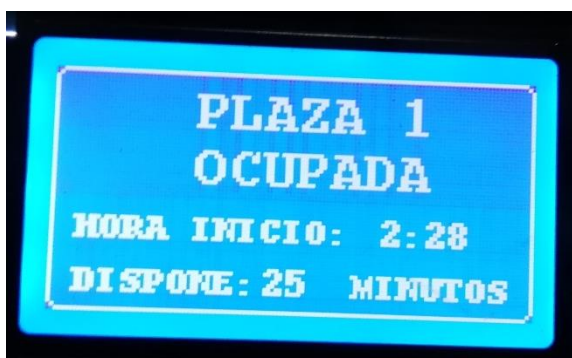
De igual forma que con las luces se realizó la validación de la programación de ciertos formatos en la pantalla GLCD. En la primera verificación se comprobó que se exponga información al usuario sobre la fecha, día y hora en tiempo real. En la Figura 4.17 se verifica que la pantalla muestra dicha información.



**Figura 4-17:** Pantalla GLCD principal

**Fuente:** Elaboración propia

En la segunda validación se evidenció que si un vehículo se estacionó en una plaza específica la pantalla muestra la acción que se efectuó en dicha posición, desplegando la hora en tiempo real de parqueo y el tiempo disponible para estacionarse en dicho espacio. En la Figura 4.18 se confirma la acción ocurrida en la plaza de parqueo 1 con la hora que se estacionó el vehículo y el tiempo que tiene para permanecer en el sitio sin inconvenientes.



**Figura 4-18:** Pantalla GLCD de plaza ocupada

**Fuente:** Propia

Para la tercera validación de la pantalla se confirmó que si no existiese evento alguno de parqueo en los espacios de estacionamiento la pantalla mostró que las plazas se encontraban libres. En la Figura 4.19 se aprecia el mensaje que indica la plaza 4 se encuentra libre o sin presencia vehicular.



**Figura 4-19:** Pantalla GLCD de plaza libre  
**Fuente:** Elaboración propia

#### **4.3.5 Funcionamiento del monedero electrónico**

Para la comprobación del monedero electrónico CH-926 se recolectó valores de pago por los espacios de parqueo y se comprobó la configuración previa de los 5 tipos de monedas como: son monedas de 5 centavos nacionales y americanas, monedas de 10 centavos americanas y de igual manera las monedas de 25 centavos nacionales y americanas. Las monedas no validas fueron rechazadas por la parte inferior del colector. Cada vez que se ingresó una moneda de cualquier denominación valida se constató la información desplegada en el selector. La Figura 4.20 muestra el ingreso de 10 centavos americanos al monedero.



**Figura 4-20:** Validación de 10 cent en el monedero

**Fuente:** Propia

#### 4.3.6 Envió información hacia el servidor

Como se estableció al inicio de este capítulo para la transmisión de datos se realiza de una forma lógica en una trama de datos o *String* en la Figura 4.21 se puede constató el correcto envío de información desde el parquímetro electrónico hacia el servidor se obtuvo la siguiente información: interpretada de la siguiente manera un carro pago el parqueo en el tótem 1 de la plaza 1 el día 17 de Agosto del 2018 un valor de 25 centavos a las 22 horas con un minuto y 25 segundos.



**Figura 4-21:** Envió comandos AT hacia el servidor

**Fuente:** Elaboración propia



### 4.3.7 Envío de la información a la página web

La información en la cadena *String* generada se procesó de manera adecuada en una página web. Esta página se creó desde nuestro servidor ingresando en una dirección IP específica que contiene nuestro módulo Arduino y está conectado a la PC. Para esta acción se confirmó que se está recibiendo la información de forma precisa con los valores y fechas específicas de pago. La Figura 4.22 se muestra que se generó la trama por un valor de pago.

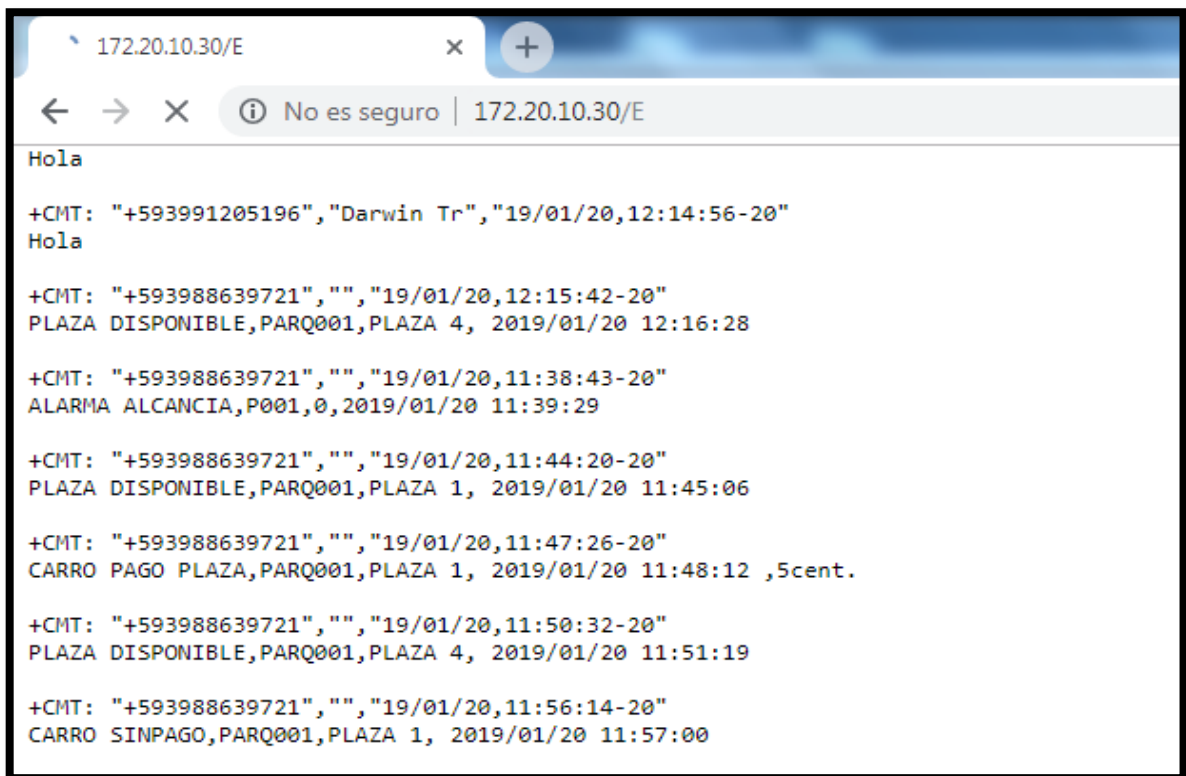


Figura 4-22: Recepción trama en la Web

Fuente: Elaboración propia

#### 4.4 Análisis de resultados

Para comprobar los resultados obtenidos al culminar el proyecto de titulación se creó una serie de instrucciones que nos sirvieron como herramienta de verificación y las llamamos lista de validación.

Para cada proceso se creó una lista de validación diferente que se procede a interpretar. Primero se verificó que los sensores de presencia ultrasónicos confirmen la presencia de vehículos, para esta acción se realizó pruebas con materiales de plástico y metal que son materiales de los cuales están compuestos los vehículos en la tabla 4.2 se puede verificar las acciones programadas para la adecuada validación.

**Tabla 4.2** Lista de validación sensores

ITEM	PRUEBA DE FUNCIONAMIENTO	VALIDACIÓN	
1	Validación con objeto plástico	√	CORRECTO
2	Validación con objeto metálico	√	CORRECTO

**Fuente:** Elaboración propia

La siguiente acción interpretada es la validación de las Luces Led RGB. Para esta tarea se procedió a verificar el cambio de tonalidad descrito en la tabla 4.3 que se explica por cada evento descrito en la lista de validación y mostro los siguientes resultados válidos.

**Tabla 4.3** Lista de validación luces RGB

<b>ITEM</b>	<b>PRUEBA DE FUNCIONAMIENTO</b>	<b>VALIDACIÓN</b>	
<b>1</b>	Verificación de color azul para plaza disponible	√	<b>CORRECTO</b>
<b>2</b>	Verificación de color rojo para plaza ocupada sin pago	√	<b>CORRECTO</b>
<b>3</b>	Verificación de color verde para plaza ocupada con pago	√	<b>CORRECTO</b>

**Fuente:** Elaboración propia

La siguiente acción que se validó fue el funcionamiento de la pantalla GLCD para este caso se generó 4 ítems en su respectiva Lista de validación que indican las acciones a validar. La pantalla desplegó todos los eventos descritos en la en la Tabla 4.4 y nos generó resultados positivos.

**Tabla 4.4** Lista de validación pantalla GLCD

<b>ITEM</b>	<b>PRUEBA DE FUNCIONAMIENTO</b>	<b>VALIDACIÓN</b>	
<b>1</b>	Pantalla de inicialización	√	<b>CORRECTO</b>
<b>2</b>	Pantalla con día, fecha y hora en el sistema	√	<b>CORRECTO</b>
<b>3</b>	Pantallas con plazas libres	√	<b>CORRECTO</b>
<b>4</b>	Pantallas con plazas ocupadas	√	<b>CORRECTO</b>

**Fuente:** Elaboración propia

Para la siguiente validación se realizó las pruebas con el monedero electrónico y se verifico que acepta las monedas que se explicaron en el Capítulo III los resultados en 10 monedas de 10 centavos fue óptimo no generó mayor problema, al igual que con 8 monedas de 5 centavos tanto nacionales como Americanas tampoco genero inconveniente alguno, pero tuvo algunos conflictos con monedas de 25 centavos americanas ya que de 10 monedas ingresadas 5 monedas el monedero electrónico las rechazó ya que para este tipo de monedas existen algunas presentaciones en la versión americana en la tabla 4.5 se aprecian los resultados obtenidos.

**Tabla 4.5** Lista de validación de Monedero Electrónico

ITEM	PRUEBA DE FUNCIONAMIENTO	VALIDACIÓN	
1	Ingreso de valor de 5 cent por 7 min	√	CORRECTO
2	Ingreso de valor de 10 cent por 14 min	√	CORRECTO
3	Ingreso de valor de 25 cent por 35 min	X	INCORRECTO
4	Rechazo de monedas no validas: 1 cent, 1 dólar, 50 cent	√	CORRECTO

Fuente: Elaboración propia

Finalmente la última validación que se realizó fue las pruebas de comunicación con los módulos SIM900 y se verificó que la información se la está procesando de forma correcta desde el parquímetro hacia el servidor y sucesivamente y la tabla 4.6 se presenta la validación y nos muestra resultados favorables.

**Tabla 4.6** Lista de validación del SIM900

<b>ITEM</b>	<b>PRUEBA DE FUNCIONAMIENTO</b>	<b>VALIDACIÓN</b>	
<b>1</b>	Envió de código <i>String</i> desde parquímetro hacia servidor central	√	CORRECTO
<b>2</b>	Envió de código <i>String</i> desde servidor central hacia parquímetro	√	CORRECTO
<b>3</b>	Problema de comunicación	√	CORRECTO

**Fuente:** Elaboración propia

## CONCLUSIONES

- Se diseñó un prototipo para el cobro y control de los sistemas de estacionamientos en zona azul con base microprocesador Arduino MEGA ADK y la comunicación a través de GSM hacia el servidor central.
- Se construyó un tótem de interrelación para que el cliente pueda cancelar y validar los valores por el tiempo de parqueo.
- Se desarrolló un sistema de monitoreo con base en el software Arduino como servidor para la verificación en tiempo real de los eventos ocurridos en los tótems de control.
- El prototipo permite la interacción entre el usuario y el sistema, a través de una pantalla GLCD en la que se presentan opciones al usuario, y mediante unos selectores se puede elegir entre las 4 plaza establecidas.
- Al utilizar los puertos de comunicación I2C, TX-RX, UART, individualmente se debe evitar utilizar los mismos pines ya que hay puertos que funcionan con estos tres tipos de configuraciones y así se evitar conflictos de comunicación.
- Se utilizó la red móvil de CLARO por su mejor cobertura en la ciudad y mejor compatibilidad con el módulo de comunicación SIM 900.
- La pulsación repetida de los botones de selección provoca desgaste en los mismos y provoca que su vida útil no sea muy prolongada.

## RECOMENDACIONES

- En lugar la pantalla GLCD y los selectores que se usan en el prototipo se podría usar una pantalla táctil.
- Si bien se han tomado en cuenta varias consideraciones relativas al funcionamiento a la intemperie del prototipo, se recomienda no exponerlo a condiciones climáticas extremas.
- Colocar un dispositivo para dar el cambio o vuelto por un valor de pago en futuras mejoras.
- Instalar un selector que pueda acoger mayor opciones de monedas como 50 cent, 1 dólar.
- De acuerdo al área de instalación verificar que la red móvil que se vaya a utilizar tenga una buena cobertura.
- Sería conveniente que el precio por estacionamiento incluya el pago de un seguro, que sirva para cubrir de alguna manera con los gastos que se produjeran en caso de ocurrir un delito mientras el auto se encuentra estacionado.
- Crear un programa base para validar por separado el funcionamiento de cada uno de los dispositivos electrónicos a usar.

## BIBLIOGRAFÍA

- Anjanappa, M. (2002). *Introduction to sensor and Actuators*. Maryland Baltimore.
- ARCOTEL. (MAYO de 2018). <http://arcotel.gob.ec/servicio-movil-avanzado-sma/>. Obtenido de <http://arcotel.gob.ec>
- Arduino. (2018). *ArduinoUNO*. Obtenido de <https://www.arduino.cc/en/Guide/ArduinoUNO>
- Consejo Metropolitano de Quito. (2012). *Ordenanza Metropolitana 0221*. Quito.
- Cuerpo de agentes de control Quito. (2018). *Notificación por incumplimiento de Normativa en Zona Azul*. Obtenido de <http://cuerpodeagentesdecontrolquito.gob.ec/gestion/zona-azul/>
- Ecuared. (Mayo de 2018). *Aplicacion WEB*. Obtenido de [http://www.ecured.cu/index.php/Aplicaci%C3%B3n\\_web](http://www.ecured.cu/index.php/Aplicaci%C3%B3n_web)
- El Comercio. (15 de Febrero de 2018). *Guayaquil-estacionamiento-parquimetro-movilidad-atm. Guayaquil ampliará áreas de estacionamiento con parquimetro*.
- Group CSS, W. (Mayo de 2018). <http://www.w3.org/html/>. Obtenido de <http://www.w3.org/html/>
- Herrador, D. (2013). *Sistemas de aparcamiento inteligente aplicado a las Smart Cities*. Madrid.
- INEC. (2016). *ANUARIO DE ESTADISTICAS DE TRANSPORTE*. QUITO.
- Jayavardhana Gubbi, R. B. (Octubre 2011). *Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions*.



Kendall, K. (2005). *Análisis y diseño de sistemas*. Mexico: Pearson Educación.

Miñarro, U. B. (2009). <http://ocw.um.es>. Obtenido de <http://osw.um.es/web/innovacion/plataformas/ocwingenierias/sistemas-embedidos/material-de-clase-1/ssee-t02.pdf>

Puche Lara, J. (2014). *Infraestructura para la comunicación Citizen - Smart City*. Manzanares.

Studio, M. V. (Mayo de 2018). <http://www.microsoft.com/visualstudio/esn> . Obtenido de <http://www.microsoft.com/visualstudio/esn>

SurOnline. (Mayo de 2018). *beneficios-funcionamientoaplicaciones-web*. Obtenido de Sur Online. (s. f.). Que es una aplicación Web. Sur Online. [Rhttp://www.suronline.net/nuevo\\_sitio/beneficios-funcionamientoaplicaciones-web.asp](http://www.suronline.net/nuevo_sitio/beneficios-funcionamientoaplicaciones-web.asp)

Tomasi, W. (2003). *Sistemas de Comunicaciones Electrónicas*. Mexico: Pearson Educación.

Verle. (2009). *Microcontroladores PIC*.

# **ANEXOS**

## ANEXO A (DATASHEET ARDUINO MEGA)



### Product Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

### Index

Technical Specifications	Page 2
How to use Arduino Programming Environment, Basic Tutorials	Page 6
Terms & Conditions	Page 7
Environmental Policies half sqm of green via Impatto Zero®	Page 7



radiospares RADIONICS



# Technical Specification

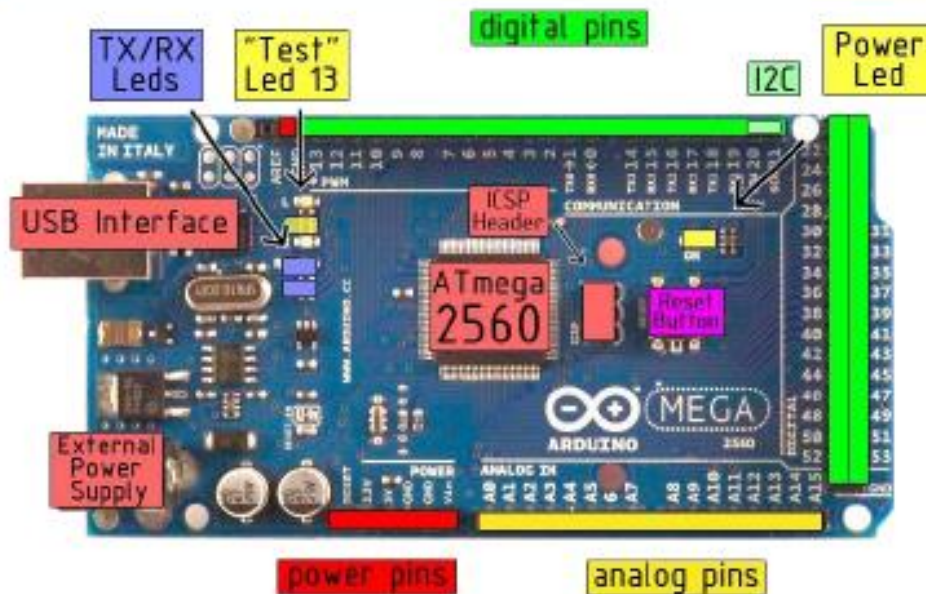


EAGLE files: [arduino-mega2560-reference-design.zip](#) Schematic: [arduino-mega2560-schematic.pdf](#)

## Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

## the board



radiospares RADIONICS



## Power

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

## Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

## Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial:** 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts:** 2 (Interrupt 0), 3 (Interrupt 1), 18 (Interrupt 6), 19 (Interrupt 4), 20 (Interrupt 3), and 21 (Interrupt 2). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM:** 0 to 13. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI:** 60 (MISO), 61 (MOSI), 62 (SS), 63 (SS). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **LED:** 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **PC:** 20 (SDA) and 21 (SCL). Support PC (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the PC pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and [analogReference\(\)](#) function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.



radiospares

RADIONICS



## Power

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega's digital pins.

The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation on the Wiring website](#) for details. To use the SPI communication, please see the ATmega2560 datasheet.

## Programming

The Arduino Mega2560 can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.



*radiospares*

**RADIONICS**



## Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

## USB Overcurrent Protection

The Arduino Mega has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

## Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila. Please note that I<sup>2</sup>C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).



*radiospares* **RADIONICS**



# How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

**Linux Install**

**Windows Install**

**Mac Install**

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

## Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>  
Arduino-0017>Examples>  
Digital>Blink**

Once you have your sketch you'll see something very close to the screenshot on the right.

In **Tools>Board** select MEGA

Now you have to go to **Tools>SerialPort** and select the right serial port, the one arduino is attached to.

```
Sketch - Arduino IDE
File Edit Sketch Tools Help
Sketch
int ledPin = 13; // LED connected to digital pin 13
// The setup() method runs once when the sketch starts
void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}
// the loop() method runs over and over again,
// as long as the Arduino has power
void loop() {
  digitalWrite(ledPin, HIGH); // set the LED on
  delay(1000); // wait 1000ms (1 second)
  digitalWrite(ledPin, LOW); // set the LED off
  delay(1000); // wait 1000ms (1 second)
}
```

Press Compile button (to check for errors) Done compiling

Upload

TX RX Flashing

Blinking Led!

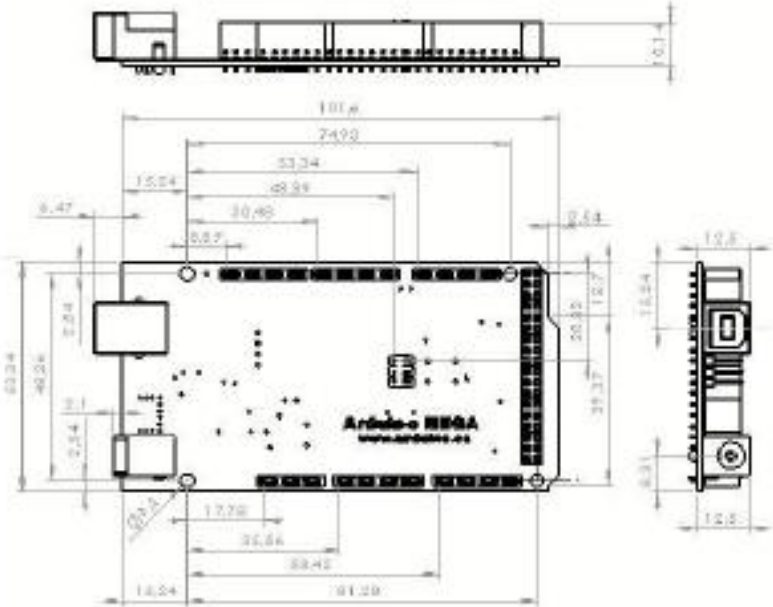
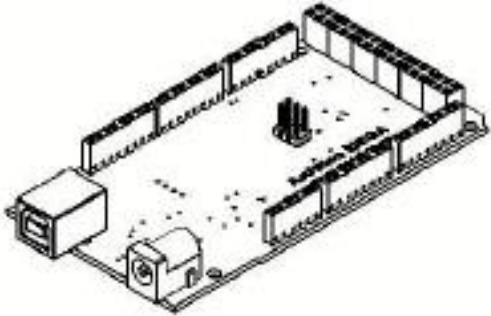


radiospares

RADIONICS







# Terms & Conditions



## 1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

1.4 Customer agree that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not extend or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino<sup>TM</sup> products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino<sup>TM</sup> products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino<sup>TM</sup> products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino<sup>TM</sup> products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

## 2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

## 3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

## 4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Matarale is subject to change without notice. Do not finalize a design with this information.



## Environmental Policies



The producer of Arduino<sup>TM</sup> has joined the Impatto Zero<sup>®</sup> policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.



radiospares

RADIONICS



## ANEXO B (DATASHEET SENSOR HC-SR04)



### SENSOR ULTRASONIDO

Los sensores ultrasónicos usan sonar para determinar la distancia de un objeto como los murciélagos o delfines lo hacen, ofrecen un excelente rango de detección sin contacto, su operación no es afectada por la luz del sol o materiales oscuros, aunque materiales acústicamente blandos son difíciles de detectar.

#### Características:

Voltaje de alimentación: +5VDC

Corriente en espera: <2mA

Corriente de trabajo: 15mA

Ángulo eficaz: <15°

Rangos de distancia: 2cm a 400cm ó 1" a 13ft

Resolución: 0.3cm

Ángulo de medida: 30°

Ancho de pulso de disparo (Trigger Input Pulse Width): 10us se aplica pulso de (10us en Trig)

Eco (Echo): salida del sensor.

Frecuencia de ultrasonido: 40KHz

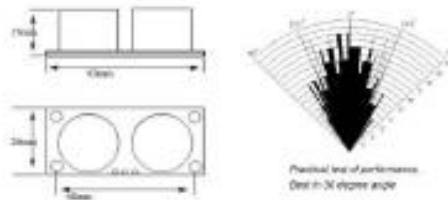
PINES DE CONEXIÓN	
	<p>VCC = +5VDC</p> <p>Trig = Trigger input of Sensor, entrada del pulso de disparo para iniciar la medición.</p> <p>Echo = Echo output of Sensor, pin de eco, que retorna un pulso proporcional a la distancia que rebota el sonido.</p> <p>GND = GND</p>

E-Mail: [info@microkitselectronica.com](mailto:info@microkitselectronica.com) - [www.microkitselectronica.com](http://www.microkitselectronica.com)

Calle: 41 No. 51 13, local 141, Teléfono: 232 77 84,

Centro Comercial Paseo Bolívar, Carabobo con los Huesos, Medellín, Antioquia, Colombia.

### Dimensiones y ángulos de desempeño:



### Operación

Para iniciar la medición el pin Trig, el de la señal de disparo (trigger) debe recibir un pulso de cinco voltios por al menos diez micro segundos, esto indicará al sensor que transmita una ráfaga de ocho ciclos ultrasónicos a 40KHz y espere por la ráfaga reflejada cuando el sensor detecta la señal de ultrasonido en el receptor enviará una señal en alto (5v) por el pin Echo, esta señal tendrá un retardo (ancho) proporcional a la distancia, así que para obtener la distancia se debe medir el Ton (tiempo en alto en el pin Echo)



Tiempo que tarda la señal en volver es igual al ancho del pulso en el pin Echo en microsegundos, para determinar la distancia a partir de este tiempo:

$$\text{Distancia en centímetros} = \frac{\text{Tiempo}}{58}$$

$$\text{Distancia en pulgadas} = \frac{\text{Tiempo}}{148}$$

O se puede utilizar la velocidad del sonido, la cual es 340m/s

E-Mail: [info@microkitselectronica.com](mailto:info@microkitselectronica.com) - [www.microkitselectronica.com](http://www.microkitselectronica.com)

Calle: 41 No. 51 15, local 141, Teléfono: 232 77 84,

Centro Comercial Paseo Bolívar, Carabobo con los Huesos, Medellín, Antioquia, Colombia.

## Programa de aplicación con Arduino UNO r3 y Arduino Duemilanove

### Usando la velocidad del sonido.

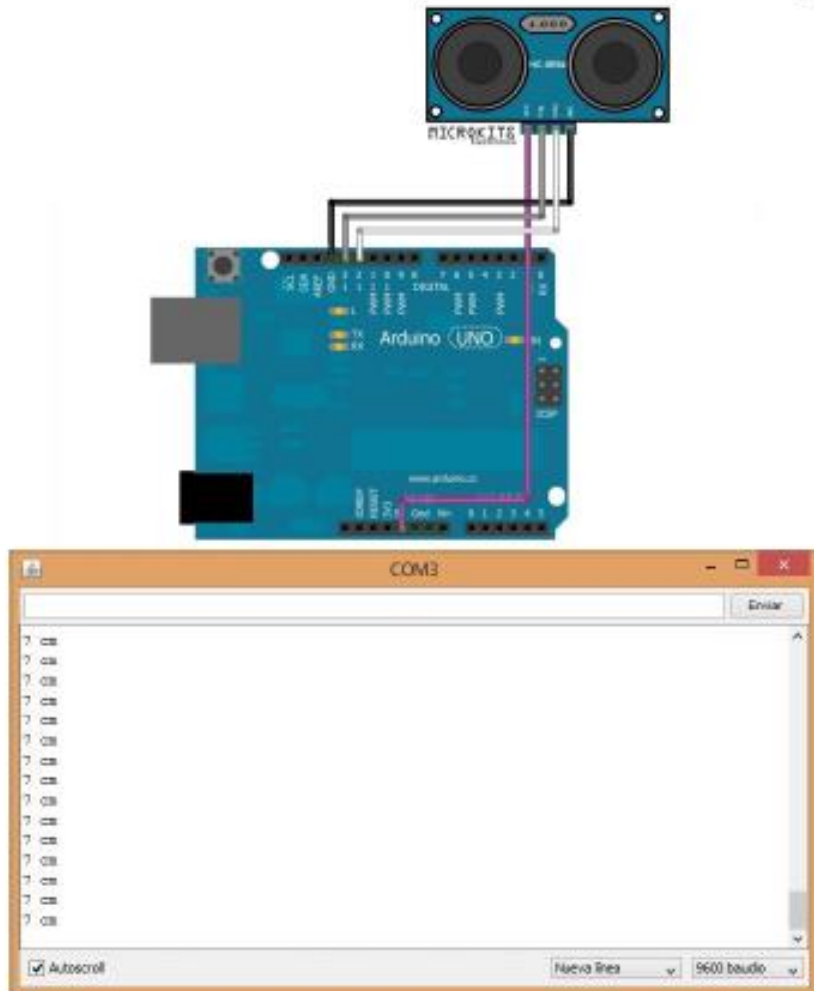
Este programa usa la constante de la velocidad del sonido 340m/s para calcular la distancia recorrida por éste, desde que sale del sensor hasta que rebota y envía los datos por el puerto serie para ser visualizado en el monitor serial del Arduino IDE, la conexión del sensor al Arduino se muestra en la última figura.

```
/* Programa para modulo ultrasonido.  
Usa la velocidad del sonido en el aire a temperatura  
ambiente 340m/s (0.034 cm/us) para completar la ecuación  
que define la distancia entre el sensor y un objeto  
capaz de rebotar la onda de sonido.  
  
Creado: Octubre del 2013  
por William Galvis.  
  
Este código de ejemplo es de dominio público  
http://www.microkitselectronica.com/images/pdf/ultrasonido.pdf  
*/  
  
#define trigPin 12  
#define echoPin 13  
  
void setup() {  
  Serial.begin(9600);  
  pinMode(trigPin, OUTPUT);  
  pinMode(echoPin, INPUT);  
}  
  
void loop() {  
  int duracion, distancia;  
  digitalWrite(trigPin, HIGH); // pulso de disparo  
  delayMicroseconds(1000); // dura 1000 microsegundos  
  digitalWrite(trigPin, LOW); // y se apaga  
  duracion=pulseIn(echoPin, HIGH); // se mide el tiempo del pin echo en alto (us)  
  distancia=(duracion/ 2)*0.034; // distancia será la mitad del tiempo por velocidad  
  Serial.print(distancia); // se envían los datos por serial  
  Serial.print(" cm \n"); // se escribe la unidad de medida y un caracter de nueva línea  
  delay(100); // envía datos cada 100 ms  
}
```

E-Mail: [info@microkitselectronica.com](mailto:info@microkitselectronica.com) - [www.microkitselectronica.com](http://www.microkitselectronica.com)

Calle: 41 No. 51 15, local 141, Teléfono: 232 77 84,

Centro Comercial Paseo Bolívar, Carabobo con los Huesos, Medellín, Antioquia, Colombia.



E-Mail: [info@microkitselectronica.com](mailto:info@microkitselectronica.com) - [www.microkitselectronica.com](http://www.microkitselectronica.com)

Calle: 41 No. 51 15, local 141, Teléfono: 232 77 84,  
Centro Comercial Paseo Bolívar, Carabobo con los Huesos, Medellín, Antioquia, Colombia,

## ANEXO C (MANUAL MONEDERO CH-926)

# Manual of CH-926

CH-926 is a multi coin selector, can accept up to 6 kinds of different coins at the same time. This type of coin selector is widely used in Vending machine, Arcade Game, Message chair, and other self-management system. CH-926 is mainly based on material, weight and size to identify coins. We use the most up to date algorithm to design software. Therefore, CH-926 is very stable and accurate even when environment changes such as temperature, and humidity etc... In order to increase the accuracy, we suggest different version of coins use different channel to set up.

## Specifications

Coin diameter : 15mm-32mm	Atmospheric pressure : 86Kpa—106Kpa
Coin thickness : 1.2mm-3.8mm	Working humidity : $\leq 95\%$
Working voltage : DC +12V $\pm 10\%$	Speed : $\leq 0.6s$
Working current : 65mA $\pm 5\%$	Accuracy rate of identification : 99.5%
Signal output : pulse	

## Features

- a. Capable of accepting all worldwide Coins and Tokens.
- b. Intelligent CPU software control, and high accuracy .
- c. Self-programming without PC.
- d. Accept 1-6 different kinds of coins at the same time.
- e. Free to set up pulses' output.
- f. Prevent not only electric shock but also electromagnetic interference.
- g. Automatic self-test for problems.

# Manual of CH-926

## The Process of Setup for Parameters

1. Press the "Add" and "minus" buttons at the same time for about three seconds, then the letter "A" will appear from the LED display.
2. Press the "setup" button once, and the letter "E" will appear. Next, use the buttons to choose how many kinds of coins you would like to use; then press the "setup" button again to finish.
3. The letter "H" will appear after pressing the button. Use the "Add" and "minus" buttons to choose how many samples you would like to insert later. Next press the "setup" button again to finish.
4. The letter "I" will appear after pressing the button. Use the "Add" and "minus" buttons to choose the amount of output's signals/pulses you want. The quantity limited is 50 times. Next, press the "setup" button to finish.
5. The letter "F" will appear after pressing the button. Use the "Add" and "minus" buttons to choose accuracy. The value is from 1-30, and 1 is the most accurate. Normally, 5-10 will be fine. Next, press the "setup" button to finish.
6. So far, you have successfully set up the first coin. please repeat all above procedures until you have set up all the coins. The letter "A" will appear again after all above procedures are finished.
7. Press the "setup" button, and the letter "E" will appear. Finally, turn off and turn on the power. The setup will be stored.

---

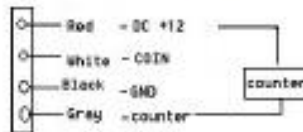
You can start sampling after the setup is finished. Please choose at least 20 coins. The sampling process will affect the accuracy of coin selector.

---

## Sampling

1. Press the "setup" button, then letter "A" will appear from the LED display.
2. Press the "setup" button again, then letter "A1" will appear. Next, start to insert sample coins. The LED display will show how many coins you insert. The letter "A1" will appear again after finished.
3. Press the "setup" button again, then the letter "A2" will appear. Next, Start to sample 2nd coin, and repeat No. 1 and No. 2 until all the coins are set up.
4. After finished the sampling, press the "setup" button. The letter "A" will appear, then turn off and turn on the power. Now you can start to use it.

## Connections' manual



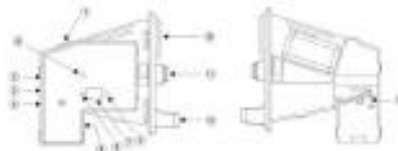


# Manual of CH-926

## FAQ:

- A. Coins can't pass through**
1. Check if the connection is correct or poor connection.
  2. Check if sampling is appropriate.
  3. Check if there is something stuck inside.
- B. Can't identify outputs**
1. Check if 2 way switches -NO, NC, - are correct.
  2. Check if signal wire is connected correctly.
  3. Check if there is pull-up resistor with signal wire(not allowed).
  4. Check if there is something stuck on the channel where the coins pass out.
- C. Coins can not pass smoothly**
1. Check if the parameter is set up appropriately.
  2. Check the accuracy switch.
  3. Check if there is something stuck on the channel where the coins pass out.

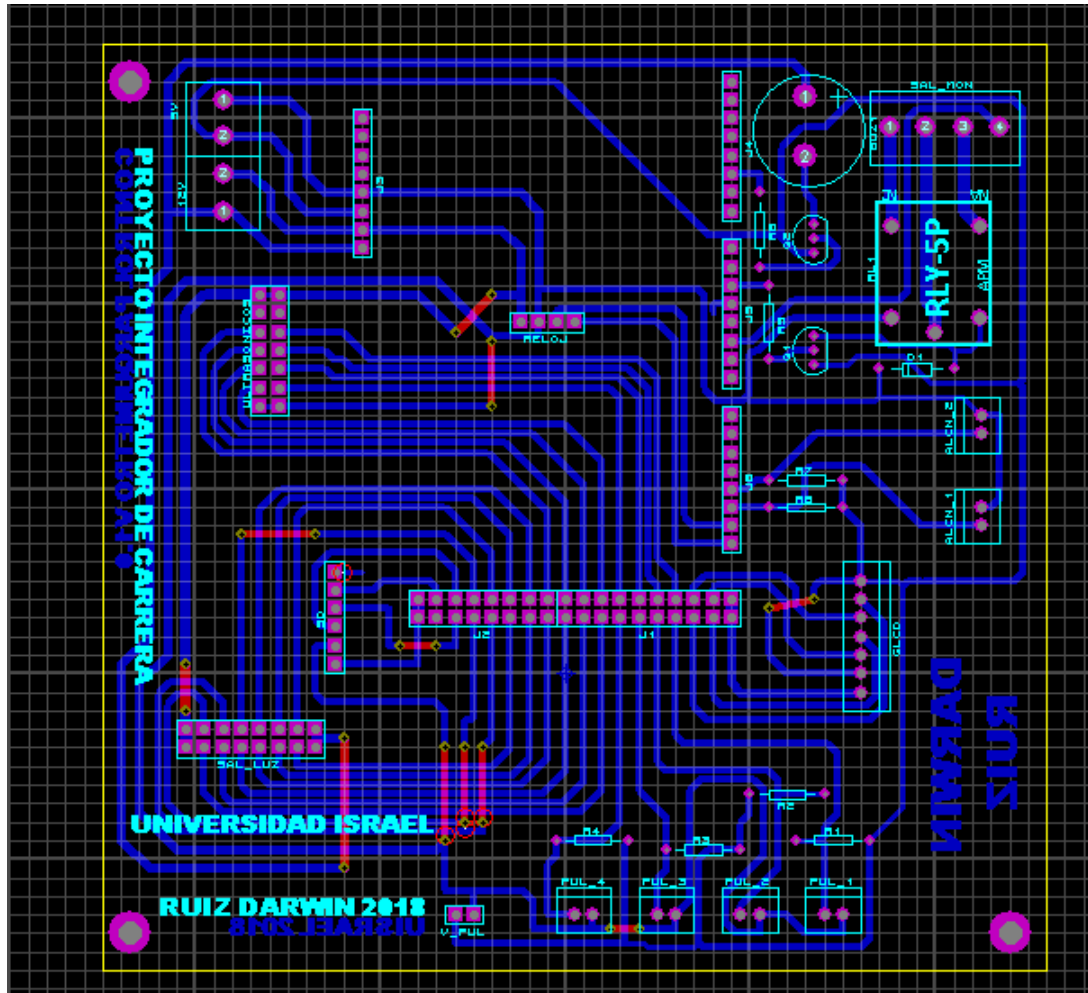
### Name and Function



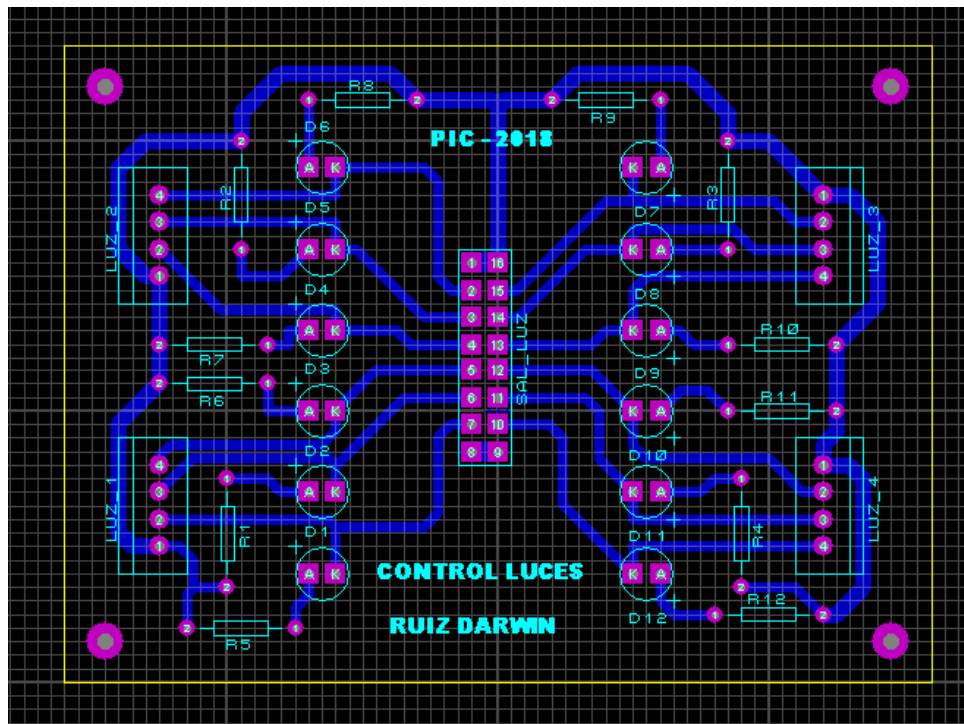
- ① LED light - The light of indication.
- ② 4pin Socket
- ③ Two way switch - signal output NO. or NC.
- ④ Three way switch - Output signal - fast: 20ms , medium: 50ms , slow: 100ms
- ⑤ 2pin Socket - Electromagnetic valve DC 12V
- ⑥ "Setup" button
- ⑦ LED display
- ⑧ "Add" button - Plus "+" value
- ⑨ "Minus" button - Minus "-" value
- ⑩ Coin slot
- ⑪ Press-button for removal of blocked coin
- ⑫ Return slot of Coin
- ⑬ Position of electromagnetic strobe

# ANEXO D (PLACAS ELECTRÓNICAS PARQUÍMETRO)

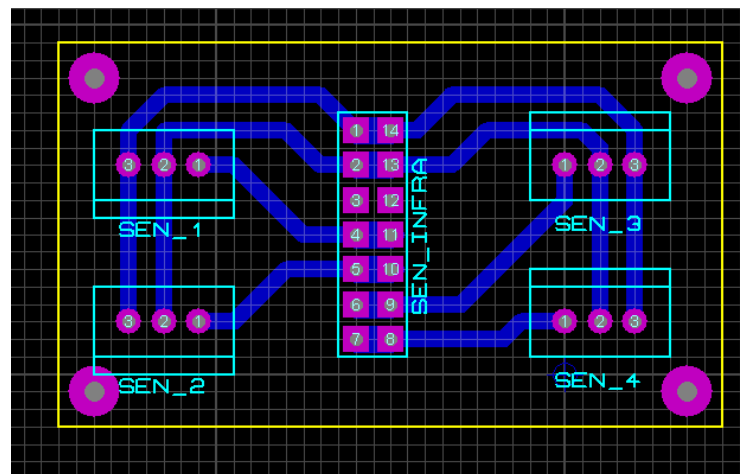
## Placa Control Principal



## Placa control luces



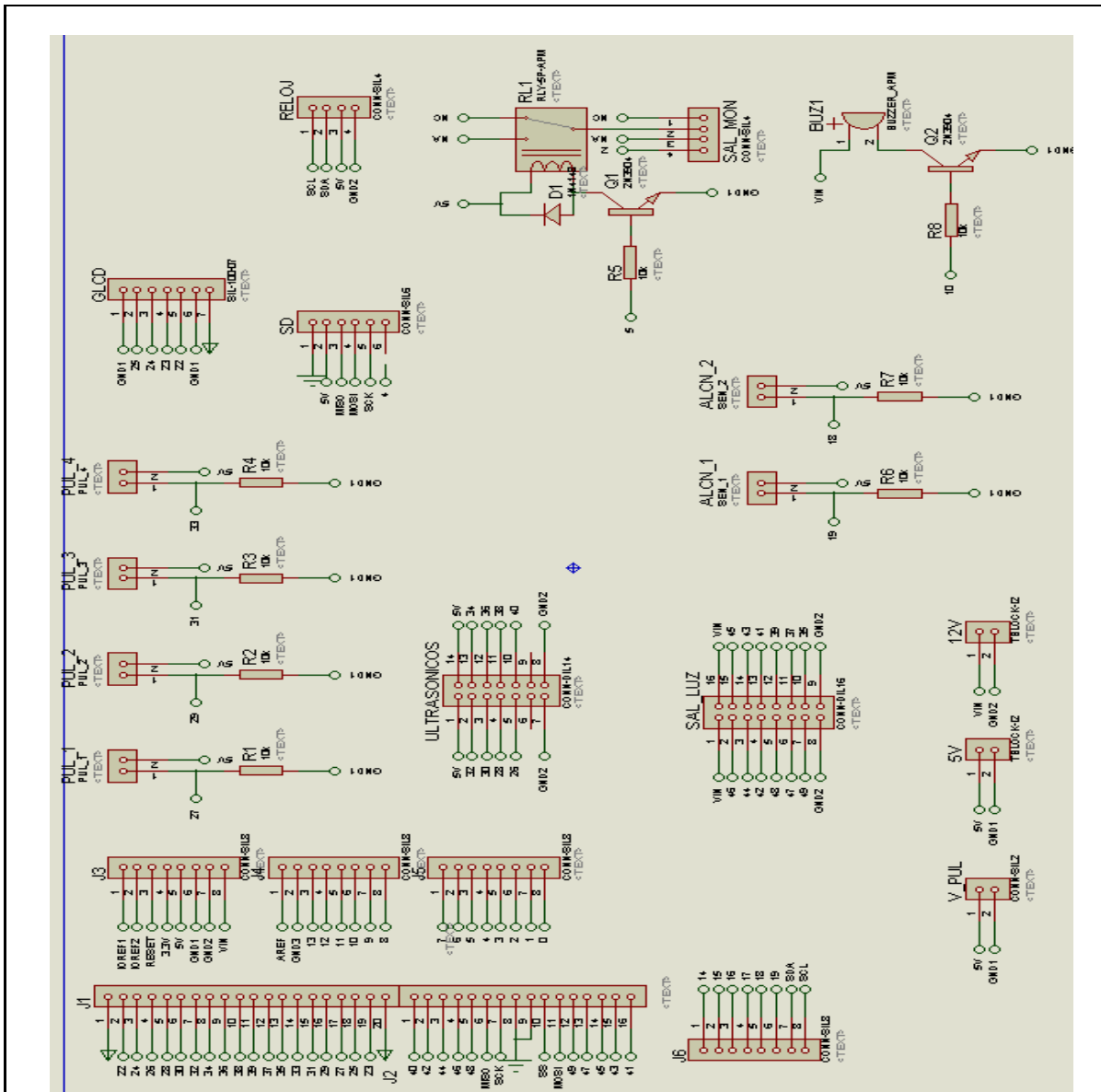
## Placa control sensores



## ANEXO E (TABLA DE COSTOS DEL PROTOTIPO)

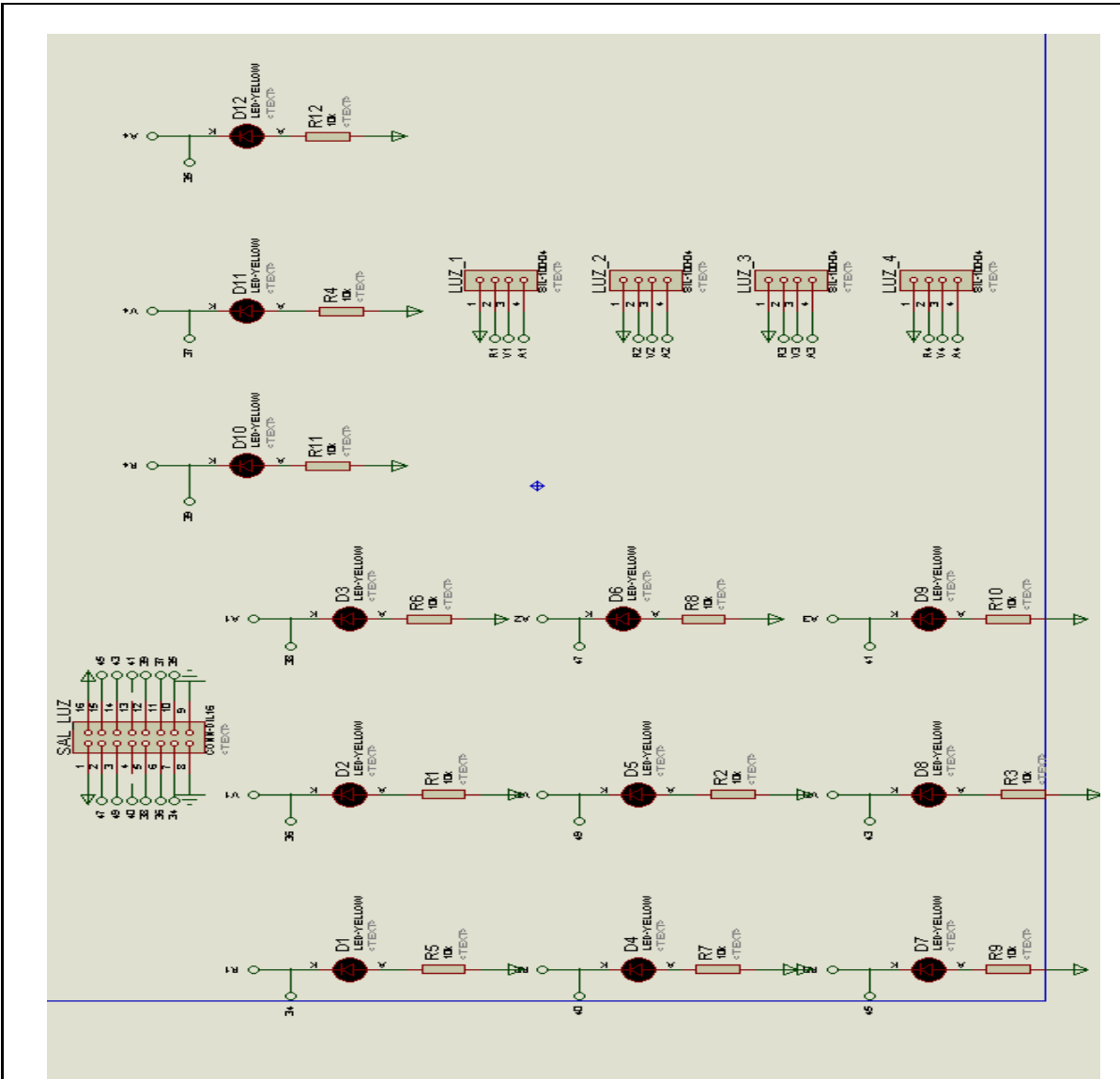
Materiales / Elementos	Cantidad	Valor unitario	Valor Total
Arduino Mega ADK	2	\$ 80,00	\$ 160,00
Shield Sim900	2	\$ 60,00	\$ 120,00
Shield Ethernet	1	\$ 40,00	\$ 40,00
Pantalla GLCD	1	\$ 30,00	\$ 30,00
Luces RGB	4	\$ 7,00	\$ 28,00
Botones Selectores	4	\$ 4,00	\$ 16,00
Fines de Carrera	2	\$ 2,00	\$ 4,00
Monedero Electrónico	1	\$ 65,00	\$ 65,00
Módulo DS3132	1	\$ 4,50	\$ 4,50
Módulo Micro SD	1	\$ 2,50	\$ 2,50
Sensores Ultrasónicos	4	\$ 4,50	\$ 18,00
Fibra de vidrio 30X20	1	\$ 4,00	\$ 4,00
Resistencias	20	\$ 0,05	\$ 1,00
Transistores	5	\$ 0,15	\$ 0,75
Pines de conexión	10	\$ 0,40	\$ 4,00
Cable UTP	30	\$ 0,70	\$ 21,00
Cajas plásticas	5	\$ 2,50	\$ 12,50
Tiempo desarrollo	1	\$ 200,00	\$ 200,00
Estructura Metálica	1	\$ 100,00	\$ 100,00
<b>TOTAL</b>			<b>\$ 832,00</b>

## ANEXO F (DIAGRAMAS ELECTRÓNICOS)




### UNIVERSIDAD TECNOLÓGICA ISRAEL

<b>ESC . 1.1</b>	<b>Observación:</b> Diseño en Proteus	<b>Fecha:</b> 29-02-2019	<b>Cuadro de Referencia:</b> Diagrama Esquemático
	<b>Autor:</b> Darwin Ruiz G.		<b>Tutor:</b> PhD. Alfonso Zozaya
	<b>DIAGRAMA ESQUEMATICO TARJETA PRINCIPAL DE CONTROL DEL PARQUÍMETRO ELECTRÓNICO</b>		



**UNIVERSIDAD TECNOLÓGICA ISRAEL**

	<b>ESC 1.1</b>	<b>Observación:</b> Diseño en Proteus	<b>Fecha:</b> 29-02-2019	<b>Cuadro de Referencia:</b> Diagrama Esquemático
	<b>Autor:</b> Darwin Ruiz G.			
<b>DIAGRAMA ESQUEMATICO          CONTROL DE LUCES DEL          PARQUÍMETRO ELECTRÓNICO</b>				

ANEXO F (MANUAL USUARIO PARQUÍMETRO)

## MANUAL DE USUARIO

PROTOTIPO PARA EL CONTROL DE PLAZAS VEHICULARES PARA LA ZONA AZUL DEL SECTOR DE LA MARISCAL EN LA CIUDAD DE QUITO “PARQUIMETRO ELECTRÓNICO”.



QUITO, ENERO 2019

## INDICE

<b>1.INTRODUCCIÓN</b> .....	3
<b>2.PARTES DEL TÓTEM</b> .....	4
2.1 Pantalla Visualización Principal .....	4
2.2 Selectores de Plazas .....	4
2.3 Luces Led de alarmas a eventos.....	5
2.4 Monedero Electrónico.....	6
<b>3.MODOS DE USO DEL PARQUÍMETRO ELECTRÓNICO</b> .....	6
3.1 Plaza Libre .....	6
3.2 Plaza ocupada y sin pago .....	7
3.3 Plaza ocupada y pagada .....	8
3.4 Monedas a ingresar .....	9

## ÍNDICE DE FIGURAS

<b>Figura 1:</b> Vista frontal Tótem .....	3
<b>Figura 2:</b> Vista pantalla GLCD.....	4
<b>Figura 3:</b> Ubicación de Pulsadores .....	5
<b>Figura 4:</b> Ubicación de Luces Led.....	5
<b>Figura 5:</b> Ubicación del monedero .....	6
<b>Figura 6:</b> Caso Plaza Libre .....	7
<b>Figura 7:</b> Caso Plaza Ocupada y sin pago .....	8
<b>Figura 8:</b> Caso Plaza Ocupada y sin pago .....	9
<b>Figura 9:</b> Monedas Validas.....	9



El presente documento contiene la información necesaria para el uso y operación del “Parquímetro Electrónico”. Se recomienda leer este manual para una correcta manipulación.

## 1. INTRODUCCIÓN

EL parquímetro electrónico se construyó en una estructura metálica para soportar condiciones climáticas externas, también tiene la seguridad necesaria para no ser afectado por factores externos o vandálicos.

El control de plazas disponibles para la Zona Azul “Parquímetro Electrónico” es un tótem de interacción para el pago por las estaciones de parqueo. Este establece comunicación con un web server para conocer los eventos sucedidos en dicha estación de pago.

El tótem de control presenta 4 luces Led indicadoras, 4 botones de selección iluminados, una pantalla GLCD y un monedero electrónico los cuales permiten el control, verificación y pago los espacios de parqueo. En la Figura 1 se muestra la parte frontal del Tótem del Parquímetro Electrónico.



**Figura 23:** Vista frontal Tótem

**Fuente:** Elaboración propia

## 2. PARTES DEL TÓTEM

### 2.1 Pantalla Visualización Principal

En la parte superior izquierda del tótem se presenta una pantalla GLCD de color azul en la que se despliega diferentes mensajes, como la hora, día, fecha en tiempo real, en la figura 2 se presenta la pantalla principal del parquímetro electrónico. Cada 3 segundos las pantallas cambian de acuerdo al evento sucedido.



**Figura 24:** Vista pantalla GLCD

**Fuente:** Elaboración propia

### 2.2 Selectores de Plazas

En la parte central del módulo se encuentran 4 botones con luces Led blancas que permitirán la selección de la plaza de parqueo, estos se encuentran rotulados para la correcta selección del espacio de estacionamiento en la figura 3 se muestra la ubicación de los electores.



**Figura 25:** Ubicación de Pulsadores

**Fuente:** Elaboración propia

### 2.3 Luces Led de alarmas a eventos

El módulo de control consta de 4 luces Led de colores Rojo, Azul y verde que indican los eventos sucedidos en las plazas de parqueo. En la figura 4 se presenta dichas luces que se encuentran en el lado derecho del Tótem.



**Figura 26:** Ubicación de Luces Led

**Fuente:** Elaboración propia

## 2.4 Monedero Electrónico

Consta también de un monedero electrónico para la validación de monedas al ingresar los valores por el pago del espacio de parqueo. Se aceptan 5 tipos de monedas como son 5 cent. Nacionales y Americanos, 10 cent. Americanos y 25 cent. Nacionales y todas las versiones de Americanos. Si las monedas no corresponden a estos tipos el selector las rechaza automáticamente y las devuelve al cliente. En la figura 5 se muestra la ubicación de dicho monedero en el módulo de control.



Figura 27: Ubicación del monedero

Fuente: Elaboración propia

## 3. MODOS DE USO DEL PARQUÍMETRO ELECTRÓNICO

### 3.1 Plaza Libre

Este mensaje se despliega en la pantalla GLCD siempre y cuando el sensor no detecte presencia vehicular en cualquier espacio de parqueo, de igual manera la luz led indicadora se mantiene azul ya que el espacio se encuentra disponible. Al pulsar el selector 1 nos desplegará este mensaje. En la figura 6 se presenta la pantalla y el color de la luz indicadora. Este mensaje

puede aparecer el cualquiera de los 4 espacios de parqueo siempre que no existiera algún vehículo sobre el sensor.



**Figura 28:** Caso Plaza Libre

**Fuente:** Elaboración propia

### **3.2 Plaza ocupada y sin pago**

Cuando un vehículo se encuentra ocupando una plaza de parqueo inmediatamente el sensor detecta la presencia vehicular y envía una alerta de evento a la pantalla GLCD de plaza ocupada con la hora de inicio de parqueo y los minutos que dispone para estacionarse, la luces indicadoras tomaran el color Rojo que representa que la plaza se encuentra ocupada y sin pago alguno en la figura 7 se presenta un caso puntual que un vehículo ocupa la plaza 2 con una hora de inicio de 00:13 y se encuentra estacionado un minuto sin haber cancelado algún valor por parquearse. Al presionar el selector 2 siempre mostrará dicho mensaje hasta que se ingrese algún valor.



**Figura 29:** Caso Plaza Ocupada y sin pago

**Fuente:** Elaboración propia

### **3.3 Plaza ocupada y pagada**

Como en el caso anterior al estacionarse un vehículo y ocupar una plaza de parqueo inmediatamente el sensor detecta la presencia vehicular y envía una alerta de evento a la pantalla GLCD de plaza ocupada con la hora de inicio de parqueo y los minutos que dispone para estacionarse, la luces indicadoras tomaran el color Rojo que representa que la plaza se encuentra ocupada y sin pago alguno pero en este caso el cliente fue al Tótem y cancelo un valor de 10 cent, selecciono la plaza en la que se ubicó he inmediatamente esta información es procesada y el vehículo ya está ocupando y pagando un valor por 15 minutos, también la luz indicadora cambio a una tonalidad Verde que refiere a estar pagada la plaza correspondiente. En la figura 8 se presenta otro caso puntual que un vehículo ocupa la plaza 3 con una hora de inicio de 00:06 y tiene 14 minutos para estacionarse.



**Figura 30:** Caso Plaza Ocupada y sin pago

**Fuente:** Elaboración propia

### 3.4 Monedas a ingresar

El monedero es un selector de monedas automáticos y se encuentra configurado para el ingreso de 5 tipos de monedas como se explicó en la parte superior de este documento, es así que en la figura 9 se muestra cuáles son las monedas óptimas para el pago de los espacios de parqueo.



**Figura 31:** Monedas Validas

**Fuente:** Elaboración propia

