



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

ESCUELA DE INGENIERÍA EN MECATRÓNICA

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO  
DE INGENIERO EN MECATRÓNICA

TEMA:

“PLATAFORMA ABIERTA PARA DESARROLLO DE  
DISPOSITIVOS DE TARIFACIÓN VEHICULAR: SOFTWARE PARA  
AJUSTES”

AUTOR: HÉCTOR MAURICIO YÉPEZ PONCE

DIRECTOR: CARLOS XAVIER ROSERO CHANDI

IBARRA-ECUADOR

JULIO 2019



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

## 1. IDENTIFICACIÓN DE LA OBRA

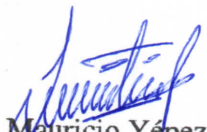
El cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

<b>DATOS DEL AUTOR</b>			
<b>CÉDULA DE IDENTIDAD:</b>	1003798541		
<b>APELLIDOS Y NOMBRES:</b>	YÉPEZ PONCE HÉCTOR MAURICIO		
<b>DIRECCIÓN:</b>	CDLA. YANAYACU - OTAVALO		
<b>EMAIL:</b>	hmyepezp@utn.edu.ec - mauroyepez32@gmail.com		
<b>TELÉFONO FIJO:</b>	062903072	<b>TELÉFONO MÓVIL:</b>	0979444269
<b>DATOS DE LA OBRA</b>			
<b>TÍTULO:</b>	"PLATAFORMA ABIERTA PARA DESARROLLO DE DISPOSITIVOS DE TARIFACIÓN VEHICULAR: SOFTWARE PARA AJUSTES"		
<b>AUTOR:</b>	HÉCTOR MAURICIO YÉPEZ PONCE		
<b>FECHA (AAAA-MM-DD):</b>	2019-07-19		
<b>SÓLO PARA TRABAJOS DE GRADO</b>			
<b>PROGRAMA:</b>	PREGRADO		
<b>TÍTULO POR EL QUE OPTA:</b>	INGENIERO EN MECATRÓNICA		
<b>ASESOR/DIRECTOR:</b>	CARLOS XAVIER ROSERO CHANDI		

## 2. CONSTANCIAS

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

Ibarra, a los 19 días del mes de Julio de 2019.

  
Héctor Mauricio Yépez Ponce  
C.I.: 1003798541



UNIVERSIDAD TÉCNICA DEL NORTE  
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS  
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A  
FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, Héctor Mauricio Yépez Ponce con cédula de identidad Nro. 1003798541, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículos 4, 5 y 6, en calidad de autor (es) de la obra o trabajo de grado denominado "PLATAFORMA ABIERTA PARA DESARROLLO DE DISPOSITIVOS DE TARIFACIÓN VEHICULAR: SOFTWARE PARA AJUSTES", que ha sido desarrollado para optar por el título de Ingeniero en Mecatrónica, en la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Técnica del Norte.

Ibarra, Julio de 2019

Héctor Mauricio Yépez Ponce  
C.I.: 1003798541



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**  
**CERTIFICACIÓN**

En calidad de director del trabajo de grado “PLATAFORMA ABIERTA PARA DESARROLLO DE DISPOSITIVOS DE TARIFACIÓN VEHICULAR: SOFTWARE PARA AJUSTES”, presentado por el egresado HÉCTOR MAURICIO YÉPEZ PONCE, para optar por el título de Ingeniero en Mecatrónica, certifico que el mencionado proyecto fue realizado bajo mi dirección.

Ibarra, Julio de 2019



**Carlos Xavier Rosero**  
**DIRECTOR DE TESIS**



UNIVERSIDAD TÉCNICA DEL NORTE  
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS  
DECLARACIÓN

Yo, Héctor Mauricio Yépez Ponce con cédula de identidad Nro. 1003798541, declaro bajo juramento que el trabajo aquí escrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Universidad Técnica del Norte - Ibarra, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

Ibarra, julio de 2019

Héctor Mauricio Yépez Ponce  
C.I.: 1003798541

## **Agradecimiento**

Agradezco primero a la Universidad Técnica del Norte. La Facultad de Ingeniería en Ciencias Aplicadas por haberme permitido ingresar a su mundo de conocimientos y a todos los docentes que han dejado en mí un granito de arena en cada clase, para que vaya recolectándolos uno por uno hasta poder construir una montaña de conocimientos que finalmente está dando frutos.

Agradezco a mi tutor Msc. Ing. Xavier Rosero por ser una persona paciente y brindarme su apoyo y conocimientos, pero más que nada por ser un buen docente y persona de bien, ya que sin ello no se consigue nada en la vida, gracias a su ejemplo y ayuda he podido salir adelante con la tesis.

A mis padres y hermanos por siempre confiar en mí, por siempre brindarme el apoyo económico y moral para cada día salir adelante con los estudios, por darme todo el amor del mundo, valores y ejemplo de personas humanas para formarme como un hombre de bien.

A mis compañeros y compañeras de clase que con sus ocurrencias me han hecho sonreír en los buenos y malos momentos, y como no, dar gracias a nuestra inolvidable Facultad que sus aulas han sido fieles testigos de mis triunfos y fracasos en mi largo camino por alcanzar una profesión para defenderme en un futuro como una persona útil para la sociedad.

*Héctor Yépez*

## **Dedicatoria**

A Dios, Por darme la vida, por estar a mis espaldas en cada paso que doy, por fortalecer mi corazón e iluminar nuestra mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el período de estudio.

A mis padres, Por ser un ejemplo de perseverancia, e inculcarme los valores que me han ayudado en toda circunstancia de la vida, por dejar todo su sudor y dedicación para que sea un hombre de bien, y finalmente por siempre brindarme su calor y amor constante de padres y amigos.

A mis hermanos y hermanas, Que siempre confiaron en mí, por su apoyo y por estar conmigo, y para que vean en mí un ejemplo a seguir.

*Héctor Yépez*



# Resumen

En el Ecuador para cambiar los parámetros de los datos medidos de un taxímetro, el propietario debe dirigirse a la autoridad metrológica, lugar en el cual se procede a desconectar totalmente el dispositivo del vehículo y conectarlo mediante un cable de datos (USB, RS232, entre otros) al instrumento que realizará la configuración, resultando algo incómodo para el propietario del medio de transporte. Con el fin de facilitar este proceso, se procede a realizar una interfaz gráfica que permita conectarse inalámbricamente con la memoria EEPROM del microcontrolador donde se encuentran almacenados todos los parámetros que permiten el funcionamiento del taxímetro. Con la ayuda de la interfaz se puede visualizar los datos y modificarlos de una manera cómoda, rápida e intuitiva. Para el diseño del trabajo mencionado, se establecen requerimientos funcionales y no funcionales basados en las necesidades del sistema y de los usuarios. De esta manera se busca reemplazar los métodos tradicionales que se manejan en la sociedad actualmente.

# Abstract

In Ecuador, to change the parameters of the measured data of a taximeter, the owner must go to the metrological authority, where the device is completely disconnected from the vehicle and connected using a data cable (USB, RS232, among other things) to the instrument that will carry out the configuration, which is uncomfortable for the owner of the means of transport. In order to facilitate this process, it is necessary to perform a graphical interface that allows to connect wirelessly with the EEPROM memory of the microcontroller where all the parameters that allow the operation of the taximeter are stored. With the help of the interface you can visualize the data and modify them in a comfortable, fast and intuitive way. For the design of the aforementioned work, functional and non-functional requirements are established based on the needs of the system and the users. In this way it seeks to replace the traditional methods that are currently used in society.

# Índice general

<b>Índice general</b>	<b>XI</b>
<b>Índice de figuras</b>	<b>XIV</b>
<b>Índice de cuadros</b>	<b>XV</b>
<b>Lista de Programas</b>	<b>XVI</b>
<b>Introducción</b>	<b>1</b>
<b>1. Revisión Literaria</b>	<b>4</b>
1.1. Normativa Vigente . . . . .	4
1.2. Antecedentes . . . . .	5
1.2.1. Taxímetro S700 . . . . .	6
1.2.2. Taxímetro M1 plus . . . . .	6
1.2.3. Taxímetro Optronix TX-10 . . . . .	7
1.2.4. Taxímetro Tango XP . . . . .	7
1.3. Interfaz Gráfica . . . . .	7
1.3.1. Características de una Interfaz Gráfica . . . . .	8
1.4. Sistema Embebido . . . . .	9
1.5. Comunicación . . . . .	9
1.5.1. Wi-Fi . . . . .	10
1.5.2. Modelos de Referencia . . . . .	10

1.5.2.1.	Modelo OSI . . . . .	11
1.5.2.2.	Modelo TCP/IP . . . . .	12
1.6.	Sockets . . . . .	13
<b>2.</b>	<b>Descripción del Sistema</b>	<b>15</b>
2.1.	Requerimientos del Sistema . . . . .	15
2.1.1.	Requerimientos Funcionales . . . . .	15
2.1.2.	Requerimientos no Funcionales . . . . .	16
2.2.	Diagrama de bloques . . . . .	18
<b>3.</b>	<b>Arquitectura del Software</b>	<b>19</b>
3.1.	Subsistema para Interfaz Gráfica de Usuario . . . . .	19
3.2.	Subsistema para Configuración de Fecha y Hora . . . . .	21
3.3.	Subsistema para Transferencia de Datos . . . . .	21
<b>4.</b>	<b>Implementación del Sistema Computacional</b>	<b>23</b>
4.1.	Implementación y Pruebas de Funcionamiento . . . . .	23
4.1.1.	Implementación de la Interfaz Gráfica . . . . .	23
4.1.2.	Funcionamiento de la Interfaz Gráfica . . . . .	25
4.1.2.1.	Ingreso y Verificación . . . . .	25
4.1.2.2.	Lectura de Datos . . . . .	25
4.1.2.3.	Envío de Datos . . . . .	26
<b>5.</b>	<b>Conclusiones</b>	<b>28</b>
5.1.	Conclusiones . . . . .	28
5.2.	Recomendaciones . . . . .	29
5.3.	Trabajo futuro . . . . .	30
	<b>Bibliografía</b>	<b>31</b>

<b>Apéndice</b>	<b>35</b>
.A. Software . . . . .	36
.A.1. Interfaz Gráfica (taximetro.py) . . . . .	36
.A.2. Código Arduino (taximetro.ino) . . . . .	52
.2. Diseño PCB . . . . .	65

# Índice de figuras

1.1. Modelo de referencia OSI. . . . .	11
2.1. Diagrama de bloques del sistema . . . . .	17
4.1. Ventanas de interfaz gráfica . . . . .	24
4.2. Ventana de ingreso . . . . .	25
4.3. Confirmación de usuario . . . . .	25
4.4. Ventana de configuración . . . . .	26
4.5. Lectura de datos . . . . .	26
4.6. Envío de Datos . . . . .	26
4.7. Pestaña de servicio técnico . . . . .	27

# Índice de cuadros

1.1. Taxímetros Homologados . . . . .	5
1.2. Características Arduino Mega 2560 y Rasberry pi3 . . . . .	9
1.3. Protocolos de comunicación inalámbrica . . . . .	10
3.1. Características principales módulo ESP8266 . . . . .	21
4.1. Botones de la interfaz gráfica . . . . .	25

# Lista de Programas

1.	Interfaz gráfica . . . . .	36
2.	Código arduino . . . . .	52



# Introducción

## Problema

El taxímetro es un instrumento de medición y control instalado en los vehículos de servicio de transporte que indica gradualmente el valor que debe pagar el usuario [1]. Desde el año 1982 en Ecuador se incluyó el uso obligatorio del taxímetro, sin embargo, se produjeron inconsistencias durante varios años que impedían su implementación [2]. Finalmente, en el año 2013 la ANT (Agencia Nacional de Tránsito) resolvió el uso obligatorio del taxímetro para todas las ciudades del país [3].

Con la rápida construcción urbana, los taxis se han convertido en un medio de transporte indispensable en la vida de las personas [4]. Actualmente, en la ciudad de Ibarra existen aproximadamente 1000 taxis convencionales y 400 ejecutivos de las diferentes cooperativas y compañías [5]. El taxímetro aplicado a este medio de transporte presenta varios inconvenientes, entre ellos se encuentran el elevado costo de adquisición, constantes calibraciones debido a desconfiguración del firmware, fallas técnicas en el hardware e incumplimiento de la normativa en [3]. Es por esto que la mayoría de transportistas denuncian irregularidades en el uso del taxímetro, ocasionando que el cliente no perciba una tarifa justa [5].

Considerando la problemática citada es necesario el desarrollo de una plataforma abierta (hardware y software), de bajo costo, para experimentación sobre tarifado vehicular. Este dispositivo permitiría que en la academia se investigue acerca de algoritmos de tarifación. Sería el

primer paso para el desarrollo de taxímetros que cumplan con la normativa en [3] y puedan ser certificados.

## Objetivos

El objetivo principal de este proyecto consiste en desarrollar una plataforma de software libre para experimentación sobre tarificación vehicular.. Los siguientes objetivos específicos son también realizados:

- Determinar la funcionalidad de los taxímetros existentes en el medio.
- Desarrollar el algoritmo para el software de configuración que cumpla con la funcionalidad requerida.
- Realizar el software de interfaz humano-máquina usando software libre en base a criterios de usabilidad.
- Implementar el sistema computacional para su experimentación en condiciones reales de funcionamiento.

## Alcance

Esta plataforma se desarrollará con software libre, y su funcionalidad será la mencionada en la norma INEN en [3]. Los ensayos para la certificación con la norma se realizarán en un trabajo posterior, ya que se encuentran fuera del enfoque del presente desarrollo.

El hardware del taxímetro se conectará por comunicación inalámbrica con el software que se realizará dentro de este proyecto. Este programa es una interfaz humano-máquina que permitirá leer y establecer parámetros de configuración para las siguientes funciones del taxímetro:

- Tarifas de usuario (tiempo inicial, distancia inicial, valor mínimo, incremento por cargos suplementarios, valor de la tarifa por tiempo y distancia).

- Constante del transductor K (impulsos por kilómetro).
- Actualización y modificación del programa.
- Información general (fecha de revisión, datos del propietario, identificación del vehículo, hora y fecha).
- Datos que contendrá el recibo (identificación de tarifa, costo del viaje, cargos suplementarios, distancia y duración del viaje, fecha y hora del viaje, número de identificación del taxi).

Para mejor entendimiento de las funciones, se recomienda referirse a [3].

## **Justificación**

La importancia de esta investigación radica en solucionar la problemática que actualmente sufren los propietarios de taxis luego de la adquisición e instalación de un taxímetro en sus unidades. En el país un taxímetro e impresora se venden por separado y a un costo elevado en relación a las prestaciones que ofrecen. Por tal motivo en esta investigación se desarrollará un taxímetro de bajo costo, que incluya mecanismos para obtener medidas fiables y consistentes con un mínimo error, disminuyendo así las constantes calibraciones.

El desarrollo de esta plataforma abierta permitiría establecer los parámetros de configuración del taxímetro de manera inalámbrica mediante una interfaz gráfica que reemplazaría a los métodos tradicionales que se usan actualmente, los cuales implican el desmontaje del equipo resultando incómodo para el propietario del medio de transporte.

Además, esta plataforma servirá como base para realizar posteriores modificaciones de hardware y software que permitirían obtener taxímetros homologados cumpliendo los parámetros establecidos en la norma INEN en [3].

# Capítulo 1

## Revisión Literaria

### 1.1. Normativa Vigente

La ANT (Agencia Nacional de Tránsito) en el año 2013 resolvió el uso obligatorio de taxímetros en los vehículos que prestan servicio de transporte en todas las ciudades del país, ya sea bajo la modalidad convencional o ejecutiva. El INEN (Instituto Ecuatoriano de Normalización) será el encargado de emitir el certificado de aprobación del modelo luego de realizar la verificación de ensayos y cumplimiento de la normativa.

De conformidad con la norma INEN en [3], establece que el taxímetro debe estar equipado con una interfaz de usuario que permita el intercambio de información entre un usuario humano y el taxímetro. La misma que debe permitir visualizar y configurar los siguientes parámetros:

- Información general ( Información del propietario).
- Marcas de verificación (Constante k, número de serie del taxímetro, identificación del vehículo, fecha de revisión, fecha y hora real).
- Información Arancelaria: Los parámetros de la(s) tarifa(s).

El software de configuración debe integrar una interfaz de protección que garantice la transmisión y actualización de datos medidos y por programa, en contra de cambios intencionales, no intencionales y accidentales.

Se debe utilizar un archivo auditable o un contador de eventos que se incremente cada vez que existan intervenciones para cambiar los distintos parámetros del taxímetro y debe permitir el acceso a esta información especificando la fecha y la persona autorizada que realizó la intervención.

Los registros y datos relevantes deben ser almacenados en la memoria del taxímetro o en un dispositivo de almacenamiento externo que deberán estar protegidos adecuadamente contra cualquier tipo de cambio. Cuando la memoria alcance su capacidad máxima, ninguna otra intervención será posible sin romper el sello físico y los datos nuevos podrán reemplazar a los datos antiguos, siempre y cuando el propietario haya autorizado la sobre escritura.

## 1.2. Antecedentes

La ANT en [6] describe los diferentes modelos de taxímetros homologados en el país con las respectivas empresas proveedoras de estos dispositivos de medición, control y seguridad, como se detalla en la Tabla 1.2.

Tabla 1.1: Taxímetros Homologados

NUM	MARCA	MODELO
1	VIRTUALTEC	VIRLOC 10
2	FUL-MAR	TANGO XP
3	DIGITAX 550	M07
4	DIGITAX 550	M11
5	LAKEDRIVER	SLIM-800
6	TAXSYM	M09
7	DIGITAX	F3 PLUS
8	DIGITAX	M1 PLUS
9	SEMSA	PLATINUM
10	MICROTEK	MT12
11	MICROTEK	MTMK

12	TAXITRONIC	TX-40
13	TAXITRONIC	TX-80 SKYGLASS
14	INTELLIGENT TAXIMETER	D10
15	DIVUTAXI	I700-3G
16	STALIN-TAX	ST-11
17	CENTRODYNE	S700
18	OPTRONIC	TX-10
19	ARIEL TAX	MILENIO
20	ATA PRIMUS	RS-01
21	ATA PRIMUS	S-01

Los taxímetros descritos anteriormente varían dependiendo de las funcionalidades que poseen sin embargo el tipo de comunicación que utilizan para la configuración de parámetros y actualización de firmware son semejantes, a continuación se detalla algunos ejemplos.

### **1.2.1. Taxímetro S700**

Taxímetro fabricado por la empresa Centrodyne, posee 16 tarifas estándar que realizan un cambio automático, límite de velocidad programable, 3 niveles de contraseña de seguridad y la comunicación con la computadora es mediante bluetooth, USB o RS232 [7].

### **1.2.2. Taxímetro M1 plus**

Taxímetro que lleva la marca Digitax tiene la forma de un retrovisor, la comunicación se lo puede realizar por medio de dos puertos serie RS232 a cualquier velocidad en baudios de 300 a 115200, usando el protocolo COMMTAX2 que permite la administración remota del taxímetro, lector de tarjeta de crédito magnético, tarjeta de chip y funciones de impresora descritas en [8].

### **1.2.3. Taxímetro Optronic TX-10**

Dispositivo 100% diseñado y construido en Ecuador por la empresa Electrónica Industrial Optronic CIA. LTDA. Es multitarifario, es decir, puede operar con tarifas diferentes dependiendo de la hora y fecha del día, la comunicación es por Serial TTL, RS232 o RS485 y se puede conectar a varios periféricos como impresora y GPS (Sistema de posicionamiento geográfico) [9].

### **1.2.4. Taxímetro Tango XP**

Este taxímetro cuenta con 6 tarifas independientes horarias y automáticas, un software de máxima flexibilidad que se puede modificar de acuerdo a cambios en la reglamentación y ordenanzas vigentes. Además para realizar cambios en el software de configuración es necesario disponer de un módulo JTAG que permite la comunicación con la computadora a través de los puertos COM que posee [10].

## **1.3. Interfaz Gráfica**

Cuando existen dos sistemas cualesquiera, que se deben comunicar entre ellos, la interfaz será el mecanismo, el entorno o la herramienta que hará posible dicha comunicación. Por lo tanto la interfaz de usuario deberá ser capaz de hablar pero no de forma verbal, sino a través de los elementos gráficos que la conforman[11]. El diseño de interacción es un aspecto fundamental en el desarrollo de una interfaz gráfica. Este diseño busca implicar al usuario en todo el proceso de diseño, observando cómo actúan y reaccionan los usuarios ante eventos que se van presentando en el transcurso de la interacción. Para solucionar este tipo de inconvenientes es necesario un trabajo multidisciplinario, como ingenieros, psicólogos, sociólogos, diseñadores gráficos entre otros, y de esta manera lograr un producto final que sea fácil de comprender, estéticamente agradable, que pueda satisfacer las necesidades del usuario y cumpla con las funcionalidades requeridas [12]. Cuando se toman decisiones en el diseño de las interfaces de usuario, se deben

tener en cuenta las capacidades físicas y mentales de las personas que utilizarán el software, por tal razón se debe considerar los siguientes aspectos en el proceso de diseño de una interfaz gráfica [13].

- **Análisis del usuario.**- Comprender las tareas que éste realiza, identificar y establecer los requisitos.
- **Prototipado del sistema.**- Desarrollar varios prototipos del sistema y exponerlos a los usuarios, quienes pueden entonces guiar la evolución de la interfaz.
- **Evaluación de la interfaz.**- Esta actividad permitirá recopilar información sobre las experiencias reales de los usuarios con la interfaz.

Finalmente la usabilidad de un producto dependerá de los siguientes aspectos: efectividad, eficiencia, seguro al usar, fácil de aprender y fácil de recordar [12].

### **1.3.1. Características de una Interfaz Gráfica**

Para lograr lo mencionado anteriormente en el diseño de una interfaz gráfica se debe tomar en cuenta las características que se muestran en [14].

- **Sencilla.**- Colocar los elementos necesarios, evitando la saturación de los mismos.
- **Clara.**- La información debe estar organizada y fácil de localizar.
- **Flexible.**- Debe ser clara y entenderse fácilmente, con formatos similares para que puedan ser compatibles con otras plataformas.
- **Consistente.**- Tener una semejanza de diseño entre pestañas o capítulos del programa.
- **Intuitiva.**- El usuario interactúe de manera segura sin tener que adivinar las funciones o como ejecutar una acción.
- **Coherente.**- La parte visual como gráficos, colores y demás elementos deben ser adecuados al tipo de información que se desea transmitir.



Si el diseño de interacción, facilidad de uso y manejo del mensaje visual, logran una buena inmersión entre el usuario y la interfaz, se habrá cumplido con el objetivo que conlleva el diseño de una interfaz gráfica [12].

## 1.4. Sistema Embebido

“Se conoce como sistema embebido a un circuito electrónico computarizado que está diseñado para cumplir una labor específica en un producto” [37]. A continuación en la Tabla 1.2 se detallan las características de dos marcas utilizadas para el desarrollo de aplicaciones electrónicas en software libre.

Tabla 1.2: Características Arduino Mega 2560 y Raspberry pi3

Características	Arduino Mega 2560	Raspberry Pi3
Precio	15 dólares	60 dólares
Memoria Mbps	0.002 MB	1 GB
Velocidad de reloj	16 MHz	700 MHz
Multitarea	No	Si
Voltaje de entrada	7 a 12V	5V
Memoria Flash	32KB	SD (2 a 16GB)
Entorno de desarrollo integrado	Arduino	Scratch, IDLE, cualquier soporte de linux

Raspberry Pi es superior a Arduino, pero eso es sólo cuando se trata de aplicaciones de software. La simplicidad de Arduino hace que éste sea una apuesta mucho mejor para proyectos de hardware.

## 1.5. Comunicación

Para poder transmitir información inalámbricamente desde el computador hacia un microcontrolador existen varios módulos con diferentes protocolos de comunicación como nos muestra [15] en la Tabla 1.3. Que permiten conectar nodos, sensores o dispositivos sin necesidad de una conexión física, es decir son redes sin cables.

Tabla 1.3: Protocolos de comunicación inalámbrica

Technology	Bluetooth	ZigBee	Z-Wave	6LoWPAN	Wi-Fi
Frequency, GHz	2,4	2,4;0,915;0,868;	up to 1,0	1,0; 2,4	2,1;5,0
Speed, Mbps	up to 24	up to 0,25;	up to 0,1	up to 0,250	up to 300
Approx. range, m	up to 100	up to 100;	30	800	up to 100
The cost of the module	120	170	330	500	100

### 1.5.1. Wi-Fi

Wi-Fi (Wireless Fidelity) es un conjunto de estándares de la IEEE 802.11 [16] creadas para redes locales inalámbricas, las cuales se utilizan para acceso a internet y redes privadas. Este estándar permite trabajar con velocidades de transmisión de 1 Mbps y 2 Mbps, y existen varias versiones del estándar IEEE 802.11, como IEEE 802.11b/a/g/n donde su fin es añadir nuevas técnicas de modulación que permiten alcanzar mayores velocidades de transmisión y conectividad de información [17]. La principal ventaja que supone una red Wireless frente a una de cables, es la movilidad y su fácil instalación. Actualmente, muchos usuarios y empleados de empresas requieren acceder en forma remota a sus archivos, trabajos y recursos. La red Wireless permite hacerlo sin realizar ninguna tarea compleja de conexión o configuración, y evita que cada usuario viaje hasta su empresa o su casa para poder acceder a los recursos de su red de datos [18].

### 1.5.2. Modelos de Referencia

Hay dos arquitecturas que han sido determinantes en el desarrollo de los estándares de comunicación: el modelo de referencia OSI y el modelo de referencia TCP/IP. Los protocolos asociados con el modelo OSI casi no se utilizan, pero su modelo se ha convertido en una base para clasificar las funciones de comunicación. Por otro lado el protocolo TCP/IP, el modelo en sí no se utiliza mucho, pero los protocolos son usados ampliamente para la interconexión de sistemas [19].

### 1.5.2.1. Modelo OSI

Modelo de referencia OSI (Open Systems Interconnection) se ocupa de la conexión de sistemas abiertos a la comunicación con otros sistemas. El modelo OSI tiene siete capas como se muestra en la Fig. 1.1. Aunque el modelo (en parte) es muy usado, los protocolos asociados han estado en el olvido desde hace tiempo [20].

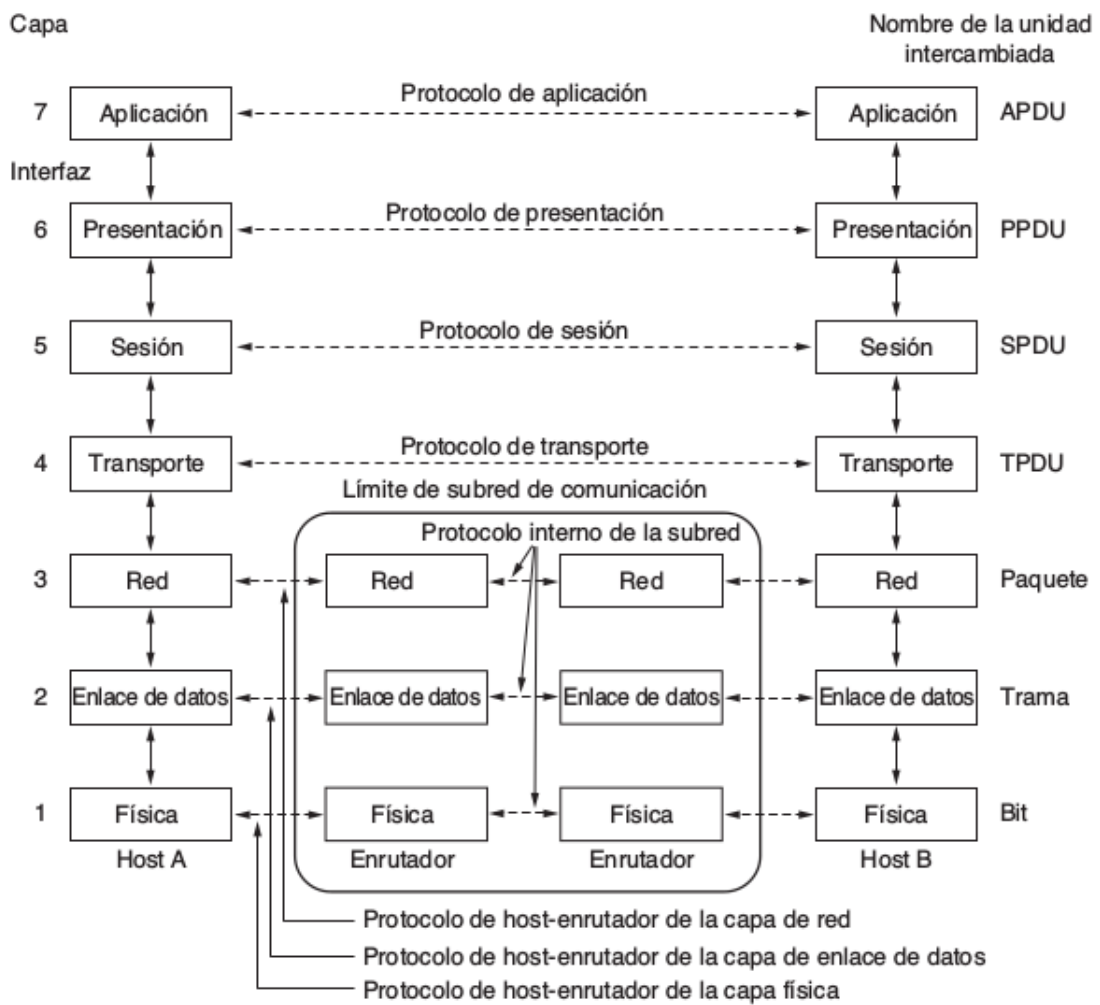


Figura 1.1: Modelo de referencia OSI.

- Capa física.- Se encarga de la transmisión de cadena de bits no estructurados sobre el medio físico.

- Capa de enlace de datos.- Realiza la transferencia de datos a través de un enlace físico, enviando bloques de datos ordenados en tramas, llevando a cabo la sincronización, control de errores y de flujo.
- Capa de red.- Es la responsable del establecimiento, mantenimiento y cierre de las conexiones.
- Capa de transporte.- Encargada de proporcionar seguridad en la transferencia de datos, además procedimientos de recuperación de errores y control de flujo origen-destino.
- Capa de sesión.- Controla la comunicación entre aplicaciones, es la encargada de establecer, gestionar y cerrar las conexiones.
- Capa de presentación.- Se enfoca en la sintaxis y la semántica de la información transmitida y pueda ser interpretada en diferentes tipos de máquinas.
- Capa de aplicación.- Un protocolo de aplicación muy utilizado es HTTP ( HyperText Transfer Protocol), el cual forma la base para la World Wide Web, que permite transferir archivos, enviar y recibir la información deseada.

#### **1.5.2.2. Modelo TCP/IP**

El protocolo TCP/IP es el líder de la comunicación en red. Puede utilizarse para establecer conexiones entre diferentes computadoras y servidores. Este tipo de interoperabilidad es una de las principales ventajas de TCP/IP. Además funciona independientemente del sistema operativo con una arquitectura escalable, cliente/servidor [21]. El modelo TCP/IP consta de cuatro capas basadas en [20], que se detallan a continuación.

- La capa de enlace.- Describe que enlaces se deben llevar a cabo para cumplir con las necesidades de esta capa de interred sin conexión, es una interfaz entre los hosts y los enlaces de transmisión.

- La capa de interred.- Se encarga de definir un formato de paquete y un protocolo oficial llamado IP (Internet Protocol), además de un protocolo complementario llamado ICMP (Internet Control Message Protocol ) que le ayuda a funcionar. La tarea de la capa de interred es entregar los paquetes IP a su destino.
- La capa de transporte.- Diseñada para permitir que los nodos de origen y de destino, lleven a cabo una conversación, para esto se necesitan dos protocolos de transporte.
  - El primero, TCP (Transmission Control Protocol), permite que un flujo de bytes originado en una máquina se entregue sin errores a cualquier otra máquina en la interred. También maneja el control de flujo para que un emisor rápido no pueda inundar a un receptor lento con más mensajes de los que pueda manejar.
  - UDP (User Datagram Protocol), Se utiliza mucho en las consultas de petición - respuesta del tipo cliente-servidor, y en las aplicaciones en las que es más importante una entrega oportuna que una entrega precisa.
- Capa de aplicación.- Contiene la lógica necesaria para posibilitar la transferencia de ficheros usando protocolos de alto nivel entre las distintas aplicaciones de usuario.

## 1.6. Sockets

Un socket implementa una arquitectura cliente/servidor que permite una comunicación bidireccional entre procesos ubicados en diferentes máquinas. Su objetivo es proporcionar un método general IPC (Inter-Process Communication) en la capa de transporte, y lograr que varios procesos se ejecuten en la misma máquina. La forma de referenciar un socket es a través de una dirección IP y un número de puerto que permita realizar una conexión TCP entre los computadores para lograr transferir datos e interactuar entre ellos [22].

Las propiedades de un socket dependerá del protocolo de comunicación a implementarse.

- Protocolo TCP.- Orientado a la conexión que garantiza la transmisión de octetos sin errores ni omisiones y conservando el orden en que fue transmitido.

- Protocolo UDP.- Es un protocolo no orientado a la conexión, sin garantía de entrega y los mensajes tienen una alta probabilidad de llegar en un orden que no fueron enviados, es por esto que es adecuado unicamente para el envío de mensajes frecuentes pero no importantes [23].

# Capítulo 2

## Descripción del Sistema

### 2.1. Requerimientos del Sistema

Representan las necesidades de los clientes y las personas involucradas en el desarrollo del sistema. Estos requisitos describen lo que un sistema debe hacer, información específica sobre los servicios que proporciona y las restricciones con las cuales debe operar. Se clasifican en funcionales y no funcionales [24].

#### 2.1.1. Requerimientos Funcionales

Dependen del tipo de software a desarrollar y hacen referencia a las actividades y servicios que el sistema debe realizar. Además están estrechamente relacionados con las entradas, las salidas de cada proceso y el almacenamiento de información en el sistema [12]. A continuación se presentan los requerimientos funcionales de la Plataforma Abierta para Desarrollo de Dispositivos de Tarifación Vehicular:

- El software posee una interfaz gráfica de usuario, que muestra los parámetros de configuración del taxímetro almacenados en la memoria EEPROM del microcontrolador, permitiendo al usuario interactuar con los mismos. La interfaz funciona a través de la consola de comandos de Linux y despliega todas las acciones que realiza mientras se ejecuta la

aplicación.

- La interfaz de usuario y el microcontrolador se comunican inalámbricamente mediante un módulo wifi accediendo a los diferentes parámetros de configuración del taxímetro establecidos en la normativa INEN en [3].
- La aplicación posee un subsistema para definir la hora y fecha actual del módulo RTC (Real–Time Clock) conectado al microcontrolador, los cambios se los puede ejecutar de manera manual a través de la interfaz gráfica o automáticamente tomando como referencia la hora del computador y para realizarlo es necesario de un subsistema de transferencia remota de datos [24].

### **2.1.2. Requerimientos no Funcionales**

Surgen de las necesidades del usuario y representan las restricciones que el sistema debe tener, no se relacionan directamente con los servicios específicos, sin embargo el incumplimiento de estos requisitos haría que el sistema no funcione de la manera esperada o fuera inútil [25]. Existen estándares para la evaluación de la calidad de un sistema como el ISO/IEC 9126 (International Standard Organization, 2001), que propone un listado de atributos que garantizan la calidad de operabilidad del sistema [26]. A continuación se presentan los requerimientos no funcionales de la Plataforma Abierta para Desarrollo de Dispositivos de Tarifación Vehicular:

- Usabilidad.- El software de ajustes posee un diseño sencillo, claro, interactivo y dividido en pestañas para mejor visibilidad.

El tiempo estimado de aprendizaje por el usuario es menor de 30 minutos debido a su fácil operación.

- Eficiencia.- El envío y recepción de datos desde la interfaz ubicada en el computador hacia el microcontrolador tiene una capacidad máxima de 64 bytes, este valor corresponde al tamaño del buffer de entrada que posee el microcontrolador utilizado.

La aplicación responde en un rango menor a 5 segundos.



- Seguridad.- El protocolo de comunicación es TCP/IP, por tal motivo la dirección IP y el número del puerto son definidos en la programación del software.

La interfaz posee una contraseña de acceso que impide el ingreso de personas no autorizadas a manipular el sistema.

- Disponibilidad.- El software de ajustes se encuentra almacenado en el computador de tal manera que puede estar en funcionamiento las 24 horas.

El microcontrolador estará conectado a la batería del vehículo y tomando en cuenta que el consumo energético de los componentes es mínimo puede funcionar durante todo el día.

El computador se comunica mediante wifi con el microcontrolador a través del módulo esp8266 por lo que la disponibilidad de red será de un 99,9%, siempre y cuando no se exceda la distancia máxima entre los dispositivos.

- Portabilidad.- Usar herramientas y lenguajes basados en software libre dan como resultado una instalación sencilla, sin costo y de fácil mantenimiento. Esto facilita realizar mejoras, corregir errores y adaptar a plataformas sin costo alguno.

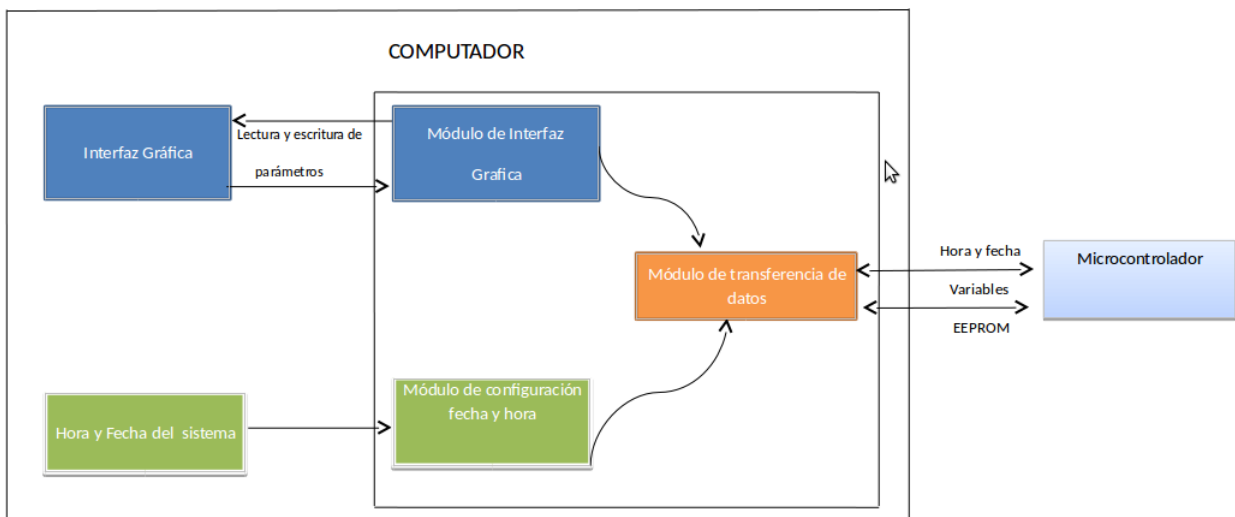


Figura 2.1: Diagrama de bloques del sistema

## **2.2. Diagrama de bloques**

El diagrama de bloques del sistema se visualiza en la Fig. 2.1. Cada uno de los subsistemas se describen en el capítulo 3.

# Capítulo 3

## Arquitectura del Software

El software de ajustes funciona con un sistema operativo de código abierto denominado Ubuntu. Una distribución de Linux, fácil de usar, segura, estable y gratuita que puede ser fácilmente descargada desde internet[27]. El lenguaje de programación para el desarrollo de la interfaz de usuario y transferencia de datos es python. Un lenguaje de programación interpretado, interactivo, orientado a objetos que proporciona estructuras de datos de alto nivel, escritura dinámica con una sintaxis muy simple y elegante que lo convierte en un lenguaje de programación potente y de propósito general[28]. El programa en el computador está compuesto por tres principales subsistemas que son:

- Subsistema para interfaz gráfica de usuario
- Subsistema para configuración de fecha y hora
- Subsistema para transferencia de datos

### 3.1. Subsistema para Interfaz Gráfica de Usuario

Tkinter es un paquete estándar incluido en Python para la creación de interfaces de usuario (GUI), funciona con la mayoría de las plataformas (Linux, OSX, Windows). Para poder usarla se utiliza el módulo de python *tkinter* [29]. La forma de llamar a los elementos que posee esta

librería varía dependiendo la versión de python empleada en este caso se utiliza Python 3.x. La fortaleza de Tkinter es su simplicidad de diseño, que se complementa con una variedad de herramientas como: botones, etiquetas, gráficas, textos, ventanas, pestañas, etc. que proporcionan la interacción entre el software y el hombre. La interfaz gráfica está compuesta por cinco pestañas, en cada una se encuentran distribuidas varias etiquetas y cuadros de texto que permiten leer y escribir lo siguiente:

- Información general.- Datos principales del propietario y del vehículo.
- Parámetros.- Valores que dependen del tipo de taxímetro, como la identificación de programa y valores utilizados para el cálculo del costo y redondeo.
- Tarifas.- Monto fijado por el municipio de cada ciudad mediante una ordenanza para el cobro del servicio que presta el medio de transporte a la ciudadanía.
- Servicio técnico.- El día de la próxima inspección, lectura del último mantenimiento y la configuración de fecha y hora actual de forma manual o automática.

Finalmente se observan cuatro botones que cumplen con una función específica en cada uno de los subsistemas que conforman el software y son:

- Lectura.- Lee los valores almacenados en la memoria EEPROM del microcontrolador y los despliega de manera ordenada en los cuadros de texto.
- Enviar.- Envía la información colocada hacia el microcontrolador.
- Borrar.- Limpia los valores escritos en cada segmento para su posterior lectura o escritura de datos.
- Salir.- Cierra la ventana y corta la comunicación entre la interfaz gráfica y la placa arduino.

## 3.2. Subsistema para Configuración de Fecha y Hora

Permite obtener la hora y fecha del sistema y enviar esos datos al RTC, esto es posible usando el paquete *datetime* [30] y *calendar* [31] que amplían las posibilidades del módulo *time* [32], para manipular las expresiones de tiempo.

## 3.3. Subsistema para Transferencia de Datos

La transferencia de datos desde el computador hacia el microcontrolador es mediante Wi-Fi a través del módulo ESP8266 que tiene las características que se detallan en la tabla 3.1. Considerando las características presentadas en la tabla 3.1, la transferencia de datos se realiza

Tabla 3.1: Características principales módulo ESP8266

Items	Parameters
Operating Voltage	2.5V 3.6V
Protocols	802.11 b/g/n
Frecuency Range	WiFi 2.4 GHz
Wi-Fi Mode	Station/SoftAP/SoftAP+Station
Security	WPA/WPA2
Network Protocols	IPv4, TCP/UDP/HTTP

mediante el protocolo TCP/IP. Para usar este tipo de conexión se importa en python el módulo *socket* [33]. Para realizar la comunicación se necesita un socket servidor (ESP8266) y un socket cliente (Computador), que al establecer conexión permitirá la escritura y lectura de parámetros entre ambos. La cantidad de bytes para envío y recepción de datos dependerá del tamaño del búffer del microcontrolador. Cuando la dirección IP y el número de puerto entre emisor y receptor sea establecida correctamente el servidor emitirá una petición al cliente para entablar la comunicación. Una vez que el cliente responde se procede a la lectura y escritura de los parámetros de configuración del taxímetro. Para evitar pérdida de información durante la transmisión se optó por despachar los datos en partes, de esta manera se logra una lectura y escritura ordenada y sin errores. Finalmente se utilizaron tres módulos adicionales en el diseño del software de ajustes:

- Módulo *sys* [34], se encarga de proveer variables y funcionalidades, directamente relacionadas con el intérprete.
- Módulo *os* [35], accede a funcionalidades dependientes del sistema operativo, sobre todo manipula la estructura de directorios.
- Módulo *threading* [36], ejecuta varias operaciones simultáneamente en un mismo espacio de proceso.

# Capítulo 4

## Implementación del Sistema Computacional

### 4.1. Implementación y Pruebas de Funcionamiento

#### 4.1.1. Implementación de la Interfaz Gráfica

El desarrollo de una interfaz gráfica garantiza al usuario acceder y enviar información de manera sencilla sin la necesidad de tener un conocimiento profundo de la estructura del software, la interfaz consta de tres ventanas, como se puede observar en la Fig 4.1. a) La consola del sistema operativo del computador (terminal de Linux), que lleva a cabo la ejecución de la aplicación y visualización de los valores que se envían hacia el microcontrolador. b) Ingreso y verificación para el acceso a la interfaz. c) Un panel con cinco pestañas que contienen los principales parámetros de configuración de un taxímetro. Las acciones que efectúa la interfaz gráfica son: controlar que el ingreso sea realizado por personas autorizadas esto se lo resuelve mediante un usuario y contraseña, extraer los datos almacenados en la memoria EEPROM del microcontrolador, enviar los valores de configuración necesarios para el funcionamiento del taxímetro, actualización de hora y fecha del RTC, borrar los datos ubicados en cada cuadro de texto una vez realizada la lectura y cerrar las ventanas emergentes de la aplicación.

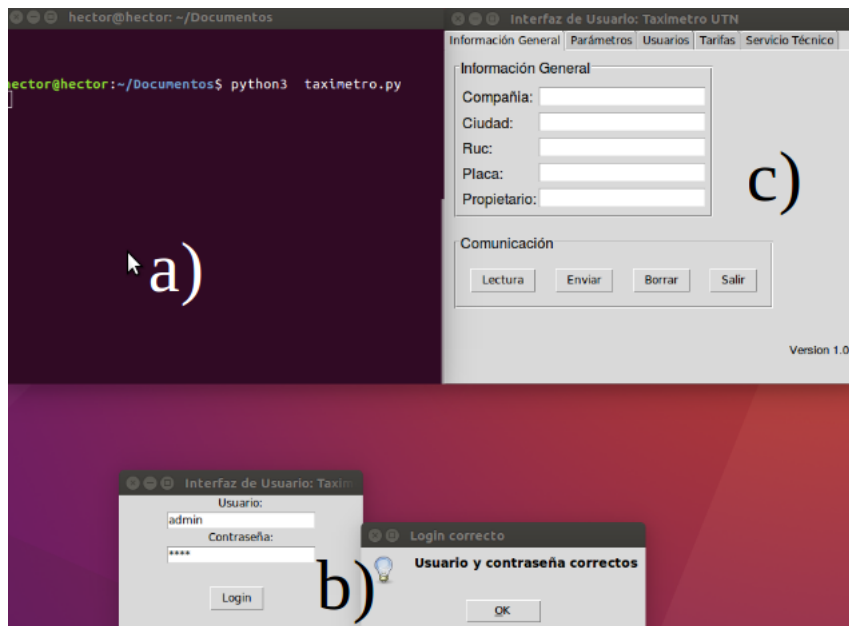


Figura 4.1: Ventanas de interfaz gráfica

Para hacer uso de la interfaz gráfica es necesario realizar los siguientes pasos:

- Encender el módulo ESP8266.
- Conectar el computador al red inalámbrica creada por el módulo wifi.
- Encender el microcontrolador.
- Colocar el taxímetro en modo configuración, esto se realiza seleccionando en el dispositivo la opción *libre*, y posteriormente la opción *modo de configuración*.
- Por medio del terminal de linux se accede a la dirección en donde se encuentra ubicado el archivo y se escribe `python3 taximetro.py` seguido de la tecla enter.
- Ingresar usuario y contraseña de verificación.

Si todos los pasos descritos anteriormente se realizaron correctamente el usuario puede interactuar con la interfaz gráfica. Para una mejor comprensión en la tabla 4.1 se describe el nombre y la actividad que realiza cada uno de los botones.



Tabla 4.1: Botones de la interfaz gráfica

Nombre	Actividad
Lectura	Extrae los valores almacenados en la memoria EEPROM del microcontrolador
Enviar	Envía los valores ingresados en la interfaz hacia el microcontrolador
Borrar	Limpia todos los cuadros de texto
Salir	Cierra la interfaz gráfica

## 4.1.2. Funcionamiento de la Interfaz Gráfica

A continuación se describe el funcionamiento de la interfaz gráfica.

### 4.1.2.1. Ingreso y Verificación

Al introducir el usuario y contraseña designados previamente y presionando el boton *Login* de la Fig. 4.2. Se procede a validar los datos ingresados y si son correctos se despliega una ventana emergente de confirmación como se muestra en la Fig. 4.3.

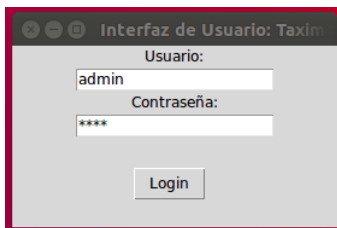


Figura 4.2: Ventana de ingreso

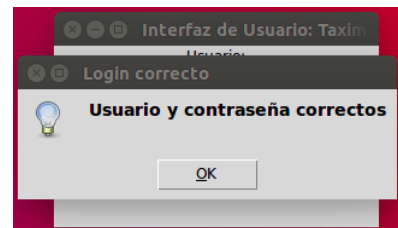


Figura 4.3: Confirmación de usuario

### 4.1.2.2. Lectura de Datos

Una vez confirmado que el usuario es el correcto se despliega la ventana de configuración con cinco pestañas como nos muestra la Fig. 4.4, presionando el botón *Lectura* se desplegarán los datos que se encuentran almacenados en la memoria EEPROM, si es primera vez que se ejecuta la operación todos los casilleros despliegan un valor de cero, como se puede ver en la Fig. 4.5.

El proceso y resultado será idéntico en las cuatro pestañas restantes.

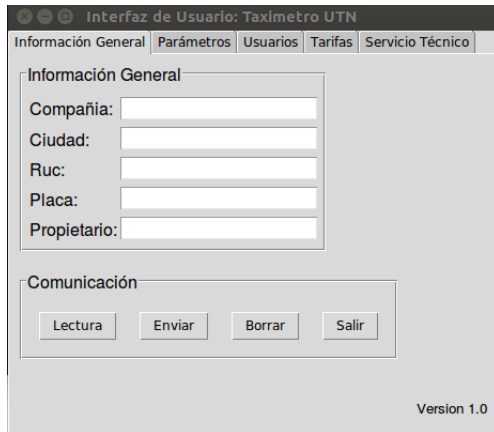


Figura 4.4: Ventana de configuración

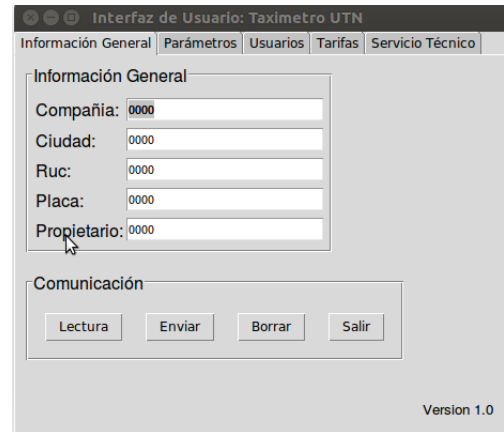


Figura 4.5: Lectura de datos

#### 4.1.2.3. Envío de Datos

Para enviar datos es necesario ingresar en cada cuadro de texto la información correspondiente y al pulsar el botón *Enviar*, los datos se enviarán por medio del módulo wifi hacia la memoria de la placa arduino, como se puede apreciar en la Fig 4.6. De igual forma se realizará en cada una de las pestañas.

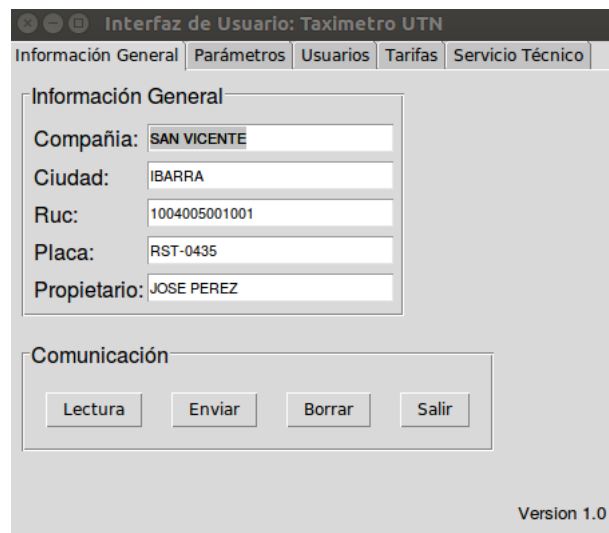


Figura 4.6: Envío de Datos

El proceso culminará en la última pestaña obligatoriamente, puesto que esta última pestaña

indica la cantidad de veces que el dispositivo a sido modificado y registra el nombre de la persona la modificación de valores por medio de la interfaz, como se puede ver en la Fig. 4.7

Interfaz de Usuario: Taxímetro UTN

Información General Parámetros Usuarios Tarifas Servicio Técnico

Fecha y Hora

Configuración Manual

dd/MM/aaaa 16 7 2019

H:mm:ss 9 59 47

Servicio Técnico

Fecha de próxima inspección

dd/MM/aaaa 13 08 2019

Última revisión técnica

dd/MM/aaaa 16/07/2019

Contador de eventos 26

Responsable PABLO PONCE

Comunicación

Lectura Enviar Salir

Version 1.0

Figura 4.7: Pestaña de servicio técnico

# Capítulo 5

## Conclusiones

### 5.1. Conclusiones

En este trabajo se desarrolló una plataforma de software libre que permite configurar los datos relevantes de un taxímetro como: información general del propietario y del vehículo, marcas de verificación para el cálculo de costo entre ellos los impulsos por kilómetro que realiza el automóvil, ajuste de tarifas y finalmente actualización de la hora y fecha. Los cambios se llevan a cabo mediante una interfaz desarrollada en Ubuntu. Cabe mencionar que todas estas modificaciones obedecen a la norma INEN 2663:2013 citada en [3].

Se determinó la funcionalidad mediante los manuales de usuario de cinco taxímetros existentes en el mercado como: Intelligent taximeter, S700, M1 plus, Optronic TX-10 y Tango XP. Luego de leer sus principales características se pudo concluir que cuatro de los taxímetros mencionados anteriormente usan el protocolo por cable RS232 y el restante se comunica por medio de un módulo llamado JTAG que se conecta a un puerto COM de la computadora. El uso de este tipo de comunicación resulta incómodo para los propietarios del medio de transporte que deben desmantelar el dispositivo cada vez que se realice una modificación en los parámetros, por tal motivo el sistema desarrollado tiene comunicación inalámbrica.

Se desarrolló un algoritmo que realiza las configuraciones de manera inalámbrica mediante el módulo ESP8266 que fue configurado como servidor usando el protocolo TCP-IP, permitiendo al sistema el intercambio de información entre computador que contiene la interfaz gráfica y la placa Arduino Mega 2560 encargada de ejecutar todo el proceso cuando el taxímetro se encuentre en operación.

La interfaz se realizó utilizando el módulo Tkinter que es la interfaz estándar del lenguaje de programación Python, ese módulo permitió crear una aplicación que interactúa amigablemente con el usuario sin olvidar las medidas de seguridad que eviten cambios accidentales de los parámetros.

Toda la funcionalidad del sistema computacional puede ser evidenciada mediante la pantalla del taxímetro la cual muestra las configuraciones que haga el usuario por medio de la interfaz gráfica y el taxímetro está listo para entrar en modo de trabajo.

## **5.2. Recomendaciones**

- Antes de realizar la comunicación verificar que la velocidad en baudios del módulo ESP8266 y la placa Arduino sea la misma ya que de esta manera se logra una transferencia de datos sin pérdida de información.
- El tamaño del buffer serial del microcontrolador es un aspecto fundamental para determinar la cantidad de información que se puede enviar desde la interfaz gráfica, para evitar inconvenientes se recomienda enviar los datos por partes, organizando la información de la interfaz en diferentes pestañas.
- Verificar que el protocolo de comunicación entre microcontrolador y computador sea realizado correctamente.

### **5.3. Trabajo futuro**

Se podría implementar en las compañías distribuidoras la tecnología IoT (Internet of Things), y de esta manera se lograría monitorear desde la empresa cualquier cambio anormal que pudiera presentarse en el taxímetro o realizar las configuraciones optimizando tiempo tanto para el propietario del medio de transporte como para el proveedor del dispositivo.

# Bibliografía

- [1] Resolución N°020-DIR-2013-ANT. *Reglamento de aplicación para la homologación, instalación y uso del taxímetro en el servicio de transporte comercial en taxis convencionales y ejecutivos*. Quito, Ecuador, 2013.
- [2] M. Cuasapaz, “Construcción e implementación de un prototipo de un taxímetro digital con impresora facturadora utilizando tecnología GPS y el desarrollo de un firmware en un microcontrolador”, Tesis Pregrado, Escuela Politécnica Nacional, Quito, Ecuador, Abril.2013.
- [3] NTE INEN 2663:2013, *Taxímetros. Requisitos metrológicos y técnicos, procedimientos de ensayo*. Quito, Ecuador, 2013.
- [4] B. Wu, and A. An, “FPGA Design and Implementation of Taximeter Anti -Fraud System”, in *IEEE Youth Academic Annual Conference of Chinese Association of Automation.*, pp. 654–659, May.2018.
- [5] C. Morales, “Diseño, construcción e implementación de un taxímetro con almacenamiento de viajes mediante GPS en el vehículo Chevrolet corsa wind 1.4 ”, Tesis Pregrado, Universidad Técnica del Norte, Ibarra, Ecuador, Marzo.2016.
- [6] Agencia Nacional de Tránsito (Diciembre, 2018).[Online]. Available: <https://www.ant.gob.ec/index.php/descargable/file/6007-direcciones-de-empresas-distribuidoras-autorizadas-por-la-ant-al-19-12-2018>
- [7] Centrodyne Inc.(2013).[Online]. Available: <http://www.centrodyne.com/es/products/taximeters/s700>

- [8] Digitax Automotive Electronics (2019).[Online]. Available: <http://www.digitax.com/products/Taximetro-M1.html>
- [9] Optronic (2014).[Online]. Available: <http://optroniconline.com/category/taximetros/>
- [10] Ful-Mar S.A.(2008-2019).[Online]. Available: <http://www.ful-mar.com.ar/es/producto.php?producto=2—6>
- [11] Y. Martinez and C. Córdoba, “Diseño de Interfaz de Usuario para Creación de Sistemas Multimedia para Apoyar el Desarrollo del Lenguaje”, in *Tecnología y Diseño*, vol. 9, pp. 39-55, 2018.
- [12] J. Preece, Y. Rogers and H. Sharp, “Interaction Design: beyond human-computer interaction”. G. Crockett, Ed. Estados Unidos de América: John Wiley y Sons, Inc., 2002.
- [13] I. Sommerville “Ingeniería del Software”. 7ma.ed. Madrid: Pearson Educación. S.A., 2005, pp. 344-346.
- [14] L. Luna, “El diseño de interfaz gráfica de usuario para publicaciones digitales”, in *Revista UNAM*, vol. 5, no. 7, pp. 7-12, 2004.
- [15] A. Tulenkov *et al.*, “The Features of Wireless Technologies Application for Smart House Systems”, in *4th IEEE International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems*, Lviv, Ukraine, September, 2018.
- [16] International Standard ISO/IEC/IEEE 18880, *Information technology — Ubiquitous green community control network protocol*. New York, USA 2015.
- [17] D. Cuji, D. Pazmiño, “Diseño e implementación de una red WSN (Wireless Sensor Network) basado en los protocolos Zigbee, Wifi y Zigbee Mesh, para el monitoreo de variables climáticas en el invernadero ubicado en el barrio Rumipamba del navas, cantón Salcedo, provincia de Cotopaxi. ”, Tesis Pregrado, Universidad de las Fuerzas Armadas - ESPE, Latacunga, Ecuador, Julio.2015.



- [18] M. León, ‘Wireless hacking, protegido con IDS, firewalls y Honeypots. ’, Tesis Pregrado, Universidad Tecnológica Israel, Cuenca, Ecuador, 2011.
- [19] W. Stallings “Comunicaciones y Redes de Computadores”. 7ma.ed. Madrid: Pearson Educación, S.A., 2004, pp. 29-31,40-41.
- [20] A. Tanenbaum, D Wetherall “Redes de Computadoras”. 5ta.ed. México: Pearson Educación, S.A., 2012, pp. 36-40.
- [21] K. Mahabaleshwarkar, N. Mundada, A. Chavan and A Panage, “TCP/IP Protocol Acceleration”, in *International Conference on Computer Communication and Informatics*, Coimbatore, India, January, 2012.
- [22] M. Xue, C. Zhu, “The Socket Programming and Software Design for Communication Based on Client/Server”, in *Pacific-Asia Conference on Circuits, Communications and System*, 2009.
- [23] M. Donahoo, and K Calvert, “TCP/IP Sockets in C”, 2da.ed. USA: Elsevier Inc, 2009, pp. 1-8, 53-54.
- [24] I. Sommerville “Ingeniería del Software”. 9na.ed. México: Pearson Educación. S.A., 2011, pp. 85-39.
- [25] L. Cysneiros, E. Yu, “Non-Functional Requirements Elicitation”, USA, January, 2004.
- [26] H. Cervantes, P. Velasco, L. Careaga “Arquitectura de Software: Conceptos y ciclo de desarrollo”, México, January, 2015.
- [27] B. Al Housani, B. Mutrib and H. Jaradi, “The Linux Review-Ubuntu Desktop Edition-version 8.10”, in *International Conference on the Current Trends in Information Technology (CTIT)*, Dubai, 2009, pp. 1-6.
- [28] M. Sanner, “Python: A Programming Language for Software Integration and Development”, in *Journal of Molecular Graphics and Modelling*, November, 1998.

- [29] Python, “Graphical User Interfaces with Tk”, 2019. [En línea], Disponible en:<https://docs.python.org/3/library/tk.html>. [Accedido: 06-Marzo-2019].
- [30] Python, “datetime — Basic date and time types”, 2019. [En línea], Disponible en:<https://docs.python.org/3/library/datetime.html>. [Accedido: 06-Marzo-2019].
- [31] Python, “calendar — General calendar-related functions”, 2019. [En línea], Disponible en:<https://docs.python.org/3/library/calendar.html#module-calendar>. [Accedido: 06-Marzo-2019].
- [32] Python, “time — Time access and conversions”, 2019. [En línea], Disponible en:<https://docs.python.org/3/library/time.html#module-time>. [Accedido: 06-Marzo-2019].
- [33] Python, “socket — Low-level networking interface”, 2019. [En línea], Disponible en:<https://docs.python.org/3/library/socket.html>. [Accedido: 06-Marzo-2019].
- [34] Python, “sys — System-specific parameters and functions”, 2019. [En línea], Disponible en:<https://docs.python.org/3/library/sys.html>. [Accedido: 06-Marzo-2019].
- [35] Python, “os — Miscellaneous operating system interfaces”, 2019. [En línea], Disponible en:<https://docs.python.org/3/library/os.html>. [Accedido: 06-Marzo-2019].
- [36] Python, “threading — Thread-based parallelism”, 2019. [En línea], Disponible en:<https://docs.python.org/3/library/threading.html>. [Accedido: 06-Marzo-2019].
- [37] G. Atethortúa, “Programación de sistemas embebidos en C: teoría y prácticas aplicadas a cualquier microcontrolador”. Ed. México: Alfaomega Grupo Editor, 2009.

# Apéndice

Este apéndice incluye el código de la interfaz gráfica que incluyen las funciones que realiza la Plataforma Abierta para Desarrollo de Dispositivos de Tarifación vehicular: "Software para Ajustes"

## **.A. Software**

### **.A.1. Interfaz Gráfica (taximetro.py)**

---

#### Programa 1: Interfaz gráfica

---

```
import socket #libreria para usar socket
import sys
import os
from tkinter import * #libreria para crear interfaz en tkinter
from tkinter import ttk
from threading import Thread #permite que dos o mas tareas se puedan realizar simultaneamente
from datetime import datetime, date, time, timedelta
import calendar
from tkinter.messagebox import showinfo
from tkinter.messagebox import showerror
import sqlite3

ahora = datetime.now() # Obtiene fecha y hora actual
fecha=date.today()
hoy=fecha.strftime ("%d/ %m/ %Y")

host, port = "192.168.4.1", 80 #ip del esp8266 puerto para la comunicacion tcp/ip
BUFSIZ = 1024 #cantidad de bytes
client = socket.socket(socket.AF_INET, socket.SOCK_STREAM) #parametros establecidos para crear
    un socket que utiliza comunicacion tcp
client.connect((host, port)) #socket client se conecta al servidor creado por arduino

data1=sys.argv[1:] #primer dato que recibe desde el arduino
data2=(sys.argv[2:]) #segundo dato que recibe desde el arduino
data3=(sys.argv[3:]) #tercer dato que recibe desde el arduino
data4=(sys.argv[4:]) #cuarto dato que recibe desde el arduino etc ,etc
data5=(sys.argv[5:])
data6=(sys.argv[6:])
```

```

data7=(sys.argv[7:])
data8=(sys.argv[8:])
data9=(sys.argv[9:])
data10=(sys.argv[10:])
def lectura1():
    valor='2'+':'
    try:
        message = valor.encode()
        print('sending {!r}'.format(message))
        client.send(message)
        amount_received=0
        amount_expected=len(message)

        while amount_received<amount_expected:

            data1=client.recv(BUFSIZ).decode()
            data2=client.recv(BUFSIZ).decode()
            data3=client.recv(BUFSIZ).decode()
            data4=client.recv(BUFSIZ).decode()
            data5=client.recv(BUFSIZ).decode()

            amount_received += len(message)

            comp.insert(END,data1) #escribe en el cuadro de texto
                la variable data
            ciu.insert(END,data2)
            ruc.insert(END,data3)
            aut.insert(END,data4)
            prop.insert(END,data5)

    finally:
        pass

def lectura2(): #la tarea receive
    valor='3'+':'
    try:
        message = valor.encode()
        print('sending {!r}'.format(message))
        client.send(message)
        amount_received=0
        amount_expected=len(message)

```

```

while amount_received < amount_expected :

    data1=client.recv(BUFSIZ).decode()
    data2=client.recv(BUFSIZ).decode()
    data3=client.recv(BUFSIZ).decode()
    data4=client.recv(BUFSIZ).decode()
    data5=client.recv(BUFSIZ).decode()
    data6=client.recv(BUFSIZ).decode()

    amount_received += len(message)

    ser.insert(END, data1)
    k.insert(END, data2)
    vcr.insert(END, data3)
    vmax.insert(END, data4)
    pla.insert(END, data5)
    nreg.insert(END, data6)

finally :
    pass

def lectura3(): #1a tarea receive
    valor='4'+':'
    try :
        message = valor.encode()
        print('sending {!r}'.format(message))
        client.send(message)
        amount_received=0
        amount_expected=len(message)

        while amount_received < amount_expected :

            data1=client.recv(BUFSIZ).decode()
            data2=client.recv(BUFSIZ).decode()
            data3=client.recv(BUFSIZ).decode()
            data4=client.recv(BUFSIZ).decode()
            data5=client.recv(BUFSIZ).decode()

            amount_received += len(message)

            master.insert(END, data1)
            user1.insert(END, data2)
            con1.insert(END, data3)
            user2.insert(END, data4)

```

```

con2.insert(END, data5)

finally:
    pass

def lectura4(): #1a tarea receive
    valor='5'+':'
    try:
        message = valor.encode()
        print('sending {!r}'.format(message))
        client.send(message)
        amount_received=0
        amount_expected=len(message)

        while amount_received<amount_expected:

            data1=client.recv(BUFSIZ).decode()
            data2=client.recv(BUFSIZ).decode()
            data3=client.recv(BUFSIZ).decode()
            data4=client.recv(BUFSIZ).decode()
            data5=client.recv(BUFSIZ).decode()
            data6=client.recv(BUFSIZ).decode()
            data7=client.recv(BUFSIZ).decode()
            data8=client.recv(BUFSIZ).decode()
            data9=client.recv(BUFSIZ).decode()
            data10=client.recv(BUFSIZ).decode()

            amount_received += len(message)

            tm1.insert(END, data1)
            ta1.insert(END, data2)
            Vk1.insert(END, data3)
            me1.insert(END, data4)
            ex1.insert(END, data5)
            tm2.insert(END, data6)
            ta2.insert(END, data7)
            Vk2.insert(END, data8)
            me2.insert(END, data9)
            ex2.insert(END, data10)

    finally:
        pass

def lectura5(): #1a tarea receive
    valor='6'+':'

```

```

try:
    message = valor.encode()
    print('sending {!r}'.format(message))
    client.send(message)
    amount_received=0
    amount_expected=len(message)

    while amount_received<amount_expected:

        data1=client.recv(BUFSIZ).decode()
        data2=client.recv(BUFSIZ).decode()
        data3=client.recv(BUFSIZ).decode()
        data4=client.recv(BUFSIZ).decode()
        data5=client.recv(BUFSIZ).decode()

        amount_received += len(message)

        frevday.configure(state='normal')
        frevmonth.configure(state='normal')
        frevyear.configure(state='normal')
        furday.configure(state='normal')
        contador1.configure(state='normal')
        frevday.insert(END, data1)
        frevmonth.insert(END, data2)
        frevyear.insert(END, data3)
        furday.insert(END, data4)
        contador1.insert(END, data5)
        frevday.configure(state='disabled')
        frevmonth.configure(state='disabled')
        frevyear.configure(state='disabled')
        furday.configure(state='disabled')
        contador1.configure(state='disabled')

    finally:
        pass

def enviar1(): #cuando se presiona la tecla enviar hace lo siguiente
    valor='7'+':'+comp.get()+':'+ciu.get()+':'+ruc.get()+':'+aut.get()+':'+prop.get()
    try:
        dato=valor.encode()#transforma el string "valor" en bytes y lo guarda en dato
        # client.sendall(dato) #envia el valor hacia el servidor (arduino)
        client.send((dato))
        #print('sending {!r}'.format(dato))

```



```

        print('sending {}'.format(dato))

    finally:
        pass

def enviar2(): #cuando se presiona la tecla enviar hace lo siguiente
    valor='8'+':'+ser.get()+':'+k.get()+':'+vcr.get()+':'+vmax.get()+':'+pla.get()+':'+
        nreg.get()
    try:
        dato=valor.encode()
        # client.sendall(dato)
        client.send((dato))
        #print('sending {}'.format(dato))
        print('sending {}'.format(dato))

    finally:
        pass

def enviar3(): #cuando se presiona la tecla enviar hace lo siguiente
    valor='9'+':'+master.get()+':'+user1.get()+':'+con1.get()+':'+user2.get()+':'+con2.get
    ()
    try:
        dato=valor.encode()
        # client.sendall(dato)
        client.send((dato))
        #print('sending {}'.format(dato))
        print('sending {}'.format(dato))

    finally:
        pass

def enviar4(): #cuando se presiona la tecla enviar hace lo siguiente
    valor='10'+':'+tm1.get()+':'+ta1.get()+':'+Vk1.get()+':'+me1.get()+':'+ex1.get()+':'+
        tm2.get()+':'+ta2.get()+':'+Vk2.get()+':'+me2.get()+':'+ex2.get()
    try:
        dato=valor.encode()
        # client.sendall(dato)
        client.send((dato))
        #print('sending {}'.format(dato))
        print('sending {}'.format(dato))

    finally:
        pass

```

```

def enviar5(): #cuando se presiona la tecla enviar hace lo siguiente
    furday.configure(state='normal')
    furday.delete(0,END)
    furday.insert(END,hoy)
    furday.configure(state='disabled')
    contador1.configure(state='normal')

    valor='11'+ ':' +day.get()+ ':' +month.get()+ ':' +year.get()+ ':' +hora.get()+ ':' +minuto.get
        (+)'+ ':' +segundo.get()+ ':' +frevday.get()+ ':' +frevmonth.get()+ ':' +frevyear.get()+ ':' +
        furday.get()+ ':' +contador1.get()+ ':' +responsable.get()

    try:
        dato=valor.encode()
        # client.sendall(dato)
        client.send((dato))
        #print('sending {}'.format(dato))
        contador1.configure(state='disabled')
        print('sending {}'.format(dato))

    finally:
        pass

def click():
    frevday.config(state='normal')
    frevmonth.config(state='normal')
    frevyear.config(state='normal')

def click1():
    day.config(state='normal')
    month.config(state='normal')
    year.config(state='normal')
    hora.config(state='normal')
    minuto.config(state='normal')
    segundo.config(state='normal')

def borrar1():
    comp.delete(0,END)
    ciu.delete(0,END)
    ruc.delete(0,END)
    aut.delete(0,END)
    prop.delete(0,END)

def borrar2():

```

```

    ser.delete(0,END)
    k.delete(0,END)
    vcr.delete(0,END)
    vmax.delete(0,END)
    pla.delete(0,END)
    nreg.delete(0,END)

def borrar3():
    master.delete(0,END)
    user1.delete(0,END)
    con1.delete(0,END)
    user2.delete(0,END)
    con2.delete(0,END)

def borrar4():
    tm1.delete(0,END)
    ta1.delete(0,END)
    Vk1.delete(0,END)
    me1.delete(0,END)
    ex1.delete(0,END)
    tm2.delete(0,END)
    ta2.delete(0,END)
    Vk2.delete(0,END)
    me2.delete(0,END)
    ex2.delete(0,END)

def Salir():
    valor= '12'+':':
    try:
        dato=valor.encode()
        # client.sendall(dato)
        client.send((dato))
        #print('sending {}'.format(dato))
        print('sending {}'.format(dato))
    finally:
        ventana.quit() #cierra la interfaz
        client.close() #cierra la conexion del socket

def login():
    # Connect to database
    db = sqlite3.connect('/home/hector/Documentos/usuarios/login.db')
    c = db.cursor()
    usuario = cajal.get()

```

```

    contr = caja2.get()
    c.execute('SELECT * FROM usuarios WHERE usuario = ? AND pass = ?', (usuario , contr))

    if c.fetchall():
        showinfo(title = "Login correcto", message = "Usuario y contrasena correctos")
        ingreso=caja1.get()
        caja1.configure(state='disabled')
        caja2.configure(state='disabled')
        botonlogin.place_forget()
        ventana2.destroy()

    else:
        showerror(title = "Login incorrecto", message = "Usuario o contrasena
            incorrecta")
        Salir()
    c.close()

ventana2 = Tk() #Ventana
ventana2.title("Interfaz de Usuario: Taximetro UTN")
ventana2.geometry('270x150')

Label(ventana2, text = "Usuario:").pack()

caja1 = Entry(ventana2)
caja1.pack()

Label(ventana2, text = "Contrasena:").pack()
caja2 = Entry(ventana2, show = "*")
caja2.pack()
botonlogin = Button (text = "Login", command = login)
botonlogin.place(x=100, y=100)
ventana2.mainloop() #ejecuta la ventana

ventana = Tk() #Ventana
ventana.title("Interfaz de Usuario: Taximetro UTN")
ventana.geometry('470x420')
informacion=ttk.Notebook(ventana)
informacion.pack(fill='both',expand='yes')
pes1=ttk.Frame(informacion)
pes2=ttk.Frame(informacion)
pes3=ttk.Frame(informacion)
pes4=ttk.Frame(informacion)

```

```

pes5=ttk.Frame(informacion)
informacion.add(pes1, text='Informacion General')
informacion.add(pes2, text='Parametros')
informacion.add(pes3, text='Usuarios')
informacion.add(pes4, text='Tarifas')
informacion.add(pes5, text='Servicio Tecnico')
version = Label(informacion, text="Version 1.0",font=("Helvetica",10))
version.place(x=395,y=390)

info=Frame(pes1)
info.configure('disable')
info=LabelFrame(pes1, text="Informacion General",font=("Helvetica",12),padx=5, pady=5)
info.grid(column=0,row=0,padx=(10,10),pady=(5,5),sticky=W)

etiqueta = Label(info, text="Compania:",font=("Helvetica",12))
etiqueta.grid(column=0, row=0,sticky=W)
comp = Entry(info, width=30,font=("Helvetica",8))
comp.grid(column=1, row=0,sticky=W,pady=(3,3))

etiqueta = Label(info, text="Ciudad:",font=("Helvetica",12))
etiqueta.grid(column=0, row=1,sticky=W)
ciu = Entry(info, width=30,font=("Helvetica",8))
ciu.grid(column=1, row=1,sticky=W)

etiqueta = Label(info, text="RUC:",font=("Helvetica",12))
etiqueta.grid(column=0, row=2,sticky=W)
ruc = Entry(info, width=30,font=("Helvetica",8))
ruc.grid(column=1, row=2,sticky=W)

etiqueta = Label(info, text="Aut. SRI:",font=("Helvetica",12))
etiqueta.grid(column=0, row=3,sticky=W)
aut = Entry(info, width=30,font=("Helvetica",8))
aut.grid(column=1, row=3,sticky=W)

etiqueta = Label(info, text="Propietario:",font=("Helvetica",12))
etiqueta.grid(column=0, row=4,sticky=W)
prop = Entry(info, width=30,font=("Helvetica",8))
prop.grid(column=1, row=4,sticky=W)

info1=Frame(pes2)
info1=LabelFrame(pes2, text="Parametros",font=("Helvetica",12),padx=5, pady=5)
info1.grid(column=0,row=0,padx=(10,10),pady=(5,5),sticky=NW, rowspan=2)

```

```

etiqueta1 = Label(info1 , text="Num. de Serie:",font=("Helvetica",12))
etiqueta1.grid(column=0, row=0,sticky=W)
ser = Entry(info1 , width=20,font=("Helvetica",8))
ser.grid(column=1, row=0,sticky=W)

etiqueta1 = Label(info1 , text="Constante K:",font=("Helvetica",12))
etiqueta1.grid(column=0, row=1,sticky=W)
k = Entry(info1 , width=20,font=("Helvetica",8))
k.grid(column=1, row=1,sticky=W)

etiqueta2 = Label(info1 , text="Velocidad Critica:",font=("Helvetica",12))
etiqueta2.grid(column=0, row=2,sticky=W)
vcr = Entry(info1 , width=20,font=("Helvetica",8))
vcr.grid(column=1, row=2,sticky=W)

etiqueta2 = Label(info1 , text="Velocidad Maxima:",font=("Helvetica",12))
etiqueta2.grid(column=0, row=3,sticky=W)
vmax = Entry(info1 , width=20,font=("Helvetica",8))
vmax.grid(column=1, row=3,sticky=W)

etiqueta1 = Label(info1 , text="Placa:",font=("Helvetica",12))
etiqueta1.grid(column=0, row=4,sticky=W)
pla = Entry(info1 , width=20,font=("Helvetica",8))
pla.grid(column=1, row=4,sticky=W)

etiqueta2 = Label(info1 , text="Num.de registro:",font=("Helvetica",12))
etiqueta2.grid(column=0, row=5,sticky=W)
nreg = Entry(info1 , width=20,font=("Helvetica",8))
nreg.grid(column=1, row=5,sticky=W)

user=Frame(pes3)
user=LabelFrame(pes3 , text="Usuarios",font=("Helvetica",12),padx=5, pady=5)
user.grid(column=0,row=0,padx=(10,0),pady=(5,5),sticky=NW)

etiqueta1 = Label(user , text="Llave Master:",font=("Helvetica",12))
etiqueta1.grid(column=0, row=0,sticky=W)
master = Entry(user , width=20,font=("Helvetica",8))
master.grid(column=1, row=0,sticky=W)

etiqueta1 = Label(user , text="Chofer 1:",font=("Helvetica",12))
etiqueta1.grid(column=0, row=1,sticky=W)
user1= Entry(user , width=20,font=("Helvetica",8))
user1.grid(column=1, row=1,sticky=W)

```

```

etiqueta2 = Label(user , text="Contrasena 1:",font=("Helvetica",12))
etiqueta2.grid(column=0, row=2,sticky=W)
con1= Entry (user , width=20,font=("Helvetica",8))
con1.grid(column=1, row=2,sticky=W)

etiqueta2 = Label(user , text="Chofer 2:",font=("Helvetica",12))
etiqueta2.grid(column=0, row=3,sticky=W)
user2 = Entry (user , width=20,font=("Helvetica",8))
user2.grid(column=1, row=3,sticky=W)

etiqueta1 = Label(user , text="Contrasena 2:",font=("Helvetica",12))
etiqueta1.grid(column=0, row=4,sticky=W)
con2= Entry (user , width=20,font=("Helvetica",8))
con2.grid(column=1, row=4,sticky=W)

vp=Frame(pes4)
vp=LabelFrame(pes4 , text="Tarifas",font=("Helvetica"))
vp.grid(column=0,row=0,sticky=NW,rowspan=4)

tarifa=Frame(vp)
tarifa=LabelFrame(vp, text="Tarifa Diurna",font=("Helvetica",12),padx=5, pady=5)
tarifa.grid(column=0,row=0,padx=(10,10),pady=(5,5),sticky=NW)

etiqueta = Label(tarifa , text="Tarifa minima")
etiqueta.grid(column=0, row=0,sticky=W)
tm1 = Entry (tarifa , width=10) #cuadros de texto
tm1.grid(column=1, row=0,sticky=W)

etiqueta1 = Label(tarifa , text="Arranque")
etiqueta1.grid(column=0, row=1,sticky=W)
ta1 = Entry (tarifa , width=10)
ta1.grid(column=1, row=1,sticky=W)

etiqueta2 = Label(tarifa , text="Valor por Km")
etiqueta2.grid(column=0, row=2,sticky=W)
Vkl = Entry (tarifa , width=10)
Vkl.grid(column=1, row=2,sticky=W)

etiqueta3 = Label(tarifa , text="Minuto de espera")
etiqueta3.grid(column=0, row=3,sticky=W)

```

```

mel = Entry(tarifa , width=10)
mel.grid(column=1, row=3, sticky=W)

etiqueta3 = Label(tarifa , text="Extras")
etiqueta3.grid(column=0, row=4, sticky=W)
ex1 = Entry(tarifa , width=10)
ex1.grid(column=1, row=4, sticky=W)

tarifa1=Frame(vp)
tarifa1=LabelFrame(vp, text="Tarifa Nocturna y Feriados", font=("Helvetica",12), padx=5, pady=5)
tarifa1.grid(column=1, row=0, padx=(10,10), pady=(5,5), sticky=SW)

etiqueta = Label(tarifa1 , text="Tarifa minima")
etiqueta.grid(column=0, row=0, sticky=W)
tm2 = Entry(tarifa1 , width=10) #cuadros de texto
tm2.grid(column=1, row=0, sticky=W)

etiqueta1 = Label(tarifa1 , text="Arranque")
etiqueta1.grid(column=0, row=1, sticky=W)
ta2 = Entry(tarifa1 , width=10)
ta2.grid(column=1, row=1, sticky=W)

etiqueta2 = Label(tarifa1 , text="Valor por Km")
etiqueta2.grid(column=0, row=2, sticky=W)
Vk2 = Entry(tarifa1 , width=10)
Vk2.grid(column=1, row=2, sticky=W)

etiqueta3 = Label(tarifa1 , text="Minuto de espera")
etiqueta3.grid(column=0, row=3, sticky=W)
me2 = Entry(tarifa1 , width=10)
me2.grid(column=1, row=3, sticky=W)

etiqueta3 = Label(tarifa1 , text="Extras")
etiqueta3.grid(column=0, row=4, sticky=W)
ex2 = Entry(tarifa1 , width=10)
ex2.grid(column=1, row=4, sticky=W)

va=Frame(pes5)
va=LabelFrame(pes5 , text="Fecha y Hora", font=("Helvetica",12), padx=5, pady=5)
va.grid(column=0, row=0, padx=(10,10), pady=(5,5), sticky=NW)
check1 = Checkbutton(va, text="Configuracion Manual", command=click1)
check1.grid(column=0, row=0, sticky=W, columns=4)

```



```

etiqueta = Label(va, text="dd/MM/aaaa")
etiqueta.grid(column=1, row=1, sticky=W)

day = Entry(va, width=5, fg="black")
day.insert(END, ahora.day)
day.configure(state='disabled')
day.grid(column=2, row=1, sticky=W)

month = Entry(va, width=5)
month.insert(END, ahora.month)
month.configure(state='disabled')
month.grid(column=3, row=1, sticky=W)

year = Entry(va, width=5)
year.insert(END, ahora.year)
year.configure(state='disabled')
year.grid(column=4, row=1, sticky=W)

etiqueta = Label(va, text="H:mm:ss")
etiqueta.grid(column=1, row=2, sticky=W)

hora = Entry(va, width=5)
hora.insert(END, ahora.hour)
hora.configure(state='disabled')
hora.grid(column=2, row=2, sticky=W)

minuto = Entry(va, width=5)
minuto.insert(END, ahora.minute)
minuto.configure(state='disabled')
minuto.grid(column=3, row=2, sticky=W)

segundo = Entry(va, width=5)
segundo.insert(END, ahora.second)
segundo.configure(state='disabled')
segundo.grid(column=4, row=2, sticky=W)

Frev=Frame(pes5)
Frev=LabelFrame(pes5, text="Servicio Tecnico", font=("Helvetica", 12), padx=5, pady=5)
Frev.grid(column=0, row=1, padx=(10, 10), pady=(5, 5), sticky=SW)

check = Checkbutton(Frev, text="Fecha de proxima inspeccion", command=click)
check.grid(column=0, row=0, sticky=W, colspan=4)

```

```

etiqueta = Label(Frev , text="dd/MM/aaaa")
etiqueta.grid(column=0, row=1, sticky=W)

frevday = Entry(Frev , width=5, state='disabled')
frevday.grid(column=1, row=1, sticky=W)

frevmonth = Entry(Frev , width=5, state='disabled')
frevmonth.grid(column=2, row=1, sticky=W)

frevyear = Entry(Frev , width=5, state='disabled')
frevyear.grid(column=3, row=1, sticky=W)

etiqueta = Label(Frev , text="ultima revision tecnica", font=("Helvetica",12), pady=5)
etiqueta.grid(column=0, row=2, sticky=W, columnspan=4)
etiqueta = Label(Frev , text="dd/MM/aaaa")
etiqueta.grid(column=0, row=3, sticky=W)

furday = Entry(Frev , width=20)
furday.grid(column=1, row=3, sticky=W, columnspan=4)
furday.configure(state='disabled')

etiqueta = Label(Frev , text="Contador de eventos")
etiqueta.grid(column=0, row=4, sticky=W)

contador1 = Entry(Frev , width=20)
contador1.grid(column=1, row=4, sticky=W, columnspan=4)
contador1.configure(state='disabled')

etiqueta = Label(Frev , text="Responsable")
etiqueta.grid(column=0, row=5, sticky=W)

responsable = Entry(Frev , width=20)
responsable.grid(column=1, row=5, sticky=W, columnspan=4)

botones=Frame(pes1)
botones=LabelFrame(pes1 , text="Comunicacion", font=("Helvetica",12), padx=5, pady=5)
botones.grid(column=0, row=2, padx=(10,10), pady=(10,10), rowspan=2)

boton=Button(botones , text="Lectura", command=lectural) #boton
boton.grid(column=0, row=1, padx=(10,10), pady=(10,10))

boton2=Button(botones , text="Enviar", command=enviar1)#boton
boton2.grid(column=1, row=1, padx=(10,10), pady=(10,10))

```

```

boton2=Button ( botones , text="Borrar" ,command=borrar1 )#boton
boton2 . grid ( column=2,row=1 ,padx=(10,10) ,pady=(10,10))

boton3=Button ( botones , text="Salir" ,command=Salir , activebackground="lavender")#boton
boton3 . grid ( column=3,row=1 ,padx=(10,10) ,pady=(10,10))

botones=Frame ( pes2 )
botones=LabelFrame ( pes2 , text="Comunicacion" ,font=("Helvetica" ,12) ,padx=5, pady=5)
botones . grid ( column=0,row=2 ,padx=(10,10) ,pady=(10,10) ,rowspan=2)

boton=Button ( botones , text="Lectura" ,command=lectura2 ) #boton
boton . grid ( column=0,row=1 ,padx=(10,10) ,pady=(10,10))

boton2=Button ( botones , text="Enviar" ,command=enviar2)#boton
boton2 . grid ( column=1,row=1 ,padx=(10,10) ,pady=(10,10))

boton2=Button ( botones , text="Borrar" ,command=borrar2)#boton
boton2 . grid ( column=2,row=1 ,padx=(10,10) ,pady=(10,10))

boton3=Button ( botones , text="Salir" ,command=Salir)#boton
boton3 . grid ( column=3,row=1 ,padx=(10,10) ,pady=(10,10))

botones=Frame ( pes3 )
botones=LabelFrame ( pes3 , text="Comunicacion" ,font=("Helvetica" ,12) ,padx=5, pady=5)
botones . grid ( column=0,row=2 ,padx=(10,10) ,pady=(10,10) ,rowspan=2 ,sticky=↖)

boton=Button ( botones , text="Lectura" ,command=lectura3 ) #boton
boton . grid ( column=0,row=1 ,padx=(10,10) ,pady=(10,10))

boton2=Button ( botones , text="Enviar" ,command=enviar3)#boton
boton2 . grid ( column=1,row=1 ,padx=(10,10) ,pady=(10,10))

boton2=Button ( botones , text="Borrar" ,command=borrar3)#boton
boton2 . grid ( column=2,row=1 ,padx=(10,10) ,pady=(10,10))

boton3=Button ( botones , text="Salir" ,command=Salir)#boton
boton3 . grid ( column=3,row=1 ,padx=(10,10) ,pady=(10,10))

botones=Frame ( pes4 )
botones=LabelFrame ( pes4 , text="Comunicacion" ,font=("Helvetica" ,12) ,padx=5, pady=5)
botones . grid ( column=0,row=4 ,padx=(10,10) ,pady=(10,10) ,rowspan=2 ,sticky=↖)

```

```

boton=Button( botones , text="Lectura" ,command=lectura4 ) #boton
boton . grid ( column=0,row=1,padx=(10,10) ,pady=(10,10))

boton2=Button( botones , text="Enviar" ,command=enviar4)#boton
boton2 . grid ( column=1,row=1,padx=(10,10) ,pady=(10,10))

boton2=Button( botones , text="Borrar" ,command=borrar4)#boton
boton2 . grid ( column=2,row=1,padx=(10,10) ,pady=(10,10))

boton3=Button( botones , text="Salir" ,command=Salir)#boton
boton3 . grid ( column=3,row=1,padx=(10,10) ,pady=(10,10))

botones=Frame( pes5 )
botones=LabelFrame( pes5 , text="Comunicacion" ,font=("Helvetica" ,12) ,padx=5, pady=5)
botones . grid ( column=0,row=3,padx=(10,10) ,pady=(10,10) ,rowspan=2)

boton=Button( botones , text="Lectura" ,command=lectura5 ) #boton
boton . grid ( column=0,row=1,padx=(10,10) ,pady=(10,10))

boton2=Button( botones , text="Enviar" ,command=enviar5)#boton
boton2 . grid ( column=1,row=1,padx=(10,10) ,pady=(10,10))

boton3=Button( botones , text="Salir" ,command=Salir)#boton
boton3 . grid ( column=3,row=1,padx=(10,10) ,pady=(10,10))

ventana . mainloop ()
client . close ()

```

---

## **.A.2. Código Arduino (taximetro.ino)**

---

### Programa 2: Código arduino

---

```

int eeAddress = 1; // Compania char [50]
int eeAddress1 = 51; //Ciudad char [30]
int eeAddress2 = 81; //Ruc char [13]
int eeAddress3 = 95; //aut. char [10]
int eeAddress4 = 106; //Propietario char [45]

```

```

int eeAddress5 = 150; // Numero de serie [15]
int eeAddress6 = 165; //Constante K float 4 bytes
int eeAddress7 = 170; //Velocidad Critica int 2 bytes
int eeAddress8 = 172; //Velocidad Maxima int 2 bytes
int eeAddress9 = 174; //Placa char[10]
int eeAddress10 = 184; // Numero de registro char [10]

int eeAddress11=200; // Tarifa Diurna
int eeAddress12=205; //Arranque
int eeAddress13=210; //Valor por km
int eeAddress14=215; //Minuto de espera
int eeAddress15=220; //Extras

int eeAddress16=225; // Tarifa Nocturna
int eeAddress17=230; //Arranque
int eeAddress18=235; //Valor por km
int eeAddress19=240; //Minuto de espera
int eeAddress20=245; //Extras

int eeAddress21 = 250; //Llave Master [4]
int eeAddress22 = 255; //Chofer 1 [20]
int eeAddress23 = 275; //Contrasena 1 [4]
int eeAddress24 = 280; //Chofer 2 [20]
int eeAddress25 = 300; //Contrasena 2 [4]

int eeAddress26 = 310; // Dia de proxima revision int [2]
int eeAddress27 = 314; // Mes de proxima revision int [2]
int eeAddress28 = 318; // Ano de proxima revision int [2]
int eeAddress29 = 322; // Fecha de ultima revision int [12]

int eeAddress30 = 340; // contador de eventos int [2]
int eeAddress31 = 345; // Persona de verificacion char [20]
int eeAddress32 = 400; // Persona de verificacion char [20]
int eeAddress33 = 455; // contador de eventos compara int[2]

char tarifas [5];
int vlc;// vlm;
int aa, mes, dia, hora, minuto, segundo, eventos, eventos1;
int dr;//mr, ar;
int connectionId;
char compania[50];//char propietario[45];
char ciu [30];

```

```

char ruc[15]; //char aut[11]; //char numserie[15]; char fecha[12];
char placa[10]; //char numreg[10];
char master[5]; //char contra1[5]; char contra2[5];
char chofer1[20]; //char chofer2[20];
char compania1[50]="";
unsigned long tiempo1=0;
unsigned long tiempo2=0;
unsigned long tiempoSegundos=0;

char B[5];
char C[5];
String cad="";
String cad1="";
int weekDay;
int h, m, a, b;

float temp, temp1, temp2, arra, porkm, mines, ext, arra1, porkm1, mines1, ext1, td;
float arra2, porkm2, mines2, ext2, val1, val2, val3, val4;

void setup()
{
  pinMode(ENTER, INPUT_PULLUP);
  pinMode(ATRAS, INPUT_PULLUP);
  pinMode(ABAJO, INPUT_PULLUP);
  pinMode(DERECHA, INPUT_PULLUP);
  pinMode(IZQUIERDA, INPUT_PULLUP);
  pinMode(ARRIBA, INPUT_PULLUP);
  pinMode(PULSOS, INPUT_PULLUP);

  Serial.begin(115200); ///////For Serial monitor
  Serial.print("Iniciando SD ...");
  if (!SD.begin(4))
  {
    Serial.println("No se pudo inicializar");
    return;
  }
  Serial.println("inicializacion exitosa");
  if (!rtc.begin())
  {
    while (1);
  }
  if (rtc.lostPower())

```

```

    {
        rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
    }

    if (epd.Init(lut_full_update) != 0)
    {
        return;
    }
    epd.ClearFrameMemory(0xFF); // bit set = white, bit reset = black
    epd.DisplayFrame();
    epd.ClearFrameMemory(0xFF); // bit set = white, bit reset = black
    epd.DisplayFrame();

    if (epd.Init(lut_partial_update) != 0)
    {
        return;
    }
    tiempo1=millis();
    EEPROM.get(eeAddress4, compania);
    escritura();
}
void(*resetFunc)(void)=0;
void loop()
{
    ENter();
    ATras();
    left();
    right();
    ABajo();
    ARriba();
    if (configuracion==1)
    {
        sendData("AT+CWMODE=3\r\n", 200, DEBUG); // configure as access point
        sendData("AT+CIPMUX=1\r\n", 200, DEBUG); // configure for multiple connections
        sendData("AT+CIPSERVER=1,80\r\n", 200, DEBUG); // turn on server on port 80
        configuracion=2;
    }
    while(configuracion == 2)
    {
        if (Serial2.available())
        {
            if (Serial2.find("+IPD, "))
            {

```

```

delay(300);
connectionId = Serial2.read() - 48;
String cadena="";
String elemento1, elemento2, elemento3, elemento4, elemento5, elemento6, elemento7, elemento8,
    elemento9;
String elemento10, elemento11, elemento12, elemento13, elemento14;

while(Serial2.available())
{
    cadena=Serial2.readString();
    elemento1=s.separa(cadena,':',0);
    elemento2=s.separa(cadena,':',1);
    int variable=elemento2.toInt();

        if(variable==2)
        {
            epd.SetFrameMemory(IMAGE_RECEIVE);
            epd.DisplayFrame();
            String enviar1 = String(EEPROM.get(eeAddress, compania));
            espsend(enviar1);
            String enviar2 = String(EEPROM.get(eeAddress1, ciu));
            espsend(enviar2);
            String enviar3 = String(EEPROM.get(eeAddress2, ruc));
            espsend(enviar3);
            delay(50);
            String enviar4 = String(EEPROM.get(eeAddress3, ruc));
            espsend(enviar4);
            delay(50);
            String enviar5 = String(EEPROM.get(eeAddress4, compania));
            espsend(enviar5);
            delay(50);
        }
        if(variable==3)
        {
            String enviar6 = String(EEPROM.get(eeAddress5, ruc));
            espsend(enviar6);
            delay(50);
            String enviar7 = String(EEPROM.get(eeAddress6, td));
            espsend(enviar7);
            delay(50);
            String enviar8 = String(EEPROM.get(eeAddress7, vlc));
            espsend(enviar8);
            delay(50);
        }
    }
}

```



```

String enviar9 = String(EEPROM.get(eeAddress8 , vlc));
espsend(enviar9);
delay(50);
String enviar10 = String(EEPROM.get(eeAddress9 , placa));
espsend(enviar10);
delay(50);
String enviar11 = String(EEPROM.get(eeAddress10 , placa));
espsend(enviar11);
delay(50);
}
if(variable==4)
{
String enviar1 = String(EEPROM.get(eeAddress21 , master));
espsend(enviar1);
delay(50);
String enviar2 = String(EEPROM.get(eeAddress22 , chofer1));
espsend(enviar2);
delay(50);
String enviar3 = String(EEPROM.get(eeAddress23 , master));
espsend(enviar3);
delay(50);
String enviar4 = String(EEPROM.get(eeAddress24 , chofer1));
espsend(enviar4);
delay(50);
String enviar5 = String(EEPROM.get(eeAddress25 , master));
espsend(enviar5);
delay(50);
}
if(variable==5)
{
String enviar1 = String(EEPROM.get(eeAddress11 , td));
espsend(enviar1);
delay(50);
String enviar2 = String(EEPROM.get(eeAddress12 , td));
espsend(enviar2);
String enviar3 = String(EEPROM.get(eeAddress13 , td));
espsend(enviar3);
delay(50);
String enviar4 = String(EEPROM.get(eeAddress14 , td));
espsend(enviar4);
delay(50);
String enviar5 = String(EEPROM.get(eeAddress15 , td));
espsend(enviar5);

```

```

delay (50);
String enviar6 = String (EEPROM.get (eeAddress16 ,td));
espsend (enviar6);
delay (50);
String enviar7 = String (EEPROM.get (eeAddress17 ,td));
espsend (enviar7);
delay (50);
String enviar8 = String (EEPROM.get (eeAddress18 ,td));
espsend (enviar8);
delay (50);
String enviar9 = String (EEPROM.get (eeAddress19 ,td));
espsend (enviar9);
delay (50);
String enviar10 = String (EEPROM.get (eeAddress20 ,td));
espsend (enviar10);
delay (50);
}

if (variable==6)
{
String enviar1 = String (EEPROM.get (eeAddress26 ,dr));
espsend (enviar1);
delay (50);
String enviar2 = String (EEPROM.get (eeAddress27 ,dr));
espsend (enviar2);
delay (50);
String enviar3 = String (EEPROM.get (eeAddress28 ,dr));
espsend (enviar3);
delay (50);
String enviar4 = String (EEPROM.get (eeAddress29 ,ruc));
espsend (enviar4);
delay (50);
String enviar5 = String (EEPROM.get (eeAddress30 ,eventos));
espsend (enviar5);
delay (50);
}

```

*//////////////////////////////// recibir datos //////////////////////////////////*

```

if (variable==7)
{
epd . SetFrameMemory (IMAGE_ENVIO);
epd . DisplayFrame ();
elemento3=s . separa (cadena ,' : ' ,2);

```

```

elemento4=s.separa(cadena,':',3);
elemento5=s.separa(cadena,':',4);
elemento6=s.separa(cadena,':',5);
elemento7=s.separa(cadena,':',6);

elemento3.toCharArray(compania,50);
EEPROM.put(eeAddress,compania);
elemento4.toCharArray(ciu,30);
EEPROM.put(eeAddress1,ciu);
elemento5.toCharArray(ruc,15);
EEPROM.put(eeAddress2,ruc);
elemento6.toCharArray(ruc,15);
EEPROM.put(eeAddress3,ruc);//autorizacionsri
elemento7.toCharArray(compania,50);//propietario
EEPROM.put(eeAddress4,compania);

}
if(variable==8)
{
elemento3=s.separa(cadena,':',2);
elemento4=s.separa(cadena,':',3);
elemento5=s.separa(cadena,':',4);
elemento6=s.separa(cadena,':',5);
elemento7=s.separa(cadena,':',6);
elemento8=s.separa(cadena,':',7);

elemento3.toCharArray(ruc,15);
EEPROM.put(eeAddress5,ruc);//numero de serie
td=elemento4.toFloat();
//elemento4.toCharArray(tarifas,5);
EEPROM.put(eeAddress6,td);
vlc=elemento5.toInt();
EEPROM.put(eeAddress7,vlc);
vlc=elemento6.toInt();
EEPROM.put(eeAddress8,vlc);
elemento7.toCharArray(placa,10);
EEPROM.put(eeAddress9,placa);
elemento8.toCharArray(placa,10);
EEPROM.put(eeAddress10,placa);

}
if(variable==9)
{

```

```

elemento3=s.separa(cadena,':',2);
elemento4=s.separa(cadena,':',3);
elemento5=s.separa(cadena,':',4);
elemento6=s.separa(cadena,':',5);
elemento7=s.separa(cadena,':',6);

elemento3.toCharArray(master,5);
EEPROM.put(eeAddress21, master);
elemento4.toCharArray(chofer1,20);
EEPROM.put(eeAddress22, chofer1);
elemento5.toCharArray(master,5);
EEPROM.put(eeAddress23, master);
elemento6.toCharArray(chofer1,20);
EEPROM.put(eeAddress24, chofer1);
elemento7.toCharArray(master,5);
EEPROM.put(eeAddress25, master);
}
if(variable==10)
{
elemento3=s.separa(cadena,':',2);
elemento4=s.separa(cadena,':',3);
elemento5=s.separa(cadena,':',4);
elemento6=s.separa(cadena,':',5);
elemento7=s.separa(cadena,':',6);
elemento8=s.separa(cadena,':',7);
elemento9=s.separa(cadena,':',8);
elemento10=s.separa(cadena,':',9);
elemento11=s.separa(cadena,':',10);
elemento12=s.separa(cadena,':',11);

td=elemento3.toFloat();
EEPROM.put(eeAddress11, td);
td=elemento4.toFloat();
EEPROM.put(eeAddress12, td);
td=elemento5.toFloat();
EEPROM.put(eeAddress13, td);
td=elemento6.toFloat();
EEPROM.put(eeAddress14, td);
td=elemento7.toFloat();
EEPROM.put(eeAddress15, td);
td=elemento8.toFloat();
EEPROM.put(eeAddress16, td);

```

```

        td=elemento9.toFloat();
        EEPROM.put( eeAddress17 , td);
        td=elemento10.toFloat();
        EEPROM.put( eeAddress18 , td);
        td=elemento11.toFloat();
        EEPROM.put( eeAddress19 , td);
        td=elemento12.toFloat();
        EEPROM.put( eeAddress20 , td);
    }
    if (variable==11)
    {
        elemento3=s.separa(cadena,':',2);
        elemento4=s.separa(cadena,':',3);
        elemento5=s.separa(cadena,':',4);
        elemento6=s.separa(cadena,':',5);
        elemento7=s.separa(cadena,':',6);
        elemento8=s.separa(cadena,':',7);
        elemento9=s.separa(cadena,':',8);
        elemento10=s.separa(cadena,':',9);
        elemento11=s.separa(cadena,':',10);
        elemento12=s.separa(cadena,':',11);
        elemento13=s.separa(cadena,':',12);
        elemento14=s.separa(cadena,':',13);

        dia=elemento3.toInt();
        mes=elemento4.toInt();
        aa=elemento5.toInt();
        hora=elemento6.toInt();
        minuto=elemento7.toInt();
        segundo=elemento8.toInt();
        dr=elemento9.toInt();
        EEPROM.put( eeAddress26 , dr);
        dr=elemento10.toInt();
        EEPROM.put( eeAddress27 , dr);
        dr=elemento11.toInt();
        EEPROM.put( eeAddress28 , dr);
        elemento12.toCharArray( ruc ,15);
        EEPROM.put( eeAddress29 , ruc);
        eventos=elemento13.toInt();
        eventos=eventos+1;
        EEPROM.put( eeAddress30 , eventos);
        elemento14.toCharArray( compania ,50); //propietario
        EEPROM.put( eeAddress31 , compania);
    }
}

```

```

        rtc.adjust(DateTime(aa, mes, dia, hora, minuto, segundo));

    }
    if (variable==12)
    {
        ISRenter=0;
        derecha=1;
        Flagoperacion=false;
        inicio1();
        configuracion=3;
        resetFunc();
    }
}
}
}

if (Flagg==true)
{
    escritura();
    Flagg=false;
}
}
String sendData(String command, const int timeout, boolean debug)
{
    String response = "";
    Serial2.print(command);
    long int time = millis();
    while ( (time + timeout) > millis())
    {
        while (Serial2.available() > 0)
        {
            char c = Serial2.read(); // read the next character.
            response += c;
        }
    }
    if (debug)
    {
        Serial.print(response); //displays the esp response messages in arduino Serial2 monitor
    }
    return response;
}
}

```

```

void calendario()
{
    DateTime now = rtc.now();
    weekDay = now.dayOfTheWeek();
    h = now.hour();
    m = now.minute();
    if(weekDay == 6 || weekDay == 0)
    {
        a=2;
    }
    else if(weekDay ==1 || weekDay ==2 || weekDay==3|| weekDay==4 || weekDay==5)
    {
        if((h >= 6) || (h < 17))
        {
            a=1;
        }
        if((h >= 17) || (h<6))
        {
            a=2;
        }
    }
    EEPROM.get(eeAddress11, temp1);
    EEPROM.get(eeAddress12, arra1);
    EEPROM.get(eeAddress13, porkm1);
    EEPROM.get(eeAddress14, mines1);
    EEPROM.get(eeAddress15, ext1);

    EEPROM.get(eeAddress16, temp2);
    EEPROM.get(eeAddress17, arra2);
    EEPROM.get(eeAddress18, porkm2);
    EEPROM.get(eeAddress19, mines2);
    EEPROM.get(eeAddress20, ext2);
    if(a==1)
    {
        temp=temp1;
        arra=arra1;
        porkm=porkm1;
        mines=mines1;
        ext=ext1;
    }
    if(a==2)
    {
        temp=temp2;
    }
}

```

```

    arra=arra2;
    porkm=porkm2;
    mines=mines2;
    ext=ext2;
}
val1=arra+0.00;
val2=porkm+0.00;
val3=mines+0.00;
val4=ext+0.00;
total=temp+0.00;
}
void escritura()
{
    DateTime now = rtc.now();
    EEPROM.get( eeAddress30 , eventos );
    EEPROM.get( eeAddress33 , eventos1 );
    EEPROM.get( eeAddress31 , compania );
    EEPROM.get( eeAddress32 , companial );
    String aar="";
    String aar1="";
    String ev="";
    String ev1="";
    ev=String( eventos );
    ev1=String( eventos1 );
    aar=String( compania );
    aar1=String( companial );

    if( aar != aar1 || ev!=ev1)

{
    myFile = SD.open("Cambios.txt", FILE.WRITE); //abrimos el archivo
    if (myFile)
    {
        myFile.print("Responsable= ");
        myFile.print( compania );
        myFile.print(' ');
        myFile.print("Fecha= ");
        myFile.print( now.year() , DEC );
        myFile.print( '/' );
        myFile.print( now.month() , DEC );
        myFile.print( '/' );
        myFile.print( now.day() , DEC );
        myFile.print(' ');
    }
}

```



```
myFile.print(now.hour(), DEC);  
myFile.print(':');  
myFile.print(now.minute(), DEC);  
myFile.print(':');  
myFile.print(now.second(), DEC);  
myFile.println();  
myFile.close();  
EEPROM.put(eeAddress32, compania);  
EEPROM.put(eeAddress33, eventos);  
}  
}  
}
```

---

## .2. Diseño PCB

