



**UNIVERSIDAD NACIONAL DE CHIMBORAZO**

**FACULTAD DE INGENIERÍA**

**ESCUELA DE INGENIERÍA ELECTRÓNICA Y  
TELECOMUNICACIONES**

**"Trabajo de grado previo a la obtención del Título de Ingeniero en  
Electrónica y Telecomunicaciones"**

**TRABAJO DE GRADUACIÓN**

**Título del proyecto:**

**“DISEÑO E IMPLEMENTACIÓN DE UN ENTRENADOR DE  
MICROCONTROLADORES PARA COMUNICACIÓN VÍA RED  
CELULAR”**

**AUTORES:**

MILTON MARCELO TINGO TIERRA

FRANKLIN GEOVANNY QUISI GUALLI

**Director:**

ING. Fabián Gunsha

**Riobamba – Ecuador**

**2015**

Los miembros del Tribunal de Graduación del proyecto de investigación de título:

**“DISEÑO E IMPLEMENTACIÓN DE UN ENTRENADOR DE MICROCONTROLADORES PARA COMUNICACIÓN VÍA RED CELULAR”**

Presentado por:

Milton Marcelo Tingo Tierra

Franklin Geovanny Quisi Gualli

Y dirigida por:


Ing. Fabián Celso Gunsha Maji

Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación con fines de graduación escrito en la cual se ha constatado el cumplimiento de las observaciones realizadas, remite la presente para uso y custodia en la biblioteca de la Facultad de Ingeniería de la UNACH.

Para constancia de lo expuesto firman:

Ing. Carlos Peñafiel

**Presidente del Tribunal**



Firma

Ing. Fabián Gunsha Maji

**Director del proyecto**



Firma

Ing. Deysi Inca

**Miembro del Tribunal**



Firma

## AUTORÍA DE LA INVESTIGACIÓN

“La responsabilidad del contenido del tema de investigación de Graduación, nos corresponde exclusivamente a: Milton Marcelo Tingo Tierra, Franklin Geovanny Quisi Gaulli y del Director de Proyecto: Ing. Fabián Gunsha Maji; y el patrimonio intelectual de la misma universidad Nacional de Chimborazo”.



---

Autor: Milton Tingo  
060405825-5



---

Autor: Franklin Quisi  
060417259-3

## **AGRADECIMIENTO I**

A Dios por la fortaleza e inteligencia para poder culminar mis estudios.

A la Universidad Nacional de Chimborazo y todos los docentes de la Escuela de Ingeniería en Electrónica y Telecomunicaciones por impartirnos sus conocimientos.

A nuestro Director de Tesis Ingeniero Fabián Gunsha por su ayuda para la culminación de este proyecto.

*Milton Tingo*

## **AGRADECIMIENTO II**

Mi agradecimiento muy especial a Dios por conceder inteligencia, sabiduría por guiarme y cuidarme cada día de mi vida, también a toda mi familia quienes siempre estaban a mi lado haciendo todo lo posible para alcanzar este sueño.

Un agradecimiento a la Universidad Nacional de Chimborazo a sus autoridades y todos sus docentes quienes nos compartieron sus valiosos conocimientos de la mejor manera.

También agradecer al Director de Tesis Ing. Fabián Gunsha, por ayudarme en la realización de este proyecto, a mis amigos y compañeros quienes me estaban apoyando en los momentos buenos y malos.

*Franklin Quisi*

## DEDICATORIA I

A Dios y la Virgen Santísima por no haberme dejado caer en los momentos más difíciles de mi vida.

A mi madre Angelita, ya que de ella aprendí que lo importante en la vida es la humildad, voluntad y la fuerza para poder seguir siempre adelante sin importar las caídas en el camino de la vida.

A mi padre Marco, a mi hermano Stiven y toda mi familia, ya que ellos han sido siempre el apoyo incondicional durante toda mi vida.

A ti Mónica, la mujer que sin darme cuenta se ha convertido en lo más bonito de mi vida, que ha logrado sacar lo mejor que hay en mí.

A todos mis amigos y compañeros que estuvieron conmigo durante todos los años en la universidad.

*Milton Tingo*

## DEDICATORIA II

A Dios, por ser mi padre Celestial, por cuidarme y guiarme cada instante de mi vida.

A mis padres, quienes siempre me brindaron su apoyo incondicional, que día a día lucharon por sacarme adelante, por confiar, darme lo necesario para poder culminar con esta etapa de mi vida.

A mi esposa: por tu amor, paciencia, ternura, comprensión, bondad y sacrificio.

A mis hermanos y hermana quienes siempre estaban a mi lado ayudándome también porque nunca se apartaron de mi lado.

A mis tíos, y familiares quienes confiaron en mis capacidades, esfuerzo y sacrificio constante, para la culminación de esta meta.

También a todos quienes conforman la Iglesia Centro Evangélico “La Grey de Dios” quienes siempre confiando ayudándome todos los días de mi vida.

*Franklin Quisi*

## ÍNDICE GENERAL

PORTADA.....	I
FIRMAS DEL TRIBUNAL.....	II
AUTORÍA DE LA INVESTIGACIÓN.....	III
AGRADECIMIENTO I.....	IV
AGRADECIMIENTO II.....	V
DEDICATORIA I.....	VI
DEDICATORIA II.....	VII
ÍNDICE GENERAL.....	VIII
ÍNDICE DE FIGURAS.....	XII
ÍNDICE DE TABLAS.....	XIV
RESUMEN.....	XV
SUMARY.....	XVI
INTRODUCCIÓN.....	XVII
CAPÍTULO I.....	1
1. FUNDAMENTACIÓN TEÓRICA.....	1
1.1. ENTRENADOR DE MICROCONTROLADORES.....	1
1.1.1. MÓDULOS ENTRENADORES COMERCIALES.....	1
1.1.1.1. ENTRENADOR PIC TRAINER PARA MICROS DE 18 PINES.....	1
1.1.1.2. KIT DE SISTEMA EB88.....	2
1.1.1.3. TARJETA DE DESARROLLO ENTRENADORA PIC CON PROGRAMADOR.....	3
1.1.1.4. ATMEGA TRAINING & DEVELOPMENT BOARD.....	4
1.1.1.5. ENTRENADOR THE EASYPICV7 CONNECTIVITY.....	4
1.2. TELEFONÍA CELULAR.....	5
1.2.1. GSM (Sistema Global para las comunicaciones Móviles ).....	6
1.2.1.1. ARQUITECTURA DE RED DEL SISTEMA GSM.....	6
1.2.1.2. MENSAJES DE TEXTO.....	7
1.2.2. GPRS (Sistema General de Paquetes vía Radio).....	8
1.2.2.1. ARQUITECTURA DE RED DEL SISTEMA GPRS.....	8
1.3. EL INTERNET DE LAS COSAS (IoT).....	9
1.3.1. UBIDOTS.....	10
1.4. APP INVENTOR.....	11
1.5. SHIELDS GPRS/GSM.....	12



1.5.1.	MÓDULO GPRS QUADDBAND.....	12
1.5.2.	MÓDULO GPRS+GPS QUADDBAND (SIM908) .....	12
1.5.3.	MÓDULO 3G/GPRS+GPS PARA ARDUINO/RASPBERRY Y PI... 13	
1.5.4.	MÓDULO GPRS/ GSM SIMCOM SIM900, SHIELD V1.0 .....	14
1.5.5.	MÓDULO GSM/GPRS TINYSINE .....	14
1.6.	MÓDULO GPRS/GSM SIM900 DE SIMCOM.....	15
1.6.1.	CARACTERÍSTICAS GENERALES DE SIM900.....	17
1.7.	COMANDOS AT .....	19
1.7.1.	SINTAXIS DE LOS COMANDOS AT.....	19
1.7.1.1.	<i>COMANDOS AT GENERALES</i> .....	20
1.7.1.2.	<i>COMANDOS PARA INFORMACIÓN DEL EQUIPO</i> .....	20
1.7.1.3.	<i>COMANDOS DEL SERVICIO DE RED</i> .....	20
1.7.1.4.	<i>COMANDOS DE SEGURIDAD</i> .....	20
1.7.1.5.	<i>COMANDOS PARA REALIZAR LLAMADAS</i> .....	21
1.7.1.6.	<i>COMANDOS PARA SMS</i> .....	21
1.7.1.7.	<i>COMANDOS PARA ENVIÓ DE DATOS:</i> .....	21
1.8.	SENSORES.....	22
1.8.1.	SENSOR DE TEMPERATURA LM35.....	22
1.8.2.	SENSOR LDR.....	23
1.8.3.	SENSOR DE GAS MQ-6.....	25
1.8.4.	SENSOR DE HUMEDAD Y TEMPERATURA DHT11 .....	27
1.9.	MICROCONTROLADOR .....	28
1.9.1.	PIC 16F877A.....	28
1.9.2.	PIC 18F2550.....	30
1.9.3.	ATmega 328P.....	32
	CAPÍTULO II .....	34
2.	METODOLOGÍA .....	34
2.1.	TIPO DE ESTUDIO .....	34
2.2.	POBLACIÓN Y MUESTRA.....	34
2.2.1.	POBLACIÓN .....	34
2.2.2.	MUESTRA.....	35
2.2.3.	HIPÓTESIS .....	36
2.4.	PROCEDIMIENTOS.....	37
2.4.1.	DISEÑO DE LAS PLACAS ELECTRÓNICAS PARA EL ENTRENADOR.....	37
2.4.1.1.	<i>DISEÑO DE PLACA: MÓDULO SIM900</i> .....	38

2.4.1.1.1.	Circuito electrónico para módulo SIM900.....	38
2.4.1.1.2.	Placa electrónico para módulo SIM900 .....	39
2.4.1.2.	<i>DISEÑO DE PLACA: T-SV651, MAX232 Y MAX485</i> .....	40
2.4.1.2.1.	Circuito electrónico: Transeiver SV651, MAX232 y MAX485 .....	40
2.4.1.2.2.	Placa electrónica: Transeiver SV651, MAX232 y MAX485.....	41
2.4.1.3.	<i>DISEÑO PLACA: TECLADO, PULSADORES Y DIPSWITCHS</i> .....	41
2.4.1.3.1.	Circuito electrónico: Teclado, pulsadores y dipswitchs.....	41
2.4.1.3.2.	Placa electrónica: Teclado, pulsadores y dipswitchs .....	43
2.4.1.4.	<i>ACOPLAMIENTO DE ELEMENTOS ELECTRÓNICOS SOBRE LAS PLACAS DEL ENTRENADOR</i> .....	44
2.4.1.5.	<i>MONTAJE DE LOS DISTINTOS MÓDULOS SOBRE EL ENTRENADOR</i> .....	44
2.4.2.	<i>DISEÑO DE UN SISTEMA DE CONTROL DE TEMPERATURA DE UNA AVÍCOLA VÍA RED CELULAR</i> .....	45
2.4.2.1.	<i>CIRCUITO DE MONITOREO - SENSORES</i> .....	45
2.4.2.2.	<i>CIRCUITO DE CONMUTACIÓN - ACTUADORES</i> .....	46
2.4.2.3.	<i>COMUNICACIÓN CON EL MÓDULO GSM/GPRS</i> .....	47
2.4.2.4.	<i>CIRCUITO DEL MICROCONTROLADOR</i> .....	48
2.4.2.5.	<i>PROGRAMACIÓN DEL MICROCONTROLADOR</i> .....	48
2.4.2.5.1.	Programación de mensajes de alerta enviados por los sensores .....	48
2.4.2.5.2.	Envío de mensajes de textos .....	51
2.4.2.5.3.	Interrupción PIR1 para lectura de mensajes de texto.....	51
2.4.2.5.4.	Lectura de mensaje para activación de actuadores y consulta de magnitudes de los sensores. ....	53
2.4.2.6.	<i>DISEÑO DE LA APLICACIÓN EN ANDROID</i> .....	55
2.4.2.6.1.	Configuración del entorno de programación Android .....	55
2.4.2.6.2.	Programación de la aplicación en Android .....	55
2.5.	<i>PROCESAMIENTO Y ANÁLISIS</i> .....	58
2.5.1.	<i>ESTUDIO DEL ENTRENADOR DE MICROCONTROLADORES PARA COMUNICACIONES VÍA RED CELULAR</i> .....	58
2.5.2.	<i>ANÁLISIS DE LA COMUNICACIÓN DEL SISTEMA</i> .....	59
2.5.3.	<i>ADMINISTRACIÓN DE LA INTERFAZ DE CONTROL</i> .....	59
2.5.4.	<i>ANÁLISIS DEL SISTEMA DE CONTROL REMOTO DE LA TEMPERATURA DE UNA AVÍCOLA VÍA RED CELULAR</i> .....	59
2.6.	<i>COMPROBACIÓN DE LA HIPÓTESIS</i> .....	60
2.6.1.	<i>PLANTEAMIENTO DE LA HIPÓTESIS:</i> .....	60
2.6.2.	<i>ESTABLECIMIENTO DE NIVEL DE SIGNIFICANCIA</i> .....	61

2.6.3.	MUESTRA ENCONTRADA .....	61
2.6.4.	CÁLCULO DE LA MEDIA Y LA VARIANZA .....	63
2.6.5.	ANÁLISIS DE LAS REGIONES CRÍTICAS .....	66
2.6.6.	CÁLCULO DEL VALOR CRÍTICO O ZC.....	68
2.6.7.	REPRESENTACIÓN GRÁFICA DE LA REGLA DE DECISIÓN .....	70
CAPÍTULO III.....		71
3.	RESULTADOS.....	71
CAPÍTULO IV.....		75
4.	DISCUSIÓN .....	75
CAPÍTULO V .....		76
5.	CONCLUSIONES Y RECOMENDACIONES.....	76
5.1.	CONCLUSIONES .....	76
5.2.	RECOMENDACIONES .....	77
CAPÍTULO VI.....		78
6.	PROPUESTA.....	78
6.1.	TÍTULO DE LA PROPUESTA.....	78
6.2.	INTRODUCCIÓN .....	78
6.3.1.	OBJETIVO GENERAL .....	79
6.3.2.	OBJETIVO ESPECÍFICO .....	79
6.4.	DESCRIPCIÓN DEL PROBLEMA .....	79
6.5.	DISEÑO ORGANIZACIONAL .....	80
6.6.	MONITOREO Y EVALUACIÓN DE LA PROPUESTA .....	80
7.	BIBLIOGRAFÍA .....	81
CAPÍTULO VIII.....		83
8.	ANEXOS .....	83
ANEXO 1.- ESQUEMAS DE CIRCUITOS DE LAS PLACAS EN EL SIMULADOR.....		83
ANEXO 2. DISEÑO DE PLACAS ELECTRÓNICAS PARA ENTRENADOR		86
ANEXO 3.- PRÁCTICAS DE LABORATORIO .....		89
ANEXO 4.- CONDICIONES CLIMÁTICAS ADECUADAS DEL GALPÓN DE AVES .....		124
ANEXO 5.- PROGRAMA GLOBAL DEL MICROCONTROLADOR PARA EL CONTROL DE TEMPERATURA DE UNA AVÍCOLA VÍA RED CELULAR .....		126
ANEXO 6.- PRÁCTICAS PROPUESTAS PARA COMPROBAR EFICIENCIA DEL ENTRENADOR.....		137

## ÍNDICE DE FIGURAS

FIGURA N° 1.- Entrenador PIC TRAINER.....	2
FIGURA N° 2.- Tarjeta de desarrollo KIT EB88 .....	3
FIGURA N° 3.-Tarjeta de desarrollo entrenadora de PIC´s .....	3
FIGURA N° 4.-ATmega Training & Development Board.....	4
FIGURA N° 5.-The EasyPICv7 connectivity" .....	5
FIGURA N° 6.-Evolución de la comunicación celular.....	6
FIGURA N° 7.-Arquitectura de red del sistema GSM .....	7
FIGURA N° 8.-Estructura básica de la red SMS.....	8
FIGURA N° 9.-Arquitectura de red del sistema GPRS .....	9
FIGURA N° 10.-Concepto del “Internet de las cosas” .....	10
FIGURA N° 11.-Plataforma Ubidots .....	10
FIGURA N° 12.-Diagrama de funcionamiento de App Inventor .....	11
FIGURA N° 13.-Módulo GPRS Quadband para Arduino.....	12
FIGURA N° 14.-Módulo GPRS+GPS Quadband para Arduino y Raspberry PI (SIM908).....	13
FIGURA N° 15.-Módulo 3G/GPRS+GPS para Arduino/Raspberry PI y sus accesorios .....	13
FIGURA N° 16.-Módulo GPRS/GSM Quadband (SIM900) .....	14
FIGURA N° 17.-Módulo GSM/GPRS TinySine.....	15
FIGURA N° 18.-SIM900 de SIMCom .....	15
FIGURA N° 19.-Diagrama en Bloques del modulo .....	16
FIGURA N° 20.-Asignación de Pines del módulo SIM900 .....	16
FIGURA N° 21.- Sensor LM35 .....	22
FIGURA N° 22.-Sensor Fotosresistor LDR .....	23
FIGURA N° 23.- Conexión de LDR.....	24
FIGURA N° 24.-Sensor de gas MQ-6 .....	25
FIGURA N° 25.-Circuito para MQ-6 .....	25
FIGURA N° 26.-Características de sensibilidad del MQ-6.....	27
FIGURA N° 27.- Sensor DHT11y su conexión.....	27
FIGURA N° 28.- Comunicación del DHT11 y el microcontrolador .....	28
FIGURA N° 29.- Distribución de pines del 16F877A.....	29
FIGURA N° 30.- Distribución de pines del 18F2550.....	31
FIGURA N° 31.-Distribución de pines del ATmega 328P.....	33
FIGURA N° 32.-Interfaz de ISIS – Proteus.....	38
FIGURA N° 33.-Interfaz de ARES – Proteus.....	38
FIGURA N° 34.- Circuito de reguladores 7805 y 7833 .....	39
FIGURA N° 35.-Placa para módulo SIM900 en 3D .....	39
FIGURA N° 36.- Circuito para Transeiver SV651 .....	40
FIGURA N° 37.-Circuito MAX232.....	40
FIGURA N° 38.- Circuito MAX485.....	41
FIGURA N° 39.-Placa para Transeiver SV651, MAX232 y MAX485 en 3D .....	41
FIGURA N° 40.-Circuito de teclado.....	42
FIGURA N° 41.-Circuitos Dipswitchs .....	42
FIGURA N° 42.-Circuito de botones.....	43
FIGURA N° 43.-Placa para Teclado, pulsadores y dipswitcs en 3D.....	43

FIGURA N° 44.-Elaboración de la placa.....	44
FIGURA N° 45.-Diseño del entrenador.....	44
FIGURA N° 46.-Entradas y salidas del control de temperatura de una avícola...	45
FIGURA N° 47.-Circuito de Sensores .....	45
FIGURA N° 48.-Circuito de bobina de relé .....	46
FIGURA N° 49.-Circuito de potencia para conmutación de actuadores .....	47
FIGURA N° 50.-Topología de conexión de módulo SIM900 a la PC .....	48
FIGURA N° 51.-Circuito de control digital para el monitoreo de sensores y manipulación de relés.....	49
FIGURA N° 52.-Flujo grama de programación de mensajes de alerta enviados por sensores .....	49
FIGURA N° 53.-Programación de mensajes de alerta enviada por un sensor.....	50
FIGURA N° 54.-Programa de envío de mensajes de texto.....	51
FIGURA N° 55.-Programa de interrupción PIR1 para lectura de datos .....	52
FIGURA N° 56.-Programa de búsqueda de carácter de inicio de mensaje .....	52
FIGURA N° 57.-Flujograma de programación del microcontrolador para activación de actuadores .....	53
FIGURA N° 58.-Programa para el control de actuador 1 .....	54
FIGURA N° 59.-Programa para consulta de magnitudes de sensores.....	54
FIGURA N° 60.-Conexión de PC - teléfono con sistema Android .....	55
FIGURA N° 61.-Lógica de programación para la aplicación Android .....	56
FIGURA N° 62.-Pantalla de inicio de la aplicación en teléfono .....	57
FIGURA N° 63.-Arranque de la aplicación .....	57
FIGURA N° 64.-Pantalla principal de la aplicación.....	57
FIGURA N° 65.-Diagrama de procedimientos.....	58
FIGURA N° 66.-Etapas para el control de actuadores .....	58
FIGURA N° 67.-Etapas para la información desde el entrenador .....	58
FIGURA N° 68.-Control por mensajes de texto .....	59
FIGURA N° 69.-Control de la temperatura de una avícola vía red celular .....	60
FIGURA N° 70.-Valor crítico lateral derecho .....	67
FIGURA N° 71.-Valores críticos bilaterales .....	68
FIGURA N° 72.-Representación de límites de valores críticos.....	68
FIGURA N° 73.-Representación gráfica de decisión por armado de práctica ....	70
FIGURA N° 74.-Representación gráfica de decisión por latencia .....	70
FIGURA N° 75.-Práctica propuesta sin entrenador .....	71
FIGURA N° 76.-Práctica propuesta con el entrenador.....	72
FIGURA N° 77.-Diagrama del sistema del proyecto .....	80
FIGURA N° 78.- Circuito Electrónico para placa SIM900 .....	83
FIGURA N° 79.- Circuito para placa Transeiver SV651, MAX232 y MAX485.	84
FIGURA N° 80.-Circuito para placa: teclado, pulsadores y dipswitchs.....	85
FIGURA N° 81.-Diseño de placa para SIM900 .....	86
FIGURA N° 82.- Placa para SIM900 .....	86
FIGURA N° 83.-Diseño de placa para Transeiver SV651, MAX232 y MAX485	87
FIGURA N° 84.-Placa para Transeiver SV651, MAX232 y MAX485.....	87
FIGURA N° 85.-Diseño de placa para Teclado, pulsadores y dipswitchs .....	88
FIGURA N° 86.-Placa para Teclado, pulsadores y dipswitchs .....	88

## ÍNDICE DE TABLAS

Tabla N° 1.-Características de SIM900 .....	17
Tabla N° 2.-Características del PIC 16F877A .....	29
Tabla N° 3.-Características del 18F2550 .....	30
Tabla N° 4.-Características del ATmega 328P .....	32
Tabla N° 5.-Operacionalización de variables .....	37
Tabla N° 6.-Datos de tiempo de armado de práctica .....	61
Tabla N° 7.-Datos medidos en tiempo de Latencia .....	62
Tabla N° 8.-Cálculo de la media y desviación para los datos de armado de la práctica .....	63
Tabla N° 9.-Valores de varianza y desviación estándar del tiempo de armado de práctica .....	64
Tabla N° 10.-Calculo de la media y varianza de latencias.....	65
Tabla N° 11.-Regla de decisión para entrenador .....	66
Tabla N° 12.-Datos que determinan los valores críticos.....	67
Tabla N° 13.-Pruebas de comunicación .....	72
Tabla N° 14.-Materiales utilizados en el proyecto.....	73

## **RESUMEN**

El propósito de este trabajo de investigación es el diseño e implementación de un entrenador de microcontroladores para comunicaciones, que permitan el desarrollo de nuevas y varias aplicaciones con microcontroladores en el área de las telecomunicaciones, principalmente en las comunicaciones vía red celular.

Tras ensamblar el entrenador, se pudo realizar distintas prácticas con los microcontroladores y la red de telefonía móvil, que permitan desarrollar un sistema de control de temperatura a través de un teléfono celular para una avícola, con sensores y dispositivos electrónicos que contiene el entrenador.

El trabajo culmina con las pruebas de funcionamiento de los dispositivos utilizados del entrenador y si se mejoró la rapidez del armado de las practicas, demostrando que esta herramienta es eficiente.



UNIVERSIDAD NACIONAL DE CHIMBORAZO  
FACULTAD DE INGENIERÍA  
CENTRO DE IDIOMAS



Lic. Maritza Larrea

17 de Diciembre del 2015

### SUMMARY

Design and implement communications coach microcontrollers is the main purpose of this research to let development of new and various microcontroller applications in the telecommunication areas, mainly in communications cellular network.

After assembling the trainer, it could carry out various practices with microcontrollers and mobile network, in order to develop a system of temperature control through a cell phone to poultry, with sensors and electronic devices containing the coach.

The work ends with performance testing of devices used the coach and if the speed of the assembly of improved practices, showing that this tool is efficient.





## INTRODUCCIÓN

Con el pasar de los últimos años se ha notado considerablemente la evolución de las telecomunicaciones, entre todas, la más destacada la telefonía móvil que día a día provee nuevos servicios a los usuarios.

Es necesario mantenerse a la par con la tecnología, por esta razón se ha visto la necesidad de desarrollar una herramienta que cuente con dispositivos electrónicos, para construir sistemas embebidos con circuitos integrados programables.

El diseño de un sistema de control de temperatura de una avícola vía red celular, demuestra que los dispositivos del entrenador comandados por un microcontrolador, permiten realizar sistemas útiles para la actualidad.

Detalla los circuitos de monitoreo que están encargados de las mediciones de la temperatura de una avícola, procesadas por el microcontrolador que envía mensajes de texto a un teléfono móvil, indicando si alguna magnitud se encuentra fuera de un rango previamente establecido. Los circuitos de control manejan la activación o desactivación de algún actuador en el galpón, cada vez que el usuario envía un mensaje desde su teléfono móvil, indicando la acción que requiere para tener el control del sistema.

# CAPÍTULO I

## 1. FUNDAMENTACIÓN TEÓRICA

### 1.1. ENTRENADOR DE MICROCONTROLADORES

Herramienta didáctica compuesta por un conjunto de circuitos electrónicos pre-ensamblados que poseen la capacidad de conectarse a un microcontrolador para que éste pueda controlarlos. Para que un entrenador sea robusto y eficiente debe poseer una serie de características como:

- **Modular.** Los distintos módulos deben ser compatibles entre sí y el microcontrolador.
- **Contar con fuente de alimentación.** Evita así la dependencia de otras herramientas.
- **Soportar una gama de dispositivos programables.** Debe romper los límites de trabajar con un solo integrado o un solo fabricante.

#### 1.1.1. MÓDULOS ENTRENADORES COMERCIALES

En el mercado e internet se puede encontrar múltiples entrenadores de diferentes fabricantes. A continuación se presenta a los más destacados.

##### *1.1.1.1. ENTRENADOR PIC TRAINER PARA MICROS DE 18 PINES*

Se observa en la **figura N°1**, es muy básico porque implementa el módulo de entradas y salidas del PIC<sup>1</sup> 16F628A de Microchip. Este entrenador posee ICSP<sup>2</sup> para la programación del dispositivo. La placa mide 9.5 x 6.7 centímetro y se comercializa listo para funcionar a un costo de 30 dólares. (Gonzalez Rojas & Mendez Caviativa, 2011)

---

<sup>1</sup> Controlador de Interfaz Periférico

<sup>2</sup> In-Circuit Serial Programming



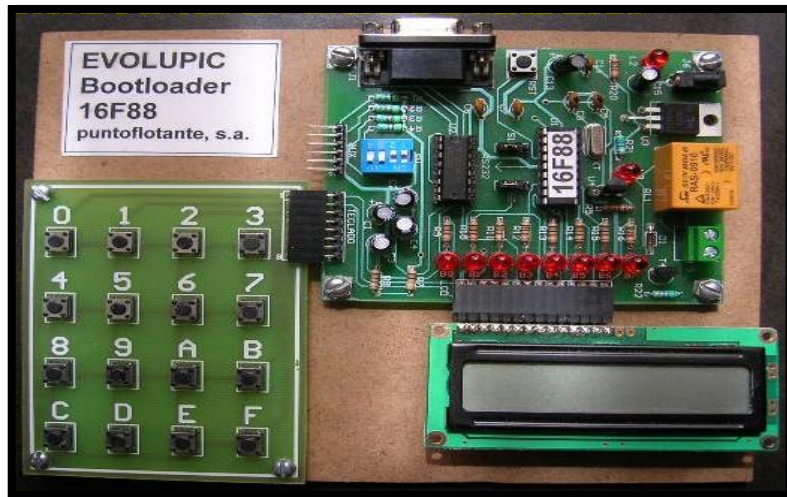


FIGURA N° 2.- Tarjeta de desarrollo KIT EB88  
 FUENTE:[http://articulo.mercadolibre.com.mx/MLM-492395691-kit-microcontrolador-pic-16f88-tarjeta-de-desarrollo-\\_JM](http://articulo.mercadolibre.com.mx/MLM-492395691-kit-microcontrolador-pic-16f88-tarjeta-de-desarrollo-_JM)

### ***1.1.1.3. TARJETA DE DESARROLLO ENTRENADORA PIC CON PROGRAMADOR***

Se observa en la **figura N°3**; ideal para usar con cualquier microcontrolador de 40 pines. Cuenta con: sensor de temperatura, teclado 4x4, juego de LED's, entradas analógicas de 7 segmentos, puerto serial, Buzzer, conectores para LCD 2x16, LCD grafica 128x64. (ElectrotekMEGA, 2014)



FIGURA N° 3.-Tarjeta de desarrollo entrenadora de PIC's  
 FUENTE: [http://electrotekmega.com/producto/tarjeta-desarrollo-entrenadora-microcontroladores-pic/#lightbox\[gal\]/0/](http://electrotekmega.com/producto/tarjeta-desarrollo-entrenadora-microcontroladores-pic/#lightbox[gal]/0/)

#### ***1.1.1.4. ATMEGA TRAINING & DEVELOPMENT BOARD***

Desarrollado para microcontroladores ATmel AVR. La placa tiene dos ISP programador y JTAG<sup>9</sup>, conectados al puerto serie de una PC para depurar y cargar el código con la ayuda del software AVR Studio IDE.

Esta tarjeta funciona con el microcontrolador ATmega128 y se encuentra en una tarjeta secundaria extraíble. Dentro de los componentes se encuentran: 2 puertos RS-232, 1 puerto RS-485, LCD, fuente de 9VDC, oscilador, 4 display de siete segmentos, 4 pulsadores, 8 LEDs, buzzer, chip de tiempo real PCF8563, IIC E2PROM<sup>10</sup> AT24C01, amplificador dual LM358 para A/D y un potenciómetro para ajustar la señal A/D (MCPros, 2009). Se observa en la **figura N°4**.

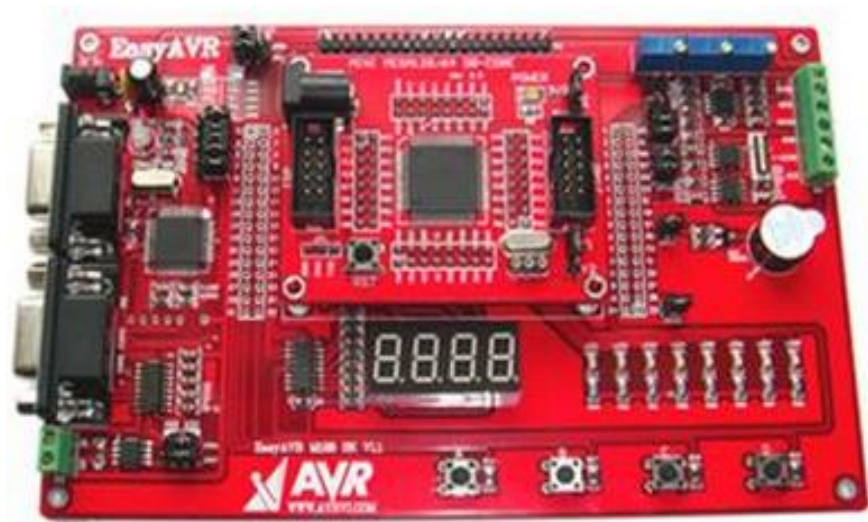


FIGURA N° 4.-ATmega Training & Development Board  
FUENTE: [http://microcontrollershop.com/product\\_info.php?products\\_id=3404](http://microcontrollershop.com/product_info.php?products_id=3404)

#### ***1.1.1.5. ENTRENADOR THE EASYPICV7 CONNECTIVITY***

Comercializado por Mikroelektronika, es la séptima generación de esta placa de desarrollo PIC. Contiene un microcontrolador PIC 18F45K22 que cuenta con 16 MIPS<sup>11</sup> de operación, 32 Kbytes de memoria de programa lineal, además tiene 36 E/S de uso general, 30 pines de A/D, 3 contadores de tiempo de 8 bits,

---

<sup>9</sup> Joint Test Action Group

<sup>10</sup> Electrically Erasable Programmable Read-Only Memory

<sup>11</sup> Microprocessor without Interlocked Pipeline Stages



temporizadores de 16 bits. También cuenta con par de CCP<sup>12</sup>, comparadores y módulos MSSP<sup>13</sup> que pueden ser SPI o I2C<sup>14</sup>.

Está compuesta por : “UART vía RS232, sensor de temperatura, 36 LEDs, 36 pulsadores, LCD, 4 display de siete segmentos, pantalla touch, conector USB<sup>15</sup> y PS2<sup>16</sup>, un CD con aplicaciones, drivers y ejemplos desarrollados en C. Se observa en la **figura N°5.**” (Gonzalez Rojas & Mendez Caviativa, 2011). Además cuenta con una serie de tarjetas que se pueden utilizar tales como: Wifi, GPS<sup>17</sup>, RS485, DAC<sup>18</sup>, lector de MicroSD, entre otras.



FIGURA N° 5.-The EasyPICv7 connectivity"  
FUENTE: <http://www.mikroe.com/easypic/>

## 1.2. TELEFONÍA CELULAR

Las tecnologías de comunicaciones inalámbricas y móviles han ido evolucionando con el pasar del tiempo, pero la que ha tenido mayor desarrollo en los últimos años ha sido los sistemas de comunicación celular, como muestra la **figura N°6.**

---

<sup>12</sup> Multitud de Control de Producciones

<sup>13</sup> Microsoft Smooth Streaming Protocol

<sup>14</sup> Inter-Circuitos Integrados

<sup>15</sup> Bus Serial Universal

<sup>16</sup> Sistema personal/ 2

<sup>17</sup> Sistema de Posicionamiento Global

<sup>18</sup> Convertidor Analógico Digital

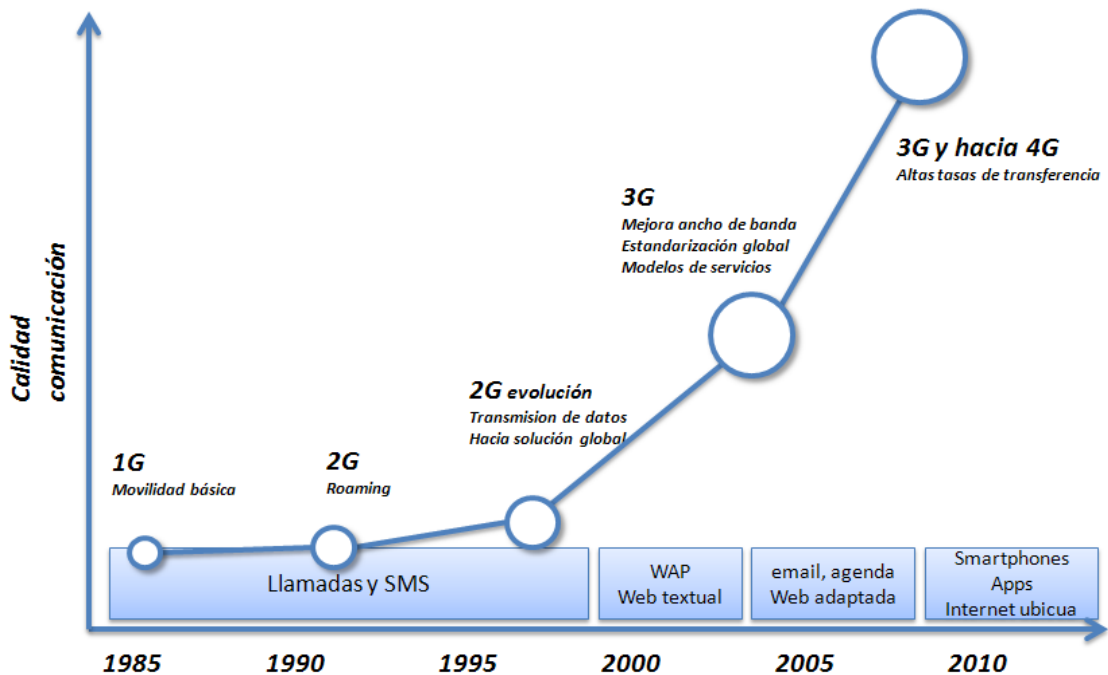


FIGURA N° 6.-Evolución de la comunicación celular

FUENTE: <https://altransmart.files.wordpress.com/2011/11/grafic-evolucionic3b3.png>

### 1.2.1. GSM (Sistema Global para las comunicaciones Móviles )

Estándar conocido como de segunda generación (2G), su principal característica es que son comunicaciones totalmente digitales.

#### 1.2.1.1. ARQUITECTURA DE RED DEL SISTEMA GSM

Una red GSM está constituida por una tarjeta SIM<sup>19</sup> en el teléfono celular, se encarga de identificar en la red celular gracias a un número de identificación denominado IMEI<sup>20</sup>, compuesto por 15 dígitos. Cada tarjeta SIM posee un número de identificación denominado IMSI<sup>21</sup>, para que el abonado sea identificado en el sistema.

En la **figura N°7**, se visualiza la arquitectura de una red GSM compuesta por subsistemas con funciones específicas. Es el caso del BSS<sup>22</sup> que se encarga de las conexiones de la Estaciones Base (BTS) a un centro de control (BSC<sup>23</sup>). También

<sup>19</sup> Módulo de identificación de abonado

<sup>20</sup> Identificador internacional de equipos móviles

<sup>21</sup> Identificador internacional de abonados móviles

<sup>22</sup> Subsistema de estación base

<sup>23</sup> Controlador de Estación Base

el caso del NSS<sup>24</sup>, donde se conectan físicamente los BSC y también se añade Internet a la telefonía pública, a este punto se lo conoce como MSC<sup>25</sup>, y se vincula una serie de base de datos para proveer funciones adicionales como:

- “HLR<sup>26</sup>: almacena la información de abonados.
- VLR<sup>27</sup> contiene información de usuario que no son abonados locales.
- EIR<sup>28</sup>: contiene la lista con los dispositivos móviles.
- AuC<sup>29</sup>: su misión es verificar las identidades de los usuarios.” (Castro Dominguez, s.f)

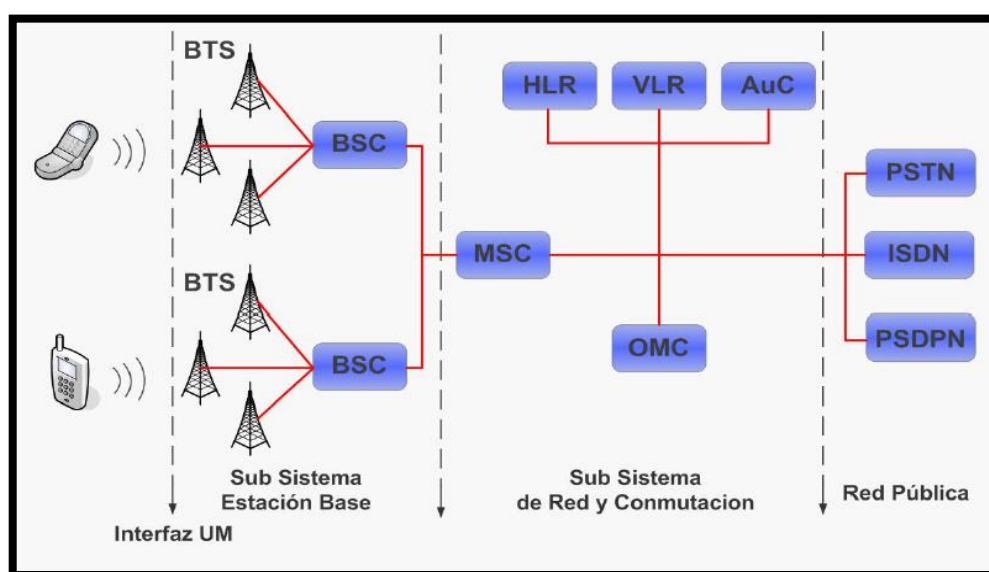


FIGURA N° 7.-Arquitectura de red del sistema GSM  
 FUENTE: [https://www.academia.edu/9921597/PROYECTO\\_FIN\\_DE\\_CARRERA](https://www.academia.edu/9921597/PROYECTO_FIN_DE_CARRERA)

### 1.2.1.2. MENSAJES DE TEXTO

El servicio de mensajes cortos o SMS<sup>30</sup>, servicio que brinda un sistema de red GSM, y permite al usuario el envío y recepción de mensajes de texto que tenga una longitud de 160 caracteres, los que pueden ser palabras, números, o combinados y no contiene imágenes.

<sup>24</sup> Subsistema de red y conmutación  
<sup>25</sup> Centros de conmutación móvil  
<sup>26</sup> Registro de Posición Base  
<sup>27</sup> Registro de Posición Visitante  
<sup>28</sup> Registro de identificación del equipo  
<sup>29</sup> Centro de Autenticación  
<sup>30</sup> Short Message Service



Su arquitectura se basa en los siguientes elementos, ver **figura N°8**:

- “SMC o SME<sup>31</sup>: permite enviar o recibir mensajes cortos.
- SMSC<sup>32</sup>: almacena y envía los mensajes cortos entre los distintos SMC.
- SMS-Gateway: es un MSC capaz de recibir un mensaje de un SMSC, interrogando al HLR sobre la información de encaminamiento y enviando el SMS al MSC.” (Cuzco Carrión & Layana Vallejo, 2012)

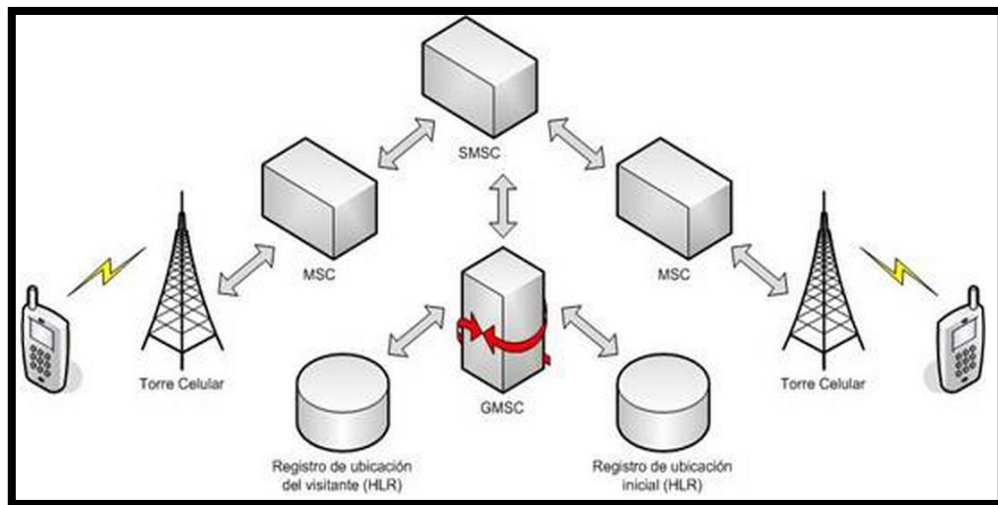


FIGURA N° 8.-Estructura básica de la red SMS

FUENTE: <http://dspace.ups.edu.ec/bitstream/123456789/4120/1/UPS-GT000375.pdf>

### 1.2.2. GPRS (Sistema General de Paquetes vía Radio)

GPRS es un estándar de telefonía de segunda generación, porque es intercalable con la estructura del sistema GSM, que introduce un sistema de conmutación de paquetes.

#### 1.2.2.1. ARQUITECTURA DE RED DEL SISTEMA GPRS

En la **figura N°9**, se muestra la estructura funcional del sistema GPRS, basado en la adición de nuevos nodos sobre la infraestructura correspondiente a GSM. A dichos nodos se les conoce como GSN<sup>33</sup>, ubicados en una red de transporte:

<sup>31</sup> Entidades de mensajería corta

<sup>32</sup> Centro de servicio de mensajería corta

<sup>33</sup> Nodo de soporte GPRS

- El router SGSN<sup>34</sup>: gestiona las direcciones de las terminales de celdas y proporciona la transferencia de la interfaz de paquetes con la pasarela GGSN.
- La pasarela GGSN<sup>35</sup>: se conecta con otras redes de datos (*Internet*), proporcionar una dirección IP a las terminales móviles durante toda la conexión. (CCM.net, 2015)

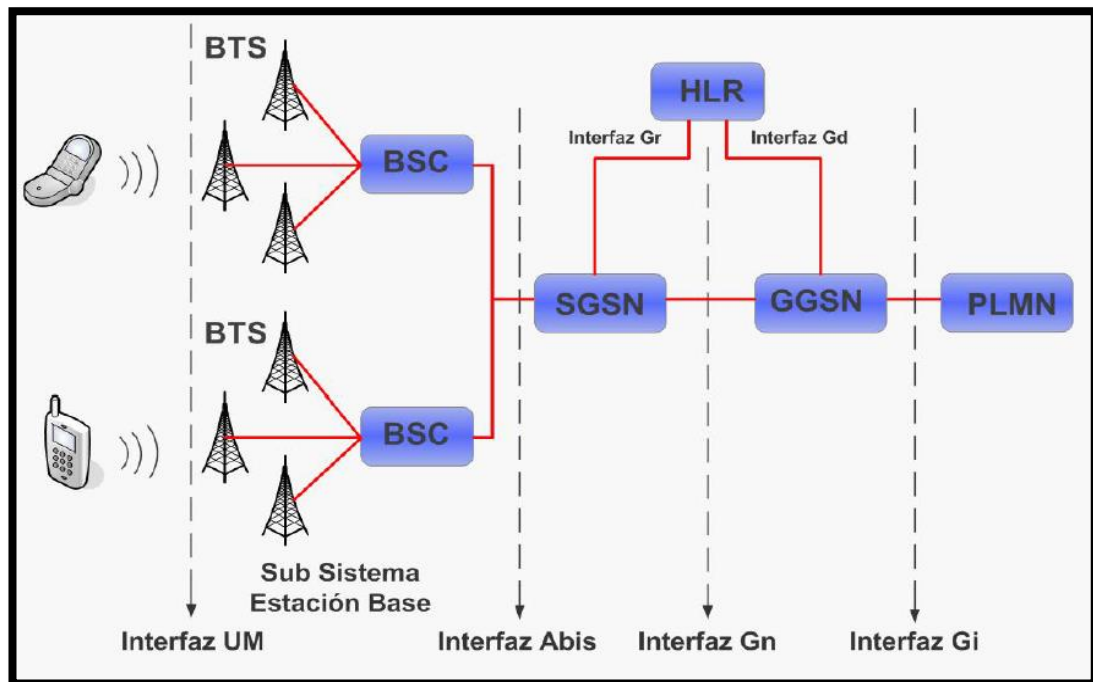


FIGURA N° 9.-Arquitectura de red del sistema GPRS  
 FUENTE: [https://www.academia.edu/9921597/PROYECTO\\_FIN\\_DE\\_CARRERA](https://www.academia.edu/9921597/PROYECTO_FIN_DE_CARRERA)

### 1.3.EL INTERNET DE LAS COSAS (IoT)

La expresión Internet de las cosas, hace referencia a la interconexión entre objetos de consumo o de uso cotidiano (electrodomésticos, ropa, libros, productos alimenticios, etc.) que a través de ciertos dispositivos son capaces de conectarlos a la red. La **figura N°10** se resume el concepto de IoT. (Castro Dominguez, s.f)

<sup>34</sup> Nodo de soporte de servicios GPRS

<sup>35</sup> Nodo de soporte de pasarela GPRS

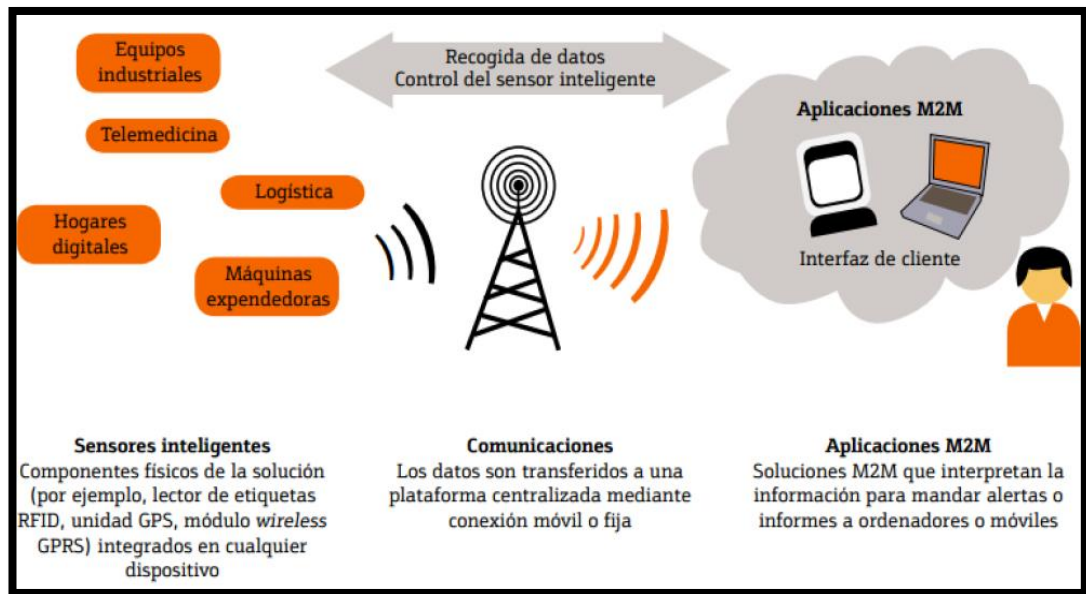


FIGURA N° 10.-Concepto del “Internet de las cosas”

FUENTE: [https://www.academia.edu/9921597/PROYECTO\\_FIN\\_DE\\_CARRERA](https://www.academia.edu/9921597/PROYECTO_FIN_DE_CARRERA)

### 1.3.1. UBIDOTS

Ubidots, servidor que permite almacenar datos de sensores y visualizarlos en tiempo real a través de una página web. También puede configurar alertas Email o SMS dependiendo del valor de los sensores, como ejemplo: enviar un SMS cuando un garaje esté abierto o enviar un email cada vez que haya un movimiento en un departamento (UbiMaker, 2014). A continuación en la **figura N°11** se aprecia el entorno de trabajo de Ubidots.

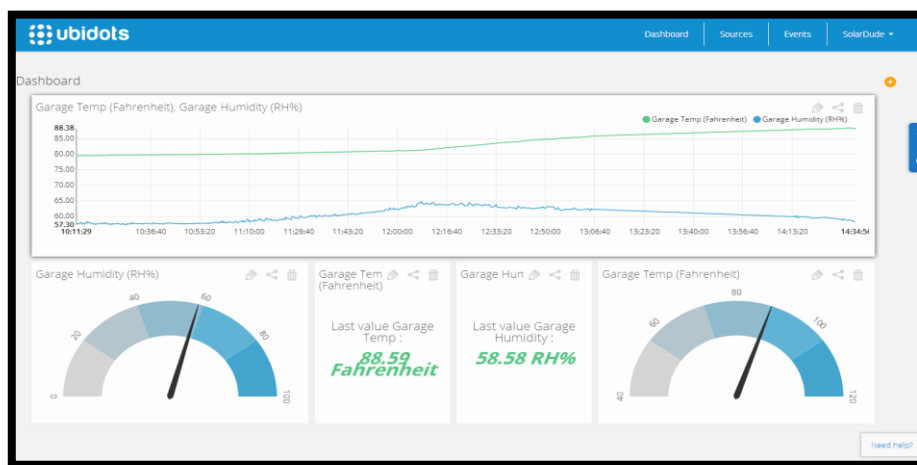


FIGURA N° 11.-Plataforma Ubidots

FUENTE: <https://community.particle.io/t/example-logging-and-graphing-data-from-your-spark-core-using-google/2929/105>

## 1.4. APP INVENTOR

“App Inventor permite desarrollar aplicaciones para teléfonos basados en sistema operativo Android mediante un navegador web” (MIT App Inventor, s.f). En la **figura N°12** se ve el diagrama de funcionalidad de App Inventor.

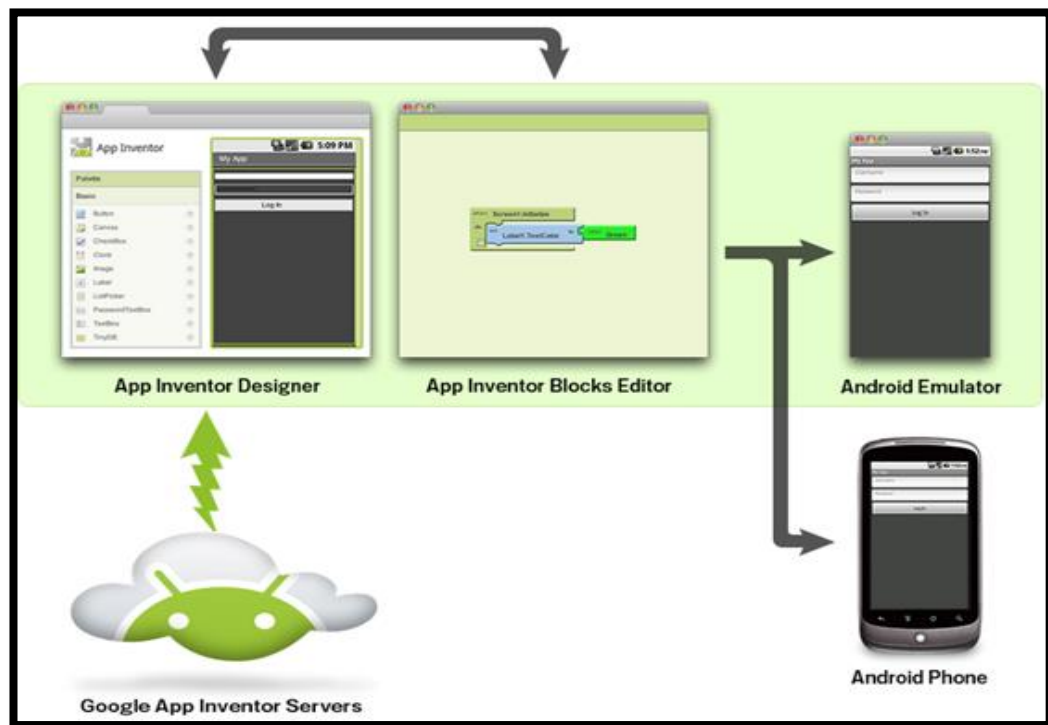


FIGURA N° 12.-Diagrama de funcionamiento de App Inventor

FUENTE:<http://recursostic.educacion.es/observatorio/web/en/software/programacion/1090-uso-de-appinventor-en-la-asignatura-de-tecnologias-de-la-comunicacion-y-la-informacion>

Para el desarrollo de aplicaciones se trabaja en:

- “El Diseñador, donde se selecciona los componentes para la aplicación.
- El Editor de Bloques, donde se ensamblan los bloques de programa para especificar cómo deben comportarse sus componentes” (MIT App Inventor, s.f). La programación se desarrolla ensamblando piezas como un rompecabezas.

Una vez terminado el trabajo se empaqueta y produce una aplicación para instalar en un celular o en un emulador de teléfono con Android.

## 1.5. SHIELDS GPRS/GSM

Un shield es un módulo en forma de placa impresa que se puede conectar desde sus puntos de conexión hacia otro circuito para ampliar sus capacidades, permitiendo que con un mismo circuito pueda generar distintas aplicaciones.

### 1.5.1. MÓDULO GPRS QUADBAND

Permite obtener los servicios de un modem GPRS gracias a que integra un módulo HILO SAGEM. En la **figura N°13** se visualiza la placa física del shield, para establecer comunicación se necesita de una antena externa.



FIGURA N° 13.-Módulo GPRS Quadband para Arduino  
FUENTE: <http://shieldlist.org/libelium/gprs-quadband>

Este módulo permite las funciones que se busca para las aplicaciones en el entrenador, pero debido a su precio y a su poca información, queda descartada.

### 1.5.2. MÓDULO GPRS+GPS QUADBAND (SIM908)

Nuevo modelo de shield GPRS para Arduino y Raspberry integra un módulo SIM908, ofrece las tecnologías GPS<sup>36</sup> y GPRS, necesita de antenas adicionales para establecer las comunicaciones. En la **figura N°14** se adjunta una imagen de dicho shield. Debido a sus costos se lo deja un lado, pero no puede quedar descartado en su totalidad puesto que si se necesita tener funciones de posicionamiento puede ser un modelo interesante.

---

<sup>36</sup> Sistema de Posicionamiento Global



FIGURA N° 14.-Módulo GPRS+GPS Quadband para Arduino y Raspberry PI (SIM908)  
FUENTE: <https://www.cooking-hacks.com/documentation/tutorials/geolocation-tracker-gprs-gps-geoposition-sim908-arduino-raspberry-pi#raspberrypi>

### 1.5.3. MÓDULO 3G/GPRS+GPS PARA ARDUINO/RASPBERRY Y PI

Este es el modelo más completo de entre todos los shield GPRS disponibles. Gracias a su módulo SIM5218, integra servicios 3G y tecnología GPS. Su precio es bastante elevado (*\$250 USD*), pero admite más funcionalidades en comparación con el resto de shields, permite la conexión de una cámara para la toma de imágenes (Castro Dominguez, s.f). En la **figura N°15** se ve el aspecto que presenta este shield. A pesar que se trata de la placa con más prestaciones, no se puede optar por ella en primer lugar por su precio, después por la tecnología con las que cuenta (*GPS* y *3G*), que quizá no se llegara hacer uso.

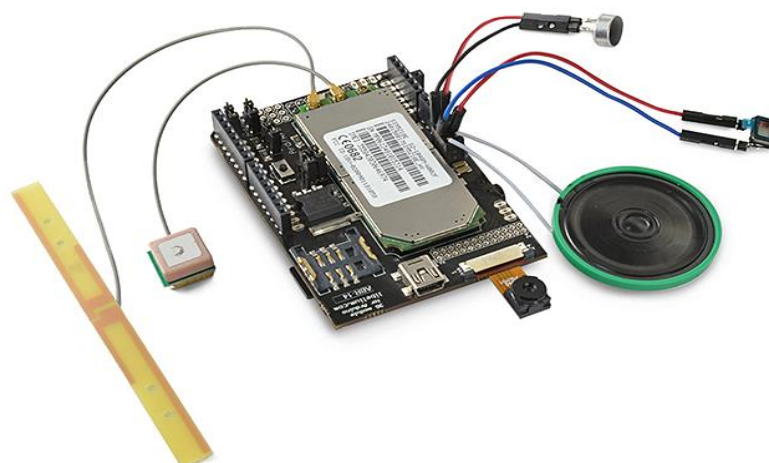


FIGURA N° 15.-Módulo 3G/GPRS+GPS para Arduino/Raspberry PI y sus accesorios  
FUENTE: <https://www.cooking-hacks.com/documentation/tutorials/3g-gps-shield-arduino-raspberry-pi-tutorial/>



#### 1.5.4. MÓDULO GPRS/ GSM SIMCOM SIM900, SHIELD V1.0

Esta placa puede convertir a un circuito electrónico en una plataforma que proporciona comunicación a través de la red telefonía celular GPRS/ GSM. Integra un módulo SIM900 de SIMCom, permite establecer llamadas, enviar SMS, MMS y audio a través de UART, incluso permite comunicación de datos a través de los protocolos TCP<sup>37</sup>, UDP<sup>38</sup>, HTTP<sup>39</sup> o FTP<sup>40</sup>, todo esto mediante comandos AT (GSM 07.07, 07.05). Cuatri banda (*Quadband*) con frecuencias de 850, 900, 1800 y 1900 MHz con el propósito que pueda funcionar en todos los países del mundo. En la **figura N°16** se adjunta una imagen de este módulo:



FIGURA N° 16.-Módulo GPRS/GSM Quadband (SIM900)  
FUENTE: <http://es.aliexpress.com/w/wholesale-arduino-gprs-shield.html>

#### 1.5.5. MÓDULO GSM/GPRS TINYSINE

El módulo de TineSine se basa en el módulo SIM900 de SIMCOM y utiliza la red telefónica celular GSM para enviar o recibir datos desde una ubicación remota, esto a través de cualquiera de los tres métodos: servicio de SMS, audio, servicio GPRS. Está configurado y controlado a través de su UART usando comandos AT. Además de las características de comunicación tiene 12 GPIO<sup>41</sup>, 2

---

<sup>37</sup> Transmission Control Protocol

<sup>38</sup> User Datagram Protocol

<sup>39</sup> Protocolo de Transferencia de Hipertexto

<sup>40</sup> Protocolo de Transferencia de Archivos

<sup>41</sup> Entrada/Salida de Propósito General

PWM<sup>42</sup> y un ADC (TinySine Electronics, 2014). Se puede observar en la **figura N°17** su estructura física.

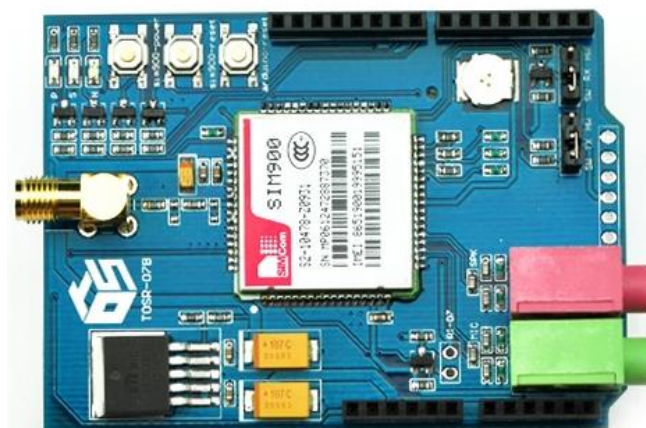


FIGURA N° 17.-Módulo GSM/GPRS TinySine

FUENTE: [http://www.tinyosshop.com/index.php?route=product/product&product\\_id=464](http://www.tinyosshop.com/index.php?route=product/product&product_id=464)

Su precio no supera los \$50 USD y es por ello que no cabe duda que es uno de los más baratos en el mercado, y ofrece todos los servicios que se necesita para poder realizar distintas aplicaciones sobre las comunicaciones móviles celulares.

## 1.6. MÓDULO GPRS/GSM SIM900 DE SIMCOM

En la **figura N°18** se muestra el aspecto físico del SIM900 y la **figura N°19** el diagrama de bloque de la funcionalidad del módulo.



FIGURA N° 18.-SIM900 de SIMCom

FUENTE: <http://www.soselectronic.com/?str=371&artnum=81028&name=simcom-sim900>

---

<sup>42</sup> Modulación por ancho de pulsos



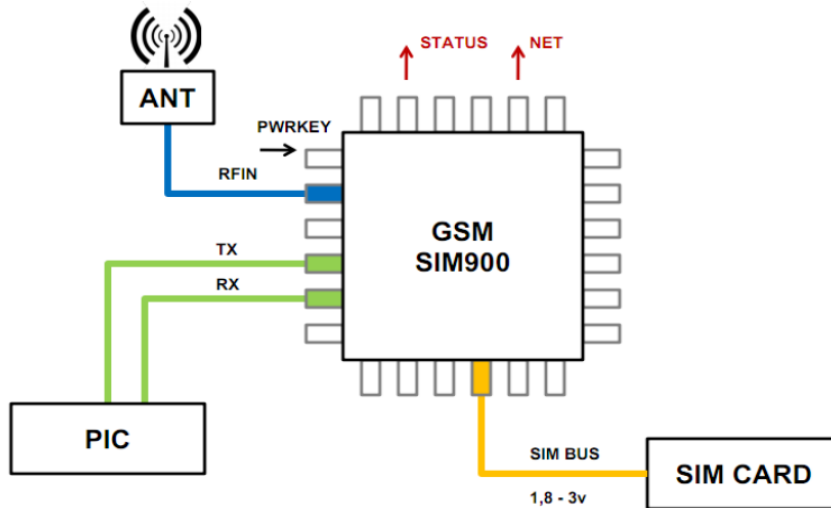


FIGURA N° 19.-Diagrama en Bloques del modulo  
 FUENTE: <http://dspace.uazuay.edu.ec/bitstream/datos/3289/1/10063.pdf>

La referencia de todos los pines del módulo SIM900, se puede observar en la figura N°20.

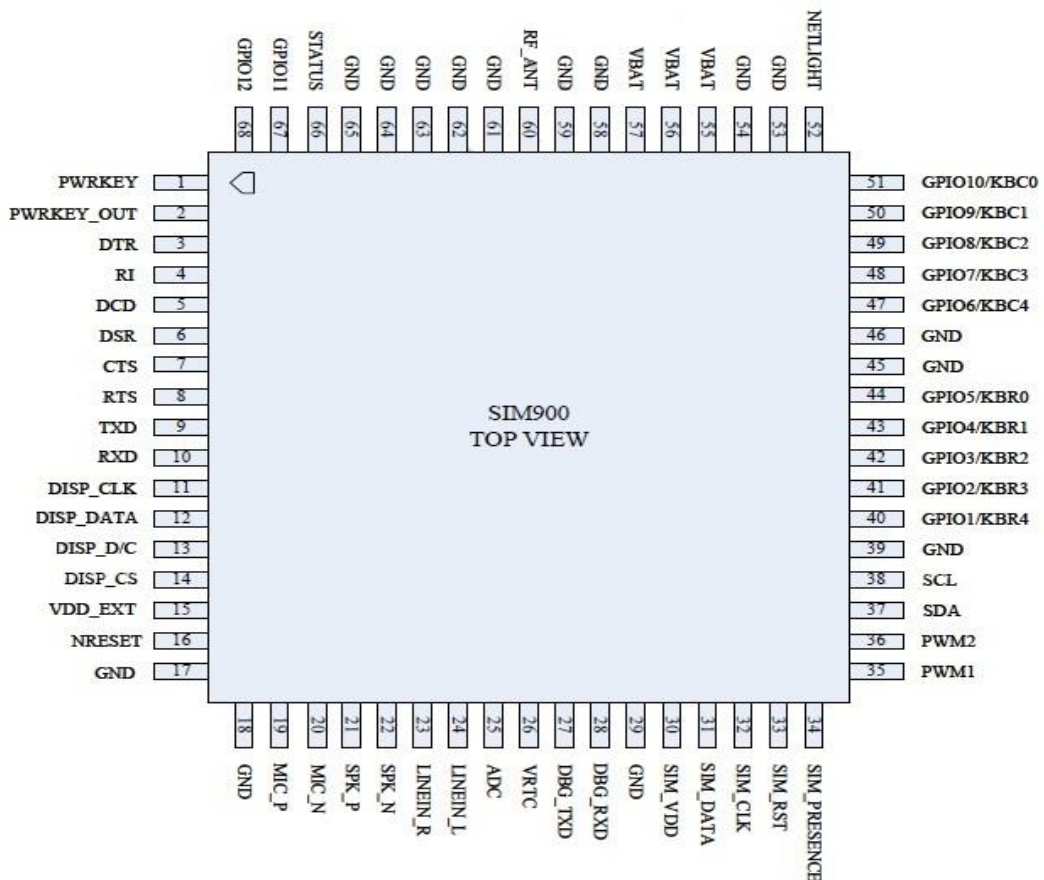


FIGURA N° 20.-Asignación de Pines del módulo SIM900  
 FUENTE: <http://www.open-electronics.org/gsm-remote-control-part-4-sim900/>

## 1.6.1. CARACTERÍSTICAS GENERALES DE SIM900

Tabla N° 1.-Características de SIM900

FUENTE: [http://mt-system.ru/sites/default/files/documents/sim908\\_hardware\\_design\\_v2.00.pdf](http://mt-system.ru/sites/default/files/documents/sim908_hardware_design_v2.00.pdf)

Característica	Implementación
Fuente de alimentación	3.2V ~4.8V
Ahorro de energía	Consumo típico en modo de reposo es 1.0mA
Bandas de frecuencia	<ul style="list-style-type: none"> <li>• SIM900 cuatri banda: GSM 850, EGSM<sup>43</sup> 900, DCS<sup>44</sup> 1800, PCS<sup>45</sup> 1900. SIM900 puede buscar las 4 bandas automáticamente.</li> <li>• Cumple con la Fase GSM 2/2+.</li> </ul>
Potencia de transmisión	<ul style="list-style-type: none"> <li>• Clase 4 (2W) en GSM 850 y EGSM 900</li> <li>• Clase 1 (1W) en DCS 1800 y PCS 1900</li> </ul>
Conectividad GPRS	<ul style="list-style-type: none"> <li>• GPRS clase 10 multi-slot (default).</li> <li>• GPRS clase 8 multi-slot (opcional).</li> </ul>
Rango de temperatura	<ul style="list-style-type: none"> <li>• Normal: -30°C ~ +80°C</li> <li>• Restringida: -40°C ~-30°C y +80°C ~+85°C</li> <li>• Temperatura de almacenamiento: -45°C ~ +90°C</li> </ul>
Datos GPRS	<ul style="list-style-type: none"> <li>• Transferencia de enlace descendente de datos: máx. 85.6 kbps.</li> <li>• Transferencia de enlace ascendente de datos: máx. 42.8 kbps.</li> <li>• Esquema de codificación: CS-1, CS-2, CS-3, CS-4</li> <li>• Integra el protocolo TCP/IP<sup>46</sup></li> <li>• Soporte de paquetes de canal de control de difusión (<i>PBCCH</i><sup>47</sup>).</li> </ul>
CSD <sup>48</sup>	Soporte de transmisión CSD

<sup>43</sup> GSM Extendida

<sup>44</sup> Digital Cellular Service

<sup>45</sup> Servicio de Comunicación Personal

<sup>46</sup> Transmission Control Protocol/Internet Protocol

<sup>47</sup> Packet Broadcast Control Channel

<sup>48</sup> Circuit Switched Data

USSD <sup>49</sup>	Servicios de datos suplementarios no estructurados.
SMS	<ul style="list-style-type: none"> <li>• MT<sup>50</sup>, MO<sup>51</sup>, CB<sup>52</sup>, texto y el modo PDU.</li> <li>• Almacenamiento de SMS: tarjeta SIM</li> </ul>
FAX	Grupo 3 clase 1
Interface SIM	Soporta tarjeta SIM: 1.8V, 3V
Antena externa	Antenna pad
Funciones de audio	<p>Modos de códec de voz:</p> <ul style="list-style-type: none"> <li>• Half Rate (ETS<sup>53</sup> 06, 20)</li> <li>• Full Rate (ETS 06, 10)</li> <li>• Enhanced Full Rate (ETS 06.50/ 06.60/ 06.80)</li> <li>• Cancelación de eco</li> <li>• Supresión de ruido</li> </ul>
Puerto serie y puerto de depuración	<p><b>Puerto Serie:</b></p> <ul style="list-style-type: none"> <li>• Interfaz de modem completa con líneas de estado y control, desequilibradas, asíncrono.</li> <li>• 1200 bps a 115200 bps.</li> <li>• Puede ser utilizado para los comandos AT o flujo de datos.</li> <li>• Capacidad multiplex de acuerdo con GSM 07.10 Protocolo multiplexor.</li> <li>• Autobauding, soporta velocidades de 1200 a 57600 bps.</li> </ul> <p><b>Puerto de depuración:</b></p> <ul style="list-style-type: none"> <li>• DBG_TXD interfaz de modem nulo y DBG_RXD.</li> <li>• Puede ser utilizad para la depuración y actualización del firmware.</li> </ul>
Características físicas	<p>Tamaño: 24*24*3 mm</p> <p>Peso: 3.4g</p>

<sup>49</sup> Unstructured Supplementary Services Data

<sup>50</sup> Terminal móvil

<sup>51</sup> Mobile Originated

<sup>52</sup> Cell Broadcast

<sup>53</sup> Telecommunications Standards

## 1.7.COMANDOS AT

De la abreviatura de Attention, lenguaje por medio de instrucciones codificadas que permite la comunicación entre un Modem y el hombre. Permitiendo realizar llamadas, enviar SMS, y muchas otras opciones de acuerdo a un juego de comandos AT específicos.

### 1.7.1. SINTAXIS DE LOS COMANDOS AT

“Los comandos AT están compuestos por cadenas de caracteres ASCII que para su ejecución se debe anteponer la palabra AT a excepción de los comandos de pause y de repetición de comando anterior.” (Alvarez Pineda, 2013)

Su estructura para su envío es la siguiente:

AT+	CMGF=1	<CR>
<i>PREFIJO</i>	<i>COMANDO</i>	<i>SUFIJO</i>

- El prefijo es la cadena de carácter AT seguido del signo +.
- El sufijo es <CR> retorno de carro, que equivale a un ENTER.
- El comando es la instrucción que se dará al modem, con el signo igual (=) junto al comando se configura algún parámetro y con el signo de interrogación (?) se pide información. Al combinar los signos igual-interrogación se obtiene el rango de opciones posibles que se puede configurar el modulo. (Alvarez Pineda, 2013)

La respuesta del modem ante un comando es la siguiente:

<CR><LF>	OK	<CR><LF>
<i>PREFIJO</i>	<i>CÓDIGO</i>	<i>SUFIJO</i>

- El Código es la respuesta del módulo, OK si es correcto o ERROR si es operación fallida.
- El prefijo y sufijo son el salto de línea y retorno de carro.

Existen comandos que realizan alguna acción y se los escribe así:

AT	<X>
SUFIJO	ACCIÓN

- <X> es la acción que realizará el modem.

A continuación se indican algunos comandos AT que serán útiles para la configuración y utilización del módulo.

#### ***1.7.1.1. COMANDOS AT GENERALES***

- **AT**: para comprobar que el módulo responde correctamente a las instrucciones.
- **ATE**: eliminar eco.
- **ATI**: información del producto.

#### ***1.7.1.2. COMANDOS PARA INFORMACIÓN DEL EQUIPO***

- **AT+CGMI**: identificar nombre del fabricante.
- **AT+CIMI**: obtener el IMSI<sup>54</sup>.
- **AT+CGSN**: recabar el IMEI<sup>55</sup>.
- **AT+IPR?**: consultar el baud rate actual.
- **AT+IPR=?**: opciones de baud rate.
- **AT+IPR=< >**: determinar nuevo baud rate.

#### ***1.7.1.3. COMANDOS DEL SERVICIO DE RED***

- **AT+CPAS**: leer estado de modem.
- **AT+CSQ**: calidad de la señal.
- **AT+CREG**: registrarse a una red
- **AT+COPS?**: verificar el nombre de la compañía telefónica.

#### ***1.7.1.4. COMANDOS DE SEGURIDAD***

---

<sup>54</sup> Identificación de la Estación Móvil

<sup>55</sup> International Mobile System Equipment Identity

- **AT+CPIN?**: para introducir código PIN<sup>56</sup>.
- **AT+CPWD**: cambiar clave.

#### **1.7.1.5. COMANDOS PARA REALIZAR LLAMADAS.**

- **ATD09XXXXXXXX;** : realizar llamada telefónica.
- **ATA**: contestar llamada.
- **ATH**: colgar llamada.
- **ATM**: conexión/desconexión de altavoz.

#### **1.7.1.6. COMANDOS PARA SMS**

- **AT+CMGF=1**: selección de modo SMS.
- **AT+CMGS="0912345678"**: escribir número que se va enviar SMS.
- **>0x1A**: indicar fin de SMS.
- **AT+CPMS**: seleccionar lugar de almacenamiento de los SMS.
- **AT+CPMS="SM"**: leer memoria de SIM.
  - **+CPMS:<X ><Y>**
  - X**: número de SMS almacenado.
  - Y**: número máximo de SMS que se puede almacenar.
- **AT+CMGR**: leer un SMS almacenado.
- **AT+CMGW**: almacenar mensaje en memoria.
- **AT+CMSS**: enviar mensaje almacenado.

#### **1.7.1.7. COMANDOS PARA ENVIÓ DE DATOS:**

- **AT+CGATT**: conectar o desconectar servicio GPRS.
- **AT+CGATT?**: comprobar si la MS está conectada a la red GPRS.
- **AT+CSTT**: empezar la tarea y set APN<sup>57</sup>, nombre de usuario, contraseña.
- **AT+CSTT=<APN>**: un parámetro de cadena (se lo incluye entre comillas), que indica el nombre del punto de acceso GPRS.
- **AT+CIICR**: traer conexión wireless con GPRS o CSD<sup>58</sup>.

---

<sup>56</sup> Número de Identificación Personal.

<sup>57</sup> Access Point Name

<sup>58</sup> Circuit Switched Data

- **AT+CIFSR:** obtener dirección IP local.
- **AT+CIPSPRT:** set puntual de ‘>’ cuando el módulo envía datos.
- **AT+CIPSTART:** iniciar la conexión TCP<sup>59</sup> o UDP<sup>60</sup>.
- **AT+CIPCLOSE:** TCP o UDP cerrar.

## 1.8. SENSORES

### 1.8.1. SENSOR DE TEMPERATURA LM35

Producidos por National Semiconductor Corporation, adecuados para intervalos de -55°C a 150°C y no requieren ninguna calibración externa. Se polariza de 4 a 30 voltios con salida de 10 mV por grado Celsius. Se observa en la **figura N°21** su apariencia física y su distribución de pines.

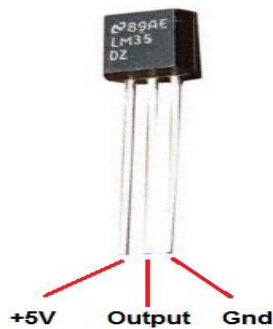


FIGURA N° 21.- Sensor LM35

FUENTE: <http://www.learningaboutelectronics.com/Articles/LM35-temperature-sensor-circuit.php>

Para determinar el valor de temperatura se debe convertir en un número digital el voltaje de salida (*máximo 1,5mV cuando está a 150°C*), utilizando el módulo ADC del microcontrolador. Todo esto se verá definido por las siguientes relaciones, en las que como ejemplo se toma la referencia de 5V y entrega un resultado binario de 1024 cuando el voltaje es de 5V:

$$\frac{1024}{5V} = \frac{Radc}{Vadc}$$

Ecuación 1

Despejando Vadc:

---

<sup>59</sup> Transmission Control Protocol

<sup>60</sup> User Datagram Protocol

$$V_{adc} = \frac{(5V)(R_{adc})}{1024}$$

Ecuación 2

De la relación del sensor: 10mV por grado Celsius:

$$\frac{10mV}{1^{\circ}C} = \frac{V_{adc}}{n^{\circ}C}$$

Ecuación 3

De esta ecuación se puede deducir que la temperatura es:

$$n^{\circ}C = \frac{(1^{\circ}C)(V_{adc})}{10mV}$$

Ecuación 4

Reemplazando la ecuación 2 en la 4, se tiene:

$$n^{\circ}C = \frac{(5)(R_{adc})}{10.23} = 0.48828125R_{adc}$$

Ecuación 5

En la programación del microcontrolador se utilizara esta relación en la conversión AD.

### 1.8.2. SENSOR LDR<sup>61</sup>

Dispositivo que cambia su valor de resistencia en función de la luz, cuanto mayor sea la luz menor es la resistencia. Se lo puede apreciar en la **figura N°22**.



FIGURA N° 22.-Sensor Fotoresistor LDR

FUENTE: <http://spanish.alibaba.com/product-gs/large-20mm-ldr-photoresistor-sensor-photo-sensor-502336427.html>

---

<sup>61</sup> Light Dependent Resistor



Al hacer la lectura en el analógico del microcontrolador se obtiene valores de 0 a 1024, que son números no descriptivos. Lo que se quiere es una salida en Lux, el LDR no es muy preciso en estas lecturas, pero existe una relación áspera que transforma la resistencia del LDR en lux, que es:

$$Lux = 500/R_{ldr}$$

Ecuación 6

- $R_{ldr}$  es la resistencia del LDR.

Del divisor de voltaje de la **figura N°23** se implementa la siguiente ecuación:

$$V_{out} = \frac{R_{ldr} * V_{cc}}{R1 + R_{ldr}}$$

Ecuación 7

Despejando  $R_{ldr}$ :

$$R_{ldr} = \frac{V_{out} * R1}{V_{cc} - V_{out}}$$

Ecuación 8

Remplazando la ecuación 8 en la 6, se tiene:

$$Lux = \frac{500(V_{cc} - V_{out})}{V_{out} * R1}$$

Ecuación 9

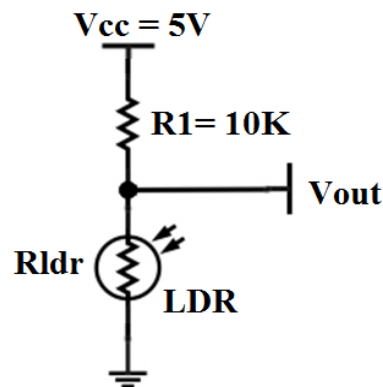


FIGURA N° 23.- Conexión de LDR

FUENTE: <http://www.tr3sdland.com/2011/11/componentes-el-sensor-ldr/>

Como el valor de salida  $V_{out}$  ingresa a una entrada AD del microcontrolador se puede asumir como  $V_{adc}$ , con un voltaje de referencia de 5V y remplazar en la ecuación 2.

### 1.8.3. SENSOR DE GAS MQ-6

Sensor resistivo de alta sensibilidad y respuesta rápida. Utilizado para detectar GLP<sup>62</sup>, butano, metano y alcohol. Detecta concentraciones en rangos dentro de las 200 a 10000 ppm<sup>63</sup>. Se lo observa en la **figura N° 24**.



FIGURA N° 24.-Sensor de gas MQ-6  
FUENTE: <http://www.oddwires.com/mq-6-lpg-sensor/>

El circuito para las conexiones del sensor se muestra en la **figura N° 25**. La resistencia de protección del sensor y la resistencia ajustable forman una resistencia de carga ( $R_L$ ), que forma un divisor de voltaje con la resistencia del sensor ( $R_s$ ).

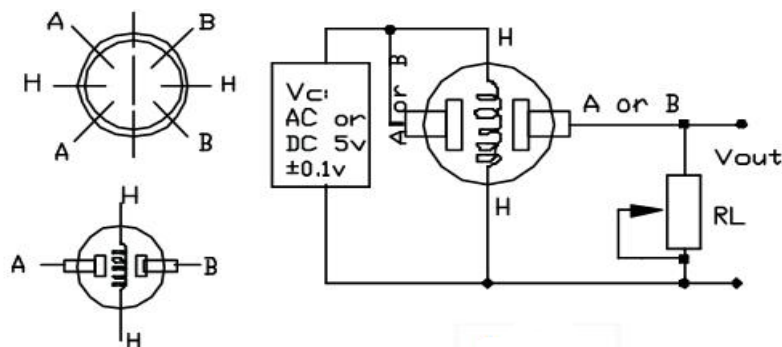


FIGURA N° 25.-Circuito para MQ-6  
FUENTE: <http://playground.arduino.cc/Main/MQGasSensors>

<sup>62</sup> Gas Licuado de Petróleo

<sup>63</sup> Particiones Por Millón

Como  $V_{out}$  ingresa a una entrada AD del microcontrolador se puede asumir como  $V_{adc}$  de la ecuación 2, con un voltaje de referencia de 5V. Para obtener la concentración de gas se debe calcular la resistencia del sensor por medio del divisor de voltaje con la siguiente ecuación:

$$R_s = \frac{V_c - V_{out}}{V_{out}}$$

Ecuación 10

En la **figura N° 26** se muestra la gráfica de la relación de la resistencia del sensor ( $R_s/R_o$ ), versus la concentración de gas en ppm para MQ-6.  $R_o$  es el valor de  $R_s$  a 1000ppm, basado en el grafico  $R_o$  es igual a 1. En la siguiente ecuación se muestra la relación entre  $R_s$  y el valor de concentración de GLP:

$$GLP_{ppm} = 1000 * R_s^{1/\alpha}$$

Ecuación 11

Donde,  $\alpha$  muestra el valor de la pendiente en la **figura N°25**, el valor se lo puede obtener con la siguiente ecuación:

$$\alpha = \frac{\log x_1 - \log x_2}{\log y_1 - \log y_2}$$

Ecuación 12

Tomando los puntos de la gráfica, se tiene que la pendiente de la concentración de GLP es igual a -1. Y reemplazando y simplificando en la ecuación 12 se tiene:

$$GLP_{ppm} = \frac{1000}{R_s} ppm$$

Ecuación 13

El mismo método se aplica para calcular la concentración de gas de alcohol, se obtiene la siguiente ecuación:

$$ALCOHOL_{ppm} = \frac{0.4}{R_s^{1.54}} ppm$$

Ecuación 14

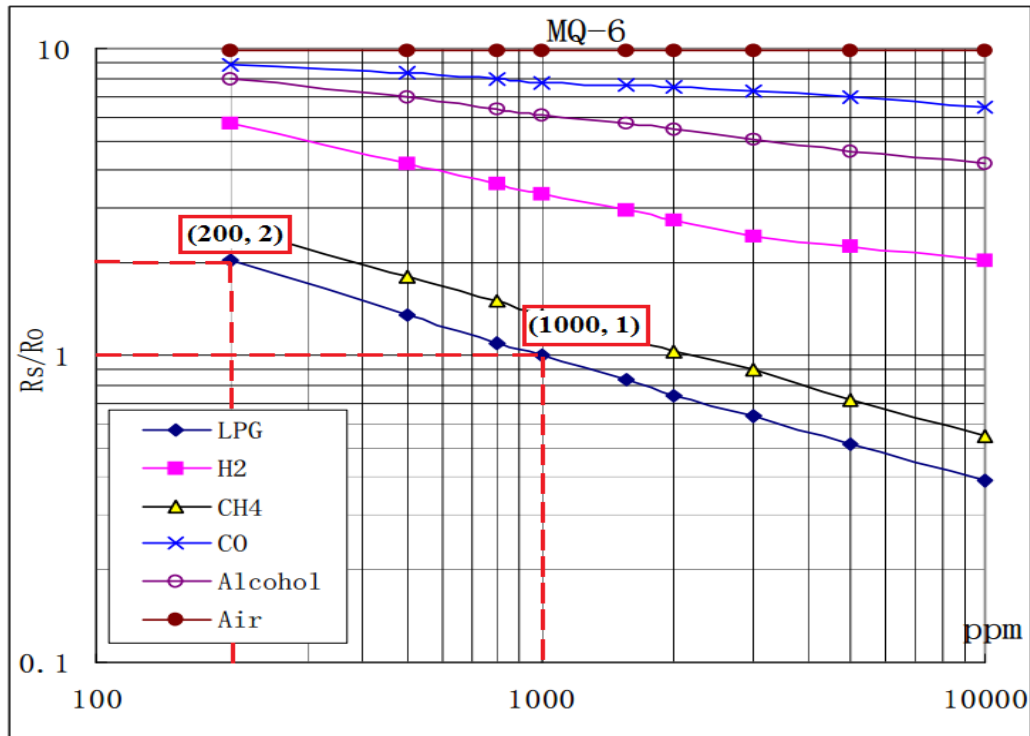


FIGURA N° 26.-Características de sensibilidad del MQ-6  
 FUENTE: <https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-6.pdf>

#### 1.8.4. SENSOR DE HUMEDAD Y TEMPERATURA DHT11

Sensor con salida de datos digital calibrada de la temperatura y humedad. No requiere interfaces como I2C o SPI, utiliza su propio protocolo para comunicarse por un solo conductor hasta los pines de I/O en un microcontrolador. En la **figura N°27** se observa su aspecto físico y como se debe conectar.

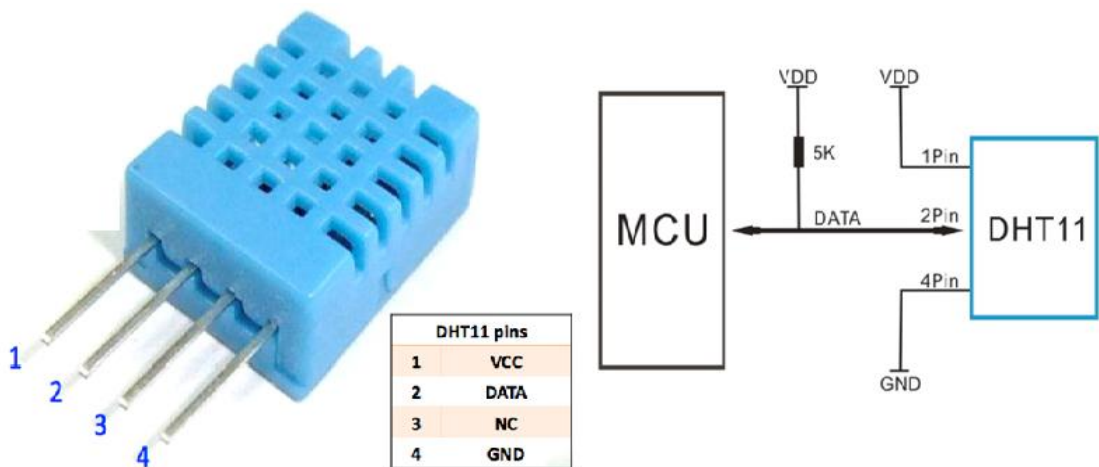


FIGURA N° 27.- Sensor DHT11 y su conexión  
 FUENTE: <http://electronics.stackexchange.com/questions/140670/dht11-temperature-humidity-sensor>

Para su funcionamiento, el microcontrolador debe iniciar la comunicación con el DHT11 manteniendo la línea de datos en estado bajo durante 18ms. Luego el sensor envía una respuesta con un pulso bajo de 80µs y luego deja flotar la línea de datos por otros 80µs (Jesús, 2013). En la **figura N°28**, se observa estas réplicas.

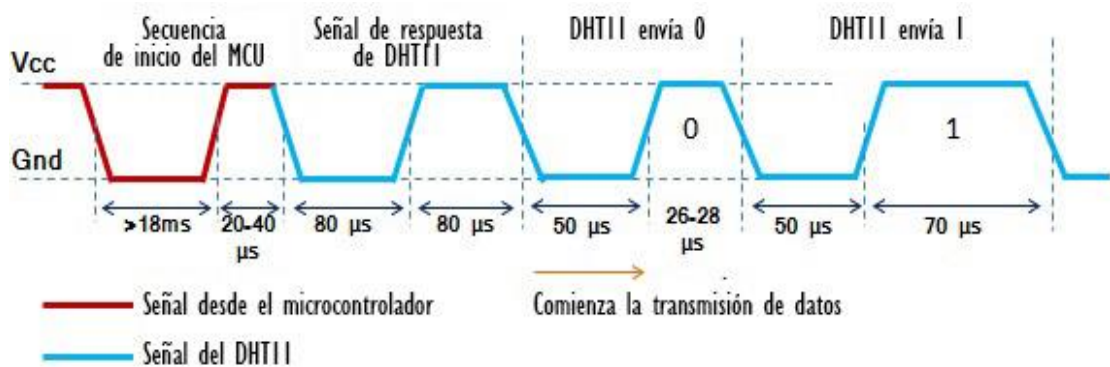


FIGURA N° 28.- Comunicación del DHT11 y el microcontrolador  
 FUENTE: <http://www.geekfactory.mx/tutoriales/tutoriales-pic/dht11-con-pic/>

## 1.9. MICROCONTROLADOR

“Un microcontrolador es un dispositivo electrónico encapsulado de alto nivel de integración. Los microcontroladores se pueden adquirir comercialmente de diferentes casas fabricantes como: Atmel, Microchip, Freescale, Motorola, Intel y Philips.” (Clavijo Mendoza, 2011)

### 1.9.1. PIC 16F877A

“El PIC 16F877A es un microcontrolador de Microchip Technology fabricado en tecnología CMOS, su consumo de potencia es muy bajo y además es completamente estático. Tiene una memoria de programa tipo FLAH, lo que presenta gran facilidad en el desarrollo de prototipos” (Morlote, 2013).

En la **figura N°29** se observa la distribución de los pines del 16F877A y en la **Tabla N°2** se indica las características más relevantes de este microcontrolador

### 40-Pin PDIP

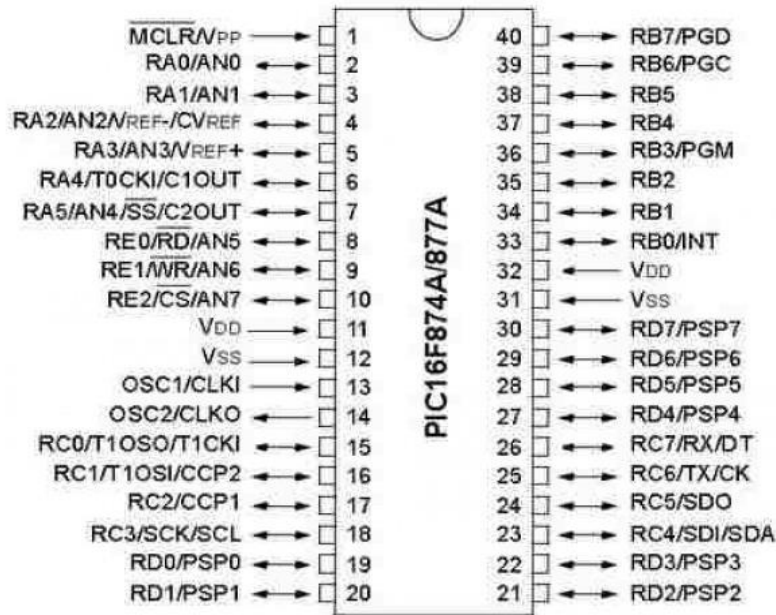


FIGURA N° 29.- Distribución de pines del 16F877A  
 FUENTE: <http://akizukidenshi.com/download/PIC16F87XA.pdf>

Tabla N° 2.-Características del PIC 16F877A  
 FUENTE: <http://akizukidenshi.com/download/PIC16F87XA.pdf>

CARACTERÍSTICA	
Frecuencia máxima de operación	DX-20MHz
Resets (y delays)	POR, BOR, (PWRT, OST)
Memoria de programa flash (palabras de 14 bits)	8K
Posiciones de RAM de datos	368 bytes
Posición EEPROM de datos	256 bytes
Interrupciones	15
Puertos E/S	A, B, C, D, E
Número de pines	40
Timers	3
Módulos CCP <sup>64</sup>	2
Comunicaciones Serie	MSSP, USART
Comunicaciones Paralelo	PSP

<sup>64</sup> Captura/Comparar/módulo PWM

Líneas de entrada analógicas de 10 bits	8
Comparadores analógicos	2
Conjunto de instrucciones	35 instrucciones
Longitud de instrucciones	14 bits
Arquitectura	Harvard
CPU	Risc
Canales PWM	2
Paquetes	40-pin PDIP <sup>65</sup>

### 1.9.2. PIC 18F2550

De Microchip Technology Inc., microcontrolador que alcanza velocidades de hasta más de 16 MIPS<sup>66</sup>, dispone de un puerto USB y soporta transferencias de control, interrupción e isócronas. Se lo encuentra en encapsulado DIP, para el montaje sobre placas; la funcionalidad de este PIC puede verse en la **Tabla N°3**.

Tabla N° 3.-Características del 18F2550  
FUENTE: <http://ww1.microchip.com/downloads/en/DeviceDoc/39632e.pdf>

<b>CARACTERÍSTICA</b>	<b>ESPECIFICACIÓN</b>
Frecuencia máxima de operación	DC-48MHz
Memoria de programa (Bytes)	32768
Memoria de programa (Instrucciones)	16384
Memoria de datos (Bytes)	2048
Memoria EEPROM de datos (Bytes)	256
Fuentes de Interrupción	19
Puertos E/S	A, B, C, (E)
Número de pines	28
Timers	4
Módulos CCP	0
Comunicaciones Serie	MSSP, USART mejorado

<sup>65</sup> Pin Dual in-line package

<sup>66</sup> Millones de instrucciones por segundo

Módulo de Bus Serial Universal (USB)	1
Comunicaciones Paralelo	No
Líneas de entrada analógicas de 10 bits	10 canales de entrada
Comparadores	2
Resets (y delays)	POR, BOR, Reset Instruction, (PWRT, OST), MCLR (opcional), WDT
Programable detector de baja tensión	Si
Programable reset Brown-out	Si
Conjunto de instrucciones	75 instrucciones
Paquetes	28-pin PDIP
Periféricos avanzados	<ul style="list-style-type: none"> <li>• CAN<sup>67</sup></li> <li>• USB</li> <li>• Ethernet</li> </ul>

En la **figura N°30** se observa la distribución de los pines del 18F2550.

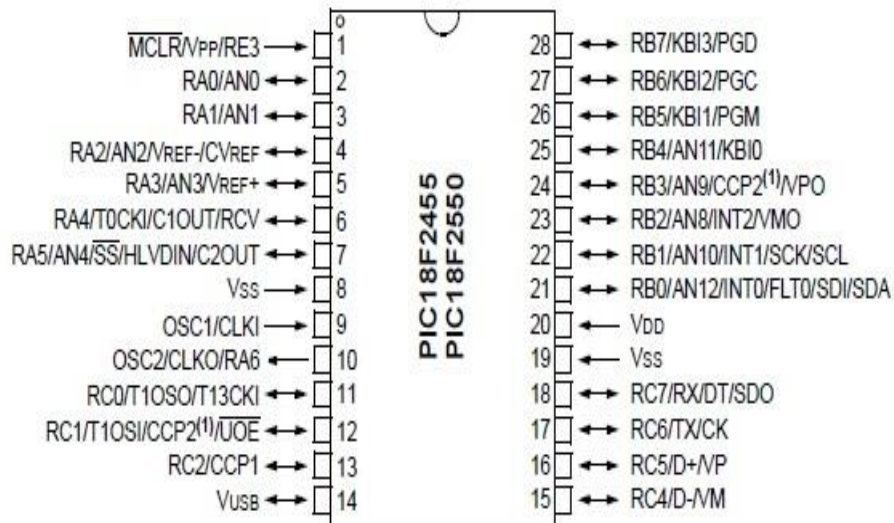


FIGURA N° 30.- Distribución de pines del 18F2550  
FUENTE: <http://ww1.microchip.com/downloads/en/DeviceDoc/39632e.pdf>

<sup>67</sup> Control Área Network



### 1.9.3. ATmega 328P

El ATmega 328P es un microcontrolador de alto rendimiento que está basado en un microcontrolador RISC, construido por Atmel y pertenece a la serie megaAVR y es el más común y utilizado en Arduino; alcanza una respuesta de 1MIPS, balanceando el consumo de energía y velocidad de proceso. En la **Tabla N°4**, se muestran las características de este microcontrolador.

Tabla N° 4.-Características del ATmega 328P

FUENTE: [http://www.atmel.com/images/atmel-8271-8-bit-avr-microcontroller-atmega48a-48pa-88a-88pa-168a-168pa-328-328p\\_datasheet\\_complete.pdf](http://www.atmel.com/images/atmel-8271-8-bit-avr-microcontroller-atmega48a-48pa-88a-88pa-168a-168pa-328-328p_datasheet_complete.pdf)

CARACTERÍSTICA	ESPECIFICACIÓN
Frecuencia máxima de operación	20MHz
Tamaño de núcleo	8-bits
Capacidad de memoria de programa	32KB (16K x 16)
Tipo de memoria de programa	Memoria FLASH
Capacidad de RAM	2K x 8
Capacidad de EEPROM	1K x 8
Número de pines	28
Puertos E/S	23
Conectividad	I <sup>2</sup> C, SPI, UART/USART
Alimentación de voltaje	1.8 V – 5.5 V
Rangos de temperatura	-40°C a 86°C
Características especiales	-Calibración de oscilador interna -Interrupción interna y externa. -Detección programable de falla.
Grados de velocidad	<ul style="list-style-type: none"><li>• 0 – 4 MHz a 1.8 – 5.5V</li><li>• 0 – 10 MHz a 2.7 – 5.5V</li><li>• 0 – 20 MHz a 4.7 – 5.5V</li></ul>
Consumo de energía a 1 MHz, 1.8V, 25°C	-Modo Activo: 0.2mA -Modo Power-down: 0.1µA -Modo Power-save: 0.75µA

En la **figura N°31** se observa la distribución de los pines del microcontrolador.

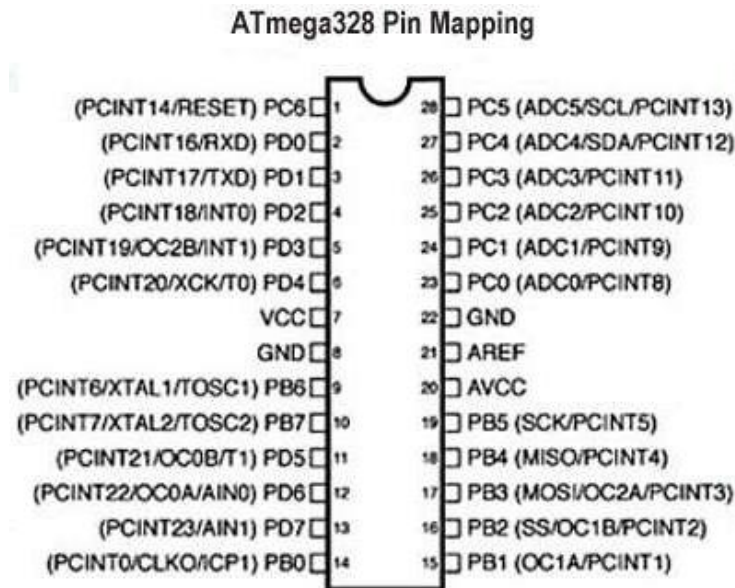


FIGURA N° 31.-Distribución de pines del ATmega 328P  
 FUENTE: <http://www.qtechknow.com/Tutorials/3>

## CAPÍTULO II

### 2. METODOLOGÍA

Este capítulo presenta la metodología manejada para el desarrollo de la investigación.

#### 2.1.TIPO DE ESTUDIO

- **Empírica:** se basa en la recolección de datos de los módulos que intervendrán en el entrenador, para poder analizar sus características y encaminarlas al diseño de los módulos.
- **Metodología de medición:** para determinar la eficiencia de los módulos de comunicación y demostrar su correcto funcionamiento.
- **Cuasi-experimental:** se tendrá el control de forma parcial de las variables de la investigación, puesto que, se depende de las características de los distintos tipos de módulos para realizar el diseño del entrenador.
- **Cuantitativa:** al término de la presente investigación se escogerá un grupo de estudiantes, a quienes se hará desarrollo de alguna práctica para determinar si el entrenador satisface las necesidades de los educandos, para realizar aplicaciones con los módulos de comunicación.

#### 2.2.POBLACIÓN Y MUESTRA

##### 2.2.1. POBLACIÓN

Para el uso del entrenador, la población está representada por los estudiantes de la Escuela de Electrónica y Telecomunicaciones de la Universidad Nacional de Chimborazo, en especial para aquellas personas que trabajan en aplicaciones y programación de microcontroladores.

En el desarrollo de la prácticas de monitoreo de sensores por medio de la red celular, la población es determinada por los datos obtenidos en las pruebas a distancias de comunicación entre el usuario y el sistema, estableciendo los tiempos de latencia.

### 2.2.2. MUESTRA

Para determinar el tamaño de la muestra por el uso del entrenador, es establecida de acuerdo al cálculo con la siguiente formula según Abraham de Moivre (1667 - 1754):

$$n = \frac{N\sigma^2 Z^2}{(N - 1)e^2 + \sigma^2 Z^2}$$

Donde los parámetros son:

- $n$  = El tamaño de la muestra.
- $N$  = Tamaño de la población, N° de estudiantes que reciben la materia de microcontroladores.
- $\sigma$  = Desviación estándar de la población que, generalmente cuando no se tiene su valor, suele utilizarse un valor constante de 0,5.
- $Z$  = Valor obtenido mediante niveles de confianza. Es un valor constante que, si no se tiene su valor, se lo toma en relación al 95% de confianza equivalente a 1,96 (como más usual) o en relación al 99% de confianza equivale 2,58; valor que queda a criterio del investigador.
- $e$  = Limite aceptable de error muestral que, generalmente cuando no se tiene su valor, suele utilizarse un valor que varía entre el 1% (0,01) y 9% (0,09), valor que queda a criterio del encuestador. (Galarza, 2015)

$$n = \frac{(20)(0.5)^2(1.96)^2}{(20 - 1)(0.09)^2 + (0.5)^2(1.96)^2}$$

$$n = 17,237$$

$$n = 17 \text{ muestras}$$

En el caso del desarrollo de la práctica la población carece de registros identificables, porque el número de pruebas puede ser infinito, por lo que la muestra se establece con la siguiente formula:

$$n = \frac{Z_{\infty}^2 * p * q}{i^2}$$

Donde cada parámetro es:

- $Z_{\infty}$  =Distribución de Gauss donde  $Z_{\infty} = 0.05 = 1.96$
- $p$  = prevalencia esperada de parámetro a evaluar, el valor tomado es un porcentaje alto de prevalencia  $p=0.9$  es decir el 90 %.
- $q = 1-p$
- $i$  = error considerado que se prevé cometer es del 10%,  $i=0.1$

$$n = \frac{1.96^2 * 0.9 * (1 - 0.9)}{0.1^2}$$

$$n = \frac{0.3457}{0.01}$$

$$n = 34.57$$

$$n = 35 \text{ muestras}$$

### 2.2.3. HIPÓTESIS

*“Con el diseño e implementación de un entrenador de microcontroladores para comunicaciones, usando la red celular se logrará el monitoreo de sensores para el desarrollo de las prácticas.”*

La hipótesis planteada es de tipo descriptiva, ya que involucra dos variables y declara que el entrenador de microcontroladores para comunicaciones diseñado e implementado podrá desarrollar prácticas de monitoreo de sensores por medio de la red celular.

### 2.3. OPERACIONALIZACIÓN DE VARIABLES

Tabla N° 5.-Operacionalización de variables  
FUENTE: Los Autores

VARIABLES	CONCEPTO	CATEGORÍA	INDICADORES	TÉCNICAS E INSTRUMENTO
INDEPENDIENTE	<b>Módulo GSM</b>  Dispositivo del entrenador que permite la comunicación con la red celular, cumpliendo los estándares de comandos AT, para poder desarrollar las prácticas.	-Comunicación -Red celular -Comandos AT -Desarrollo	-Prácticas y experimentación.  -Aplicaciones de comunicación vía red celular.  -Guía para realizar prácticas y posteriormente evaluarlas.	<b>Técnicas:</b> -Observación -Experimentación -Mediciones convencionales  <b>Instrumentos:</b> -Fichas de observación -Material experimental -Unidades de medida -Libretas de apuntes
DEPENDIENTE	<b>Entrenador</b>  Herramienta para desarrollar prácticas con microcontroladores. -Transportable -Modular -Autoalimentado -Independiente -Gama de microcontroladores.	-Equipo -Prácticas -Modular	-Montaje y pruebas del entrenador.  -Ensamblaje del módulo de comunicación vía red celular	<b>Técnicas:</b> -Observación -Análisis documental  <b>Instrumentos:</b> -Lista de cotejo -Análisis de contenido -Libretas de apuntes

## 2.4.PROCEDIMIENTOS

Refiere los pasos para el proceso de la investigación que serán ampliados a continuación, con el diseño de placas para el entrenador y desarrollo de un sistema de monitoreo de sensores y conmutación de relés controlados desde un teléfono celular, aplicado al control de las condiciones climáticas de un galpón de aves.

### 2.4.1. DISEÑO DE LAS PLACAS ELECTRÓNICAS PARA EL ENTRENADOR

Se diseñaron las placas electrónicas en el simulador Proteus 8.1 Professional, porque permite simular los circuitos y también el diseñar las placas de acuerdo a las características que se busca obtener.

- En la interfaz de ISIS de Proteus, buscar los elementos electrónicos para el simular el circuito, ir a *Component Mode* dentro de la paleta *DEVICES*, seleccionar *P* y explorar para encontrar los elementos. **Figura N°32.**

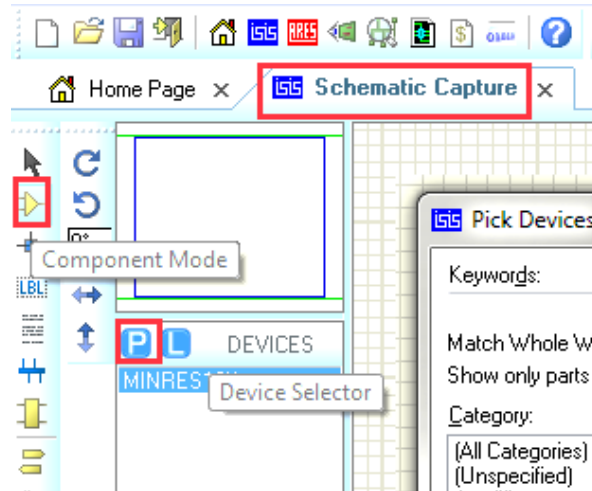


FIGURA N° 32.-Interfaz de ISIS – Proteus  
FUENTE: Autores

- Traspasar todos los elementos a la interfaz ARES de Proteus, ir a *Component Mode*, e ir seleccionando los componentes; organizarlos de la mejor manera para que ocupe el área de la placa. Mandar a rutear hasta conseguir la forma de la placa final, sin ningún error. **Figura N°33.**

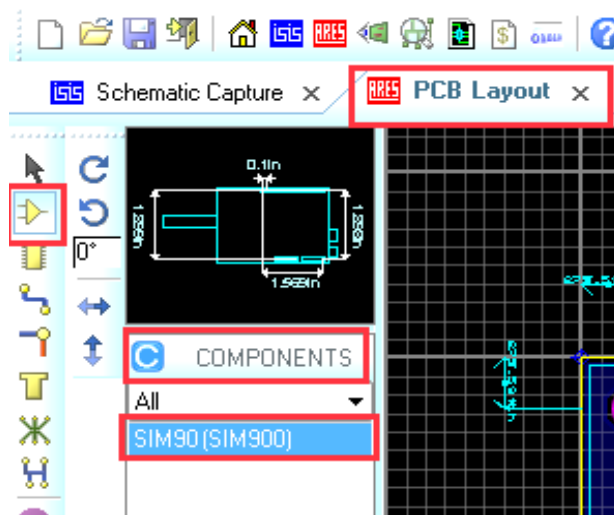


FIGURA N° 33.-Interfaz de ARES – Proteus  
FUENTE: Autores

#### 2.4.1.1.DISEÑO DE PLACA: MÓDULO SIM900

##### 2.4.1.1.1. Circuito electrónico para módulo SIM900





## 2.4.1.2. DISEÑO DE PLACA: T-SV651, MAX232 Y MAX485

### 2.4.1.2.1. Circuito electrónico: Transeiver SV651, MAX232 y MAX485

Para el transeiver SV651 conectar un LED indicador de activación, y una regleta para las salidas y la alimentación, ver **Figura N°36**.

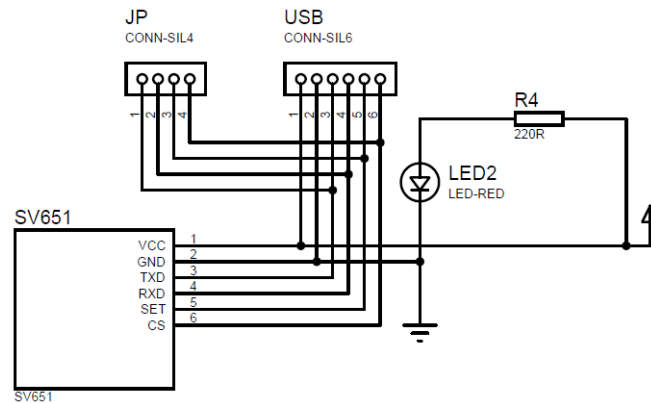


FIGURA N° 36.- Circuito para Transeiver SV651  
FUENTE: Autores

El circuito MAX232 posee condensadores de 10uF conectados como se observa en la **Figura N°37**, así también el circuito posee un LED indicador de conexión y una regleta para las salidas del MAX.

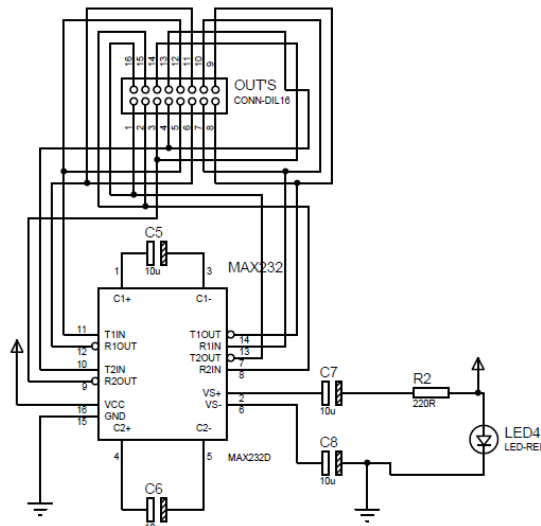


FIGURA N° 37.-Circuito MAX232  
FUENTE: Autores

Para transmisiones full-duplex se juntó el circuito del MAX485, que posee un LED indicador de activación y una regleta para las salidas, sus conexiones se pueden apreciar en la **Figura N°38**.

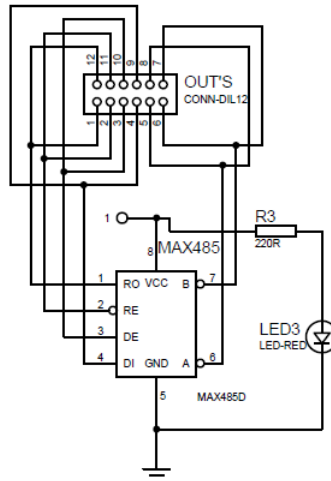


FIGURA N° 38.- Circuito MAX485  
FUENTE: Autores

El esquema también tiene reguladores 7805 y 7833, para alimentar al transeiver, los MAX y para dejar salidas que sirvan de alimentación para otros circuitos, ver **Figura N°34**. El esquema de la placa en el **ANEXO 1** (circuito 2).

#### 2.4.1.2.2. Placa electrónica: Transeiver SV651, MAX232 y MAX485

Incorporar los elementos de tal forma que se ocupe la demisión de la placa (10x 10 cm), acomodando por separado los distintos circuitos y dejando las regletas de salida en los bordes de la placa. El diseño se o puede observar en el **ANEXO 2** (placa N°2) y la placa vista en 3D en la **Figura N°39**.

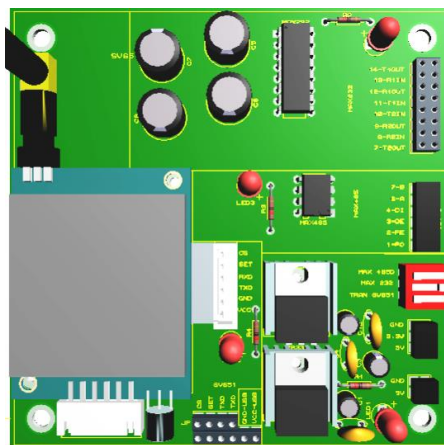


FIGURA N° 39.-Placa para Transeiver SV651, MAX232 y MAX485 en 3D  
FUENTE: Autores

#### 2.4.1.3.DISEÑO PLACA: TECLADO, PULSADORES Y DIPSWITCHS

##### 2.4.1.3.1. Circuito electrónico: Teclado, pulsadores y dipswitchs

El teclado es de matriz 4x4 y tiene conectado resistencias a los pines 1, 2, 3 y 4 para generar las señales lógicas; además tiene un switch para seleccionar la lógica que se desea trabajar: positiva o negativa, ver **Figura N°40**.

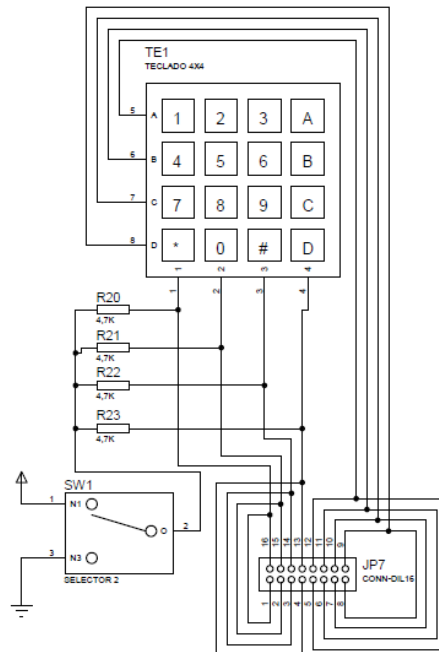


FIGURA N° 40.-Circuito de teclado  
FUENTE: Autores

Se dotó de dos circuitos para que los dipswitchs tengan lógica positiva o negativa, ver **Figura N°41**.

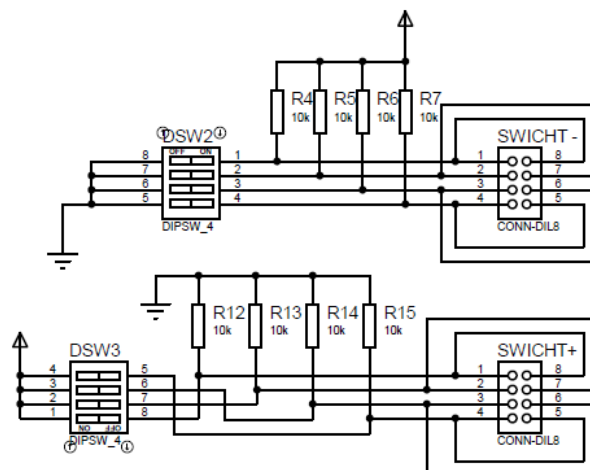


FIGURA N° 41.-Circuitos Dipswitchs  
FUENTE: Autores

Al igual que los dipswitchs los botones tiene lógica positiva o negativa para lo cual se armó el esquema de la **Figura 42**.

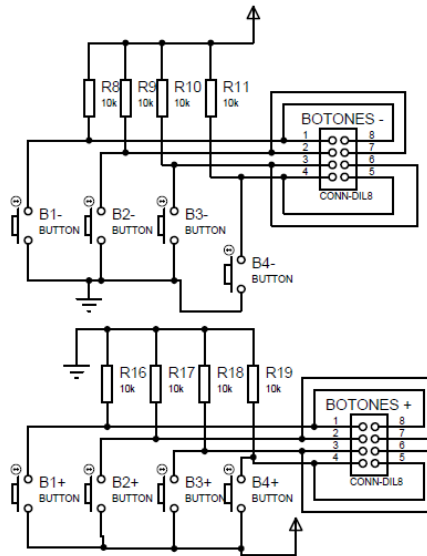


FIGURA N° 42.-Circuito de botones  
FUENTE: Autores

También tiene reguladores 7805 y 7833, para alimentar el sistema y dejar salidas que sirvan de alimentación para otros circuitos, ver **Figura N°34**. El esquema completo del circuito se observa en el **ANEXO 1** (circuito 3).

#### 2.4.1.3.2. Placa electrónica: Teclado, pulsadores y dipswitchs

Acoplar los distintos elementos que interceden en el esquema utilizando el espacio dispuesto para la placa, acomodando por separado los distintos circuitos y dejando las regletas de salida en los bordes de la placa. El diseño se o puede observar en el **ANEXO 2** (placa N°3) y la placa vista en 3D en la **Figura N°43**.

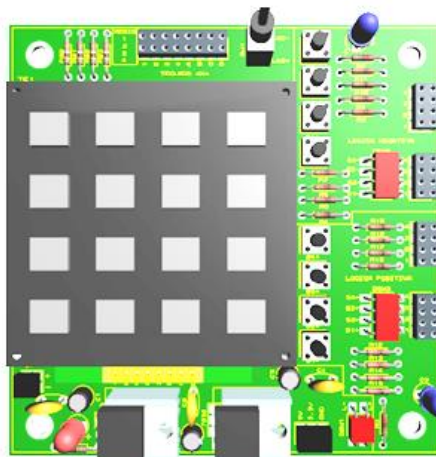


FIGURA N° 43.-Placa para Teclado, pulsadores y dipswitchs en 3D  
FUENTE: Autores

#### 2.4.1.4. ACOPLAMIENTO DE ELEMENTOS ELECTRÓNICOS SOBRE LAS PLACAS DEL ENTRENADOR

- Colocar todos los elementos sobre la placa.
- Soldar todos los elementos ver **Figura N°44**.

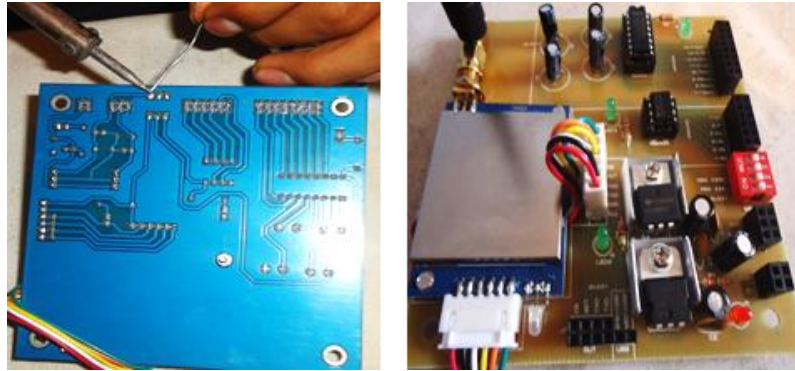


FIGURA N° 44.-Elaboración de la placa  
FUENTE: Autores

#### 2.4.1.5.MONTAJE DE LOS DISTINTOS MÓDULOS SOBRE EL ENTRENADOR

Acomodar los módulos de los distintos circuitos que componen el entrenador para realizar las distintas prácticas. **Figura N°45**.

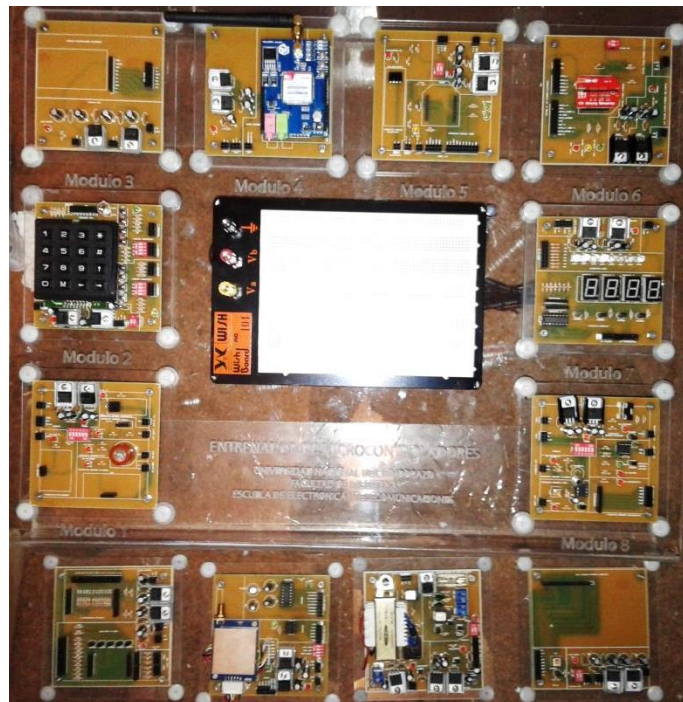


FIGURA N° 45.-Diseño del entrenador  
FUENTE: Autores

## 2.4.2. DISEÑO DE UN SISTEMA DE CONTROL DE TEMPERATURA DE UNA AVÍCOLA VÍA RED CELULAR

Este diseño, fue desarrollado para el monitoreo y control de las condiciones climáticas en una avícola, operado por medio de la red celular. Consta de un módulo GSM/GPRS (*SIM900*) anteriormente descrito, el cual es gestor para permitir la comunicación con la red de telefonía móvil y un microcontrolador. **Figura N°46.**

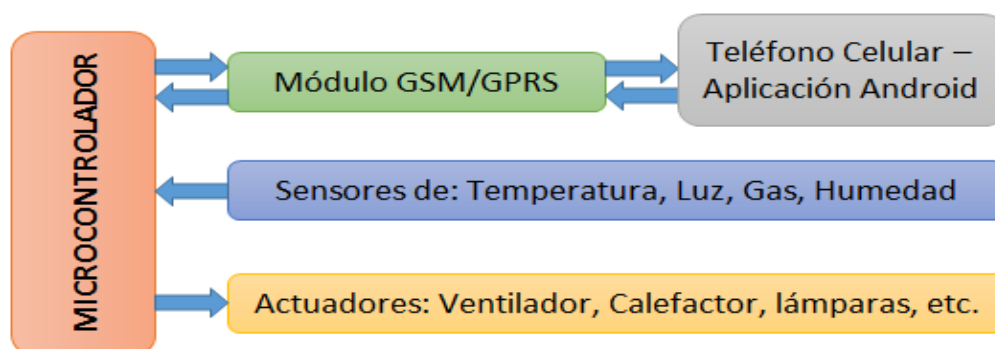


FIGURA N° 46.-Entradas y salidas del control de temperatura de una avícola  
FUENTE: Autores

### 2.4.2.1. CIRCUITO DE MONITOREO - SENSORES

El circuito de sensores, consta de un LM35 que medirá la temperatura, MQ-3 encargado de censar el gas, un LDR para determinar la intensidad de luz y un DHT11 administrador de los niveles de humedad. **Figura N°47.**

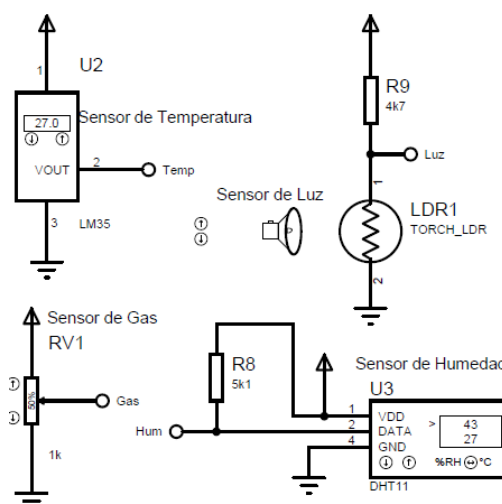


FIGURA N° 47.-Circuito de Sensores  
FUENTE: Autores

### 2.4.2.2. CIRCUITO DE CONMUTACIÓN - ACTUADORES

El circuito de conmutación está compuesto por un optoacoplador, que se basa su funcionamiento en señales luminosas para pasar señales de circuito a otro sin conexiones eléctricas, evitando así las señales de rebote generadas por las cargas protegiendo así lo diferentes sistemas que comanda el microcontrolador.

Para conectar el relé que servirá como interruptor para activar o desactivar el actuador, se debe encajar un transistor 2N3904 que proporcione la corriente necesaria para la bobina del relé, **Figura N°48**, el diodo sirve para eliminar los picos de tensión que se producen cuando se conecta o desconecta la bobina.

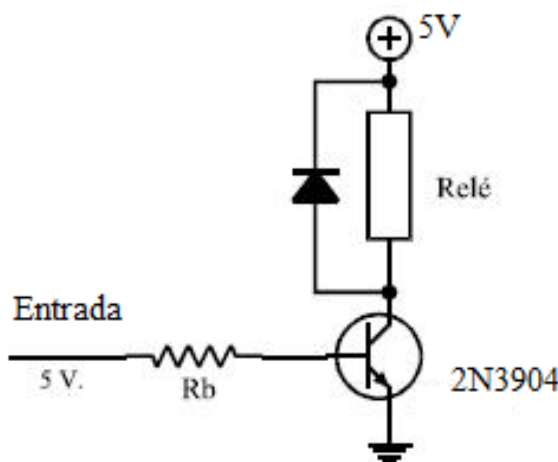


FIGURA N° 48.-Circuito de bobina de relé  
FUENTE: Autores

Revisando la hoja de datos del transistor se tiene los siguientes valores:  $I_C^{68}=100mA$ ,  $V_{CEO}^{69}=45V$  y su  $hFE^{70}=110$ , todo servirá para calcular la resistencia para colocar en la base. Como la corriente que necesita la bobina del relé es de 80mA, el suministro de corriente a la base es:

$$I_B = \frac{I_C}{hFE} = \frac{80mA}{110} = 0,72mA$$

Para asegurar que el transistor va a conmutar se debe suministrar una  $I_B=1mA$ .

<sup>68</sup> Corriente máxima que puede soportar el transistor

<sup>69</sup> Máximo voltaje que soporta el transistor

<sup>70</sup> Ganancia del transistor que se define como  $I_C/I_B$

Hay que tomar en cuenta la caída de tensión que tiene el voltaje que se aplica a la Base al atravesar el transistor por unión Base – Emisor es  $V_{BE}=0,7$ .

$$V_B = 5V - 0,7V = 4,3V$$

Aplicando ley de Ohm se puede encontrar la resistencia de la base:

$$R_B = \frac{4,3V}{1mA} = 4300\Omega$$

Comercialmente se puede encontrar resistencias de  $4,7K\Omega$  y  $3,9K\Omega$ , por lo que el diseñador puede tomar la decisión de cual utilizar.

El circuito completo de conmutación como se observa en la **Figura N°49**. Utilizar este circuito para todos los actuadores.

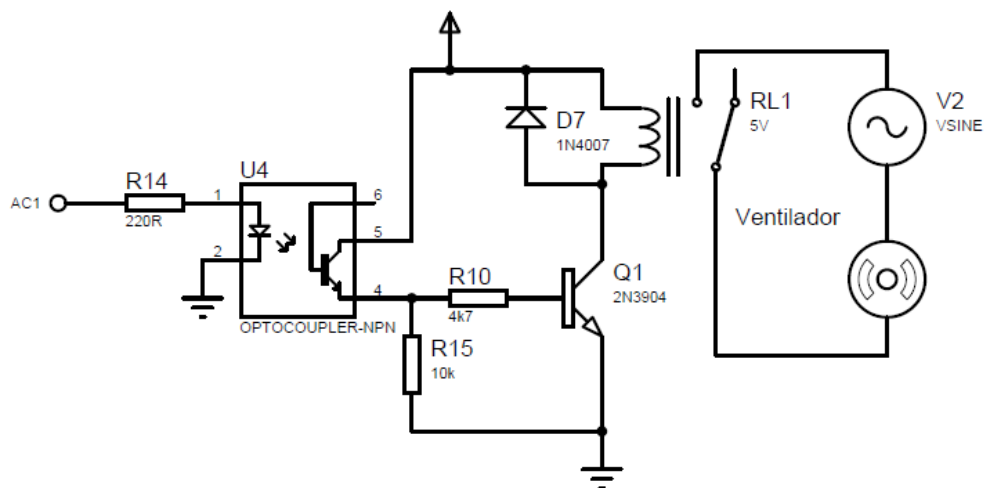


FIGURA N° 49.-Circuito de potencia para conmutación de actuadores  
FUENTE: Autores

### 2.4.2.3.COMUNICACIÓN CON EL MÓDULO GSM/GPRS

En primera pretensión se requiere verificar las funcionalidades del módulo SIM900 antes de intentar conectar a otro circuito, con la ayuda de una interfaz UART y un terminal serial, usando comandos AT se puede interactuar con el ejemplar. Para su proceso guiarse en el **ANEXO 3**, *Práctica de laboratorio N°1.1: Comunicación con el módulo SIM900 mediante comandos AT manuales*. En la **Figura N°50**, se observa la topología de la conexión de esta práctica.



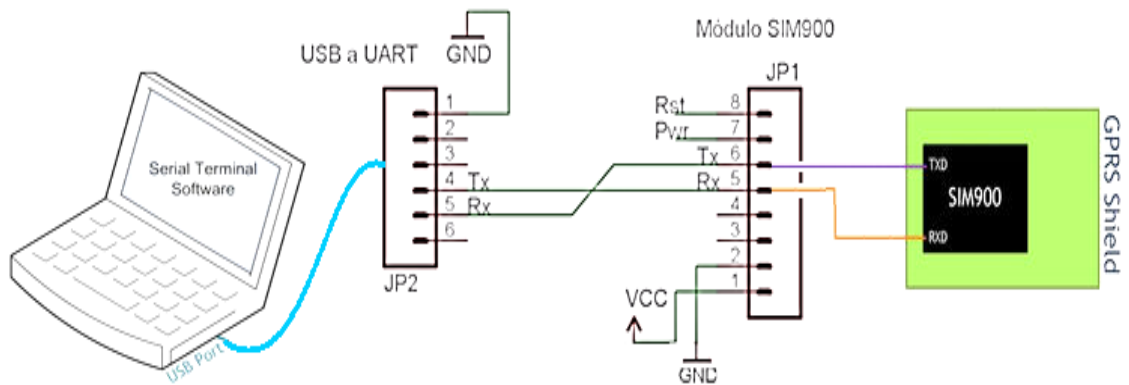


FIGURA N° 50.-Topología de conexión de módulo SIM900 a la PC  
FUENTE: Autores

#### 2.4.2.4.CIRCUITO DEL MICROCONTROLADOR

El circuito que se muestra en la **Figura N°51**, ejecutará el control para monitorear y manipular la temperatura de una avícola, está compuesto por un microcontrolador PIC 18F2550, que administra los siguientes procesos:

- Encendido del PIC.
- Recolectar información de los sensores.
- Indicadores de alerta.
- Reset del sistema.
- Comunicación Usart con el módulo GSM/GPRS.
- Procesamiento de la información recibida por el módulo.
- Envío de mensajes de texto.
- Salidas al circuito de conmutación para el control de actuadores.

#### 2.4.2.5.PROGRAMACIÓN DEL MICROCONTROLADOR

##### 2.4.2.5.1. Programación de mensajes de alerta enviados por los sensores

El diagrama de flujo de la **Figura N°52**, pauta la lógica que el microcontrolador cumple con la etapa de enviar mensajes de alerta, cuando algún sensor determine magnitudes fuera de su rango previamente establecido.

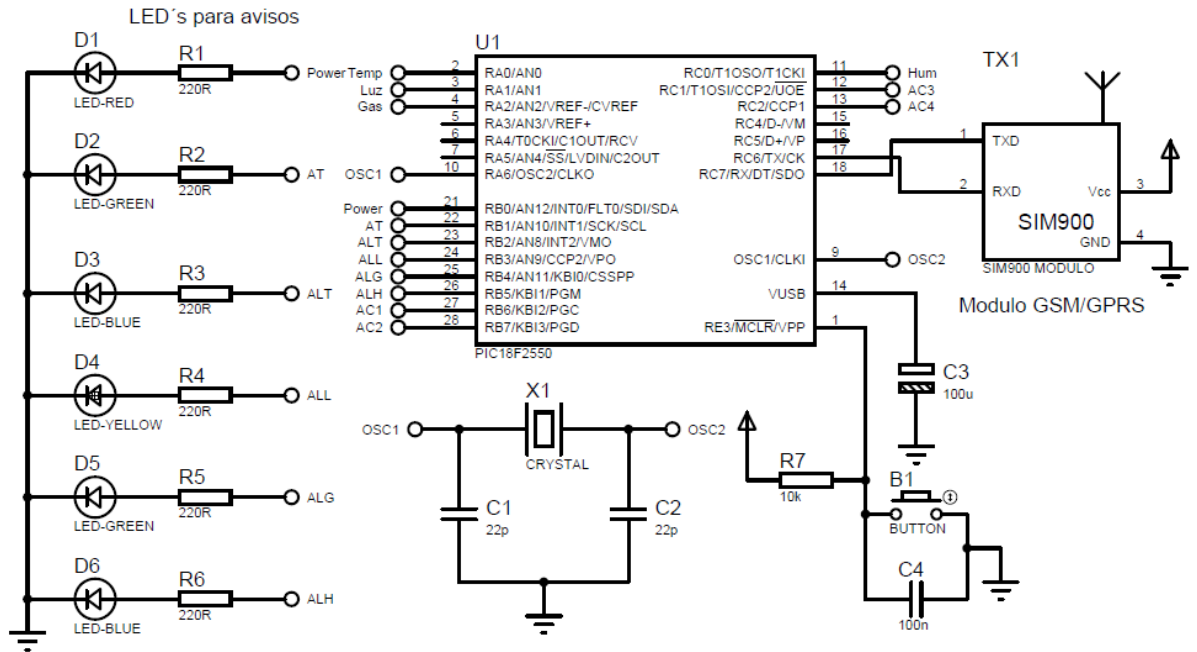


FIGURA N° 51.-Circuito de control digital para el monitoreo de sensores y manipulación de relés.  
FUENTE: Autores

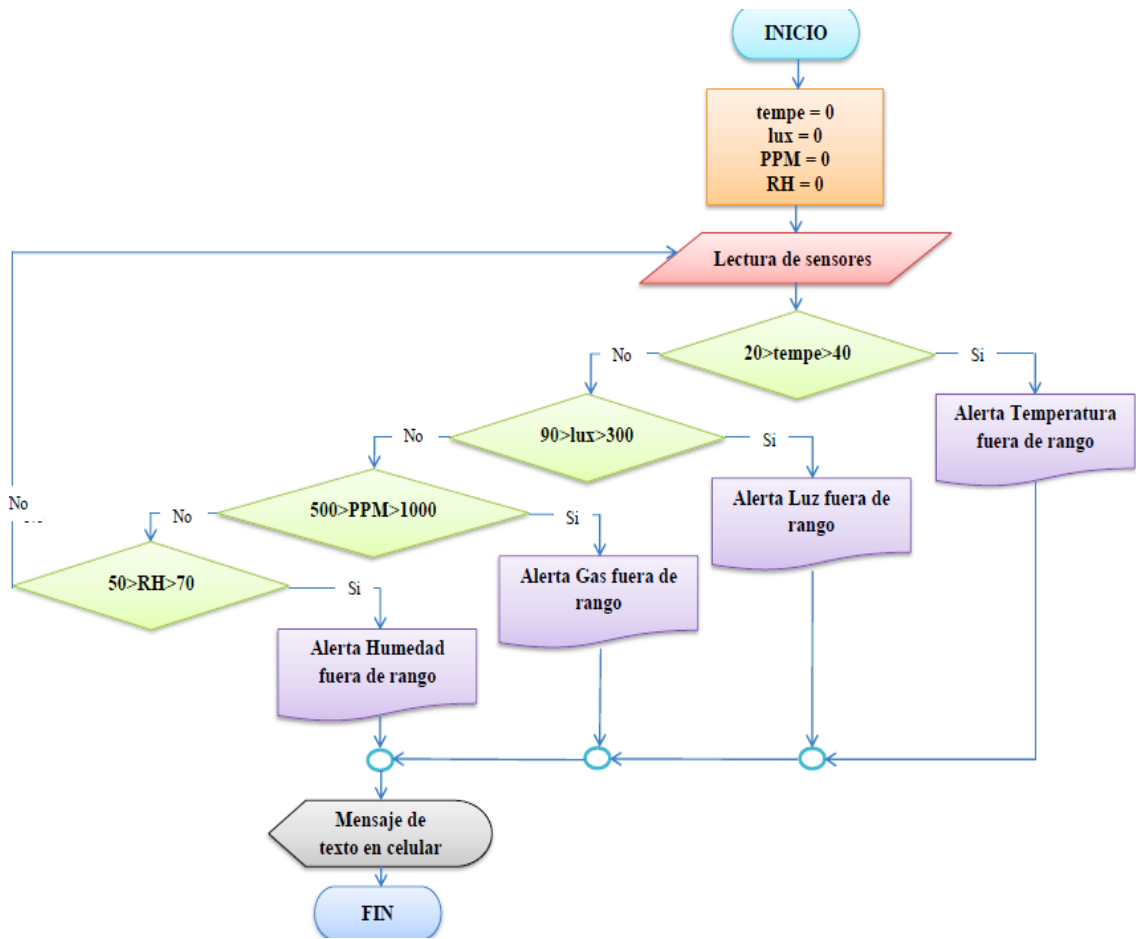


FIGURA N° 52.-Flujo grama de programación de mensajes de alerta enviados por sensores  
FUENTE: Autores

Tomando en cuenta los niveles de temperatura adecuados que debe poseer un galpón de aves (**ANEXO 4**), se determina los siguientes rangos que deben cumplir los sensores:

- La temperatura debe ser mayor a 20°C y menor a 40°C.
- La intensidad de luz debe estar en un rango de 90 a 300 Lux.
- La Humedad a niveles de 50 a 70%.
- Los niveles de Gas entre 500 y 1000 PPM.

En la **Figura N°53**, se ve la programación para el envío de mensajes de alerta desde el sensor de temperatura, además tienen un mensaje de aviso cuando se encuentre dentro del rango ya establecido, hacer la misma estructura para los demás sensores tomando en cuenta los rangos antes expuestos.

```

//////////TEMPERATURA
if((tempe>=20)&&(tempe<=40)){ //Rango de comparacion de temperatura
  PORTB.F2=0;
  if(aux1==0){
    SMSenvia(); //subrutina de envio de SMS
    Send_to_GSM("Temperatura dentro de rango!!\r\n");
    Send_to_GSM1(temperatura); //empaquetado de valor SMS
    Delay_ms(500);
    UART1_Write(0x1A); // Cntrl+Z (0x1A ASCII valor de Cntrl+Z en Hex)
    UART1_Write(0x0D); // Terminacion de CR (ASCII valor Enter en Hex)
    Delay_ms(200);
    aux1=1;
  }
}
else{
  PORTB.F2=1;
  if(aux1==1){
    SMSenvia(); //subrutina de envio de SMS
    Send_to_GSM("Alerta - Temperatura Fuera de rango!!\r\n");
    Send_to_GSM1(temperatura); //empaquetado de valor SMS
    Delay_ms(500);
    UART1_Write(0x1A); //Cntrl+Z (0x1A ASCII valor de Cntrl+Z en Hex)
    UART1_Write(0x0D); // Terminacion de CR (ASCII valor Enter en Hex)
    Delay_ms(200);
    aux1=0;
  }
}
}

```

FIGURA N° 53.-Programación de mensajes de alerta enviada por un sensor  
FUENTE: Autores

#### 2.4.2.5.2. Envío de mensajes de textos

En esta parte se indica la estructura y comandos AT que posibiliten el envío de mensajes de texto desde el sistema. **Figura N°54.**

```
void SMSenvia(){
    PORTB.F1=1;           // Indicador de envío de comandos AT
    UART1_Write_Text("ATZ\r\n");//Enviar cadena "ATZ"
    Delay_ms(750);       // pequeño retraso 5000milli
    UART1_Write_Text("ATZ\r\n");//Enviar cadena "ATZ"
    Delay_ms(750);       // pequeño retraso 500milli
    ///Desactivar eco de comandos
    UART1_Write_Text("ATE\r\n");// Enviar "ATE" para desactivar el modo de eco
    Delay_ms(750);       // pequeño retraso 500milli
    UART1_Write_Text("AT+CMGF=1\r\n");//Cadena para establecer el modo de SMS
    Delay_ms(500);       // pequeño retraso 500milli
    /// Enviar mensaje
    /// Enviar mensaje al número celular: 0123456789
    UART1_Write_Text("AT+CMGS=\"0123456789\r\n");//(Escriba su número de
    //teléfono celular en lugar de 0123456789)

    Delay_ms(500);
    PORTB.F1=0;         //Indicador de fin de comandos AT
    return;
}
```

FIGURA N° 54.-Programa de envío de mensajes de texto  
FUENTE: Autores

#### 2.4.2.5.3. Interrupción PIR1 para lectura de mensajes de texto

Para el proceso de lectura de mensajes de texto que llegan al sistema, es sustancial establecer una rutina que identifique la cadena de caracteres del SMS y filtrar los datos de interés. Para guardar los datos se implementara una lectura de datos por medio de interrupciones, ver **Figura N°55**, que este en capacidad de recibir y almacenar los caracteres en un búfer de datos.

Una vez almacenados los datos en el búfer correspondiente empieza a buscar los datos que se necesitan para empezar la acción encomendada, el carácter que indica el inicio del mensaje es el asterisco (\*); ver **Figura N°56.**

```

//// Variables para recibir SMS
char Dato;
char Dato2;
char Bufer[255];
char Bufer2[]="U ";
unsigned short Pos=0;
unsigned short Bandera=0;
int i, j;
//// Declaracion de la funcion de interrupcion
void interrupt(){
//// Se evalua si la interrupción es disparada por recepcion serial
if(PIR1.F5){
//// Se lee el Dato de entrada
Dato=UART1_Read();
switch(Dato){
case 13: Pos=0; // Se recibe el caracter enter
Bandera=1;
break;
case 10: break; // Se recibe el retroceso del carro
default: Bufer[Pos++]=Dato; //se guarda los datos
Bufer[Pos++]=0; // Se establece el fin de cadena
break;
}
}
}
////////////////////////////////////

```

FIGURA N° 55.-Programa de interrupción PIR1 para lectura de datos  
FUENTE: Autores

```

void main()
{
//// SE CONFIGURAN LOS PUERTOS
//// se activa la interrupción por recepcion serial
PIE1=0b00100000;
//// Se desactivan las demas fuentes de interrupcion
PIE2=0;
//// Se apagan las banderas de interrupcion
PIR2=0;
PIR1=0;
//// Configuracion del puerto serial a 9600 bps
UART1_Init(9600); // Inicio de modulo Usart
//// Se activan las interrupciones globales y perifericas
INTCON=0b11000000;
while(1)
{
//// LEE SMS
//// Se evalua si una trama ha llegado
if(Bandera){
//// Aqui se debe analizar el contenido de la informacion del Bufer
for (i=0;i<=255;i++){
if(Bufer[i]=='*'){
j=i+2;
Dato2=Bufer[j];
UART1_Write(Dato2);
i=255;
}
}
Bandera=0; //se apaga la Bandera de llegada
}
//// Aquí preguntar por datos que contine el buffer
} ////////////////////////////////////

```

FIGURA N° 56.-Programa de búsqueda de carácter de inicio de mensaje  
FUENTE: Autores

#### 2.4.2.5.4. Lectura de mensaje para activación de actuadores y consulta de magnitudes de los sensores.

La programación del microcontrolador encargada del control de los actuadores y la consulta de magnitudes de los sensores, compara el carácter leído por la interrupción PIR1 con otro carácter para determinar la acción que debe realizar, el diagrama de flujo de la **Figura N°57** establece la lógica en caminada; determinar previamente los caracteres para cada acción:

- \*Q – Activación de actuador 1, y \*R – Desactivación de actuador 1.
- \*S – Activación de actuador 2, y \*T – Desactivación de actuador 2.
- \*U – Activación de actuador 3, y \*V – Desactivación de actuador 3.
- \*W – Activación de actuador 4, y \*X – Desactivación de actuador 4.
- \*Z – Consulta de magnitudes de sensores.

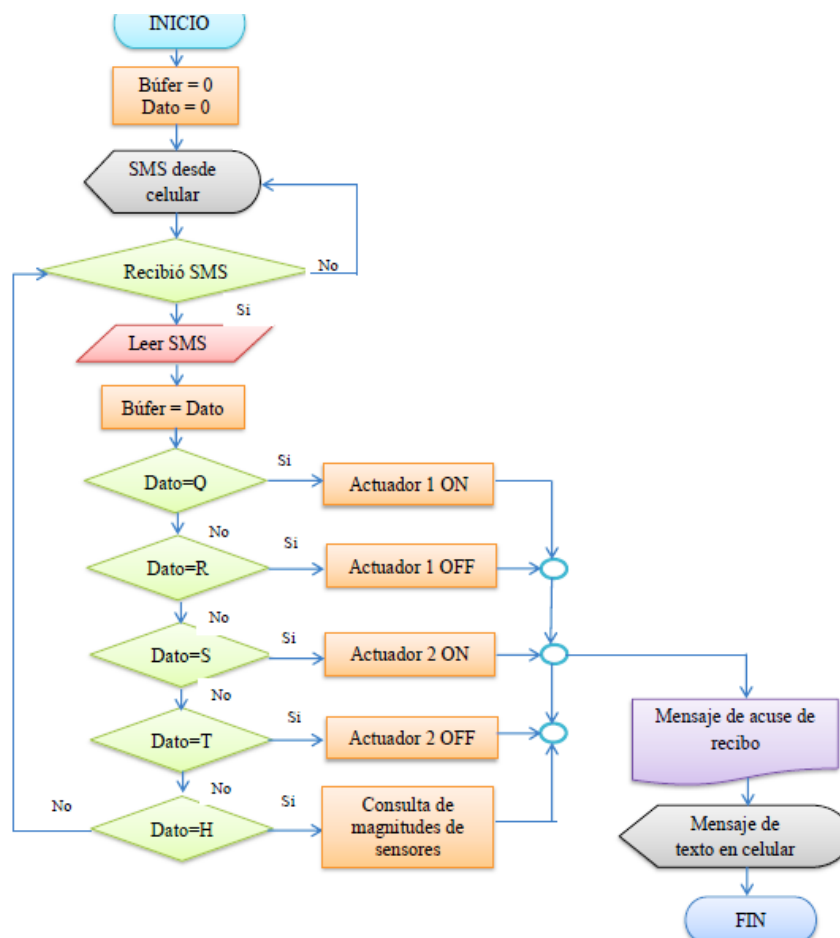


FIGURA N° 57.-Flujograma de programación del microcontrolador para activación de actuadores  
FUENTE: Autores

Parte del programa se ve en la **Figura N°58** este controla el actuador 1, usar la misma lógica de programación para los demás actuadores.

```

//// Aquí preguntar por datos que contine el buffer
//// LEE DATO2 PARA ACTUADORES
//Primer actuador
if(Dato2=='A'){ //Si mensaje contine *A
    PORTB.F5=1; //Actuador 1 ON
    SMSenvia(); // subrutina de envio de SMS
    Send_to_GSM("ACTIVO ACTUADOR 1!!!\r\n");
    Delay_ms(200);
    UART1_Write(0x1A); // Cntrl+Z (0x1A ASCII valor de Cntrl+Z en Hex)
    UART1_Write(0x0D); // Terminacion de CR (ASCII valor Enter en Hex)
    Delay_ms(500);
    Dato2=Bufere2;
}
if(Dato2=='B'){ //Si mensaje contine *B
    PORTB.F5=0; //Actuador 1 OFF
    SMSenvia(); // subrutina de envio de SMS
    Send_to_GSM("DESACTIVO ACTUADOR 1!!!\r\n");
    Delay_ms(200);
    UART1_Write(0x1A); // Cntrl+Z (0x1A ASCII valor de Cntrl+Z en Hex)
    UART1_Write(0x0D); // Terminacion de CR (ASCII valor Enter en Hex)
    Delay_ms(500);
    Dato2=Bufere2;
}

```

FIGURA N° 58.-Programa para el control de actuador 1  
FUENTE: Autores

A continuación se ve el programa de consulta de magnitudes de los sensores.  
**Figura N°59.**

```

////CONSULTA DE SENSORES
if(Dato2=='Z'){
    SMSenvia(); // subrutina de envio de SMS
    Send_to_GSM("CONSULTA DE SENSORES!!!\r\n");
    Send_to_GSM1(temperatura);
    Send_to_GSM2(luz);
    Send_to_GSM3(gas);
    Send_to_GSM4(humed);
    Delay_ms(200);
    UART1_Write(0x1A); // Cntrl+Z (0x1A ASCII valor de Cntrl+Z en Hex)
    UART1_Write(0x0D); // Terminacion de CR (ASCII valor Enter en Hex)
    Delay_ms(1000);
    Dato2=Bufere2;
}

```

FIGURA N° 59.-Programa para consulta de magnitudes de sensores  
FUENTE: Autores

Para mejor entendimiento el programa global se encuentra en el **ANEXO 5.**

#### 2.4.2.6.DISEÑO DE LA APLICACIÓN EN ANDROID

Para mayor facilidad del control y monitoreo de las condiciones climáticas, se creó una interfaz que permita al usuario manipular el sistema desde el teléfono móvil, para lo cual se utilizó App Inventor una plataforma basada en un servicio web de Google que permite almacenar los proyectos, maneja librerías Open Blocks de Java para el desarrollo visual de las aplicaciones.

##### 2.4.2.6.1. Configuración del entorno de programación Android

Para poder desarrollar la aplicación tomar en cuenta la *Práctica de laboratorio N°1.5: Diseño de una aplicación en sistema Android, para el control de una interfaz gráfica mediante mensajes de texto*; del **Anexo 3**. En la **figura N°60**, se observa la topología de los elementos que intervienen en la práctica.

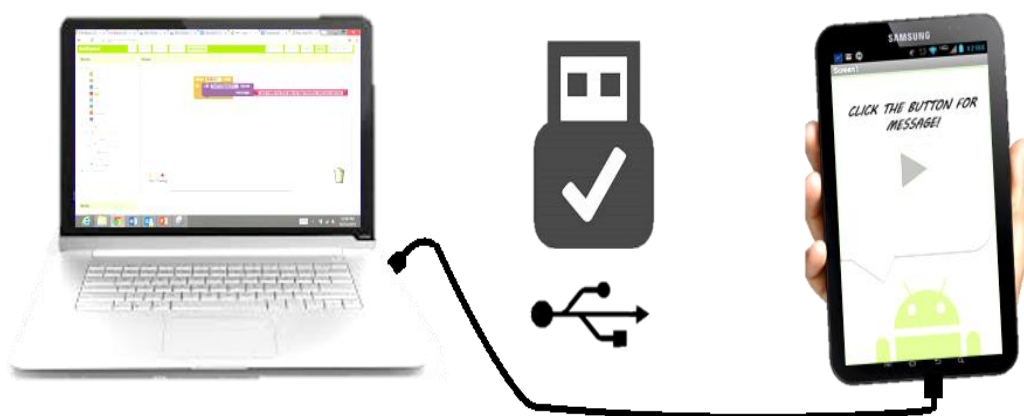


FIGURA N° 60.-Conexión de PC - teléfono con sistema Android

FUENTE: <https://sites.google.com/site/appinventormegusta/instalacion/conexion-a-un-telefono-o-tableta-con-un-cable-usb>

##### 2.4.2.6.2. Programación de la aplicación en Android

La aplicación consta de botones que controlan las funciones de encendido o apagado de los actuadores, así también para solicitar un registro de los valores de medición de los sensores y esto se visualiza en la pantalla. La programación es la misma que la práctica de laboratorio N°1.5 del **Anexo 3**, con la diferencia que cuenta con botones y luces indicadoras para cada actuador, sigue la lógica de programación del diagrama de la **Figura N°61**.



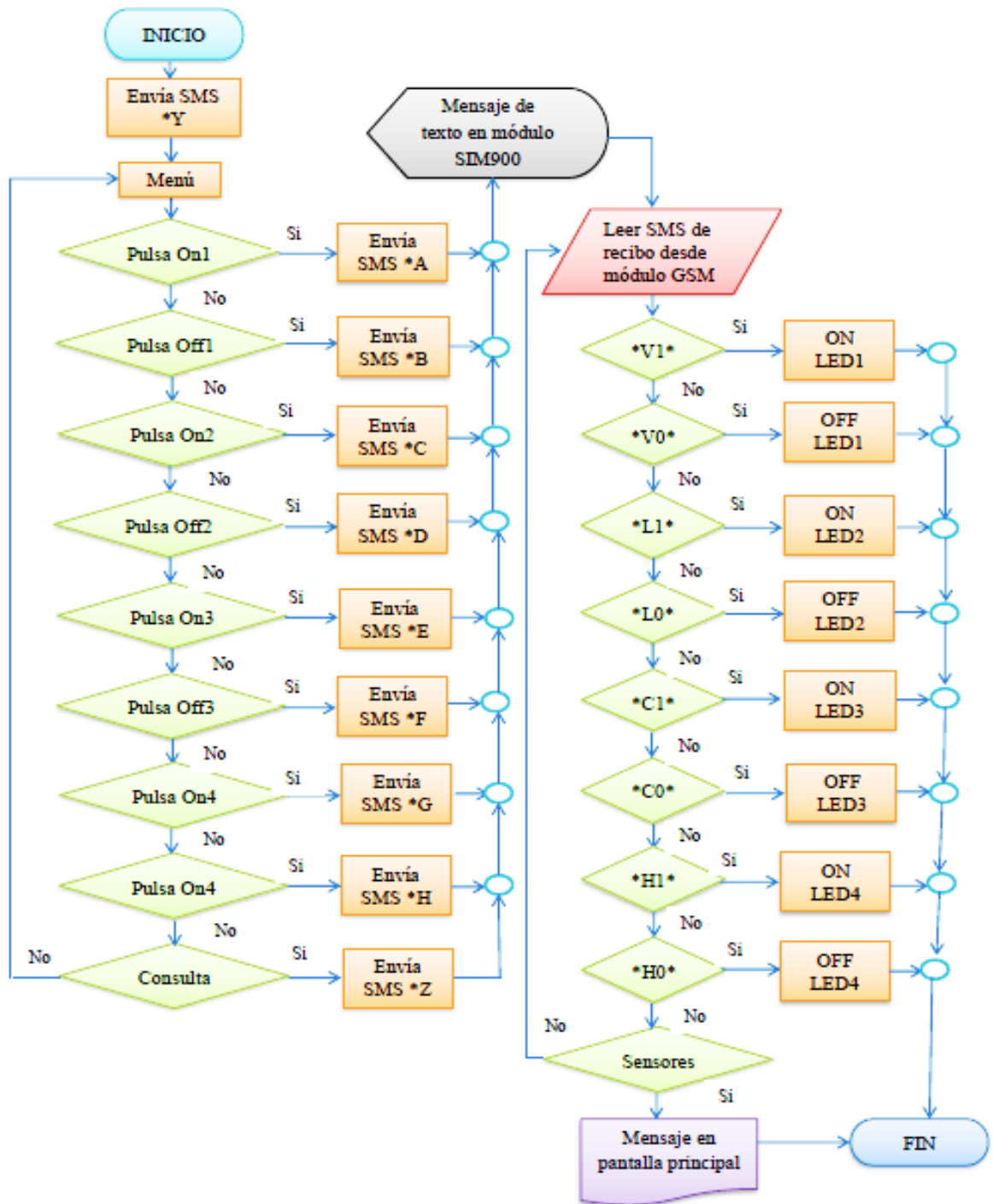


FIGURA N° 61.-Lógica de programación para la aplicación Android  
FUENTE: Autores

- Cuando la aplicación inicia solicita el nombre del usuario y clave para verificar y evitar la manipulación de otras personas (ver **Figura N°62**).
- Ingresar y envía un mensaje con contenido \*Y (ver **Figura N°63**), para consultar las magnitudes de los sensores y los actuadores que se encuentran activados o desactivados.



FIGURA N° 62.-Pantalla de inicio de la aplicación en teléfono  
FUENTE: Autores

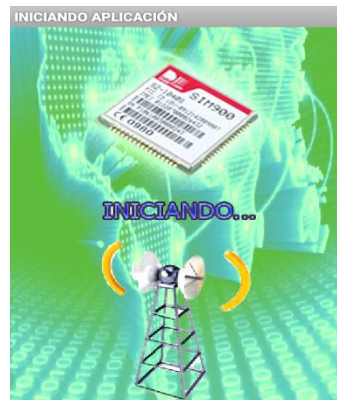


FIGURA N° 63.-Arranque de la aplicación  
FUENTE: Autores

- Al recibir el mensaje de acuse de recibo desde el módulo SIM900, se encienden o apagan los LED's indicadores de cada actuador y muestra las magnitudes medidas por los sensores (ver **Figura N°64**)



FIGURA N° 64.-Pantalla principal de la aplicación  
FUENTE: Autores

## 2.5.PROCESAMIENTO Y ANÁLISIS

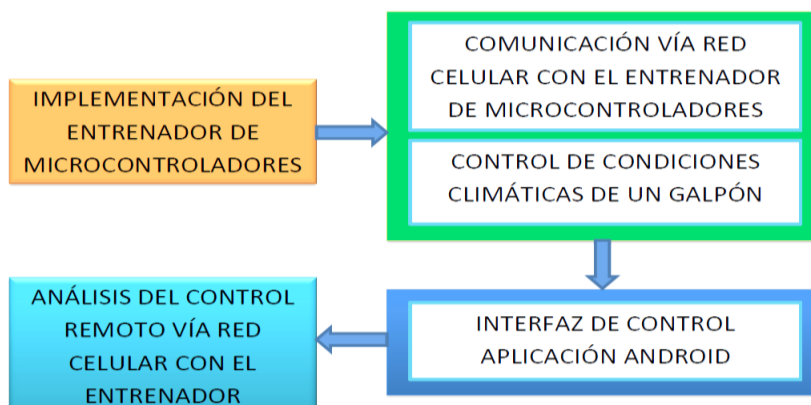


FIGURA N° 65.-Diagrama de procedimientos  
FUENTE: Autores

En la **Figura N°65**, se observa el proceso para el desarrollo de la investigación, con las etapas y herramientas utilizadas en cada etapa del proyecto mencionado.

### 2.5.1. ESTUDIO DEL ENTRENADOR DE MICROCONTROLADORES PARA COMUNICACIONES VÍA RED CELULAR

El entrenador se comunica con un teléfono móvil, a través de la red celular GSM, encargada de enviar los mensajes de texto de un dispositivo al otro. Para realizar la comunicación el entrenador como el teléfono deben tener acceso a una red telefónica por medio de su tarjeta SIM, ya que los datos enviados por la aplicación se deben recibir por el entrenador para cumplir la acción indicada, y así también el entrenador enviar mensajes de información al usuario sin importar el lugar que se encuentre. Como se muestra en la **Figura N°66** y **Figura N°67**, los bloques de procesos para que el control del sistema funcione correctamente.



FIGURA N° 66.-Etapas para el control de actuadores  
FUENTE: Autores

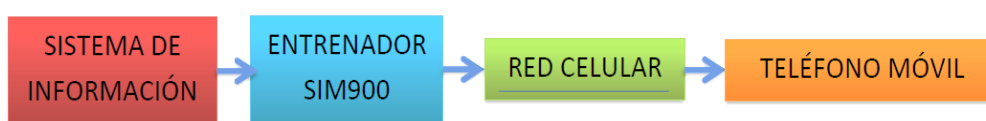


FIGURA N° 67.-Etapas para la información desde el entrenador  
FUENTE: Autores

### 2.5.2. ANÁLISIS DE LA COMUNICACIÓN DEL SISTEMA

El módulo SIM900 cumple con estándares de red celular GSM/GPRS, la correcta configuración por medio de comandos AT garantiza la comunicación entre dispositivos de un sistema y el teléfono móvil.

El uso de una tarjeta SIM para la comunicación es un requerimiento importante, ya que permite el acceso a la red celular por medio de diferentes estaciones base, para así lograr el control remoto desde diferentes lugares cuando el usuario lo requiera, se necesita definir el número de teléfono del módulo para que el usuario realice monitorios desde diferentes teléfonos.

### 2.5.3. ADMINISTRACIÓN DE LA INTERFAZ DE CONTROL

El control remoto de la temperatura de una avícola vía red celular utiliza mensajes de texto, que se emiten desde el teléfono móvil hacia el módulo SIM900 y viceversa, la interfaz permite interactuar de manera fácil al usuario con botones e indicadores de los actuadores. (**Figura 68**)



FIGURA N° 68.-Control por mensajes de texto  
FUENTE: Autores

### 2.5.4. ANÁLISIS DEL SISTEMA DE CONTROL REMOTO DE LA TEMPERATURA DE UNA AVÍCOLA VÍA RED CELULAR

Los mensajes de texto enviados por el teléfono a través de la red celular, hacia el sistema de control remoto que tiene implementado un módulo SIM900 para recibir los mensajes de texto, siempre que tenga una tarjeta SIM incorporada.

El módulo GSM debe tener una comunicación UART con el microcontrolador y así controlar los actuadores conectados al mismo y monitorear las condiciones ambientales por medio de los sensores; en la programación del microcontrolador se debe indicar el número de teléfono del celular que va enviar los mensajes de texto de acuse de recibo. Ver la **Figura N° 69**.

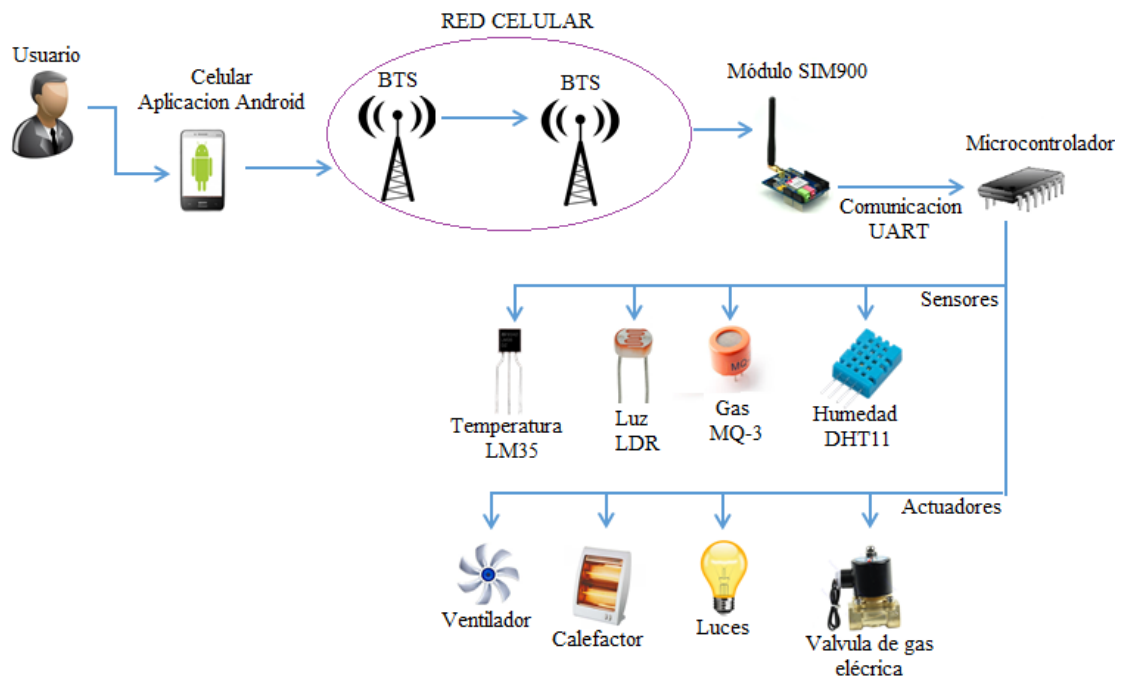


FIGURA N° 69.-Control de la temperatura de una avícola vía red celular  
FUENTE: Autores

## 2.6.COMPROBACIÓN DE LA HIPÓTESIS

Las pruebas realizadas siguen una sinopsis de afirmaciones cercanas de una población, para afirmar o negar la hipótesis.

### 2.6.1. PLANTEAMIENTO DE LA HIPÓTESIS:

#### Hipótesis Nula Ho:

Con el diseño e implementación de un entrenador de microcontroladores para comunicaciones, usando la red celular se logrará el monitoreo de sensores para el desarrollo de las prácticas.

#### Hipótesis Alternativa H1:

Con el diseño e implementación de un entrenador de microcontroladores para comunicaciones, usando la red celular no se logrará el monitoreo de sensores para el desarrollo de las prácticas.

### 2.6.2. ESTABLECIMIENTO DE NIVEL DE SIGNIFICANCIA

Para alcanzar un resultado confiable para la investigación, se toma un valor de significancia en relación al 95% de confianza equivalente a 1.96, siendo un valor a criterio del investigador, que permitirá el resultado sea aceptable.

### 2.6.3. MUESTRA ENCONTRADA

Se toma de la población una muestra 17 pruebas en el armado de las prácticas: *“Uso de teclado de matriz 4x4 y visualización en el LCD de caracteres”* y *“Uso de módulo-teclado y visualización en el módulo-LCD del entrenador de micorontroladores”*, que se indica en el **Anexo 6**; haciendo referencia al tiempo de demora para completar la práctica, sin y con el entrenador para así probar su eficiencia. **Tabla N°6.**

Tabla N° 6.-Datos de tiempo de armado de práctica  
FUENTE: Autores

N° prueba de armado	TIEMPO DE ARMADO DE LA PRACTICA PROPUESTA EN MINUTOS	
	Sin entrenador	Con entrenador
Prueba 1	21	7
Prueba 2	23	9
Prueba 3	22	8
Prueba 4	23	8
Prueba 5	21	7
Prueba 6	22	8
Prueba 7	22	7
Prueba 8	25	10
Prueba 9	23	8
Prueba 10	22	8
Prueba 11	22	7
Prueba 12	24	7
Prueba 13	22	8

<b>Prueba 14</b>	21	7
<b>Prueba 15</b>	24	8
<b>Prueba 16</b>	24	8
<b>Prueba 17</b>	22	7
<b>Sumatoria</b>	383	132

En el caso del cálculo de la muestra en una población desconocida se realiza un total de 35 pruebas de comunicación entre el teléfono celular y el módulo GSM/GPRS, en base al tiempo de recepción (SMS) y distancia de envío del mensaje de texto como se indica en la **Tabla N°7**.

Tabla N° 7.-Datos medidos en tiempo de Latencia  
FUENTE: Autores

N° prueba	TIEMPO DE RESPUESTA DEL MÓDULO SIM900 EN SEGUNDOS						
	DISTANCIA						
	Teléfono móvil – Módulo SIM900			Módulo SIM900 – Teléfono móvil			
	10m	2Km	4Km	10m	2Km	4Km	SUMA
Prueba 1	5	8	6	8	7	7	41
Prueba 2	5	8	7	7	7	7	41
Prueba 3	5	8	6	8	7	7	41
Prueba 4	5	7	7	8	7	7	41
Prueba 5	5	7	7	8	8	7	42
Prueba 6	6	7	7	8	7	7	42
Prueba 7	5	7	7	8	8	8	43
Prueba 8	5	7	6	8	7	7	40
Prueba 9	5	8	7	7	7	7	41
Prueba 10	6	8	6	8	7	7	42
Prueba 11	5	7	7	8	7	7	41
Prueba 12	5	8	7	8	8	7	43
Prueba 13	6	7	7	8	7	7	42
Prueba 14	5	8	7	8	8	8	44
Prueba 15	5	8	7	7	7	7	41
Prueba 16	6	8	6	8	7	7	42
Prueba 17	5	7	7	8	7	7	41
Prueba 18	5	8	7	8	8	7	43
Prueba 19	6	7	7	8	7	7	42
Prueba 20	5	8	7	8	8	8	44
Prueba 21	5	7	6	8	7	7	40
Prueba 22	5	7	6	8	7	7	40

Prueba 23	5	7	7	9	8	7	43
Prueba 24	6	8	7	8	7	7	43
Prueba 25	7	8	7	8	6	7	43
Prueba 26	5	8	7	8	7	8	43
Prueba 27	5	8	7	7	7	7	41
Prueba 28	5	7	8	8	7	7	42
Prueba 29	6	7	8	8	8	7	44
Prueba 30	5	7	6	7	7	8	40
Prueba 31	5	8	7	7	7	7	41
Prueba 32	5	7	7	8	8	7	42
Prueba 33	5	8	7	7	8	7	42
Prueba 34	5	8	7	7	8	7	42
Prueba 35	5	8	7	8	8	7	43
Sumatoria	184	264	239	273	256	250	1466
Promedio General	5,26	7,54	6,82	7,80	7,31	7,14	41,89

#### 2.6.4. CÁLCULO DE LA MEDIA Y LA VARIANZA

De forma estadística se realiza el cálculo de la media de las pruebas realizadas, usando las siguientes formulas:

$$\bar{X} = \frac{\sum_{i=1}^n X_i}{n}$$

Dónde:

- $\bar{X}$  o  $\bar{Y}$ : Media Aritmética.
- $n$  : Número de muestra.
- $X_i$ : Valor de prueba

En la **Tabla N°8** se muestra los valores del cálculo de la media y desviación, para los datos de armado de la práctica.

Tabla N° 8.-Cálculo de la media y desviación para los datos de armado de la práctica  
FUENTE: Autores

N° de prueba	Media $\bar{X}$	Desviación $(X_i - \bar{X})^2$	Media $\bar{Y}$	Desviación $(Y_i - \bar{Y})^2$
Prueba 1	22,53	2,3391	7,76	0,5848



<b>Prueba 2</b>	22,53	0,2215	7,76	1,5260
<b>Prueba 3</b>	22,53	0,2803	7,76	0,0554
<b>Prueba 4</b>	22,53	0,2215	7,76	0,0554
<b>Prueba 5</b>	22,53	2,3391	7,76	0,5848
<b>Prueba 6</b>	22,53	0,2803	7,76	0,0554
<b>Prueba 7</b>	22,53	0,2803	7,76	0,5848
<b>Prueba 8</b>	22,53	6,1038	7,76	4,9965
<b>Prueba 9</b>	22,53	0,2215	7,76	0,0554
<b>Prueba 10</b>	22,53	0,2803	7,76	0,0554
<b>Prueba 11</b>	22,53	0,2803	7,76	0,5848
<b>2Prueba 12</b>	22,53	2,1626	7,76	0,5848
<b>Prueba 13</b>	22,53	0,2803	7,76	0,0554
<b>Prueba 14</b>	22,53	2,3391	7,76	0,5848
<b>Prueba 15</b>	22,53	2,1626	7,76	0,0554
<b>Prueba 16</b>	22,53	2,1626	7,76	0,0554
<b>Prueba 17</b>	22,53	0,2803	7,76	0,5848
<b>Sumatoria</b>		22,2353		11,0588

Para el cálculo de la varianza se utilizó la siguiente formula:

$$S^2 = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n - 1}$$

Donde los parámetros resultan:

- $S^2$ : Varianza.
- S: Desviación estándar.

Dando como resultado los siguientes valores, **Tabla N°9**:

Tabla N° 9.-Valores de varianza y desviación estándar del tiempo de armado de práctica  
FUENTE: Autores

	<b>Sin entrenador</b>	<b>Con entrenador</b>
<b>Varianza</b>	1,3897	0,6912
<b>Desviación estándar</b>	1,1789	0,8314

Para las pruebas de latencia de comunicación entre teléfono celular módulo SIM900, se realiza el cálculo de la media y la varianza en cada una de las pruebas realizadas con las formulas estadísticas. **Tabla N°10**.

Formula de promedio:

$$\bar{X} = \frac{\sum_{i=1}^n X_i}{n}$$

Ecuación de Varianza total:

$$\sigma = \frac{\sum_{i=1}^n (V_i - \bar{X})^2}{n - 1}$$

Tabla N° 10.-Calculo de la media y varianza de latencias  
FUENTE: Autores

<b>N° de prueba</b>	<b>Valor</b>	<b>Media</b>	<b>Varianza</b>
Prueba 1	41	41,91	0,83
Prueba 2	41	41,91	0,83
Prueba 3	41	41,91	0,83
Prueba 4	41	41,91	0,83
Prueba 5	42	41,91	0,01
Prueba 6	42	41,91	0,01
Prueba 7	43	41,91	1,19
Prueba 8	40	41,91	3,65
Prueba 9	41	41,91	0,83
Prueba 10	42	41,91	0,01
Prueba 11	41	41,91	0,83
Prueba 12	43	41,91	1,19
Prueba 13	42	41,91	0,01
Prueba 14	44	41,91	4,37
Prueba 15	41	41,91	0,83
Prueba 16	42	41,91	0,01
Prueba 17	41	41,91	0,83
Prueba 18	43	41,91	1,19
Prueba 19	42	41,91	0,01
Prueba 20	44	41,91	4,37
Prueba 21	40	41,91	3,65
Prueba 22	40	41,91	3,65
Prueba 23	43	41,91	1,19
Prueba 24	43	41,91	1,19
Prueba 25	43	41,91	1,19
Prueba 26	43	41,91	1,19
Prueba 27	41	41,91	0,83

<b>Prueba 28</b>	42	41,91	0,01
<b>Prueba 29</b>	44	41,91	4,37
<b>Prueba 30</b>	40	41,91	3,65
<b>Prueba 31</b>	41	41,91	0,83
<b>Prueba 32</b>	42	41,91	0,01
<b>Prueba 33</b>	42	41,91	0,01
<b>Prueba 34</b>	42	41,91	0,01
<b>Prueba 35</b>	43	41,91	1,19

Realizando los cálculos correspondientes se obtiene una varianza total de = 1,27

### 2.6.5. ANÁLISIS DE LAS REGIONES CRÍTICAS

En primera instancia se trata de un problema de inferencia estadística, acerca de la diferencia de medias de tiempo entre el rendimiento de armado de prácticas sin el entrenador y con el mismo.

Considerando las varianzas conocidas se tiene que:

- $\mu_1$  = Rendimiento de armado de práctica sin entrenador.
- $\mu_2$  = Rendimiento de armado de práctica con entrenador.

$$H_0 = \mu_1 < \mu_2$$

$$H_1 = \mu_1 \geq \mu_2$$

Como la hipótesis alterna indica  $\mu_1$  es mayor o igual que  $\mu_2$  entonces la región crítica es una cola hacia la derecha, con un valor de significancia de:  $\alpha=0,05$ ; ver **Figura N°70**. Y la regla de decisión se encuentra dada la **Tabla N°11**.

Tabla N° 11.-Regla de decisión para entrenador  
FUENTE: Autores

Se rechaza $H_0$ si:	$Z_c > 1,6449$
Se acepta $H_0$ si:	$Z_c \leq 1,6449$

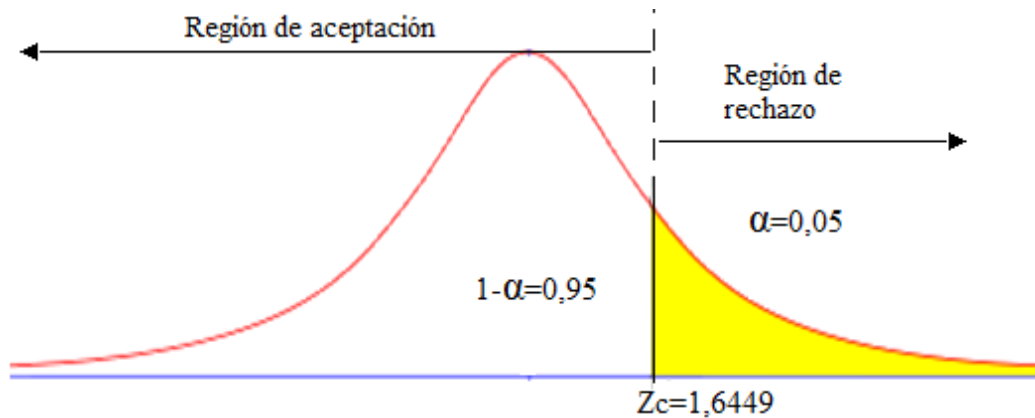


FIGURA N° 70.-Valor crítico lateral derecho  
FUENTE: Autores

Para probar la eficiencia de los módulos SIM900 con la pruebas de latencia de comunicación, revisar los valores de la **Tabla N°12**, con los valores críticos se tiene con un valor de significancia  $\alpha=0,05$  en este caso. Considerando que los valores de latencia, carece de registros identificables, porque el número de pruebas puede ser infinito se tiene que:

$$H_0 = \mu_t \neq \mu_{Real}$$

$$H_1 = \mu_1 \neq \mu_{Real}$$

Tabla N° 12.-Datos que determinan los valores críticos  
FUENTE: Autores

<b>1- α</b>	<b>α/2</b>	<b>Z α/2</b>
0,90	0,05	1,645
0,95	0,025	1,96
0,99	0,005	2,575

La **Figura N°71**, muestra la representación de estos valores, es bilateral porque toman valores positivos y negativos en la recta.

En la **Figura N°72** se muestran los límites de valores críticos para rechazar la hipótesis según el valor de significancia elegido.

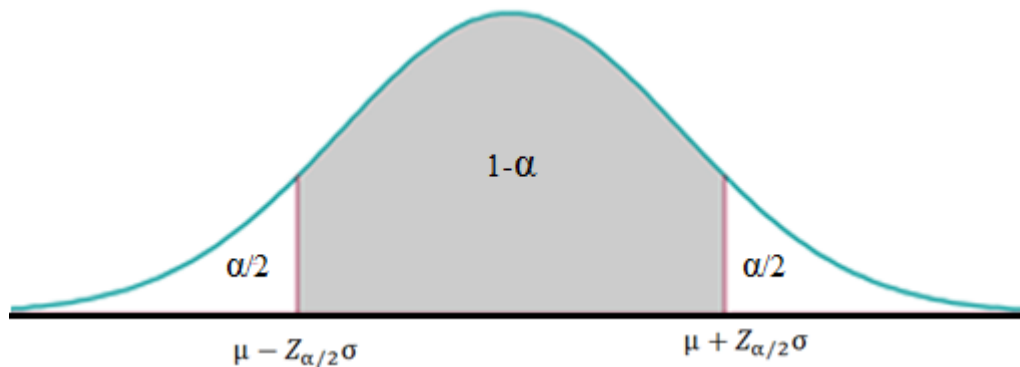


FIGURA N° 71.-Valores críticos bilaterales  
FUENTE: Autores

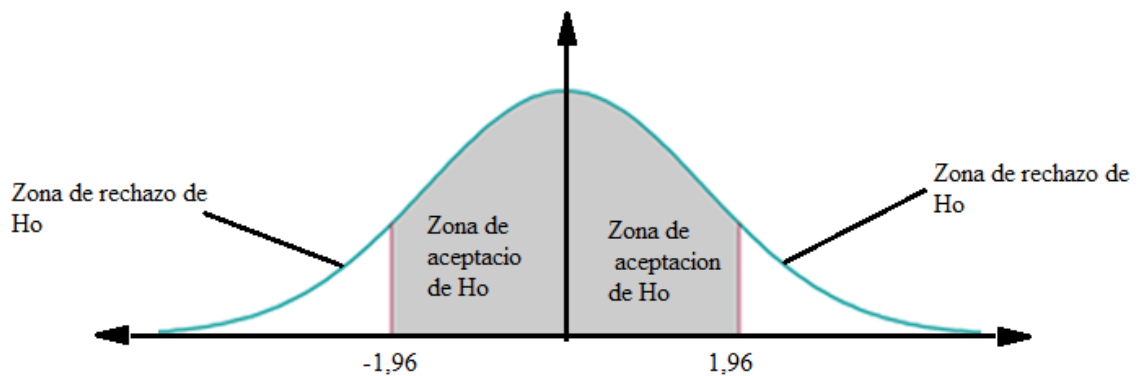


FIGURA N° 72.-Representación de límites de valores críticos  
FUENTE: Autores

### 2.6.6. CÁLCULO DEL VALOR CRITICO O ZC

Con la información de datos obtenida por los tiempos de demora en armados de prácticas y bajo el supuesto de la hipótesis nula es verdadera, la estadística de prueba es:

$$Z_c = \frac{\bar{X} - \bar{Y}}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

Con las siguientes variables:

- Medias de muestras  $\bar{X}$ ,  $\bar{Y}$ .

- Variación estándar =  $\sigma$ .
- Número de muestras =  $n$ .

Remplazando por los valores obtenidos:

$$Z_c = \frac{22,2353 - 11,0588}{\sqrt{\frac{1,3897}{17} + \frac{0,6912}{17}}}$$

$$Z_c = \frac{11,18}{0,3499}$$

**Valor crítico o  $Z_c = 31,9452$**

Con la recolección de datos de latencia obtenidos y analizados estadísticamente se procede a calcular el valor crítico, con la siguiente formula:

$$Z_c = \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{n}}}$$

Donde sus variables son:

- $\bar{X}$ : Media de valores aleatorios.
- $\mu$ : Promedio de los valores óptimos, en este caso se tomara el dato del promedio de la prueba general.
- $\sigma$ : Variación estándar.
- $n$ : Numero de muestras

$$Z_c = \frac{41,91 - 41,89}{\frac{1,27}{\sqrt{35}}}$$

$$Z_c = \frac{0,02}{0,214}$$

$Z_c = 0,0931 = \text{Valor crítico}$

## 2.6.7. REPRESENTACIÓN GRÁFICA DE LA REGLA DE DECISIÓN

El valor crítico obtenido se encuentra fuera de la región de aceptación de la hipótesis nula como se demuestra en la **Figura N°73**, tomando en cuenta el valor de confiabilidad de 95%, comprobando que la hipótesis nula es verificada de manera negativa, tomando como pruebas los tiempos de demora en armados de prácticas, se puede decir que el rendimiento de armado de práctica con entrenador es mucho mejor.

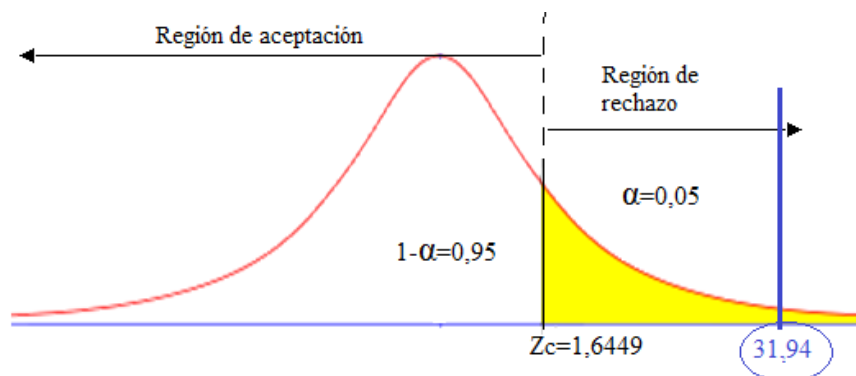


FIGURA N° 73.-Representación gráfica de decisión por armado de práctica  
FUENTE: Autores

Tomando en cuenta los tiempos de latencia de comunicación entre el teléfono y módulo SIM900 a distintas distancias con su promedio total y el promedio general del tiempo, su valor crítico se encuentra en la zona de aceptación de la hipótesis nula como se demuestra en la **Figura N°66**, tomando en cuenta que la confiabilidad en este caso es de un 95%; comprobando que la hipótesis nula es verificada de manera positiva. Por lo tanto el grado de eficiencia del módulo y su comunicación con el dispositivo móvil es alto.

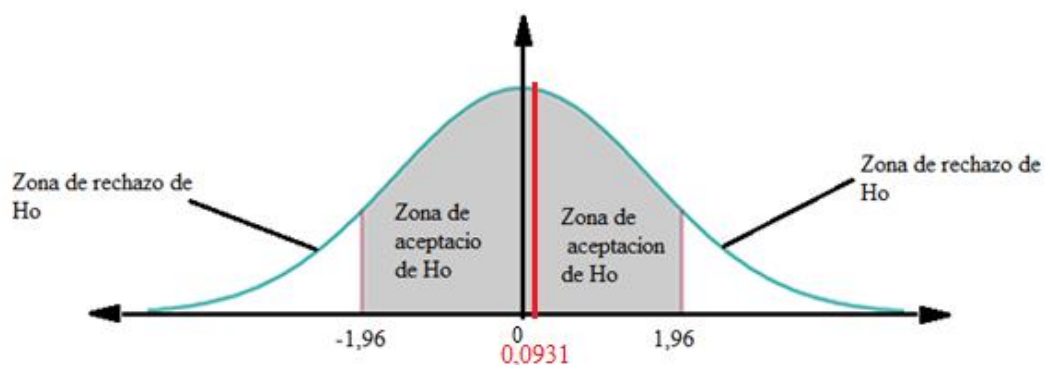


FIGURA N° 74.-Representación gráfica de decisión por latencia  
FUENTE: Autores

## CAPÍTULO III

### 3. RESULTADOS

#### 3.1. PRUEBAS DE TIEMPO EN EL ARMADO DE PRÁCTICAS CON EL ENTRENADOR

Una vez diseñadas las placas de los diferentes módulos se procedió con la estructuración del entrenador para realizar las pruebas de armado de la práctica propuesta, ver **Anexo 6**; esto se la realizo a 17 estudiantes de la Escuela de electrónica y Telecomunicaciones que cursan la asignatura de microcontroladores, obteniendo resultados satisfactorios ya que eficientemente se mejoró el proceso de armado de la práctica.

En la **Figura N°67** se muestra la práctica armada sin el entrenador y en la **Figura N°68** la misma práctica en el entrenador.

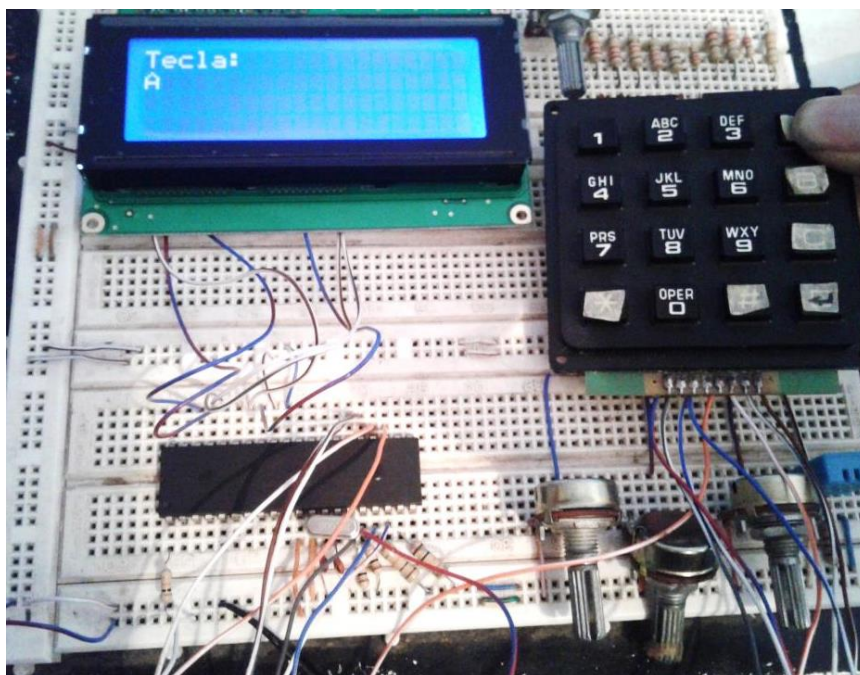


FIGURA N° 75.-Práctica propuesta sin entrenador  
FUENTE: Autores



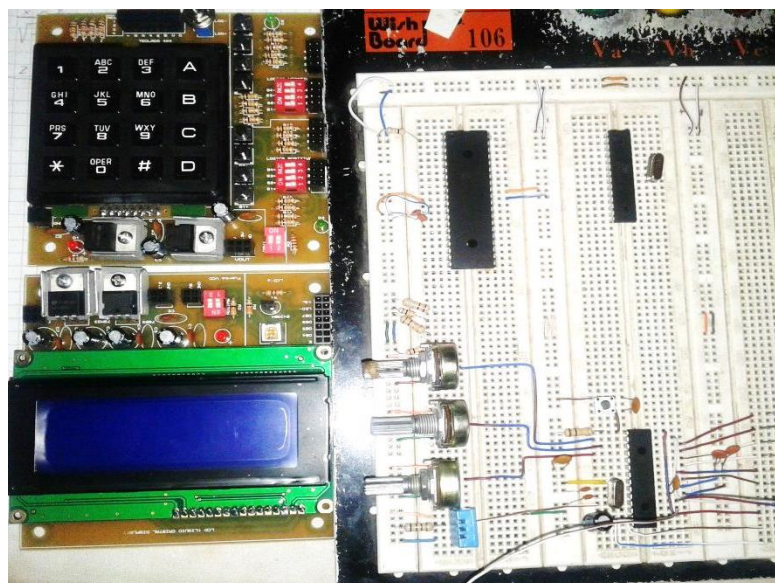


FIGURA N° 76.-Práctica propuesta con el entrenador  
FUENTE: Autores

Se nota la gran utilidad del entrenador para la facilidad del armado de las prácticas con los microcontroladores, permitiendo a los estudiantes preocuparse más en la programación del microcontrolador que en el armado de la práctica.

### 3.2. PRUEBAS DE LATENCIA DE COMUNICACIÓN POR RED CELULAR

Las pruebas de comunicación, entre el teléfono móvil y el módulo SIM900, se realizaron desde distintos lugares de la ciudad de Riobamba, teniendo el módulo en el domicilio que queda ubicado junto a la Hostería el Toril y el teléfono móvil en distintos lugares de la ciudad, se detalla en la **Tabla N°13**.

Tabla N° 13.-Pruebas de comunicación  
FUENTE: Autores

DIRECCIÓN	DISTANCIA	TIEMPO TELÉFONO - MÓDULO SIM900	TIEMPO MÓDULO SIM900 - TELÉFONO
Domicilio	25 m	5 seg	8,16 seg
UNACH	1,445Km	7,59 seg	9,72 seg
Parque Alfaro	2,688Km	7,66 seg	7,54 seg
Cdla. La Primavera	3,601Km	6,43 seg	6,57 seg
Mercado Mayorista	3,209Km	7,07 seg	7,03 seg

### 3.3.ANÁLISIS FINANCIERO

La **Tabla N°14** muestra una lista de materiales utilizados para la implantación de los entrenadores.

Tabla N° 14.-Materiales utilizados en el proyecto  
FUENTE: Autores

<b>CANT</b>	<b>DETALLE</b>
50	Condensadores
8	Dip swicht
2	Display -Matrices 8x5
4	Display´s 7 segmetos
2	Display´s Alfanuméricos
2	Fuetes de voltaje 9v-3A
2	Giroscopios
2	Integrado MAX232
2	Integrado MAX485
2	LCD´s 4x16
4	Led´s RGB
40	Leds de colores
2	Módulos bluethooth
2	Módulos ethernet
2	Módulos SIM900
2	Módulos wifi
2	Módulos xbee
4	Molex
4	Operacionales LM358
2	Opto acopladores ECG3100
8	Potenciómetros de precisión
2	Protoboards
8	Pulsadores
2	Puentes rectificadores
30	Regletas dobles
24	Reguladores 7805
24	Reguladores 7833
50	Resistencias
2	Sensores DHT11
2	Sensores LM35
2	Sensores MQ-3
2	Switch de 2 posiciones
2	Teclado 4x4
2	Transeiver SV651
2	Transformadores
4	Zócalos

1	Pic 16F877A
1	Pic 18F2550
1	Atmega328p
1	Cable USB a RS232
4	Optoacoplador 4n25
2	Cristal de 20mhz
1	Cristal de 16mhz
4	Relés
	Elementos de soldado
	Placas de prueba
	Acrílicos

El entrenador se lo desarrolló de forma individual, por esta razón el presupuesto alcanzo un costo de \$1 500 USD, por cada persona.

## CAPÍTULO IV

### 4. DISCUSIÓN

El desarrollo del entrenador de microcontroladores para comunicaciones, está fundamentado en la actualización de tecnología de medios de comunicación, a través de una exhaustiva recolección de información de los tipos que hoy en día presentan una alta tendencia de uso, se logró encontrar los módulos apropiados que permitan el desarrollar de prácticas y complementar los conocimientos en los temas de telecomunicaciones y microcontroladores.

Se puso énfasis en las comunicaciones vía red celular, el uso más frecuente de esta tecnología es la intercomunicación de los teléfonos móviles por voz, mensajes de texto o internet, siempre y cuando se encuentren dentro del área de cobertura de la operadora, la cobertura sigue extendiéndose permitiendo que los usuarios realicen comunicación desde cualquier lugar. Las operadoras ofrecen distintos servicios de comunicación por red celular, pero el que se utilizó como punto de estudio en el presente proyecto es el de mensajería corta (SMS), ya que es el más económico y accesible para cualquier usuario, no depende de activación de datos u otro servicio.

El sistema utilizado para el monitoreo y control remoto de la temperatura de una avícola vía red celular cumple con un esquema lógico, donde debe cumplir el envío de SMS para la activación de los actuadores y la lectura de los mensajes que llegan desde el módulo con la información de las mediciones de los sensores. El software que se utilizó para el desarrollo de la aplicación, está basado en los usuarios de los teléfonos con sistema Android, por la facilidad de manejo que brinda este sistema.

Así pues finalizado los equipos, son capaces de cumplir con todas las expectativas planteadas de forma eficiente.

## **CAPÍTULO V**

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

#### **5.1. CONCLUSIONES**

- El entrenador de microcontroladores para comunicaciones vía red celular, permite realizar una comunicación de forma eficiente.
- Podrá desarrollar sistemas embebidos para controlar, supervisar u operar equipos o maquinarias.
- El sistema de control de temperatura de una avícola vía red celular, en el entrenador facultó monitorear magnitudes de sensores y conmutar relés.
- El módulo SIM900 asiste al entrenador de microcontroladores, la conexión a la red de telefonía GSM/GPRS, para realizar comunicación con dispositivos móviles.
- Los servicios de mensajería corta que brindan las operadoras permiten que sea posible la comunicación remota desde un teléfono móvil hacia el sistema de monitoreo y control.
- La eficiencia de la comunicación remota, es dependiente de la velocidad de transmisión y la distancia a la que se encuentran los dispositivos móviles.

## 5.2.RECOMENDACIONES

- La configuración del módulo SIM900, es necesario hacerla de forma manual con los comando AT necesarios, por seguridad y para confirmar que las características estén acordes a la necesidad del usuario.
- Para que el módulo GSM trabaje de forma estable alimentar con el voltaje adecuado de 5Vcc y la corriente necesaria de 1.5mA en sleep mode.
- Disponer de saldo suficiente en la tarjeta SIM utilizada en el módulo, para realizar las distintas pruebas.
- Es necesario estar dentro del área de cobertura de la operadora para realizar la comunicación.
- La programación del microcontrolador debe poseer las conversiones adecuadas para las mediciones de los sensores y los comandos AT necesarias para la lectura y envío de mensajes de texto.
- Verificar que el teléfono móvil sea compatible con la aplicación, tomando en cuenta que debe contar con sistema Android versión 2.3 como mínimo.
- Para visualizar las acciones en la aplicación esperar el tiempo necesario para que el microcontrolador envíe un mensaje de respuesta de la acción realizada.
- Para interconectar un sistema digital con un sistema de potencia, es necesario hacer un optoacoplamiento, para garantizar el aislamiento eléctrico.

# **CAPÍTULO VI**

## **6. PROPUESTA**

### **6.1.TÍTULO DE LA PROPUESTA**

“DISEÑO E IMPLEMENTACIÓN DE UN ENTRENADOR DE MICROCONTROLADORES PARA COMUNICACIÓN VÍA RED CELULAR”

### **6.2.INTRODUCCIÓN**

Para actualizar las tecnologías de medios de comunicación, se necesita determinar los tipos de comunicación que en la actualidad se encuentran en auge, que permiten la trasmisión y recepción de información a corta y larga distancia, para el desarrollo de prácticas y complementar los conocimientos en los temas de telecomunicaciones y microcontroladores. Una vez determinados los medios más relevantes, se escogen los equipos adecuados que permitan el desarrollo de aplicaciones para estudios y desarrollo.

Para desarrollar los objetivos planteados, se deberá desarrollar en el entrenador un esquema de diseño aplicativo para lograr que el desempeño del equipo sea eficiente y cumpla con todas las expectativas: comunicación por medio de la red celular, monitoreo de las magnitudes medidas por los sensores y el control remoto utilizando un teléfono móvil para el mando de actuadores.

Con la revisión anticipada que se concebirá de los módulos de comunicación vía red celular, tomando en cuenta las ventajas y los precios, se podrá determinar el Shell que se debe utilizar para culminar el proyecto de manera confortable y lograr establecer comunicación por medio de la red celular.

## **6.3.OBJETIVOS**

### **6.3.1. OBJETIVO GENERAL**

Diseñar e implementar un entrenador de microcontroladores para realizar comunicaciones vía red celular.

### **6.3.2. OBJETIVO ESPECIFICO**

- Desarrollar una aplicación de comunicación vía red celular utilizando el entrenador de microcontroladores, para poder monitorear sensores y conmutar relés por un teléfono móvil.
- El entrenador servirá para el desarrollo de sistemas embebidos.
- Realizar un manual de prácticas para que el usuario compruebe los distintos tipos de prácticas que se pueden realizar en el entrenador.
- La arquitectura del entrenador quedará abierto para el desarrollo con otros tipos de microcontroladores.

## **6.4.DESCRIPCIÓN DEL PROBLEMA**

La propuesta se fundamenta en la necesidad y falta de equipos en los laboratorios para realizar estudios y mejorar las prácticas con microcontroladores para aplicaciones con las telecomunicaciones, las mismas que servirán para el desarrollo de aplicaciones útiles para monitorear sensores y controlar relés para la activación de actuadores por medio de la red celular, permitiendo que un teléfono móvil sirva como control remoto de todo un sistema.

El sistema deberá emitir hacia un teléfono celular mensajes de alerta, indicando que alguna magnitud medida por los sensores se encuentra fuera del rango establecido previamente, leer los mensajes que reciba y si este contiene alguna instrucción que indique el usuario desde su móvil, realizar la acción encomendada como: la activación de los actuadores y el monitoreo de las magnitudes medidas por los sensores.



## 6.5.DISEÑO ORGANIZACIONAL

En la **Figura N°69** se observa, el esquema organizacional de las personas que participan en el desarrollo y la implementación de la propuesta.

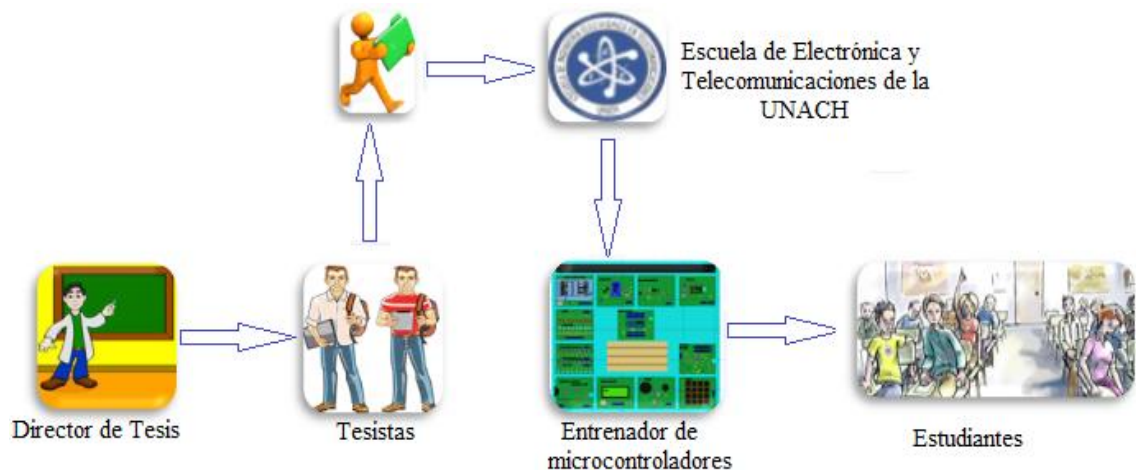


FIGURA N° 77.-Diagrama del sistema del proyecto  
FUENTE: Autores

## 6.6. MONITOREO Y EVALUACIÓN DE LA PROPUESTA

Para el monitoreo y evaluación de la propuesta, se realizara pruebas con los estudiantes con prácticas, tomando en cuenta los tiempos de demora y la facilidad que se tiene para desarrollar las prácticas con en el entrenador. También se ejecutara comunicaciones con el dispositivo de comunicación celular y el teléfono móvil, haciendo un seguimiento para el acceso a la red para el monitoreo y control de las condiciones climáticas de un área, determinando la confiabilidad de transmisión y recepción de los mensajes de texto desde diferentes distancias, para evaluar el sistema, además se realizará un manual de prácticas con el avance del desarrollo del proyecto, para que el usuario pueda realizar los trabajos y aplicaciones en el entrenador.

## CAPÍTULO VII

### 7. BIBLIOGRAFÍA

- CCM.net.* (Mayo de 2015). Obtenido de <http://es.ccm.net/contents/680-estandar-gprs-servicio-general-de-paquetes-de-radio>
- Alvarez Pineda, J. A. (2013). *Diseño, Construcción y Configuración de un Sistema de Control mediante SMS aplicado a Domotica*. Cuenca: Universidad del Azuay.
- Arduino Lab. (diciembre de 2012). *Menos media y mas comunicacion*. Obtenido de Placa Arduino Uno: <http://www.menosmedia.org/spip.php?article43>
- Canché, Á., & Mukul, J. (2013). *LocaPet. Localización Satelital para Mascotas*. Yucatán: ITM.
- Castro Dominguez, A. (s.f). *Sistema d control de temperatura a traves de arduino y la tecnologia GPRS/GSM*. E.U.I.T.TELECOMUNICACION.
- Clavijo Mendoza, J. R. (2011). *Diseño y simulación de sistemas microcontrolados en lenguaje C*. Colombia: ISBN.
- Cuzco Carrión, G., & Layana Vallejo, W. (2012). *Sistema de alarma automatización y control de equipos a distancia a través de línea telefonica y pagina web*. Guayaquil: UPS, Guayaquil.
- Dujmovic, M. (2006). *Diseño de un entrenador de microcontroladores PIC para el laboratoriao de microcontroladores del Instituto Universitario Jesús Obrero*. Sartenejas: Universidad Simon Bolívar.
- ElectrotekMEGA. (2014). *Comercio electronico y soluciones de ingenieria*. Obtenido de [http://electrotekmega.com/producto/tarjeta-desarrollo-entrenadora-microcontroladores-pic/#lightbox\[gal\]/0/](http://electrotekmega.com/producto/tarjeta-desarrollo-entrenadora-microcontroladores-pic/#lightbox[gal]/0/)
- Galarza, C. (19 de mayo de 2015). *INVESTIGACION DE CIENCIA Y TECNOLOGIA*. Obtenido de <http://cienciaytecnologiapanchos.blogspot.com/2015/05/los-fundamentos-matematicos-de-la.html>
- Gonzalez Rojas, K. V., & Mendez Caviativa, J. A. (2011). *Diseño y Fabricación de un modulo entrenador implementado a la familia de microcontroladores dsPIC33F*. Bucaramanga: Universidad Pontifica Boliviana .

- INNCOMEX. (2013). *Comunicaciones y Electronica - Equipamento de microcomputadoras*. Obtenido de [http://www.inncomex.com.mx/productos/pdf/mc\\_cic-500.pdf](http://www.inncomex.com.mx/productos/pdf/mc_cic-500.pdf)
- Jesús, R. (16 de septiembre de 2013). *Geek Factory*. Obtenido de Sensor DHT11 con PIC: <http://www.geekfactory.mx/tutoriales/tutoriales-pic/dht11-con-pic/>
- MCPros. (22 de mayo de 2009). *MicroController Pros LLC*. Obtenido de [http://microcontrollershop.com/product\\_info.php?products\\_id=3404](http://microcontrollershop.com/product_info.php?products_id=3404)
- MIT App Inventor. (s.f). *Aprende App Inventor*. Obtenido de ¿Qué es App Inventor?: <https://sites.google.com/site/aprendeappinventor/>
- Morlote, H. (2013). *Estudio de la estructura interna del PIC 16F877*. Obtenido de Blog de Electrónica: <https://cifpn1hectorm.wordpress.com/2013/04/10/estudio-de-la-estructura-interna-del-pic-16f877/>
- Prieto Donate, F. (s.f.). Obtenido de <http://bibing.us.es/proyectos/abreproy/11372/fichero/Memoria%252F03+-+GPRS.pdf>
- termotasajero-FUDOC. (8 de septiembre de 2014). *Manual práctico del pollo de engorde*. Obtenido de SlideShare: <http://es.slideshare.net/raulboadarojas/manual-prctico-del-manejo-de-pollo-de-engorde>
- TinySine Electronics. (2014). *TinySine*. Obtenido de GSM/GPRS Shield Version 1.0 Datasheet: [http://www.tinyosshop.com/index.php?route=product/product&product\\_id=464](http://www.tinyosshop.com/index.php?route=product/product&product_id=464)
- UbiMaker. (27 de julio de 2014). *instructables*. Obtenido de Como controlar un LED remotamente con Arduino y Ubidots: <http://www.instructables.com/id/Como-controla-un-LED-remotamente-con-Arduino-y-Ubi/>

# CAPÍTULO VIII

## 8. ANEXOS

### ANEXO 1.- ESQUEMAS DE CIRCUITOS DE LAS PLACAS EN EL SIMULADOR

Circuito N°1: Módulo GSM/GPRS SIM900

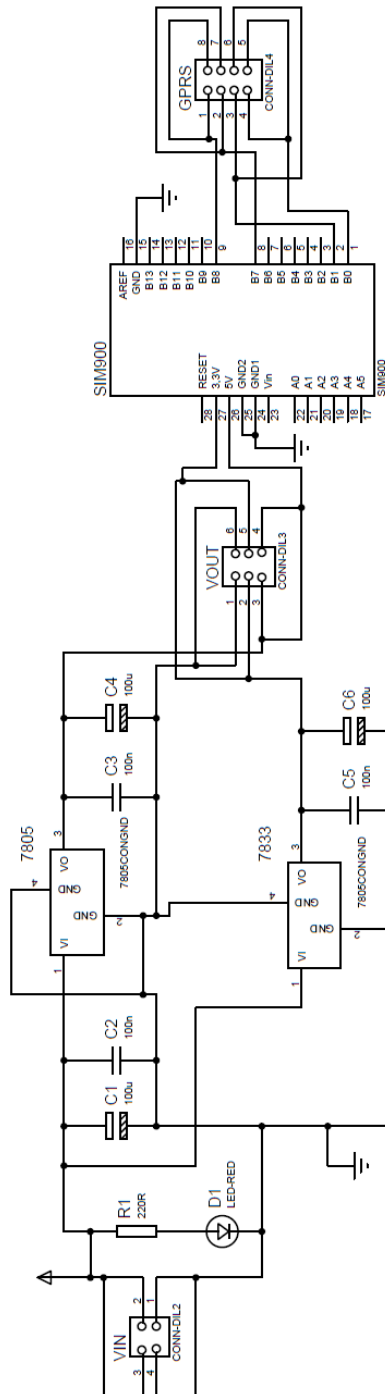


FIGURA N° 78.- Circuito Electrónico para placa SIM900  
FUENTE: Autores

## Circuito N°2: Transceiver SV651, MAX232 y MAX485

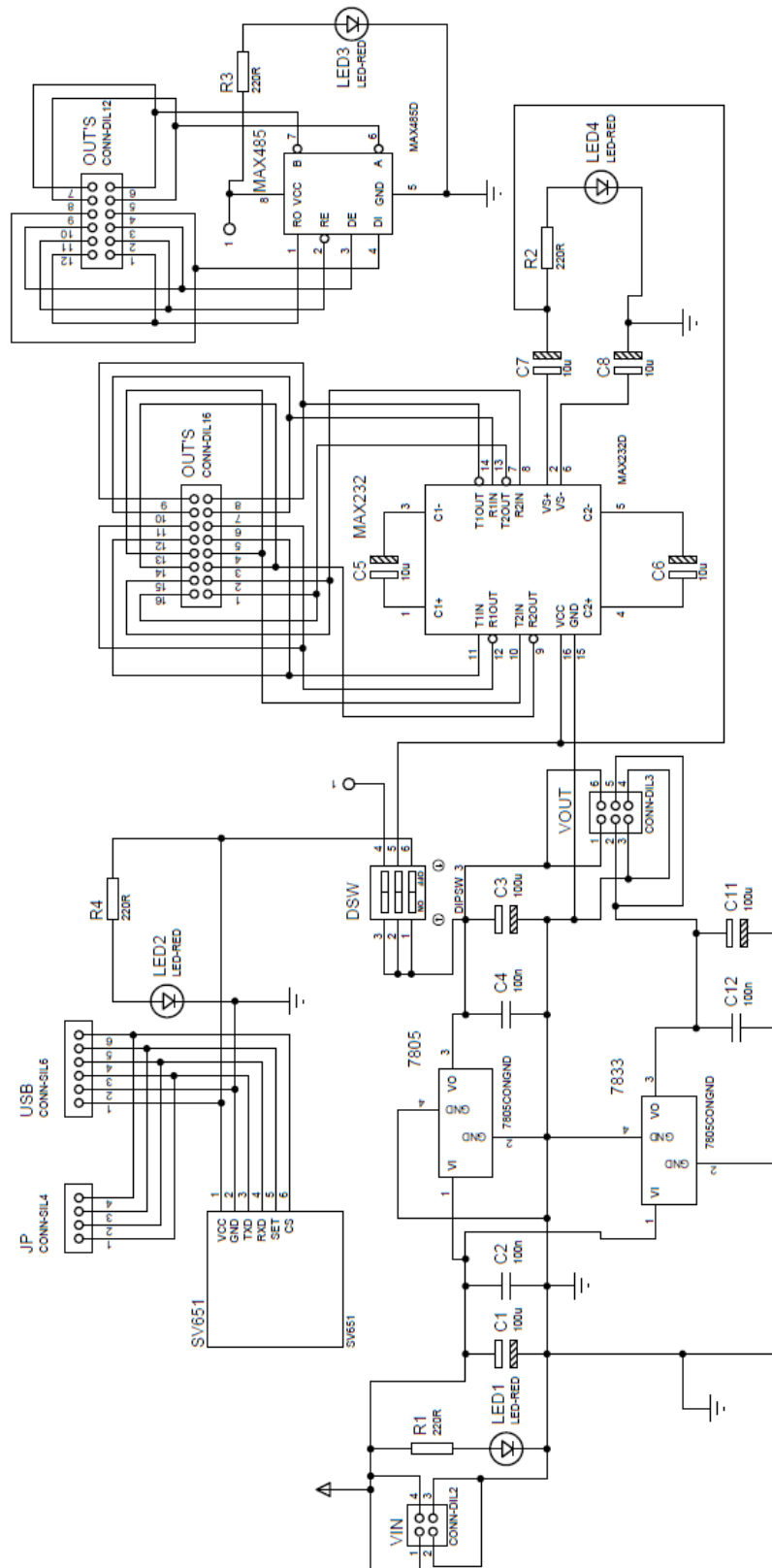


FIGURA N° 79.- Circuito para placa Transceiver SV651, MAX232 y MAX485  
FUENTE: Autores

### Circuito N°3: Teclado, pulsadores y dipswitches

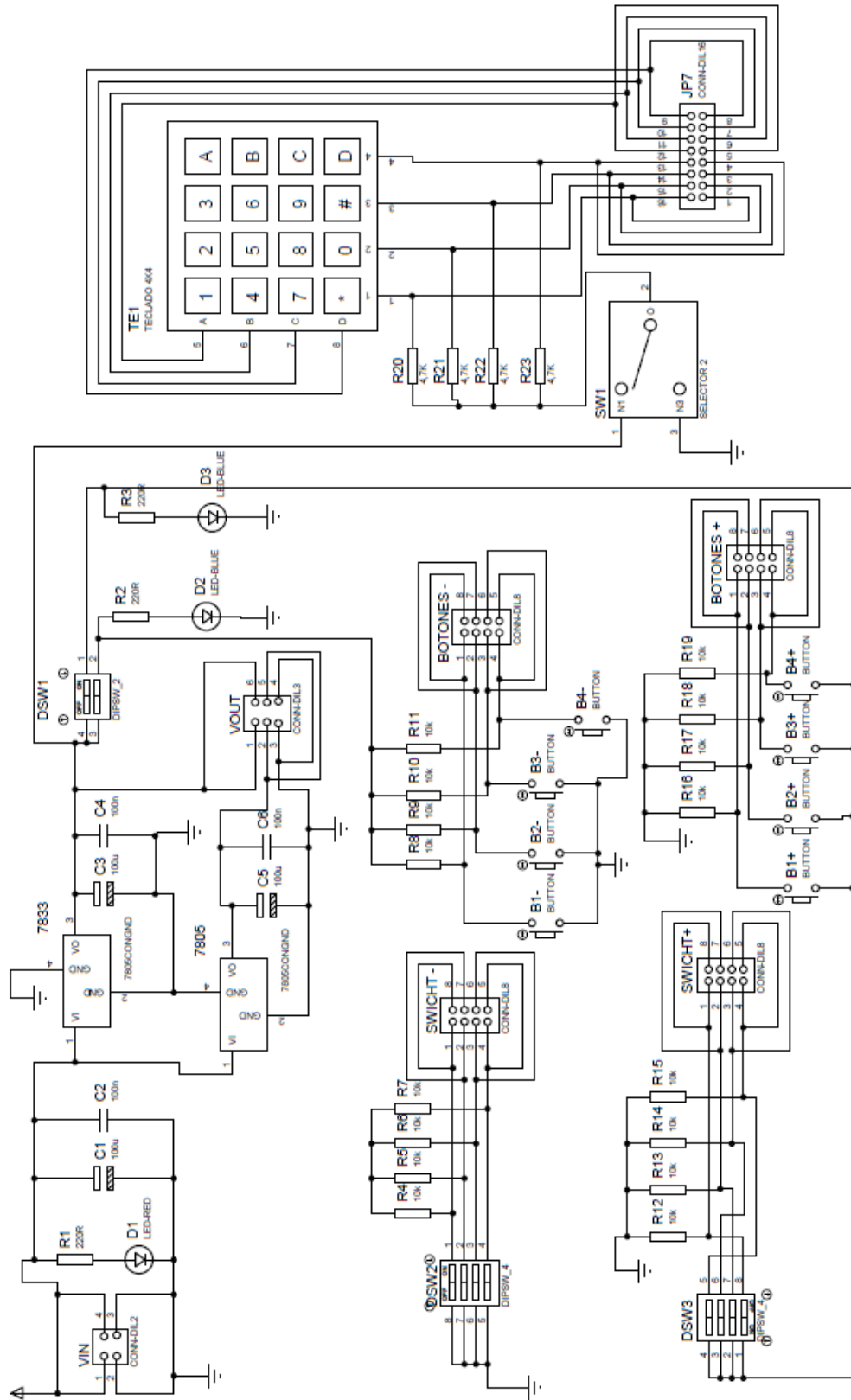


FIGURA N° 80.-Circuito para placa: teclado, pulsadores y dipswitches  
FUENTE: Autores

## ANEXO 2. DISEÑO DE PLACAS ELECTRÓNICAS PARA ENTRENADOR

### Placa N°1: Módulo GSM/GPRS SIM900

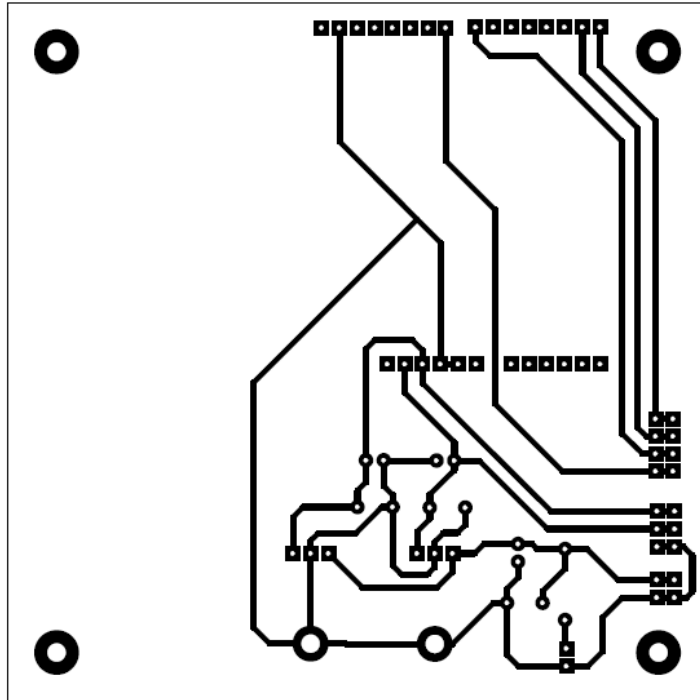


FIGURA N° 81.-Diseño de placa para SIM900  
FUENTE: Autores

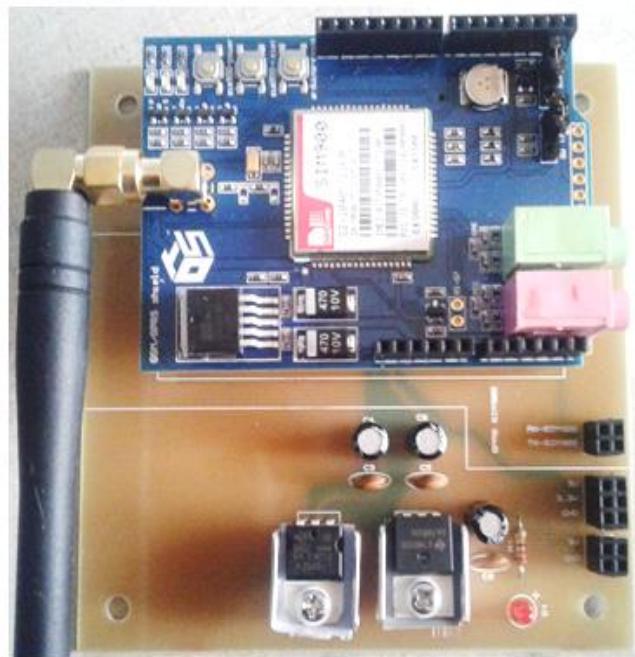


FIGURA N° 82.- Placa para SIM900  
FUENTE: Autores

**Placa N°2: Transeiver SV651, MAX232 y MAX485**

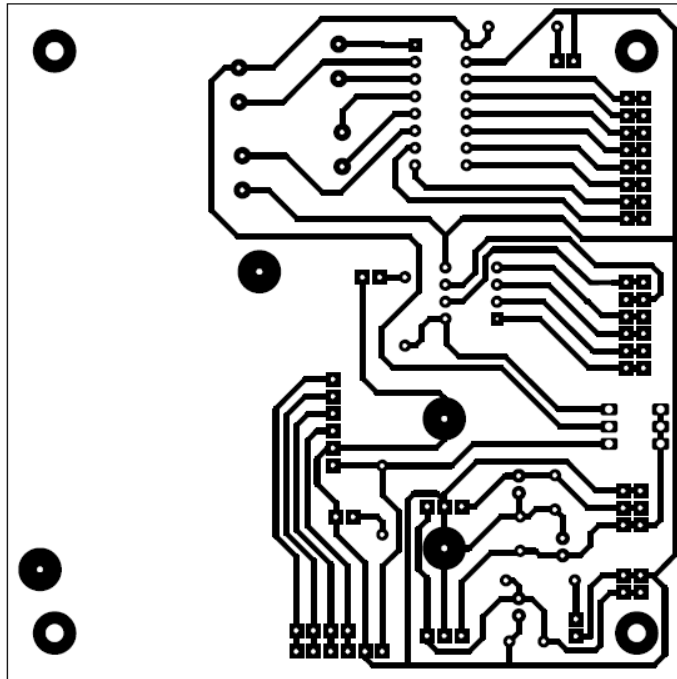


FIGURA N° 83.-Diseño de placa para Transeiver SV651, MAX232 y MAX485  
FUENTE: Autores

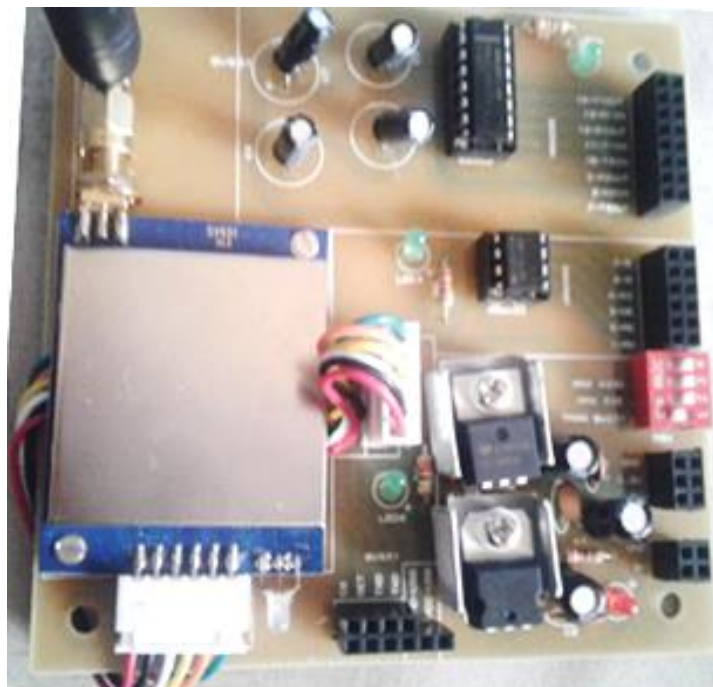


FIGURA N° 84.-Placa para Transeiver SV651, MAX232 y MAX485  
FUENTE: Autores



## Placa N°2: Teclado, pulsadores y dipswitchs

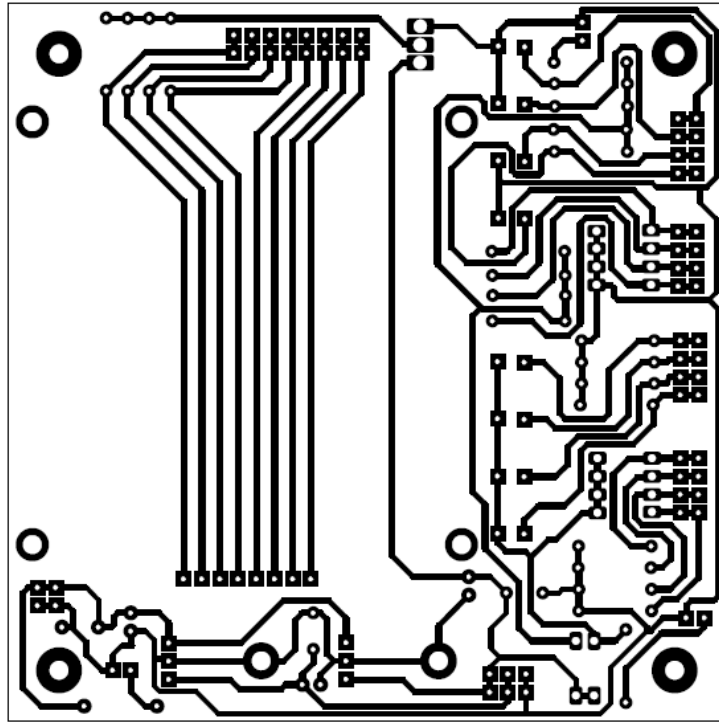


FIGURA N° 85.-Diseño de placa para Teclado, pulsadores y dipswitchs  
FUENTE: Autores



FIGURA N° 86.-Placa para Teclado, pulsadores y dipswitchs  
FUENTE: Autores

## ANEXO 3.- PRÁCTICAS DE LABORATORIO

### Práctica de laboratorio 1.1: Comunicación con el módulo SIM900 mediante comandos AT manuales.

#### Topología:

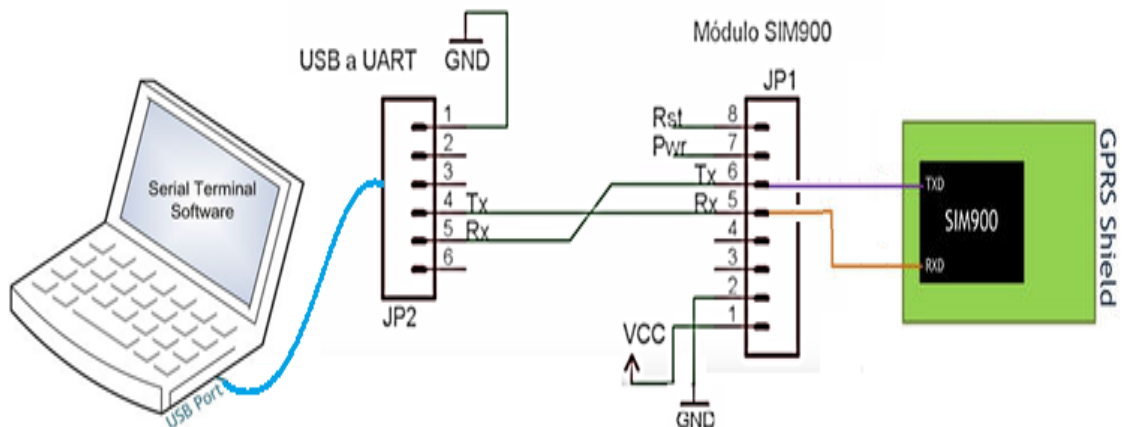


FIGURA N°1.1.- Conexión del módulo SIM900 a la PC mediante USB a UART

#### Objetivos de aprendizaje

- Establecer comunicación PC – módulo SIM900.
- Manipulación de comandos básicos AT.
- Configurar del módulo SIM900.
- Realizar llamadas y envío de SMS.

#### Información Básica/ Situación

Antes de comenzar una práctica de comunicación vía red celular en la que utiliza un módulo SIM900, se debe verificar la funcionalidad de este ejemplar. Asegurándose que el dispositivo tenga una velocidad de transmisión igual con la que se va a trabajar, cuente con la tarjeta SIM, disponga de saldo suficiente para poder realizar llamadas y enviar mensajes, entre otras características que se debe revisar. En esta práctica se proporciona un procedimiento para la revisión de funciones básicas del módulo SIM900, interactuando por medio de comandos AT y la ayuda de un terminal serial.

**Nota:** El terminal serial es un software que permite la comunicación dispositivo - ser humano de acuerdo a los comandos que se digite. Existe en internet varios software como: Hype terminal, Putty, TeraTerm,

RealTerm, entre otros. La presente práctica utiliza el TeraTerm, por la facilidad de manejo, no requiere instalación porque es portable y de descarga gratis desde:

- <http://osdn.jp/projects/tssh2/downloads/63335/teraterm-4.87.zip/>
- <http://logmett.com/index.php?/download/free-downloads.html>
- <http://www.4shared.com/rar/pzK6pOiUba/teraterm-483.html?>

## Recursos necesarios

- Módulo SIM900
- Computador
- Cable USB de serial RS232
- Teléfono Celular.

## Parte 1: Conexión de Modulo SIM900–Computador con cable RS232

En la parte 1, conectará el módulo GSM/GPRS a una PC; para establecer comunicación.

### Paso 1: Conectar el cable al módulo SIM900.

En la figura N°1.1 se divide el diagrama de cómo se debe conectar el cable. Y en la figura N°1.2 la conexión física del cable al módulo. Al término del armado conectar el USB a la PC.

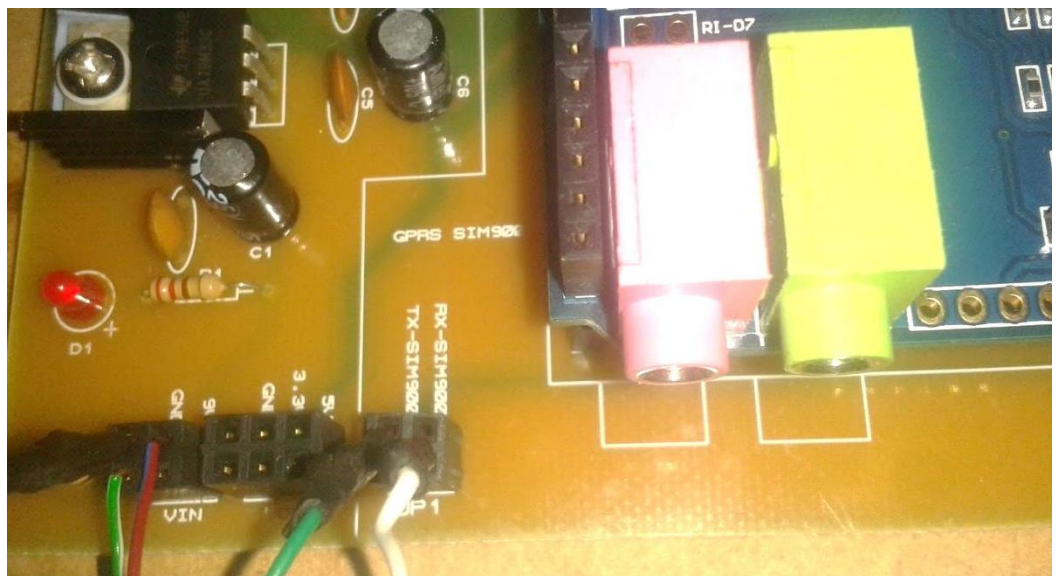


FIGURA N° 1.2.- Conexión física del SIM900 a la PC

## Paso 2: Revisar Puerto COM del USB en la PC.

Si utiliza una PC con Microsoft Windows 7 hacer los siguientes pasos:

- Dar clic en el icono de *Inicio de Windows* y acceder al *Panel de control*.
- Abrir el *Administrador de dispositivos*.
- Dar clic en el enlace de árbol *Puertos (COM y LPT)* para expandirlo.
- Conectado el USB se visualiza el *Prolific USB-to-Serial Comm Port (COM#)* que le corresponde. Ver figura N°1.3.

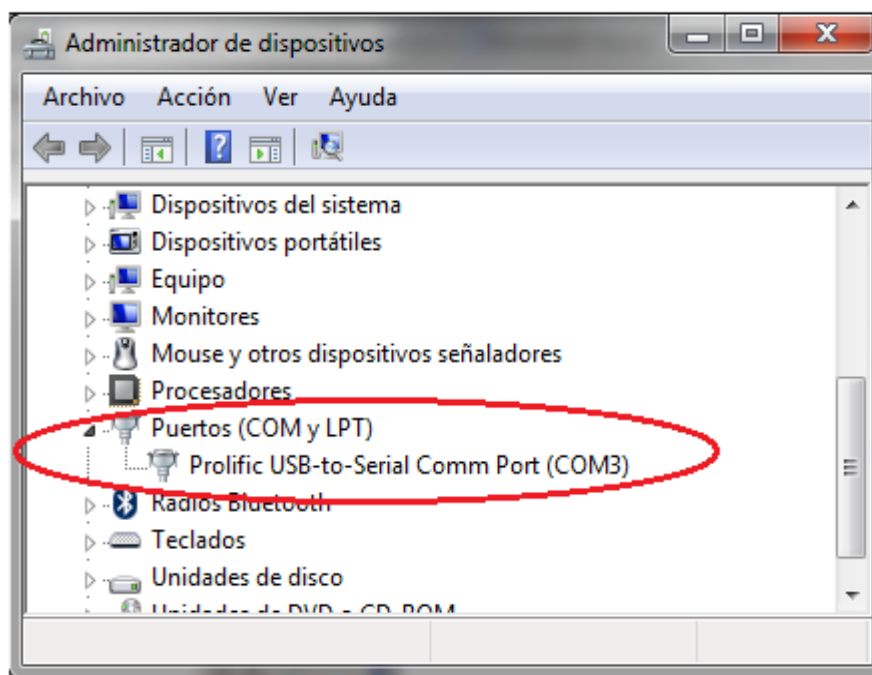


FIGURA N° 1.3.- Administrador de dispositivos

## Parte 2: Configuración de software de terminal serial (TeraTerm).

El software TeraTerm debe ser configurado para poder establecer la comunicación con el Módulo GSM.

### Paso 1: Inicio TerraTerm.

En la carpeta descargada buscar el TerraTerm y abrirlo con doble clic. Aparecerá un cuadro de dialogo *New connection*, hacer clic en opción **Serial**. Verificar si el puerto COM es correcto y dar clic en **OK**. Ver figura N°1.4.

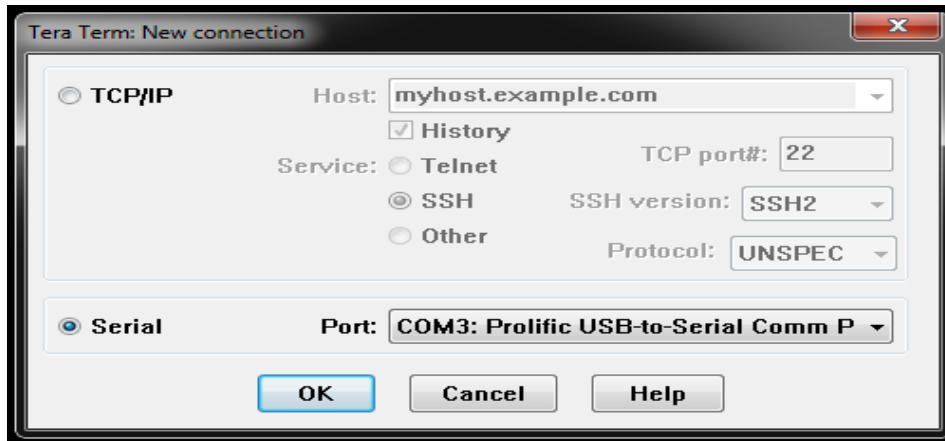


FIGURA N° 1.4.- Inicio TeraTerm

### Paso 2: Activación del retorno de carro y salto línea (CR+LF).

Ir a *Setup*, seleccionar *Terminal...* y en la ventana que se abre hacer la selección de *CR+LF* en *Transmit* como se ve la figura 1.5; y dar clic en *OK*.

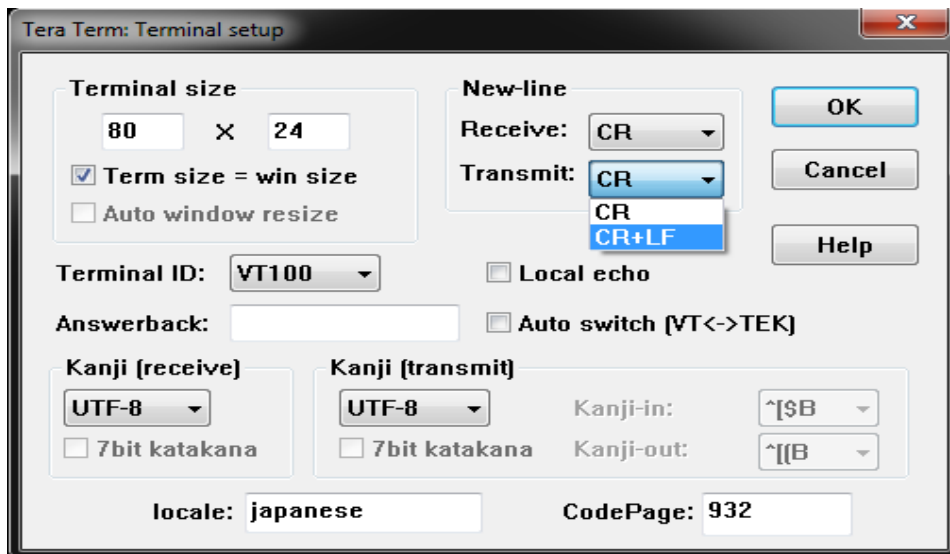


FIGURA N° 1.5.- Activación CR+LF del TeraTerm

### Paso 3: Configuración de la Velocidad de Transmisión.

Clic en *Setup*, elegir *Serial port....* Configurar la ventana como se observa en la figura N°1.6 y dar *OK*. El Baud rate depende de la velocidad de transmisión a la que este configurado el módulo SIM900, por lo general es de 9600 baudios. Si el módulo es nuevo por defecto está a 0 baudios (detección automática del baude rate) ó a 115200 baudios esto depende del

fabricante por lo que se recomienda revisar el manual, para poder configurarlo según su requerimiento.

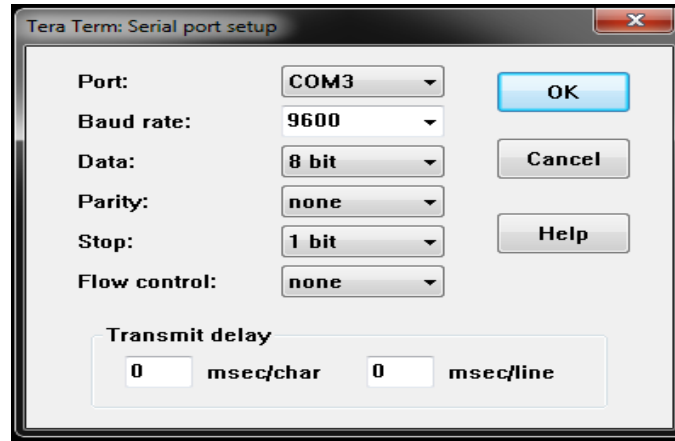


FIGURA N° 1.6.- Configuración de velocidad transmisión

### Parte 3: Configuración del módulo SIM900 y manipulación de comandos AT básicos.

Una vez que el software este configurado y las conexiones finalizadas se puede empezar a interactuar con el módulo SIM900.

#### Paso 1: Inicio de comunicación PC - SIM900.

Encender el módulo SIM900 pulsando el botón *power* de la placa por más de dos segundos. A continuación se recibe un mensaje de bienvenida; si el módulo es nuevo se visualiza la respuesta como en la figura N°1.7; y si ya es de uso recurrente su respuesta será como se ve en la figura N°1.8, siempre y cuando tenga una tarjeta SIM caso contrario enviara un mensaje de que no tiene tarjeta SIM.

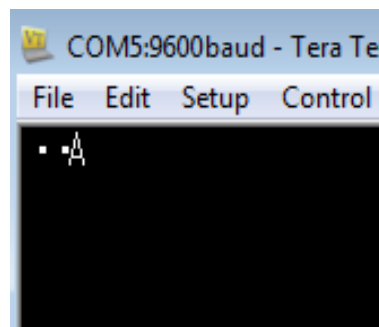


FIGURA N° 1.7.- Respuesta de encendido de Módulo SIM900 (nuevo)



```
..  
RDY  
  
+CFUN: 1  
  
+CPIN: READY  
  
+PACSP: 1  
  
Call Ready
```

FIGURA N° 1.8.- Respuesta de encendido de Módulo Sim900 (reincidente - configurado)

**Nota:** en caso que no aparezca nada en la pantalla (TeraTerm), revisar las conexiones y configuraciones del software.

### Paso 2: Primer comandos AT

Iniciar la interacción verificando si el módulo SIM900 responde correctamente a las instrucciones con el comando *AT* y presionando *ENTER*. Ver figura N°1.9. Responderá *OK* si el comando es correcto y *ERROR* si no lo es.

```
AT  
  
OK  
AT  
  
OK  
AT r  
  
ERROR
```

FIGURA N° 1.9.- Comando de atención

### Paso 2: Consultar opciones de Baud rate que soporta el módulo SIM900.

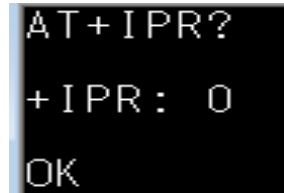
Digitar el comando *AT+IPR=?* y presionar *ENTER*, se obtendrá una respuesta como se ve en la figura N°1.10.

```
OK  
AT+IPR=?  
  
+IPR: ( ), (0,1200,2400,4800,9600,19200,38400,57600,115200)  
  
OK
```

FIGURA N° 1.10.- Consulta de Baud rate que acepta el SIM900

### Paso 3: Consulta de Baud rate actual del SIM900.

Escribir el comando *AT+IPR?* y oprimiendo ENTER se verificará el Baud rate que tiene el SIM900. Ver figura N°1.11; la respuesta dependerá de la configuración del módulo.

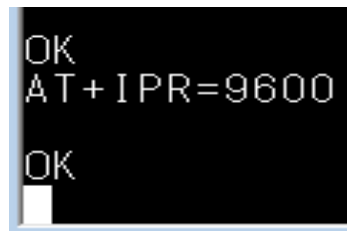


```
AT+IPR?  
+IPR: 0  
OK
```

FIGURA N° 1.11.-Baud rate actual

### Paso 4: Configuración de nuevo Baud rate.

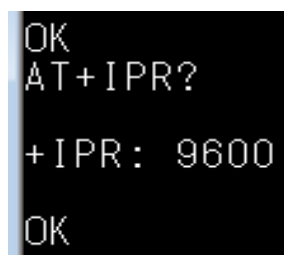
Digitar el comando *AT+IPR= <nuevo Baud rate>* y *ENTER*. El Baud rate con el que se trabajará es de 9600 baudios, por esto se configura como se visualiza en la figura N°1.12.



```
OK  
AT+IPR=9600  
OK
```

FIGURA N° 1.12.-Nuevo Baud rate

Revisar nuevamente con el comando *AT+IPR?*, se debe observar la respuesta del nuevo Baud rate. Ver figura N°1.13.



```
OK  
AT+IPR?  
+IPR: 9600  
OK
```

FIGURA N° 1.13.- Nuevo Baud rate

**Nota:** siempre que cambie la velocidad de transmisión del Módulo también se debe cambiar en el software terminal serial (Tera Term) para que esté a la misma velocidad y se pueda seguir interactuando.



### Paso 5: Revisar información del Producto

Para este paso se debe correr el comando *ATI* y *ENTER*. Ver figura N°1.14 para verificar respuesta.

```
ATI
SIM900 R11.0
OK
```

FIGURA N° 1.14.-Informacion del Producto

### Paso 6: Información del fabricante.

Ejecutar el comando *AT+CGMI* y *ENTER*. Ver figura N°1.15

```
AT+CGMI
SIMCOM_Ltd
OK
```

FIGURA N° 1.15.-Informacion de fabricante

### Paso 7: Nombre de la operadora de telefonía celular.

Escribir el comando *AT+COPS?* Y con *ENTER*. Ver figura N°1.16.

```
AT+COPS?
+COPS: 0,0,"PORTA GSM"
OK
```

FIGURA N° 1.16.-Nombre de la operadora

### Paso 8: Recabar número del IMEI.

Digitar el comando *AT+CGSN* mas *ENTER*. Ver figura N°1.17.

```
AT+CGSN
013950007225403
OK
```

FIGURA N° 1.17.- IMEI del módulo Sim 900

## Parte 4: Hacer/aceptar llamadas, enviar/recibir SMS.

Determinará si se puede o no establecer comunicación con el mundo por medio de la red celular utilizando alguno de los servicios brindados por la operadora.

### Paso 1: Realizar una llamadas desde el módulo SIM900.

Es posible realizar una llamada a cualquier número de teléfono celular desde el módulo GSM/GPRS utilizando el comando **ATD09XXXXXXXX;** y **ENTER**. Ver figura N°1.18.

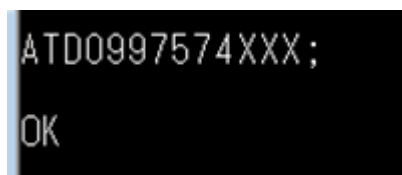


FIGURA N° 1.18.- Haciendo llamada desde SIM900 (TeraTerm)

Al presionar ENTER inicia la llamada y el teléfono a quien se hace la llamada empezara a timbrar. Ver figura N°1.19.

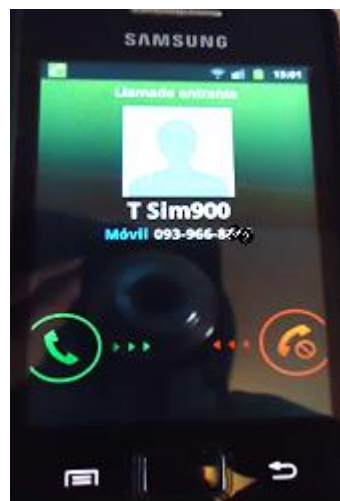


FIGURA N° 1.19.-Llamada entrante en teléfono celular

### Paso 2: Aceptar/rechazar llamadas en el módulo SIM900.

Desde un teléfono celular marcar el número de la tarjeta SIM insertada en el módulo GSM/GPRS y realizar una llamada. Ver figura N°1.20.



FIGURA N° 1.20.- Marcado y llamando a SIM900

El módulo SIM900 empezará mostrar un mensaje (*RING*) en el serial terminal, indicando que tiene una llamada entrante. Ver figura N°1.21

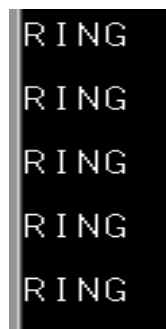


FIGURA N° 1.21.-Llamada entrante en SIM900

Para contestar la llamada digitar el comando *ATA* (*de Answer*) y *ENTER*, desde ese momento el SIM900 contesta y la llamada está en proceso. Para cortar la llamada escribir el comando *ATH* (*de Hung-up*) y *ENTER*. Ver figura N°1.22.



FIGURA N° 1.22.-Contestar y colgar llamada en SIM900

### Paso 3: Enviar SMS desde el módulo SIM900.

El envío de SMS desde el módulo es posible con los siguientes pasos:

- Digitar el comando **AT+CMGS="09XXXXXXXX"**
- Tecla **ENTER**.
- Seguido aparecerá el signo de *mayor que* (>), es el indicador del inicio del mensaje que se va enviar.
- Al terminar de escribir el mensaje se debe enviar el carácter **Control-Z** para ello se debe pulsar simultáneamente las teclas: <Ctrl> + <Z>.
- Si dispone de saldo o paquete de mensajes se enviará el SMS y el modulo mostrará un mensaje de **OK**. Ver figura N°1.23.
- Revisar el teléfono celular para confirmar la llegada del mensaje. Ver figura N°1.24.

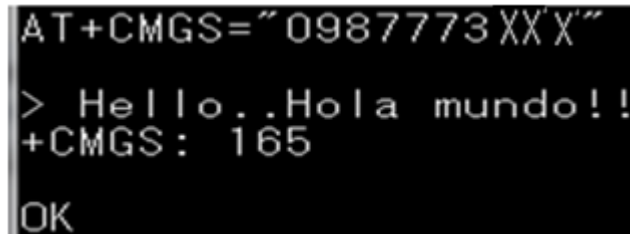


FIGURA N° 1.23.- Envío de SMS desde SIM900



FIGURA N° 1.24.-Mensaje recibido en teléfono celular

### Paso 4: Recibir SMS en el módulo SIM900

Escribir un mensaje de texto desde un teléfono celular y enviar al número del SIM del módulo (ver figura 1.25). En el software de terminal serial se visualiza una cadena de caracteres que indica: **+CMT:** "El número de quien se recibió el mensaje", "el nombre (si es que está registrado)", "la fecha y hora" y al final el mensaje de texto. Ver figura 1.26.



FIGURA N° 1.24.-Mensaje enviado desde teléfono celular



FIGURA N° 1.25.-Mensaje recibido en módulo SIM900

## PARTE 5: Apagado del modulo

Simplemente presionar el botón de **power** de la placa por más de más de dos segundos. En el terminal serial se visualizará un mensaje de **NORMAL POWER DOWN**. Ver figura 1.26.

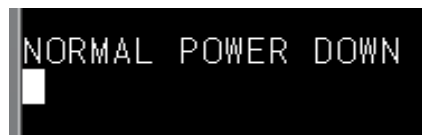


FIGURA N° 1.26.- Módulo SIM900 APAGADO

## Práctica de laboratorio 1.5: Diseño de una aplicación en sistema operativo Android, para el control de una interfaz gráfica mediante mensajes de textos.

### Topología:

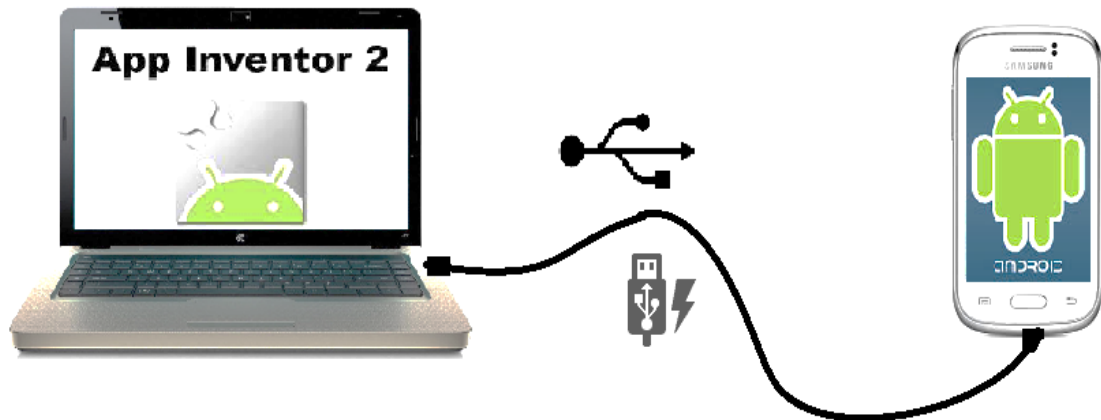


FIGURA N° 5.1.- Conexión PC con un teléfono con sistema operativo Android.

### Objetivos de aprendizaje

- Desarrollar aplicaciones en Android.
- Crear archivos .APK para instalar en teléfonos con sistema operativo Android.
- Conocer las principales ventanas de la plataforma App inventor 2.
- Utilizar los elementos principales para el envío y recepción de sms.

### Información Básica

Android es un sistema operativo basado en el núcleo Linux, desarrollado para dispositivos electrónicos inteligentes (en especial *Smartphones*). La plataforma es adaptable para pantallas de mayor resolución, su almacenamiento es SQLite, una base de datos liviana, que es usada para propósitos de almacenamiento de datos.

Todas sus aplicaciones están escritas en lenguaje de programación Java. Pero Google posee la plataforma de App Inventor, para crear aplicaciones de forma visual y con herramientas básicas. El sistema es gratuito y se puede descargar de la web.

## Situación

En esta práctica se diseñará una aplicación que permita el control del encendido y apagado de un led en la interfaz gráfica de la aplicación, mediante mensajes de texto.

Se utilizará un teléfono celular que posea sistema operativo Android, puesto que, la plataforma App Inventor 2 solo genera archivos .APK que funcionan solo en dicho sistema operativo.

**Nota 1:** Para esta práctica se debe tener conexión permanente a internet, la plataforma a utilizar se encuentra en la siguiente dirección de la página web.

- <http://ai2.appinventor.mit.edu/>

## Recursos necesarios

- Computador.
- Conexión a internet.
- Teléfono con sistema operativo Android.
- Cable de datos.

## Parte 1: Ingreso a la plataforma App Inventor 2.

Para crear una aplicación en App Inventor2, es necesario tener una cuenta de correo electrónico en Gmail, para lo cual si no lo posee crear uno.

### Paso 1: Buscar App Inventor 2.

Buscar en Google **App Inventor 2**, y dar clic en la primera opción. Ver figura N°5.2.

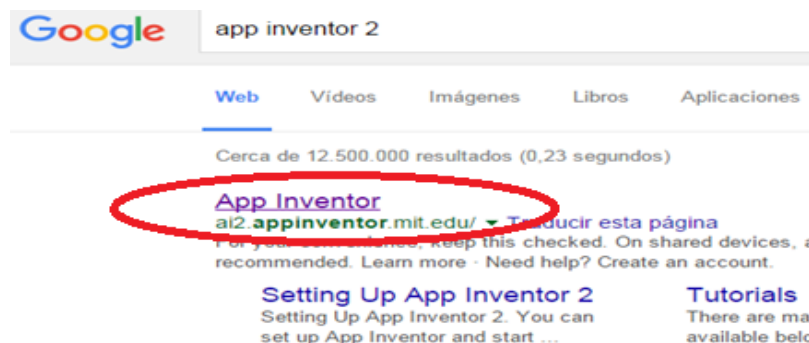


FIGURA N° 5.2.- Búsqueda de App Inventor 2 en Google



## Paso 2: Ingreso a cuenta Gmal.

Al término del **paso 1**, aparecerá la petición de ingresar a la cuenta Gmail.  
Ver Figura N°5.3.

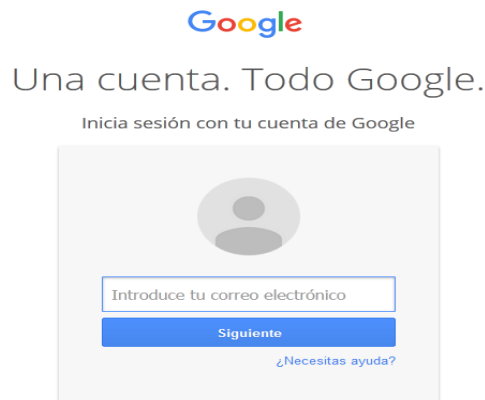


FIGURA N° 5.3.- Ingresar cuenta de correo electrónico de Gmail.

## Paso 3: Aceptar términos de servicio.

Finalizado el **paso 2**, se abre la ventana de los términos de servicio en donde se debe aceptar las condiciones. Ver Figura N°5.4.

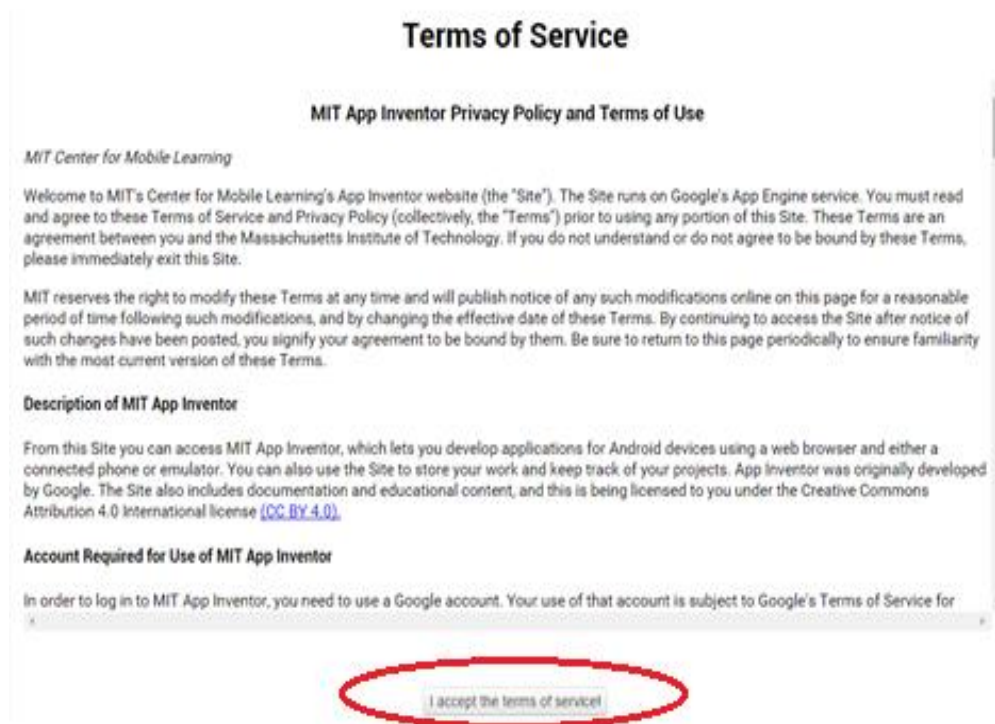


FIGURA N° 5.4.- Terminos de servicio de App Inventor



## Paso 4: Ventana de App Inventor 2.

Aceptado los términos automáticamente se crea una cuenta en la plataforma en App Inventor 2 y se visualiza la siguiente ventana, ver Figura N°5.5.

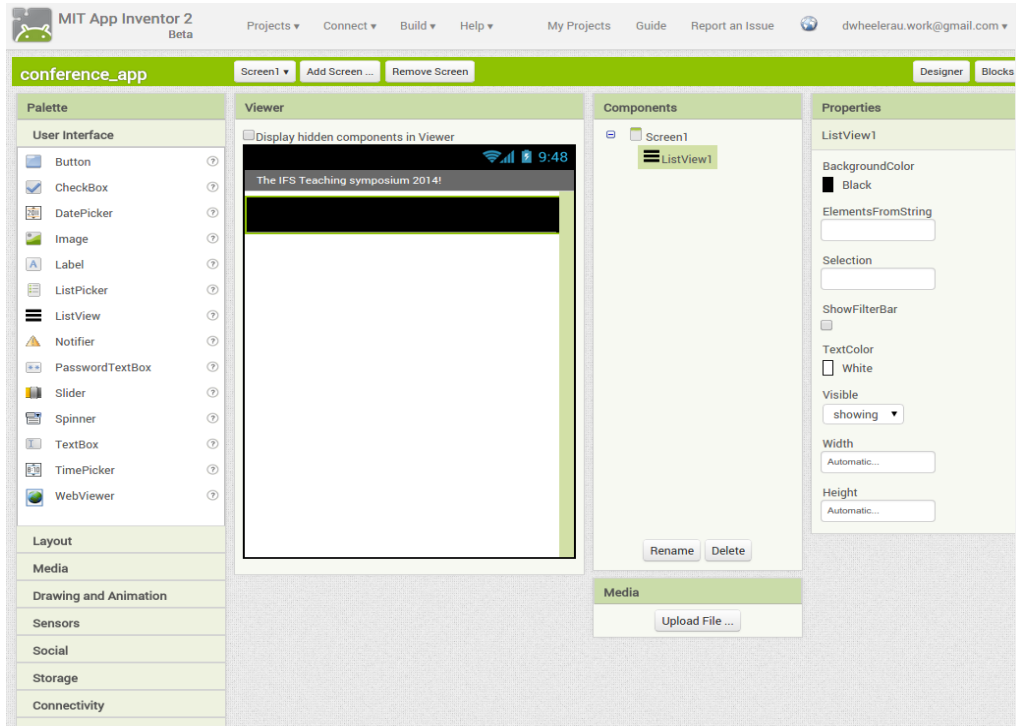


FIGURA N° 5.5.- Ventana de App Inventor 2.

## Paso 5: Cambio de idioma de la plataforma.

Ir a los iconos de la parte superior de la plataforma, por lo general se activan e inglés, por lo tanto buscar el icono que dice **English**, dar clic y seleccionar **Español**.

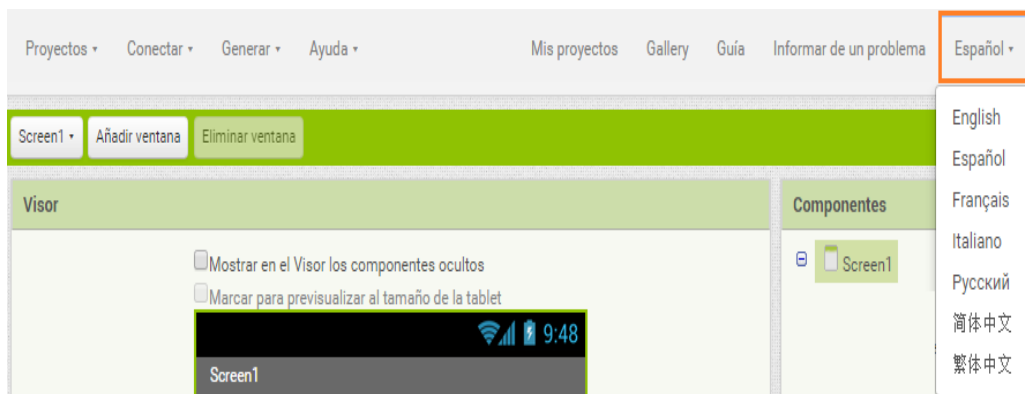


FIGURA N° 5.6.- Cambio de idioma de la plataforma App Inventor2.

## Parte 2: Desarrollo de la aplicación para Android.

### Paso 1: Crear un nuevo proyecto.

En la parte superior izquierda dar clic en la pestaña **Proyecto**, seleccionar la opción **Comenzar un proyecto nuevo...** Ver Figura N°5.7.

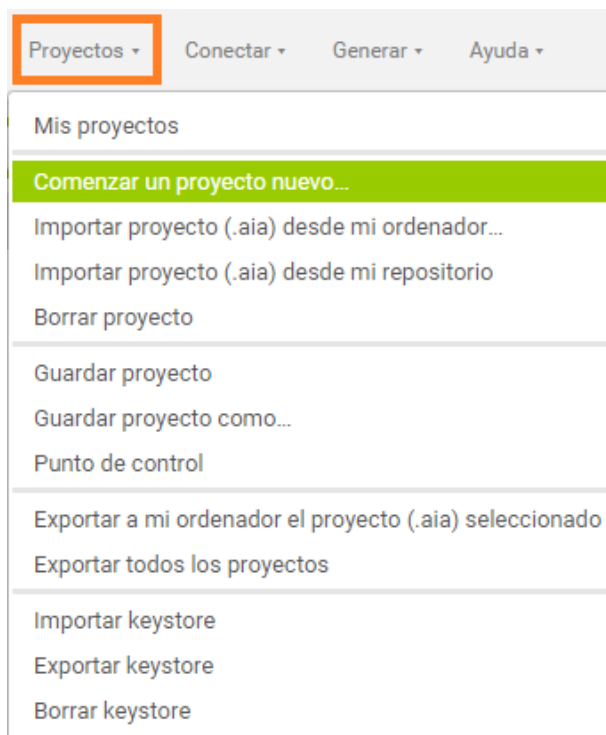


FIGURA N° 5.7.- Comenzar un nuevo proyecto

Seguido a esto aparecerá un cuadro de dialogo, donde se debe poner un nombre a la aplicación, ver Figura N°5.8. Una vez nombrado dar clic en **Aceptar**.

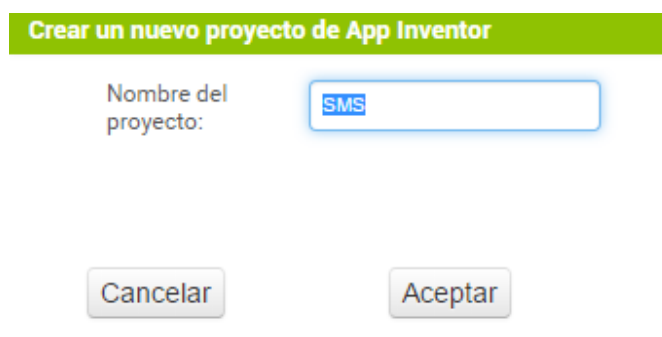


FIGURA N° 5.8.- Nombre de nueva aplicación.

## Paso 2: Ventanas principales de App Inventor 2.

En la figura N°5.9, se observa resaltados algunos bloques y ventanas que servirán de herramientas para el diseño de nuestra aplicación.

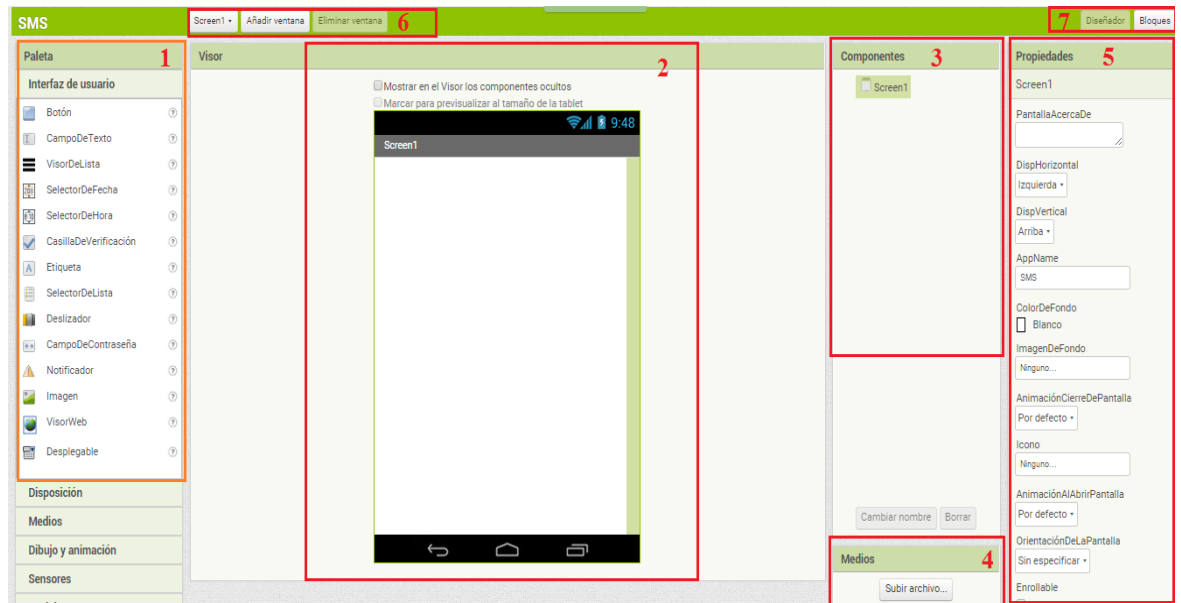


FIGURA N° 5.9.- Bloques y ventanas de la plataforma App Inventor2.

1. **Paleta:** Para seleccionar elementos y componentes de acuerdo a las necesidades, arrastrarlos al visor para incluir en la aplicación.
2. **Visor:** Lugar para añadir componentes y elementos para constituir la apariencia de la aplicación.
3. **Componentes:** Ventana que enlista los componentes incorporados a la aplicación.
4. **Medios:** Permite añadir imágenes, sonidos o videos a la aplicación.
5. **Propiedades:** Sirve para cambiar las propiedades de un componente seleccionado en la lista de componentes.
6. **Screen (Pantallas):** Crea ventanas dentro de la aplicación.
7. **Diseñador – Bloques:** Botones para cambiar del diseñador al editor de bloques o viceversa.

## Parte 3: Interfaz gráfica de aplicaciones para Android.

### Paso 1: Diseño de la interfaz gráfica de la aplicación.

Arrastrar desde el cuadro *Paleta* el elemento **Etiqueta** a la ventana del *Visor*, este permite mostrar el texto para la señalización adecuada para el usuario. Hacer esto en dos ocasiones. Ver figura N°5.10.

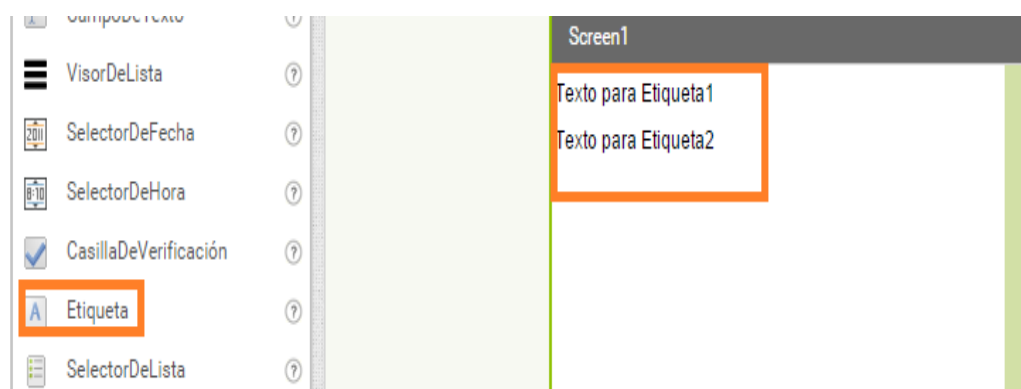


FIGURA N° 5.10.- Elemento Etiqueta en el Visor

Añadir **Botón** a la aplicación, este elemento realiza alguna acción cuando el usuario lo toca o pulsa. En el caso de esta aplicación enviará un mensaje de texto. Hacer esto en dos ocasiones. Ver figura N° 5.11.

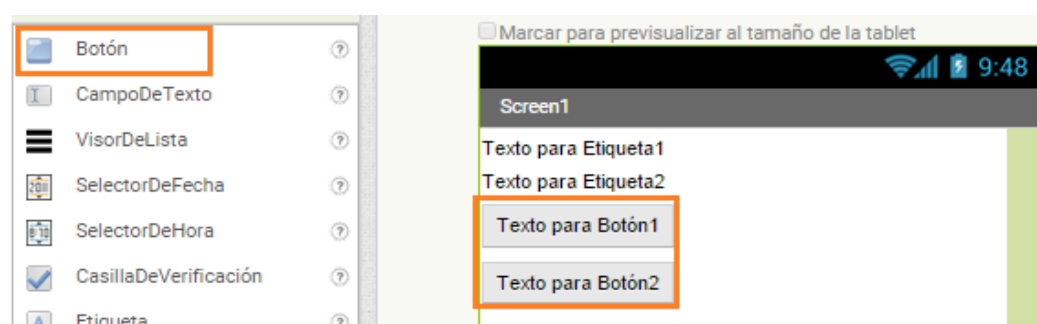


FIGURA N° 5.11.- Botón en la aplicación

Agregar el icono de **Imagen**, para representar imágenes que los usuarios seleccionan y manipulan. Ver figura N°5.12.

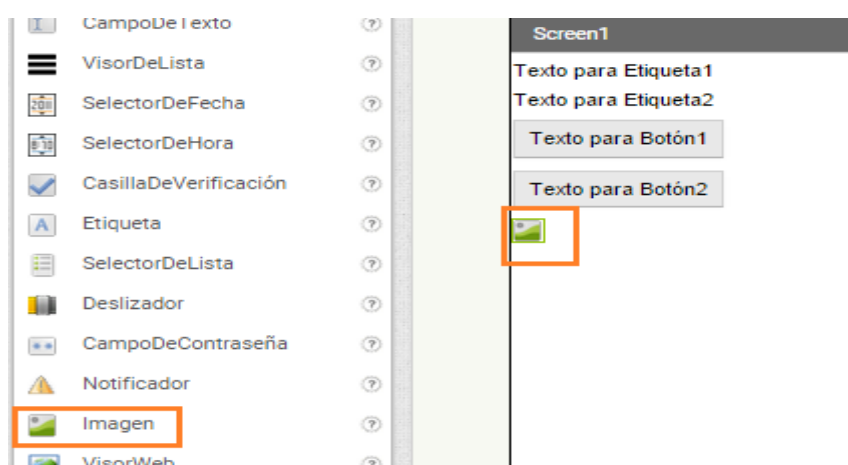


FIGURA N° 5.12.- Imagen en el Visor

## Paso 2: Capacidad de enviar mensajes de texto.

En *Paleta*, seleccionar la opción **Social** y arrastrar el elemento **EnviarTexto** a la pantalla del *Visor*, este posibilita la capacidad de enviar SMS.

**Nota:** el icono de EnviarTexto se lo puede observar en la parte baja del visor y no dentro de este, porque es un icono oculto. Ver Figura N°5.13.

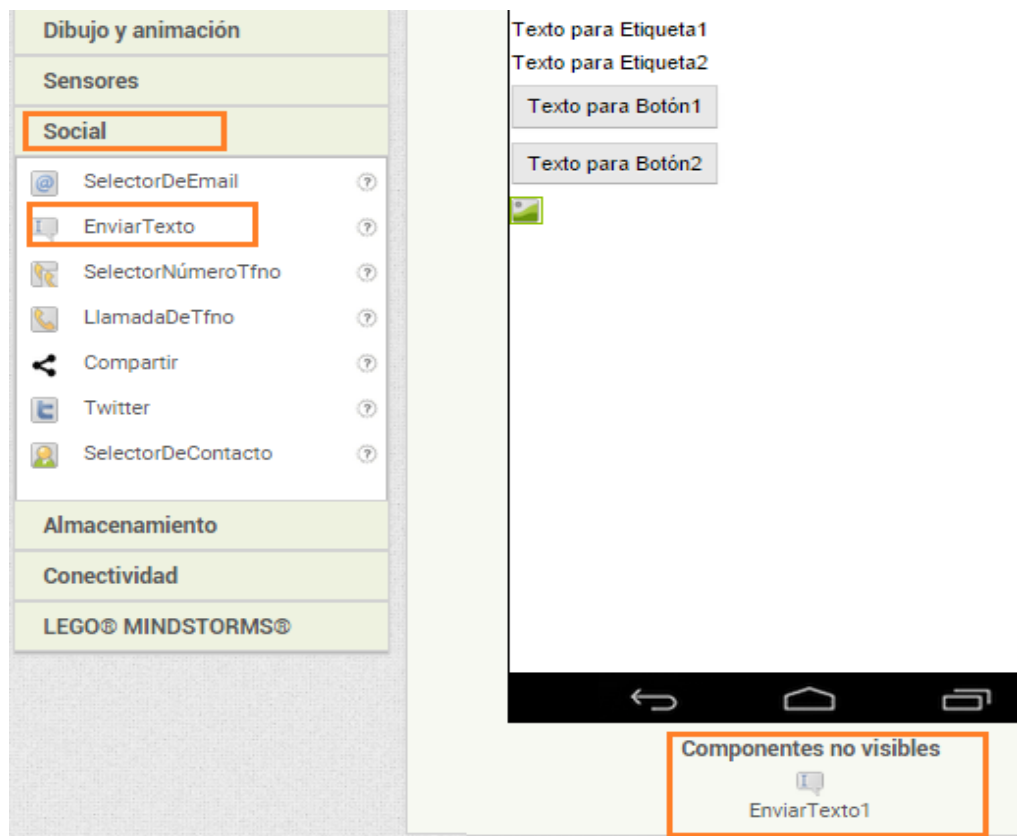


FIGURA N° 5.13.- Elemento enviar texto en la aplicación

## Paso 3: Cambio de las propiedades de los elementos del Visor.

Seleccionar el elemento a cambiar las propiedades en la ventana de *Componentes* y modificarlo en la ventana *Propiedades*, ver figura N°5.14.

En la figura se realiza los cambios para las Etiquetas, estos pueden variar de acuerdo a las necesidades y gustos del usuario. Así también se lo puede hacer con el Botón y entre otros elementos. Para determinar y saber qué es lo que hace cada uno de las variables de la ventana *Propiedades* se recomienda ir variando cada uno e ir observando lo cambios producidos.

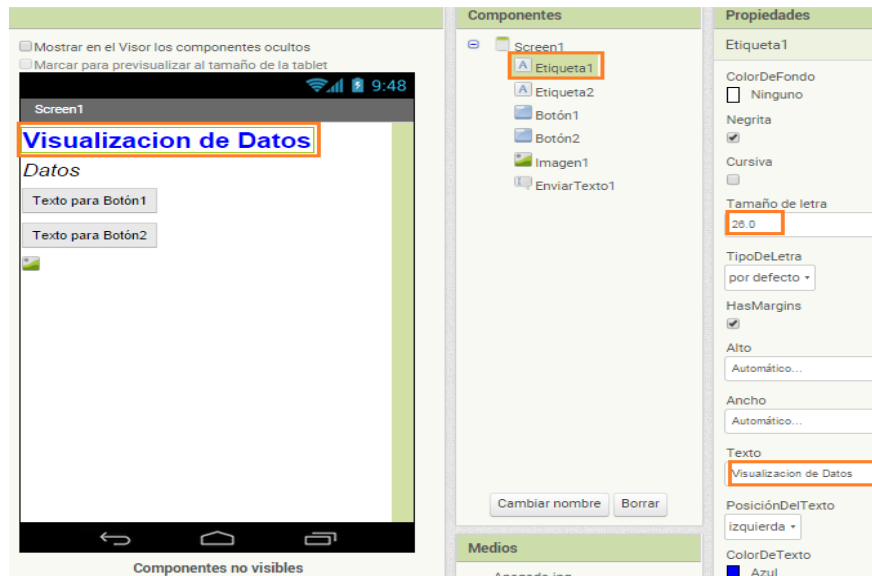


FIGURA N° 5.14.- Cambio los nombres de los elementos.

#### Paso 4: Subir imágenes a la plataforma.

Para la práctica se necesita imágenes para el botón y una representación de lámpara o luz (leds, focos, etc.), a quien se va hacer el control; es por esto que se requiere tener previamente imágenes referentes al tema.

**Nota:** tomar en cuenta el tamaño de la imagen, para evitar que ocupe toda la pantalla en vano.

Ir a la ventana **Medios**, clic en **subir archivo...**, clic en **seleccionar archivo** y buscar la ubicación de la imagen a subir, ver figura N°5.15. Seleccionar la imagen clic en **Abrir** y también en **Aceptar**.

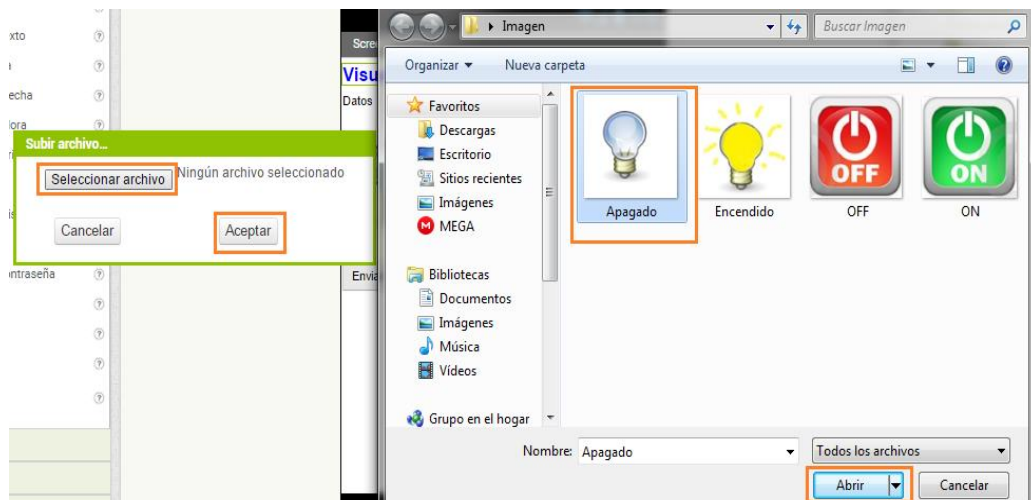


FIGURA N° 5.15.- Subir imágenes a la aplicación

Hacer esto para todas las imágenes necesarias. Al termino de esto irán apareciendo en la ventana *Medios* los archivos que ha ido subiendo. Ver figura N°5.16.

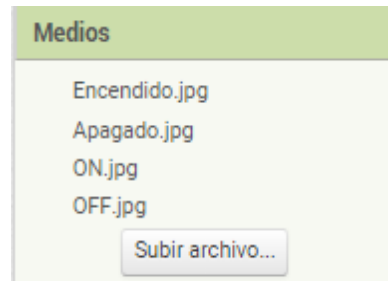


FIGURA N° 5.16.- Lista de elementos subidos a la plataforma.

### Paso 5: Carga de imágenes a la aplicación.

En el **Visor** o en la ventana *Componentes* dar clic en el elemento que se va a poner una imagen. Ir a *propiedades* buscar la variable **Foto** o **Imagen**, desplegar la paleta y seleccionar la imagen que desee para ese elemento. Ver figura N°5.17.

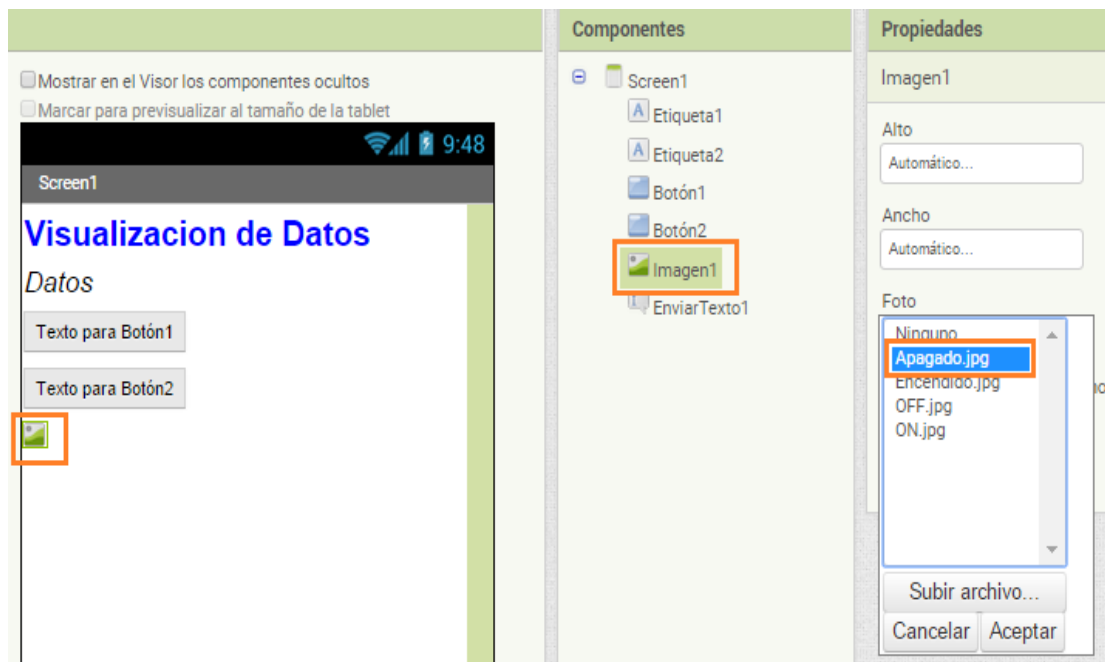


FIGURA N° 5.17.- Ubicar imagen en un elementos

Hacer esto para la *Imagen1* así también para los botones para mejor presentación de la aplicación. Ver figura N°5.18





FIGURA N° 5.18.- Elementos con sus imágenes

### Paso 6: Ordenar elementos de la aplicación.

Como se visualiza en la figura N°5.18, los elementos se encuentran desordenados y ocupan espacio en vano y la aplicación no tiene una buena presentación y lo mejor es ordenarlos adecuadamente.

Para ello ir a Paleta, seleccionar **Disposición** y arrastrar al Visor alguna de las opciones acorde a la necesidad o gusto del usuario. Ver figura N°5.19.



FIGURA N° 5.19.- Disposición en el Visor

Arrastrar los elementos a ordenar a la *Disposición* seleccionada. Hacer esto hasta que la aplicación tome la apariencia deseada del usuario. Ver figura N°5.20.





FIGURA N° 5.20.- Apariencia de la aplicación

#### Parte 4: Programación de la aplicación.

La aplicación se programa mediante bloques o piezas en el editor de bloques. Encaja las piezas como un rompecabezas para establecer el comportamiento de la aplicación. Arrastrando los bloques para utilizarlos en el programa.

##### Paso 1: Ventana “Editor de bloques” para la programación.

En la figura N°5.21 se observa la ventana del **Editor de Bloques**, donde cuenta con una serie de elementos entre los principales son los siguientes:

1. Bloques de uso general.
2. Bloques correspondientes a los elementos incluidos en la aplicación.

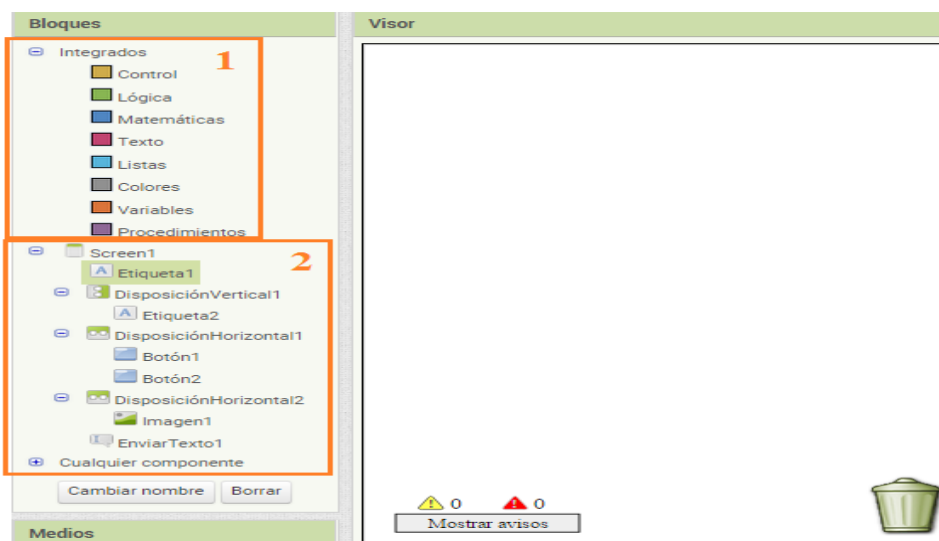


FIGURA N° 5.20.- Ventana del Editor de Bloques

## Paso 2: Programación de Botón 1 (Encendido - On).

Ir a **Bloques** y dar clic **Botón 1**, seleccionar el bloque necesitado y arrastrarlo a la ventana del Visor. Ver figura N°5.21.

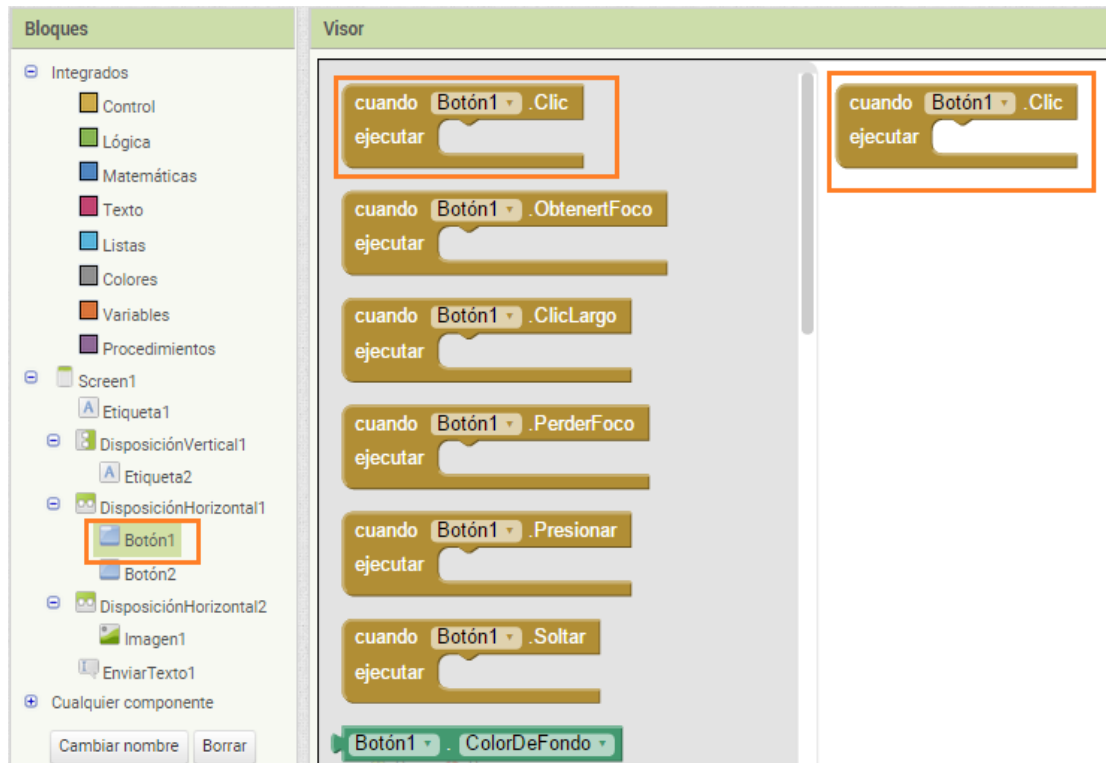


FIGURA N° 5.21.- Opciones del elemento botón

Después ir al elemento **EnviarTexto 1** y seleccionar los bloques como se observa en la figura N°5.22.

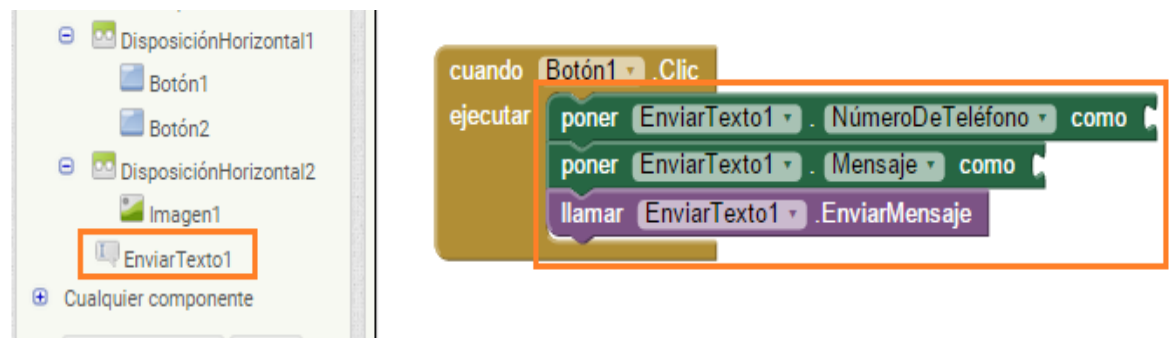


FIGURA N° 5.22.-Bloques de Enviar Texto

Se completa la programación para el envío de mensajes, con un bloque de texto donde indicará el número de teléfono a enviar el mensaje y el contenido.

Para esto ir al bloque **Texto** dar clic y seleccionar la primera opción. Para cambiar el contenido dar clic en el bloque y escribir el número de teléfono y el contenido del mensaje de texto. Ver figura N°5.23

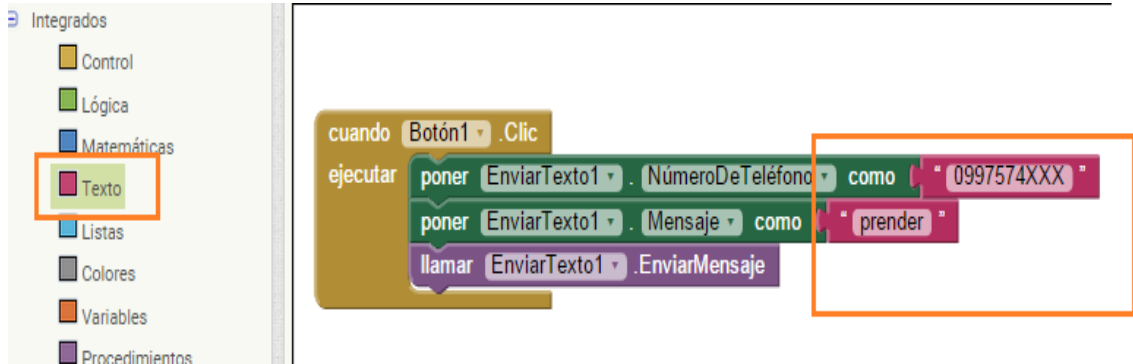


FIGURA N° 5.23.-Bloques de Texto para envío de sms.

Lo siguiente es situar una condición **IF** (si), para verificar que el botón 1 fue pulsado indicando con la imagen de la interfaz gráfica que fue encendido o apagado el foco en el caso de la presente aplicación.

Dirigirse a **Control** y seleccionar la primera opción. Ver figura N°5.24.

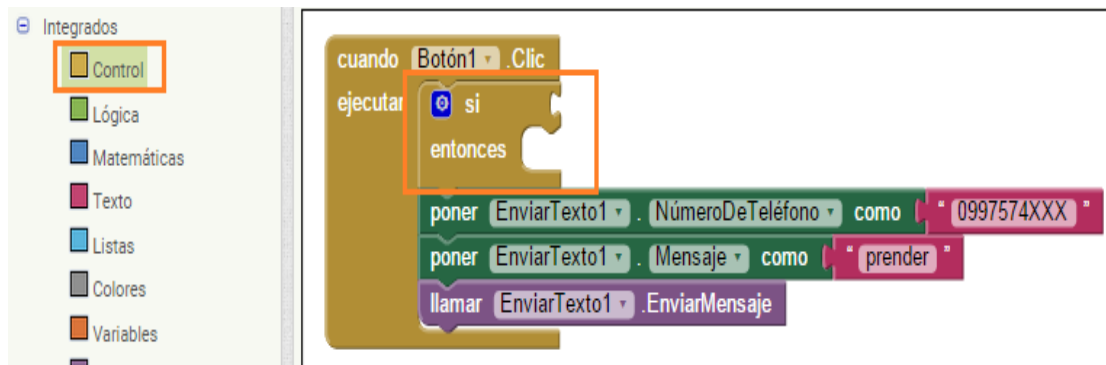


FIGURA N° 5.24.- Condición IF

Seleccionar la condición que debe cumplir, para esto ir a **Lógica** y escoger la condición igual y ubicar en la posición **SI**.



La idea es hacer un juego de cambio de imágenes para simular que se encienda o se apague el foco al presionar los botones. Para esto ir a **Bloques** dar clic en el elemento **Imagen1** y con lo bloques necesario de este elemento y la ayuda de los bloques de Texto, armarlos como se ve en la figura N°5.25.

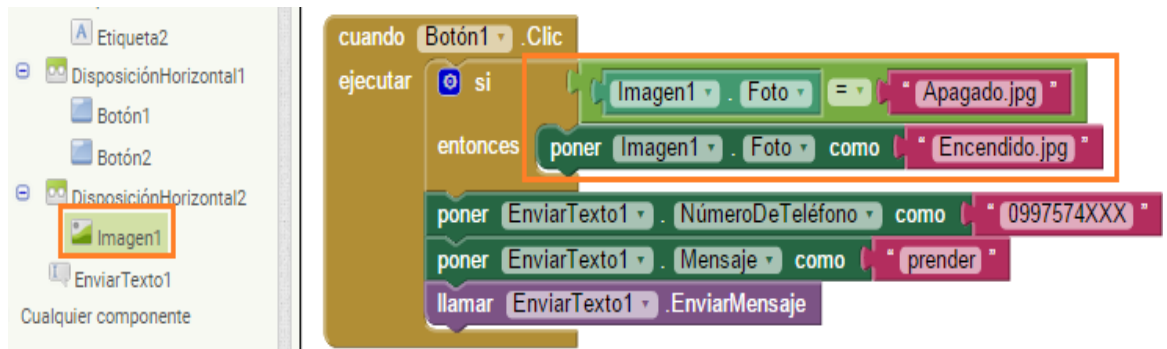


FIGURA N° 5.25.- Bloques de imagen en la programación.

### Paso 3: Programación de Botón 2 (Apagado – Off).

Para la configuración de Botón 2 realizar lo mismo que el Botón 1, con la diferencia en el juego de imágenes y el contenido del mensaje de texto. Ver figura N°5.26.



FIGURA N° 5.26.- Configuración de botón 2

### Paso 4: Recepción de mensajes de texto.

En esta parte se lee el mensaje de texto que llegue al teléfono celular para encender o apagar el foco, siempre y cuando lea una palabra especificada por parte del usuario.

Lo primero es ir a el elemento **EnviarTexto1**, donde se encuentra la opción recibir mensaje. Ver figura N°5.27

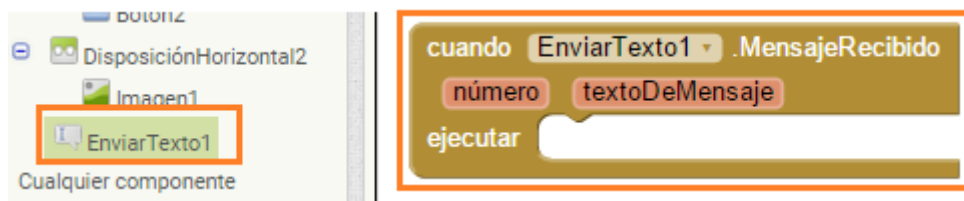


FIGURA N° 5.27.- Bloque recibir mensaje de texto

Después se debe ir al elemento **Etiqueta2**, escoger la opción como se ve en la figura N°5.28.

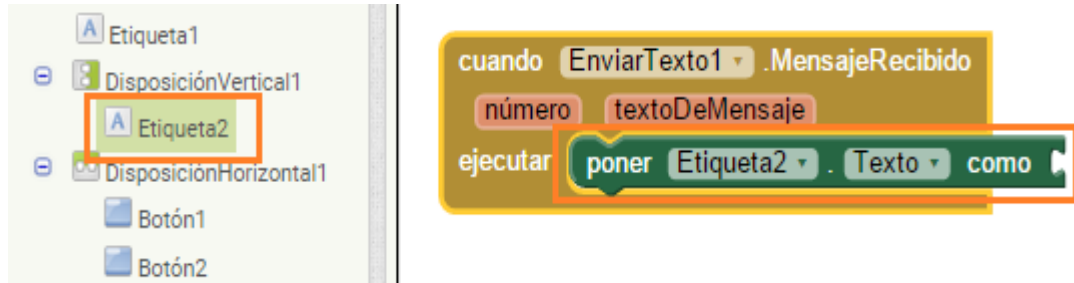


FIGURA N° 5.28.-Bloque de variable de Etiqueta2

Diríjase al elemento **Variable** y escoger el bloque **tomar**, desplegar la pestaña de este y seleccionar la opción **textoDeMensaje**. Ver figura N°5.29.



FIGURA N° 5.29.- Bloque de variable Tomar

El siguiente paso es agregar una condicional **IF ELSE**, para esto ir a **Control** y seleccionar la condición **SI**. Ver figura N°5.30.

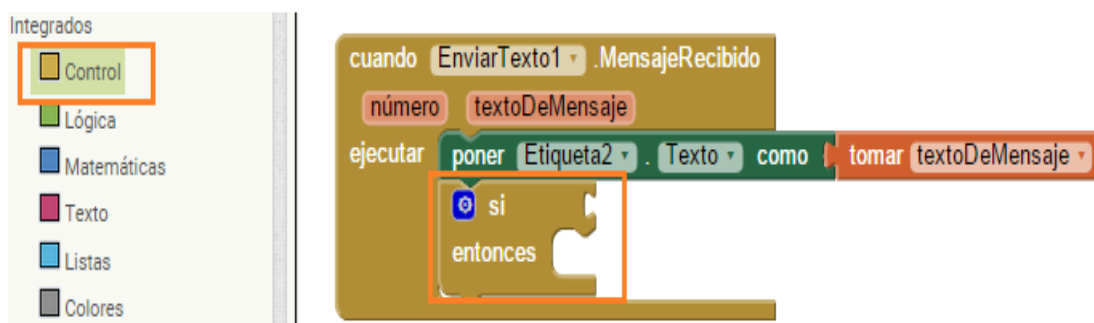


FIGURA N° 5.30.-Condición SI

Dar clic en la configuración de la condición **SI** y arrastrar el bloque **SI NO** a la mini ventana del elemento de la condición. Ver figura N°5.31.



FIGURA N° 5.31.- Configuración de condición IF

Ahora se diríjase al elemento **Texto** y agregar el siguiente bloque que se ve en la figura N°5.32, y colocarlo en la posición *SI*.

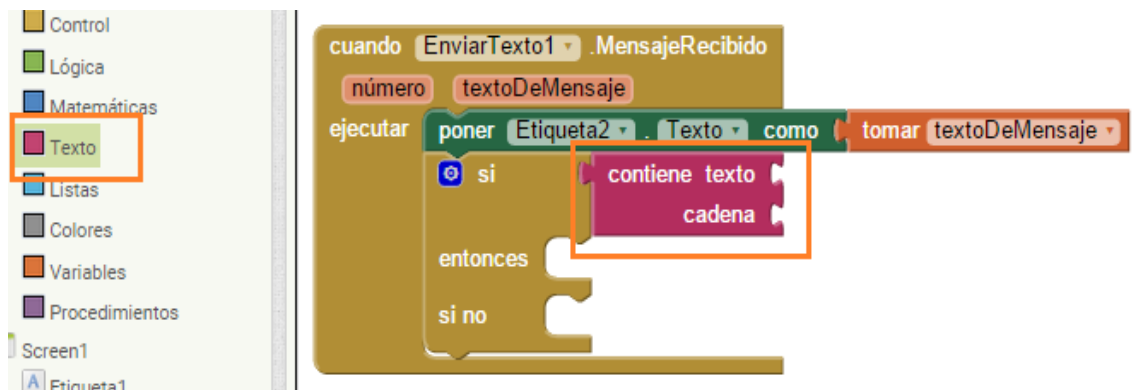


FIGURA N° 5.32.- Bloque contenido de Texto

Del elemento **Etiqueta2** y seleccionar el bloque que se ve en la figura N°5.33 que permitirá leer los mensajes de textos recibidos.

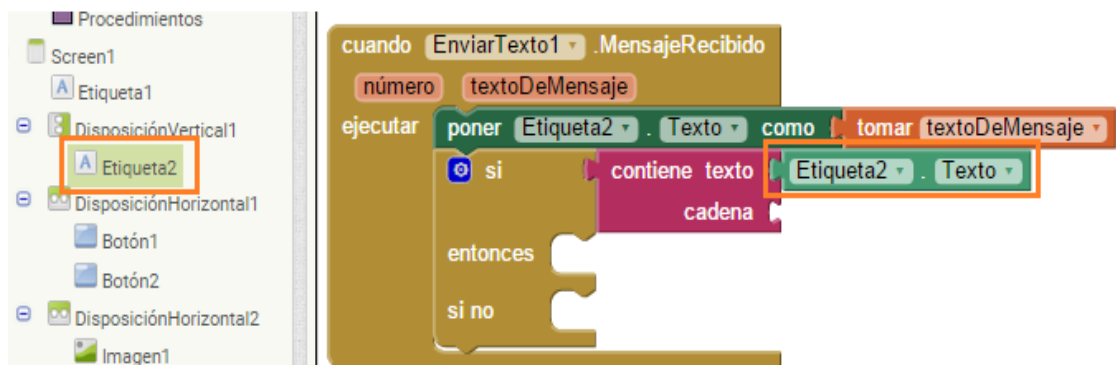


FIGURA N° 5.33.-Bloque de Etiqueta 2

El siguiente bloque de Texto permite extraer la palabra que desee leer de todo el mensaje de texto. Ver figura N°5.34.

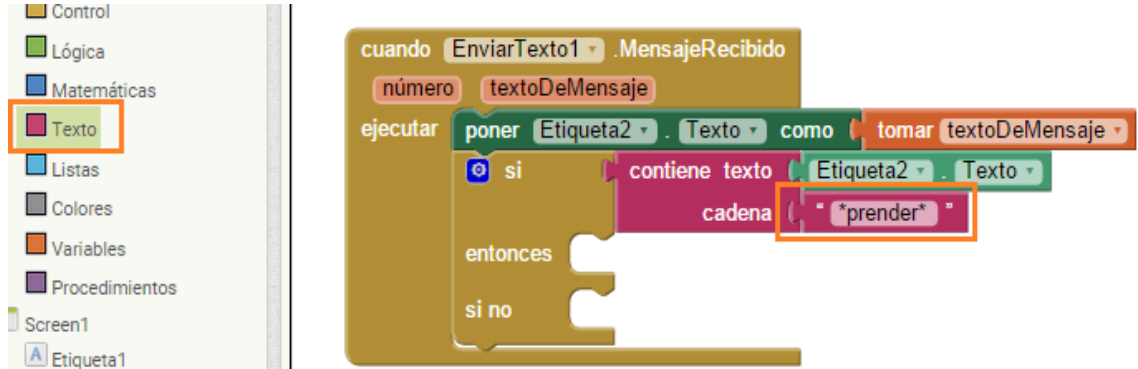


FIGURA N° 5.34.-Bloque de Texto para leer palabra del Mensaje de texto

Diríjase al elemento **Imagen1**, agregar el siguiente bloque que sirve para agregar una imagen. Arrastrarlo hasta la posición *Entonces*. Ver figura N°5.35.

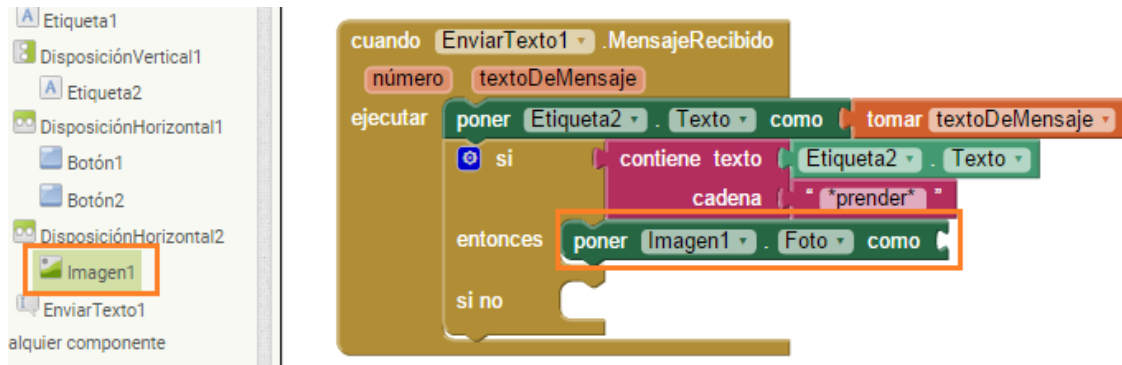


FIGURA N° 5.35.- Bloque de Imagen

Ingresar el nombre de la imagen con el bloque de Texto. Ver figura N°5.36.

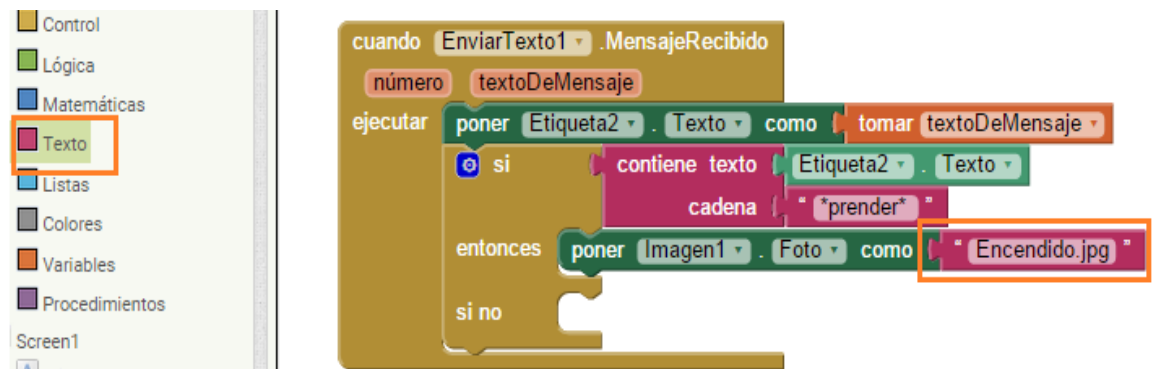


FIGURA N° 5.36.- Bloque de texto de la imagen a utilizar



A continuación se presenta el armado completo de los bloques a utilizar para la recepción y lectura de mensajes de texto. Ver figura N°5.37.

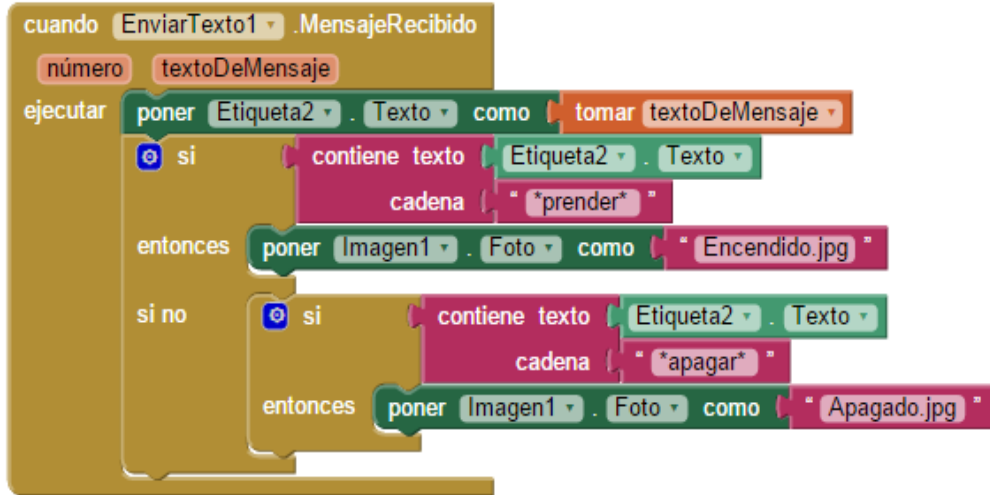


FIGURA N° 5.37.-Armado de bloques para recepción de mensajes de texto

## Parte 5: Carga de aplicación a teléfono celular.

### Paso 1: Generación de archivo .apk.

1. Dar clic en la pestaña Generar en la parte superior de la ventana. Ver figura N°5.38.

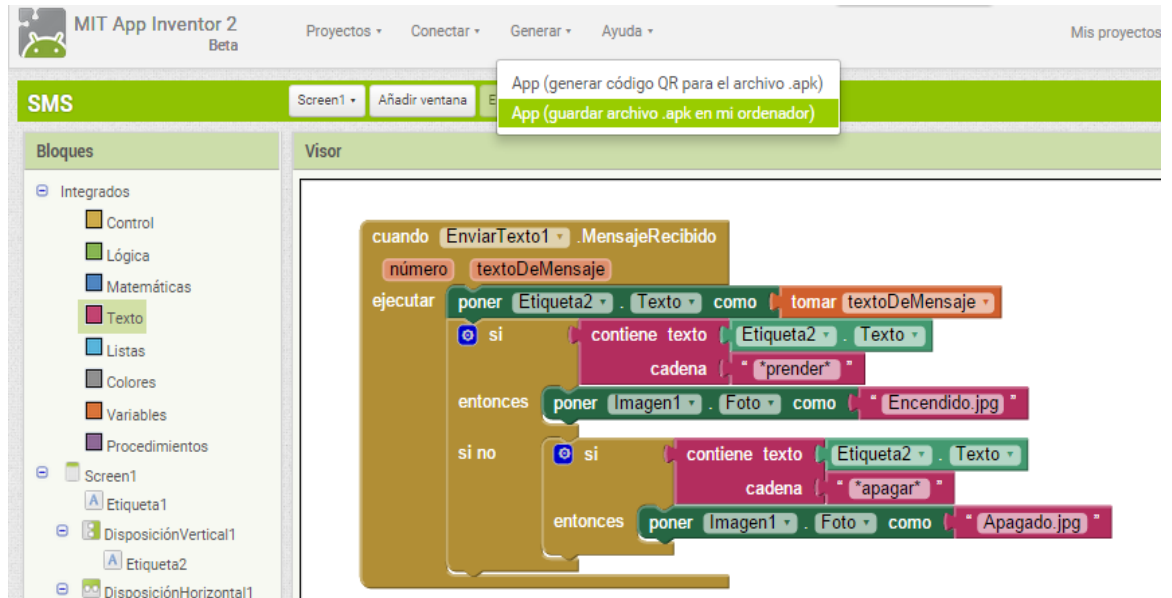


FIGURA N° 5.38.- Generar archivo .apk



2. Dar clic en App (**guardar archivo .apk en mi ordenador**) y se generará automáticamente una descarga de archivo .apk. Ver figura N°5.39 y N°5.40.

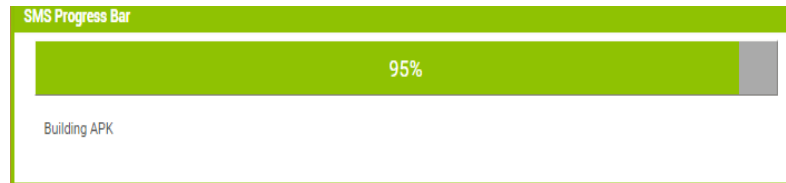


FIGURA N° 5.39.- Generación de archivo .apk

Buscar el archivo en la carpeta de descargas, y cargarlo en el teléfono celular. Ver figura N°5.41.

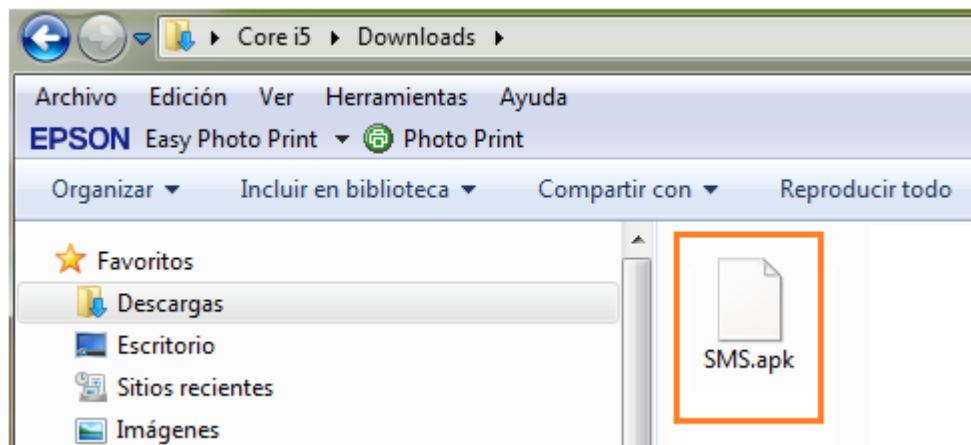


FIGURA N° 5.41.- Archivo .apk

3. Instalar el programa como cualquier aplicación desde la carpeta que se encuentre el archivo. Ver figura N°5.42.

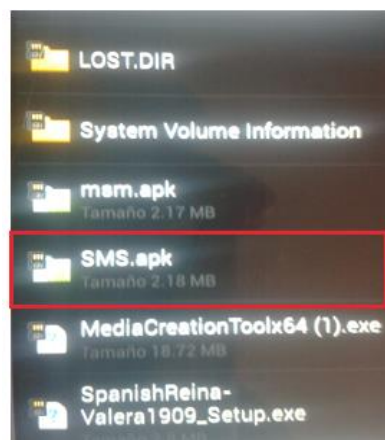


FIGURA N° 5.42.-Archivo .apk en el teléfono celular

4. En el organizador de aplicaciones aparecerá un nuevo icono. Ver figura N°5.43.



FIGURA N° 5.43.- Aplicación instalada

5. Abrir la aplicación y proceder a realizar los controles de la aplicación. Ver figura N°5.44.



FIGURA N° 5.44.- Aplicación en el teléfono celular

## Paso 2: Pruebas de la aplicación.

1. Presionar el botón **ON**, el foco se encenderá y enviará un mensaje de texto al número de teléfono preestablecido, con el contenido escrito previamente escrito. Ver figura N°5.45.



FIGURA N° 5.45.- Botón ON, encendido de foco y mensaje de texto en teléfono

2. Al presionar el botón OFF, el foco se apaga y de igual manera se envía un mensaje de texto. Ver figura N°5.46.

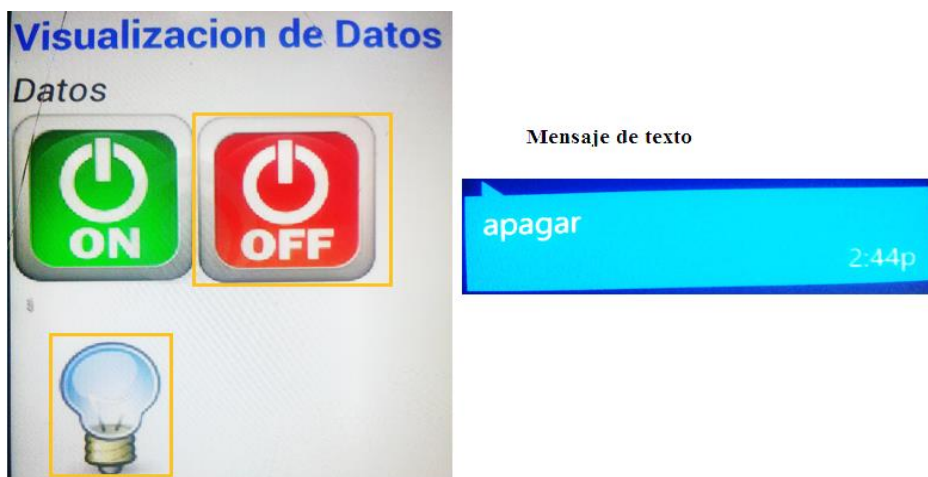


FIGURA N° 5.46.- Botón OFF, apagado de foco y mensaje en teléfono

3. Se puede controlar el foco con mensajes de texto enviados desde otro teléfono celular. Con un contenido **\*prender\*** el foco se enciende. Ver figura N°5.47.



FIGURA N° 5.47.-Mensaje de texto para encender foco.

4. Y con un contenido **\*apagar\*** el foco se apaga. Ver figura N°5.48.

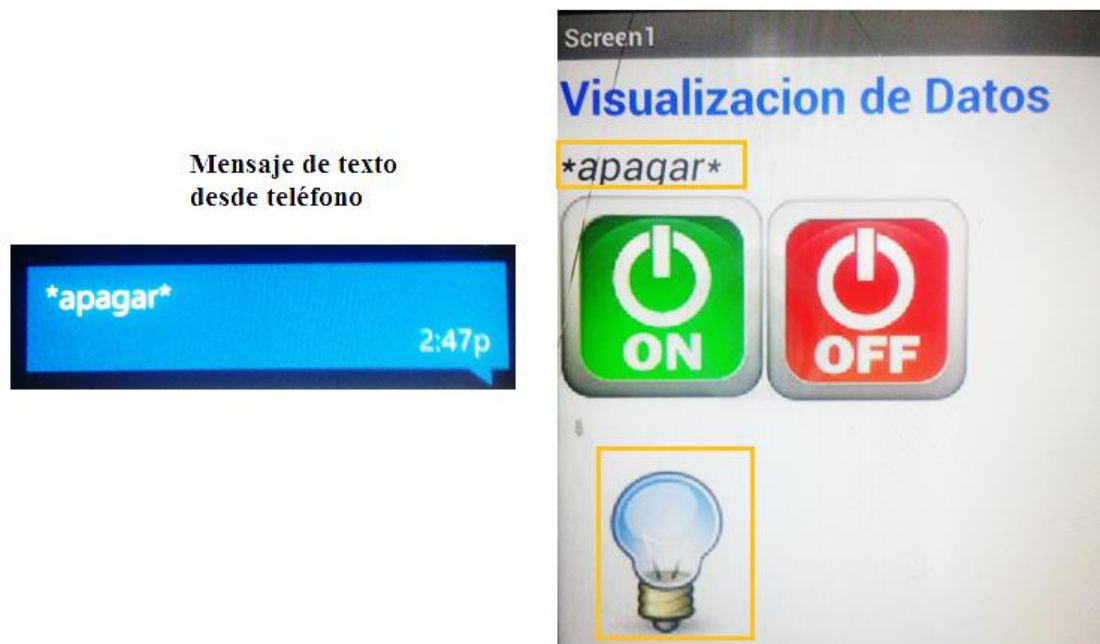


FIGURA N° 5.48.- Mensaje de texto para apagado de foco

## **ANEXO 4.- CONDICIONES CLIMÁTICAS ADECUADAS DEL GALPÓN DE AVES**

El objetivo es proporcionar a las aves un medio ambiente que le permita lograr el máximo rendimiento, velocidad de crecimiento óptima y uniforme, y buena eficiencia alimenticia con rendimiento en carne, asegurándonos de no afectar adversamente la salud ni el bienestar de las aves.

### **Condiciones Climáticas**

El clima se refiere a las condiciones medioambientales como temperatura, humedad, iluminación y ventilación que influye directamente sobre el galpón y sobre los animales ya que pueden ser fuente o causa de enfermedades y retrasos en la producción.

Los pollos a diferencia de otros animales, no están bien adaptados para producir calor, no pueden adaptarse rápidamente a cambios bruscos de temperatura, ni pueden comportarse bien si la temperatura ambiental es muy alta o muy baja.

#### **1. Temperatura Ambiental**

La temperatura ideal dentro del galpón es de 24 a 26°C. Cuando los pollos se someten a temperaturas mayores a 26°C aumenta el consumo de agua, disminuye el consumo de alimento y su habilidad de ganar peso. En regiones con temperaturas muy altas se deben tomar medidas para reducir la tensión por calor.

#### **2. Humedad**

La humedad relativa dentro del galpón depende del agua eliminada por los pollitos en la respiración y en las heces, del grado de humedad del medio ambiente de la temperatura dentro del galpón y de la ventilación. El punto óptimo de humedad relativa en el galpón varía entre el 50% y el 70%, sino se controla la humedad se pueden presentar los siguientes problemas:

- Humedades altas: Camas compactas, altos niveles de amoniaco, pollos muy sensibles al frio o calor y a las enfermedades (coccidiosis, enfermedades respiratorias).
- Humedades bajas: Camas polvosas, que también pueden ocasionar enfermedades de tipo respiratorio.

### **3. Ventilación**

El ambiente exterior debe tener una adecuada ventilación para así poder remover periódicamente el exceso de humedad, el gas carbónico producido por la respiración de las aves y de gas amoniaco producido por las heces. La ventilación natural del galpón va a estar determinada por la velocidad del viento, una buena ventilación debe llenar los dos requisitos:

- Que la velocidad del viento a nivel de las aves sea baja.
- Que la renovación del ambiente del galpón se efectúe por igual en toda el área.

### **4. Iluminación**

La recomendación general es suministrar 23 horas de luz continua, con una hora de oscuridad al día, que permita que los pollos se acostumbren a la oscuridad sorpresiva en caso de un apagón, para evitar mortandad por amontonamiento.

La intensidad de luz debe variar la edad del pollo, durante la primera semana debemos proveer 60 vatios por cada 20 metros cuadrados, para ir reduciendo la intensidad en forma tal que a los 21 días se tenga 15 vatios por cada 20 metros cuadrados para el resto del engorde.

Niveles muy altos de luminosidad aumentan la actividad de los pollos,, se producen problemas de canibalismo, magulladuras y ascitis aviar.

Tomado de: (termotasajero-FUDOC, 2014)

## ANEXO 5.- PROGRAMA GLOBAL DEL MICROCONTROLADOR PARA EL CONTROL DE TEMPERATURA DE UNA AVÍCOLA VÍA RED CELULAR

```

////////////////////////////////////
/// Requerimientos para sensor digital DHT11
sbit Data at RC0_bit;
sbit DataDir at TRISC0_bit;
unsigned short k;
/// Variables para procesos de Humedad y temperatura
unsigned short T_Byte1, T_Byte2, RH_Byte1, RH_Byte2;
char humed[] = "Humedad es 00.0 %";
/// Para sensores analógicos
unsigned int Radc0 = 0; // Guarda los datos de la lectura del AN0
unsigned int Radc1 = 0; // Guarda los datos de la lectura del AN1
unsigned int Radc2 = 0; // Guarda los datos de la lectura del AN2
/// Para sensores analógicos
float volt; // Variable para procesos de temperatura
unsigned float tempe; // Variable para procesos de temperatura
float vout, lux; // variables para sensor de luz
float sensorValue, Vo, Rs, Ro, ratio, PPM; // Variables para procesos sensor Gas

char temperatura[]="Temperatura=" ";
char luz[]="I-luz=" ";
char gas[]="GLP=" ";
/// Auxiliares para control de envio de SMS
int aux1=0, aux2=0, aux3=0, aux4=0;

////variables para recibir SMS
char Dato;
char Dato2;
char Bufer[255];
//char Bufer2[]="Ñ ";
unsigned short Pos=0;
unsigned short Bandera=0;
int i, j;

/// declaracion de la funcion de interrupcion
void interrupt( ){
//se evalua si la interrupcion disparada por recepcion serial
if(PIR1.F5){
//se lee el Dato de entrada
Dato=UART1_Read();
switch(Dato){
case 13: Pos=0; //se recibe el caracter enter
Bandera=1;
break;
case 10: break; //se recibe el retroceso del carro
default:Bufer[Pos++]=Dato; //se guarda los datos
//Pos++;
Bufer[Pos++]=0; //se establece el fin de cadena
break;
}
}
}
/// LECTURA DE SENSORES
/// SENSOR DIGITAL
void DHT11StartSignal(){
DataDir = 0;
Data = 0;
Delay_ms(25);
Data = 1;
Delay_us(30);
DataDir = 1;
}
///Chekea la respuesta de sensor
unsigned short DHT11CheckResponse(){
k = 150;
while(!Data){
Delay_us(2);
}
}

```

```

    k--;
    if(k<1) return 0; // Se acabó el tiempo
}
k = 150;
while(Data){
    Delay_us(2);
    k--;
    if(k<1) return 0; // Se acabó el tiempo
}
return 1;
}
////Lectura de sensor Digital DHT11
unsigned short DHT11ReadByte(){
    int i;
    unsigned short num = 0;
    DataDir = 1;
    for (i=0; i<8; i++){
        while(!Data);
        Delay_us(40);
        if(Data) num |= 1<<(7-i);
        while(Data);
    }
    return num;
}
////SENSORES ANALOGICOS
SAnalogicos(){
    ////TEMPERATURA - LM35
    Radc0=adc_read(0); // Lectura del Canal AN0
    volt=Radc0*0.48828125; //ecuacion 5 - tesis
    tempe=volt;
    FloatToStr(tempe,temperatura+12); // Conversion de tipo Float a Text
    temperatura[20]='°'; // Visualizacion del signo de Grados
    temperatura[21]='C'; // Visualizacion "C" para Celsius
    //////////////////////////////////////
    ////LUZ - LDR////
    Radc1=adc_read(1); // Lectura del Canal AN1
    vout=Radc1*0.0048828125; // Ecuacion 2 - tesis
    lux=(500*(5-vout))/(vout*10); // Ecuacion 9 - tesis
    FloatToStr(lux,luz+6); // Conversion de tipo Float a Text
    luz[14]='L'; // Visualizacion de L
    luz[15]='U'; // Visualizacion de U
    luz[16]='X'; // Visualizacion de X
    //////////////////////////////////////
    ////GAS -MQ3//
    Radc2 = adc_read(2); // Lectura del Canal AN2
    Vo=Radc2*0.0048828125; // Ecuacion 2 - tesis
    Rs=((5.00-Vo)/Vo); // Ecuacion 10 - tesis
    PPM=1000/Rs; // Ecuacion 13 - tesis
    FloatToStr(PPM,gas+4); // Conversion de tipo Float a Text
    gas[11]='P'; // Visualizacion de P
    gas[12]='P'; // Visualizacion de P
    gas[13]='M'; // Visualizacion de M
    //////////////////////////////////////
    return;
}
////PARA EMPAQUETAR TEXTO DE SMS
void Send_to_GSM(const char *s)
{
    // Enviar comando o cadena de datos
    while(*s) {
        UART1_Write(*s++);
    }
    // Terminacion por CR(Enter)
    //UART1_Write(0x0D);
    return;
}
////////////////////////////////////
void Send_to_GSM1( char *m)
{
    // Send command or data string
    while(*m) {
        UART1_Write(*m++);
    }
}

```



```

// Termination by CR(Enter)
  UART1_Write(0x0D);
  return;
}
//////////
void Send_to_GSM2( char *n)
{
// Send command or data string
  while(*n) {
    UART1_Write(*n++);
  }
// Termination by CR(Enter)
  UART1_Write(0x0D);
  return;
}
//////////
void Send_to_GSM3( char *p)
{
// Send command or data string
  while(*p) {
    UART1_Write(*p++);
  }
// Termination by CR(Enter)
  UART1_Write(0x0D);
  return;
}
//////////
void Send_to_GSM4( char *q)
{
// Send command or data string
  while(*q) {
    UART1_Write(*q++);
  }
// Termination by CR(Enter)
  UART1_Write(0x0D);
  return;
}
void Send_to_GSM5(const char *r)
{
// Send command or data string
  while(*r) {
    UART1_Write(*r++);
  }
// Termination by CR(Enter)
  UART1_Write(0x0D);
  return;
}
//////////Subrutina de envio de SMS//////////
void SMSenvia(){
  PORTB.F1=1; // Indicador de envio de comandos AT
  UART1_Write_Text("ATZ\r\n"); //Enviar cadena "ATZ"
  Delay_ms(500); // pequeño retraso 500milli
  UART1_Write_Text("ATZ\r\n"); //Enviar cadena "ATZ"
  Delay_ms(500); // pequeño retraso 500milli
  ///Desactivar eco de comandos
  UART1_Write_Text("ATE\r\n"); // Enviar "ATE" para desactivar el modo de eco
  Delay_ms(500); // pequeño retraso 500milli
  /// Establecer tipo de mensaje como TEXTO
  UART1_Write_Text("AT+CMGF=1\r\n"); //Cadena para establecer el modo de SMS a TEXTO
  Delay_ms(500); // pequeño retraso 500milli
  /// ENVIAR MENSAJE
  /// Enviar mensaje al número celular: 0123456789
  UART1_Write_Text("AT+CMGS=\"0997574792\"\r\n");//(Escriba su número de teléfono celular en lugar de
0123456789)
  Delay_ms(500); // pequeño retraso 500milli
  PORTB.F1=0; //Indicador de fin de comandos AT
  return;
}
void main()
{
//SE CONFIGURAN LOS PUERTOS
TRISB = 0; // Configuración del puerto B
TRISC.F1=0; // Configuración del puerto 1 de C

```

```

TRISC.F2=0;          // Configurcion del puerto 2 de C
PORTB=0b00000000;
PORTC.F1=0;
PORTC.F2=0;
TRISA = 0xFF;      // PORTA como entradas
ADCON1 = 0b1011;  // activacion de analogicos

//repcion serial
PIE1=0b00100000;
//se desactivan las demas fuentes de interrupcion
PIE2=0;
//se apagan las banderas de interrupcion
PIR2=0;
PIR1=0;
/////
ADC_Init();        // Iniciar modulo ADC
UART1_Init(9600); //inicio de modulo Usart
delay_ms(100);    // Retardo para estabilizar Modulos
//se activan las interrupciones globales y perifericas
INTCON=0b11000000;

PORTB.F0=1; //Indicador de funcinamiento de Mikro
//////////
Delay_ms(10000);
PORTB.F1=1; //Indicador de funcinamiento de Mikro
/////COMANDOS AT
UART1_Write_Text("ATZ\r\n"); // Enviar "AT" cadena luego esperar a que un pequeño retraso
Delay_ms(500);
UART1_Write_Text("ATZ\r\n"); // Enviar "AT" cadena luego esperar a que un pequeño retraso
Delay_ms(500);
// Desactivar el eco de comandos
UART1_Write_Text("ATE\r\n"); // Enviar "ATE" para desactivar el modo de eco
Delay_ms(500);
// Establecer tipo de mensaje como TEXTO
UART1_Write_Text("AT+CMGF=1\r\n"); // Enviar "AT + CMGF = 1" cadena para establecer el modo de SMS a
TEXTO
Delay_ms(500);
// recepcion de mensajes d texto
UART1_Write_Text("AT+CNMI=1,2,0,0,0\r\n"); // prepara par la recepcion mensajes
Delay_ms(500);
PORTB.F1=0; // Indicador de fin de comandos AT

////PRIMERA LECTURA DE SENSORES//
///Sensores Analogicos
SAnalogicos();
///CONTROL DE ENVIO DE SMS////////
///auxiliar SMS temperatura
if((tempe<=25)||(tempe>=35)){ //Rango de comparacion de temperatura
    aux1=0;
}
else{
    aux1=1;
}
///auxiliar SMS luz
if((lux<=90)||(lux>=300)){ //Rango de comparacion de luz
    aux2=0;
}
else{
    aux2=1;
}
///auxiliar SMS Gas
if((PPM<=300)||(PPM>=1000)){ //Rango de comparacion de Gas
    aux3=0;
}
else{
    aux3=1;
}
while(1)
{
    PORTB.F0=1; //Indicador de funcinamiento de Mikro
    Dato2=0;
    /// LECTURA DE SENSORES
    ///Sensor Digital

```

```

DHT11StartSignal();
if(!DHT11CheckResponse()) continue;
RH_Byte1 = DHT11ReadByte();
RH_Byte2 = DHT11ReadByte();
DHT11ReadByte(); /* Checksum */
// Set hum
humed[12] = RH_Byte1/10 + 48;
humed[13] = RH_Byte1%10 + 48;
humed[15] = RH_Byte2/10 + 48;
///

```

```

//gas
if((PPM>=300)&&(PPM<=1000)){
PORTB.F4=0;
if(aux3==0){
SMSenvia();
Send_to_GSM("Nivel de gas dentro de rango!!!\r\n");
SAnalogicos();
Send_to_GSM1(gas);
Delay_ms(200);
UART1_Write(0x1A);
UART1_Write(0x0D);
Delay_ms(1000);
aux3=1;
}
}
else{
PORTB.F4=1;
if(aux3==1){
SMSenvia();
Send_to_GSM5("!!!ALERTA!!!");
Send_to_GSM("Gas Fuera de rango!!!\r\n");
SAnalogicos();
Send_to_GSM1(gas);
Delay_ms(200);
UART1_Write(0x1A);
UART1_Write(0x0D);
Delay_ms(1000);
aux3=0;
}
}
/////////HUMEDAD
if ((RH_Byte1 > 45)&&(RH_Byte1 < 65)){
PORTB.F5 =0;
if(aux4==0){
SMSenvia();
Send_to_GSM("Humedad dentro de rango!!!\r\n");
Send_to_GSM1(humed);
Delay_ms(200);
UART1_Write(0x1A);
UART1_Write(0x0D);
Delay_ms(200);
aux4=1;
}
}
else{
PORTB.F5 =1;
if(aux4==1){
SMSenvia();
Send_to_GSM("Alerta - Humedad Fuera de rango!!!\r\n");
Send_to_GSM1(humed);
Delay_ms(200);
UART1_Write(0x1A);
UART1_Write(0x0D);
Delay_ms(200);
aux4=0;
}
}

////LEE SMS
//se evalua si una trama ha llegado
if(Bandera){
//aqui se debe analizar el contenido de la informacion del Bufer
for (i=0;i<=255;i++){
if(Bufer[i]=='*'){
j=i+2;
Dato2=Bufere[j];
UART1_Write(Dato2);
i=255;
}
}
Bandera=0; //se apaga la Bandera de llegada
}
}

```

```

//LEE DATO2 PARA ACTUADORES
//Primer actuador
if(Dato2=='Q'){ //ACTIVA 1
  PORTB.F6=1;
  SMSenvia();
  Send_to_GSM("A ACTIVADO VENTILADOR!!!\r\n");
  UART1_Write(0x0D);
  Send_to_GSM5("*V1*");
  Delay_ms(200);
  UART1_Write(0x1A);
  UART1_Write(0x0D);
  Delay_ms(1000);
  Dato2=0;
}
if(Dato2=='R'){ //DESACTIVA 1
  PORTB.F6=0;
  SMSenvia();
  Send_to_GSM("A DESACTIVADO VENTILADOR!!!\r\n");
  UART1_Write(0x0D);
  Send_to_GSM5("*V0*");
  Delay_ms(200);
  UART1_Write(0x1A);
  UART1_Write(0x0D);
  Delay_ms(1000);
  Dato2=0;
}
//Segundo Actuador
if(Dato2=='S'){ //ACTIVA 2
  PORTB.F7=1;
  SMSenvia();
  Send_to_GSM("A ACTIVADO CALEFACTOR!!!\r\n");
  UART1_Write(0x0D);
  Send_to_GSM5("*C1*");
  Delay_ms(200);
  UART1_Write(0x1A);
  UART1_Write(0x0D);
  Delay_ms(1000);
  Dato2=0;
}
if(Dato2=='T'){ //DESACTIVA 2
  PORTB.F7=0;
  SMSenvia();
  Send_to_GSM("A DESACTIVADO CALEFACTOR!!!\r\n");
  UART1_Write(0x0D);
  Send_to_GSM5("*C0*");
  Delay_ms(200);
  UART1_Write(0x1A);
  UART1_Write(0x0D);
  Delay_ms(1000);
  Dato2=0;
}
//Tercer Actuador
if(Dato2=='U'){ //ACTIVA 3
  PORTC.F1=1;
  SMSenvia();
  Send_to_GSM("A ACTIVADO LUZ!!!\r\n");
  UART1_Write(0x0D);
  Send_to_GSM5("*L1*");
  Delay_ms(200);
  UART1_Write(0x1A);
  UART1_Write(0x0D);
  Delay_ms(1000);
  Dato2=0;
}
if(Dato2=='V'){ //DESACTIVA 3
  PORTC.F1=0;
  SMSenvia();
  Send_to_GSM("A DESACTIVADO LUZ!!!\r\n");
  UART1_Write(0x0D);
  Send_to_GSM5("*L0*");
  Delay_ms(200);
  UART1_Write(0x1A);
  UART1_Write(0x0D);
}

```

```

    Delay_ms(1000);
    Dato2=0;
}
///Cuarto Actuador
if(Dato2=='W'){ //ACTIVA 4
    PORTC.F2=1;
    SMSenvia();
    Send_to_GSM("A ACTIVADO VENTILADOR 2!!!\r\n");
    UART1_Write(0x0D);
    Send_to_GSM5("*H1*");
    Delay_ms(200);
    UART1_Write(0x1A);
    UART1_Write(0x0D);
    Delay_ms(1000);
    Dato2=0;
}
if(Dato2=='X'){ //DESACTIVA 4
    PORTC.F2=0;
    SMSenvia();
    Send_to_GSM("A DESACTIVADO VENTILADOR 2!!!\r\n");
    UART1_Write(0x0D);
    Send_to_GSM5("*H0*");
    Delay_ms(200);
    UART1_Write(0x1A);
    UART1_Write(0x0D);
    Delay_ms(1000);
    Dato2=0;
}
///CONSULTA DE SENSORES
if(Dato2=='Z'){
    SMSenvia();
    Send_to_GSM("CONSULTA DE SENSORES!!!\r\n");
    Send_to_GSM4(humed);
    Send_to_GSM3(gas);
    Send_to_GSM2(luz);
    Send_to_GSM1(temperatura);
    Delay_ms(200);
    UART1_Write(0x1A);
    UART1_Write(0x0D);
    Delay_ms(1000);
    Dato2=0;
}
///CONSULTA DE SENSORES y ACTUADORES ////////////////
if(Dato2=='Y'){
    ///Primer caso
    if((PORTC.F2==0)&&(PORTC.F1==0)&&(PORTB.F7==0)&&(PORTB.F6==0)){
        SMSenvia();
        Send_to_GSM5("*H0*L0*C0*V0*\r\n");
        Send_to_GSM("CONSULTA DE SENSORES!!!\r\n");
        Send_to_GSM4(humed);
        Send_to_GSM3(gas);
        Send_to_GSM2(luz);
        Send_to_GSM1(temperatura);
        Delay_ms(200);
        UART1_Write(0x1A);
        UART1_Write(0x0D);
        Delay_ms(1000);
        Dato2=0;
    }
    ///Segundo caso
    if((PORTC.F2==0)&&(PORTC.F1==0)&&(PORTB.F7==0)&&(PORTB.F6==1)){
        SMSenvia();
        Send_to_GSM5("*H0*L0*C0*V1*\r\n");
        Send_to_GSM("CONSULTA DE SENSORES!!!\r\n");
        Send_to_GSM4(humed);
        Send_to_GSM3(gas);
        Send_to_GSM2(luz);
        Send_to_GSM1(temperatura);
        Delay_ms(200);
        UART1_Write(0x1A);
        UART1_Write(0x0D);
        Delay_ms(1000);
        Dato2=0;
    }
}

```

```

}
//Tercer caso
if((PORTC.F2==0)&&(PORTC.F1==0)&&(PORTB.F7==1)&&(PORTB.F6==0)){
  SMSenvia();
  Send_to_GSM5("*H0*L0*C1*V0*\r\n");
  Send_to_GSM("CONSULTA DE SENSORES!!!\r\n");
  Send_to_GSM4(humed);
  Send_to_GSM3(gas);
  Send_to_GSM2(luz);
  Send_to_GSM1(temperatura);
  Delay_ms(200);
  UART1_Write(0x1A);
  UART1_Write(0x0D);
  Delay_ms(1000);
  Dato2=0;
}
//Cuarto caso
if((PORTC.F2==0)&&(PORTC.F1==0)&&(PORTB.F7==1)&&(PORTB.F6==1)){
  SMSenvia();
  Send_to_GSM5("*H0*L0*C1*V1*\r\n");
  Send_to_GSM("CONSULTA DE SENSORES!!!\r\n");
  Send_to_GSM4(humed);
  Send_to_GSM3(gas);
  Send_to_GSM2(luz);
  Send_to_GSM1(temperatura);
  Delay_ms(200);
  UART1_Write(0x1A);
  UART1_Write(0x0D);
  Delay_ms(1000);
  Dato2=0;
}
//Quinto caso
if((PORTC.F2==0)&&(PORTC.F1==1)&&(PORTB.F7==0)&&(PORTB.F6==0)){
  SMSenvia();
  Send_to_GSM5("*H0*L1*C0*V0*\r\n");
  Send_to_GSM("CONSULTA DE SENSORES!!!\r\n");
  Send_to_GSM4(humed);
  Send_to_GSM3(gas);
  Send_to_GSM2(luz);
  Send_to_GSM1(temperatura);
  Delay_ms(200);
  UART1_Write(0x1A);
  UART1_Write(0x0D);
  Delay_ms(1000);
  Dato2=0;
}
//Sexto caso
if((PORTC.F2==0)&&(PORTC.F1==1)&&(PORTB.F7==0)&&(PORTB.F6==1)){
  SMSenvia();
  Send_to_GSM5("*H0*L1*C0*V1*\r\n");
  Send_to_GSM("CONSULTA DE SENSORES!!!\r\n");
  Send_to_GSM4(humed);
  Send_to_GSM3(gas);
  Send_to_GSM2(luz);
  Send_to_GSM1(temperatura);
  Delay_ms(200);
  UART1_Write(0x1A);
  UART1_Write(0x0D);
  Delay_ms(1000);
  Dato2=0;
}
//Septimo caso
if((PORTC.F2==0)&&(PORTC.F1==1)&&(PORTB.F7==1)&&(PORTB.F6==0)){
  SMSenvia();
  Send_to_GSM5("*H0*L1*C1*V0*\r\n");
  Send_to_GSM("CONSULTA DE SENSORES!!!\r\n");
  Send_to_GSM4(humed);
  Send_to_GSM3(gas);
  Send_to_GSM2(luz);
  Send_to_GSM1(temperatura);
  Delay_ms(200);
  UART1_Write(0x1A);
  UART1_Write(0x0D);
}

```

```

    Delay_ms(1000);
    Dato2=0;
}
//Octavo caso
if((PORTC.F2==0)&&(PORTC.F1==1)&&(PORTB.F7==1)&&(PORTB.F6==1)){
    SMSenvia();
    Send_to_GSM5("*H0**L1**C1**V1*\r\n");
    Send_to_GSM("CONSULTA DE SENSORES!!!\r\n");
    Send_to_GSM4(humed);
    Send_to_GSM3(gas);
    Send_to_GSM2(luz);
    Send_to_GSM1(temperatura);
    Delay_ms(200);
    UART1_Write(0x1A);
    UART1_Write(0x0D);
    Delay_ms(1000);
    Dato2=0;
}
//Noveno caso
if((PORTC.F2==1)&&(PORTC.F1==0)&&(PORTB.F7==0)&&(PORTB.F6==0)){
    SMSenvia();
    Send_to_GSM5("*H1**L0**C0**V0*\r\n");
    Send_to_GSM("CONSULTA DE SENSORES!!!\r\n");
    Send_to_GSM4(humed);
    Send_to_GSM3(gas);
    Send_to_GSM2(luz);
    Send_to_GSM1(temperatura);
    Delay_ms(200);
    UART1_Write(0x1A);
    UART1_Write(0x0D);
    Delay_ms(1000);
    Dato2=0;
}
//Decimo caso
if((PORTC.F2==1)&&(PORTC.F1==0)&&(PORTB.F7==0)&&(PORTB.F6==1)){
    SMSenvia();
    Send_to_GSM5("*H1**L0**C0**V1*\r\n");
    Send_to_GSM("CONSULTA DE SENSORES!!!\r\n");
    Send_to_GSM4(humed);
    Send_to_GSM3(gas);
    Send_to_GSM2(luz);
    Send_to_GSM1(temperatura);
    Delay_ms(200);
    UART1_Write(0x1A);
    UART1_Write(0x0D);
    Delay_ms(1000);
    Dato2=0;
}
//Decimo primer caso
if((PORTC.F2==1)&&(PORTC.F1==0)&&(PORTB.F7==1)&&(PORTB.F6==0)){
    SMSenvia();
    Send_to_GSM5("*H1**L0**C1**V0*\r\n");
    Send_to_GSM("CONSULTA DE SENSORES!!!\r\n");
    Send_to_GSM4(humed);
    Send_to_GSM3(gas);
    Send_to_GSM2(luz);
    Send_to_GSM1(temperatura);
    Delay_ms(200);
    UART1_Write(0x1A);
    UART1_Write(0x0D);
    Delay_ms(1000);
    Dato2=0;
}
//Decimo segundo caso
if((PORTC.F2==1)&&(PORTC.F1==0)&&(PORTB.F7==1)&&(PORTB.F6==1)){
    SMSenvia();
    Send_to_GSM5("*H1**L0**C1**V1*\r\n");
    Send_to_GSM("CONSULTA DE SENSORES!!!\r\n");
    Send_to_GSM4(humed);
    Send_to_GSM3(gas);
    Send_to_GSM2(luz);
    Send_to_GSM1(temperatura);
    Delay_ms(200);
}

```





## **ANEXO 6.- PRÁCTICAS PROPUESTAS PARA COMPROBAR EFICIENCIA DEL ENTRENADOR**

Para comprobar la eficiencia del entrenador se planteó las siguientes prácticas, y así tomar los tiempos de demora en su armado.

### **TEMA: USO DE TECLADO DE MATRIZ 4X4 Y VISUALIZACIÓN EN EL LCD DE CARACTERES.**

#### **Objetivos:**

- Armar una práctica el cual tiene un teclado de 4 filas y 4 columnas.
- Visualizar los caracteres en el LCD dependiendo de la tecla digitada en la matriz.

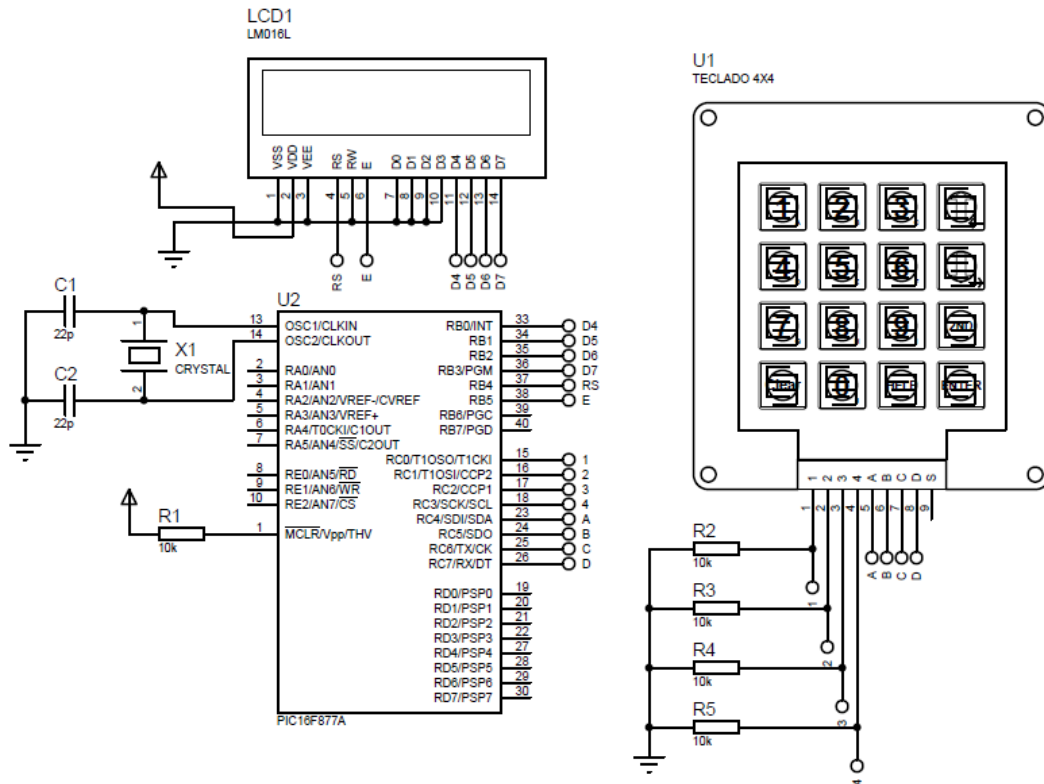
#### **Materiales:**

- 5 Resistencias de 10K o 4.7K ohm.
- 1 LCD de caracteres.
- 1 Teclado matricial 4x4.
- 1 Microcontrolador PIC 16F877A.
- 1 Cristal de cuarzo de 20 MHz
- 2 Condensadores de 22pF
- 1 Fuente de 5Vcc.
- 1 Protoboard.
- Cables para conexiones.

#### **Procedimiento:**

- 1) Para iniciar con la práctica se debe tener el circuito y de ser posible simular (*Proteus - ISIS*), para tener mayor confiabilidad del funcionamiento de la práctica.

En la siguiente imagen se puede ver el circuito que se va a construir.



Circuito 1.-Teclado y LCD

- 2) Realizar la programación del microcontrolador en el compilador Mikro C.  
 A continuación se encuentra el programa para el microcontrolador, compilado en MikroC:

```

////////////////////////////////////
char keypadPort at PORTC;
sbit LCD_RS at RB4_bit;
sbit LCD_EN at RB5_bit;
sbit LCD_D7 at RB3_bit;
sbit LCD_D6 at RB2_bit;
sbit LCD_D5 at RB1_bit;
sbit LCD_D4 at RB0_bit;

sbit LCD_RS_Direction at TRISB4_bit;
sbit LCD_EN_Direction at TRISB5_bit;
sbit LCD_D7_Direction at TRISB3_bit;
sbit LCD_D6_Direction at TRISB2_bit;
sbit LCD_D5_Direction at TRISB1_bit;
sbit LCD_D4_Direction at TRISB0_bit;

char LeerTeclado(void){
switch(Keypad_Key_Press()){
case 1: return '1';
case 2: return '2';
case 3: return '3';
case 4: return 'A';
case 5: return '4';
}
}

```

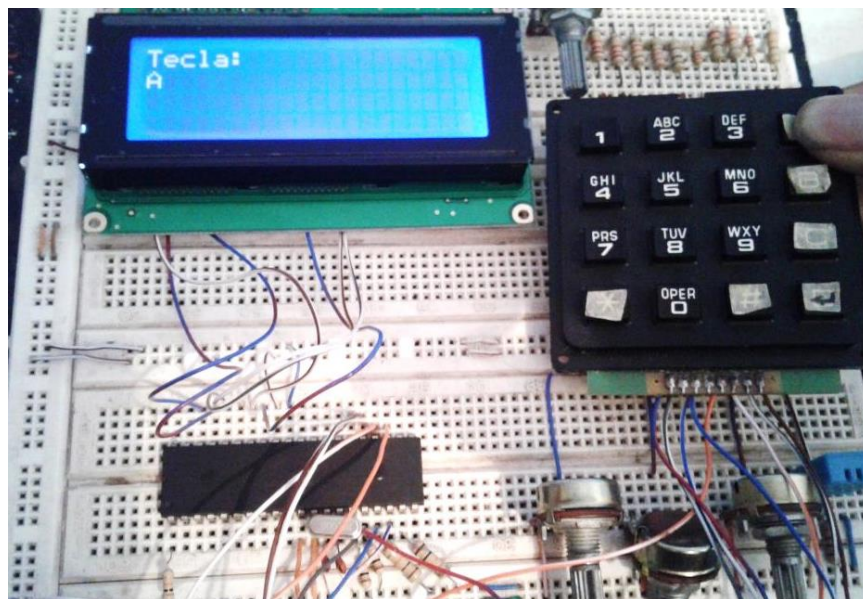
```

case 6: return '5';
case 7: return '6';
case 8: return 'B';
case 9: return '7';
case 10: return '8';
case 11: return '9';
case 12: return 'C';
case 13: return '*';
case 14: return '0';
case 15: return '#';
case 16: return 'D';
default: return 0;
}
}

void main(void){
char Tecla;
Lcd_Init();
Lcd_Cmd(_LCD_CURSOR_OFF);
Lcd_Out(1, 1, "Tecla:");
while(1){
    Tecla=LeerTeclado();
    Lcd_Chr(2, 1, Tecla);
}
}
/////////////////////////////////////////////////////////////////

```

- 3) Armar la práctica de acuerdo al **Circuito 1**, tomando en cuenta cada uno de los pines conectados al microcontrolador, los del LCD y del teclado.
- 4) Alimentar con 5 Vcc.
- 5) Digitar cada uno de las teclas para robar su funcionamiento.



**Conclusiones:**

- Se comprobó el funcionamiento del teclado matricial y su visualización de los caracteres en el LCD.
- El tiempo de armado fue extenso y la práctica fue un poco ambiguo hasta determinar cada uno de los pines de los elementos utilizados.

**TEMA: USO DE MÓDULO-TECLADO Y VISUALIZACIÓN EN EL MÓDULO- LCD DEL ENTRENADOR DE MICORONTROLADORES.****Objetivos:**

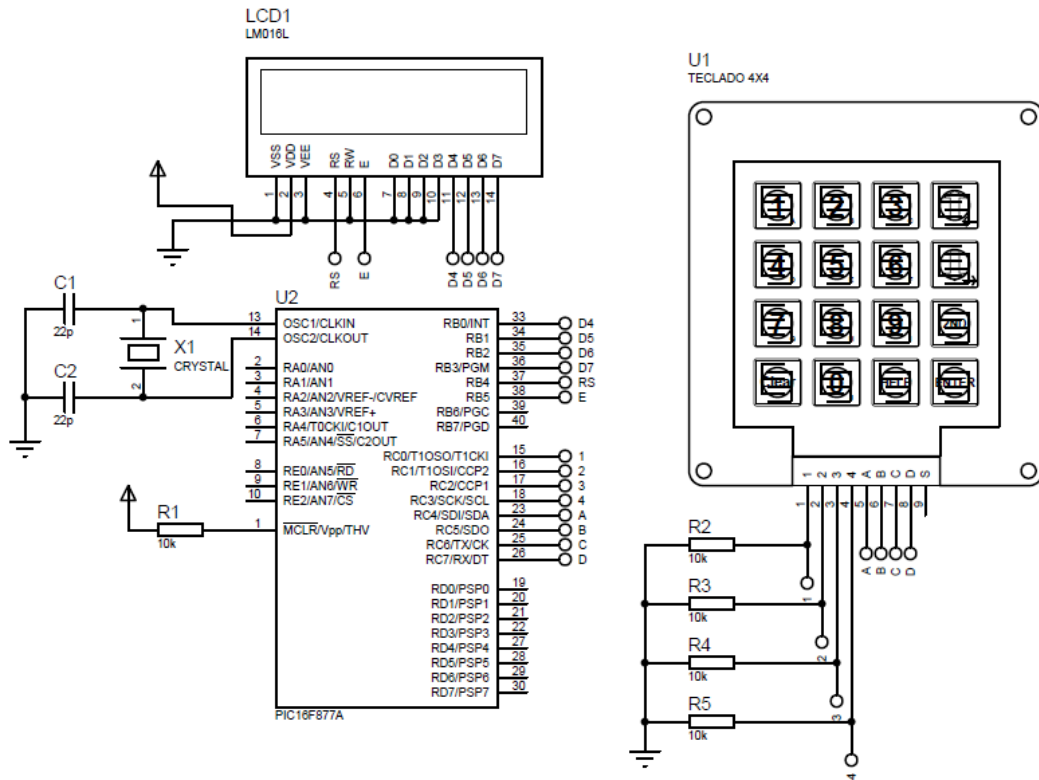
- Armar una práctica el cual está compuesto por el módulo de teclado y el módulo de LCD
- Comprobar el funcionamiento de los módulos y determinar la rapidez del armado.

**Materiales:**

- Entrenador de microcontroladores.
- 1 Módulo de teclado.
- 1 Módulo de LCD.
- 1 Microcontrolador PIC 16F877A.
- 1 Cristal de cuarzo de 20 MHz
- 2 Condensadores de 22pF
- Cables para conexiones.

**Procedimiento:**

- 1) Para iniciar con la práctica se debe tener el circuito y de ser posible simular (*Proteus - ISIS*). En la siguiente imagen se puede ver el circuito que se va a construir.



Circuito 2.-Teclado y LCD

- 2) Realizar la programación del microcontrolador en el compilador Mikro C.  
 A continuación se encuentra el programa para el microcontrolador, compilado en MikroC:

```

////////////////////////////////////
char keypadPort at PORTC;
sbit LCD_RS at RB4_bit;
sbit LCD_EN at RB5_bit;
sbit LCD_D7 at RB3_bit;
sbit LCD_D6 at RB2_bit;
sbit LCD_D5 at RB1_bit;
sbit LCD_D4 at RB0_bit;

sbit LCD_RS_Direction at TRISB4_bit;
sbit LCD_EN_Direction at TRISB5_bit;
sbit LCD_D7_Direction at TRISB3_bit;
sbit LCD_D6_Direction at TRISB2_bit;
sbit LCD_D5_Direction at TRISB1_bit;
sbit LCD_D4_Direction at TRISB0_bit;

char LeerTeclado(void){
switch(Keypad_Key_Press()){
case 1: return '1';
case 2: return '2';
case 3: return '3';
case 4: return 'A';
case 5: return '4';
case 6: return '5';

```

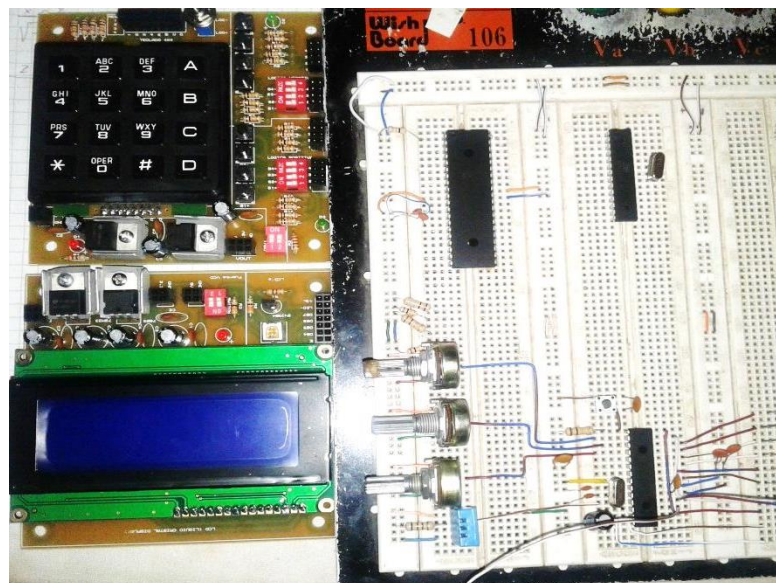
```

case 7: return '6';
case 8: return 'B';
case 9: return '7';
case 10: return '8';
case 11: return '9';
case 12: return 'C';
case 13: return '*';
case 14: return '0';
case 15: return '#';
case 16: return 'D';
default: return 0;
}
}

void main(void){
char Tecla;
Lcd_Init();
Lcd_Cmd(_LCD_CURSOR_OFF);
Lcd_Out(1, 1, "Tecla:");
while(1){
    Tecla=LeerTeclado();
    Lcd_Chr(2, 1, Tecla);
}
}
////////////////////////////////////

```

- 3) Armar la práctica de acuerdo al **Circuito 2**, tomando en cuenta cada uno de los pines conectados al microcontrolador.
- 4) Conectar el módulo de teclado desde sus pines hacia el microcontrolador.
- 5) De igual manera enlazar el módulo de LCD para la visualización.
- 6) Encender la fuente del entrenador.
- 7) Digitar cada uno de las teclas para robar su funcionamiento.



**Conclusiones:**

- Se comprobó el funcionamiento de los módulos de teclado y su visualización de los caracteres en el módulo de LCD.
- El tiempo de armado fue rápida y la práctica resulta fácil porque cada uno de los pines de los elementos utilizados se los ubica fácilmente en los módulos.