

UNIVERSIDAD TÉCNICA LATINOAMERICANA

Facultad de Ingenierías

Escuela de Ingeniería Electrónica



“FORMULACIÓN, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO DE TEMPERATURA Y HUMEDAD RELATIVA; Y EL CONTROL DE LUCES EN EL LABORATORIO DE ELECTRÓNICA DE LA UNIVERSIDAD TÉCNICA LATINOAMERICANA”.

**PARA OPTAR AL GRADO DE:
INGENIERO EN ELECTRÓNICA**

PRESENTAN:

MANUEL DE JESÚS GÁMEZ LÓPEZ

JAIME ALEJANDRO OCHOA RIVAS

JOSÉ ENMANUEL URÍAS GUZMÁN

SANTA TECLA, OCTUBRE 2020



AUTORIDADES UNIVERSITARIAS.

ING. ROSENDO MAURICIO SERMEÑO PALACIOS.
RECTOR.

ING. FRANCISCO ALFREDO CARRILLO LARREYNAGA.
VICE-RECTOR.

ING. ROOSELVET ADOLFO OSORIO.
SECRETARIO GENERAL.

LIC. CARLOS MAURICIO MOLINA RENDEROS.
FISCAL.

ING. MAURICIO ANTONIO AGUIRRE ORELLANA.
DECANO DE LA FACULTAD DE INGENIERÍA.

SANTA TECLA, OCTUBRE 2020.



Asesor del documento de Tesis

Ing. Eduardo Antonio Fermán Arana

Jurado Evaluador

Ing. Gustavo Raul Alfaro.

Ing. Armando Antonio Vividor Rivas.

Inga. Ana Gladis Revelo Verdi.

SANTA TECLA, OCTUBRE 20

**ESPACIO PARA ACTAS DE APROBACIÓN
DE GRADO DE CADA ALUMNO.**

**ESPACIO PARA ACTAS DE APROBACIÓN
DE GRADO DE CADA ALUMNO.**

**ESPACIO PARA ACTAS DE APROBACIÓN
DE GRADO DE CADA ALUMNO.**

AGRADECIMIENTOS DE JAIME ALEJANDRO OCHOA RIVAS

A mi madre Mirian Ilbea de Ochoa. Por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien, pero más que nada, por su amor.

A mi padre Gilberto Ochoa Ramírez. Por los ejemplos de perseverancia y constancia que lo caracterizan y que me ha infundado siempre, por el valor mostrado para salir adelante y por su amor.

Mis hermanas Beatriz Rivas y Wendy Jassmine Rivas, el ánimo, apoyo y alegría que me brindan me dan la fortaleza necesaria para seguir adelante.

Mariel Hernández Ramos que, desde el primer momento, colaboración y cariño sin ningún interés, son las personas por las cuales por hoy puedo afirmar que, porque ellos han estado a mi lado cada día

Ing. Eduardo Antonio Fermán Arana por su gran apoyo y motivación para la culminación de nuestros estudios profesionales y para la elaboración de esta tesis; por su tiempo compartido y por impulsar el desarrollo de nuestra formación profesional.

AGRADECIMIENTOS DE MANUEL DE JESÚS GÁMEZ LÓPEZ

Quiero dedicar este trabajo primeramente a Dios por darme la sabiduría necesaria para llevarlo a cabo y permitirme llegar hasta este momento.

A mí amada familia: María Brendalí Alvarado, Jhoseline Alejandra Gámez Alvarado, José Manuel Gámez Alvarado y José Daniel Gámez Alvarado por convertirse en mi principal motivación de superación personal. Gracias por estar para mí y apoyarme de diferentes maneras para seguir adelante como familia, cosa que agradezco y reconozco que ha sido una gran ayuda para poder llegar a la meta. ¡Gracias!

A mi madre por su infinito amor, cariño y apoyo, porque sé que muy a pesar de cualquier cosa siempre está conmigo, orando y pidiendo a nuestro Dios porque me ayude y de sabiduría para realizar cualquier proyecto que me proponga. ¡Infinitas Gracias!

A mi padre donde quiera que se encuentre porque sé que siempre está conmigo. La persona que siempre me aconsejó para que me preparara y llegará a ser un profesional. Sus consejos y su lucha por ayudarme en lo que podía los agradezco desde lo más profundo de mi ser. Ahora puedo decir padre “meta lograda”. ¡Infinitas Gracias!

A mis amigos/as por apoyarme y hacerme sonreír en los momentos difíciles, momentos duros que a veces a uno lo bloquean y siente que ya no puede más. ¡Gracias!

A mis maestros de la universidad, quienes a lo largo de cinco años compartieron sus conocimientos conmigo y me dotaron de habilidades y destrezas para enfrentarme al mundo laboral que desde hoy se abre en mi vida; a aquellos maestros que me ofrecieron también su amistad, sus horas extras para la comprensión de temas y contenidos difíciles, a ellos mil gracias por regalarme parte de su tiempo. ¡Infinitas gracias!

Y a todas las personas que, con su amor, cariño, comprensión, conocimiento y apoyo colaboraron de alguna manera para ayudarme en la finalización de este trabajo. ¡Gracias!

AGRADECIMIENTOS DE JOSÉ ENMANUEL URIAS GUZMAN

Primeramente, a Dios por haberme permitido estudiar; poder tener el tiempo para hacerlo y la sabiduría para concretar mis estudios.

A mi familia esposa e hijos por todo el apoyo brindado durante todos estos años por el esfuerzo, dedicación, cariño que me ayudaron para seguir estudiando; por el tiempo libre dedicado a mis estudios y sacrificado para con mis hijos gracias por el apoyo.

Gracias a mis padres y hermanos por el apoyo brindado durante todo este proceso por siempre incentivar me a estudiar.

Gracias a mis amigos/as por la motivación a seguir estudiando y por apoyarme.

Gracias a todos los maestros de la universidad por sus conocimientos transferidos por todas las arduas horas de enseñanza y la dedicación puesta en nosotros. A Dios. Por haberme permitido llegar hasta este punto y haberme dado salud para lograr mis objetivos, además de su infinita bondad y amor.

INDICE

INDICE DE FIGURAS.....	15
INDICE DE TABLAS	21
INTRODUCCIÓN.....	22
DESCRIPCIÓN DEL PROBLEMA.....	23
ALCANCES Y LIMITACIONES.....	24
ALCANCES.	24
LIMITANTES.....	25
REQUERIMIENTOS PARA LA CORRECTA FUNCIONALIDAD DEL PRESENTE PROYECTO.....	26
OBJETIVOS.....	27
OBJETIVO GENERAL	27
OBJETIVO ESPECÍFICO.....	27
JUSTIFICACIÓN	28
CAPÍTULO I.....	29
1. MARCO TEÓRICO	30
1.1. CONTROLADOR ARDUINO	30
1.2. CARACTERÍSTICAS TÉCNICAS DE ARDUINO UNO R3	31
1.3. CARACTERÍSTICAS TÉCNICAS DE LA TARJETA ARDUINO MEGA. ...	35
1.4. SENSORES	37
1.5. TEMPERATURA Y HUMEDAD	37
1.6. SISTEMA OPERATIVO ANDROID	40
1.7. COMPARACIÓN Y ELECCIÓN.....	41
1.8. ELECCIÓN DEL SISTEMA OPERATIVO ANDROID	43
1.9. ESTRUCTURA ANDROID	45
1.10. PROGRAMACIÓN Y DESARROLLO ANDROID	48
1.11. EVOLUCIÓN DE LOS SISTEMAS OPERATIVOS ANDROID	51
1.12. HERRAMIENTAS DE DESARROLLO EN EL IDE DE ANDRO STUDIO.....	56
1.13. REQUERIMIENTOS DE INSTALACIÓN DEL ENTORNO DE TRABAJO CON IDE ANDROID STUDIO	58
1.14. PLATAFORMA WEB (VIRTUAL).....	59

1.15. SERVICIO WEB (WEB SERVICE)	60
1.16. IMPORTANCIA DE UN WEB SERVICE	62
1.17. TECNOLOGÍAS EN UN WEB SERVICE	64
1.18. API.....	70
1.19. EJEMPLOS DE COMO FUNCIONAN LAS API.....	71
1.20. API vs SERVICIO WEB.....	73
1.21. RAZONES PARA CREAR SERVICIOS WEB.....	74
1.22. RESUMEN API vs WEB SERVICE	74
1.23. BASE DE DATOS.....	75
1.24. DIAGRAMA DE BASE DE DATOS.....	77
1.25. TARJETAS ELECTRÓNICAS DE PRUEBAS PARA CONEXIÓN DEL CONTROLADOR ARDUINO AL EXTERIOR (INTERNET).....	77
1.26. PRINCIPALES CONTROLADORES PROBADOS CON ARDUINO PARA ESTABLECER COMUNICACIÓN CON EL EXTERIOR (INTERNET).....	78
1.27. SHIELD ETHERNET W5100	79
1.27.1. CARACTERÍSTICAS DE SHIELD ETHERNET W5100	80
1.28. ARDUINO ETHERNET SHIELD - W5200.....	81
1.28.1. CARACTERÍSTICAS DE ARDUINO ETHERNET SHIELD - W5200	82
1.29. ARDUINO ETHERNET SHIELD 2 – W5500	82
1.29.1. CARACTERÍSTICAS DE ARDUINO ETHERNET SHIELD - W5500	82
1.30. VARIANTES ARDUINO ETHERNET SHIELD Rev3 CON PoE.....	81
1.30.1. CARACTERÍSTICAS DE VARIANTES ARDUINO ETHERNET SHIELD Rev3 CON PoE.	84
1.31. DETALLES DE USO DEL WIZNET W5100.....	84
1.32. PLACAS DE RED (WIFI)	87
1.32.1. CARACTERÍSTICAS DE PLACAS DE RED (WIFI).....	88
1.33. TECNOLOGÍA GSM/GPRS	89
1.34. GSM (GROUPE SPECIAL MOBILE)	89
1.35. TECNOLOGÍA 5G	92
1.35.1. CARÁCTERISTICAS DE TECNOLOGIA 5G.	93
1.36. GPRS (SERVICIO GENERAL DE PAQUETES DE RADIO).....	94
1.37. SHIELD GSM/GPRS SIM900.....	95
1.37.1. CARACTERÍSTICAS DE GPRS	97

1.37.2. COMANDOS AT PARA CONEXIÓN TCP/UDP	98
CAPÍTULO II	99
2. DESCRIPCIÓN FUNCIONAL DEL SISTEMA.....	100
2.1. ENTRADAS Y SALIDAS DEL SISTEMA ANDROID	100
2.2. INTRODUCCIÓN	100
2.3. SENSOR DE TEMPERATURA Y HUMEDAD RELATIVA	100
2.4. SENSOR LDR (Light-Dependent Resistor).....	104
2.5. SENSOR DE CORRIENTE SCT013-30.....	106
2.6. ADAPTADOR PARA SENSOR SCT013-30	109
2.7. SENSOR DE CORRIENTE SCT013-100.....	111
2.8. CIRCUITO ACONDICIONADOR DE SEÑAL	112
2.9. CONEXIÓN SCT – 013 CON ARDUINO	114
2.10. CODIGO FUENTE PARA LECTURA BÁSICA DE LA CORRIENTE	116
2.11. ACTUADORES.....	117
2.12. INTERFAZ ELECTRÓNICA DE CONTROL DE CARGAS.....	117
2.13. MÓDULO RELÉ HL-52S.....	117
2.14. ESQUEMA DEL CIRCUITO.....	119
2.15. USO DEL MÓDULO RELÉ HL-52S CON CARGA DE ALTA TENSIÓN .	121
2.16. SKETCH ARDUINO PARA CONTROL DE CARGAS DE ALTA TENSIÓN CON MÓDULO RELÉ	122
2.17. RESUMÉN DEL CAPÍTULO	123
CAPÍTULO III	124
3. APP DEL PROYECTO	125
3.1. APP.....	125
3.2. CARACTERÍSTICAS DE LA APP	127
3.3. INTERFACES GRÁFICAS DE LA APP	128
3.4. DESCRIPCIÓN DE LAS PRINCIPALES PANTALLAS DE LA APP	128
3.5. SEGURIDAD DEL SISTEMA EN GENERAL.....	132
3.6. RESUMÉN DEL CAPÍTULO.....	157
CAPÍTULO IV	158

4. PLATAFORMA WEB, WEB SERVICE Y BASE DE DATOS	159
4.1. BREVE INTRODUCCIÓN	159
4.2. PLATAFORMA WEB.....	159
4.3. LENGUAJES DE PROGRAMACIÓN.....	159
4.4. SERVIDOR WEB.....	160
4.5. WEB SERVICE	161
4.5.1. BREVE INTRODUCCIÓN	161
4.5.2. LENGUAJES DE PROGRAMACIÓN.....	161
4.5.3. FUNCIONALIDAD DEL WEB SERVICE EN EL SISTEMA	161
4.6. BASE DE DATOS.....	162
4.6.1. BREVE INTRODUCCIÓN	162
4.6.2. LENGUAJE ESTRUCTURADO DE CONSULTA (SQL).....	162
4.6.3. SISTEMAS GESTORES DE BASE DE DATOS	162
4.6.4. SOFTWARE PARA DISEÑO DE BASE DE DATOS	163
4.6.5. CREACIÓN DE LA BASE DE DATOS	165
4.6.6. DIAGRAMAS DE LA BASE DE DATOS.....	165
4.6.7. PROTOTIPO Y DISEÑO ELECTRÓNICO.....	167
4.6.8. CÓDIGO FUENTE PARA CONTROL DE SHIELD ETHERNET CON ARDUINO MEGA.....	168
4.7. RESUMÉN DEL CAPÍTULO.....	170
CAPÍTULO V.....	171
5. SISTEMA DE MONITOREO Y AUTOMATIZACIÓN	172
5.1. BREVE INTRODUCCIÓN	172
5.2. MONITORIZACIÓN DE PARÁMETROS ENTRADAS Y SALIDAS	172
5.3. ENTRADAS DEL SISTEMA.....	172
5.4. SENSOR DE MEDICIÓN DE TEMPERATURA Y HUMEDAD RELATIVA.....	173
5.5. SENSOR DEPENDIENTE DE LUZ (LDR)	174
5.6. PROCESAMIENTO DE LA INFORMACIÓN DE LOS SENSORES	174
5.7. SALIDAS DEL SISTEMA	175
5.8. DISEÑO DEL CIRCUITO	177

5.9. SOFTWARE DE MONITORIZACIÓN Y AUTOMATIZACIÓN (SKETCH).....	179
5.10. MONTAJE FÍSICO DE PRUEBAS DEL SISTEMA DE MONITORIZACIÓN Y AUTOMATIZACIÓN	182
5.11. ESQUEMA ELECTRÓNICO DEL SISTEMA	184
5.12. RESUMÉN DEL CAPÍTULO	185
CAPÍTULO VI.....	186
6. COMUNICACIÓN DEL SISTEMA.....	187
6.1. BREVE INTRODUCCIÓN	187
6.2. SHIELD O ESCUDO SIM900 GSM/GPRS.....	187
6.3. CARACTERÍSTICAS DE HARDWARE Y SOFTWARE DE LA TARJETA.....	189
6.4. FORMAS DE EJECUCIÓN DE COMANDOS AT EN LA SHIELD SIM900.....	194
6.5. PRUEBAS EXPERIMENTALES CON EL PROTOTIPO	196
6.6. VERIFICACIÓN DE LA COMUNICACIÓN	196
6.7. SKETCH BÁSICO PARA PRUEBAS DE ENVÍO DE COMANDOS AT POR TECLADO DESDE MONITOR SERIAL	197
6.8. SKETCH BÁSICO PARA REALIZAR PRUEBA DE LLAMADA CON ARDUINO Y COMANDOS AT.....	198
6.9. SKETCH BÁSICO PARA REALIZAR PRUEBA DE ENVIO DE SMS CON ARDUINO Y COMANDOS AT.....	199
6.10. SOFTWARE DE COMUNICACIONES	199
6.11. ESQUEMA DE FUNCIONAMIENTO DEL SISTEMA.....	200
6.12. PROTOTIPO Y SKETCH.....	201
6.13. SKETCH BÁSICO PARA REALIZAR PRUEBA DE CONEXIÓN A INTERNET POR GPRS A TRAVES DE ARDUINO Y COMANDOS AT.....	203
6.14. RESUMÉN DEL CAPÍTULO	206
CONCLUSIONES	208
RECOMENDACIONES	209
BIBLIOGRAFÍA.....	210
GLOSARIO	213
APÉNDICES	220

INDICE DE FIGURAS

FIGURA # 1. Placa Arduino UNO	31
FIGURA # 2. Placa Arduino MEGA	35
FIGURA # 3. Graficas Representativas de señales continuas y discretas.....	37
FIGURA # 4. Cantidad de vapor que contiene el aire.....	38
FIGURA # 5. Gráfico de vapor en agua y porcentaje de humedad relativa.....	38
FIGURA # 6. Higrómetro.....	39
FIGURA # 7. Medidor de humedad.....	40
FIGURA # 8. Ratings de los lenguajes de programación del índice TIOBE	42
FIGURA # 9. Gráfica del incremento del número de desarrolladores de los principales sistemas de operativos móviles.....	42
FIGURA # 10. Cuota del mercado móvil en España 2015-2017.....	43
FIGURA # 11. Estadísticas y sus versiones Android (agosto 2015)	44
FIGURA # 12. Estructura del sistema operativo Android.....	45
FIGURA # 13. Compilación del código JAVA.....	48
FIGURA # 14. Conversión de class a dex.....	49
FIGURA # 15. Logo herramienta de construcción” Gradle”.	56
FIGURA # 16. Motor de base de datos nativa de Android	57
FIGURA # 17. Logo librería “OrmLite”	57
FIGURA # 18. Gestor de base de datos “DB Browser for SDLITE”	58
FIGURA # 19. Plataforma virtual.....	60
FIGURA # 20. Arquitectura de los servicios Web SOAP.....	60
FIGURA # 21. Esquema representativo de Web Service.....	61
FIGURA # 22. Object JSON	67
FIGURA # 23. Array JSON.....	67
FIGURA # 24. Value JSON	68
FIGURA # 25. String JSON	68
FIGURA # 26. Number JSON	69
FIGURA # 27. Objeto JSON.....	69
FIGURA # 28. Esquema de una API	71

FIGURA # 29. Tabla de la base de datos en MYSQL	77
FIGURA # 30. Diagrama de contenedores para el sistema-base de datos	77
FIGURA # 31. Tarjeta Shield o Escudo Ethernet W5100	79
FIGURA # 32. Tarjeta Shield o Escudo Ethernet W5200	82
FIGURA # 33. Tarjeta Shield o Escudo Ethernet W5500	83
FIGURA # 34. Power over Ethernet Shield	84
FIGURA # 35. Atmel AVR ATmega328 and Wiznet WIZ811MJ Network Module Simplify Schematic	85
FIGURA # 36. The Wiznet W5100 Hardwire TCP/IP Stack Chip and WIZ811MJ Ethernet Module.	86
FIGURA # 37. ESP8266	88
FIGURA # 38. NodeMCU-esp8266.....	88
FIGURA # 39. Esp32	88
FIGURA # 40. Shield GSM/GPRS SIM900	96
FIGURA # 41. Sensor DHT22.....	101
FIGURA # 42. Terminales del sensor DHT22	102
FIGURA # 43. Conexión sensor DHT22 con Arduino Uno	103
FIGURA # 44. Código básico de programación del sensor DHT22 con Arduino Uno	104
FIGURA # 45. Diagrama de circuito LDR con Arduino (con Breadboard)	105
FIGURA # 46. Conexión directa con el sensor LDR con Arduino (sin Breadborad)	106
FIGURA # 47. Sketch Arduino para pruebas con sensor LDR	106
FIGURA # 48. Sensor SCT013-030	107
FIGURA # 49. PCB interna del SCT013-030 junto secundario del transformador	108
FIGURA # 50. Esquema equivalente	108
FIGURA # 51. Equivalente completo	109
FIGURA # 52. Ecuaciones de relación de transformación	109
FIGURA # 53. Adaptador para sensor	110
FIGURA # 54. Montaje de sensor con adaptador.....	110
FIGURA # 55. Sensor CT (Current Transformers)	111
FIGURA # 56. Forma de pinza amperimetrica del sensor de corriente	111
FIGURA # 57. Sensor SCT013-100	112

FIGURA # 58. Diagrama eléctrico acondicionador de señal para sensor SCT-013-030..	114
FIGURA # 59. Circuito acondicionador de señal para sensor SCT-013-100.....	114
FIGURA # 60. Terminales del sensor SCT-013 en Jack macho 3.5mm.....	115
FIGURA # 61. Terminales del sensor SCT-013 sin Jack macho 3.5mm.....	115
FIGURA # 62. Montaje del circuito Arduino- Sensor SCT013-030	116
FIGURA # 63. Código fuente Arduino – Sensor SCT013-030.....	116
FIGURA # 64. Código Arduino para obtener la corriente	117
FIGURA # 65. Código utilizar referencia de 1.1V en Arduino Mega	117
FIGURA # 66. Estado de un relé	118
FIGURA # 67. Módulo de relé de 2 canales HL-52S	118
FIGURA # 68. Módulo de relé de 2 canales HL-52S- alimentación y señales de control	119
FIGURA # 69. Esquema del circuito HL-52S- con puente entre el JDVcc y el pin Vcc..	120
FIGURA # 70. Esquema del circuito HL-52S- sin puente entre el JDVcc y el pin Vcc...	120
FIGURA # 71. Esquema del circuito HL-52S- señal de entrada terminal In1	121
FIGURA # 72. Conexión física del módulo relé HL-52S- con carga de alto voltaje en Arduino Mega.....	122
FIGURA # 73. Código fuente para control ON-OFF de luminaria a 120 VAC con Arduino	123
FIGURA # 74. Versión del IDE Arduino Studio utilizada para el desarrollo de la APP del proyecto.....	125
FIGURA # 75. Fichero de definición de permisos por el programador.....	126
FIGURA # 76. Portada de la APP.....	129
FIGURA # 77. LOGIN.....	130
FIGURA # 78. Registrarme	130
FIGURA # 79. Recuperar clave	130
FIGURA # 80. Menú principal que se muestra a usuario Master.....	134
FIGURA # 81. Login Master	135
FIGURA # 82. System Settings	135
FIGURA # 83. Servidor.....	136
FIGURA # 84. Información cuenta.....	137
FIGURA # 85. Perfil de usuario	137

FIGURA # 86. Confirmación borrar cuenta de usuario	138
FIGURA # 87. Solicitud permiso para encender BT	139
FIGURA # 88. Sin solicitud para encender BT	139
FIGURA # 89. Lista de dispositivos BT vinculados.....	140
FIGURA # 90. Conectando	140
FIGURA # 91. Error BT.....	140
FIGURA # 92. Conexión establecida.....	140
FIGURA # 93. Menú contextual BT	141
FIGURA # 94. Control de luces por comandos de voz.....	143
FIGURA # 95. Menú contextual voz	144
FIGURA # 96. Orden ejecutada correctamente.....	144
FIGURA # 97. Control SMS sin configuración.....	145
FIGURA # 98. Menú contextual control SMS	145
FIGURA # 99. Control SMS con confirmación	145
FIGURA # 100. Dashboard de notificación	146
FIGURA # 101. Menu contextual Dashboard SMS.....	147
FIGURA # 102. Cambio múltiple luminarias por SMS	148
FIGURA # 103. Call me (perdida).....	149
FIGURA # 104. Cambio múltiple de estado luminarias por SMS.....	149
FIGURA # 105. Dashboard ultimo estado de iluminarias.....	150
FIGURA # 106. Estado de temperatura y humedad relativa	150
FIGURA # 107. Grafica de estado de temperatura y humedad relativa.....	150
FIGURA # 108. Grafica de líneas, lecturas de temperaturas °C y RH%	151
FIGURA # 109. Grafica barra, lectura de último registro °C y RH%	152
FIGURA # 110. Grafica pastel de lectura de último registro °C y RH %.....	153
FIGURA # 111. Compartir información de °C y RH % por WhatsApp	154
FIGURA # 112. Envío SMS	155
FIGURA # 113. Ubicación de la UTLA	155
FIGURA # 114. Demostración del uso de Google Maps.....	156
FIGURA # 115. Acerca de	157
FIGURA # 116. Lectura de temperatura y humedad relativa en navegador web.....	158

FIGURA # 117. Funcionalidad del Web Service	160
FIGURA # 118. Logo XAMPP panel control	164
FIGURA # 119. Pantalla XAMPP panel control	164
FIGURA # 120. Creación de nueva base de datos con phpmyadmin	165
FIGURA # 121. Modelo de datos DHT11/DHT22.....	166
FIGURA # 122. Base de datos del sistema.....	166
FIGURA # 123. Montaje electrónico Shield Ethernet con Arduino Mega	167
FIGURA # 124. Código fuente de prueba Shield Ethernet y Arduino Mega.....	168
FIGURA # 125. Sensor DHT11.....	173
FIGURA # 126. Sensor DHT22.....	173
FIGURA # 127. Módulo de sensor LDR.....	174
FIGURA # 128. Módulo de 2 relés	176
FIGURA # 129. Contactor.....	177
FIGURA # 130. Simulación circuito de automatización (entradas y salidas)	177
FIGURA # 131. Simulación del plano de instalación eléctrica/electrónica	178
FIGURA # 132. Plano de la instalación eléctrica/electrónica	178
FIGURA # 133. Diagrama de conexión y control de luminarias desde 2 puntos.....	179
FIGURA # 134. Código fuente sensor DHT22	180
FIGURA # 135. Código fuente sensor LDR- lectura de ausencia de luz.....	181
FIGURA # 136. Código fuente sensor LDR- lectura de presencia de luz.....	181
FIGURA # 137. Montaje realizado por pruebas de funcionamiento	182
FIGURA # 138. Diseño electrónico del sistema integrado	184
FIGURA # 139.Shield SIM900 GSM/GPRS	188
FIGURA # 140. Descripción física de la tarjeta SIM900 GSM/GPRS.....	189
FIGURA # 141. Parte trasera de la tarjeta SIM900 GSM/GPRS	191
FIGURA # 142. Pin out Del chip SIM900 GSM/GPRS	191
FIGURA # 143. Chip SIM900 GSM/GPRS	192
FIGURA # 144. Conexión FTDI del escudo SIM900.....	194
FIGURA # 145. Conexión Arduino con escudo SIM900	195
FIGURA # 146. Montaje para pruebas comandos AT	195

FIGURA # 147. Sketch básico para pruebas de funcionalidad por comandos AT desde la ventana de monitor serial IDE de Arduino.....	197
FIGURA # 148. Sketch básico para realizar llamadas	198
FIGURA # 149. Sketch básico para realizar SMS.....	199
FIGURA # 150. Comunicación mediante GPRS e Internet (server-BD).....	200
FIGURA # 151. Prototipo físico del sistema y comunicaciones	202
FIGURA # 152. Prototipo para el control de luminarias por interruptor y electrónicamente con Arduino	202
FIGURA # 153. Sketch básico para enviar datos al servidor en internet por GPRS	204
FIGURA # 154. Esquema representativo del circuito de monitoreo de temperatura y humedad relativa por Ethernet.....	220
FIGURA # 155. Esquemático representativo del circuito de control de luces remotamente con Shield GSM/GPRS SIM900	221
FIGURA # 156. Esquemático representativo del circuito de control de luces localmente con Bluetooth	221
FIGURA # 157. Tarjeta Shield GSM/GPRS SIM900.....	222
FIGURA # 158. Simulación en software Proteus del sistema de control de cargas (luminarias).....	226

INDICE DE TABLAS

Tabla 1. Comparativa de especificaciones de placas Arduino	36
Tabla 2. Versiones de sistema operativo Android.....	54
Tabla 3: Comparación de especificaciones de tarjetas WIFI	88
Tabla 4. Comandos AT para la conexión TCP / UDP	98
Tabla 5. Sensores de temperatura y humedad relativa, DHT11 VS DHT22.....	101
Tabla 6. Obtenida del Datasheet SCT-013	112
Tabla 7: Ayuda de comando de voz	142
Tabla 8. Listado de software en XAMPP panel de control.....	163
Tabla 9. Descripción de los LED indicadores de la tarjeta GSM/GPRS SIM900... .	192
Tabla 10. Características de placas Arduino.....	222
Tabla 11: Comandos AT más usados en los módulos GSM/GRPS y GPS SIM900 y SIM808	224
Tabla 12. Herramientas de software utilizadas	227
Tabla 13. Principales componentes electrónicos del sistema	227
Tabla 14. Características del sensor DHT	228

INTRODUCCIÓN

Uno de los campos que ha registrado mayor evolución en los últimos años, han sido los sistemas de telecomunicación, especialmente la telefonía móvil, que día tras día ofrece más servicios a los usuarios. Por ejemplo: La telefonía IP, paquetes de datos, redes Wi-Fi, mensajes de texto, mensajes multimedia, Bluetooth, reproductores de música y videos, televisión, banca virtual, sistemas de información, entre otros.

Dentro del panorama de las comunicaciones, aparecen los sistemas de transmisión de datos inalámbricos, estos sistemas son particularmente apropiados para aplicaciones de telemetría o de computadoras portátiles, lo cual permite movilidad, con las ventajas de estar conectados a una red.

El siguiente proyecto de grado abarca todos los aspectos del proceso de diseño de un sistema que tiene como finalidad; la formulación, diseño e implementación de un sistema de monitoreo de parámetros ambientales; y el control y monitoreo de luces en el Laboratorio de Electrónica de la Universidad Técnica Latinoamericana (UTLA).

DESCRIPCIÓN DEL PROBLEMA

Actualmente existe un riesgo eminente para el equipo del Laboratorio de Electrónica y personal dentro del mismo, además de incidir en la toma de acciones ante posibles casos de sobrecalentamiento y afectar la disponibilidad del servicio brindado hacia el Mercado Meta de la UTLA, sus estudiantes.

Esto debido a que se identificó que actualmente el Laboratorio de Electrónica de la UTLA, no cuenta con un registro de parámetros ambientales como temperatura y humedad relativa. Además, no se monitorea ni se controla el estado de las luces; dando lugar a la problemática del alto riesgo de pérdidas, tanto materiales, como de carácter económico, provocando con ello un costo más elevado y no controlado por el área administrativa de la UTLA, al utilizar energía eléctrica en momentos que no es necesario. Un escenario de lo anterior, puede interpretarse/entenderse por un error humano (olvidos) por parte del personal o alumnado circulantes en el área en estudio.

ALCANCES Y LIMITACIONES.

ALCANCES.

Desarrollar un sistema de control de luces, a su vez el monitoreo de las variables involucradas en el sistema (Temperatura y humedad relativa, estado de las luces ON u OFF) en el Laboratorio de Electrónica; este sistema podrá:

- ✚ Monitorear la temperatura y humedad relativa, a su vez como el control y estado de las luminarias en el Laboratorio de Electrónica de la UTLA.
- ✚ Control ON-OFF de las luminarias localmente (vía Bluetooth - App Android) y remotamente (vía SMS/CALL - App Android).
- ✚ Control ON-OFF de las luminarias remotamente vía una llamada perdida desde la opción CALL en la interface gráfica de la App Android con notificación SMS incluida de retorno, informando en estado final de las luminarias.
- ✚ Enviar la información captadas por los diferentes sensores al servidor de base de datos en la nube utilizando la red telefónica GPRS.
- ✚ Consultar Información vía SMS a través de la App acerca del estado de las variables contraladas por el sistema (estado y control de luminarias, estado de temperatura y humedad relativa en el Laboratorio de Electrónica de la UTLA).
- ✚ Adecuar el sistema eléctrico en el Laboratorio de Electrónica de la UTLA, en donde se implementará el sistema electrónico de monitoreo de la temperatura y la humedad relativa, a su vez el control y estado ON-OFF de las luminarias.

LIMITANTES

La investigación para el desarrollo del sistema tecnológico presentado en este proyecto, estará diseñado solamente para el Laboratorio de Electrónica de la carrera Ingeniería Electrónica. La programación y manipulación vía un dispositivo móvil celular en este sistema hace necesario poseer un equipo terminal con conexión a Internet con un sistema operativo Android, para monitorear:

- ✚ El estado de las luminarias ON-OFF.
- ✚ Temperatura.
- ✚ Humedad Relativa.

REQUERIMIENTOS PARA LA CORRECTA FUNCIONALIDAD DEL PRESENTE PROYECTO:

- ✚ Conectividad a Internet segura (Se recomienda un acceso dedicado n*64K, dimensionado y/o recomendado bajo la supervisión y visto bueno del administrador de informática de la UTLA).
- ✚ El aparato móvil celular a utilizar será de uso institucional (con sistema operativo Android) en donde se instala la App, el cual deberá poseer siempre SALDO (o en su defecto un óptimo plan de voz y datos) para envío de SMS de control remotamente. Para el caso del control ON-OFF de las luminarias vía Bluetooth no será necesario poseer saldo para su óptima funcionalidad.
- ✚ La conexión Bluetooth tiene un alcance aproximado de los 10 metros suficientes para cubrir el área del Laboratorio de Electrónica de la UTLA y tener un óptimo control de las luminarias desde la App.
- ✚ El espacio (Aula) debe contar al 100% con energía eléctrica para que el sistema esté debidamente alimentado todo el tiempo.
- ✚ El sistema electrónico a implementar debe poseer un chip con saldo para la comunicación GSM a través de SMS desde el sistema eléctrico/electrónico hacia la App y viceversa. Brindar información y visualizar en un dispositivo celular a través de la App Android o como un SMS Entrante.

OBJETIVOS

OBJETIVO GENERAL

- ✚ Implementar un sistema que permita observar parámetros ambientales y controlar y vigilar el estado de luminarias de manera local y remota, mediante la transmisión de SMS a través de una App Android empleando la red Bluetooth, GSM y GPRS; para la prevención de un calentamiento de equipos instalados en el Laboratorio de Electrónica.

OBJETIVO ESPECÍFICO

- ✚ Diseñar el esquema eléctrico/electrónico para el control y monitoreo de las variables del sistema y la base de datos.
- ✚ Desarrollar la codificación de la tarjeta Arduino para su utilización con sensores de temperatura y humedad relativa, modulo Bluetooth, escudo y módulo de comunicaciones GSM/GPRS.
- ✚ Desarrollar el web service para la comunicación con el servidor web, base de datos y la App Android.

JUSTIFICACIÓN

El sistema que se implementará en el Laboratorio de Electrónica de la UTLA, permitirá monitorear el estado de la temperatura ambiente, humedad relativa y estado de luminarias (ON-OFF) en dicho lugar, teniendo de esta manera información ágil y oportuna para la toma de decisiones inmediatas.

La Importancia de poder contar con un control y monitoreo de las luces de una sala o Laboratorio, además de la verificación del estado de la temperatura y humedad relativa ambiente de dicho lugar, permite hacer eficiente el uso del recurso energético, asegurando que el consumo energético sea el necesario de acuerdo a la demanda requerida, por otra parte; de prevenir grandes pérdidas económicas en materiales y equipos electrónicos, eléctricos, neumáticos, hidráulicos, etc. que se encuentre en dicho lugar debido a excesos de temperaturas no monitoreado oportunamente.

Lo cual reduce sustancialmente el riesgo de un incendio por sobrecalentamiento de equipos con dicho monitoreo; por otra parte, se contribuirá a la economía energética previniendo así el uso innecesario de energía eléctrica.

CAPÍTULO I

MARCO TEÓRICO

1. MARCO TEÓRICO

1.1. CONTROLADOR ARDUINO

Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios. Es una plataforma de prototipos electrónica de código abierto (open-source) basada en hardware y software flexibles y fáciles de usar.

Arduino puede medir entorno mediante la recepción de entradas desde una variedad de sensores y puede afectar a su alrededor mediante el control de luces, motores y otros artefactos.

El microcontrolador de la placa se programa usando el Arduino Programming Language (basado en Wiring) y el Arduino Development Environment (basado en Pressing).

Los proyectos de Arduino pueden ser autónomos o se pueden comunicar con software en ejecución en un ordenador. Las placas se pueden ensamblar a mano o encargarlas pre-ensambladas; el software se puede descargar gratuitamente.

Los diseños de referencia del hardware (archivos CAD) están disponibles bajo licencia open-source, por lo que se tiene la libertad de adaptarlas a las necesidades puntuales de cualquier proyecto.

Microcontrolador: Un microcontrolador (abreviado μC o MCU) es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria.

Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica.

Los microcontroladores más usados en Arduino son los Atmega168, Atmega328, Atmega1280 y Atmega8 por su sencillez y bajo valor monetario que permiten el desarrollo de múltiples diseños. El Software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing y el cargador de arranque que es ejecutado en la placa. Se programa en el ordenador para que la placa controle los componentes electrónicos (sensores y actuadores).



Figura 1: Placa Arduino UNO

Arduino Uno es una placa electrónica basada en el microcontrolador ATmega328. Cuenta con 14 entradas/salidas digitales, de las cuales 6 se pueden utilizar como salidas PWM (Modulación por ancho de pulsos) y otras 6 son entradas analógicas. Además, incluye un resonador cerámico de 16 MHz, un conector USB, un conector de alimentación, una cabecera ICSP y un botón de reseteo. La placa incluye todo lo necesario para que el microcontrolador haga su trabajo, basta conectarla a un ordenador con un cable USB o a la corriente eléctrica a través de un transformador.

1.2. CARACTERÍSTICAS TÉCNICAS DE ARDUINO UNO R3

- ❖ Microcontrolador: ATmega328.
- ❖ Voltaje: 5V.
- ❖ Voltaje de entrada (recomendado): 7-12V.
- ❖ Voltaje de entrada (limites): 6-20V.
- ❖ Digital I/O Pins: 14 (de los cuales 6 son salida PWM).

- ❖ Entradas Analógicas: 6.
- ❖ DC Current per I/O Pin: 40 mA.
- ❖ DC Current parar 3.3V Pin: 50 mA.
- ❖ Flash Memory: 32 KB (ATmega328) de los cuales 0.5 KB son utilizados para el arranque.
- ❖ SRAM: 2 KB (ATmega328).
- ❖ EEPROM: 1 KB (ATmega328).
- ❖ Clock Speed: 16 MHz.

Entradas y salidas digitales/analógicas

Un sistema electrónico es cualquier disposición de componentes electrónicos con un conjunto definido de entradas y salidas. Una placa Arduino, por tanto, puede pensarse de forma simplificada como un sistema que acepta información en forma de señal de entrada, desarrolla ciertas operaciones sobre ésta y luego produce señales de salida en función a la programación grabada en la propia tarjeta Arduino.

Justamente, una de las opciones que hacen más potente a Arduino son sus entradas/salidas digitales.

En los sistemas electrónicos, una magnitud física variable se representa generalmente mediante una señal eléctrica que varía de manera tal que describe esa magnitud. Por lo general, se hace referencia a las señales continuas como señales analógicas, mientras que asociamos las señales discretas a señales digitales: el ejemplo más claro es el de las señales binarias, donde la señal sólo puede tomar dos niveles, 0 (LOW) o 1 (HIGH).

Arduino incorpora terminales digitales (señales discretas) de tal forma que se tiene un gran abanico de valores con los que trabajar (por ejemplo, 255 valores de luz en una foto sensor, siendo 0 ausencia de luz y 254 el máximo valor lumínico).

Terminales Digitales

Los terminales digitales de una placa Arduino pueden ser utilizadas para entradas o salidas de propósito general a través de los comandos de programación `pinMode ()`, `digitalRead ()`, y `digitalWrite ()`. Cada terminal tiene una resistencia pull-up que puede activarse o desactivarse utilizando `digitalWrite ()` (con un valor de HIGH o LOW, respectivamente) cuando el pin está configurado como entrada. La corriente máxima por salida es 40 mA.

- ❖ **Serial:** 0 (RX) y 1 (TX). Utilizado para recibir (RX) y transmitir (TX) datos serie TTL.
- ❖ **Interruptores externos:** 2 y 3. Estas terminales pueden ser configuradas para disparar una interrupción con un valor bajo, un pulso de subida o bajada, o un cambio de valor. Consultar la función `attachInterrupt ()` para más detalles.
- ❖ **PWM:** 3, 5, 6, 9, 10, y 11. Proporcionan salidas PWM de 8 bit con la función `analogWrite ()`. En placas con ATmega8, las salidas PWM solo están disponibles en los pines 9, 10, y 11.
- ❖ **SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estas terminales soportan comunicación SPI. Aunque esta funcionalidad esta proporcionada por el hardware, no está incluida actualmente en el lenguaje Arduino.
- ❖ **LED:** 13. En la mayoría de tarjetas Arduino hay un led en placa conectado al pin digital 13. cuando el pin tiene valor HIGH, el LED está encendido, cuando el pin está en LOW, está apagado.

Pines analógicos

Los pines de entrada analógicos soportan conversiones analógico-digital (ADC) de 10 bit utilizando la función `analogRead ()`. Las entradas analógicas pueden ser también usadas como pines digitales: entrada analógica 0 como pin digital 14 hasta la entrada analógica 5 como pin digital 19. Las entradas analógicas 6 y 7 (presentes en el Mini y el BT) no pueden ser utilizadas como pines digitales.

I2C: 4 (SDA) y 5 (SCL). Soportan comunicaciones I2C (TWI) utilizando la librería Wire (documentación en la página web de Wiring).

PINES DE ALIMENTACIÓN.

VIN. (a veces marcada como “9V”). Es el voltaje de entrada a la placa Arduino cuando se está utilizando una fuente de alimentación externa (En comparación con los 5 voltios de la conexión USB o de otra fuente de alimentación regulada). Se puede proporcionar voltaje a través de este pin. Observar que diferentes placas aceptan distintos rangos de voltaje de entrada, se recomienda ver la documentación de la placa adquirida.

5V. La alimentación regulada utilizada para alimentar el microcontrolador y otros componentes de la placa. Esta puede venir de VIN a través de un regulador en placa o ser proporcionada por USB u otra fuente regulada de 5V.

3V3. Una fuente de 3.3 voltios generada por el chip FTDI de la placa.

GND. Pines de tierra.

OTROS PINES.

AREF. Referencia de voltaje para las entradas analógicas. Utilizada con la función analogReference ().

RESET. Pon esta línea a LOW para resetear el microcontrolador. Utilizada típicamente para añadir un botón de reset a shields que bloquean el de la placa principal.

El Arduino Mega es uno de los microcontroladores con mayores capacidades de la familia Arduino. Posee 54 pines digitales que funcionan como entrada/salida; 16

entradas análogas, un cristal oscilador de 16 MHz, una conexión USB, un botón de reset y una entrada para la alimentación de la placa.

La comunicación entre la computadora y Arduino se produce a través del Puerto Serie. Posee un convertidor USB-Serie, por lo que sólo se necesita conectar el dispositivo a la computadora utilizando un cable USB como el que utilizan las impresoras.



Figura 2: Placa Arduino MEGA

1.3. CARACTERÍSTICAS TÉCNICAS DE LA TARJETA ARDUINO MEGA.

- ❖ Microcontrolador: ATmega2560
- ❖ Voltaje Operativo: 5V
- ❖ Voltaje de Entrada: 7-12V
- ❖ Voltaje de Entrada(límites): 6-20V
- ❖ Pines digitales de Entrada/Salida: 54 (de los cuales 15 proveen salida PWM)
- ❖ Pines análogos de entrada: 16
- ❖ Corriente DC por cada Pin Entrada/Salida: 40 mA
- ❖ Corriente DC entregada en el Pin 3.3V: 50 mA
- ❖ Memoria Flash: 256 KB (8KB usados por el bootloader)
- ❖ SRAM: 8KB

- ❖ EEPROM: 4KB
- ❖ Clock Speed: 16 MHz

Alimentación

- ❖ VIN: A través de este pin es posible proporcionar alimentación a la placa.
- ❖ 5V: Podemos obtener un voltaje de 5V y una corriente de 40mA desde este pin.
- ❖ 3.3V: Se puede obtener un voltaje de 3.3V y una corriente de 50mA desde este pin.
- ❖ GND: El ground (0V) de la placa.

Utilizando el lenguaje propio de Arduino este es sencillo de programar si se utiliza la interfaz IDE con el lenguaje de programación de sistemas (Lenguaje C).

A continuación, se muestra la tabla 1. La cual hace una comparativa de las placas Arduino más conocidas y utilizadas en distintos escenarios de investigación.

Nombre	Microcontrolador	Voltaje del Sistema	Frecuencia de Reloj	Digital I/O	Entradas Analógicas	Memoria Flash	Interfaz de programación
Arduino Due	AT91SAM3X8E	3,3V	84MHz	54	12	512Kb	Nativa USB
Arduino Leonardo	ATmega32U4	5V	16MHz	20	12	32Kb	Nativa USB
Arduino Uno – R3	ATmega328	5V	16MHz	14	6	32Kb	USB vía ATmega16U2
Arduino Mega	ATmega2560	5V	16MHz	54	16	256Kb	USB vía ATmega16U2
Arduino Mini	ATmega328	5V	16MHz	14	6	32Kb	Cabecera Serial

Tabla 1. Comparativa de especificaciones de placas Arduino.

Para el desarrollo de este trabajo de investigación se hará uso de la placa Arduino Mega por las características que ya se mencionaron, las cuales cubren con la demanda de pines de entrada y salida digitales, analógicos, comunicación serial, I2C, entre otros. Los cuales son necesarios para la puesta en marcha del presente trabajo.

1.4. SENSORES

Los sensores son dispositivos capaces de transformar magnitudes físicas a señales eléctricas que pueden ser interpretadas por un microcontrolador. La salida de los sensores se tiene que conectar a una de las entradas del microcontrolador. Estos dispositivos pueden ser analógicos o digitales dependiendo del tipo de información que devuelvan, que pueden ser variables continuas o discretas.

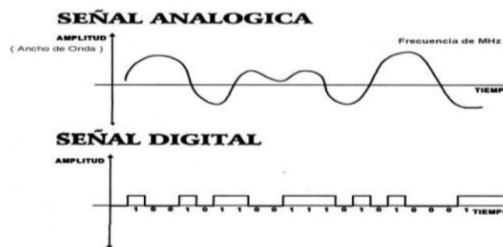


Figura 3: Gráficas representativas de señales continuas y discretas.

1.5. TEMPERATURA Y HUMEDAD

La temperatura es una magnitud referida a las nociones comunes de calor medible mediante un termómetro.

Se denomina humedad al agua que impregna un cuerpo o al vapor presente en la atmósfera.

La humedad. La mayor parte del aire que te rodea tiene agua en forma de un gas llamado vapor de agua. La humedad es la cantidad de vapor de agua que contiene ese aire, por eso también se llama "Humedad del Aire". No se puede ver la humedad porque el vapor de agua es transparente, pero si se puede sentir.

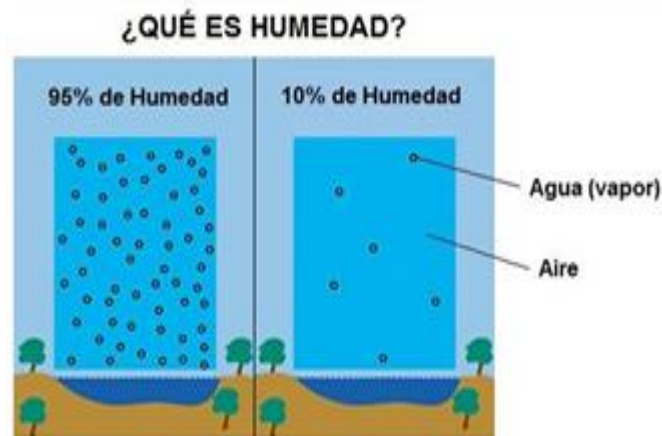


Figura 4: Cantidad de vapor que contiene el aire.

Como ya se sabe, dos terceras partes de la superficie de la tierra están cubiertas por agua: océanos, lagos y ríos, de las cuales se desprende el vapor de agua.

Esto quiere decir que el mismo aire puede estar saturado de vapor de agua a una temperatura, pero si se aumenta su temperatura, se puede introducir en él más vapor.



Figura 5: Gráfico de vapor en agua y porcentaje de humedad relativa.

A mayor temperatura, se tendrá menos humedad relativa, pero la cantidad de agua es la misma. Eso es porque al aumentar la temperatura se puede meter más agua en ese aire. La cantidad que a 100° ya no podía entrar más agua en ese aire (al 100%) si se le sube a 20°, como ahora entraría más agua, esa misma cantidad de agua ya no sería el 100% de la cantidad total que podría tener ese aire.

Tres tipos de humedad: humedad relativa, humedad absoluta y humedad específica. En cada caso el aparato que la mide es diferente.

Humedad Relativa: En la atmósfera siempre hay una cantidad de agua en forma de vapor, aunque no se pueda ver. La cantidad de agua que contiene el aire se llama humedad y se suele medir como humedad relativa.

Si el valor de la humedad es del 50%, quiere decir que en ese momento el aire contiene un 50% de vapor de agua respecto al máximo que podría contener, que sería el 100%. La humedad relativa se mide con el higrómetro que expresa la humedad relativa del aire en porcentaje.



Figura 6: Higrómetro

Humedad Específica: La humedad específica, se refiere a la cantidad de humedad en peso, que se requiere para saturar un kilogramo de aire seco, es decir, los gramos de vapor de agua contenidos en 1 kg de aire seco. Se expresa en g/kg. La humedad específica es muy similar a la humedad absoluta, excepto que esta última, está basada en gramos por metro cúbico, y la humedad específica, está basada en gramos de humedad por kilogramo de aire seco.

Humedad Absoluta: La humedad absoluta se refiere al peso del vapor de agua por unidad de volumen. Se llama humedad absoluta al peso en gramos del vapor de agua contenido en 1 metro cúbico de aire. Se expresa en gm³ (con valores medios de 10 a 12 g/m³, pero puede llegar hasta 40 g/m³). La humedad absoluta y específica se suelen utilizar, además de en meteorología, para medir por ejemplo la humedad de la

madera, en los materiales de construcción, en los cereales y los granos de café o también del papel, el material textil y el algodón.

Estas humedades se miden con unos instrumentos llamados "medidores de Humedad".



Figura 7: Medidor de Humedad

Al calentarse el aire, pesa menos y asciende, pero también a medida que asciende se va enfriando. Este enfriamiento al subir provoca que el vapor de agua se condense y forme minúsculas gotas de agua que originan las nubes.

Si el vapor de agua se condensa al amanecer, cuando bajan las temperaturas, se deposita sobre los objetos en forma de rocío. En ocasiones, el vapor de agua se condensa en una nube baja de pequeñísimas gotas de agua y forma la niebla. Si la temperatura es menor de 0°C, se hiela y forma la escarcha.

1.6. SISTEMA OPERATIVO ANDROID

Es una plataforma de software para dispositivos móviles que incluye un sistema operativo y aplicaciones de base. Android es un conjunto de herramientas y aplicaciones sujetadas a una distribución Linux para dispositivos móviles. No es un Sistema Operativo Android como tal, es de código abierto, gratuito y no requiere pago de licencias. Android es una plataforma de código abierto para terminales móviles que está basada en Linux y es desarrollada por Open Handset Alliance.

Android Studio es un entorno de desarrollo integrado, claro y robusto para la plataforma Android. Está disponible para los sistemas operativos Windows, Mac OS X y Linux. Contiene un amplio repertorio de herramientas para ayudar en el desarrollo

de las aplicaciones, así como la posibilidad de probar la aplicación en diferentes dispositivos virtuales y en diferentes versiones de Android. Hemos usado este entorno para desarrollar la gran parte del código de nuestro proyecto.

La principal ventaja de adoptar Android es que ofrece un enfoque unificado para el desarrollo de aplicaciones. Los desarrolladores solamente necesitan desarrollar para Android, y sus aplicaciones deberían poderse ejecutar en numerosos dispositivos, siempre y cuando los dispositivos utilicen este sistema operativo.

1.7. COMPARACIÓN Y ELECCIÓN

La aplicación del proyecto es realizada para dispositivos Smartphone con el sistema operativo Android. La elección de este sistema operativo se hace con la ayuda de la comparación de las características de los dos sistemas operativos más importantes hasta la fecha (iOS y Android). A la hora de buscar información sobre las características de uno y otro sistema, se llega a la conclusión de que realizar el proyecto en Android sería una opción más fiable y menos costosa. Los motivos son los siguientes:

- ✚ Android está basado en Java e iOS está basado en Objective C. Por la proximidad con los conocimientos del lenguaje JAVA se eligió el sistema operativo Android.
- ✚ Otro factor a tener en cuenta es el precio a nivel de desarrollo software. La programación para iOS requiere el uso de un MAC, así como una cuota anual de \$100 para poder contar con las herramientas y soportes necesarios para poder programar.
- ✚ Aunque iOS lleva más tiempo en el mercado, el sistema operativo Android se ha consolidado muy bien en el mercado y ha experimentado un crecimiento exponencial, en cuanto a niveles de usuarios se refiere. Y como se puede ver en

la siguiente gráfica, el lenguaje Java es más utilizado que Objective C según el índice TIOBE. Ver figura 8.

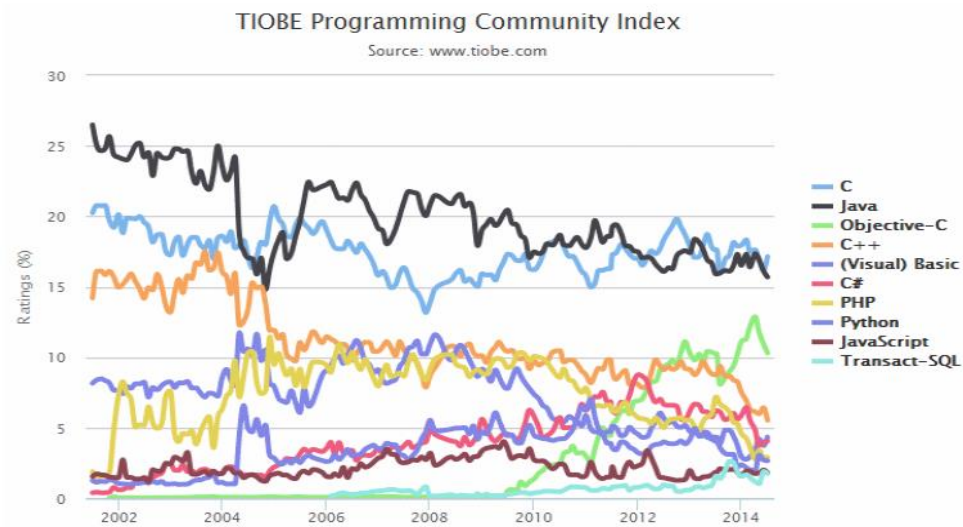


Figura 8: Ratings de los lenguajes de programación del índice TIOBE.

En cuanto a las aplicaciones, los dos sistemas principales, han incrementado el número de aplicaciones en los últimos años, pero es Android el que se ha alzado sobre iOS en este crecimiento. Android cuenta con cerca de 400 mil desarrolladores frente a los menos de 300 mil de iOS. Ver figura 9.

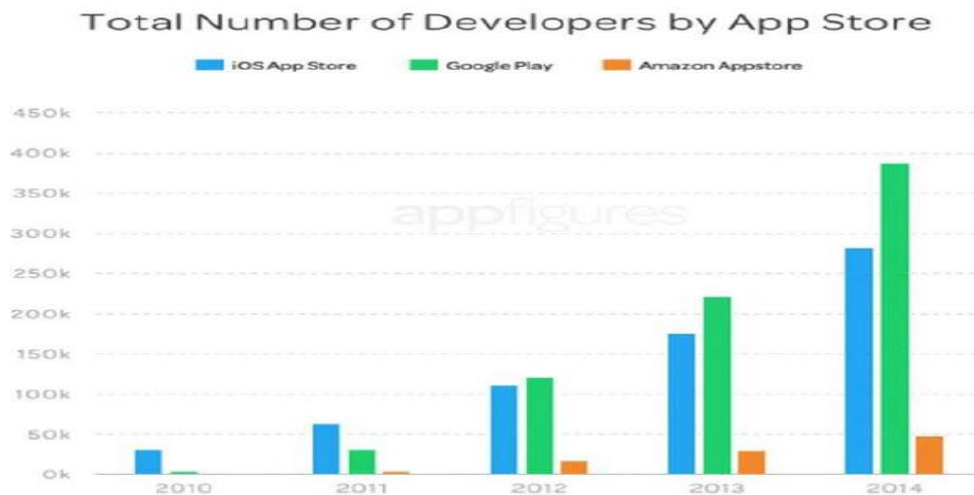


Figura 9: Gráfica del incremento del número de desarrolladores de los principales sistemas operativos móviles.

1.8. ELECCIÓN DEL SISTEMA OPERATIVO ANDROID

Cuando se inició a plantear este proyecto se quería, por una parte, que la aplicación (App) pudiera ser potencialmente utilizada, en cualquier dispositivo móvil con sistema operativo Android; por otra parte, se tenía claro que se quería hacer un desarrollo nativo de una tecnología móvil y dado que no se había tenido la oportunidad de trabajar con esta a lo largo de nuestra formación en la UTLA, se decidió tomarse como un reto el hecho de integrar esta tecnología para el desarrollo de una App, la cual forma parte importantísima de la investigación para el control y monitoreo de magnitudes físicas del sistema.

Según un informe de Kantar (Kantar es líder mundial en estudios de mercado, investigación y análisis, y trabaja para más de la mitad de las compañías que aparecen en el top 500 de la revista Fortune), Android es el sistema operativo con mayor cuota del mercado español en el 2015. Ver figura 10.

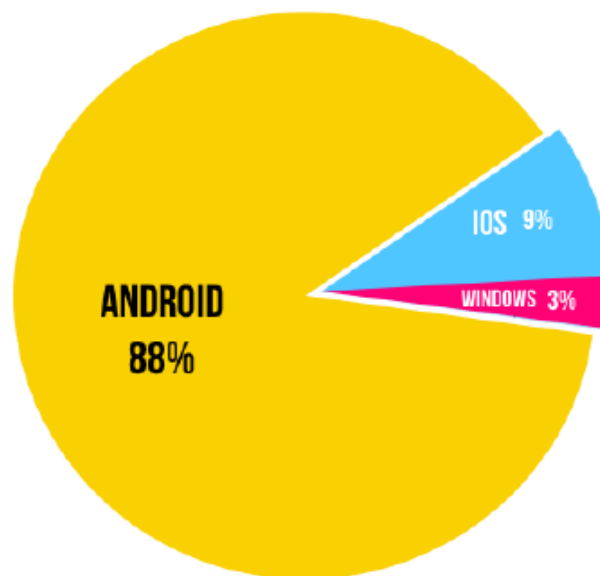


Figura 10: Cuota del mercado móvil en España 2015-2017.

Cada cierto tiempo, Google realiza estadísticas acerca de las distintas versiones Android en funcionamiento. Teniendo en cuenta la tendencia de que las últimas

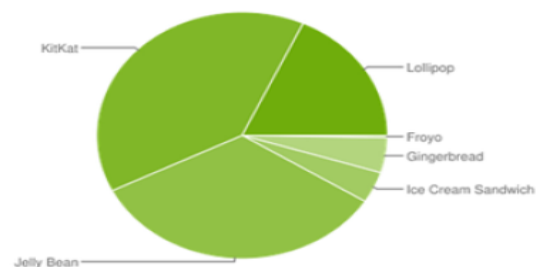
versiones lanzadas son las que acaban siendo predominantes, inicialmente pensamos hacer las dos aplicaciones en la versión Lollipop, para que tuvieran más “tiempo de vida útil”; sin embargo, después del primer encuentro en pruebas de la App, pudimos observar que las versiones de sistema operativo que la mayoría de los dispositivos móvil Android poseían eran bastante inferiores y consecuentemente decidimos elegir la versión Kitkat, que era la predominante en ese momento para implementar la aplicación usuario. Todo esto va a permitir que la App del proyecto podrá ser instalada en dispositivos smartphone con versiones de sistemas operativos Android a partir de Kitkat en adelante. Siendo actualmente la última versión Oreo. Ver figura 11.

Platform Versions

This section provides data about the relative number of devices running a given version of the Android platform.

For information about how to target your application to devices based on platform version, read [Supporting Different Platform Versions](#).

Version	Codename	API	Distribution
2.2	Froyo	8	0.3%
2.3.3 - 2.3.7	Gingerbread	10	4.6%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	4.1%
4.1.x	Jelly Bean	16	13.0%
4.2.x		17	15.9%
4.3		18	4.7%
4.4	KitKat	19	39.3%
5.0	Lollipop	21	15.5%
5.1		22	2.6%



Data collected during a 7-day period ending on August 3, 2015.

Any versions with less than 0.1% distribution are not shown.

Figura 11: Estadísticas uso versiones Android (agosto 2015).

Una vez elegido Android como sistema operativo para realizar la aplicación móvil, se detalla a continuación varios aspectos de este sistema operativo tales como la estructura, el lenguaje de programación, su desarrollo y el programa utilizado para compilar y ejecutar la aplicación.

1.9. ESTRUCTURA ANDROID

Ya se ha mencionado que Android está basado en Linux. Para ser más específicos, se habla del kernel. Android utiliza como base el kernel de Linux.

Esto no significa que, por estar basado en él, algo que se desarrolló en Linux funcione para Android, ya que por ejemplo Android no tiene soporte glibc.

La estructura consta de una serie de partes diferenciadas por colores. A continuación, se detallará cada subestructura. Ver figura 12.

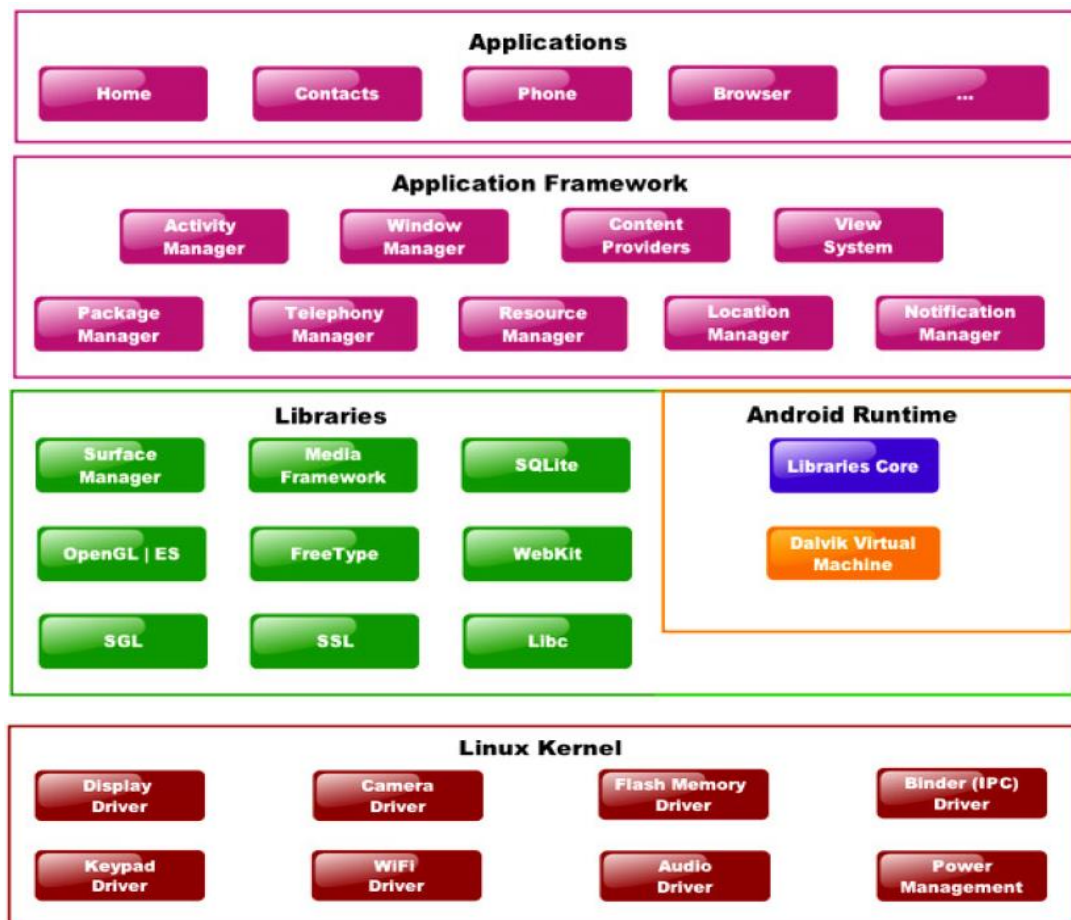


Figura 12: Estructura del sistema operativo Android

Capa del Kernel (Roja)

Aquí se encuentra el corazón de Android: el manejo de memoria, procesos, drivers, etc. Es donde se da la comunicación con el hardware. Esto sirve para no depender de los fabricantes de cada móvil. También se administran los recursos del dispositivo, memoria, energía.

Capa Librerías (Verde)

Esta capa tiene las librerías nativas de Android, están escritas en C o C++ y tienen tareas específicas.

- ✚ Surface manager: Gestión dedicada al acceso a la pantalla.
- ✚ Media Framework: Realiza las acciones de reproducción de imágenes, audio y vídeo.
- ✚ SQLite: En este punto está incluida la base de datos (BD).
- ✚ Webkit: tarea dedicada a la navegación.
- ✚ SGL: Gráficos 2D.
- ✚ OpenGL: Gráficos 3D.
- ✚ Freetype: Su tarea es enderezar vectores o imágenes.

Android Runtime (Capa naranja)

Esta capa naranja no se considera al 100% una capa. Lo que es muy importante comentar es que aquí se encuentra Dalvik, la máquina virtual de Android, que no es lo mismo que la Java Virtual Machine. Esto quiere decir que cuando se compila en Java lo que se genera solamente va a funcionar en la JVM, porque Dalvik es la máquina virtual de Android, así que el ByteCode que genera Java es inservible para Dalvik.

Algunas de las características de Dalvik son:

- ✚ Trabaja en entorno con restricción de memoria y procesador.
- ✚ Ejecuta el formato .dex.
- ✚ Convierte .class en .dx.

Application Framework (Capa rosa)

Esta capa es la es más visible para el desarrollador, ya que la mayoría de los componentes que forman parte del desarrollo se vamos a encontrar aquí.

- ✚ Activity Manager: Administra las actividades de nuestra aplicación y el ciclo de vida.
- ✚ Windows Manager: Administra lo que se muestra en la pantalla.
- ✚ Content Provider: Administra dependiendo de cómo le indiquemos algunos contenidos, puede ser información que necesitamos la encapsule para enviar o compartir.
- ✚ View: Las vistas de elementos que son parte de la interfaz gráfica, como los mapas, cuadros de texto, etc.
- ✚ Notification Manager: Administra las notificaciones.
- ✚ PackageManager: Administra los paquetes y nos permite el uso de archivos en otros paquetes.
- ✚ Telephony Manager: Administra lo que tiene que ver con la telefonía, llamadas, mensajes.
- ✚ Resource Manager: Administra recursos de la aplicación, como los xml, imágenes, sonido.
- ✚ Location Manager: Gestiona la posición geográfica.
- ✚ Sensor Manager: Gestiona los sensores que tenga el dispositivo.
- ✚ Cámara: Administra la cámara.
- ✚ Multimedia: Administra lo referente a audio, video y fotos.

Applications (Última Capa Rosa)

Aquí tenemos las aplicaciones que vienen en el dispositivo, por ejemplo: el gestor de correos, los mensajes, el market, etc.

1.10. PROGRAMACIÓN Y DESARROLLO ANDROID

Java es un lenguaje orientado al objeto que fue diseñado para funcionar de forma independiente a la arquitectura sobre la que trabaja. La compilación del código fuente (archivos .java) genera archivos de clases (archivos .class), i.e. bytecode, los cuales serán ejecutados por la máquina virtual JVM (Java Virtual Machine), utilizada como abstracción entre el hardware de la máquina y los programas Java. Los distintos archivos .class suelen ser compilados en un único archivo JAR (Java Archive). En Android sucede algo parecido.

La Open Handset Alliance optó por diseñar un sistema similar teniendo en cuenta las limitaciones de los dispositivos móviles, que en principio eran una baja capacidad de almacenamiento y procesamiento y poca potencia de cálculo. Ver figura 13.

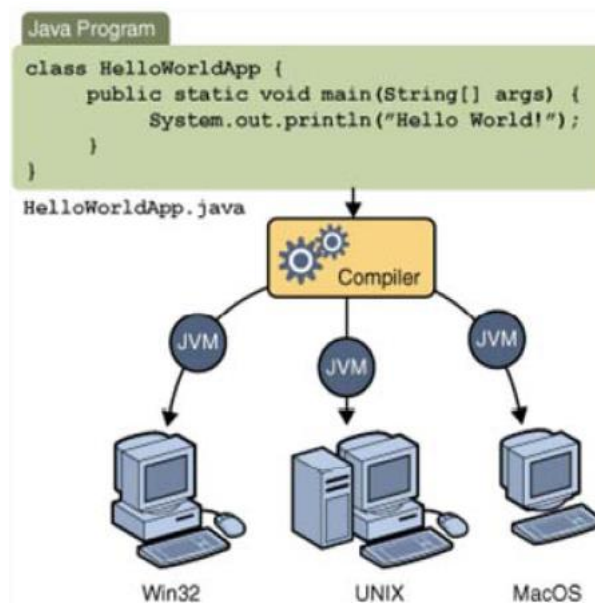


Figura 13: Compilación del código Java

La máquina virtual empleada por Android recibe el nombre de DVM (DalvikVirtual Machine).

Los programas generalmente son escritos en Java y compilados a bytecode. Posteriormente son convertidos a archivos compatibles con la DVM (archivos.dex) antes de su instalación en el dispositivo. Estos archivos son comprimidos en archivos APK (Application Package), los cuales pueden ser instalados en dispositivos Android compatibles (a la hora de crear una aplicación es necesario especificar la versión para la que está diseñada por lo que el terminal donde se instalará ha de tener una versión de Android igual o superior). Se puede observar cómo se crean los archivos con extensión. dex a partir de los archivos de clase .class.

La principal diferencia radica en la forma de empaquetar la información. Los archivos de clase son transformados en un único archivo. dex en el que se intenta desprejar la información repetida. Es más que evidente el enfoque de Android hacia dispositivos con memoria pequeña y almacenamiento similar. Ver figura 14.

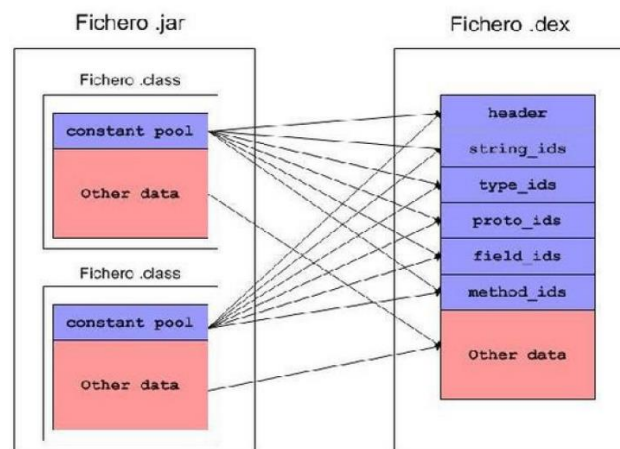


Figura 14: Conversión de class a dex

En cuanto al entorno de desarrollo, programar para Android tiene bastantes menos requisitos que hacerlo para iOS. Prueba de ello es la posibilidad de elección del sistema sobre el que se quiere programar (Unix, Windows, Mac OS, ...) y del entorno de desarrollo a utilizar. Existen varias opciones, entre las que la más destacada es sin duda alguna Eclipse. Otro programa igual de válido que Eclipse es Android Studio, surgió más tarde que Eclipse pero que desde hace unos años es los nuevos programadores e incluso los que ya programaban Android con Eclipse se están

pasando es este nuevo entorno de desarrollo como se puede ver en las opiniones de programadores de varios foros de Internet. Aquí se enumeran unas de las principales características por la cual los usuarios están decidiendo programa en un entorno como Android Studio:

- ✚ Es el futuro.
- ✚ En poco tiempo será lo único que el equipo Android recomiende.
- ✚ Está basado en IntelliJ IDEA, uno de los IDE para JAVA de primer nivel (Entre los mejores, con Eclipse y netbeans).
- ✚ Nueva forma de construir la APK. Más serio, más versátil, más potente, más actual y más parecido a un proyecto en JAVA.
- ✚ Utiliza Gradle, con ventajas como:
 - Facilita muchísimo rehusar código y recursos.
 - Facilita configurar, extender y personalizar el proceso.
 - Facilita la distribución del código y por tanto trabajar en equipos.
 - Gestiona las dependencias de una forma cómoda y potente (está basado en Maven).
 - Permite compilar desde línea de comandos.
 - Hace muy fácil crear distintas versiones de la aplicación, por ejemplo, para hacer una distribución multi-apk, para distintos dispositivos, o una versión gratis y otra de pago, o una versión de prueba que carga distintos recursos, apunta a web service distintos, usa estadísticas distintas, etc.

Otras opciones menos convencionales, pero igual de válidas pasarían por el uso de NetBeans IntelliJ IDEA.

La historial de las versiones de sistema operativo Android se inició con el lanzamiento de Android beta en noviembre de 2007. La primera versión comercial (de prueba), Android 1.0, fue lanzada en septiembre de 2008. Android es un sistema operativo móvil desarrollado por Google y la Open Handset Alliance, y ha visto un número de actualizaciones a su sistema operativo base desde su lanzamiento original. Estas actualizaciones típicamente corrigen fallos de programa y agregan nuevas

funcionalidades. Desde abril de 2009, las versiones de Android han sido desarrolladas bajo un nombre en clave y sus nombres siguen un orden alfabético: Cupcake, Donut, Éclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean, KitKat, Lollipop, Marshmallow, Nougat lanzado en agosto de 2016 y el ya anunciado "Android Oreo".

1.11. EVOLUCIÓN DE LOS SISTEMAS OPERATIVOS ANDROID

CUPCAKE (versión 1.5)

El 30 de abril de 2009, Android comenzó su curiosa denominación con postres en orden alfabético. La versión 1.5 llegó bajo el apellido Cupcake y trajo consigo interesantes novedades como la actualización de la tienda de aplicaciones Android Market Place, una interfaz más simple y organizada para el sistema, así como un renovado teclado táctil.

DONUTS (versión 1.6)

Posteriormente, el día 15 de septiembre del mismo año se liberó Android 1.6 Donut, que supuso la llegada de una mejora fundamental en la experiencia de uso de la introducción táctil. Y es que, al mismo tiempo que se mejoraba el control de la introducción táctil, esta versión de Android se actualizaba con un nuevo teclado táctil en pantalla, más sensible y preciso. Por otra parte, se optimizó la conversión de texto a voz -accesibilidad- y se introdujo el gestor de consumo energético casi como lo conocemos actualmente.

ECLAIR (versión 2.0)

Por último, el 26 de octubre de 2009 se lanzaba una tercera actualización del joven sistema operativo móvil bajo el nombre Android 2.0 Eclair. Con esta actualización, los usuarios vieron llegar un menú rápido de contactos, software para el control de la cámara, nuevas mejoras en el teclado, soporte para el estándar HTML5 en el navegador y una optimización de los gráficos.

Durante el segundo trimestre de este año 2009, Android alcanzaba una aún ridícula cuota de mercado correspondiente al 2,8% a nivel global.

FROYO (versión 2.2) Y GINGERBREAD (versión 2.3)

De nuevo, el 20 de mayo de 2010 Android se actualiza con la versión 2.2 Froyo, que supondría la llegada de soporte para múltiples idiomas en el teclado, la marcación de llamadas por comandos de voz a través de Bluetooth, el uso del smartphone como router WiFi, mejoras en la cámara y la galería y el soporte para memorias RAM de mayor capacidad.

También durante este año, pero el día 6 de diciembre, se vuelve a actualizar el sistema operativo móvil a la versión 2.3 Gingerbread, introduciendo así una importante renovación del aspecto a nivel interfaz y teclado, mejorando la gestión del consumo energético, introduciendo las llamadas de voz sobre el protocolo IP -VoIP- y, además, añadiendo soporte para la conectividad NFC (Near Field Communication o Comunicación de campo cercano).

Y en 2010, durante el cuarto trimestre del año, Android ya había alcanzado una cuota de mercado del 33% a nivel global, el crecimiento experimentado en poco más de un año fue realmente sorprendente.

HONEYCOMB (versión 3.0) e ICE CREAM SANDWICH (versión 4.0)

Manteniendo el ritmo de actualizaciones del año anterior, Android se actualiza en dos ocasiones. La primera versión de 2011 fue Android 3.0 Honeycomb, que fue además la primera en estar diseñada exclusivamente para tabletas. Con ella llegó el nuevo sistema de notificaciones, el teclado con tabulador, mejoras en la selección de texto, widgets en la pantalla de inicio y una mejor navegación web. Además, fue la primera en introducir soporte para procesadores multicore.

El 19 de octubre, sin embargo, llegó Android 4.0 Ice Cream Sandwich, que unió tabletas y teléfonos inteligentes bajo un mismo software. Las novedades de esta

versión fueron orientadas a la redimensión de widgets, la personalización de la pantalla de bloqueo y la introducción de accesos directos a aplicaciones y contactos. Ahora bien, también se introdujo la respuesta rápida de llamadas, un mejorado teclado y Android Beam NFC.

En mayo de 2011, Android estaba ya presente en más de la mitad de teléfonos inteligentes, con una cuota de mercado a nivel global del 53%.

JELLY BEAN (versión 4.1)

Entre mayo y junio de 2012, Google lanzó Android 4.1 JellyBean. Con esta versión, se optimizó notablemente el rendimiento de los dispositivos móviles y se introdujo “Project Butter”, con la intención de dar mayor fluidez a los terminales con Android. Además, Google introdujo su asistente virtual Google Now, en el que ahora centran gran parte de sus esfuerzos. Se mejoró el sistema de notificaciones y se simplificó el sistema de multitarea. Esta fue, sin duda, una de las versiones de Android que mayor evolución han dado al ecosistema.

A estas alturas, Android ya alcanzaba el 75% de cuota de mercado a nivel global.

KITKAT (versión 4.4)

Por primera vez, Google se alía con otra compañía para el nombre de su nueva versión. Así tenía comienzo la alianza entre Nestlé y Google para poner “KitKat” como nombre a esta versión, lo que vino asociado de una llamativa campaña comercial con el postre correspondiente. Esta versión, que llegó el 31 de octubre, introdujo la impresión de forma remota, los ajustes rápidos para la modificación de las pantallas de inicio, el modo inmerso para aplicaciones y la vista previa de carátulas de películas y álbumes de música en la pantalla de bloqueo, así como la botonera de reproducción. Además, también llegaron cambios importantes al gestor de correo electrónico y la aplicación de descargas.

El año 2013, Android ya alcanzaba el 78,6% de cuota de mercado a nivel global. Evidentemente, el crecimiento ha continuado, aunque mucho más lento que en años anteriores.

LOLLIPOP (versión 5.0)

Tras haber podido conocer una “vista previa” el pasado mes de junio, Android 5.0 Lollipop ha llegado ya de forma oficial. Con este último lanzamiento, y este último año de Android, el ecosistema ha crecido para llegar a los dispositivos wearables con Android Wear, a la consola integrada de los coches con Android Auto y a nuestros televisores con Android TV.

Mientras la cuota de mercado de Android continúa creciendo imparable con un último valor del 81.4%, sus competidores ya toman referencias clave de este sistema operativo. Y es que, aunque Apple llegó antes, Android ha tomado la delantera. Así, mientras que llevamos ya varios años con teléfonos inteligentes de más de 4 pulgadas de diagonal para su pantalla, ha sido este año cuando Apple ha decidido sumarse a la inevitable tendencia del mercado.

A continuación, se presenta la lista completa de versiones de sistema operativa Android hasta la fecha. Ver tabla 2.

Nombre código	Número de versión	Fecha de lanzamiento	Nivel de API
Apple Pie	1.0	23 de septiembre de 2008	1
Banana Bread	1.1	9 de febrero de 2009	2
Cupcake	1.5	25 de abril de 2009	3

Nombre código	Número de versión	Fecha de lanzamiento	Nivel de API
Donut	1.6	15 de septiembre de 2009	4
Eclair	2.0–2.1	26 de octubre de 2009	5-7
Froyo	2.2–2.2.3	20 de mayo de 2010	8
Gingerbread	2.3–2.3.7	6 de diciembre de 2010	9–10
Honeycomb	3.0–3.2.6	22 de febrero de 2011	11–13
Ice Cream Sandwich	4.0–4.0.5	18 de octubre de 2011	14–15
Jelly Bean	4.1–4.3.1	9 de julio de 2012	16–18
KitKat	4.4–4.4.4, 4.4W–4.4W.2	31 de octubre de 2013	19–20
Lollipop	5.0–5.1.1	12 de noviembre de 2014	21–22
Marshmallow	6.0–6.0.1	5 de octubre de 2015	23
Nougat	7.0 - 7.1 - 7.1.1 - 7.1.2	15 de junio de 2016	24-25

Nombre código	Número de versión	Fecha de lanzamiento	Nivel de API
Oreo	8.0 - 8.1	21 de agosto de 2017	26-27
Pie	9.0	6 de agosto de 2018	28

Tabla 2. Versiones de sistema operativo Android.

A continuación, se detallan las principales herramientas utilizadas para el desarrollo de la App de este trabajo de investigación.

1.12. HERRAMIENTAS DE DESARROLLO EN EL IDE DE ANDROID STUDIO

GRADLE



Figura 15: Logo herramienta de construcción “Gradle”.

Gradle es una herramienta para automatizar la construcción de proyectos, es decir la compilación o el testing por ejemplo, además gestiona las dependencias. Está basada en Groovy, un lenguaje de programación muy parecido a Java y está incorporada en Android Studio.

Base de datos SQLite:



Figura 16: Motor de base de datos nativa de Android

SQLite es un motor de bases de datos de código abierto, transaccional, que no necesita configuración ni requiere un servidor y se caracteriza por mantener el almacenamiento de información de forma sencilla. Es una tecnología cómoda para los dispositivos móviles a pesar de que tiene ciertas limitaciones en determinadas operaciones. Se ha utilizado este motor para crear y testear la base de datos de la aplicación Android.

ORMLite:



Figura 17: Logo librería "OrmLite"

ORMLite (Object Relational Mapping Lite) es una librería open source que permite mapear objetos Java y persistirlos en una base de datos SQL. Utiliza clases abstractas para el acceso a los datos (DAO) y soporta SQLite. Dado que nuestro proyecto usamos esa base de datos hemos utilizado esta librería para gestionar las entidades, así como los datos de las aplicaciones. Como estudiantes del Grado de Ingeniería del Software queríamos emplear el máximo de nuestros conocimientos previos, por eso decidimos persistir nuestros datos a través de JPA, sin embargo, debimos desechar esta idea dado que es un API no adecuada para Android, entonces decidimos usar ORMLite que si soporta SQLite y no nos generaba ningún problema.

DB Browser for SQLite



Figura 18: Gestor de base de datos “DB Browser for SQLITE”

DB Browser for SQLite es una herramienta open source con la que se puede gestionar archivos de bases de datos SQLite disponible para Windows, Mac OS X, Linux y FreeBSD. Nos ha permitido inspeccionar el contenido de las bases de datos que generaba nuestra aplicación, dentro de los emuladores. Cuando se importan y se abren con esta herramienta, aparecen claramente las distintas tablas con sus columnas, lo cual permite comprobar el correcto o incorrecto manejo de los datos.

1.13. REQUERIMIENTOS DE INSTALACIÓN DEL ENTORNO DE TRABAJO CON IDE ANDROID STUDIO

Los requisitos mínimos de instalación del IDE de Android Studio en diferentes plataformas y ordenadores pueden variar de una u otra. Para la creación de la Apps de este trabajo de investigación se preparó el entorno de desarrollo en una computadora con sistema operativo Windows 10, pero se explican cuáles son los requisitos mínimos en Hardware para la instalación de Android Studio en PC Mac, PC Windows y PC Linux, para que se puedan programar apps desde el entorno de Google sin problemas en cualquier ordenador.

Windows

- Windows 7/8/10 (32 o 64 bits).
- 4 GB de RAM (8 GB de RAM recomendado). Sin uso de emulador Android.
- 2 GB de espacio libre mínimo (4 GB recomendado).
- Resolución mínima de 1.280 x 800.
- Java 8.
- 64 bits y procesador Intel (emulador).

Mac

- Mac OS X 10.8.5 o superior.
- 4 GB de RAM (8 GB de RAM recomendado).
- 2 GB de espacio libre mínimo (4 GB recomendado).
- Resolución mínima de 1.280 x 800.
- Java 6.

Linux

- GNOME o KDE Desktop.
- Ubuntu u otra distribución
- 64 bits / 32 bits.
- GNU C (glibc) 2.1 o superior.
- 4 GB de RAM (8 GB de RAM recomendado).
- 2 GB de espacio libre mínimo (4 GB recomendado).
- Resolución mínima de 1.280 x 800.
- Java 8.
- 64 bits y procesador Intel (emulador).

1.14. PLATAFORMA WEB (VIRTUAL)

Una plataforma web es un sistema que permite la ejecución de diversas aplicaciones bajo un mismo entorno, dando a los usuarios la posibilidad de acceder a ellas a través de Internet.



Figura 19: Plataforma virtual

1.15. SERVICIO WEB (WEB SERVICE)

Un servicio **web** (en inglés, **web service** o **web services**) es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.



Figura 20: Arquitectura de los Servicios Web SOAP

La mayoría de los sitios webs grandes (Facebook, MySpace, Microsoft) usan aplicaciones que utilizan servicios webs (web services).

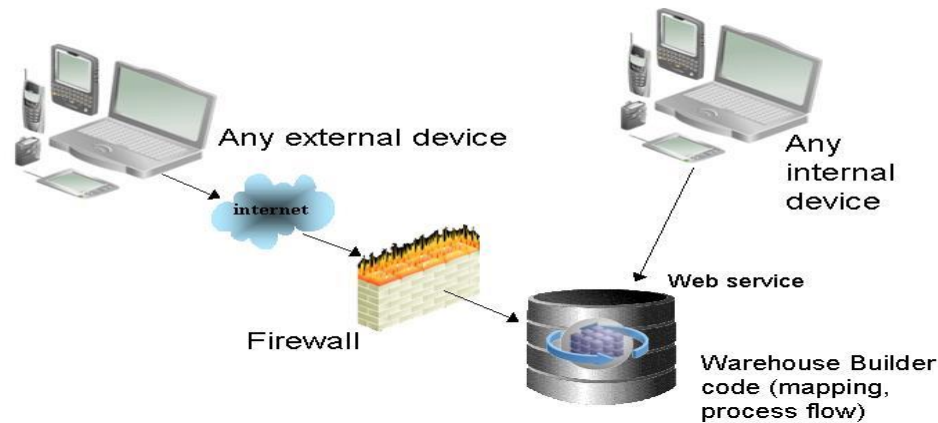


Figura 21: Esquema representativo de web Service

Un web service es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como internet.

De una manera más clara se podría decir que un web service es una función que diferentes servicios o equipos utilizan; es decir, solo se envían parámetros al servidor (lugar donde está alojado el web service) y éste responderá la petición. Entre algunas de las ventajas que se manejan de utilizar servicios webs en las aplicaciones destacan las siguientes:

- ✚ Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- ✚ Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- ✚ Al apoyarse en HTTP, los servicios Web pueden aprovecharse de los sistemas de seguridad firewall sin necesidad de cambiar las reglas de filtrado.

- ✚ Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.
- ✚ Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar y abiertos. Las especificaciones son gestionadas por una organización abierta, la W3C, por tanto, no hay secretismos por intereses particulares de fabricantes concretos y se garantiza la plena interoperabilidad entre aplicaciones.

La principal ventaja de utilizar un servicio web es que son bastante prácticos debido a que son independientes de las aplicaciones.

1.16. IMPORTANCIA DE UN WEB SERVICE

Hagamos un poco de historia. En los primeros computadores corría un solo programa a la vez, pero en la medida que en un mismo computador podían correr varios programas al mismo tiempo, surgió la necesidad de contar con mecanismo de comunicación entre ellos, esto se llamó comunicación Task to Task y, este mecanismo ha evolucionado debido a que los computadores conforman redes. Por tanto, esta comunicación debe poder efectuarse entre un programa X, que corre en el computador Alfa, y otro programa Y, que corre en el computador Beta.

Para que esta comunicación funcione, primero debe existir un medio de comunicación entre el computador Alfa y el computador Beta; esto hoy está resuelto con la Internet. Y segundo, el programa X debe saber conversar con el programa Y. Para que esto ocurra el programador a cargo de X debe conocer de Y. A su vez el programador a cargo de Y debe conocer de X, por lo menos en los que se refiere al intercambio de datos. Esto hace que, si no hay acuerdo entre el programador de X y el programador de Y, no hay comunicación posible.

La magia de los Web Services está en que el programador de X puede crear un Web Service para transferir datos sin necesidad de conocer al programador Y, ni a los programas que éste tiene a cargo. De modo que quien quiera recibir los datos solo necesita usar el Web Service y punto. Esto significa que pueden existir transferencias de datos entre distintas aplicaciones –programas- que funcionan en varios computadores, con distintos sistemas operativos, y que pertenezcan a diferentes empresas o instituciones.

A modo de ejemplo, si se ha despachado un material vía Federal Express y se quiere conocer el estado de su despacho, esta empresa pone a su disposición un Web Service.

El término Web Services describe una forma estandarizada de integrar aplicaciones WEB mediante el uso de XML, SOAP, WSDL y UDDI sobre los protocolos de la Internet. XML es usado para describir los datos, SOAP se ocupa para la transferencia de los datos, WSDL se emplea para describir los servicios disponibles y UDDI se ocupa para conocer cuáles son los servicios disponibles. Uno de los usos principales es permitir la comunicación entre las empresas y sus clientes. Los Web Services permiten a las organizaciones intercambiar datos sin necesidad de conocer los detalles de sus respectivos Sistemas de Información.

A diferencia de los modelos Cliente/Servidor, tales como un servidor de páginas Web, los Web Services no proveen al usuario una interfaz gráfica (GUI). En vez de ello, los Web Services comparten la lógica del negocio, los datos y los procesos, por medio de una interfaz de programas a través de la red. Es decir, conectan programas. Por tanto, son programas que no interactúan directamente con los usuarios. Los desarrolladores pueden por consiguiente agregar a los Web Services la interfaz para usuarios, por ejemplo, mediante una página Web o un programa ejecutable, de manera que sea posible entregarles a los usuarios una funcionalidad específica que provee un determinado Web Service.

Los Web Services permiten a distintas aplicaciones, de diferentes orígenes, comunicarse entre ellos sin necesidad de escribir programas costosos, esto porque la comunicación se hace con XML o su versión sucesora y mejorada JSON.

Los Web Services no están ligados a ningún Sistema Operativo o Lenguaje de Programación. Por ejemplo, un programa escrito en Java puede conversar con otro escrito en Perl; Aplicaciones Windows puede conversar con aplicaciones Unix. Por otra parte, los Web Services no necesitan usar browsers (Explorer) ni el lenguaje de especificación HTML.

El modelo de computación distribuida de los Web Services permite la comunicación de aplicación a aplicación. Por ejemplo, la aplicación que procesa las órdenes de compra se puede comunicar con el sistema de inventarios, tal que este último le puede informar a la aplicación de compras cuales ítems deben comprarse por estar bajo su nivel mínimo. Dado el nivel de integración que proveen para las aplicaciones, Los Web Services han crecido en popularidad y han comenzado a mejorar los procesos de negocios. De hecho, algunos postulan que los Web Services están generando la próxima evolución de la Web.

1.17. TECNOLOGÍAS EN UN WEB SERVICE

Los Web Services están contruidos con varias tecnologías que trabajan conjuntamente con los estándares que están emergiendo para asegurar la seguridad y operatividad, de modo de hacer realidad que el uso combinado de varios Web Services, independiente de la o las empresas que los proveen, este garantizado. A continuación, se describen brevemente los estándares que están ocupando los Web Services.

XML

Abreviación de Extensible Markup Language. El XML es una especificación desarrollada por W3C. Permite a los desarrolladores crear sus propios tags¹, que les permiten habilitar definiciones, transmisiones, validaciones, e interpretación de los datos entre aplicaciones y entre organizaciones.

W3C

W3C son las siglas de World Wide Web Consortium, y es una comunidad internacional donde los estados miembros trabajan para poder desarrollar estándares para el desarrollo web y así ayudar a un mejor desarrollo del Internet a nivel mundial.

W3C es reconocido a nivel mundial por ser la organización encargada de estandarizar El Lenguaje de Marcado de Hipertexto o mejor conocido como HTML, el cual es utilizado para el diseño y desarrollo de sitios web, dicho estándar inicio en el año de 1994 (es importante aclarar que antes de que W3C trabajara en HTML ya otras organizaciones habían desarrollado versiones previas).

Sin embargo, los objetivos de W3C van más allá del desarrollo de HTML y trabajan en otros proyectos a fin de poder brindar nuevas opciones para la elaboración de sitios web, tanto es así que en los últimos años han concentrados sus esfuerzos en el desarrollo de XHTML, una versión avanzada de HTML que pretende en los próximos años sustituirlo.

Por otro lado, W3C se centra en cuatro objetivos que se describen a continuación:

Web para todos: Accesibilidad para todo el mundo de forma global, sin barreras tecnológicas, limitaciones o culturales.

¹ Es un comando que va insertado en un documento, cuyo fin es especificar como el documento o parte del documento debe ser formateado o interpretado.

Web, Sobre Todo: Intenta añadir el máximo de dispositivos a la web, actualmente se está en un proceso de apertura donde las PDAs, Televisiones están iniciando su camino hacia la web.

Base de Conocimiento: Permitir que cada usuario pueda hacer un buen uso de la web.

Seguridad: Guiar al desarrollo de la web respetando las normas legales, comerciales y los aspectos sociales originados por esta tecnología.

Así de esta forma W3C se convierte en una organización de mucha importancia a nivel mundial, ya que los estándares que la misma presenta pueden influir en la forma en que el desarrollo web avance a nivel mundial.

JSON

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript, Standard ECMA-262 3rd Edition - diciembre 1999. JSON es un formato de texto que es completamente independiente del lenguaje, pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos entre diferentes sistemas operativos y lenguajes de programación.

JSON está constituido por dos estructuras:

- Una colección de pares de nombre/valor. En varios lenguajes esto es conocido como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.

- ✚ Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

Estas son estructuras universales; virtualmente todos los lenguajes de programación las soportan de una forma u otra. Es razonable que un formato de intercambio de datos que es independiente del lenguaje de programación se base en estas estructuras.

En JSON, se presentan de estas formas:

Un objeto es un conjunto desordenado de pares nombre/valor. Un objeto comienza con { (llave de apertura) y termina con } (llave de cierre). Cada nombre es seguido por: (dos puntos) y los pares nombre/valor están separados por, (coma).

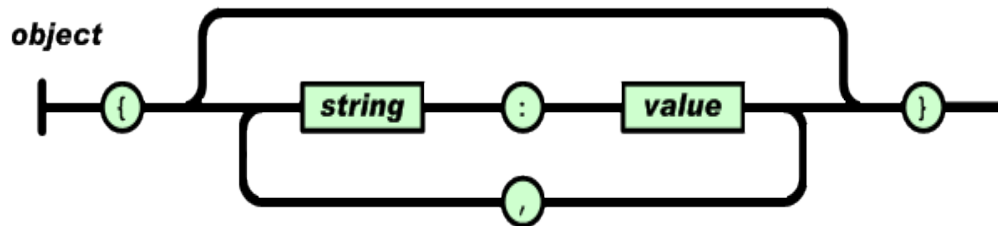


Figura 22: Object JSON

Un arreglo es una colección de valores. Un arreglo comienza con [(corchete izquierdo) y termina con] (corchete derecho). Los valores se separan por, (coma).

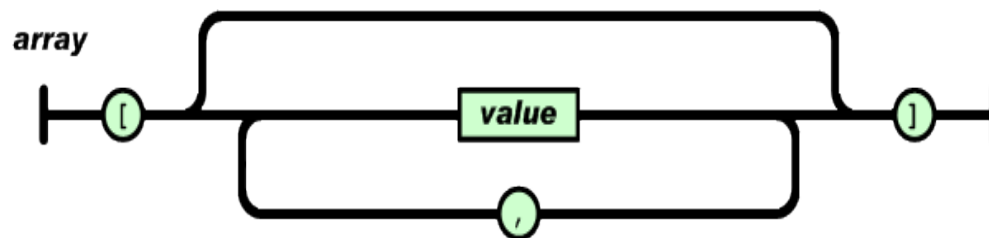


Figura 23: Array JSON

Un valor puede ser una cadena de caracteres con comillas dobles, o un número, o true o false o null, o un objeto o un arreglo. Estas estructuras pueden anidarse.

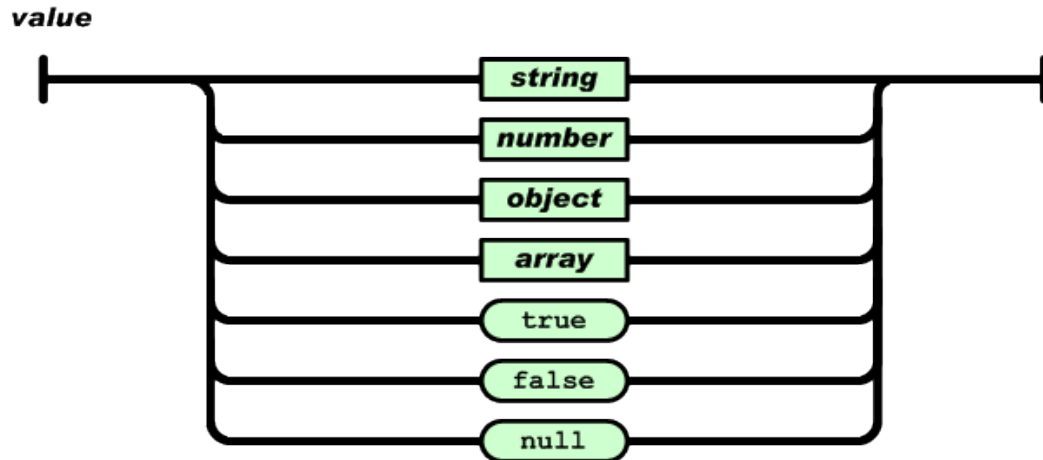


Figura 24: Value JSON

Una cadena de caracteres es una colección de cero o más caracteres Unicode, encerrados entre comillas dobles, usando barras divisorias invertidas como escape. Un carácter está representado por una cadena de caracteres de un único carácter. Una cadena de caracteres es parecida a una cadena de caracteres C o Java.

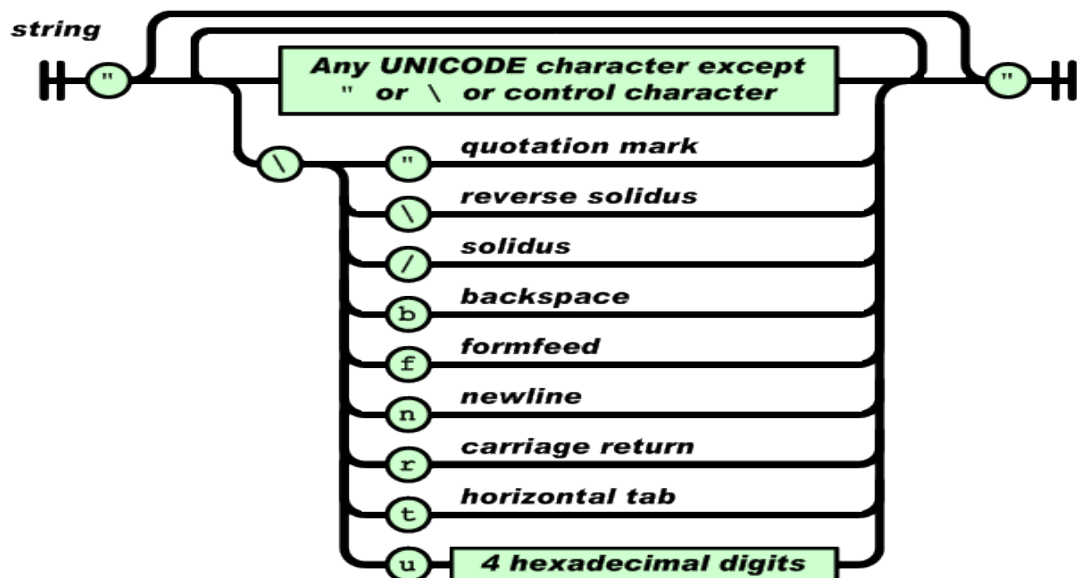


Figura 25: String JSON

Un número es similar a un número C o Java, excepto que no se usan los formatos octales y hexadecimales.

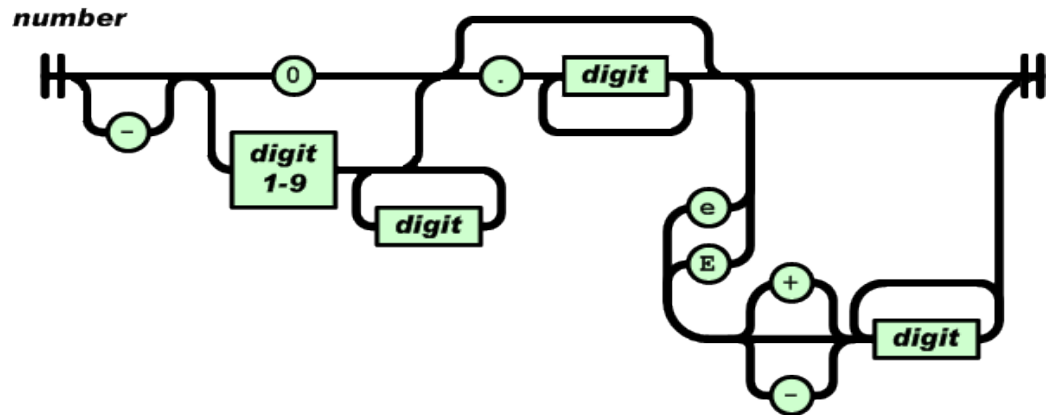


Figura 26: Number JSON

Formación de una cadena de JSON.

El principio básico es con pares atributo-valor, éstos deben estar encerrados entre llaves {,} que es lo que definen el inicio y el fin del objeto.

Ejemplo básico para formar un objeto JSON para los datos de una persona:

```
1 //Ejemplo de JSON para un objeto tipo Persona
2 {
3   "nombre":"Manuel de Jesús Gámez López",
4   "edad":33,
5   "nacionalidad":"Salvadoreño",
6   "altura":"199 cm",
7   "peso":99
8 }
```

Figura 27: Objeto JSON

Si bien es frecuente ver JSON posicionado contra XML, también es frecuente el uso de JSON y XML en la misma aplicación. Por ejemplo, una aplicación de cliente que integra datos de Google Maps con datos meteorológicos en SOAP hacen necesario soportar ambos formatos.

Sitio web online para generar cadenas, objetos o array: <http://json.parser.online.fr/>

SOAP

Abreviación de Simple Object Access Protocol, es un protocolo de mensajería construido en XML que se usa para codificar información de los requerimientos de los Web Services y para responder los mensajes antes de enviarlos por la red. Los mensajes SOAP son independientes de los sistemas operativos y pueden ser transportados por los protocolos que funcionan en la Internet, como ser: SMTP, MIME y HTTP.

WSDL

Abreviación de Web Services Description Language, es un lenguaje especificado en XML que se ocupa para definir los Web Service como colecciones de punto de comunicación capaces de intercambiar mensajes. El WSDL es parte integral de UDDI y parte del registro global de XML, en otras palabras, es un estándar de uso público (no se requiere pagar licencias para usarlo).

UDDI

Abreviación de Universal Description, Discovery and Integration. Es un directorio distribuido que opera en la Web que permite a las empresas publicar sus Web Services, para que otras empresas conozcan y utilicen los Web Services que publican, opera de manera análoga a las páginas amarillas.

1.18. API

Una API es un conjunto de funciones y procedimientos que cumplen una o muchas funciones con el fin de ser utilizadas por otro software. Las siglas API vienen del inglés Application Programming Interface y en español sería Interfaz de Programación de Aplicaciones.

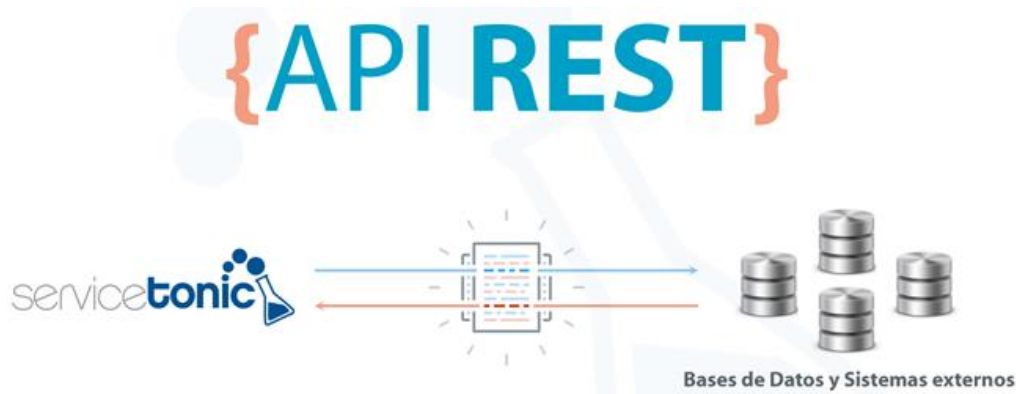


Figura 28: Esquema de una API

Las API conectan softwares diferentes.

Una “**interfaz**” es la forma en que dos aplicaciones o servicios se comunican entre sí. Lo hacen exponiendo al resto de aplicaciones el conjunto de servicios disponibles en cada una y cómo se deben acceder.

A la serie de estos servicios, se le denomina API. Por eso, las API sirven para que una aplicación pueda interactuar con otra.

1.19. EJEMPLOS DE COMO FUNCIONAN LAS API

Caso I: Agenda de reservas para MiRestaurante

El restaurante MiRestaurante ha desarrollado una aplicación para hacer reservas. Ahora, quieren hacer que los usuarios puedan hacer reservas desde la Agenda.

Entonces, escriben una API que expone un servicio para el intercambio de datos relacionados con las reservas.

En lenguaje natural

La aplicación de Agenda puede ser programada para que cuando detecta que la entrada contiene “comer en MiRestaurante” automáticamente utilice su API para

reservar. Así, lo único que se tendría que hacer es introducir la entrada en la agenda, y las aplicaciones se ocupan del resto mediante las API.

En lenguaje técnico

Si se guarda una entrada en la agenda que contenga: “Comer en MiRestaurante”, acceder mediante un POST, a la URL: <http://mirestaurante.com/api/reservar>. Así, enviará la identificación, el número de comensales, la fecha y hora de la reserva.

Caso II: Conexión entre dos Help Desks

La empresa MiEmpresa presta un servicio de soporte TI a diferentes clientes, y utiliza el help desk de ServiceTonic. Pero, las incidencias relacionadas con impresoras son derivadas a un proveedor externo, el cual usa su propia herramienta de help desk.

Lenguaje natural

Tanto ServiceTonic como la herramienta del proveedor tienen una API para gestionar las incidencias. Cuando un cliente abre una incidencia relacionada con impresoras en ServiceTonic, éste accede a la API de la herramienta del proveedor y crea un ticket. Cuando el proveedor finaliza la incidencia, su herramienta de help desk accede a ServiceTonic vía API, introduce la solución y cambia el estado de la incidencia. Así, cada empresa puede utilizar su propia herramienta de help desk. Lo único que hay que hacer es conectar ambas herramientas mediante sus respectivas API.

Lenguaje técnico

En ServiceTonic, definir una regla de negocio para que al crear una incidencia con categoría “Impresora” se acceda mediante un POST a la URL <http://proveedor.com/api/incidencias>, enviando la identificación y los datos de la incidencia. Y, en el help desk del proveedor, definir que cuando se solucione la

incidencia se acceda, mediante un PUT, a la URL <http://cliente.myservicetonic.com/api/v1/tickets/id>. Así, se enviará la solución y el nuevo estado al help desk de ServiceTonic.

1.20. API vs SERVICIO WEB

API y servicio web sirven como un medio de comunicación. La única diferencia es que un servicio web facilita la interacción entre dos máquinas a través de una red. Una API actúa como una interfaz entre dos aplicaciones diferentes para que puedan comunicarse entre sí. Una API es un método mediante el cual los proveedores externos pueden escribir programas que interactúan fácilmente con otros programas. Un servicio web está diseñado para tener una interfaz que se representa en un formato procesable por máquina que generalmente se especifica en el lenguaje de descripción de servicios web (WSDL). Normalmente, "HTTP" es el protocolo más utilizado para la comunicación. El servicio web también utiliza SOAP, REST y XML-RPC como medio de comunicación. API puede usar cualquier medio de comunicación para iniciar la interacción entre aplicaciones. Por ejemplo, las llamadas al sistema se invocan mediante interrupciones por la API del kernel de Linux.

Una API define exactamente los métodos para que un programa de software interactúe con el otro. Cuando esta acción implica el envío de datos a través de una red, los servicios web entran en escena. Una API generalmente implica llamar a funciones desde un programa de software.

En el caso de las aplicaciones web, la API utilizada está basada en la web. Las aplicaciones de escritorio, como las hojas de cálculo y los documentos de Word, utilizan API basadas en VBA y COM que no involucran el servicio web. Una aplicación de servidor como Joomla puede usar una API basada en PHP presente en el servidor que no requiere servicio web.

Un servicio web es simplemente una API envuelta en HTTP. Una API no siempre tiene que estar basada en la web. Una API consiste en un conjunto completo de reglas y especificaciones que debe seguir un programa de software para facilitar la interacción. Es posible que un servicio web no contenga un conjunto completo de especificaciones y, a veces, no pueda realizar todas las tareas que pueden ser posibles desde una API completa.

Las API se pueden exponer de varias formas, que incluyen: objetos COM, archivos DLL y .H en lenguaje de programación C / C ++, archivos JAR o RMI en Java, XML sobre HTTP, JSON sobre HTTP, etc. El método utilizado por Web El servicio para exponer la API es estrictamente a través de una red.

1.21. RAZONES PARA CREAR SERVICIOS WEB

- ✚ La principal razón para usar servicios Web es que se pueden utilizar con HTTP sobre Transmission Control Protocol (TCP) en el puerto de red 80. Dado que las organizaciones protegen sus redes mediante firewalls (que filtran y bloquean gran parte del tráfico de Internet), cierran casi todos los puertos TCP salvo el 80, que es, precisamente, el que usa los navegadores web.
- ✚ Los servicios Web utilizan este puerto, por la simple razón de que no resultan bloqueados. Es importante señalar que los servicios web se pueden utilizar sobre cualquier protocolo, sin embargo, TCP es el más común.
- ✚ Es posible brindar acceso a información (seleccionada) a terceros desde diversas herramientas y tecnologías.

1.22. RESUMEN API vs WEB SERVICE

La interfaz de programación de aplicaciones, conocida también por la sigla API del inglés Application Programming Interface, es un conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las bibliotecas de programación.

Ejemplos: API Google MAPS es una API que pone Google a disposición de todos los usuarios para hacer uso del servicio de mapas para integrar en diferentes aplicaciones para geolocalización o ubicación.

Por lo general, toda empresa que ofrece grandes servicios proporciona su propia API, por ejemplo: API FACEBOOK, API TWITTER, API INSTAGRAM, etc.

Un servicio web es un tipo de API que por lo general va a operar sobre HTTP en su gran mayoría de aplicaciones y para esto están las APIs que brindar servicio web.

- ✚ Todos los servicios web son API, pero no todas las API son servicios web.
- ✚ Servicios Web podrían no realizar todas las operaciones que una API podría llevar a cabo.
- ✚ Un servicio web o web service utiliza solo tres estilos de uso: SOAP, REST y XML-RPC para comunicación, mientras que API puede usar cualquier estilo para la comunicación.
- ✚ Un servicio web siempre necesita una red para su funcionamiento, mientras que una API no necesita una red para su funcionamiento.
- ✚ Una API facilita la interconexión directamente con una aplicación.
- ✚ Otros.

1.23. BASE DE DATOS

Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido; una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital, siendo este un componente electrónico, por tanto se ha desarrollado y se ofrece un amplio rango de soluciones al problema del almacenamiento de datos.

Existen programas denominados sistemas gestores de bases de datos, abreviado SGBD (del inglés Database Management System o DBMS), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de estos DBMS, así como su utilización y administración, se estudian dentro del ámbito de la informática.

Existe diversidad de alternativas de herramientas de software SGBD (O sistemas gestores de base de datos) para la creación de base de datos, cada uno con sus propias características y ventajas, de las cuales no se va a profundizar por no ser la prioridad de este trabajo de investigación. Por ejemplo: Microsoft SQL Server, MySQL, PostgreSQL, DB2, ORACLE, SYBASE, FIREBASE, etc.

El gestor de base de datos utilizado para la creación de la base de datos que se incorpora al sistema es MYSQL, el cual ha sido seleccionado gracias a la confiabilidad que brinda, experiencia de trabajo en este y a la diversidad de aplicaciones de diferente índole que lo utilizan, además, de ser un gestor de base de datos bien documentado y lo más importante es gratis. En la figura 30 se puede apreciar una captura del entorno web utilizado para la creación y pruebas realizadas con la base de datos diseñada para este trabajo de investigación.

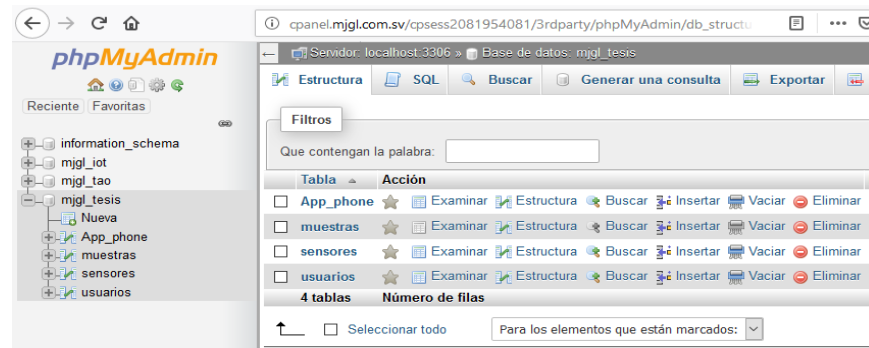


Figura 29: Tablas de la base de datos en MYSQL

1.24. DIAGRAMA DE BASE DE DATOS

En la figura 30 se presenta la estructura de la base de datos de pruebas utilizada para el almacenamiento de la información gestionada por el sistema electrónico a través de los diferentes sensores y demás elementos del sistema.

mjgl_tesis usuarios idusuario : int(11) nombre : varchar(90) apellido : varchar(70) email : varchar(100) usuario : varchar(45) clave : varchar(50) intentos : int(11) bloqueo : int(11) pregunta : varchar(100) respuesta : varchar(50) estado : varchar(40) fecha : timestamp isadmin : varchar(45) ultimoingreso : timestamp razon : varchar(200)	mjgl_tesis sensores id_sensor : int(11) descripcion : varchar(80) temperatura : decimal(10,2) humedad : decimal(10,2) luz1 : int(11) luz2 : int(11) luz3 : int(11) luz4 : int(11) luz5 : int(11) luz6 : int(11) luz7 : int(11) luz8 : int(11) test : int(11) fecha : date hora : varchar(15) mes : int(11)	mjgl_tesis App_phone codigo : int(11) descripcion : varchar(1000) numCelularCircuito : varchar(25) numCelularResponsable : varchar(25) fecha : varchar(25) hora : varchar(25)	mjgl_tesis muestras id : int(11) cantidad : int(11) numCelularCircuito : varchar(25) fecha : timestamp
---	---	--	---

Figura 30: Diagrama de contenedores para el sistema – Base de datos.

1.25. TARJETAS ELECTRÓNICAS DE PRUEBAS PARA CONEXIÓN DEL CONTROLADOR ARDUINO AL EXTERIOR (INTERNET)

Uno de los grandes retos que se afrontaba para el desarrollo de este trabajo de investigación, fue el hecho de cómo comunicar con exterior (INTERNET) el controlador del sistema para enviar la información procesada a la base de datos en el servidor de Internet, y para ello se realizaron una serie de pruebas con escudos o Shields (tarjetas electrónicas ETHERNET y WIFI) que son totalmente compatibles

con Arduino. Por tal razón, se consideró importante que se detallará la información más relevante de estas tarjetas por ser parte de la diversidad de pruebas efectuadas para lograr dicho objetivo.

Una vez lograda la conexión con la base de datos en servidor de Internet, poder tomar la información brindada por cada uno de los sensores del sistema para finalmente enviar los registros respectivos a la base de datos en el servidor.

A continuación, se hace una descripción de las diferentes posibilidades que se estudiaron para tal propósito a nivel de Hardware y Software Libre Arduino.

Existen otras posibilidades o alternativas de solución para este trabajo de investigación que no están precisamente basadas en controladores Arduino y que más bien se utilizan otros dispositivos como un Autómata programable LOGO V8 con el módulo de expansión CMR2020 para el caso de la comunicación y también es posible mediante dispositivos GSM/GPRS conocidos como RTU o Unidad de Control de Acceso Remoto).

1.26. PRINCIPALES CONTROLADORES PROBADOS CON ARDUINO PARA ESTABLECER COMUNICACIÓN CON EL EXTERIOR (INTERNET)

- ❖ Conexión a Internet por medios guiados: Para este caso se realizaron pruebas de conexión con la tarjeta SHIELD ETHERNET en sus distintas versiones.
- ❖ Conexión a Internet medios no guiados (WIFI): para este otro caso se experimentó con las tarjetas WIFI ESP8266 y ESP32.
- ❖ Conexión a Internet medios no guiados (WIFI): Por último, se estudió la comunicación vía GPRS con la tarjeta o SHIELD GSM/GPRS SIM900.

1.27. SHIELD ETHERNET W5100

El Arduino ethernet shield nos da la capacidad de conectar un Arduino a una red ethernet. Es la parte física que implementa la pila de protocolos TCP/IP.

Está basada en el chip ethernet Wiznet W5100. El Wiznet W5100 provee de una pila de red IP capaz de soportar TCP y UDP. Soporta hasta cuatro conexiones de sockets simultáneas. Usa la librería Ethernet para leer y escribir los flujos de datos que pasan por el puerto ethernet. Permite escribir sketches que se conecten a internet usando esta shield. Ver figura 31.

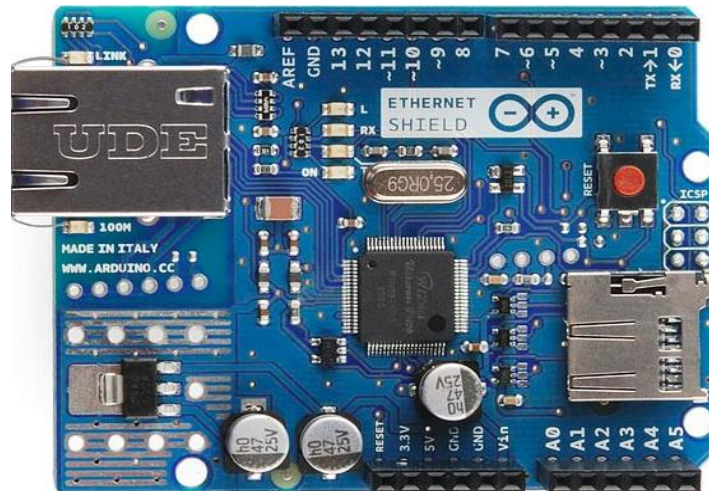


Figura 31: Tarjeta Shield o Escudo Ethernet W5100

El shield provee un conector ethernet estándar RJ45. La Ethernet Shield dispone de unos conectores que permiten conectar a su vez otras placas encima y apilarlas sobre la placa Arduino.

Arduino usa los pines digitales 10, 11, 12, y 13 (SPI) para comunicarse con el W5100 en la ethernet shield. Estos pines no pueden ser usados para e/s genéricas.

El botón de reset en la shield resetea ambos, el W5100 y la placa Arduino.

La shield contiene varios LEDs para información:

- ❖ **ON:** indica que la placa y la shield están alimentadas
- ❖ **LINK:** indica la presencia de un enlace de red y parpadea cuando la shield envía o recibe datos
- ❖ **100M:** indica la presencia de una conexión de red de 100 Mb/s (de forma opuesta a una de 10Mb/s)
- ❖ **RX:** parpadea cuando el shield recibe datos
- ❖ **TX:** parpadea cuando el shield envía datos

El jumper soldado marcado como “INT” puede ser conectado para permitir a la placa Arduino recibir notificaciones de eventos por interrupción desde el W5100, pero esto no está soportado por la librería Ethernet. El jumper conecta el pin INT del W5100 al pin digital 2 de Arduino.

El slot SD en la Shield usa la librería <http://arduino.cc/en/Reference/SD> para manejarlo. El propio chip W5100 incluye el manejo de tarjetas SD.

Para usar la Ethernet Shield solamente es necesario montarla sobre la placa Arduino. Para cargar los sketches a la placa con el Shield, conectarla al ordenador mediante el cable USB como se hace normalmente. Luego conectar el puerto Ethernet a un ordenador, switch o un router utilizando un cable Ethernet estándar (CAT5 o CAT6 con conectores RJ45). La conexión al ordenador puede requerir el uso de un cable cruzado (aunque muchos ordenadores actuales, pueden hacer el cruce de forma interna).

1.27.1. CARACTERÍSTICAS DE SHIELD ETHERNET W5100

- ❖ Opera a 5V suministrados desde la placa de Arduino
- ❖ El controlador ethernet es el W5100 con 16K de buffer interno. No consume memoria.

- ❖ El shield se comunica con el microcontrolador por el bus SPI, por lo tanto, para usarlo siempre debemos incluir la librería SPI.h:
<http://arduino.cc/en/Reference/SPI>.
- ❖ Soporta hasta 4 conexiones simultáneas
- ❖ Usar la librería Ethernet para manejar el shield:
<http://arduino.cc/en/Reference/Ethernet>
- ❖ El shield dispone de un lector de tarjetas micro-SD que puede ser usado para guardar ficheros y servirlos sobre la red. Para ello es necesaria la librería SD:
<http://arduino.cc/en/Reference/SD>
- ❖ Al trabajar con la SD, el pin 4 es usado como SS.

Arduino UNO se comunica con W5100 y la tarjeta SD usando el bus SPI a través del conector ICSP. Por este motivo los pines 10, 11, 12 y 13 en el UNO y los 50, 51, 52 y 53 en el Mega no podrán usarse. En ambas placas los pines 10 y 4 se usan para seleccionar el W5100 y la tarjeta SD. El Ethernet y el SD no pueden trabajar simultáneamente y se debe tener cuidado al usar ambos de forma conjunta.

1.28. ARDUINO ETHERNET SHIELD - W5200

Este Ethernet Shield de tamaño Arduino se basa en el chip Wiznet W5200 Ethernet y proporciona una manera fácil de poner Arduino en línea. Se adapta a todas las versiones de la placa principal de Arduino, por ejemplo, Arduino UNO y Arduino mega 2560. Para comenzar, simplemente basta con conectar este módulo a su placa Arduino, conectarlo a la red con un cable Ethernet CAT 5 y con unas pocas instrucciones simples, se puede explorar el mundo con Internet. La disposición del pasador también permite apilar otro escudo en la parte superior. Se coloca un radiador de aleación de aluminio en el chip W5200 para liberar el calor excesivo. Ver figura 33.



Figura 32: Tarjeta Shield o Escudo Ethernet W5200

1.28.1. CARACTERÍSTICAS DE ARDUINO ETHERNET SHIELD - W5200

- ❖ Directamente soportado por la biblioteca oficial de Ethernet Arduino.
- ❖ Ranura para tarjeta micro-SD adicional, que se puede usar para almacenar archivos y acceder a través de la biblioteca SD.
- ❖ Compatible con Arduino Uno y Mega (2560).
- ❖ El Wiznet W5200 proporciona una pila de red (IP) capaz de TCP y UDP.
- ❖ Admite hasta cuatro conexiones de socket simultáneas (ocho con la biblioteca actualizada).
- ❖ Utilice la biblioteca de Ethernet para escribir sketch que se conectan a Internet utilizando el escudo.
- ❖ Radiador de aleación de aluminio para liberar calor excesivo.

1.29. ARDUINO ETHERNET SHIELD 2 – W5500

El Arduino Ethernet Shield 2 conecta Arduino a Internet. Simplemente como en las versiones anteriores se requiere que enchufe este módulo en la placa Arduino, conectarlo a la red con un cable RJ45. Como siempre con Arduino, cada elemento de la plataforma (Hardware, Software y documentación) está disponible de forma gratuita y de código abierto. Ver figura 33.

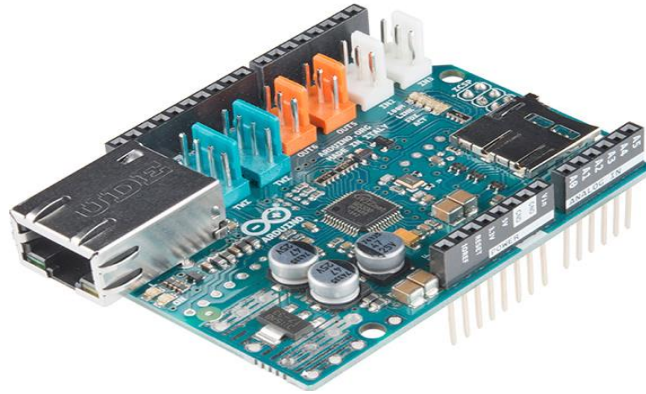


Figura 33: Tarjeta Shield o Escudo Ethernet W5500

1.29.1. CARACTERÍSTICAS DE ARDUINO ETHERNET SHIELD 2 – W5500

- ❖ Voltaje de Operación 5V.
- ❖ Controlador Ethernet W5500 con buffer interno de 23k
- ❖ Velocidad de conexión 10/100Mbps
- ❖ Conexión con Arduino por puerto SPI.
- ❖ Admite hasta 8 conexiones de socket simultáneas.
- ❖ Incorpora conectores nuevos para conexión inmediata de sensores.
- ❖ El shield dispone de un lector de tarjetas micro-SD que puede ser usado para guardar ficheros y servirlos sobre la red.
- ❖ Al trabajar con la SD, el pin 4 es usado como SS.
- ❖ Compatible con Arduino Uno y Mega (2560).
- ❖ El Wiznet W5500 proporciona una pila de red (IP) capaz de TCP y UDP.

1.30. VARIANTES ARDUINO ETHERNET SHIELD Rev3 CON PoE.

El Ethernet Shield es compatible con PoE (Power Over Ethernet) gracias a un módulo adicional que extrae la energía eléctrica del dispositivo de red Switch o HUB por medio del propio cable Ethernet que interconecta ambos dispositivos (Shield Ethernet con PoE y Switch). Ver figura 35.

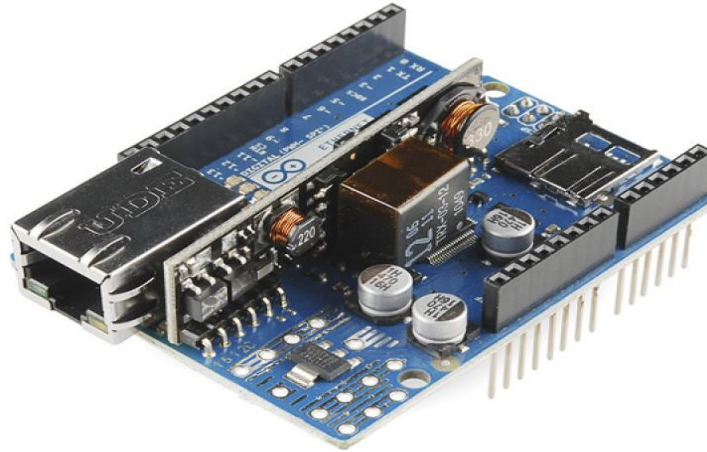


Figura 34: Power over Ethernet Shield

1.30.1. CARACTERÍSTICAS DE VARIANTES ARDUINO ETHERNET SHIELD Rev3 CON PoE.

- ❖ IEEE802.3af compliant
- ❖ Low output ripple and noise (100mVpp)
- ❖ Input voltage range 36V to 57V
- ❖ Overload and short-circuit protection
- ❖ 9V Output
- ❖ High efficiency DC/DC converter: typ 75% @ 50% load
- ❖ 1500V isolation (input to output).

1.31. DETALLES DE USO DEL WIZNET W5100

El integrado W5100 se conecta al Arduino mediante SPI. El chip Wiznet W5100 viene con tres métodos para controlar sus registros internos; los dos primeros son utilizar el bus paralelo directo o indirecto, el último es usar una conocida transferencia de datos en serie conocida como SPI (interfaz de periféricos seriales), para realizar las pruebas se utilizó la interfaz SPI para controlar el Wiznet.

La conexión SPI básica entre el módulo de red WIZNET WIZ811MJ y el microcontrolador Atmel AVR ATmega328 se muestra en la figura 35.

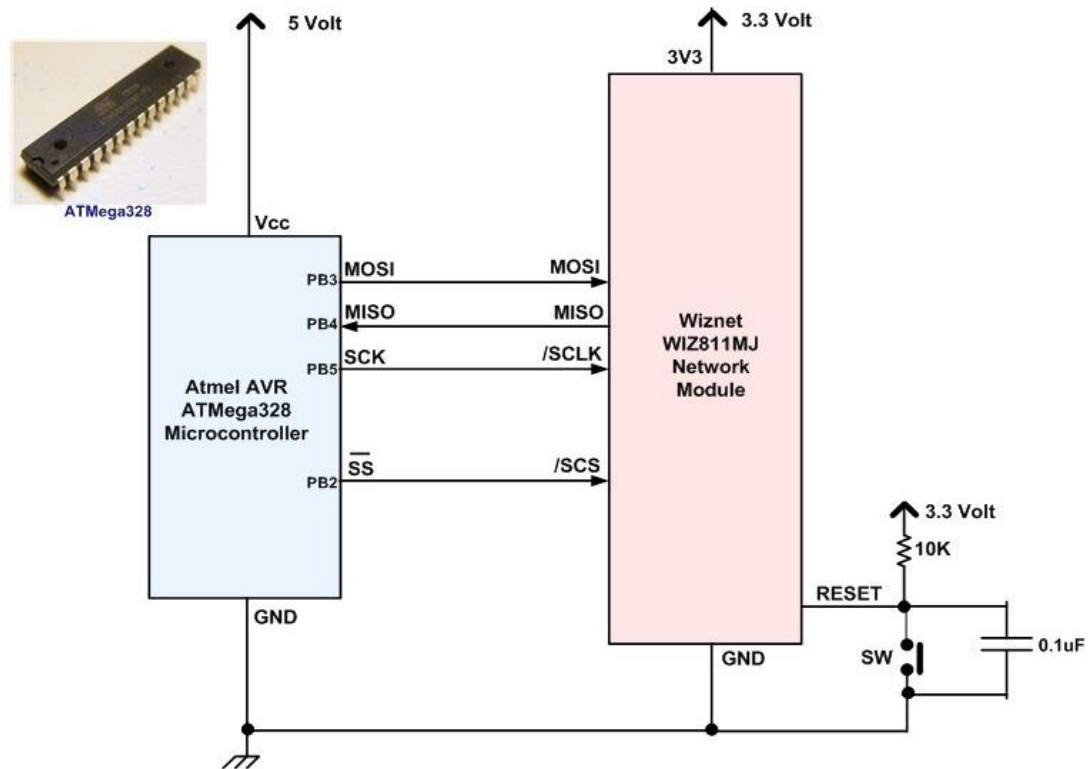


Figura 35: Atmel AVR ATmega328 and Wiznet WIZ811MJ Network Module
Simplify Schematic

El Wiznet W5100 Hardwired TCP / IP Protocol Chip

Básicamente, Wiznet W5100 implementa un completo estándar IEEE 802.3 (capa de enlace físico y de datos de Ethernet) y una potente pila de TCP / IP dentro del chip; esto hace que el chip Wiznet W5100 sea la opción adecuada para integrar el sistema integrado en Internet. La programación del chip Wiznet W5100 también es más sencilla, ya que solo se necesita escribir y leer desde y hacia los registros internos del W5100 para utilizar las características incorporadas del protocolo TCP / IP.

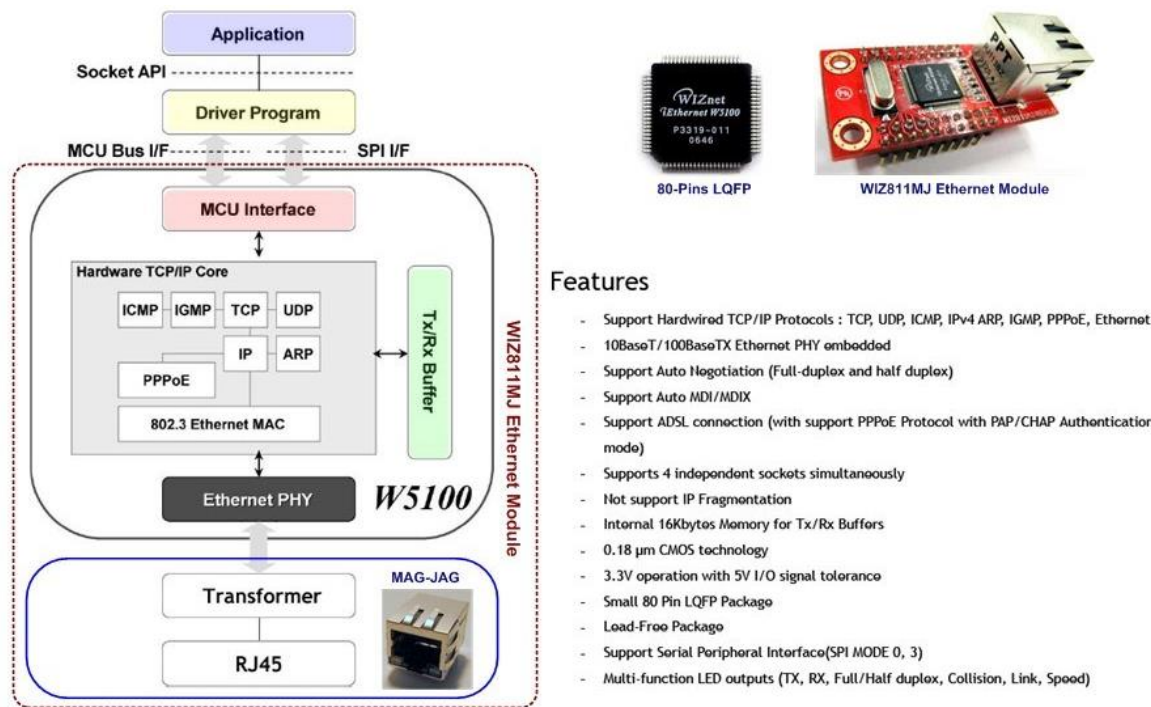


Figura 36: The Wiznet W5100 Hardwire TCP/IP Stack Chip and WIZ811MJ Ethernet Module

El Wiznet W5100 actuará como un dispositivo esclavo SPI controlado por microcontrolador ATmega328 como el SPI Maestro. Necesita el protocolo SPI al menos cuatro señales, MOSI (Master Out Serial In), MISO (Master In Serial Out), SCK (señal de reloj proporcionada por el maestro) y CS (the SPI slave chip select). El chip W5100 también proporciona el pin de interrupción.

Soporta Arduino IPv6.

El Shield oficial de Arduino basado en el WizNet 5100 implementa la pila de protocolos IPv4, por lo que está Shield no puede ser utilizada para implementar la pila IPv6.

Para implementar la pila IPv6, es necesario usar un Shield basado en Hardware que permita la gestión de las funciones de las capas de IP y Ethernet. Shields basados en el chip MicroChip ENC28J60 son adecuados para la implementación de IPv6. Esto requiere la implementación de una gestión de los estados TCP, resultando un código de Arduino más complejo.

1.32. PLACAS DE RED (WIFI)

La placa de desarrollo Node-MCU está basada en el popular chip que revolucionó el Wifi en sistemas embebidos, de la cual han surgido una serie de versiones con mayores prestaciones en funcionalidad y en Hardware. Se describen algunas de estas a continuación.

WIFI ESP8266: Con este sencillo módulo se puede realizar el prototipo de cualquier sistema para el IoT (Internet of things) en cuestión de horas. El concepto es muy similar al de Arduino, es decir, un microcontrolador conectado a través de un puente USB-Serial que interactúa con un software en la PC. Ver figura 38.

El IC - ESP8266 en el NodeMCU es más que un simple circuito integrado para WiFi. Se trata de un SoC (System on Chip) que integra en una sola pieza de silicio un procesador de aplicaciones con la electrónica necesaria para la comunicación por (Wifi). Esta placa permite aprovechar el procesador que está dentro del ESP8266 y realizar software que corre en él, no solamente usarlo como un puente entre un microcontrolador y la red wifi.

Esta placa viene cargada con el firmware NodeMCU, sin embargo, también puede usarse como una excelente plataforma para desarrollar, evaluar y experimentar con otros firmwares para el ESP8266.

Existe diversidad de tarjetas electrónicas (Controladores) que se puede utilizar para desarrollar proyectos de este tipo, se mencionan las tarjetas WIFI con las que se realizaron pruebas:

- ❖ **NODE-MCU ESP8266:** La placa de desarrollo NodeMCU está basada en el popular chip que revolucionó el Wifi en sistemas embebidos. Ver figura 39.

- ❖ **ESP32:** Es un dispositivo mucho más potente. No solo porque sea más rápido sino porque está diseñado y pensando en que sea un microcontrolador para el IoT (Internet de las Cosas o Internet of Things). Ver figura 40.



Figura 37: ESP8266

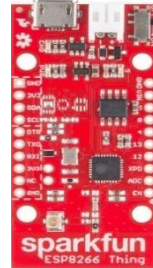


Figura 38: NodeMCU-esp8266



Figura 39: Esp32

1.32.1. CARACTERÍSTICAS DE PLACAS DE RED (WIFI)

Característica	ESP8266	ESP32
Procesador	Tensilica LX106 32 bit. A 80 MHz (hasta 160 MHz)	Tensilica Xtensa LX6 32 bit Dual-Core. A 160 MHz (hasta 240 MHz)
Memoria RAM	80 kB (40 kB disponibles)	520 kB
Memoria Flash	Hasta 4 MB	Hasta 16 MB
ROM	No	448 kB
Alimentación	3.0 a 3.6 V	2.2 a 3.6 V
Rango de temperaturas	-40 °C a 125°C	-40°C a 125°C
Consumo de corriente	80 mA (Promedio). 225 mA máximo.	80 mA (promedio). 225 mA máximo.
WIFI	802.11 b/g/n (hasta +20 dBm) WEP, WPA.	802.11 b/g/n (hasta +20 dBm) WEP, WPA.

Tabla 3: Comparación de especificaciones de tarjetas WIFI.

En la tabla 3, se presentan las principales características técnicas de los módulos WIFI. Ambas tarjetas WIFI se pueden conectar a un Access Point (Punto de Acceso) para poder establecer conexión vía red (TCP / IP) y ser un host más en la red. Son



capaces de generar un AP (Access Point o Punto de Acceso) propio y responder como un servidor web peticiones vía web a través de un navegador web conociendo su respectiva IP de configuración. La forma de trabajo de la tarjeta WIFI se define a través de la programación de la misma.

1.33. TECNOLOGÍA GSM/GPRS

En este apartado se trata de realizar una aproximación a la tecnología que se empleó en las comunicaciones del sistema. Para ello, se recurre a la segunda generación de telefonía móvil y a una serie de evoluciones que permiten la transferencia de datos a través de Internet. Del mismo modo, se presenta la tecnología GPRS considerada como la 2.5 G y que permite la transferencia de datos a través de Internet. A la vez se hace mención acerca de las características de la tarjeta electrónica Shield SIM900 GSM / GPRS.

1.34. GSM (GROUPE SPECIAL MOBILE)

Se trata de un estándar de comunicaciones, también denominado a partir de 1991 como Sistema Global de Comunicaciones Móviles, que a lo largo de diferentes fases ha ido evolucionando dentro de lo que se ha denominado como telefonía móvil de segunda generación. En Europa, el estándar GSM usa las bandas de frecuencia de 900 MHz y 1800 MHz. Sin embargo, en los estados unidos se usa la banda de frecuencia de 1900 MHz. Por ello los teléfonos portátiles que funcionan tanto en Europa como en los Estados Unidos se llaman tribanda y aquellos que funcionan solo en Europa se denominan bi-banda.

Por un lado, el estándar GSM soporta un tráfico máximo de 9,6 kbps, que permite transmisiones de voz y de datos digitales de volumen bajo: mensajes de texto (SMS, Servicio de Mensajes Cortos) o mensajes multimedia (MMS, Servicio de Mensajes Multimedia). Además, hay que mencionar que es un sistema totalmente digital, no como su antecesor que era análogo, donde se siguen empleando celdas para cubrir el territorio, donde cada celda contiene una estación base que da cobertura.

Por otro lado, se pueden especificar una serie de servicios, la arquitectura y evolución de GSM:

1. Servicios:

▪ Básicos:

Teleservicios: Dan un servicio completo dentro de la red GSM. Proporcionan capacidad para comunicarse. Es el caso de la voz, fax, mensajes cortos (SMS, EMS, MMS), llamadas de emergencia, entre otros.

Portadores (bearer): GSM proporciona solo las capas bajas de la estructura OSI. Proporcionan capacidad de transmitir señales. Por ejemplo: Transmisiones de datos entre PDA (Personal Digital Assistant) o PC mediante GSM.

▪ Suplementarios:

Son un valor añadido a los servicios portadores o a los teleservicios: limitaciones de llamadas salientes, multi conferencia, redirección de llamadas, identificación de quien llama, restricción de identificación, etc.

2. Arquitectura:

▪ MS (Mobile Station o Estación Móvil)

- TE (Terminal Equipment): Genera los datos.
- MT (Mobile Termination): Es el conversor de señal eléctrica a radio (Y al revés).
- TA (Terminal Adaptor): Es la interfaz entre el TE y el MT.
- SIM (Subscriber Identify Module): Es la tarjeta que se pone dentro del terminal.

▪ BSS (Base Station Subsystem o Subsistema de Estaciones Base).

- BTS (Base Transceiver Station): Son las estaciones base. Entre otros elementos, aquí se encuentran las antenas y los amplificadores de potencia.

- BSC (Base Station Controller): Controlador de Estaciones Base.
- NSS (Network and Switching Subsystem o Subsistema de red).
- OSS (Operation support subsystem o Subsistema de gestión):

3. Evolución:

- **GSM fase 1 (1990)**
 - Uso del terminal y la tarjeta en cualquier lugar de Europa (Itinerancia o roaming).
 - Capacidad elevada, para que no se sature rápido.
 - Servicios de voz (telefonía) y datos (hasta 9.600 kbps).
 - Frecuencias: 890-915 MHz y 935-960 MHz.
 - El costo está en la red, y no en los terminales.
 - Cifrado de información, para ofrecer seguridad.
 - Uso del sistema de señalización internacional CCITT núm. 7. Lo usan los operadores para comunicarse entre sí o entre elementos de la misma red.
 - Servicio de Mensajes Cortos (SMS).
 - Número único para llamada de emergencia.
 - Fax.
 - Restricción de llamadas.
- **GSM fase 2 (1995)**
 - Ampliación del espectro: 880-915 y 925-960 MHz.
 - Permite utilizar un codificador más rápido (half-rate speech coder). Reduce el número de bits necesario para codificar la voz, manteniendo una calidad aceptable.
 - Identificación de quien llama, llamada en espera, etc.
 - Opciones específicas para grupos de usuarios: llamada de grupo, llamada para todos.

▪ **GSM fase 2+ (1996 en adelante)**

- Terminales duales GSM/DCS.
- Interconexión con DECT (Digital European Cordless Telephony).
- CAMEL (Customized Application for Mobile Network Enhanced Logic).
- Amplificación de la funcionalidad del SIM.
- Permite llamadas de emergencia combinando un terminal GSM y un receptor GPS.
- GPRS (1997).
- UMTS (1999).
- HSDPA (3GPP reléase 5, 2001).
- HSUPA (3GPP reléase 6).
- LTE (3GPP reléase 8).

Finalmente, mencionar que es una red que sigue desplegada conviviendo con las redes 3G y 4G, de forma que en entornos rurales se sigue empleando este sistema debido a su mayor alcance y a la densidad de población.

1.35. TECNOLOGÍA 5G

Actualmente se habla y ya se está desarrollando la tecnología 5G (Probablemente se implementará a partir del 2020) la cual promete abrir la puerta a nuevos procedimientos en áreas como: quirúrgicos, medios de transporte más seguros y comunicación instantánea para los servicios de emergencia y socorro, etc.

La red inalámbrica de la próxima (5ta) generación abordará la evolución más allá del internet móvil, y alcanzará al Internet de las Cosas ²masivo hacia 2020. La evolución

² Es una red de objetos digitalizados y dotados con características particulares de computo, comunicación y sensorial, los cuales están interconectados a la nube (Internet) y/o entre ellos

más notable en comparación con las redes 4G y 4.5G (LTE avanzado) actuales es que, aparte del aumento en la velocidad de los datos, los nuevos casos de uso del Internet de las Cosas y de la comunicación requerirán nuevos tipos de desempeño mejorado; como la “latencia baja”, que brinda una interacción en tiempo real a los servicios que utilizan la nube, lo que resulta clave, por ejemplo, para los vehículos autónomos. Además, el bajo consumo de energía permitirá que los objetos conectados funcionen durante meses o años sin la necesidad de la intervención humana.

A diferencia de los servicios actuales del Internet de las Cosas que sacrifican rendimiento para sacar el máximo provecho a las tecnologías inalámbricas existentes (3G, 4G, WiFi, Bluetooth, Zigbee, etc.), las redes 5G estarán diseñadas para alcanzar el nivel de rendimiento que necesita el Internet de las Cosas masivo. Esto hará posible que se perciba un mundo completamente ubicuo y conectado.

1.35.1 CARÁCTERÍSTICAS DE TECNOLOGIA 5G.

- ✚ Una tasa de datos de hasta 10Gbps - > de 10 a 100 veces mejor que las redes 4G y 4.5G.
- ✚ Latencia de 1 milisegundo.
- ✚ Una banda ancha 1000 veces más rápida por unidad de área.
- ✚ Hasta 100 dispositivos más conectados por unidad de área (en comparación con las redes 4G LTE).
- ✚ Disponibilidad del 99.999%.
- ✚ Cobertura del 100%.
- ✚ Reducción del 90% en el consumo de energía de la red.

Las redes de telecomunicaciones de la próxima generación (5G) llegarán al mercado hacia 2020. Más allá de las mejoras en la velocidad, se espera que la tecnología 5G

realizando una tarea específica en el proceso, proporcionando información de interés al centro de datos, la cual normalmente es enviada y recibida en tiempo real para la toma de decisiones y acciones oportunas por humanos o por un sistema M2M (Machine to Machine).

desate todo un ecosistema del Internet de las Cosas masivo en el que las redes pueden satisfacer las necesidades de comunicación de miles de millones de dispositivos conectados, con un equilibrio justo entre velocidad, latencia y costo.

1.36. GPRS (SERVICIO GENERAL DE PAQUETES DE RADIO)

Es una evolución del estándar GSM por lo que en algunos casos se denomina GSM++ o GSM 2+. Dado que es un estándar de telefonía de segunda generación que permite una transición hacia la tercera generación (3G), el estándar GPRS por lo general se clasifica como 2.5 G. Permite la transferencia de datos o paquetes con una tasa de datos teórico de alrededor de 171,2 kbps hasta 114 kbps en la práctica. Gracias a su modo de transferencia en paquetes, las transmisiones de datos solo usan la red cuando es necesario. Por lo tanto, el estándar GPRS permite que el usuario reciba facturas por volumen de datos en lugar de la duración de la conexión, lo que significa especialmente que el usuario puede permanecer conectado sin costo adicional. Para el transporte de voz, el estándar GPRS emplea la arquitectura de red GSM y provee acceso a la red de datos (Especialmente Internet) por medio del protocolo IP (Internet Protocol o Protocolo de Internet) o del protocolo X.25. Además, admite características nuevas que no están disponibles en el estándar GSM y que se pueden clasificar en los siguientes tipos de servicios:

- ✚ Servicio de punto a punto (PTP): Es la capacidad de conectarse en modo cliente-servidor a un equipo en una red IP.
- ✚ Servicio de punto a multipunto (PTMP): Constituye la capacidad de enviar paquetes a un grupo de destinatarios (Multidifusión).
- ✚ Servicio de Mensajes Cortos (SMS).

Por otro lado, GSM fue pensado para transmitir voz, aunque ofrece la posibilidad de transmitir datos de hasta 9.600 bps. Se desarrollan tres sistemas que permitían obtener más velocidad aplicando algunas modificaciones a la red GSM. Estos sistemas son el HSCSD, GPRS y EDGE, y son considerados sistemas 2.5 G. Con todo esto queda

claro que GPRS no es la única tecnología de intermedia entre la segunda y tercera generación de telefonía móvil. Además, este sistema es útil para aplicaciones que requieren muchas transmisiones de pocos datos o pocas transmisiones de un volumen medio de datos, dado que GPRS permite más velocidad que GSM.

1.37. SHIELD GSM/GPRS SIM900

Conexión TCP a través de GPRS usando SIM900 y comandos AT

Dentro de las posibilidades que se tienen para establecer conexión con el servidor y la base de datos en la nube (Internet), una es la de tomar la información captada por los diferentes sensores del sistema y generar un paquete o trama para enviar esta data vía GPRS o Servicio General de Paquetes de Radio.

Las primeras tarjetas explicadas anteriormente (ETHERNET Y WIFI) obligadamente tendrían que conectarse a la nube (Internet) a través de varios elementos intermediarios.

El principal dispositivo que se requiere tener es un ROUTER, el cual, debe tener conexión a la red Internet a través de un ISP (Proveedor de Servicio de Internet). Además, requiere que se configure adecuadamente desde la web de administración del mismo, para generar el puerto lógico (IP: 80) por donde se escuchan y responden a peticiones entrantes y salientes de la información que se procesa entre dispositivo emisor y receptor.

Este último método basado en la red GPRS (Servicio General de Paquetes de Radio) no requiere de mayores configuraciones y de otros dispositivos intermediarios a nivel local. Basta con saber nombre del punto de acceso o APN, usuario y contraseña de la operadora telefónica que se esté usando.

APN (Access Point Name) es el nombre de un punto de acceso que hay que configurar para que el dispositivo (Shield GSM/GPRS SIM900) se pueda conectar a Internet

usando las redes de la operadora que se esté usando.

El Módulo GSM/GPRS es una placa de desarrollo compatible para integrar con Arduino uno R3, Arduino Mega y otras tarjetas o microcontroladores que soportan o trabajan con el protocolo de comunicación serial (Norma RS232). La comunicación de este Shield SIM900 GPRS se trabaja en las frecuencias 850/900/1800/1900 MHz.

Los módems GSM / GPRS se están volviendo muy comunes en estos días, ya que los precios son cada vez más baratos. Además de poder enviar/recibir SMS y funciones de llamada a los proyectos, también es posible establecer comunicación a Internet a través de TCP. Ver figura 40.

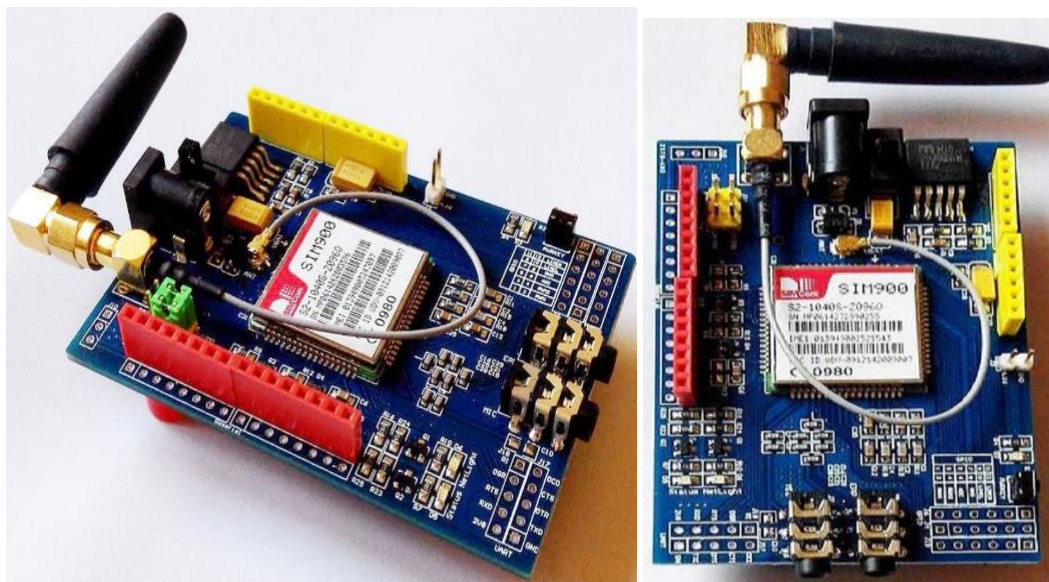


Figura 40: Shield GSM/GPRS SIM900

1.37.1. CARACTERÍSTICAS DE GPRS

- ❖ El escudo GPRS se basa en el módulo SIM900 del SIMCOM y compatible con Arduino y sus clones.
- ❖ El escudo GPRS proporciona una manera de comunicarse a través de la red de telefonía celular GSM.
- ❖ El escudo le permite trabajar con SMS, MMS, GPRS y audio a través de UART mediante el envío de comandos AT (GSM 07.07, 07.05 y SIMCOM realizada, Comandos AT).
- ❖ El escudo también tiene los 12 GPIO, 2 PWM y un ADC del módulo SIM900 (Son toda lógica 2V8) presentes a bordo.
- ❖ Quad-Band 850/900/1800/1900 MHz - Funciona en redes GSM en todos los países de todo el mundo.
- ❖ GPRS multi-slot clase 10/8.
- ❖ GPRS estación móvil de clase B.
- ❖ Cumple con la norma GSM fase 2/2 +.
- ❖ Clase 4 (2 W @ 850/900 MHz).
- ❖ Clase 1 (1 W @ 1800 / 1900MHz).
- ❖ Control a través de comandos AT - Comandos estándar: GSM 07.07 y 07.05.
- ❖ Enhanced Comandos: Comandos AT SIMCOM.
- ❖ Servicio de mensajes cortos - por lo que se puede enviar pequeñas cantidades de datos a través de la red (ASCII o hexadecimal en bruto).
- ❖ TCP Embedded / pila UDP - le permite cargar datos a un servidor web.
- ❖ RTC compatible.
- ❖ Puerto serie seleccionable.
- ❖ Jacks para altavoces y auriculares.
- ❖ Bajo consumo de energía - 1,5 mA (modo de espera).
- ❖ Rango de temperatura industrial - -40 ° C a +85 ° C
- ❖ Tamaño: aprox. 75 mm * 55 mm * 10 mm.
- ❖ Material: Metal...

1.37.2. COMANDOS AT PARA CONEXIÓN TCP/UDP

En la tabla 4, se presenta la lista de comandos AT utilizados para testing de conexión TCP / UDP con su correspondiente respuesta de ejemplo por parte del Shield SIM900 y una breve descripción de la funcionalidad que tiene cada comando. Se recomienda consultar el manual de comandos AT de su módem para más detalles.

Comando AT	Respuesta	Descripción
AT	OK	Comando de prueba. Devuelve OK si hay comunicación.
AT+CGATT?	+CGATT:n	Comprueba si GPRS está adjunta N=1 si lo está.
AT+CIPMUX=n	OK	Comando para modo de conexión: n=0 una sola conexión y n=1 para múltiples conexiones.
AT+CSTT="apn","username","pass"	OK	Establecer APN, nombre de usuario y contraseña.
AT+CIICR	OK	Se active el perfil de datos inalámbricos.
AT+CIFSR	Dirección IP	Se obtiene la dirección IP si se establece conexión.
AT+CIPSTART="TYPE" , "domain", "port"	Conexión	Inicia conexión TCP o UDP
AT+CIPSEND	>	Envía Datos TCP o UDP
AT+CIPCLOSE	OK	Cierra la conexión TCP o UDP
AT+CIPSHUT	SHUT OK	Cierra el contexto PDP del GPRS

Tabla 4. Comandos AT para la conexión TCP / UDP

CAPÍTULO II

DESCRIPCIÓN FUNCIONAL DEL SISTEMA

2. DESCRIPCIÓN FUNCIONAL DEL SISTEMA

2.1. ENTRADAS Y SALIDAS DEL SISTEMA ANDROID

2.2. INTRODUCCIÓN

En este capítulo se detallan las entradas y salidas que el proyecto incluye, realizando una descripción de los elementos que lo forman y sus características. Como todo sistema que interactúe con el entorno, se requiere un conjunto de sensores que permitan al bloque de control tener un conocimiento preciso de los parámetros que requiere para realizar el proceso que se ha codificado. De igual modo, para realizar dichos procesos se requiere un conjunto de actuadores que lleven a cabo estas tareas. Con todo, no es un objetivo de este trabajo de tesis entrar en detalle sobre los actuadores empleados o los sistemas que se activan, sino simplemente realizar una pequeña puesta en escena de estos y dar a conocer que serán activados cuando se cumplan una serie de requisitos.

2.3. SENSOR DE TEMPERATURA Y HUMEDAD RELATIVA

La captación de los valores de temperatura y humedad relativa se ha realizado con un único accesorio, el sensor DHT22. Los sensores DHT11 o DHT22 son unos pequeños dispositivos que permiten medir la temperatura y la humedad relativa del ambiente. A diferencia de otros sensores, éstos han de conectarse a pines digitales, ya que la señal de salida es digital al integrar un pequeño microcontrolador para hacer el tratamiento de la señal.

Los sensores de la gama DHT se componen de un sensor capacitivo para medir la humedad y de un termistor, ambos sensores ya calibrados por lo que no es necesario añadir ningún circuito de tratamiento de señal. Esto es, sin duda, una ventaja ya que simplifican las conexiones a realizar en la placa. Además, como los DHT han sido calibrados en laboratorios, presentan una gran fiabilidad.

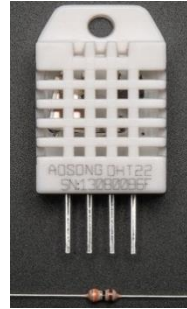


Figura 41: Sensor DHT22

Ambos sensores funcionan con ciclos de operación de duración determinada (1s en el caso del DHT11 y 2s en el caso del DHT22). En este tiempo, el microcontrolador externo (Arduino), y el microcontrolador que lleva integrado el sensor, se hablan entre sí de la siguiente manera:

- ❖ El microcontrolador (Arduino) inicia la comunicación.
- ❖ El sensor responde estableciendo un nivel bajo de 80us y un nivel alto de 80us.
- ❖ El sensor envía 5 bytes. Se produce el handshaking (establecimiento de comunicación).

En la tabla 5, se muestra las principales características de los dos modelos de sensores.

Parámetro	DHT11	DHT22
Alimentación	$3Vdc \leq Vcc \leq 5Vdc$	$3.3Vdc \leq Vcc \leq 6Vdc$
Señal de Salida	Digital	Digital
Rango de medida de Temperatura	De 0 a 50 °C	De -40°C a 80 °C
Precisión Temperatura	± 2 °C	$< \pm 0.5$ °C
Resolución Temperatura	0.1°C	0.1°C
Rango de medida Humedad	De 20% a 90% RH	De 0 a 100% RH
Precisión Humedad	4% RH	2% RH

Resolución Humedad	1%RH	0.1%RH
Tiempo de sensado	1s	2s
Tamaño	12 x 15.5 x 5.5mm	14 x 18 x 5.5mm

Tabla 5. Sensores de temperatura y humedad relativa, DHT11 VS DHT22.

En lo que se refiere al pinout o terminales, los pines del DHT22 siguen el mismo orden. De izquierda a derecha se tiene:

1. VCC: Alimentación del sensor
2. Señal: Información de temperatura y humedad
3. NC: No Conexión
4. GND: Masa del circuito

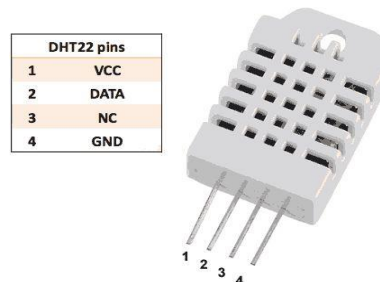


Figura 42: Terminales del Sensor DHT22

En la figura 42 se muestra un esquema de conexión y un ejemplo, donde se ve cómo se alimenta desde Arduino al sensor a través de los pines GND y VCC del mismo. Por otro lado, la salida OUTPUT se conecta a una entrada digital de Arduino, siendo necesario poner una resistencia de 10K (Pull-up) entre VCC y el PIN OUTPUT. También se hace mención que el sensor DHT22 usa su propio sistema de comunicación bidireccional mediante un único conductor, empleando señales temporizadas. Así, en cada envío de medición, el sensor envía un total de 40 bits, en 4 ms. Estos 40 bits corresponden con 2 Bytes para la medición de la humedad relativa, 2 Bytes para la medición de la temperatura, más un Byte final para la comprobación de errores (8 bit integral RH data + 8 bit decimal RH data + 8 bit integral T data + 8 bit decimal T data + 8 bit check sum). Finalmente se hace mención que se requiere la

librería DHT.h, para poder trabajar en el IDE de Arduino con más facilidad y así poder realizar las mediciones de temperatura y humedad relativa ambiente, pudiendo de esta manera facilitar la programación del sensor.

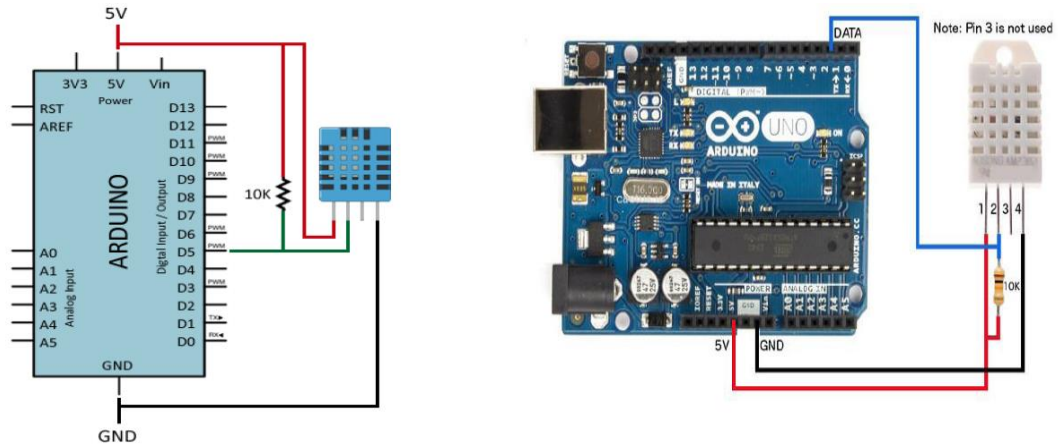


Figura 43: Conexión Sensor DHT22 con Arduino Uno

Un ejemplo de adquisición básica de la información captada del ambiente por el sensor DHT22 es el que se puede apreciar en la figura 44.

```

sketch_jan28a Arduino 1.8.8
Archivo Editar Programa Herramientas Ayuda

sketch_jan28a $
1 #include "DHT.h"
2 #define DHTYPE DHT22
3 const int DHTPin = 5;
4 DHT dht(DHTPin, DHTYPE);
5 void setup() {
6     Serial.begin(9600);
7     Serial.println("DHTxx test!");
8     dht.begin();
9 }
10
11 void loop() {
12     delay(2000);
13     float h = dht.readHumidity();
14     float t = dht.readTemperature();
15
16     if(isnan(h) || isnan(t)){
17         Serial.println("Failed to read from DHT");
18         return;
19     }
20
21     Serial.print("Humidity: ");
22     Serial.print(h);
23     Serial.print("Temperature: ");
24     Serial.print(t);
25     Serial.print(" * C");
26 }
    
```

Figura 44: Código básico de programación del sensor DHT22 con Arduino Uno

2.4. SENSOR LDR (Light-Dependent Resistor)

Se trata de un sensor que actúa como una resistencia variable en función de la luz que capta. A mayor intensidad de luz, menor resistencia y viceversa. En la figura 45 y 46 se presentan los montajes realizados para las pruebas efectuadas con respecto a la detección de luz. Además, en la figura 47 se puede apreciar el código fuente ingresado al controlador Arduino para medir dicha variable de luz.

Para conocer la cantidad de luz que el sensor capta en cierto ambiente, fue necesario medir la tensión de salida del mismo. Para ello se utilizó un divisor de tensión, colocando el punto de lectura para V_{out} entre ambas resistencias.

De esta forma se utilizó la ecuación del divisor de tensión resistivo (2 resistencias).

$$V_{out} = \frac{R_2}{R_1 + R_2} \cdot V_{in}$$

Donde:

V_{out} : Es el voltaje leído por el PIN analógico de Arduino y será convertido a un valor digital.

V_{in} : Es el voltaje de entrada (5V).

R_2 : Es el voltaje de la resistencia fija colocada (10 k Ω).

R_1 : Es el valor resistivo del sensor LDR.

A medida que el valor del sensor LDR varía, obtendremos una fracción mayor o menor del voltaje de entrada V_{in} . Ver figura 11.

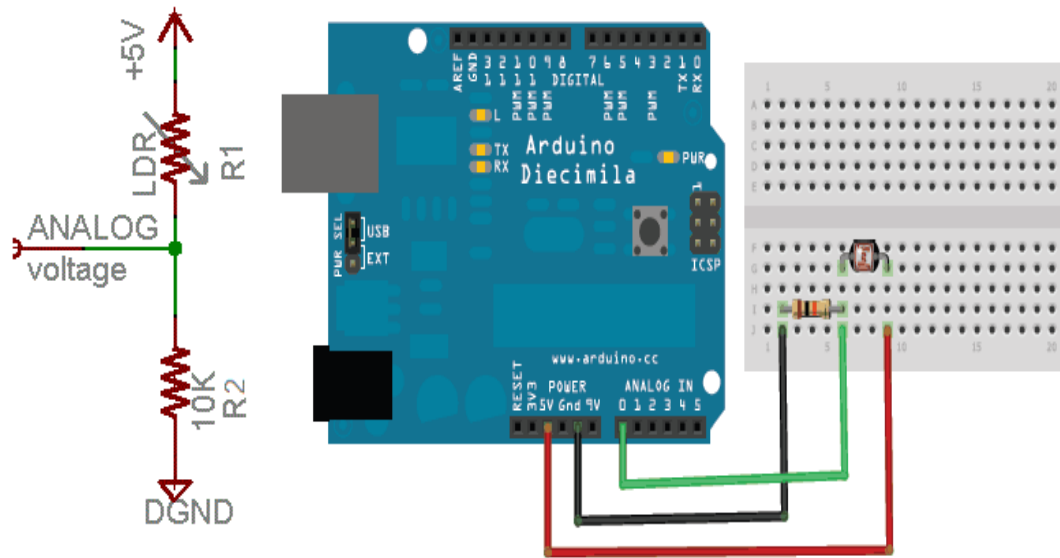


Figura 45: Diagrama de circuito LDR con Arduino (con Breadboard)

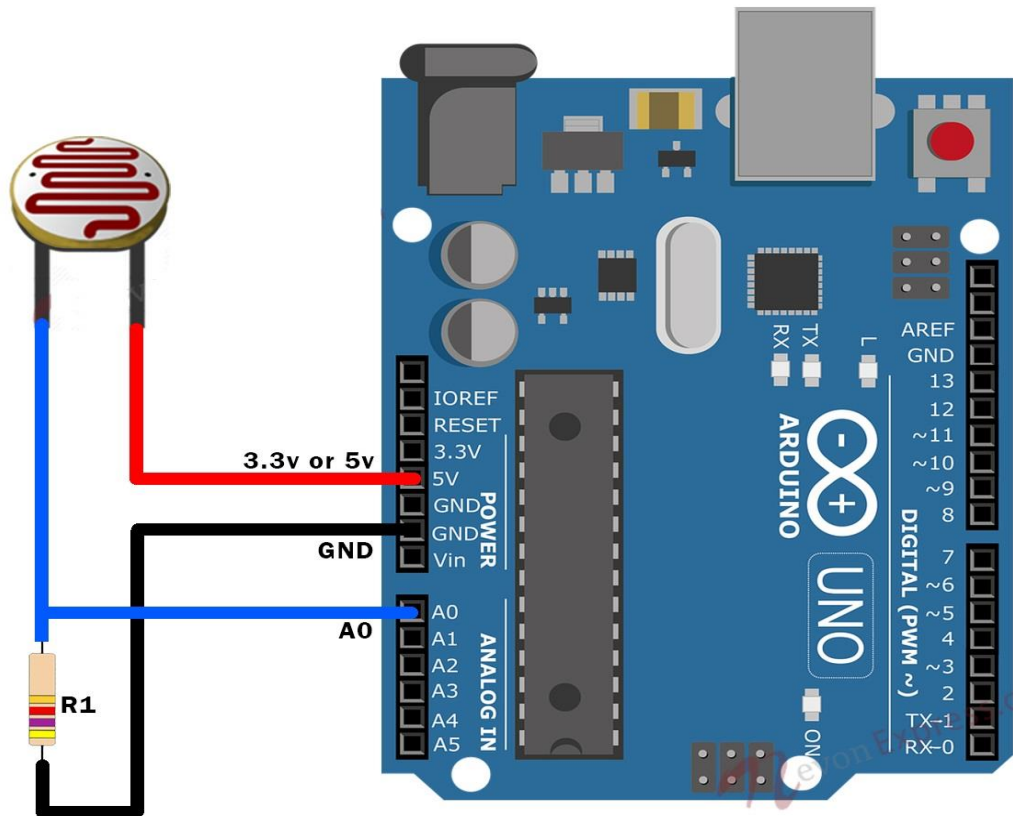


Figura 46: Conexión directa del sensor LDR con Arduino (sin Breadboard)

```
int LDR_Pin = A0; // what pin on the arduino is the LDR output connected to

void setup(){
  Serial.begin(9600);
  Serial.println("LDR Light Sensor Example Sketch | cactus.io");
}

void loop(){
  Serial.println(analogRead(LDR_Pin));
  delay(3000);
}
```

Figura 47: Sketch Arduino para pruebas con sensor LDR

Además, de los sensores DHT y el LDR se hicieron experimentos con sensores de corriente con el objetivo de comprobar si era más factible realizar la detección del estado de las cargas (luminarias) por flujo de corriente en línea o por detección de presencia de luz con el sensor LDR. Al final se optó por utilizar sensores dependientes de luz por ser más económicos, facilidad para calibración y la integración con el controlador Arduino para lograr la funcionalidad deseada. Sin embargo, a pesar de haber seleccionado el sensor LDR, se consideró adecuado incluir la información pertinente a las pruebas efectuadas con dichos sensores de corriente.

2.5. SENSOR DE CORRIENTE SCT013-30

El sensor que se ha escogido para hacer posible el cálculo de potencia consumida es el SCT013-030, un transformador de corriente-tensión para corriente alterna con una relación nominal de 30A/1V (Es decir, que para una entrada de 30 A genera una salida de 1 V, según la hoja de características). El sensor en cuestión se muestra en la figura 48.



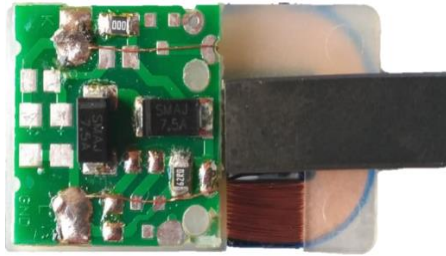
Figura 48: Sensor SCT013-030

El modo de emplear este tipo de sensores (de pinza amperimétrica) es la siguiente:

- ✚ Se abre la pinza por la pestaña indicada.
- ✚ Se toma un solo hilo de la alimentación a la que está conectada el dispositivo cuyo consumo se desea medir.
- ✚ Se hace pasar por el centro de la pinza.
- ✚ Se cierra la pinza.

La tensión de salida será medida según un cable de audio estándar de 3.5 mm.

Este sensor presenta en su interior un circuito mínimo en el que se puede apreciar cómo se hace la conversión de corriente a tensión. En primer lugar, los dos bobinados del transformador lo forman, por un lado, el propio cable por el que pasa la corriente que se va a medir (primario) y, por el otro lado, un hilo de cobre de 1 mm con un total de 1800 vueltas (secundario) sobre un núcleo de ferrita que se puede abrir para dejar pasar el cable (como se detalló anteriormente). El circuito posterior, como se puede apreciar en la figura 50, está construido sobre una placa de circuito impreso (PCB) multipropósito en la que muchas de las pistas quedan inutilizadas para esta funcionalidad, dejando un equivalente eléctrico tan sencillo como el que se observa en la figura 51, extraída directamente de la hoja de características del fabricante del sensor.



**49: PCB interna del SCT013-030
junto secundario del transformador**

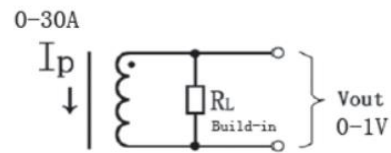


Figura 50: Esquema Equivalente

Los componentes de mayor tamaño que se pueden observar en la figura anterior son dos diodos SMD que cumplen una función de protección frente a posibles sobre intensidades en el momento de estar midiendo. Se obvian en el esquema porque en funcionamiento normal del dispositivo no actúan de ninguna manera, pero están colocados en paralelo a la resistencia de salida R_L en disposición de cátodos enfrentados, limitando así la salida a un máximo de aproximadamente ± 7.5 V según el fabricante. Por otro lado, se observa una resistencia de 0Ω que une dos pistas, para el caso de esta PCB, con un simple cortocircuito. El esquema completo con estos componentes de más se aprecia en la figura 50.

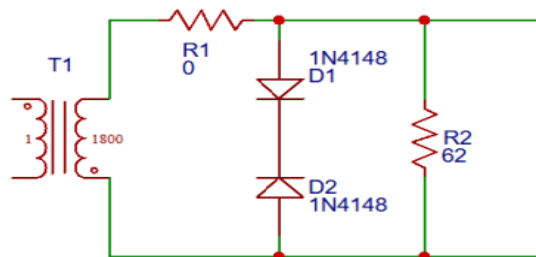


Figura 51: Equivalente completo

El primario del transformador representa el hilo por el que pasa la corriente que genera el consumo y el secundario es el bobinado de 1800 vueltas.

Se puede hacer un desarrollo sencillo para ilustrar cómo los valores proporcionados de la transformación implican que haya que usar los valores que se muestran en la figura 52. De la relación de transformación se tiene la expresión siguiente:

$$\frac{N_1}{N_2} = \frac{U_1}{U_2} = \frac{I_2}{I_1}$$

Por lo tanto, $I_2 = I_1 \frac{N_1}{N_2}$, con $N_1 = 1$ y $N_2 = 1800$.

De esta manera, para $I_1 = 30 \text{ A} \rightarrow I_2 = \frac{1}{1800} 30 \text{ A} = 0.0167 \text{ A}$, y así:

$$V_{out} = I_2 R_2 = 0.0167 * 62 \approx 1 \text{ V}$$

Figura 52: Ecuaciones de relación de transformación

2.6. ADAPTADOR PARA SENSOR SCT013-30

Como se dijo anteriormente, el sensor necesita que sólo un hilo del cable que alimente a una carga (de forma genérica) pase por su interior. Por lo general, los cables de alimentación tienen dos o más cables en el caso de cargas monofásicas: ida, retorno y, a veces, tierra de protección. Para hacer esto habría que quitar la cubierta que tiene el cable que envuelve a los dos (o tres) conductores, y posteriormente rodear con la pinza la ida o el retorno (según la primera ley de Kirchhoff, la intensidad por un hilo de ida es la misma que la de vuelta, ya que la tierra de protección en condiciones normales arrastra una corriente nula).

Para subsanar este hecho se fabrica un pequeño adaptador que sirve para la fase de pruebas con cargas monofásicas que tiene el aspecto que se puede apreciar en la figura 54 y figura 55:



Figura 53: Adaptador para sensor

Figura 54: Montaje de sensor con adaptador

Los sensores de la serie SCT-013 son sensores que trabajan como transformadores, la corriente que circula por el cable que se desea medir actúa como el devanado primario (1 espira) e internamente tiene un devanado secundario que dependiendo del modelo pueden tener hasta más de 2000 espiras.

La cantidad de espiras representa la relación entre corriente que circula por el cable y la que el sensor entrega, esta relación o proporción es la que diferencia entre los diferentes modelos de sensores SCT-013, adicionalmente pueden tener una resistencia de carga en la salida, de esta forma en lugar de corriente se trabaja con una salida voltaje como ya se mencionó anteriormente. Esto se puede apreciar mejor en la figura 55

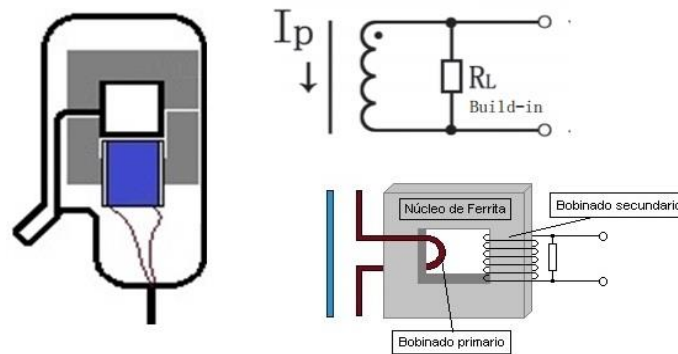


Figura 55: Sensores CT (Current transformers)

A este tipo de sensores se les conoce como sensores CT (Current Transformers), que como ya se explicó anteriormente, son transformadores, por el núcleo solo debe atravesar una sola línea, ya que si se pasarán los dos cables (Fase y Neutro) de una conexión monofásica, la lectura será cero 0, puesto que los cables tienen corrientes opuestas.

Ventajas del sensor de corriente SCT-013.

- ✚ No es necesario interrumpir (cortar o desempalmar) el cable que se va a medir, y eso es debido a que al igual que una pinza amperimétrica tiene el núcleo partido. Ver figura 56.



Figura 56: Forma de pinza amperimétrica del sensor de corriente

2.7. SENSOR DE CORRIENTE SCT013-100

Con esta variante del sensor SCT013 se puede medir hasta 100A, teniendo como salida 50 mA para una corriente de 100A, la proporción de relación es: 100A/50mA. Ver figura 57.



Figura 57: Sensor SCT013-100

El resto de la familia de sensores SCT-013 ya tienen incluida la resistencia de carga. A continuación, se muestra una lista de los diferentes modelos de sensores. Todos tienen un voltaje de salida entre 0 V y 1 V. Ver tabla 6.

Model	SCT-013-000	SCT-013-005	SCT-013-010	SCT-013-015	SCT-013-020
Input current	0-100A	0-5A	0-10A	0-15A	0-20A
Output type	0-50mA	0-1V	0-1V	0-1V	0-1V
Model	SCT-013-025	SCT-013-030	SCT-013-050	SCT-013-060	SCT-013-000V
Input current	0-25A	0-30A	0-50A	0-60A	0-100A
Output type	0-1V	0-1V	0-1V	0-1V	0-1V

Tabla 6. Obtenida del Datasheet SCT-013.

2.8. CIRCUITO ACONDICIONADOR DE SEÑAL

La salida de este sensor es una señal alterna, cuyos valores no está dentro del rango de las entradas analógicas (0 a +5V) del controlador Arduino, si bien el rango del sensor puede ser inferior, la parte negativa de la señal podría dañar la tarjeta Arduino.

Lo ideal es, que si el sensor tiene una salida de -50mA a +50mA acondicionarlo a una salida de 0 a 5V. Esto se podría realizar convirtiendo de corriente a voltaje, (a un rango de [-1V a +1V]), luego amplificarlo (a [-2.5V +2.5V]) y finalmente un sumador para eliminar la parte negativa ([0 5V]).

Otra forma seria rectificando la entrada y trabajar con la parte positiva, esto asumiendo que la señal es simétrica. Para este caso se trabajará de esta manera.

Para rectificar no se puede usar diodos, puesto que la caída de voltaje en el diodo es muy grande en comparación al voltaje de la señal. Para esto se usa un operacional, configurado en un seguidor de voltaje, se hará uso del operacional LM358, que trabaja con polaridad positiva, de esta forma se eliminará la parte negativa de la señal, si bien no es un rectificador de onda completa, pero con una rectificación de media onda basta para trabajar con Arduino.

El LM358 se alimenta con 5V, se satura con 3.5V aproximadamente, motivo por el cual no se puede amplificar hasta 5V, pero si se trabaja con Arduino no se necesita

alcanzar los 5V, se puede trabajar con la referencia de voltaje interna de 1.1V y de esta forma aprovechar en el rango completo de la lectura analógica.

Si se está trabajando con el sensor SCT-013-030, significa que la salida es a voltaje y está en un rango de $\pm 1V$, esta onda no es apta para el pin analógico A0 de Arduino, porque este pin soporta tensiones que van desde 0 a 5 voltios, entonces el circuito acondicionador que se usará es el que se puede apreciar en la figura 59.

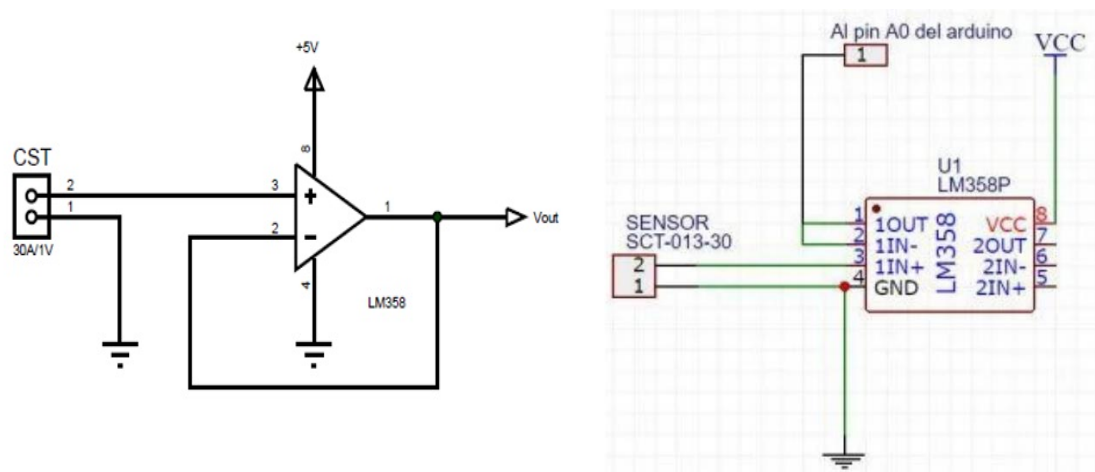


Figura 58: Diagrama eléctrico acondicionador de señal para sensor SCT-013-030

Para el segundo caso, en que el sensor es salida de corriente como el SCT-013-100, hay la necesidad de colocar una resistencia de carga para que este en un rango de 1 V ($50 \text{ mA} * 20 \Omega = 1 \text{ V}$), entonces el circuito acondicionador que se usará es el que se puede apreciar en la figura 59

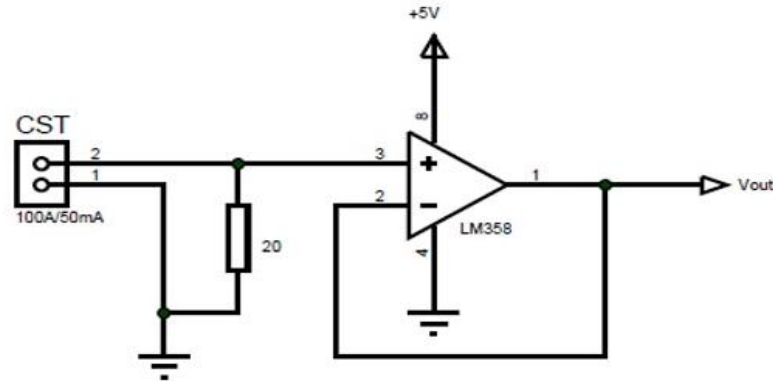


Figura 59: Circuito acondicionador de señal para sensor SCT-013-100

2.9. CONEXIÓN SCT – 013 CON ARDUINO

Primero se debe saber los pines del típico conector plug de audio 3.5 mm que se tiene en todos los auriculares y que incorpora el sensor en cuestión, conocido como Jack. Del lado del sensor de corriente SCT-013 se tiene el Jack macho con las siguientes conexiones. Ver figura 60.

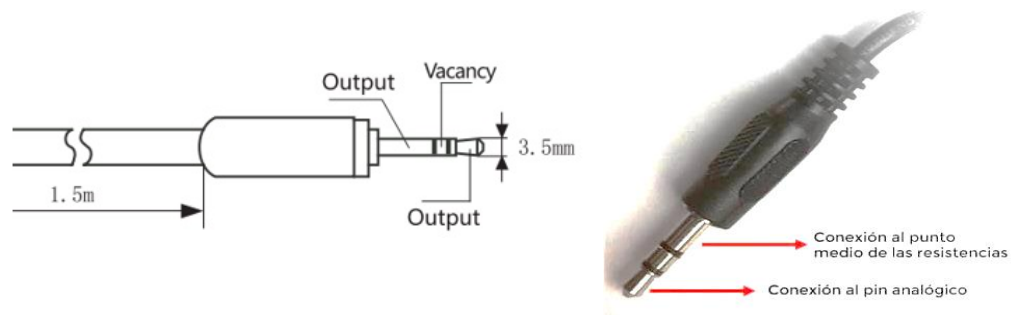


Figura 60: Terminales del sensor SCT-013 en Jack Macho 3.5 mm.

El pin central está desconectado, los otros dos son los que se ocupan para obtener la salida del sensor.

Para la opción de la figura 60 es necesario obtener un Jack hembra para hacer el empalme y conexiones más estéticas del sensor con la tarjeta Arduino.

La segunda opción es que si no se contara con el conector Jack hembra es cortar el cable para eliminar del sensor de corriente el conector Jack macho, el cual quedará como se puede apreciar en la figura 61.

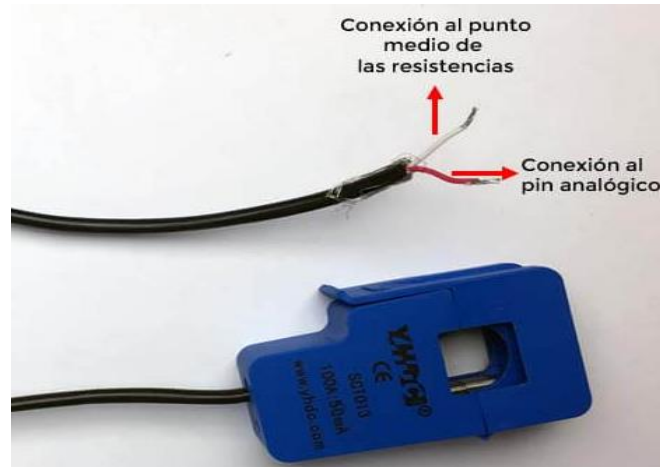


Figura 61: Terminales del sensor SCT-013 sin Jack Macho 3.5 mm.

La conexión que se utiliza es simple, solo es cuestión de montar los componentes antes mencionados como se puede apreciar en la figura 62.

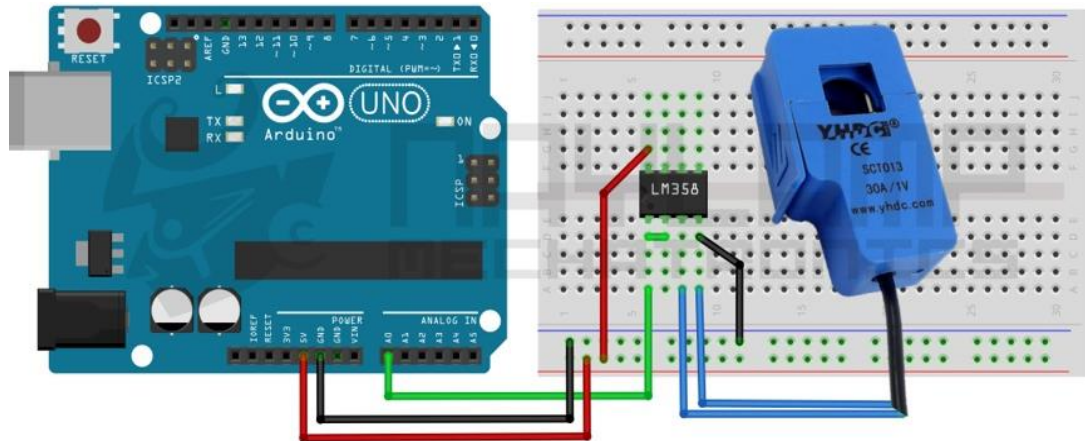


Figura 62: Montaje del circuito Arduino – Sensor SCT013 – 030

Importante recordatorio: No olvidar colocar la resistencia respectiva si se va a trabajar con un sensor con salida de corriente y además que solo se debe pasar por el sensor un solo cable de la línea que se desea medir.

2.10. CODIGO FUENTE PARA LECTURA BÁSICA DE LA CORRIENTE

Se realizará el código fuente en el IDE de Arduino para la lectura de la entrada analógica y posteriormente se escalan los valores a voltaje (voltaje del sensor) y finalmente con la relación (corriente/voltaje) que se tiene como dato del sensor se obtendrá la corriente. Ver figura 63.

```
void setup() {  
    Serial.begin(115200);  
    analogReference(INTERNAL);  
}  
  
void loop() {  
    int sensorValue = analogRead(A0); //Lectura analógica  
    float voltajeSensor = analogRead(A0) * (1.1 / 1023.0); //voltaje del sensor  
    float corriente=voltajeSensor*30.0; //corriente=VoltajeSensor*(30A/1V)  
    Serial.println(corriente,3);//enviamos por el puerto serie  
}
```

Figura 63: Código fuente Arduino – Sensor SCT013 – 030

Como se puede apreciar en la figura 63 se trata de una simple lectura analógica. En la siguiente línea se realiza el cálculo para obtener la corriente. Ver figura 654

```
float corriente=voltajeSensor*30.0; //corriente=VoltajeSensor*(30A/1V)
```

Figura 64: Código Arduino para obtener la corriente.

Dependiendo del modelo de sensor de corriente que se adquiriera se debe cambiar el 30.0 por el valor de la relación correspondiente al sensor.

Si el sensor se utiliza con la tarjeta Arduino Mega, se tienen a disposición dos referencias internas para la lectura analógica, es recomendable usar la referencia de 1.1V y para ello se utiliza la siguiente línea de código para Arduino Mega. Ver figura 65.

```
analogReference(INTERNAL1V1); //solo Arduino Mega
```

Figura 65: Código utilizar referencia de 1.1V en Arduino Mega

2.11. ACTUADORES

Un actuador es un dispositivo capaz de transformar energía hidráulica neumática o eléctrica en la activación de un proceso con la finalidad de generar un efecto sobre un proceso automatizado. Este recibe la orden de un regulador o controlador y en función a ella genera la orden para activar un elemento final de control, como por ejemplo una luminaria, válvula, motor, etc. Son los elementos que influyen directamente en la señal de salida del automatismo, modificando su magnitud según las instrucciones que reciben de la unidad de control.

2.12. INTERFAZ ELECTRÓNICA DE CONTROL DE CARGAS.

Para el control de cargas de alto voltaje usando la placa Arduino se pueden usar relés. Un relé es en realidad un interruptor que es accionado eléctricamente por un electroimán. El electroimán se activa con un voltaje bajo, por ejemplo, a 5 voltios que perfectamente pueden venir de una señal de salida desde un microcontrolador y se acciona un contacto para abrir o cerrar un circuito de alta tensión. Ver figura 66.

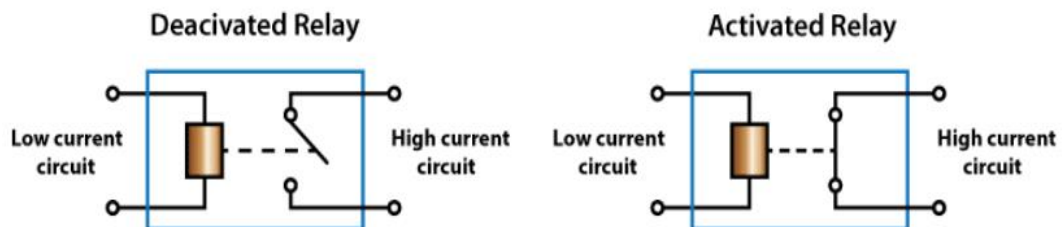


Figura 66: Estados de un relé

2.13. MÓDULO RELÉ HL-52S

El módulo relé de dos canales HL-52S, posee dos relés con clasificación de 10A @ 250/125 V AC y 10A @ 30 y 28 V CC. El conector de salida de alto voltaje tiene 3 pines, el de en medio es el pin común y, como se puede apreciar en las marcas de la

figura 68, uno de los otros dos pines es para la conexión normalmente abierta y el otro para la conexión normalmente cerrado.



Figura 67: Módulo de relé de 2 canales HL-52S

En el otro lado del módulo se tienen 2 juegos de pines. El primero tiene cuatro pines, de nomenclaturas GND, VCC, IN1 Y IN2 para alimentación del módulo y señales de entrada para control respectivamente, las cuales pueden ser inyectadas a través de un microcontrolador. El segundo conjunto de pines, de nomenclaturas JDVcc, Vcc y Gnd, en los cuales se puede observar que se encuentra un Jumper (color amarillo) con un puente entre el JDVcc y el pin Vcc, el electroimán del relé se alimenta directamente de la placa Arduino y si algo falla en la etapa relé, el microcontrolador podría dañarse por no existir aislamiento. Ver figura 68.



Figura 68: Módulo de relé de 2 canales HL-52S – alimentación y señales de control

2.14. ESQUEMA DEL CIRCUITO

Se detallan los esquemas del módulo relé HL-52S (circuito) en la configuración que se hace mención en la figura 69. Así, se puede observar que los 5 voltios del microcontrolador están conectados al pin Vcc para activar el relé a través del optoacoplador IC y también se encuentran conectados al pin JDVcc que alimenta el electroimán del relé. En este modo de configuración de conexión no existe aislamiento entre el relé y el microcontrolador. Ver figura 69.

Para aislar el microcontrolador del relé, se debe quitar el puente y conectar una fuente de alimentación separada para el electroimán al JDVcc y a la terminal tierra (GND), con esta configuración de conexiones, el microcontrolador no tiene ninguna conexión física con el relé, solo se usa la luz LED del optoacoplador IC para activar el relé. Se recomienda usar el módulo relé en esta configuración por seguridad del microcontrolador. Ver figura 70.

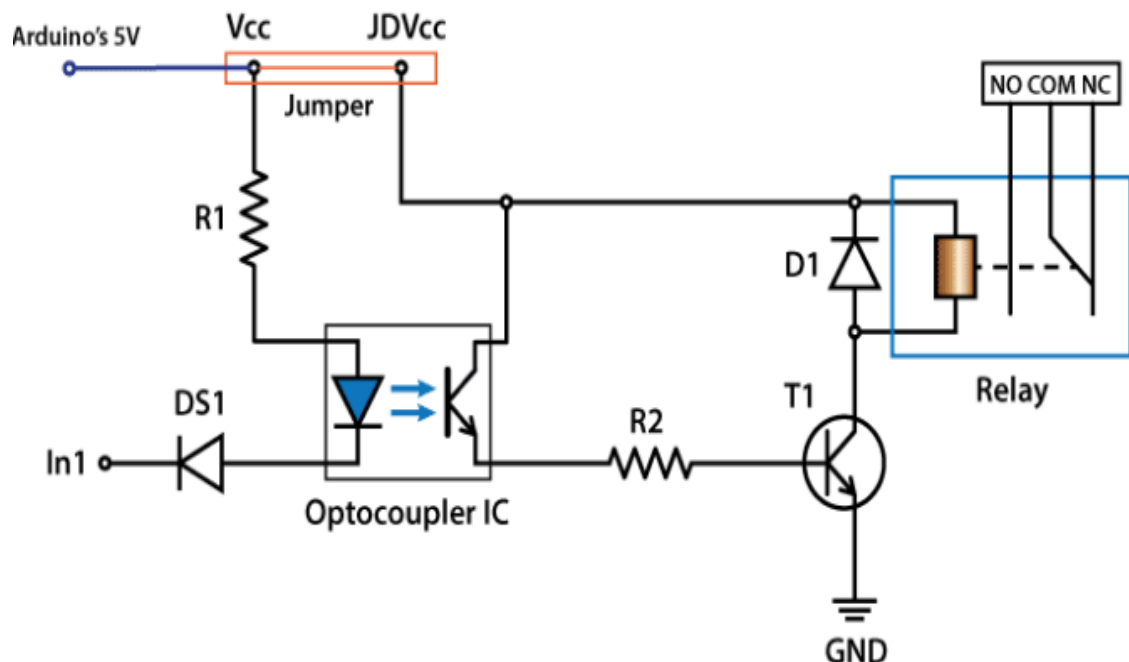


Figura 69: Esquema del circuito HL-52S - con puente entre el JDVcc y el pin Vcc

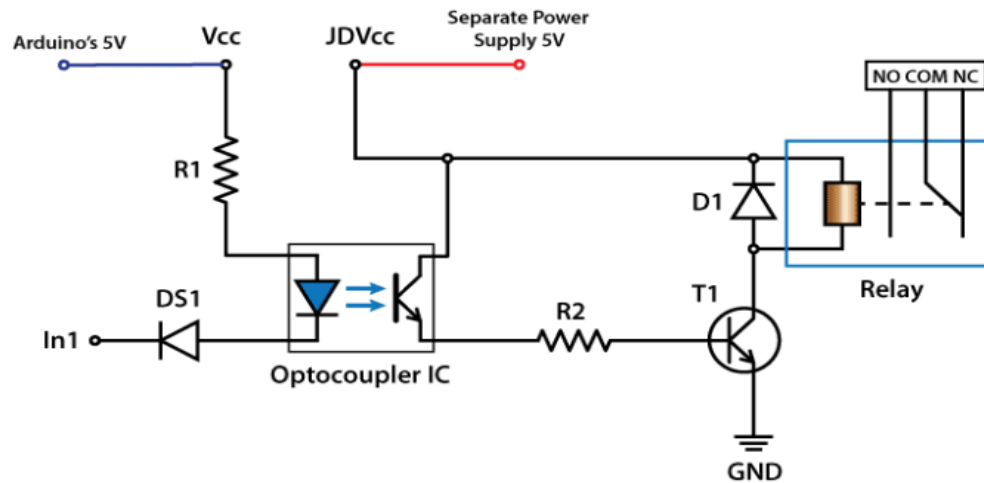


Figura 70: Esquema del circuito HL-52S – sin puente entre el JDVcc y el pin Vcc

Una última consideración a tener en cuenta a partir del esquema presentado en la figura 68, es que los pines de entrada (IN1 e IN2) del módulo relé HL-52S funcionan inversamente (Lógica negada), esto se puede apreciar en la figura 71. Como se puede observar, el relé se activará cuando el pin de entrada (In1 o In2) se lleve a estado BAJO o tierra, solo de esta manera la corriente podrá fluir desde el VCC al pin de entrada que se encuentra en estado BAJO o tierra (GND) y, el LED se iluminará provocando con ello que el relé sea ACTIVADO. Cuando el pin de entrada se ponga a estado ALTO, no habrá flujo de corriente, por lo que el LED no se encenderá y el relé estará o pasará en caso de estar ACTIVADO a estado reposo o DESACTIVADO.

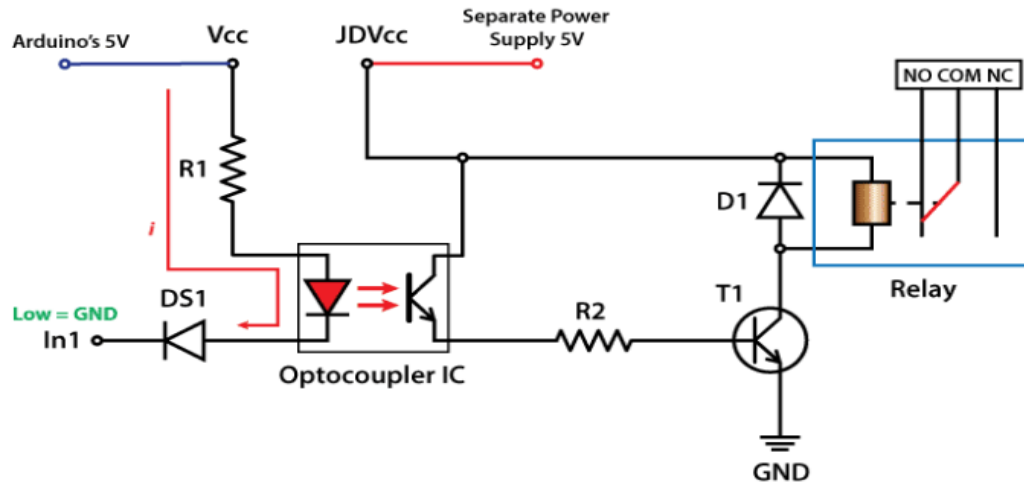


Figura 71: Esquema del circuito HL-52S – Señal de Entrada Terminal In1.

2.15. USO DEL MÓDULO RELÉ HL-52S CON CARGA DE ALTA TENSIÓN

Se recomienda primero volver a revisar los diagramas o esquemas del módulo relé HL-52S (Circuitos de figura 70 y figura 71). Como se describió anteriormente por la seguridad del microcontrolador al presentarse fallos en la etapa de relé se eligió utilizar un adaptador de 5V como una fuente de alimentación separada para el electroimán conectado al JDVcc y al pin de tierra. El pin 5V de la tarjeta Arduino se conecta al pin VCC del módulo relé y el pin 7 de Arduino al pin de entrada In1 del módulo relé para controlar el relé asociado a dicha terminal. Finalmente, para la parte de ALTO VOLTAJE se requiere un enchufe de alimentación y una extensión con dos cables, uno de los dos cables se corta y los extremos cortados se conectan en la bornera de pin común y el otro a la bornera normalmente abierto del conector de salida del módulo relé, de esta manera se obtiene como resultado el circuito de control de alto voltaje cerrado y en funcionamiento según se desee poniendo la señal correspondiente de salida en el pines 7 de Arduino y por consiguiente en pin In1 del módulo relé. Ver figura 72 y figura 67-68. Los GNDs, el GND del adaptador y GND de Arduino deben unirse.

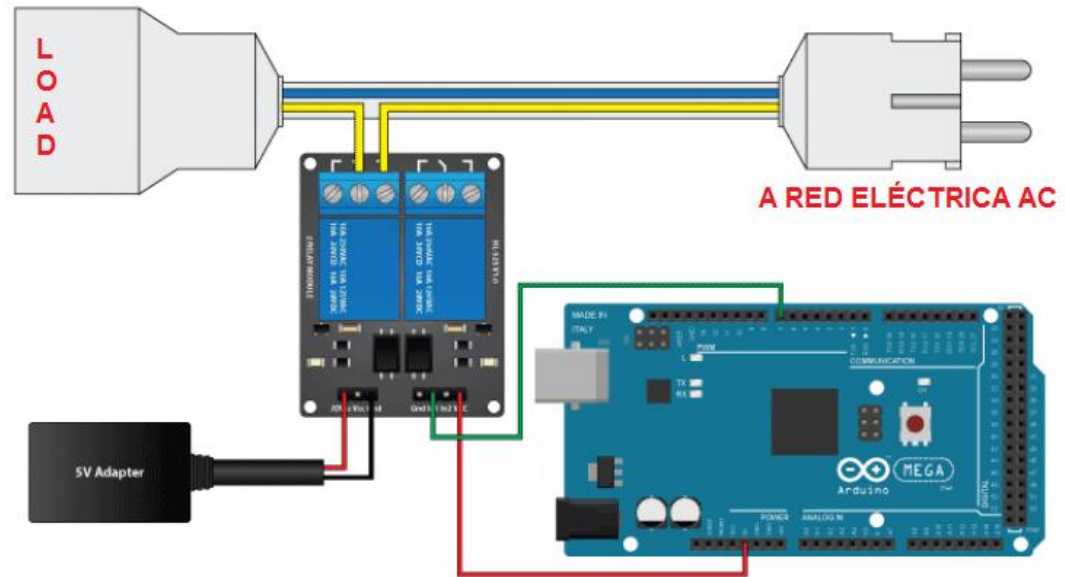


Figura 72: Conexión física de módulo relé HL-52S con carga de alto voltaje en Arduino Mega

2.16. SKETCH ARDUINO PARA CONTROL DE CARGAS DE ALTA TENSIÓN CON MÓDULO RELÉ

Se presenta un código simple para el control ON y OFF de una luminaria de 120 VAC, en el cual se usa el pin 7 de Arduino para el control del relé, de esta manera se define como salida y se escribe un programa el cual solo activará y desactivará el relé cada 3 segundos. Recordar que la entrada del módulo relé funciona con lógica negada, por lo que un estado BAJO en la entrada activará el relé y viceversa. Ver figura 73.

```

1.  int in1 = 7;
2.
3.  void setup() {
4.      pinMode(in1, OUTPUT);
5.      digitalWrite(in1, HIGH);
6.  }
7.
8.  void loop() {
9.      digitalWrite(in1, LOW);
10.     delay(3000);
11.     digitalWrite(in1, HIGH);
12.     delay(3000);
13. }

```

Figura 35: Código fuente para control ON – OFF de luminaria a 120 VAC con Arduino

2.17. RESUMÉN DEL CAPÍTULO

En este capítulo se habla de los sensores que se utilizan para que el sistema pueda captar los datos de las variables o magnitudes físicas que se necesitan monitorear y controlar. Así, se hace mención de los sensores para medir la temperatura y humedad relativa ambiente (DHT) y sensores dependientes de luz o LDR, para la detección del estado de las luminarias, además se presentan las pruebas realizadas con el sensor de corriente SCT013 para la detección de flujo de corriente en línea (potencia consumida), este último no se utilizó porque se decidió trabajar con sensores LDR.

Los sensores seleccionados son totalmente compatibles con la tecnología Arduino para facilitar la integración entre las partes y principalmente para reducir costos de todo el proyecto.

También se hace mención de los actuadores que se utilizan en el sistema, así como de las partes que lo conforman el control de cargas y se generan código fuente básico para pruebas de demostración de uso básico.

CAPÍTULO III

APP ANDROID

3. APP DEL PROYECTO

3.1. APP.

Definición App: Una App es una aplicación de software que se instala en dispositivos móviles o Tablets para ayudar al usuario en una labor concreta, a diferencia de una WebApp que no es instalable. El objetivo de una app es facilitar la consecución de una tarea determinada o asistirnos en operaciones y gestiones del día a día. Para el desarrollo de la App, se utilizó el IDE de Android Studio 3.1.3



Figura 74: Versión del IDE de Android Studio utilizada para el desarrollo de la App del proyecto.

Archivo APK: Fichero que se genera en el proceso de compilación del código JAVA y clases de Android Studio, el cual se utilizará para la instalación en dispositivos móviles o Smartphone con sistema operativo Android.

Permisos otorgados a la App: Toda aplicación que se desarrolla requiere ciertos permisos para hacer uso de ciertas funciones deshabilitas por defecto en el dispositivo y para esto fue necesario realizar dos procedimientos.

1. Al momento de estar programando es necesario darle los respectivos permisos a la App, los cuales normalmente están relacionados en función de la acción que se desea realizar. Por ejemplo: Si se desea realizar envíos de SMS desde el dispositivo móvil a través la App, es necesario que el programador haya realizado dicha especificación en el código fuente de la App.
2. Una vez instalada la aplicación en el Smartphone con sistema operativo Android es necesario entrar a los ajustes desde el dispositivo móvil y buscar en aplicaciones el nombre de la App instalada, luego buscar la opción permisos para habilitarlos, todos los permisos previamente considerados por el programador. Es obligatorio realizar por parte del usuario final el procedimiento 2 para que la App pueda realizar todas las funciones definidas por el programador.

Archivo donde el programador define los permisos que le dará a la App:
AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.utla.utla">
    <uses-permission android:name="android.permission.CALL_PHONE"/>
    <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
    <uses-permission android:name="android.permission.BLUETOOTH" />
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
    <uses-permission android:name="android.permission.SEND_SMS" />
    <uses-permission android:name="android.permission.READ_SMS" />
    <uses-permission android:name="android.permission.RECEIVE_SMS" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <!-- To auto-complete the email text field in the login form with the user's e
    <uses-permission android:name="android.permission.GET_ACCOUNTS" />
    <uses-permission android:name="android.permission.READ_PROFILE" />
    <uses-permission android:name="android.permission.READ_CONTACTS" />
    <uses-permission android:name="android.permission.READ_CONTACTS" />
    <uses-permission android:name="android.permission.WRITE_CONTACTS" />
```

Figura 75: Fichero de definición de permisos por el programador.

3.2. CARACTERÍSTICAS DE LA APP

- ✚ Seguridad en el acceso a las principales funciones de control y monitoreo.
- ✚ Usuario master e invitados (Tipos de LOGIN).
- ✚ Registro de nuevos usuarios invitados.
- ✚ Cambio de contraseña olvidada.
- ✚ Envío de SMS por red GSM para el control y monitoreo de las variables físicas del sistema.
- ✚ ON y OFF de manera simple (Una luminaria a la vez) con confirmación y sin confirmación de cambio de estado aplicado.
- ✚ ON y OFF múltiple (Varios estados de luminarias cambiados por SMS enviado) con confirmación.
- ✚ Realizar llamada perdida para el cambio de estado general de todas las luminarias del sistema con confirmación o notificación del estado final de las luces a través del DASHBOARD de luces que se ejecuta en segundo plano.
- ✚ Control de luminarias por Bluetooth.
- ✚ Control de luminarias por comandos de voz y Bluetooth.
- ✚ Dashboard para visualizar la información proveniente de los sensores vía SMS entrante. Temperatura y humedad relativa, además del estado de las luminarias.
- ✚ Dashboard para visualizar la información proveniente de los sensores vía internet (Tomada desde la base de datos en la nube)
- ✚ Opción configuración de la URL o dominio del servidor y directorio/subdirectorios donde se almacena el Portal Web, Webservice y base de datos MYSQL.
- ✚ Opción configuración para usuario MASTER. Número de teléfono del propietario de móvil, Número de teléfono del chip instalado en shield GSM/GPRS SIM900, Numero del teléfono con el que se desea compartir información vía SMS, dirección de e-mail con la que se desea compartir información y dirección MAC del dispositivo Bluetooth receptor (Instalado en el circuito).

- ✚ Compartir información a través de redes sociales como WhatsApp, e-mail, Drive, Bluetooth, entre otros.
- ✚ Interfaz de usuario amigable.
- ✚ Personalización de los datos de las cuentas de usuarios.
- ✚ Mapa Google para ubicación rápida de la UTLA.
- ✚ Menú lateral deslizante para acceso a las principales funciones de la App.
- ✚ Menú contextual en la parte superior derecha de las pantallas para el acceso a funciones principales.

3.3. INTERFACES GRÁFICAS DE LA APP

Igu: Interfaz gráfica de usuario. Está construida mediante una jerarquía de View y objetos ViewGroup.

Activity: Representan el componente principal de la interfaz gráfica de una aplicación Android. Se puede pensar en una actividad como el elemento análogo a una ventana en cualquier otro lenguaje visual.

3.4. DESCRIPCIÓN DE LAS PRINCIPALES PANTALLAS DE LA APP

PANTALLA SPLASHSCREEN: En esta interfaz gráfica se muestra una portada de carga de la App, la cual hay que esperar que transcurran más o menos 5 segundos para que automáticamente nos muestre la siguiente interfaz gráfica del LOGIN. En la figura 76 se aprecia la interface gráfica del SplashScreen.

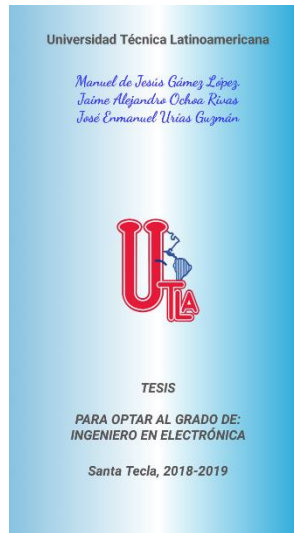


Figura 76: Portada de la App

PANTALLA LOGIN: Es esta interfaz gráfica se debe autenticar el usuario para poder tener acceso al menú principal de acciones o tareas programas en la App. Hay dos tipos de usuario que se manejan:

- + Usuario Master
- + Usuario Invitado

Usuario Master: Realizar una autenticación de acceso como usuario Master indicará a la App que muestre el menú principal de tareas programas completo y sin restricción de uso.

Usuario Invitado: Realizar una autenticación de acceso como usuario Invitado indicará a la App que muestre el menú principal de tareas programas con ciertas funcionalidades, entre las que se muestran están:

- + Información de su perfil de usuario.
- + Control de luminarias por Bluetooth.
- + Localización de la UTLA en Mapa Google. Requiere tener habilitado el sensor GPS del dispositivo móvil.
- + Acerca de.

En las siguientes figuras 77, 78 y 79 se muestra la interfaz gráfica del LOGIN, REGISTRARME Y RECUPERAR CONTRASEÑA respectivamente.

Figura 77: LOGIN

Figura 78: Registrarme

Figura 79: Recuperar clave

Desde la **PANTALLA LOGIN** el usuario deberá validar sus datos: e-mail y Password para tener acceso a las tareas o funcionalidades que brinda la App. Tener en cuenta que dependiendo del tipo de usuario que se valide así será el menú principal que se presente de tareas que se podrán ejecutar.

Desde la **PANTALLA REGISTRARME AHORA** se podrán crear nuevas cuentas usuarios. Tener en cuenta que los usuarios creados a partir de esta pantalla son de tipo invitado. El usuario Master se crea de manera automática por la misma App, los datos predeterminados de acceso que son generados de manera automática para el usuario Master se proporcionan a continuación.

Cuenta de usuario Master:

Usuario: **admin@utla.edu.sv**

Password: **admin**

Se recomienda al propietario del dispositivo móvil Android que una vez instalada la App realice de inmediato un primer acceso con el objetivo de poder editar sus datos de acceso para mayor seguridad.

Usuarios Master por dispositivo móvil podrán existir un máximo de uno, ya que sería ilógico contar con muchos usuarios Master en un mismo dispositivo, si siempre este será utilizado por el mismo propietario.

Consideraciones generales para los dos tipos de LOGIN:

-  El usuario Master podrá editar su perfil o datos personales y de acceso a su propia cuenta y de los usuarios invitados.
-  El usuario Master no se puede auto borrar.
-  El usuario Master puede editar y borrar cuentas de usuario invitados.
-  Los usuarios invitados pueden editar sus datos personales y de acceso a su cuenta. No se puede auto borrar.

PANTALLA RECUPERAR CONTRASEÑA: Desde esta opción se brinda a los usuarios registrados poder hacer un cambio de su contraseña o recuperación de la misma a través de la pregunta y respuesta de seguridad ingresada al momento de crear la cuenta. Ver figura 80 recuperar clave.

3.5. SEGURIDAD DEL SISTEMA EN GENERAL

El tema de la seguridad de acceso al control y monitoreo de las variables del sistema fue uno de los elementos que, durante todo el desarrollo en conjunto de la App y la programación del controlador, la tarjeta Arduino Mega; la cual tiene apilada un Shield GSM/GPRS SIM900 que hace la función de un dispositivo móvil transmisor y receptor de los datos proporcionados por los distintos sensores estuvo en consideración en todo momento debido a la importancia que amerita esta, ya que no cualquier usuario con dispositivo móvil Android podrá ejecutar acciones de control y monitoreo remoto y a continuación se describe cómo se ha manejado la misma:

- ✚ El primer nivel de seguridad es el LOGIN de acceso a las principales funcionalidades de control y monitoreo que brinda la App. Ver figura 77.
- ✚ El segundo y principal nivel de seguridad se aplicó para cuando se presente el siguiente caso: Suponer que cualquier usuario pudiera obtener de una fuente u otro el archivo APK o el instalador de dicha App y lo ejecute en su dispositivo móvil Android (No autorizado) para realizar acciones de control y monitoreo remoto; para este escenario se decidió crear en el controlador una base de datos de números de celular autorizados para evitar el robo o manipulación de datos por usuarios no autorizados. Cualquiera puede instalar la App si por descuido o por intensión se comparte el instalador en un dispositivo móvil Android (Usuario no autorizado) y luego obtener los datos de acceso como Master o crear un usuario Invitado y validarse. Si fuera este el caso y el usuario no autorizado intenta ejecutar acciones de control (ON / OFF) o peticiones de información de los sensores, el circuito de control y monitoreo recibirá dicho SMS o llamada, pero

este lo primero que hace es verificar en su base de datos de números de celulares autorizados, si existe, y en caso de no encontrarse dicho número de celular (Número del dispositivo móvil no autorizado donde se instaló la App) el dispositivo electrónico simplemente ignorara cualquier acción o petición ejecutada por usuarios no autorizados con la App.

- ✚ Aún más allá, Una vez instalada la App también será necesario que se conozca el número del chip instalado en el circuito para establecer comunicación con este desde la App. Por todo lo anterior se hace énfasis en hacer mención de lo siguiente.

Todo sistema es vulnerable por las siguientes razones:

- ✚ Su creador no pensó en implementar medidas de seguridad.
- ✚ Se hace un mal uso de la información de seguridad de acceso a dicho sistema, ya sea intencionalmente por hacer algún daño o probar dicha seguridad, por descuidos y confianza en permitir que otros usuarios manipulen dichos datos.

Se recomienda a la UTLA o usuarios autorizados hacer un uso cuidadoso y privado de la App de este trabajo de investigación, así como de otros datos que deben ser idealmente manejados como confidenciales.

- ✚ El archivo APK de la App de este proyecto. Solo personal autorizado.
- ✚ El número de celular o del chip instalado en el circuito de control y monitoreo. Contacto privado.
- ✚ Los datos de la cuenta de usuario Master.
- ✚ El dispositivo móvil, ya que si un usuario autorizado prestare su celular o lo extraviare y se empezara a manipular dicha App no habría manera inmediata de evitar que el sistema responda a cualquier petición realizada de control y monitoreo más que la de apagar y reprogramar el sistema dando de baja al número de celular que tiene acceso de control y monitoreo autorizado.

En la figura 80 se muestra el menú principal para un acceso de usuario Master

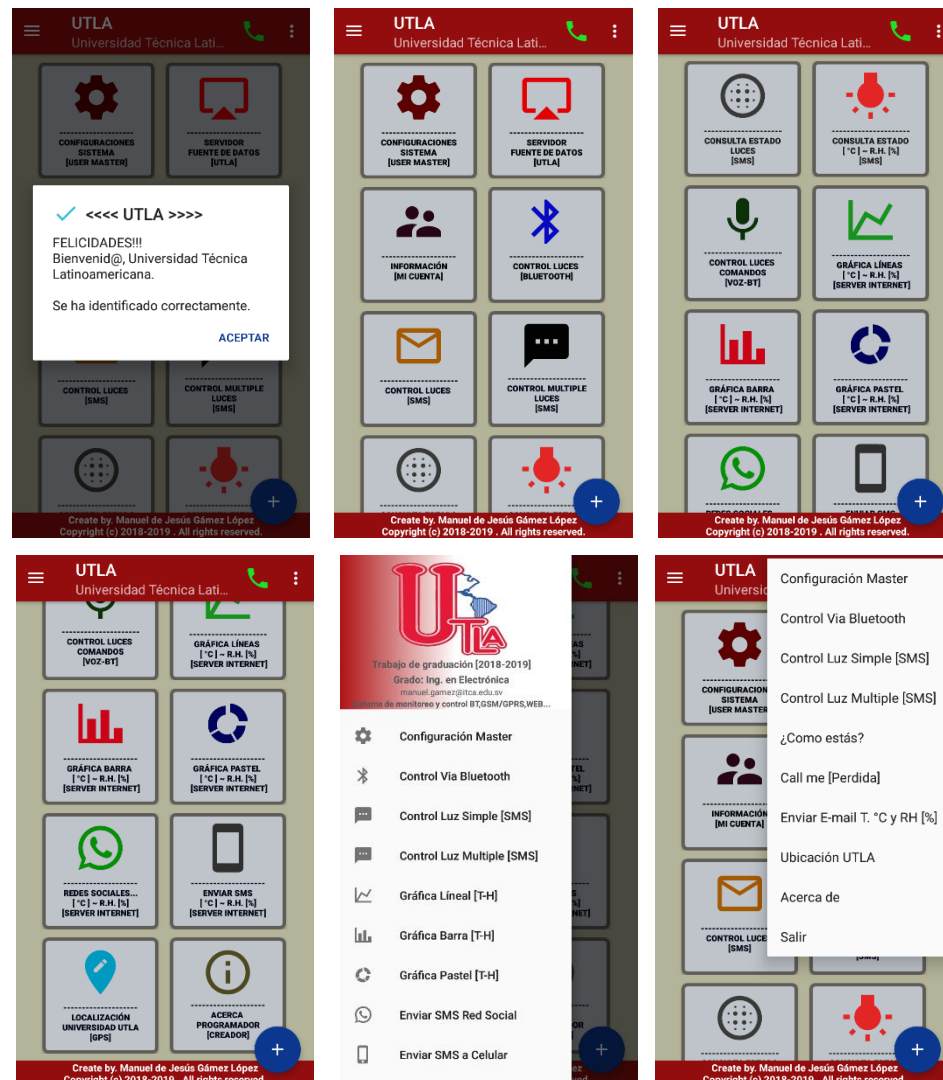


Figura 80: Menú principal que se muestra a usuario Master

A continuación, se describe cada una de las funcionalidades de la App.

Opción configuraciones del sistema:



Al dar clic sobre este botón será necesario volver a validar los datos de acceso como usuario Master. En la figura 81 se muestra la interface gráfica para el ingreso de los datos y su verificación. Si los datos ingresados son correctos se pasará a la interfaz gráfica que se muestra en la figura 83.

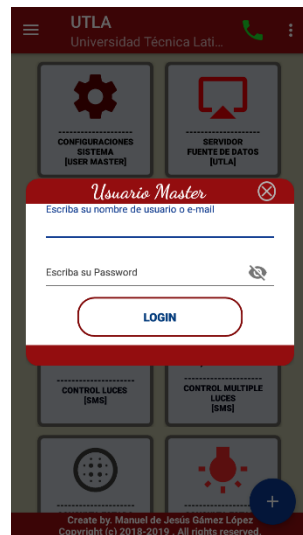


Figura 81: Login Master



Figura 82: System Settings

Descripción de los datos solicitados en la figura 82 System Settings.

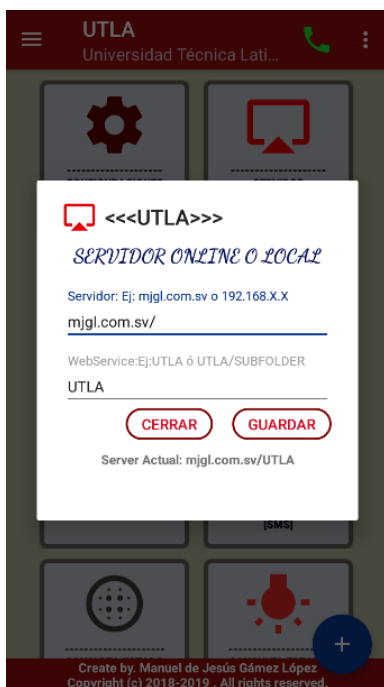
- + Número de teléfono del propietario del dispositivo móvil.
- + Número de teléfono del sistema electrónico.
- + Número de teléfono con el que se compartirá información vía SMS.
- + Dirección de correo con el que se compartirá información.
- + Dirección MAC del dispositivo Bluetooth Receptor.

Todos los datos son obligatorios para un correcto funcionamiento de las demás tareas que se pueden ejecutar a través de la App. En caso de ingresar datos incorrectos a este formulario puede ocasionar que ciertos módulos o procesos de la App no funcionen correctamente. Por ejemplo: Imaginar el caso en que, en el campo número de teléfono del sistema electrónico se introduzca un número de celular distinto al que realmente está instalado en el Shield GSM/GPRS SIM900 apilado al controlador Arduino Mega, sucederá que los SMS de control y monitoreo sean enviados a ese número incorrecto si fuera un número telefónico válido, nunca llegará al sistema electrónico para que este pueda ejecutar acciones de control y monitoreo en las variables del sistema y así mismo es de importante completar todos los demás campos de esta pantalla con información correcta.

Opción servidor - fuente de datos UTLA:



Al dar clic sobre este botón se mostrará la pantalla que se puede apreciar en la figura 83.



En esta pantalla se solicita en el primer campo el nombre de dominio o dirección IP del servidor donde se encuentra almacenado la base de datos, web service y portal web, además se debe ingresar en el segundo campo el nombre de directorio o subdirectorio donde se encuentre los ficheros PHP que forma el web service, el cual permite establecer comunicación entre la App y la base de datos del sistema. Al igual que la pantalla de System settings esta información debe ser introducida estando seguro de que dicha información es la correcta.

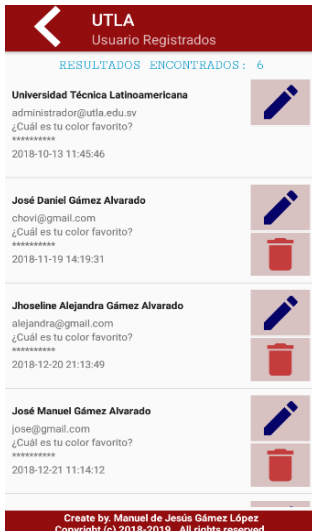
Figura 83: Servidor

En el caso que los datos introducidos sean incorrectos, muchas funcionalidades de la App estarán inservibles. Por ejemplo: No se podrá ver los registros almacenados en la base de datos sobre las variables del sistema, por ejemplo: la temperatura y humedad relativa a través de las diferentes gráficas que se han incorporado a esta. Así mismo hay muchas otras funcionalidades de esta App que dependen de una correcta inserción de los datos en este formulario o pantalla.

Opción información de mi cuenta:



Al dar clic sobre este botón se mostrará la pantalla que se puede apreciar en la figura 46.

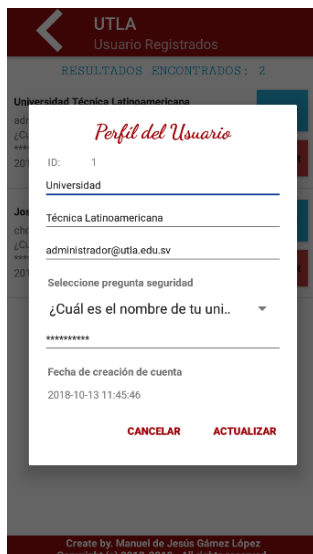


En esta pantalla se muestra la lista de usuarios Invitados y usuario Master que pueden iniciar sesión desde la ventana LOGIN. La vista de esta pantalla es la de un usuario Master y por tanto este puede realizar las actividades mencionadas en el apartado anterior de nombre: **Consideraciones generales para los dos tipos de LOGIN.**

Al dar clic en el botón con icono de LAPIZ se muestra la pantalla que se puede apreciar en la figura 84.

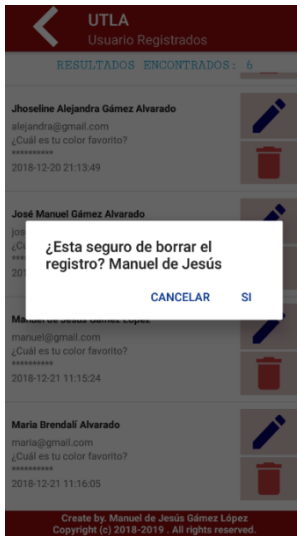
Figura 84: Información cuenta

En la figura 85, perfil de usuario se pueden realizar los cambios necesarios a los datos personales y de acceso de cada cuenta de usuario registrada o borrarla si se desea, exceptuando la cuenta de usuario Master, de la cual simplemente se pueden editar algunos datos.



Basta con hacer los cambios necesarios a los datos y dar clic en el botón ACTUALIZAR para guardar. En caso de no desear hacer ningún cambio clic en CANCELAR.

Figura 85: Perfil de usuario



En la figura 68, se puede visualizar el mensaje que aparece cuando se le da clic al bot3n con icono de PAPELERA DE RECICLAJE y si se est3 seguro de borrar la cuenta se da clic en la opci3n SI, caso contrario en CANCELAR.

**Figura 86: confirmaci3n
Borrar cuenta de usuario**

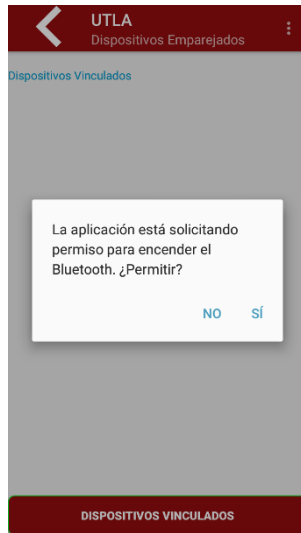
La primera fase de pantallas que se presentan y describen anteriormente, ha consistido en hacer notar, la importancia de la informaci3n ingresada en las ventanas de configuraciones, las cuales son necesarias para que la App puede ofrecer una funcionalidad 3ptima. Por tanto, se hace 3nfasis en realizar una correcta configuraci3n con los datos ingresados en cada ventana o pantalla de la App para garantizar una funcionalidad 3ptima de la misma.

La segunda entrega de pantallas o interfaces gr3ficas que a continuaci3n se van a mostrar y a describir comprende el hecho de la principal raz3n de la creaci3n de la App y est3 relacionado al control y monitoreo de las variables del sistema por Bluetooth, SMS y llamadas utilizando la red GSM y GPRS.

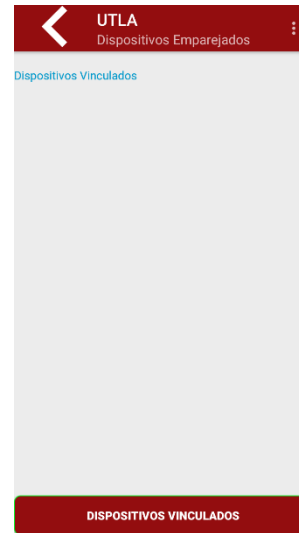
Opci3n Control de Luces Bluetooth:



Al dar clic sobre esto bot3n se mostrar3 la pantalla que se puede apreciar en la figura 87.



**Figura 87: Solicitud Permiso
Para encender BT**



**Figura 88: Sin solicitud
para encender BT**

En la figura 87 se observa que se hace una solicitud de permiso para encender el Bluetooth, se presentará si se detecta en el momento de la carga de dicha pantalla, que el dispositivo móvil Android tiene apagado su Bluetooth. Es necesario dar clic en el botón SI para encender el Bluetooth del dispositivo móvil y poder así proseguir con lo que se detalla más adelante. En el caso de haber presionado el botón NO simplemente el Bluetooth del dispositivo móvil seguirá APAGADO. Se puede encender desde el menú contextual que se encuentra en la parte superior derecha de la pantalla. Ver figuras 93 y 95: Menú contextual (Referencia los 3 puntitos color blanco en forma perpendicular).

La figura 88 se presentará si se detecta que el dispositivo móvil Android ya tiene en estado ENCENDIDO el Bluetooth. Bastara proseguir con lo que se detalla a continuación.

Una vez encendido el Bluetooth de su dispositivo móvil desde las distintas opciones que ofrece la App o de manera manual (sin App), es necesario dar clic en el botón DISPOSITIVOS VINCULADOS para que se listen los mismos. Ver figura 89.



En la figura 89, se puede apreciar la lista de dispositivos que ya previamente han sido vinculados con el dispositivo móvil Android. En caso de no mostrarse ninguno será necesario realizar la vinculación o apareamiento entre el módulo Bluetooth instalado en el sistema electrónico y el Bluetooth del dispositivo móvil Android de manera manual (Sin la App). Los datos del módulo Bluetooth instalado en el sistema electrónico son:

Nombre: HC-05

MAC: 20:16:11:28:26:40

Clave predeterminada: 0000 o 1234

Figura 89: Lista de Dispositivo BT vinculados

Una vez se muestre el nombre del dispositivo Bluetooth HC-05, instalado en el circuito, es necesario dar clic sobre este y se mostrara la pantalla que se puede apreciar en la figura 90:

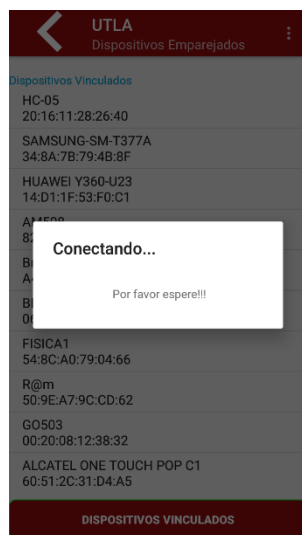


Figura 90: Conectando

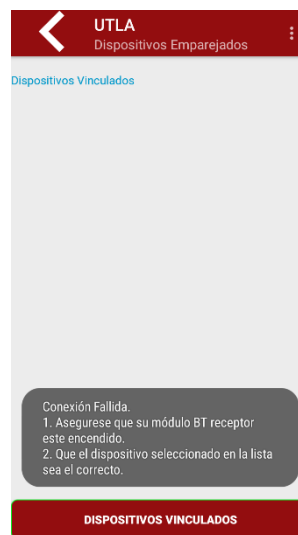


Figura 91: Error BT



Figura 92: Conexión establecida

La figura 90 conectando, se presenta en el momento de hacer clic sobre el nombre del dispositivo HC-05 e indica que se está tratando de establecer conexión vía socket con el Bluetooth instalado en el sistema electrónico. En caso de lograr establecer una

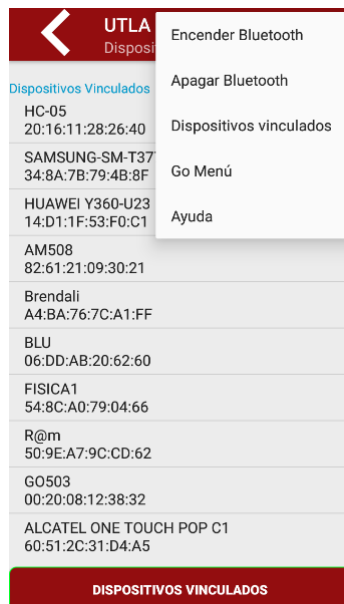
conexión correcta se mostrar la pantalla que se puede apreciar en la figura 90. Conexión establecida, mostrando inmediatamente el estado actual de las luminarias y las lecturas de temperatura y humedad relativa.

Además, se puede dar clic sobre cada icono de luminaria (Color amarillo ON y color gris OFF) para cambiar el estado actual de la luminaria seleccionada o cliqueada.

Además, se puede estar consultado vía Bluetooth las variables de temperatura y humedad relativa dando clic sobre el icono del botón correspondiente que se muestra en esta misma pantalla.

La figura 89 Error BT, se presentará al dar clic sobre el nombre del dispositivo Bluetooth HC-05 por las siguientes razones:

- ✚ El dispositivo Bluetooth este apagado o fuera de alcance.
- ✚ Que el dispositivo cliqueado o seleccionado en la lista no sea el correcto.



En la figura 93, se puede apreciar el menú contextual que se activa al dar clic sobre los tres puntitos ubicados en la parte superior derecha de la pantalla. Desde este menú se puede Encender el Bluetooth, Apagar el Bluetooth, Ver la lista de dispositivos vinculados, retornar al menú principal y una pequeña ayuda.

IMPORTANTE: Solo puede haber una conexión establecida por vez con el dispositivo Bluetooth instalado en el circuito.

Figura 93: Menú contextual BT

Control de luminarias por Bluetooth y comandos de voz

Otra opción que se adicionó a la App es la de poder controlar y monitorear las variables del sistema vía Bluetooth es por comando de voz. A continuación, se listan en la tabla 7, los comandos bajo los cuales la App permite ejecutar una orden.

Descripción	Comando de voz
Cambio estado foco 1	Foco 1 o foco uno
Cambio estado foco 2	Foco 2 o foco dos
Cambio estado foco 3	Foco 3 o foco tres
Cambio estado foco 4	Foco 4 o foco cuatro
Cambio estado foco 5	Foco 5 o foco cinco
Cambio estado foco 6	Foco 6 o foco seis
Cambio estado foco 7	Foco 7 o foco siete
Cambio estado foco 8	Foco 8 o foco ocho
Cambio estado de todas luces	Luces o luces, UTLA, utla
Retornar al menú	Retornar menú

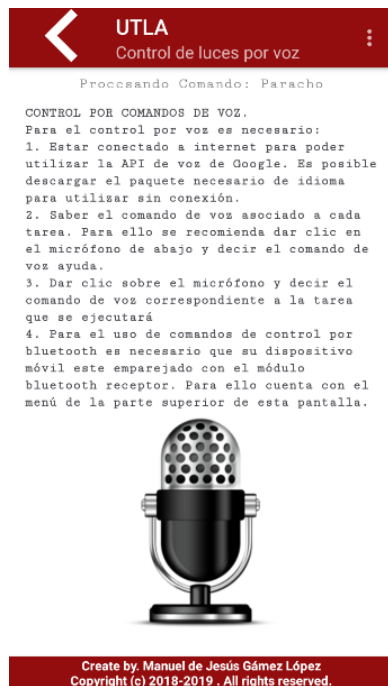
Tabla 7: Ayuda de comando de voz.

Para poder hacer uso de los comandos de voz para el control de luminarias vía Bluetooth es necesario entrar a la siguiente opción:

Opción Control de Luces por Bluetooth y Comandos de Voz.



Al dar clic sobre este botón se mostrará la pantalla que se aprecia en la figura 94.



La figura 94 muestra una lista de pasos a considerar para poder ejecutar comando de voz.

- ✚ Estar conectado a internet o poseer paquete de datos para poder hacer uso del servicio de reconocimiento de voz de Google.
- ✚ Conocer la lista de comandos de voz. Puede decir ayuda al hacer clic en el micrófono para saber lista de comandos.
- ✚ Dar clic sobre el micrófono y decir el comando de voz correspondiente a la tarea que se desea ejecutar. Ver tabla 8, ayuda de comandos de voz.

Figura 94: Control de luces

Por comandos de voz

Para el uso de comandos de voz para el control de luminarias vía Bluetooth, es necesario que su dispositivo móvil este emparejado con el módulo Bluetooth instalado en el circuito. Para ello se cuenta con el menú contextual ubicado en la parte superior de la pantalla. Ver figura 95.



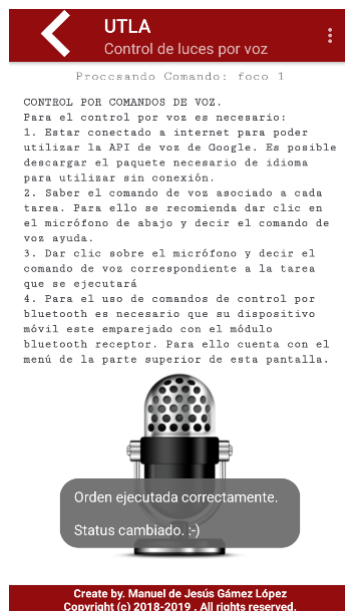
En la figura 95, se puede apreciar el menú contextual, el cual permite realizar tareas como: Encender el Bluetooth, enlazar con el Bluetooth, apagar el Bluetooth, solicitar ayuda de comandos de voz, salir de la pantalla.

IMPORTANTE: Para este menú contextual la primera opción tiene doble funcionalidad Encender/Enlazar BT y estas funcionan de la siguiente manera:

Si el Bluetooth del dispositivo móvil Android está apagado entra la función Encender y solicitara confirmación. Si el Bluetooth este encendido entra en función Enlazar BT.

Figura 95: Menú contextual Voz

Una vez entra en función la opción Enlazar BT, ya se pueden ejecutar órdenes de cambio de estado de luminarias por comandos de voz. Ver tabla 8, ayuda de comando de voz. Para el caso en que se haya ejecutado correctamente un comando de voz se presentará un Toash (alerta o notificación) con el texto: Orden ejecutada correctamente.



En la figura 96, se puede apreciar el Toash o notificación con el contenido: Orden ejecutada correctamente. Status cambiado :-)) indicando que se ha ejecutado correctamente la orden dada. Estar atento al cambio en las luminarias del Laboratorio de Electrónica según corresponda la orden.

Figura 96: Orden ejecutada

Correctamente

Opción Control de Luces SMS:



Al dar clic sobre este botón se mostrará la pantalla que se aprecia en la figura 97.



Figura 97: Control SMS
Sin confirmación



Figura 98: Menú contextual
control SMS

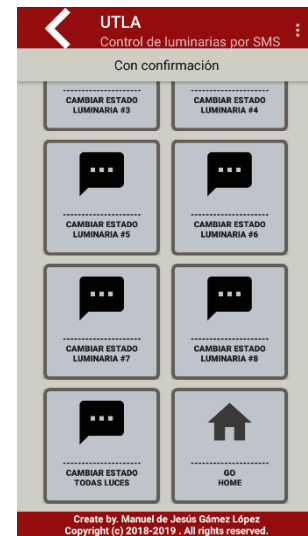


Figura 99: Control SMS
Con confirmación

En la figura 97 control SMS sin confirmación, se puede tener control del estado ON y OFF de cada luminaria de manera individual y no se solicita al circuito que notifique que ha efectuado dicho cambio. Basta con dar clic en el botón correspondiente a la luminaria que se desea cambiar su estado de ON a OFF o viceversa. Se requiere poseer saldo en el chip del dispositivo Android para enviar el SMS de control.

En la figura 98 menú contextual control SMS, se puede visualizar la opción CALL ME y EXIT.

CALL ME: Al dar clic en esta opción se le solicita al celular (circuito Shield SIM900 GSM/GPRS) que realice una llamada perdida. La idea de haber puesto dicha función en la App es para poder verificar dos cosas a través de esta opción.

1. Verificar la conexión del celular (circuito) con la red GSM. En el caso de recibir dicha llamada entonces se puede decir que está conectado a la red GSM.
2. Además, se verifica si hay saldo en el chip instalado en el celular (circuito).

En la figura 99 control SMS con confirmación, se puede tener control del estado ON y OFF de cada luminaria de manera individual y se solicita al circuito automáticamente que notifique que ha efectuado dicho cambio. Basta con dar clic en el botón correspondiente a la luminaria que se desea cambiar su estado de ON a OFF o viceversa. Se requiere poseer saldo en el chip del dispositivo Android para enviar el SMS de control y en el chip del celular (circuito) para poder enviarnos la confirmación o notificación de cambio de estado. Ver figura 100.

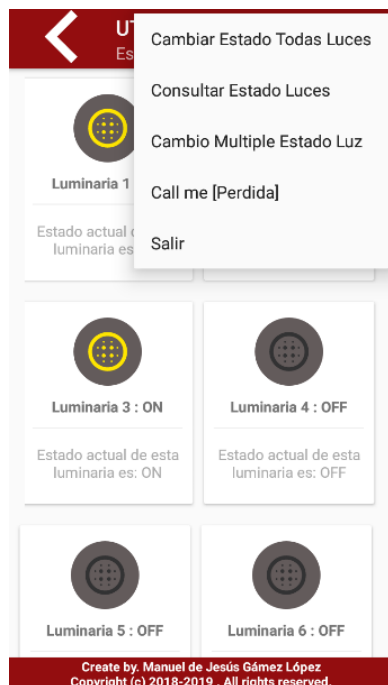


En la figura 100, se puede apreciar el DASHBOARD donde se muestra el cambio realizado y el estado de las demás luminarias. Desde este DASHBOARD de notificación se puede ir cambiando el estado de cada luminaria dando clic en cada icono de luminaria. Al hacer un cambio de estado de luminaria desde esta pantalla siempre se recibirá notificación del cambio efectuado. Se requiere poseer saldo por SMS enviado desde la App y del circuito (celular) por cada cambio efectuado. Además, se posee un menú contextual ubicado en la parte superior de esta pantalla. Ver figura 101.

Figura 100: Dashboard
De notificación

En la figura 101, se puede apreciar un menú contextual, el cual cuenta con las siguientes opciones: Cambiar estado todas luces, consultar estado luces, cambio múltiple estado luz, call me [Perdida] y salir.

Al dar clic sobre la opción **“Cambio Estado Todas Luces”**, se dará la orden de cambiar el estado de todas luces, las que se encuentren en estén ON pasarán a OFF y viceversa, si todas se encuentran en estado ON pasarán al estado OFF y viceversa.

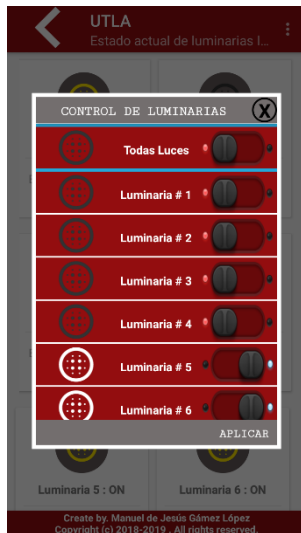


Al dar clic sobre la opción **“Consultar Estado Luces”**, se dará la orden para consultar el estado actual de las luminarias. Esta opción es útil al presentarse el caso en que alguien más remotamente vía SMS o localmente vía Bluetooth realiza cambios y se desea saber de esos cambios remotamente desde otro dispositivo.

Figura 101: Menú contextual

DASHBOARD SMS

Al dar clic sobre la opción **“Cambio Múltiple Estado Luz”**, desde esta opción se pueden dar órdenes para realizar cambios combinados o múltiples de estado de luminarias por SMS, teniendo la ventaja esta opción respecto a la otra **“Cambio de Luz Simple”** de poder cambiar estado de varias luminarias secuencialmente o intercalas por SMS.



En la figura 102, se puede apreciar la interfaz gráfica para realizar un cambio de estado múltiple de manera muy fácil, basta con dar clic en el/los switch correspondiente a la/las luminarias que se desee cambiar su estado y finalmente presionar el botón APLICAR para enviar la orden. Para el uso de esta pantalla es necesario contar con saldo en ambos celulares (Dispositivo Android y circuito celular) ya que en ambos casos se envían SMS.

**Figura 102: Cambio Múltiple
Luminarias por SMS**

Al dar clic sobre la opción **“Call me [Perdida]” de la figura 101**, se puede enviar una orden para solicitar una llamada perdida al celular (circuito) con una duración de más o menos 3 segundos, esta opción fue añadida para poder comprobar la conexión del celular (circuito) con la red GSM. Ver figura 102.

Al dar clic en la opción **“Salir” de la figura 103**, permite cerrar la venta de notificación o DASHBOARD y retornar al menú principal.



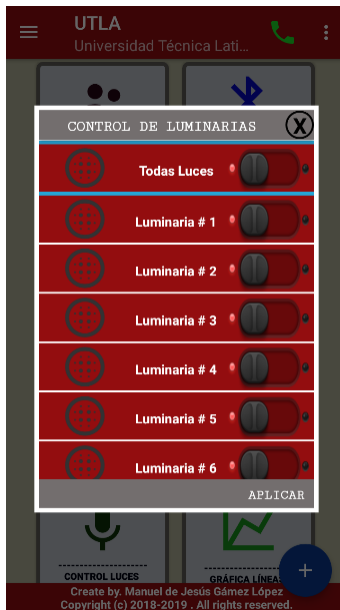
En la figura 103, se puede apreciar la llamada perdida que se recibirá del celular (circuito) al presionar la opción correspondiente.

Figura 103: Call me [Perdida]

Opción Control Múltiple Luces SMS.



Al dar clic sobre este botón se mostrará la pantalla que se aprecia en la figura 104.



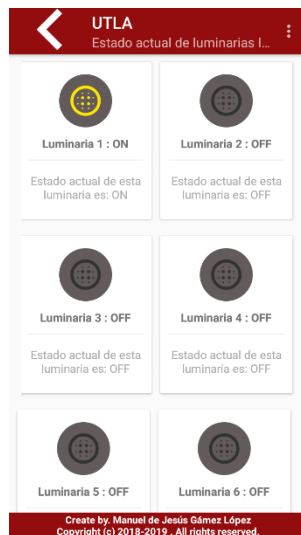
En la figura 104, se puede apreciar la interfaz gráfica que se presenta para realizar un cambio de estado de luminarias combinado o múltiple, ya sea en orden secuencial o intercalado vía SMS. Para el uso de esta pantalla es necesario que ambos celulares posean saldo, esta pantalla recibirá notificación (DASHBOARD) del último estado de las luminarias, siempre que se ejecute cualquier orden al presionar el botón APLICAR.

**Figura 104: Cambio múltiple
Estado luminarias por SMS**

Opción Consulta Estado Luces SMS



Al dar clic sobre este botón se enviará la orden al celular (código), el cual responderá con un SMS con la información del estado de las luminarias. Ver figura 105.



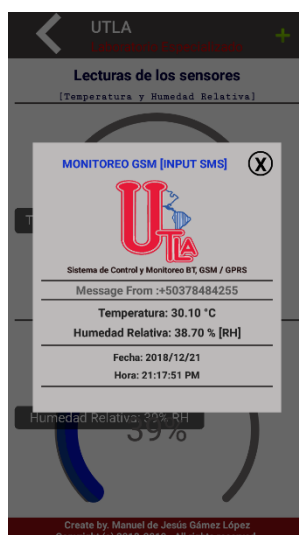
En la figura 105, se puede apreciar la interfaz gráfica con el último estado de las luminarias. Se genera de manera automática al detectarse que el celular (circuito) envía un SMS que contiene información acerca del estado de las variables del sistema. Esta función es ejecutada en segundo plano.

Figura 105: DASHBOARD
Último estado de luminarias

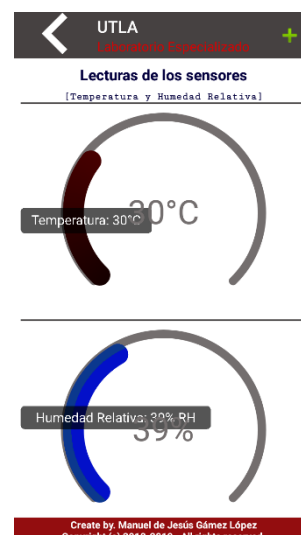
Opción Consulta Estado de Temperatura y Humedad Relativa.



Al dar clic sobre este botón se enviará la petición de consulta del estado de temperatura y humedad relativa, el cual responderá con un SMS con la información de las lecturas del sensor DHT22. Ver figuras 106 y 107.



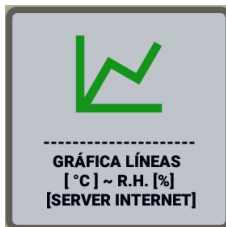
**Figura 106: Estado de temperatura
Y humedad relativa**



**Figura 107: Gráfica de estado de temperatura
y humedad Relativa**

En la figura 106, se puede apreciar la pantalla que se presenta al hacer uso de la opción correspondiente con los valores de la lectura del sensor DHT22, la temperatura °C y humedad relativa RH % en una ventana emergente, la cual se puede cerrar para observar la pantalla que se aprecia en la figura 109, en esta se puede observar la gráfica con los valores de temperatura y humedad relativa. Clic en el botón con icono “+” (color verde, parte superior) para volver a ejecutar la ventana emergente. Esta función es ejecutada en segundo plano de manera automática al mandar la orden cliqueando en la opción correspondiente (Consulta Estado [°C] ~ R.H [%]).

Opción Gráfica de Líneas °C y RH %.



Al dar clic sobre este botón se mostrará la pantalla que se puede apreciar en la figura 108.

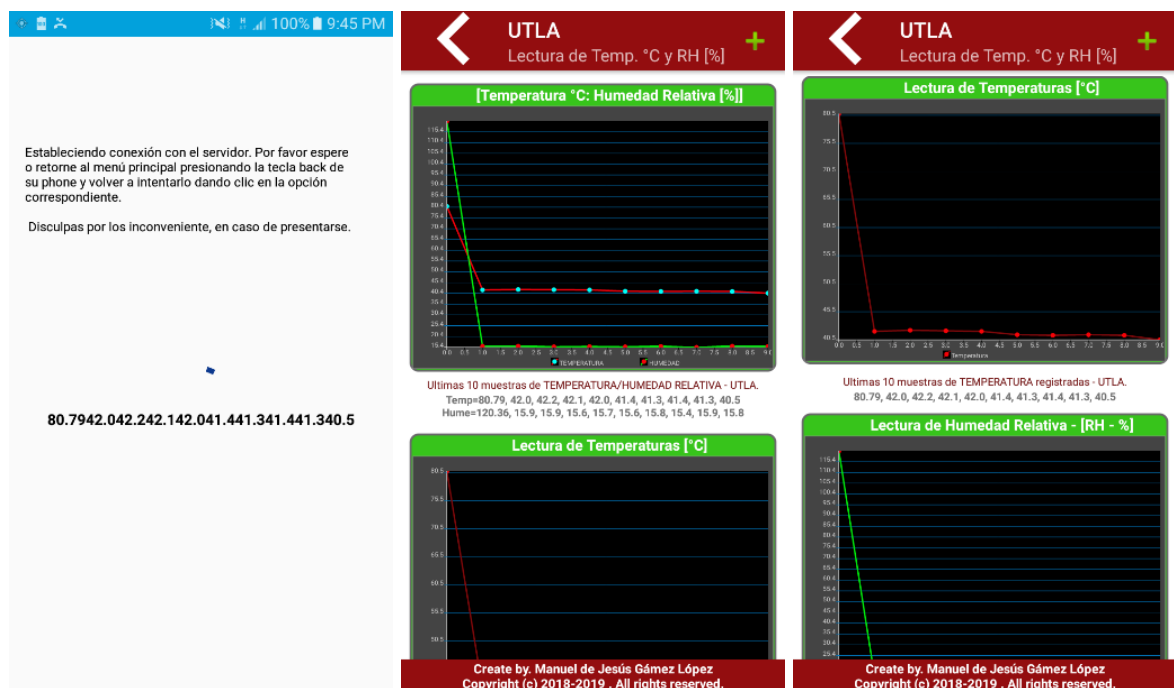
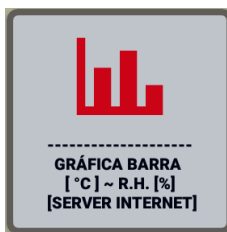


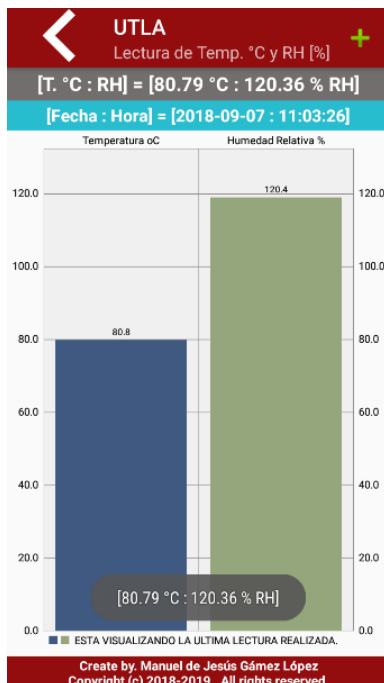
Figura 108: Gráfica de líneas, lecturas de temperatura °C y RH %.

En las gráficas que se pueden apreciar en la figura 108, se muestran las últimas 10 lecturas de temperatura °C y humedad relativa RH % almacenadas en la base de datos. Al dar clic en el icono “+” (color verde, parte superior) se hace una nueva búsqueda/consulta en la base de datos, de los últimos diez registros y se actualizan los datos en las gráficas. Se requiere que el dispositivo móvil Android esté conectado a internet o posea un plan de datos debido a que la consulta se hace a la base de datos alojada en el servidor online.

Opción Gráfica de Barras °C y RH %.



Al dar clic sobre este botón se mostrará la pantalla que se aprecia en la figura 109.



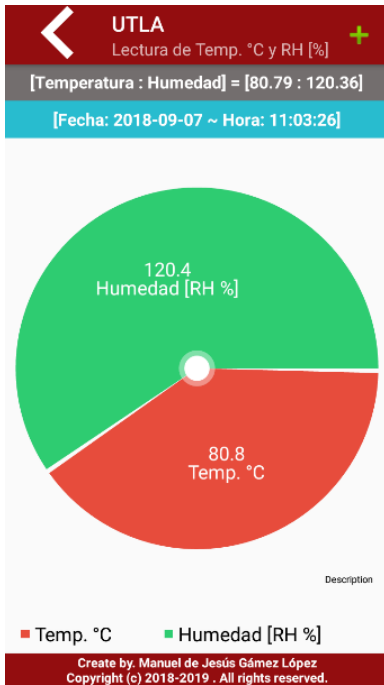
En la figura 109, se puede apreciar una gráfica de barras con los valores de la lectura del sensor DHT22, temperatura °C y humedad relativa RH % del último registro almacenado en la base de datos. Al dar clic en el icono “+” (color verde, parte superior) se hace una nueva búsqueda en la base de datos del último registro y se actualizan los datos en la gráfica. Se requiere que el dispositivo móvil Android esté conectado a internet o cuente con un plan de datos activado.

Figura 109: Gráfica Barra, lectura de último registró °C y RH %

Opción Gráfica de Pastel °C y RH %.



Al dar clic sobre este botón se mostrará la pantalla que se aprecia en la figura 110.



En la figura 110, se puede visualizar una gráfica de pastel con los valores de la lectura del sensor DHT22, temperatura °C y humedad relativa RH % del último registro almacenado en la base de datos. Al dar clic en el icono “+” (color verde, parte superior) se hace una nueva búsqueda/consulta en la base de datos del último registro y se actualizan los datos en la gráfica. Se requiere que el dispositivo móvil Android esté conectado a internet o cuente con un plan de datos activado.

Figura 110: Gráfica Pastel de lectura

de último registro °C y RH %

Opción Compartir Redes Sociales °C y RH %.



Al dar clic sobre este botón se mostrará la pantalla que se aprecia en la figura 111.

Desde esta opción se puede compartir información que es tomada de la base de datos (último registro) de temperatura °C y humedad relativa RH % a través de contactos en redes sociales, entre otros. Por ejemplo: WhatsApp, e-mail, drive y Bluetooth. Se requiere que el dispositivo móvil Android esté conectado a internet o cuente con un plan de datos activado.

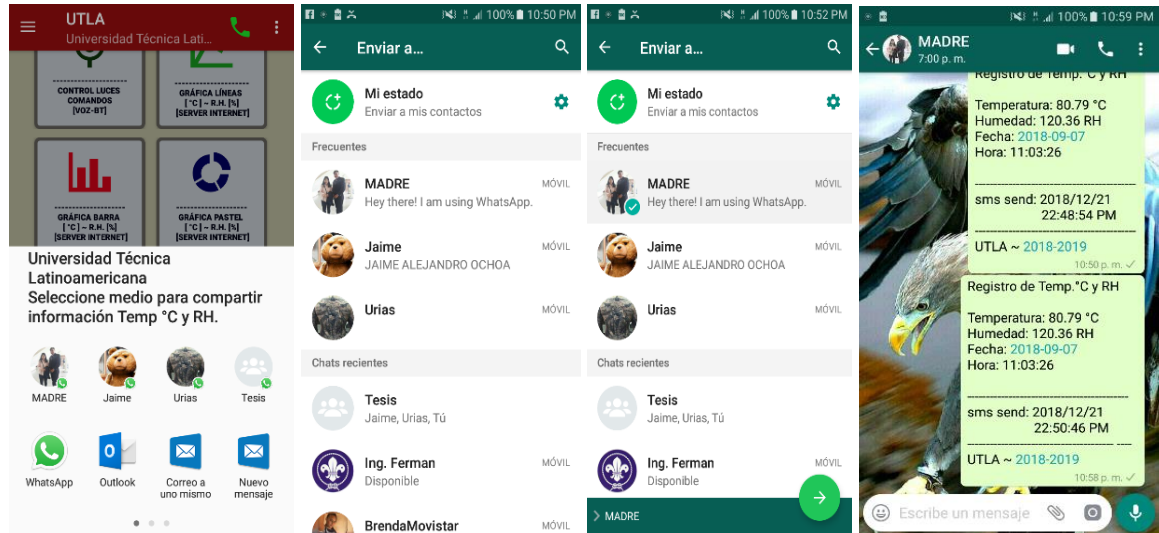
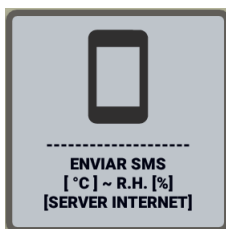


Figura 111: Compartir información de °C y RH % por WhatsApp

Opción Compartir por SMS °C y RH %.



Al dar clic sobre este botón se mostrará la pantalla que se aprecia en la figura 112.

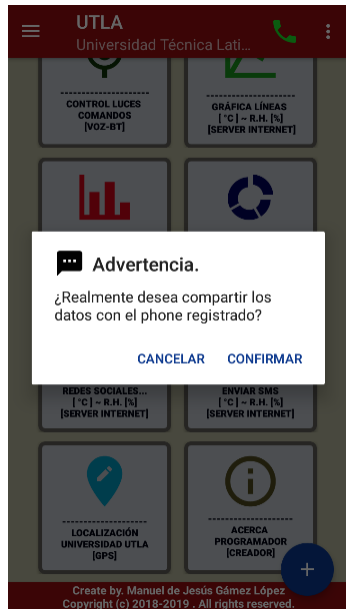


Figura 112: Envío SMS

En la figura 112, se puede apreciar el cuadro de dialogo que se lanza, en el cual, se solicita confirmación de envío de SMS. En el caso de presionar el botón CONFIRMAR se enviará el SMS con el último registro de °C y RH % almacenado en la base de datos. Se requiere que el dispositivo móvil Android posea conexión a internet o cuenta con un plan de datos activo para poder consultar a la base de datos y que el chip posea saldo para poder enviar el SMS al número de celular especificado en la opción configuración.

Opción Localización UTLA.



Al dar clic sobre este botón se mostrará la pantalla que se aprecia en las figuras 113 y 114.

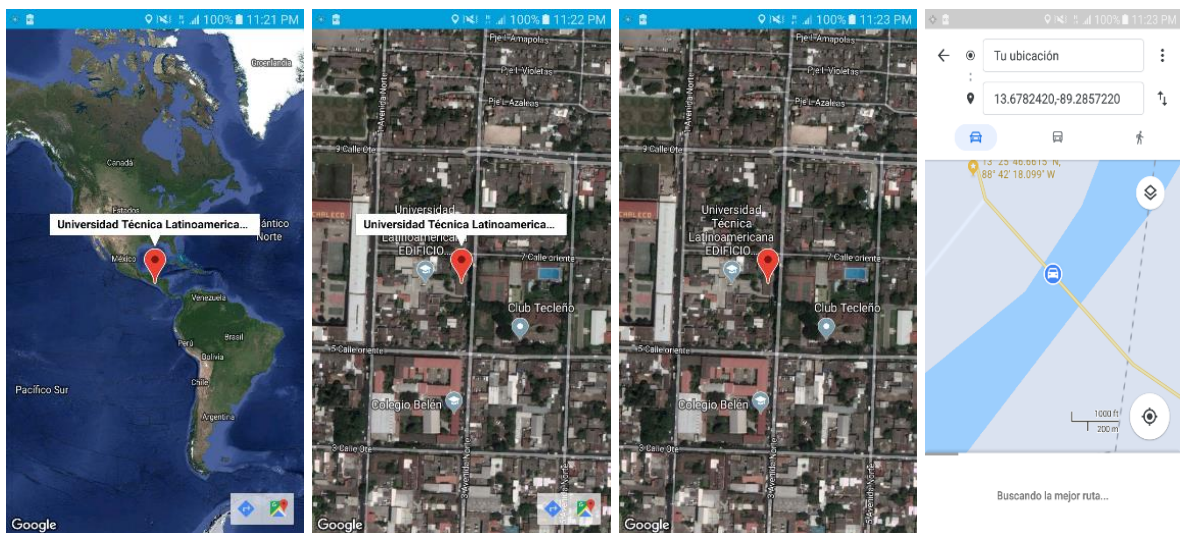


Figura 113: Ubicación de la UTLA.

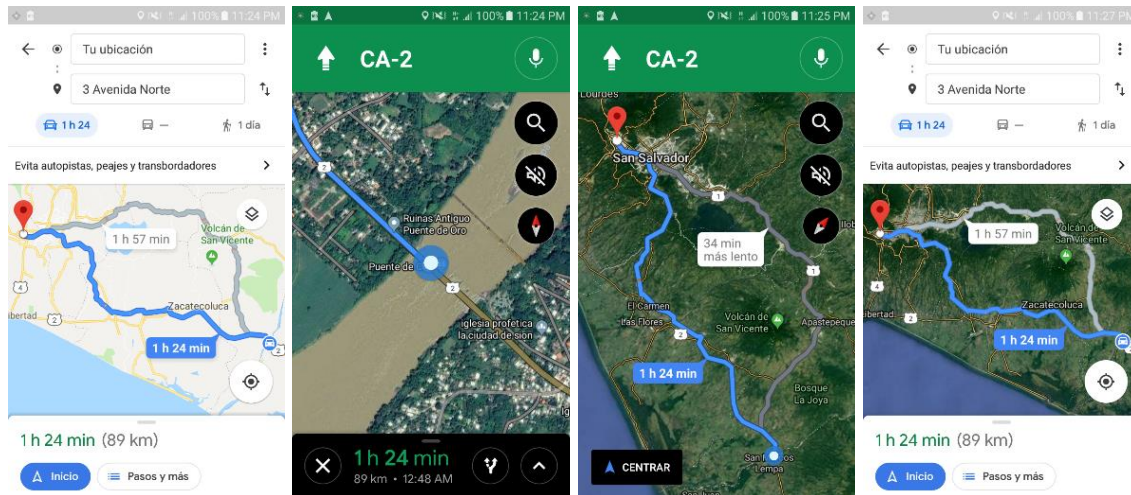


Figura 114: Demostración de uso de google map.

En las figuras 113 y 114, ubicación de UTLA, se puede apreciar que se ha realizado un uso completo del servicio de Google Maps para poder guiar a cualquier usuario (persona) desde el lugar donde se encuentre hasta la universidad. Trazándole rutas, tiempo en llegar, cantidad de kilómetros (distancia desde donde se encuentra hasta la universidad), mapa con opción de modo satélite, etc. Se requiere que el dispositivo móvil Android tenga habilitado el sensor GPS y que posea conexión a internet o plan de datos activo.



El icono de teléfono color verde ubicado en la barra de la parte superior (menú principal) permite poder realizar una llamada perdida al celular (circuito), el cual tiene la función de **“cambiar el estado de todas las luminarias”** de su estado original al inverso, además, el celular (circuito) notifica de dichos cambios a través de un SMS que hace que se muestre la pantalla del DASHBOARD con el ultimo estado de las luminarias. Se requiere saldo en el chip del dispositivo móvil Android, para poder realizar la llamada perdida y saldo en el chip instalado en el celular (circuito) para que pueda notificar dicho cambio a través de un SMS como respuesta.

Opción Acerca Programador.



Al dar clic sobre este botón se mostrará la pantalla que se aprecia en la figura 115.



En esta pantalla se visualiza información acerca del desarrollador de la App.

Figura 115: Acerca de

3.6. RESUMÉN DEL CAPÍTULO

En este capítulo se ha abordado toda la temática respecto al desarrollo de la aplicación móvil (App), se describe y se presentan las principales pantallas del sistema de control y monitoreo por SMS y CALL por medio de la red GSM desde la App. Se reitera que, para el uso de la App y el sistema electrónico que se diseñó, es necesario que ambos dispositivos electrónicos posean saldo para SMS/CALL y un plan de datos o conectividad a internet para la conexión con la base de datos en el servidor online.

Para el caso en que el control y monitoreo de las variables del sistema sea local vía Bluetooth no es necesario ningún servicio de datos o saldo.

CAPÍTULO IV

PLATAFORMA WEB, WEB SERVICE Y BASE DE DATOS

4. PLATAFORMA WEB, WEB SERVICE Y BASE DE DATOS

4.1. BREVE INTRODUCCIÓN

En este capítulo se encuentran la base de dato que almacenará los valores obtenidos por el sensor a la vez se podrá visualizar en cualquier dispositivo con conexión a internet por medio de una IP asignada en el sketch de Arduino.

4.2. PLATAFORMA WEB

Una plataforma web es más que una página web. Una plataforma incluye elementos adicionales a la página web tales como aplicaciones, carritos de compras, formularios, convertidores, instancias de aprobación y alguna otra solución específica para la necesidad del cliente y el mercado. En nuestro proyecto la página web nos mostrara los valores de temperatura del sensor DHT11 o DHT22 instalado en el Laboratorio de Electrónica como lo muestra la siguiente figura.

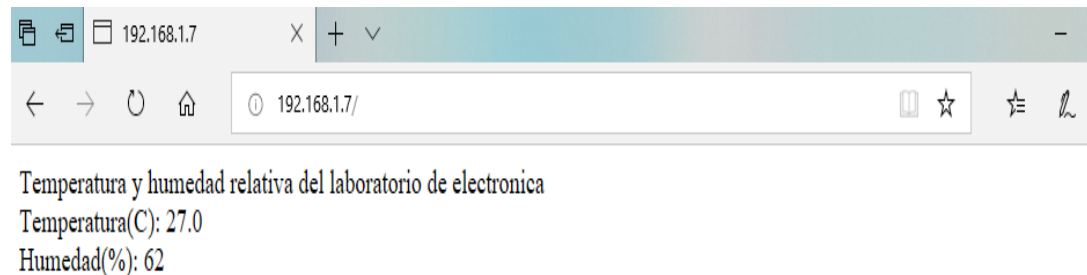


Figura 116. Lectura de temperatura y humedad relativa en navegador web

4.3. LENGUAJES DE PROGRAMACIÓN

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache o IIS con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas.

Ejemplo de sintaxis del código fuente escrito para pruebas de almacenamiento de las lecturas de temperatura y humedad relativa en la base de datos MySQL:

Script PHP

```
<?PHP
 = $_GET['h'];
$t = $_GET['t'];

$db_user = 'laboratorio utla';
$db_pass = 'utla2019';

$db = new
PDO('mysql:host=DB_URL;dbname=DB_NAME', $db_user, $db_pass);

$sql = "INSERT INTO dht11 (humidity, temperature) VALUES (:humidity,
:temperature)";

$query = $db->prepare($sql);

$query->execute(array(':humidity' => $h, ':temperature' => $t));

?>
```

4.4. SERVIDOR WEB

El Arduino Ethernet Shield permite conectar el Arduino a Internet. Está basado en el chip Wiznet W5100 con un buffer interno de 16k que es el encargado de dar una dirección IP a Arduino. Tiene un conector Ethernet RJ45 con una velocidad de 10/100Mb. Se usa la librería <Ethernet.h> para escribir el programa. También tiene un slot de tarjetas micro-SD para almacenar datos para servir a la red. Para esto, se usa la librería <SD.h>. Para comunicarse con la placa Arduino, utiliza el puerto SPI (pins 10, 11, 12 y 13) con la librería <SPI.h>, y el pin 4 si trabaja con una tarjeta micro-SD. Estos pines no pueden ser usados como entradas o salidas digitales. También tiene un botón de Reset que resetea tanto la placa Arduino como el W5100. Gracias a la tarjeta electrónica Ethernet Shield apilada en un controlador Arduino Mega, se puede usar como servidor web o como cliente web, todo depende del programa escrito y grabado en la tarjeta Arduino.

4.5. WEB SERVICE

4.5.1. BREVE INTRODUCCIÓN

Un Web Service es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. En nuestro circuito nos servirá para enviar información a través de la conectividad a la red. Además:

- ✚ Para conectar página web con sensor.
- ✚ La codificación de los mensajes debe ser en un lenguaje estándar como XML o JSON.

4.5.2. LENGUAJES DE PROGRAMACIÓN

El lenguaje de programación utilizado para el desarrollo del web Service ha sido PHP, por su gran utilidad en diversidad de aplicaciones web y la gran diversidad de fuentes de información respecto al manejo de este lenguaje de programación.

4.5.3. FUNCIONALIDAD DEL WEB SERVICE EN EL SISTEMA

El web service desarrollado para esta investigación permitirá establecer comunicación entre los diferentes componentes del sistema. Por ejemplo:

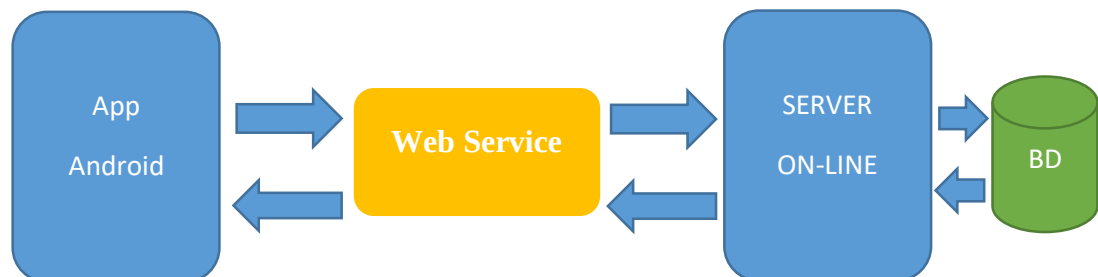


Figura 117. Funcionalidad del Web Service

4.6. BASE DE DATOS

4.6.1. BREVE INTRODUCCIÓN

En el entorno del mercado actual, la competitividad y la rapidez de maniobra de una empresa son imprescindibles para su éxito. Para conseguirlo existe cada vez una mayor demanda de datos y, por tanto, más necesidad de gestionarlos. Esta demanda siempre ha estado patente en empresas y sociedades, pero en estos años se ha disparado debido al acceso multitudinario a las redes integradas en Internet y a la aparición de los dispositivos móviles que también requieren esa información. Nos almacenará los datos obtenido del sensor para poder consultar luego o cuando lo deseemos también se podrá graficar en una página web si deseamos.

4.6.2. LENGUAJE ESTRUCTURADO DE CONSULTA (SQL)

El lenguaje estructurado de consultas, más comúnmente llamado SQL, lo conforman una serie de comandos, cláusulas y funciones que permiten realizar cualquier operación sobre la información almacenada en la base de datos.

SQL no es exactamente un lenguaje de programación, pero lleva implícita la complejidad de estas herramientas. En realidad, los sistemas gestores de bases de datos como Microsoft Access o el mismo OpenOffice Base tienen como principal fin ocultar el lado oscuro de este lenguaje mediante el uso de utilidades gráficas. Aunque si tu intención es llegar a dominar el mundo de las bases de datos es imprescindible conocer ciertos aspectos de SQL

4.6.3. SISTEMAS GESTORES DE BASE DE DATOS

Un Sistema Gestor de Base de Datos (SGBD, en inglés DBMS: DataBase Management System) es un sistema de software que permite la definición de bases de datos; así como la elección de las estructuras de datos necesarios para el almacenamiento y búsqueda de los datos, ya sea de forma interactiva o a través de un

lenguaje de programación. Un SGBD relacional es un modelo de datos que facilita a los usuarios describir los datos que serán almacenados en la base de datos junto con un grupo de operaciones para manejar los datos.

Los SGBD relacionales son una herramienta efectiva que permite a varios usuarios acceder a los datos al mismo tiempo. Brindan facilidades eficientes y un grupo de funciones con el objetivo de garantizar la confidencialidad, la calidad, la seguridad y la integridad de los datos que contienen, así como un acceso fácil y eficiente a los mismos.

4.6.4. SOFTWARE PARA DISEÑO DE BASE DE DATOS

Registro de datos a la base de datos usando el escudo shield Ethernet y controlador Arduino.

Para esta investigación, se realizaron las pruebas de almacenamiento de registros de los datos en la base de datos utilizando Arduino Ethernet Shield. Se almacenan los datos del sensor de temperatura y humedad relativa DHT22 en la base de datos y luego los mostraremos en el navegador web.

Para crear la base de datos, se utilizó el servidor XAMPP PANEL CONTROL, el cual incluye todas las aplicaciones necesarias para el diseño de la base de datos en MySQL, web service y plataforma web, utilizando el lenguaje de programación PHP. Se listan en la siguiente tabla las aplicaciones que se incluye predeterminadamente en XAMPP:

Servidor Web	Apache
Servidor de aplicaciones	Lenguaje de programación PHP (Preprocesador de hipertexto)
Gestor de base de datos	MySQL
Herramienta de diseño de base datos.	MySQL
Otros.	

Tabla 8. Listado de software en XAMPP PANEL CONTROL



Figura 118. Logo XAMPP PANEL CONTROL

Instalación y configuración de XAMPP.

- Descargar XAMPP PANEL CONTROL
- Iniciar el instalador de XAMPP
- Luego de instalado, ejecutar XAMPP PANEL CONTROL como administrador, haciendo clic derecho sobre el icono correspondiente en el escritorio y seleccionar la opción “Ejecutar como administrador”.
- Se presentará la pantalla que se muestra en la figura 119.
- Iniciar los servicios Apache y MySQL.

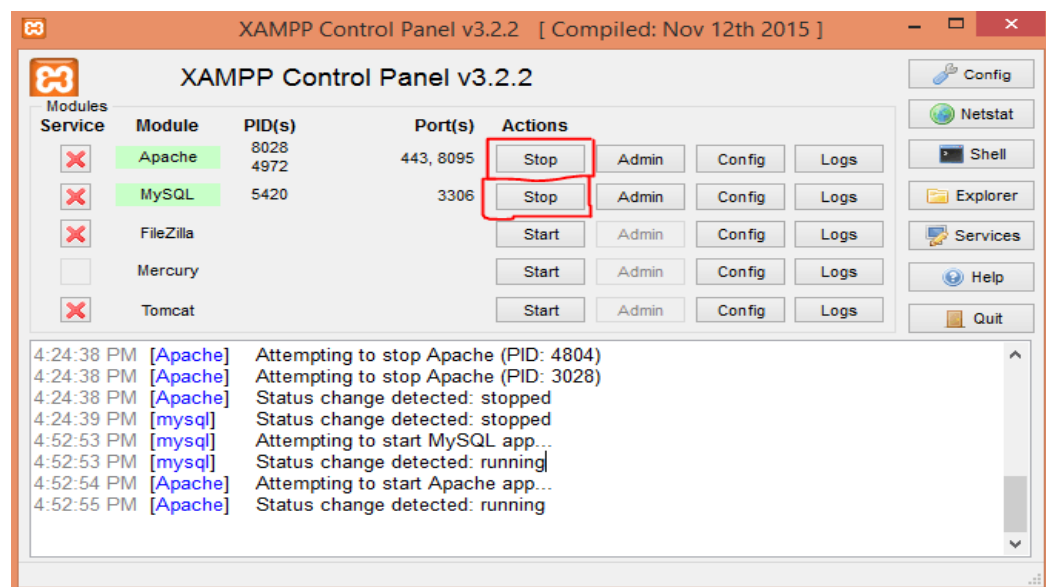


Figura 119. Pantalla XAMPP PANEL CONTROL

4.6.5. CREACIÓN DE LA BASE DE DATOS

Lo primero es abrir el navegador web y escriba "**localhost:80/phpmyadmin/**", Solicitará un nombre de usuario y contraseña, el nombre de usuario predeterminado es "**root**" y la contraseña en blanco. Después de dar el nombre de usuario y la contraseña, se abrirá el administrador grafico **phpmyadmin**.

Luego se busca la pestaña de bases de datos y se le da cree nueva base de datos. Se le da el nombre de la base de datos como "**laboratorioutla**". En la figura 120, se puede apreciar el administrador de base de datos PHPMYADMIN desde un navegador web.



Figura 120. Creación de nueva base de datos con phpmyadmin

4.6.6. DIAGRAMAS DE LA BASE DE DATOS

Cuando se utiliza una base de datos para gestionar información, se está plasmando una parte del mundo real en una serie de tablas, registros y campos ubicados en un ordenador; creándose un modelo parcial de la realidad. Antes de crear físicamente estas tablas en el ordenador se debe realizar un modelo de datos.

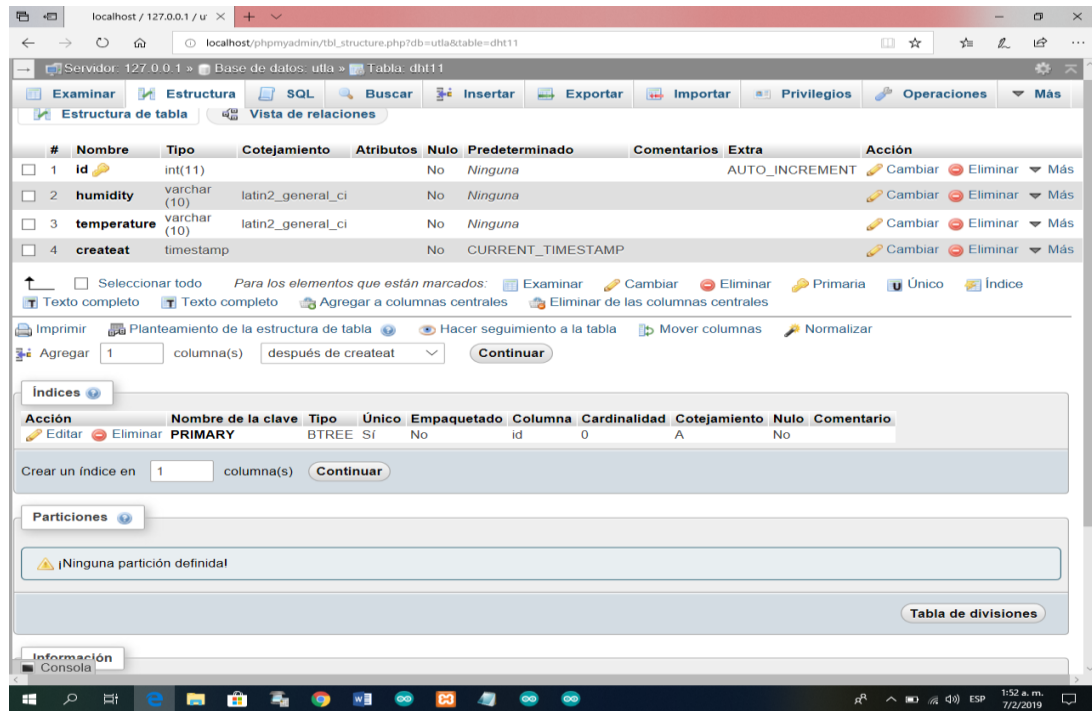


Figura 121. Modelo de datos DHT11/DHT22

mjgl_tesis usuarios idusuario : int(11) nombre : varchar(90) apellido : varchar(70) email : varchar(100) usuario : varchar(45) clave : varchar(50) # intentos : int(11) # bloqueo : int(11) pregunta : varchar(100) respuesta : varchar(50) estado : varchar(40) fecha : timestamp isadmin : varchar(45) ultimoingreso : timestamp razon : varchar(200)	mjgl_tesis sensores id_sensor : int(11) descripcion : varchar(80) # temperatura : decimal(10,2) # humedad : decimal(10,2) # luz1 : int(11) # luz2 : int(11) # luz3 : int(11) # luz4 : int(11) # luz5 : int(11) # luz6 : int(11) # luz7 : int(11) # luz8 : int(11) # test : int(11) fecha : date hora : varchar(15) mes : int(11)	mjgl_tesis App_phone codigo : int(11) descripcion : varchar(1000) numCelularCircuito : varchar(25) numCelularResponsable : varchar(25) fecha : varchar(25) hora : varchar(25)	mjgl_tesis muestras id : int(11) # cantidad : int(11) fecha : timestamp
---	---	--	---

Figura 122. Base de datos del sistema

4.6.7. PROTOTIPO Y DISEÑO ELECTRÓNICO.

En la figura 123, se puede apreciar el montaje físico realizado para efectuar pruebas de comunicación y funcionalidad vía Ethernet, tomando información proveniente del sensor DHT11. Pudiendo demostrar con ello, que se obtuvieron los resultados esperados después de realizar una diversidad de pruebas y correcciones necesarias para lograr la funcionalidad deseada.

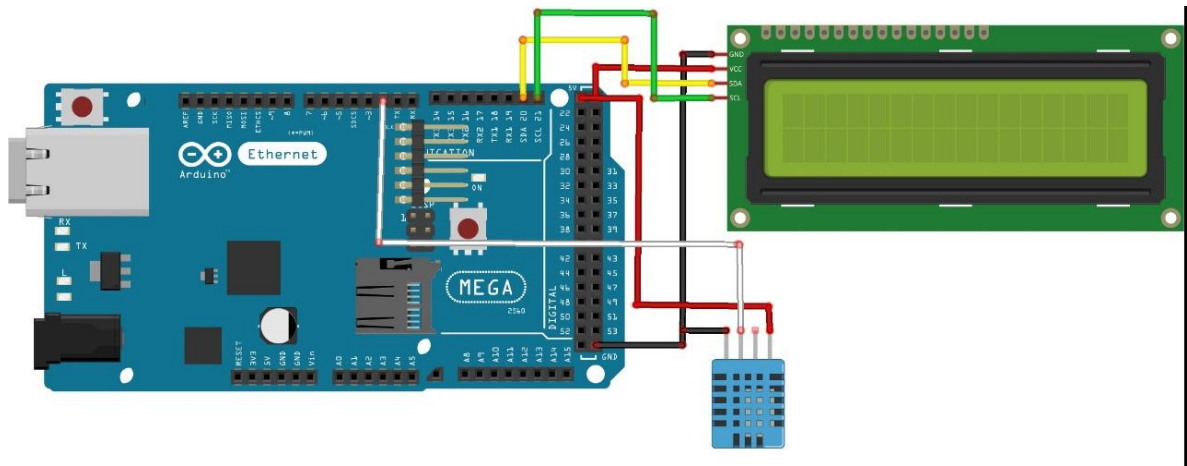


Figura 123. Montaje electrónico Shield Ethernet con Arduino MEGA

4.6.8. CÓDIGO FUENTE PARA CONTROL DE SHIELD ETHERNET CON ARDUINO MEGA.

```

sketch_feb08b Arduino 1.8.1
Archivo Editar Programa Herramientas Ayuda

sketch_feb08b g
1 #include <Wire.h>
2 #include <LiquidCrystal_I2C.h>
3 #include <DHT.h>
4 #include <SPI.h>
5 #include <Ethernet.h>
6
7 //I2C LCD:
8 LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
9 DHT dht(12, DHT11);
10
11 // Se introducen la ip y la mac.
12 byte mac[] = {
13   0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
14 IPAddress ip(192,168,1,7);
15
16 // Inicializa la libreria ethernet, el puerto 80 esta por defecto para HTTP
17 EthernetServer server(80); //Abrir el puerto 125 en tu router si quieres poder verlo fuera de la red local.
18
19 void setup()
20 {
21   // Abrir comunicacion serial y esperar que el puerto responda
22   Serial.begin(9600);
23   while (!Serial) {
24     ; // Esperar a que el puerto serial se conecte (solo cuando usamos Arduino Leonardo)
25   }
26   //Inicializar la conexion Ethernet y el servidor
27   Ethernet.begin(mac, ip);
28   server.begin();
29   Serial.print("server is at ");
30   Serial.println(Ethernet.localIP());
31
32   // PREPARAR LA COMUNICACION SERIAL
33   lcd.begin(16,2);
34   Serial.println("Prueba sensor dht11 laboratorio de electrónica Utlá");
35
36   // PREPARAR LA LIBRERIA PARA COMUNICARSE CON EL SENSOR
37   dht.begin();
38 }
39
40 void loop() {
41   // Reading temperature or humidity takes about 250 milliseconds!
42   // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
43   int h = dht.readHumidity();
44   int t = dht.readTemperature();
45
46   // set the cursor to (0,0):
47   lcd.setCursor(0, 0);
48   // print from 0 to 9:
49
50   lcd.print("Temp: ");
51   lcd.print(t);
52   lcd.print("C");

```

```

54 // set the cursor to (16,1):
55 lcd.setCursor(0,1);
56 lcd.print("Humidity: ");
57 lcd.print(h);
58 lcd.print("%");
59
60 Serial.print("Temp: ");
61 Serial.print(t);
62 Serial.print("C, Humidity: ");
63 Serial.print(h);
64 Serial.println("%");
65
66 {
67 //Escucha a clientes de entrada
68 EthernetClient client = server.available();
69 if (client) {
70     Serial.println("new client");
71
72     boolean currentLineIsBlank = true;
73     while (client.connected()) {
74         if (client.available()) {
75             char c = client.read();
76             Serial.write(c);
77             // Si la linea en la solicitud http esta en blanco y el cliente esta disponible se puede enviar una
78             if (c == '\n' && currentLineIsBlank) {
79                 client.println("HTTP/1.1 200 OK");
80                 client.println("Content-Type: text/html");
81                 client.println("Connection: close");
82                 client.println();
83                 client.println("<!DOCTYPE HTML>");
84                 client.println("<html>");
85                 // Agregar una etiqueta meta que recargue la pagina cada segundo
86                 client.println("<meta http-equiv='refresh' content='5'>");
87                 client.print("Temperatura y humedad relativa del laboratorio de electronica");
88
89                 client.println("<br />");
90
91                 //Obtener las lecturas del sensor e imprimir el resultado
92                 // La lectura de la temperatura o de la humedad tarda aproximadamente 250 milisegundos en el sensor
93                 float h = dht.readHumidity(); //Guarda la lectura de la humedad en la variable float h
94                 float t = dht.readTemperature(); //Guarda la lectura de la temperatura en la variable float t
95
96                 //Escribimos en la página los valores de temperatura y humedad que hemos leído previamente
97                 client.print("Temperatura(C): ");
98                 client.println(t,1);
99                 client.println("<br />");
100                 client.print("Humedad(%): ");
101                 client.println(h, 0);
102                 client.println("<br />");
103
104                 //Termina lectura del sensor
105                 client.println("</html>");
106
107                 break;
108             }
109             if (c == '\n') {
110                 //Linea en blanco
111                 currentLineIsBlank = true;
112             }
113             else if (c != '\r') {
114                 //Si hay un caracter en la linea en blanco
115                 currentLineIsBlank = false;
116             }
117             }
118         }
119     }
120     //Darle al navegador web tiempo para recibir los datos
121     delay(1000); //Tiene un retardo de 1000 ms = 1s para repetir el ciclo void loop.
122     //Se cierra la conexcion
123     client.stop();
124     Serial.println("client disconnected");

```

```
122 }  
123 } //Termina loop  
124
```

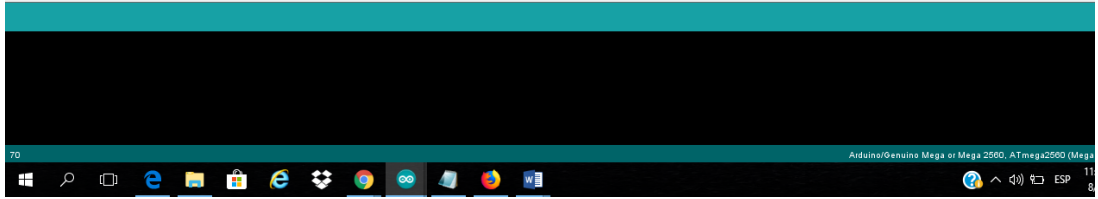


Figura 124. Código fuente de prueba Shield Ethernet y Arduino Mega

4.7. RESUMÉN DEL CAPÍTULO

En este capítulo se ha presentado la información relacionada a las herramientas de software desarrolladas como parte de la funcionalidad del sistema, para la visualización de la información desde la App y la plataforma web, se aprecian las evidencias de las pruebas realizadas, en cuanto a la funcionalidad, durante el proceso de desarrollo e integración de las partes que comprende el sistema, Hardware (Circuito eléctrico y electrónico), Software (Plataforma web y web service) y la base de datos.

Es posible la conexión del controlador Arduino con servidores locales e Internet con base de datos para el almacenamiento, procesamiento y análisis de los datos proporcionados por los sensores, haciendo uso Software y Hardware libre abaratando costos en comparación a otros sistemas existentes y además se añade valor agregado haciendo una integración con otras tecnologías (Aplicación móvil Android) para la visualización de la información obtenida del ambiente por los diferentes sensores y el control de variables TODO/NADA del sistema.

CAPÍTULO V

SISTEMA DE MONITOREO Y AUTOMATIZACIÓN

5. SISTEMA DE MONITOREO Y AUTOMATIZACIÓN

5.1. BREVE INTRODUCCIÓN

A lo largo de este capítulo se detallan todas las fases de diseño del sistema de automatización tanto a nivel de Hardware como de Software. Así, se demuestra los prototipos ensamblados para la realización de las pruebas efectuadas, para que el sistema realice los procesos requeridos por fase.

5.2. MONITORIZACIÓN DE PARÁMETROS ENTRADAS Y SALIDAS

Para el desarrollo de la monitorización de los parámetros que involucra este sistema se trató de emplear sensores que fueran posible conectar directamente con la placa Arduino y evitar así, el hecho de crear circuitos de adaptación en la medida que fuera posible o al menos evitar circuitos complejos de adaptación por cuestiones económicas y reducción del tamaño del prototipo final. Por otro lado, se mencionó anteriormente que los pines de Arduino entregan corrientes muy bajas de salida que normalmente oscilan entre 25 mA a un máximo de 40 mA por pin, por tanto, las salidas de Arduino requieren ser amplificadas para poder actuar sobre los procesos que se manejarán con este sistema, consecuentemente, a continuación, se procede a detallar como serán manejadas las entradas y salidas del sistema, teniendo presente lo anterior.

5.3. ENTRADAS DEL SISTEMA

Mencionar que en el capítulo I ya se explicó los tipos de sensores que se utilizarán, así como sus características, de modo que solo queda exponer como quedan conectados y como se tratará la información que se maneja en dicho sistema. Así la información a tener en cuenta es la siguiente.

5.4. SENSOR DE MEDICIÓN DE TEMPERATURA Y HUMEDAD RELATIVA

Los sensores DHT11 y DHT22 son sensores digitales de temperatura y humedad relativa, se pueden implementar con cualquier microcontrolador. Utiliza un sensor capacitivo de humedad y un termistor para medir el aire circundante y solo un pin para la lectura de los datos. Tal vez la desventaja de estos es la velocidad de las lecturas y el tiempo que hay que esperar para tomar nuevas lecturas (nueva lectura después de 2 segundos), pero esto no es tan importante puesto que la temperatura y humedad relativa son variables que no cambian muy rápido en el tiempo.

Sensor DHT11

Este sensor trabaja con un rango de medición de temperatura de 0 a 50 °C con precisión de ± 2.0 °C y un rango de humedad de 20% a 90% RH con precisión de 4% RH. Los ciclos de lectura deben ser como mínimo 1 o 2 segundos. Ver figura 125.

Sensor DHT22

El rango de medición de temperatura es de -40°C a 80 °C con precisión de ± 0.5 °C y rango de humedad de 0 a 100% RH con precisión de 2% RH, el tiempo entre lecturas debe ser de 2 segundos. Ver figura 126.

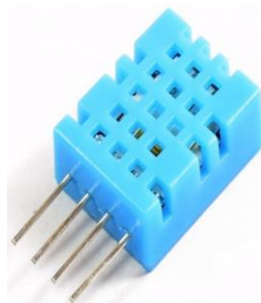


Figura 125: Sensor DHT11

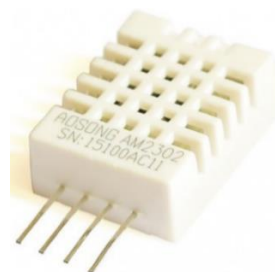


Figura 126: Sensor DHT22

Como se puede identificar la diferencia entre estos sensores es solo el rango, precisión y para el caso del DHT11 puede soportar ciclos más rápidos de lectura.

El sensor utilizado es el DHT22 por las características ya mencionadas para el monitoreo de las variables temperatura y humedad relativa en el Laboratorio de Electrónica, el cual está conectado al controlador Arduino como se puede apreciar en la figura 42.

5.5. SENSOR DEPENDIENTE DE LUZ (LDR)

El módulo sensor LDR se utiliza para detectar la intensidad de la luz. Se asocia con el pin de entrada analógica en Arduino y el pin de salida digital etiquetado como OUT en la placa LDR. Cuando hay luz, la resistencia del sensor LDR será baja proporcionalmente con la intensidad de la luz medida e inversamente sino se detecta oscuridad. El sensor tiene una perilla tipo tornillo (color dorado) desde el cual se puede ajustarse la sensibilidad del LDR hacia la luz. Ver figura 196.



Figura 127: Modulo del sensor LDR

Este sensor se ha utilizado para el monitoreo del estado de las luminarias del Laboratorio de Electrónica, el cual está conectado al controlador Arduino como se puede apreciar en la figura 46.

5.6. PROCESAMIENTO DE LA INFORMACIÓN DE LOS SENSORES

Para el manejo de la información por el sistema de manera automática y por parte de los usuarios autorizados de manera manual, se ha dispuesto en el sistema las siguientes opciones:

- ✚ El sistema automáticamente queda programado para estar enviando registros con los valores de las lecturas que realizan los sensores a la base de datos en el servidor.
- ✚ Se dota al sistema con la opción de contar con un potenciómetro/perilla para variar o controlar los tiempos de lecturas y almacenamiento del registro en la base de datos. Además, se pone un SWITCH (interruptor) para habilitar o deshabilitar esta función de control de tiempo para las lecturas.
- ✚ Finalmente, se incorpora la opción para realizar peticiones de consulta de información captada por los sensores desde la App vía SMS y CALL utilizando la red GSM.

Por tanto, los sensores y todo el sistema deben estar activo permanentemente y será la programación realizada y grabada a la tarjeta Arduino la que maneje la información recibida.

5.7. SALIDAS DEL SISTEMA

En este caso, dado que hay que operar sobre un sistema de control de cargas de entre 120 Vac a 220 Vac, y teniendo en cuenta que los pines de salida de la tarjeta Arduino suministran una corriente máxima de 40 mA, se hace necesario amplificar dichas señales. De esta manera se evaluó de entre las opciones posibles para la amplificación de la señal electrónicamente la más económica y factible para lograrlo, decidiendo finalmente, realizarlo mediante el empleo de relés conectados a los pines de salida de la tarjeta Arduino. Así, se emplearon módulos de relés (Teniendo en disposición: Tarjetas con 1 relé, con 2 relés, 4 relés, 8 relés, etc.). En la figura 127, se puede apreciar un módulo de dos relés que se utilizó para efectuar las pruebas junto con su esquema eléctrico y detalle de los componentes electrónicos que esta incorpora dentro de los cuales se mencionan algunos: Optoacopladores, transistores, diodos, y resistencias.

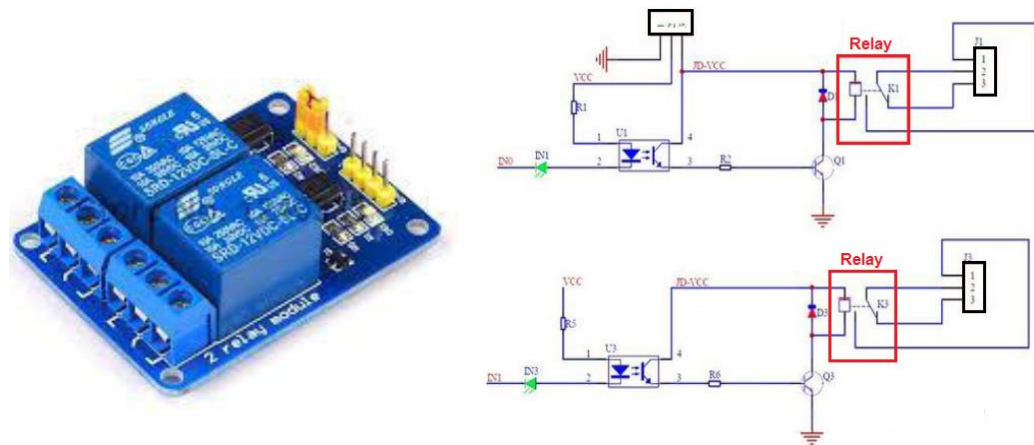


Figura 128: Módulo de 2 relés

Las características de este módulo relé permiten conectarlo directamente a la placa Arduino. En todo caso es necesario que las tarjetas de relés sean alimentadas a través de una fuente de voltaje externa ya que puede darse el escenario en que estén activos varios relés al mismo tiempo y esto puede provocar un comportamiento no deseado en la tarjeta Arduino por sobre demanda de corriente. Consecuentemente se hace necesario colocar una fuente de voltaje que cumpla con la demanda necesaria de tensión y corriente.

Por último, hay que mencionar que este tipo de relé puede soportar una corriente máxima de salida de 10A a 220 Vac, por lo que en caso que sea necesaria más corriente o tensión para activar otras cargas con mayor demanda, se hace necesario recurrir a contactores de mayor potencia, como el que se puede apreciar en la figura 129, de los cuales serían pilotados por el sistema mediante las tarjetas de relés conectando su bobina de control a los contactos de salida del relé. Con todo, queda fuera de los objetivos del proyecto profundizar en estos aspectos, por lo que solo se comenta a modo de conocimiento para el lector y evidenciar la investigación realizada al respecto.



Figura 129: Contactor

5.8. DISEÑO DEL CIRCUITO

Esta parte consiste en mostrar los principales diagramas o diseños de circuitos utilizados para la implementación del sistema, de este modo basta con conectar los sensores que se comentaron en el capítulo I junto con los relés que se encargan de accionar el sistema de control de luminarias.

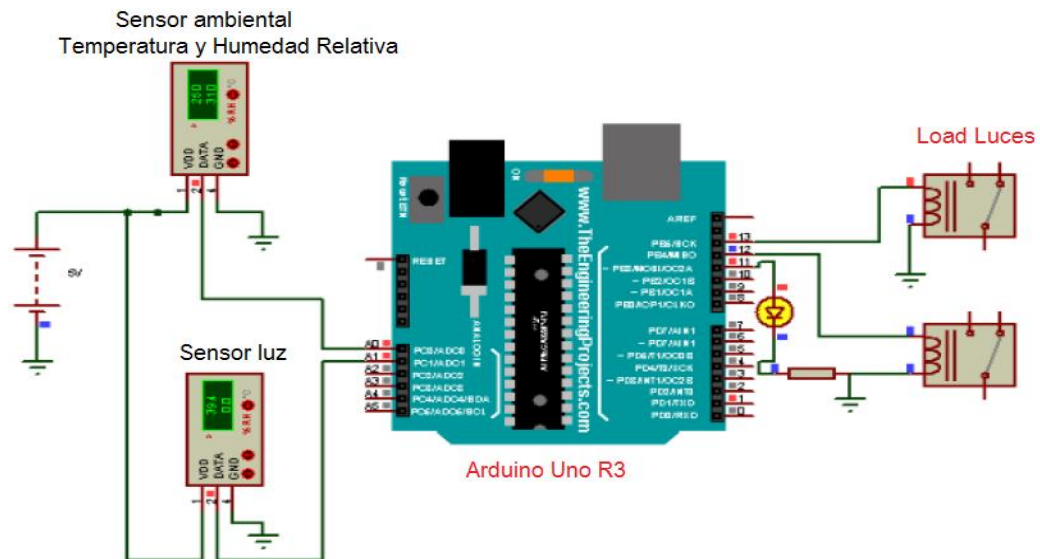


Figura 130: Simulación circuito de automatización (Entradas y Salidas)

Además, se presentan los diagramas de la instalación eléctrica que se realizó para

pruebas en el prototipo, juntamente con los correspondientes relés para el control de las luminarias tanto a nivel local desde los interruptores comunes de pared y electrónicamente desde Arduino vía Bluetooth, SMS y CALL (Red GSM) a través de la App.

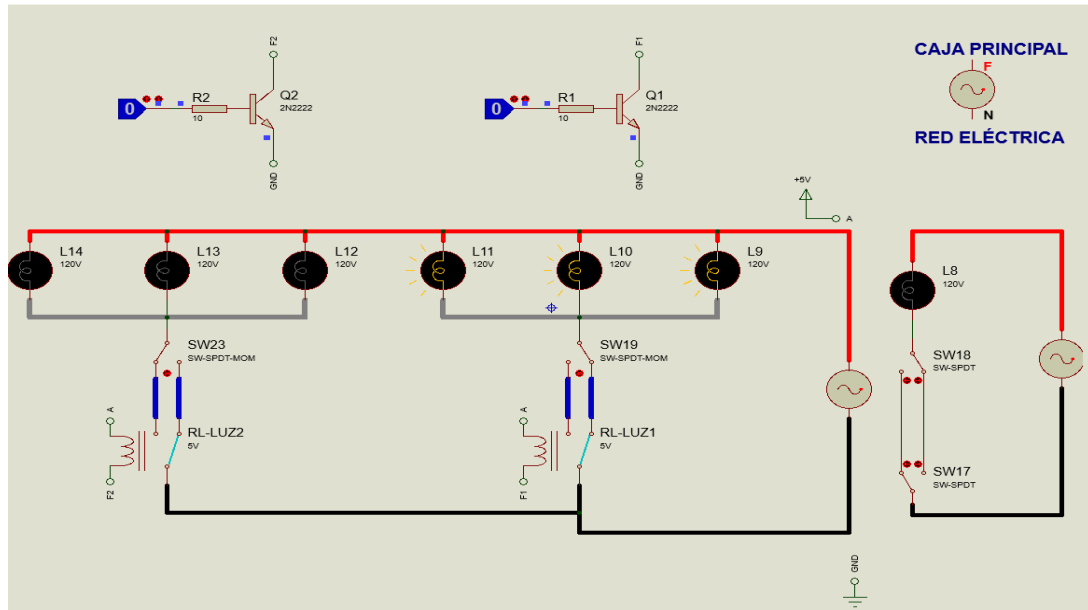


Figura 131: Simulación del plano de la instalación eléctrica/electrónica

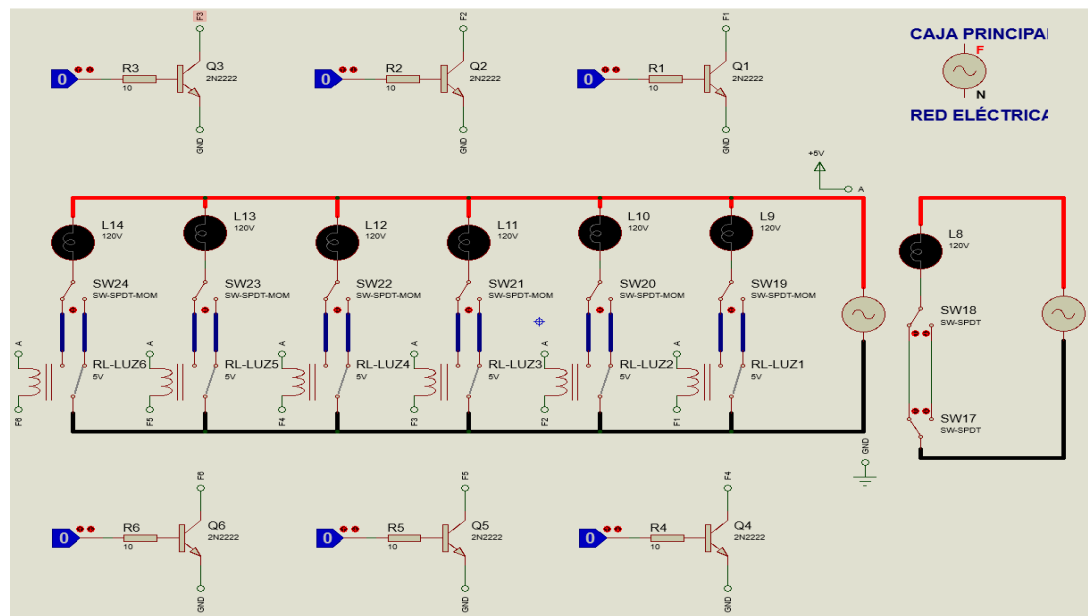


Figura 132: Plano de la instalación eléctrica/electrónica

En las figuras 131 y 132, se muestran las correspondientes intervenciones electrónicas que se realizan a la instalación eléctrica para la automatización de las luminarias.

La principal idea de las intervenciones realizadas es tener un control de luminarias parecido al control que se realiza de una misma luminaria desde dos puntos diferentes. Ver figura 133.

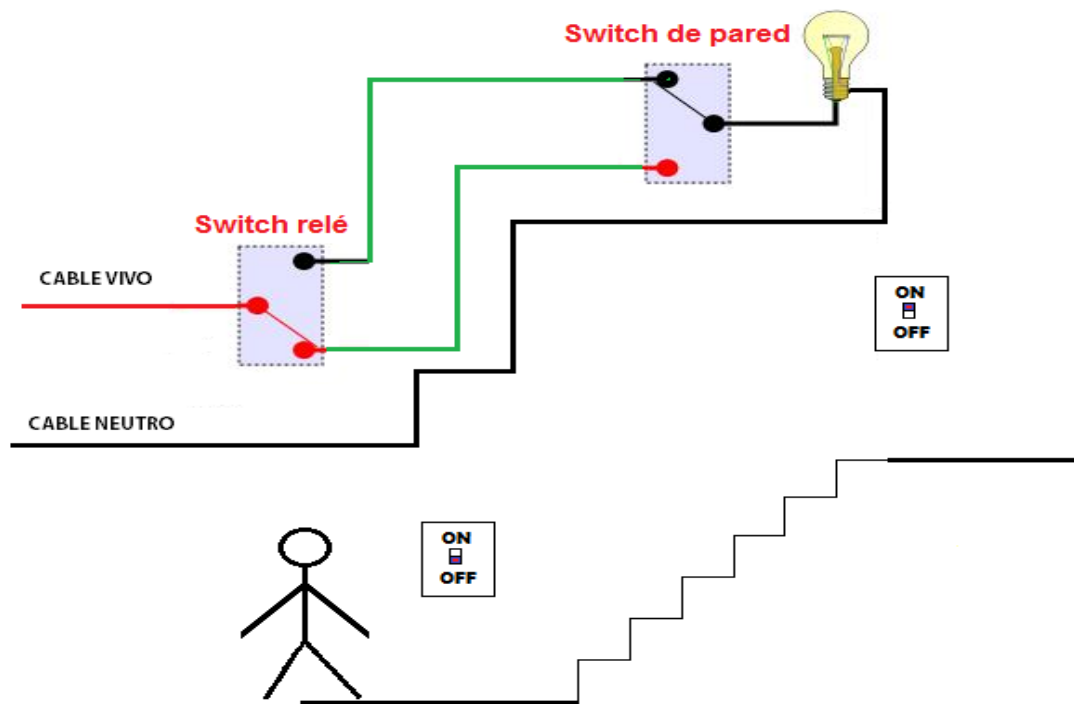


Figura 133: Diagrama de conexión y control de luminarias desde 2 puntos

5.9. SOFTWARE DE MONITORIZACIÓN Y AUTOMATIZACIÓN (SKETCH)

El código realizado para la medición de las variables del sistema ha sido escrito en el IDE de Arduino. En la figura 134, se puede ver la versión básica de lecturas de temperatura y humedad relativa a través del sensor DHT22.

```
#include "DHT.h"

#define DHTPIN 2    // Pin donde está conectado el sensor

//#define DHTTYPE DHT11    // Descomentar si se usa el DHT 11
#define DHTTYPE DHT22    // Sensor DHT22

DHT dht(DHTPIN, DHTTYPE);

void setup() {
    Serial.begin(9600);
    Serial.println("Iniciando...");
    dht.begin();
}

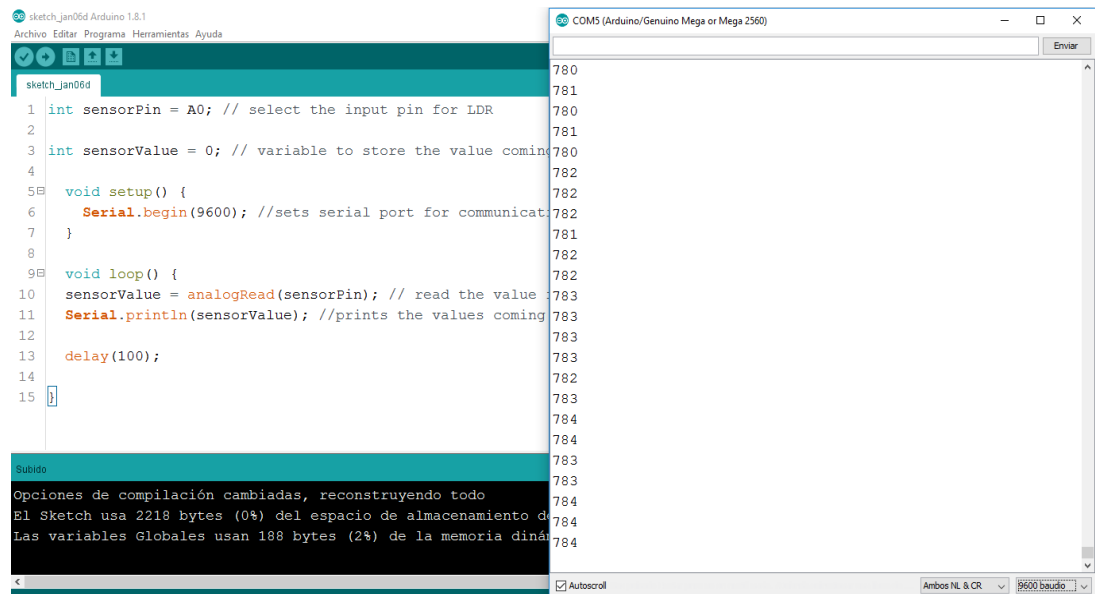
void loop() {
    delay(2000);
    float h = dht.readHumidity(); //Leemos la Humedad
    float t = dht.readTemperature(); //Leemos la temperatura en grados Celsius
    float f = dht.readTemperature(true); //Leemos la temperatura en grados Fahrenheit
    //-----Enviamos las lecturas por el puerto serial-----
    Serial.print("Humedad ");
    Serial.print(h);
    Serial.print(" %t");
    Serial.print("Temperatura: ");
    Serial.print(t);
    Serial.print(" *C ");
    Serial.print(f);
    Serial.println(" *F");
}
```

Figura 134: Código fuente sensor dht22

Para el caso que se desee trabajar con el sensor DHT11 en vez del sensor DHT22 con el código fuente mostrado en la figura 136, bastará con quitar el comentario a la línea: **“#define DHTTYPE DHT11”** y se pone en comentario la línea **“//#define DHTTYPE DHT22”**. Poner como comentario una línea de código es como que no existiera escrita para el compilador³, pero si para el programador. Por último, mencionar que el sensor DHT11 solo puede realizar y devolver lecturas de información ambiente en forma de dato entero, no es posible con este sensor las lecturas de datos flotantes o decimal, solamente el sensor DHT22 devuelve lecturas con valores más exactos por manejar datos flotantes.

En las figuras 135 y 136 se muestran las lecturas del módulo sensor LDR en ausencia y presencia de luz (detección) correspondientemente. La versión completa e integrada en un único archivo “Sketch.ino” se anexará al final del documento o se compartirá como un archivo en el CD/DVD que se entregará.

³ Entidad encargada de convertir el código fuente escrito en lenguaje de alto nivel a código máquina.

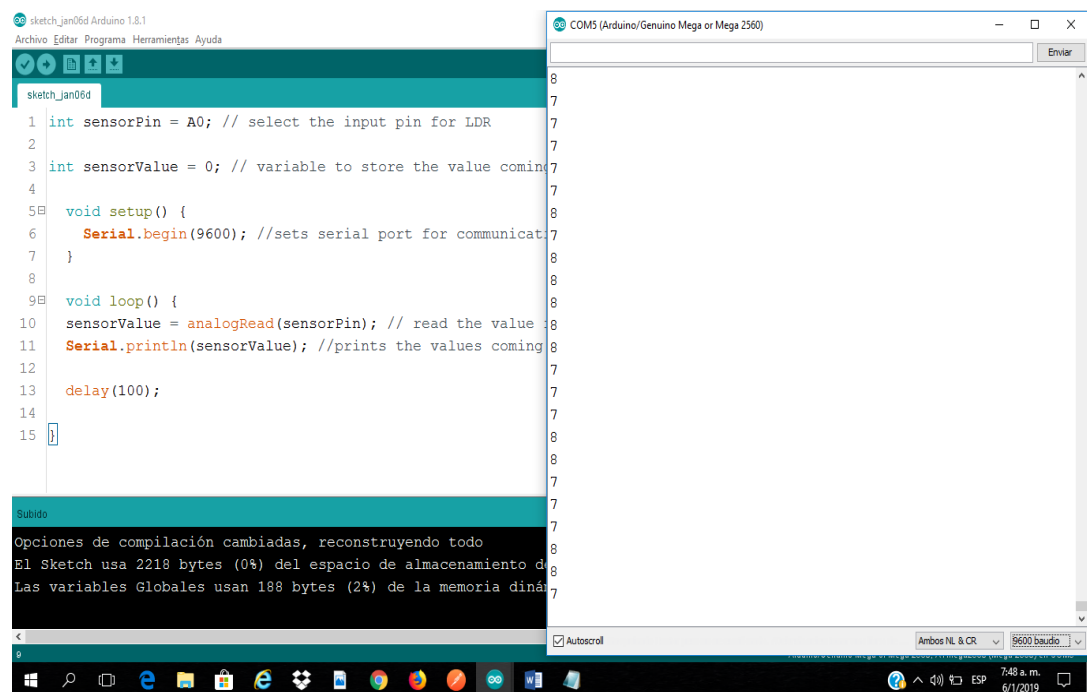


The screenshot shows the Arduino IDE interface. The main editor displays the following code:

```
1 int sensorPin = A0; // select the input pin for LDR
2
3 int sensorValue = 0; // variable to store the value coming from the sensor
4
5 void setup() {
6   Serial.begin(9600); //sets serial port for communication
7 }
8
9 void loop() {
10  sensorValue = analogRead(sensorPin); // read the value from the sensor
11  Serial.println(sensorValue); //prints the values coming from the sensor
12
13  delay(100);
14
15 }
```

The serial monitor on the right shows the output of the program, displaying the value 780 repeatedly. The status bar at the bottom indicates 'Ambos NL & CR' and '9600 baudo'.

Figura 135: Código fuente sensor LDR – Lectura de ausencia de luz



The screenshot shows the Arduino IDE interface. The main editor displays the following code:

```
1 int sensorPin = A0; // select the input pin for LDR
2
3 int sensorValue = 0; // variable to store the value coming from the sensor
4
5 void setup() {
6   Serial.begin(9600); //sets serial port for communication
7 }
8
9 void loop() {
10  sensorValue = analogRead(sensorPin); // read the value from the sensor
11  Serial.println(sensorValue); //prints the values coming from the sensor
12
13  delay(100);
14
15 }
```

The serial monitor on the right shows the output of the program, displaying the value 7 repeatedly. The status bar at the bottom indicates 'Ambos NL & CR' and '9600 baudo'.

Figura 136: Código fuente sensor LDR – Lectura de presencia de LUZ

Finalmente, se menciona que los Sketch y diagramas presentados son solo ejemplos básicos de fases por las que se pasó para llegar al resultado deseado en cuanto a funcionalidad del sistema en general.

5.10. MONTAJE FÍSICO DE PRUEBAS DEL SISTEMA DE MONITORIZACIÓN Y AUTOMATIZACIÓN

A continuación, en la figura 137 se puede apreciar el montaje realizado para efectuar las pruebas de funcionamiento deseado para el sistema de control y monitoreo de las luminarias eléctricamente por medio de los interruptores clásicos de pared y electrónicamente por medio de la tarjeta Arduino.

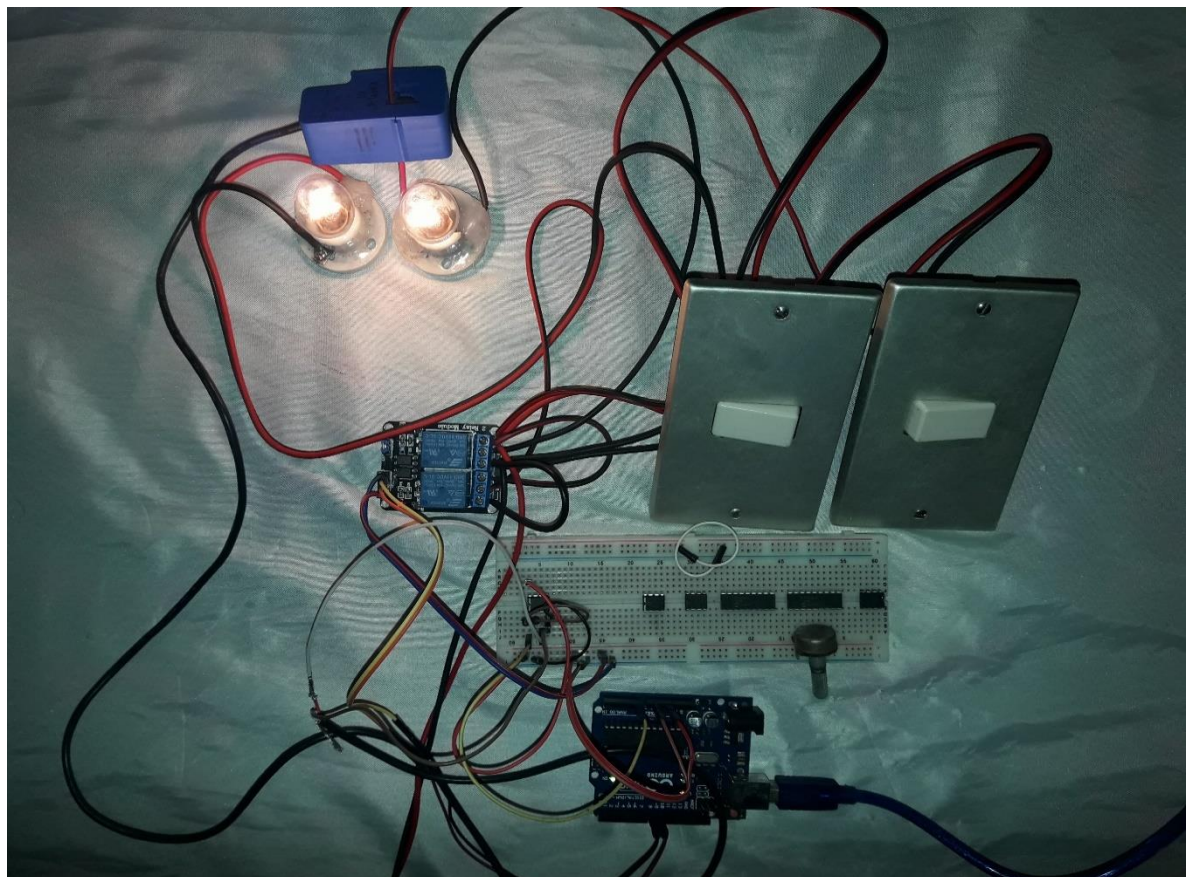


Figura 137: Montaje realizado para pruebas de funcionamiento

En la figura 139 se puede apreciar la integración y conexiones realizadas de los diferentes componentes del sistema. Dentro de los elementos están:

- ❖ **Tarjeta Arduino Mega:** Cerebro del sistema.
- ❖ **Escudo o Shield GSM/GPRS:** Tarjeta electrónica que dota al sistema de la comunicación, por un lado, con la base de datos en el servidor de Internet por medio de GPRS y, por otro lado, con la App desarrollada por medio de SMS y CALL a través de la red GSM.
- ❖ **Sensores:** Responsables de captar y proporcionar la información de las variables del sistema. Variables de detección de estado de luminarias y variables ambientales: Temperatura y humedad relativa.
- ❖ **Actuadores:** Módulo relés y cargas (luminarias).
- ❖ **Módulo relés:** Tarjeta electrónica para el control ON y OFF de las luminarias electrónicamente por medio de pulsos TTL desde el controlador Arduino.
- ❖ **LEDs:** Indicadores de estado de comunicación y luminarias.
- ❖ **Pantalla LCD 20X4:** Pantalla de cristal líquido donde se muestran mensajes acerca de las variables del sistema o información captada por los sensores.
- ❖ **Escudo o Shield Ethernet:** Tarjeta electrónica utilizada para pruebas de comunicación por medios guiados o Ethernet para el envío de la información de los sensores a la base de datos en el servidor de Internet.
- ❖ **Power Adapter 5 Vdc/2A:** Fuente de alimentación del sistema electrónico.
- ❖ **Cable cat 5e o superior Ethernet:** Cable de red utilizado para conectar escudo Ethernet con el dispositivo Router.

5.11. ESQUEMA ELECTRÓNICO DEL SISTEMA

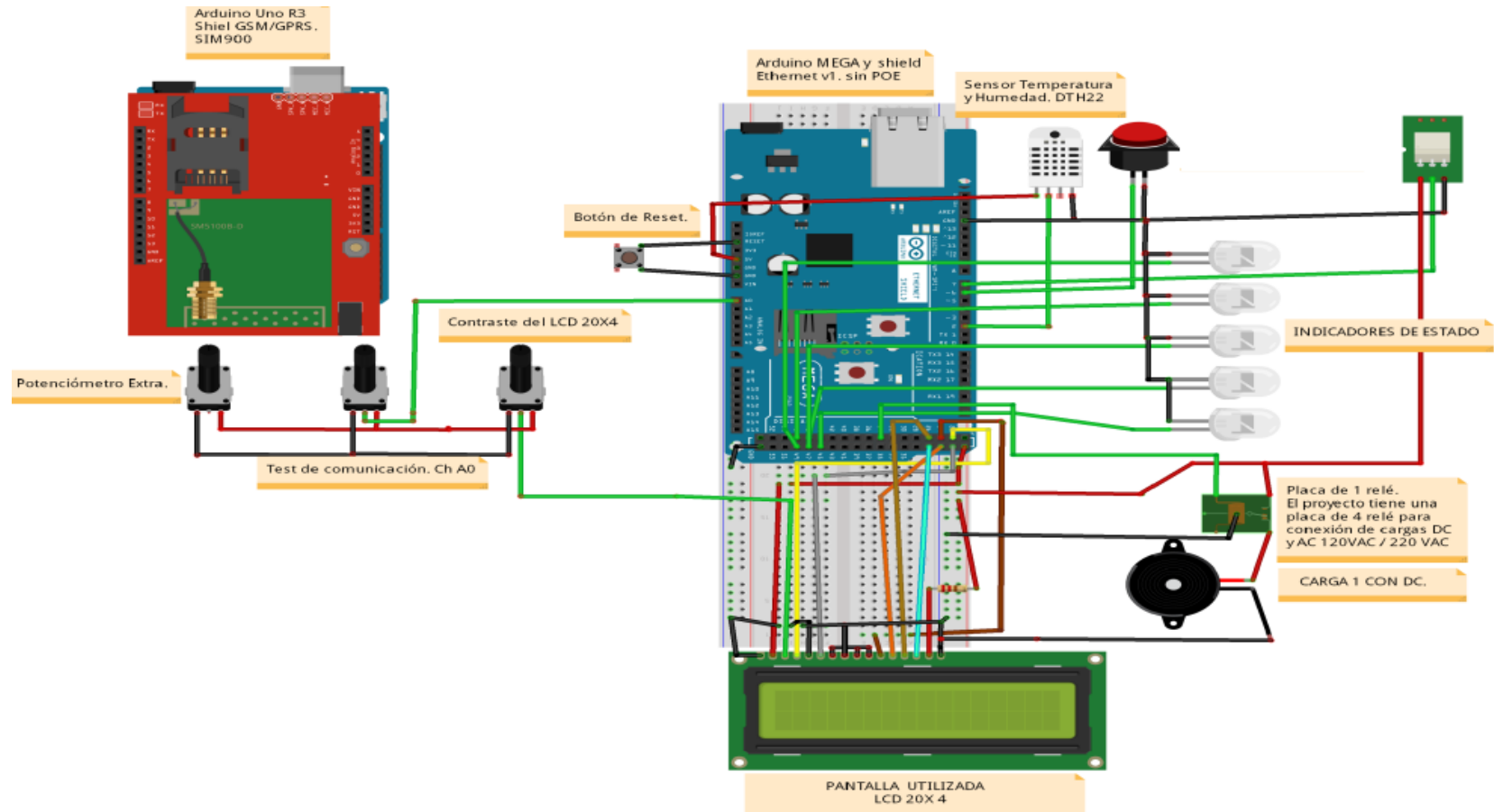


Figura 138: Diseño electrónico del sistema integrado

5.12. RESUMÉN DEL CAPÍTULO

A lo largo de este capítulo se han mostrado los diferentes componentes electrónicos y esquemas de conexión eléctricos y electrónicos utilizados para las pruebas, tanto del Hardware como del Software necesarios para el monitoreo y automatización. Así, se han detallado las entradas y salidas del sistema empleando los sensores y actuadores que fueron comentados en el capítulo I.

Por otro lado, se han realizados montajes de los circuitos empleando Hardware similar al que se empleará en el proyecto, con la finalidad de poder realizar simulaciones en el Software Proteus más aproximadas a la realidad.

Finalmente, se menciona que la parte de implementación práctica y simulaciones del sistema expuesto en este capítulo confirman el correcto funcionamiento de lo que se planteaba de forma teórica. En consecuencia, se puede afirmar que el sistema cumple con las perspectivas esperadas y se logra el objetivo planteado.

CAPÍTULO VI

COMUNICACIÓN DEL SISTEMA

6. COMUNICACIÓN DEL SISTEMA

6.1. BREVE INTRODUCCIÓN

En este capítulo se desarrolla el sistema de comunicaciones que finalmente se empleó de entre las posibles opciones estudiadas y mencionadas en el capítulo I, del marco teórico. En los capítulos anteriores se ha descrito el proceso empleado para el desarrollo del sistema de monitorización y automatización, por consiguiente, es dotar al sistema de las comunicaciones que permitan conocer los valores medidos y poder interactuar de forma local y remota. Para ello, se empleó la tarjeta electrónica GSM/GPRS SIM900, la cual, dota al sistema con la capacidad para hacer uso de la red GSM y GPRS para establecer comunicación vía SMS/CALL y con el mundo exterior (Internet) respectivamente. Se utilizó la red GPRS, a través de la tarjeta SIM insertada al escudo SIM900, de la cual, se tuvo que investigar los datos necesarios de la operadora telefónica (APN, usuario y contraseña) movistar el Salvador, necesarios para establecer conexión con la base de datos del servidor en Internet utilizando el protocolo TCP (Protocolo de Control de Transferencia), de forma que el usuario puede interactuar con el sistema mediante su terminal de telefonía móvil. Por tanto, en los siguientes puntos se presenta todo el desarrollo de esta parte del proyecto, incluyendo los argumentos necesarios y suficientes, sin entrar en una profundidad excesiva que no aporte utilidad al trabajo.

6.2. SHIELD O ESCUDO SIM900 GSM/GPRS

A partir de este punto se procede a presentar la tarjeta electrónica o Shield SIM900 GSM/GPRS para trabajar con el controlador Arduino.

En la figura 139 se puede apreciar la tarjeta electrónica (escudo SIM900) que será encargada de permitir al sistema las comunicaciones necesarias con el usuario a través de la red GSM y GPRS de manera local y remota.



Figura 139: SHIELD SIM900 GSM/GPRS

Se trata de una tarjeta o módulo de expansión que es totalmente compatible con cualquier microcontrolador que soporte el protocolo de comunicación serial Rx y Tx (Recepción y Transmisión respectivamente), debido a que este escudo maneja la comunicación de esta manera y viendo su flexibilidad de adaptación con casi cualquier controlador se decidió integrarlo con Arduino. Teniendo la ventaja de trabajarlo con Arduino, de poder apilar dicho escudo sobre Arduino debido a que los conectores que posee están alineados con esta finalidad. En la figura 140. Se describen los elementos principales de la tarjeta.

Este Shield o escudo tiene la capacidad de hacer las funciones básicas y necesarias de un dispositivo móvil cualquiera, dentro de las cuales se puede mencionar: SMS entrantes y salientes, hacer y recibir llamadas y por su puesto incorpora dentro de sus características la tecnología GPRS, tecnología que ha sido la implementada para el envío de la información que es captada y alimentada al sistema por medio de los sensores al controlador Arduino y finalmente se envía a la base de datos en el servidor de la nube o Internet, por medio de la red GSM/GPRS.

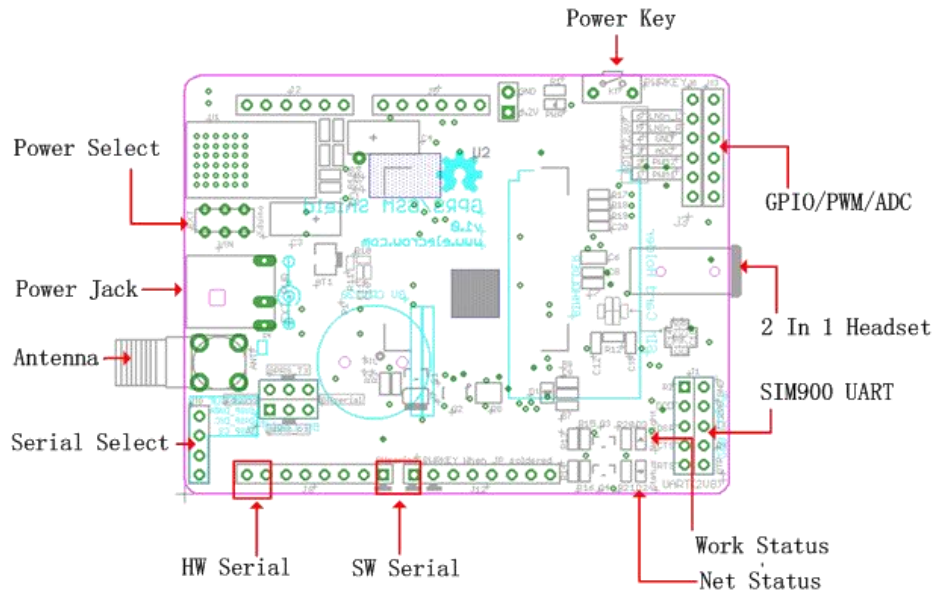


Figura 140: Descripción física de la tarjeta SIM900 GSM/GPRS

6.3. CARACTERÍSTICAS DE HARDWARE Y SOFTWARE DE LA TARJETA

Los elementos principales que integra la tarjeta son:

HARDWARE:

- ✚ Chip SIM900: Referencia datasheet del chip. Aquí se podrán encontrar todas las características y funcionalidades.
- ✚ Antena: Tiene un zócalo para conectar una antena mediante un terminal para cable coaxial, directamente al chip.
- ✚ Zócalo tarjeta SIM: En la parte inferior de la tarjeta se encuentra el zócalo donde se debe insertar la tarjeta SIM desbloqueada. Además, en este mismo lado se encuentra el alojamiento para una pila de botón de 3 V, la cual, permite mantener la información cuando la tarjeta no tiene alimentación. Ver figura 141.

Conector de alimentación tipo Jack: Para alimentar la tarjeta es con 5 Vdc,

normalmente, la intensidad operativa es de 450 mA, pero se pueden dar momentos en los que se requiera hasta 2 A, normalmente ocurre cuando se envía un SMS o se realizan llamadas. Por lo anterior la fuente de alimentación debe poder suministrar este pico de corriente en el momento que se requiera, en caso que se ponga como fuente de alimentación una que no cumple con las demandas de corrientes en los momentos deseados, simplemente el Shield o escudo se reinicia y no realiza la tarea deseada.

UART: Comunicación serie asíncronas por Hardware mediante los pines D0 y D8 o mediante software por los pines D7 y D8. Esta característica se configura mediante unos jumpers que se incorporan en dicha tarjeta (Normalmente color amarillo).

-  Pines de Entada y Salida: Permite la conexión con Arduino u otro controlador que soporte comunicación serial (UART) y presenta características similares. Hay pines analógicos, digitales y PWM (Modulación por Ancho de Pulso).
-  Conectores para micrófono y auricular: Conectores de tipo minijack para emplearlos en llamadas telefónicas.
-  Leds indicadores: Indican el estado de la comunicación, estado de la señal de cobertura, estado de elementos de la tarjeta. Ver tabla 11.

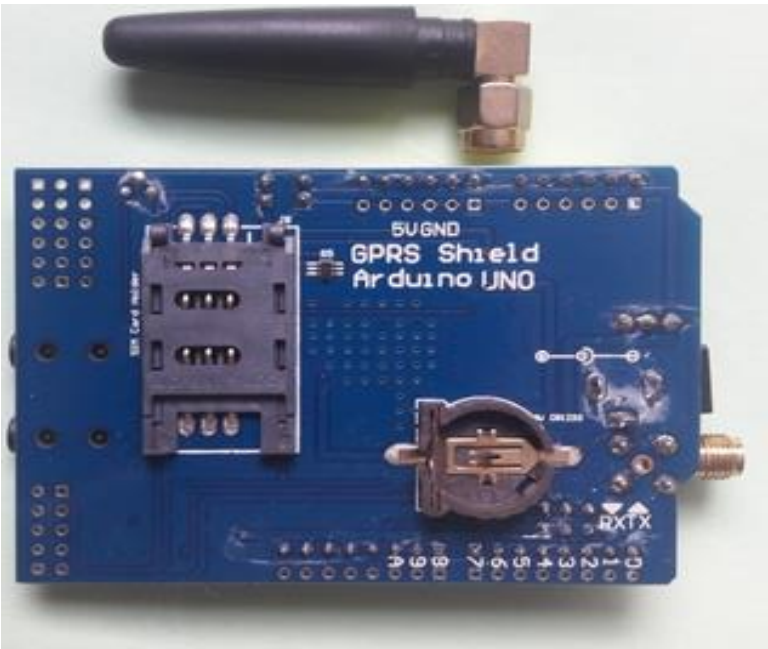


Figura 141: Parte trasera de la tarjeta SIM900 GSM/GPRS

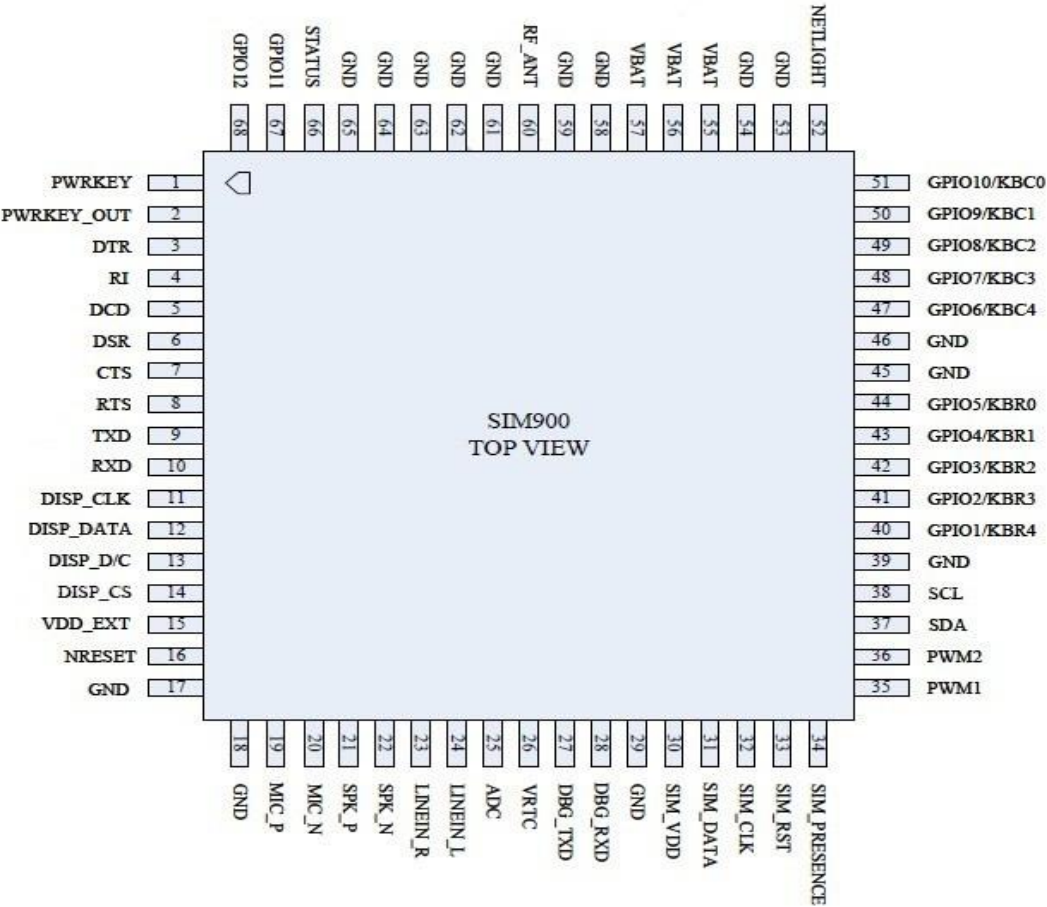


Figura 142: Pinout del chip SIM900 GSM/GPRS



Figura 143: Chip SIM900 GSM/GPRS

LED	Estado	Descripción
PWR	ON	Power of the GPRS Shield is on
	OFF	Power of the GPRS Shield is off
Status	ON	SIM900 is on
	OFF	SIM900 is off
Netlight	64 ms On/800 ms	SIM900 has not registered to a network.
	OFF	
	64 ms On/3000ms	SIM900 has registered to a network.
	OFF	
	64 ms On/3000ms	GPRS communication.
	OFF	
	OFF	SIM900 is not running.

Tabla 9. Descripción de los LED indicadores de la tarjeta GSM/GPRS SIM900

SOFTWARE:

Para hacer posible la programación de la tarjeta Shield SIM900, es obligatorio emplear comandos AT y pese a que la lista de comandos es bastante extensa, en este caso se emplearán los comandos para poder delegar a la tarjeta las funciones necesarias para alcanzar los objetivos propuestos al inicio del proyecto. Hay que aclarar que son los comandos AT y para ello, se puede decir que son los que se conocen como el conjunto de comandos Hayes (Es un lenguaje desarrollado por la compañía Hayes Communications, la cual prácticamente se convirtió en estándar abierto de comandos para configurar y parametrizar módems. Los caracteres AT, que preceden a todos los comandos, significa “ATENCIÓN”, e hicieron que se conociera también a este conjunto de comandos como comandos AT.

Por un lado, aunque la lista de comandos es muy extensa, los comandos principales que se emplearán son los siguientes:

AT: Este comando sirve para verificar si el módulo SIM900 está funcionando adecuadamente para entrar en modo comando. Al enviar AT el SIM el SIM deberá contestar con un OK.

AT+CGMI: Muestra el nombre del fabricante.

ATI: Muestra la información del producto.

AT+IPR=?: Muestra el Baud Rate en el que puede operar el SIM.

AT+IPR=XXXX: Configura la frecuencia deseada.

AT+COPS?: Muestra el nombre de la compañía telefónica.

AT+CGSN: Muestra el IMEI del chip utilizado.

AT+CSCS?: Tipo de texto.

AT+CSCS="XXX": Configura el tipo de texto.

AT+CMGF?: Muestra el formato de un mensaje, ya sea DPU(0) o SMS(1).

AT+CMGF=1; Modo texto para SMS.

AT+CMGS=XXXXXXXXX: Envía un SMS y se despliega el símbolo mayor que ‘>’

Se escribe el contenido del mensaje y al finalizar se presiona Ctrl + Z y retorna OK si el SMS se envió correctamente.

AT+CMGL=ALL: Sirve para ver todos los mensajes que han llegado al SIM.

ATDXXXXXXX: Sirve para realizar una llamada a cualquier teléfono móvil.

ATA: Sirve para contestar una llamada.

ATH: Sirve para colgar una llamada.

6.4. FORMAS DE EJECUCIÓN DE COMANDOS AT EN LA SHIELD SIM900

Se puede realizar la programación y verificación de la funcionalidad de la tarjeta SIM900 GSM/GPRS conectándola a un PC mediante un conversor FTDI, que tiene la función de convertir la señal del puerto USB a comunicación serial (RS232) o bien empleando el conversor de la tarjeta Arduino. Para el caso en que la conexión y prueba de comandos sea a través de una tarjeta electrónica FTDI, se puede recurrir a software emulador de puerto serie como, por ejemplo: Putty, TCP232, termite, Coolterm, HyperTerminal, Null-modem emulador, RealTerm, Terminal, etc. Para el envío de comandos AT y recepción del mensaje de respuesta por la Shield SIM900. En la figura 145 se puede apreciar la conexión del Shield GSM/GPRS SIM900 con un conversor FTDI.

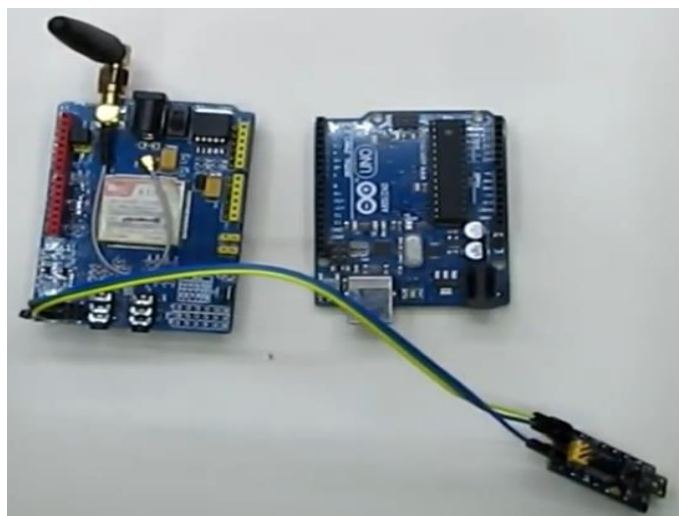


Figura 144: Conexión FTDI del escudo SIM900

Para el caso de utilizar el conversor FTDI que incorpora de fábrica la tarjeta Arduino las conexiones del escudo SIM900 con la tarjeta Arduino se pueden apreciar en figura 145.

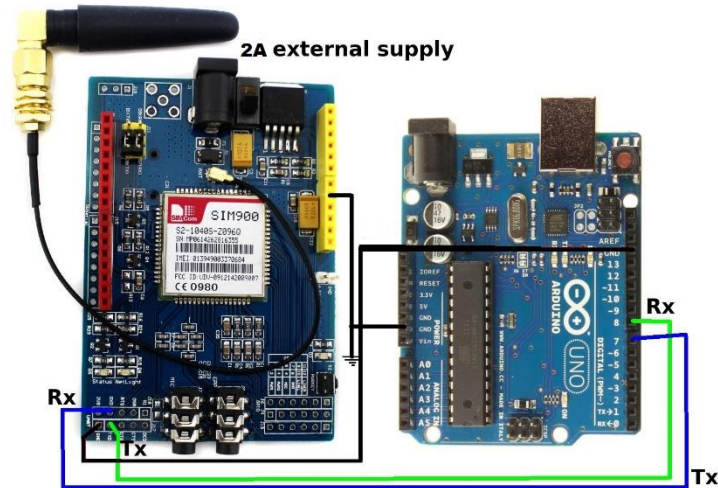


Figura 145: Conexión Arduino con escudo SIM900

En la figura 145: se puede apreciar como el escudo SIM900 se encuentra conectado con el módulo FTDI. Bastaría con conectar la tarjeta FTDI a través de un cable USB al ordenador para iniciar las pruebas con comandos AT desde un software emulador de puerto serie. En la figura 145 conexión de Arduino con SIM900 bastaría al igual que para el primer caso con conectar por medio de un cable USB a la tarjeta Arduino para iniciar las pruebas de funcionalidad a través de comandos AT, la diferencia de esta última forma de conexión del escudo SIM900 está en que en vez de utilizar un software emulador de puerto serie se utiliza el monitor serial del IDE de Arduino.

- ✚ La segunda opción está relacionada con la conexión que se realiza en la figura 145 conexión de Arduino con escudo SIM900 y esta consiste en la programación del sketch en el IDE de Arduino, donde los comando AT son interpretados directamente por el microcontrolador ATMEGA 328P-PU que incorpora la tarjeta Arduino y, mediante la comunicación serie se envían a la tarjeta SIM900 para que se ejecuten, es decir, que en el propio Sketch cargado

a Arduino se incluye los comandos AT, de forma que dependiendo de qué programa se esté ejecutando, se puede dar una orden específica o combinada a la tarjeta SIM900 encargada de la comunicación GSM/GPRS.

Por último, se hace mención a que todo lo anteriormente expuesto sobre el escudo SIM900, FTDI y Arduino será necesario emplearlo en el desarrollo del prototipo del sistema y en las simulaciones. Así, en los siguientes apartados se presenta el desarrollo de todo el proceso de diseño del prototipo.

6.5. PRUEBAS EXPERIMENTALES CON EL PROTOTIPO

En primer lugar, se procede comprobando la operatividad del escudo GSM/GPRS SIM900. Para ello, se realizan los siguientes pasos.

1. Se alimenta la tarjeta a través de un adaptador de 5VDC / 2A a través del conector Jack y se coloca el selector de alimentación en la posición de alimentación externa.
2. Se acopla la Antena correspondiente para el escudo SIM900.
3. Se inserta la tarjeta SIM de una operadora telefónica desbloqueada en su ranura.
4. Se seleccionan los pines D0 y D1 como puerto de comunicación serie, mediante el Jumper correspondiente.
5. Se ejecuta el Software emulador de puerto serie para el envío de comandos AT al escudo GSM/GPRS SIM900.

6.6. VERIFICACIÓN DE LA COMUNICACIÓN

Una vez realizado los pasos anteriores para la comprobación de la correcta operatividad del escudo SIM900, es necesario realizar el montaje básico de conexiones entre la tarjeta Arduino el escudo GSM/GPRS SIM90, ver figura 147.



Figura 146: Montaje para pruebas comandos AT.

Una vez realizadas las conexiones de la figura 146 ya es posible empezar a enviar comandos AT al módulo de comunicaciones y para ello es necesario ejecutar el IDE de Arduino y abrir la ventana de monitor serial de Arduino.

6.7. SKETCH BÁSICO PARA PRUEBAS DE ENVÍO DE COMANDOS AT POR TECLADO DESDE MONITOR SERIAL

```
sketch_oct22a
#include <SoftwareSerial.h>
SoftwareSerial SIM900(7, 8); //Selección de pines

void setup()
{
  SIM900.begin(19200);
  Serial.begin(19200);
}

void loop()
{
  //Enviamos y recibimos datos
  if (Serial.available() > 0)
    SIM900.write(Serial.read());
  if (SIM900.available() > 0)
    Serial.write(SIM900.read());
}
```

```
COM10 (Arduino/Genuino Uno)
Enviar
AT
OK
AT+CFUN="7126"
OK
+CREG: 2
+CREG: 1
RING
RING
```

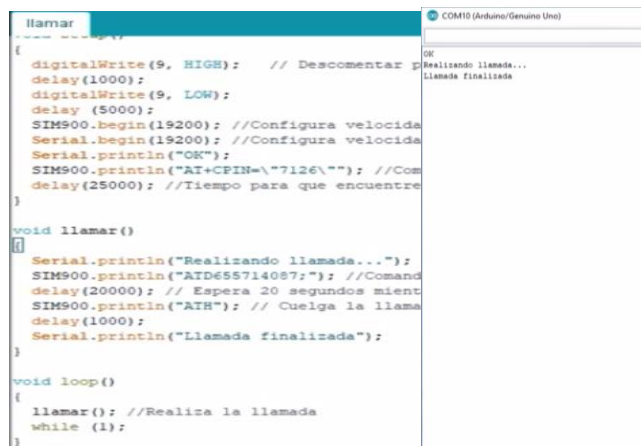
Figura 147: Sketch básico para pruebas de funcionalidad por comandos AT

Desde la ventana del monitor serial del IDE de Arduino

En la figura 147 se presenta un programa básico (Sketch) escrito en el IDE de Arduino que se utiliza únicamente para enviar comandos AT por teclado desde la ventana del monitor serial de Arduino. Además, se puede visualizar la respuesta que devuelve el módulo según el comando que se envié. La lista de comandos AT es enorme y en las pruebas realizadas solamente se utilizaron los comandos necesarios para cubrir la funcionalidad propuesta en este trabajo. Ver tabla 12 de comandos AT.

6.8. SKETCH BÁSICO PARA REALIZAR PRUEBA DE LLAMADA CON ARDUINO Y COMANDOS AT

Se presenta el Sketch básico escrito en el IDE de Arduino para realizar una llamada sin tener que escribir los comandos desde la ventana monitor serial.



```

llamar
//--- sketch ---
{
  digitalWrite(9, HIGH); // Desconectar p
  delay(1000);
  digitalWrite(9, LOW);
  delay (5000);
  SIM900.begin(19200); //Configura veloci
  Serial.begin(19200); //Configura veloci
  Serial.println("OK");
  SIM900.println("AT+CPIN=\"7126\""); //Com
  delay(25000); //Tiempo para que encuentre
}

void llamar()
{
  Serial.println("Realizando llamada...");
  SIM900.println("ATD655714087"); //Comand
  delay(20000); // Espera 20 segundos mient
  SIM900.println("ATH"); // Cuelga la llama
  delay(1000);
  Serial.println("Llamada finalizada");
}

void loop()
{
  llamar(); //Realiza la llamada
  while (1);
}
  
```

Serial Monitor (COM10 (Arduino/Genuino Uno))

```

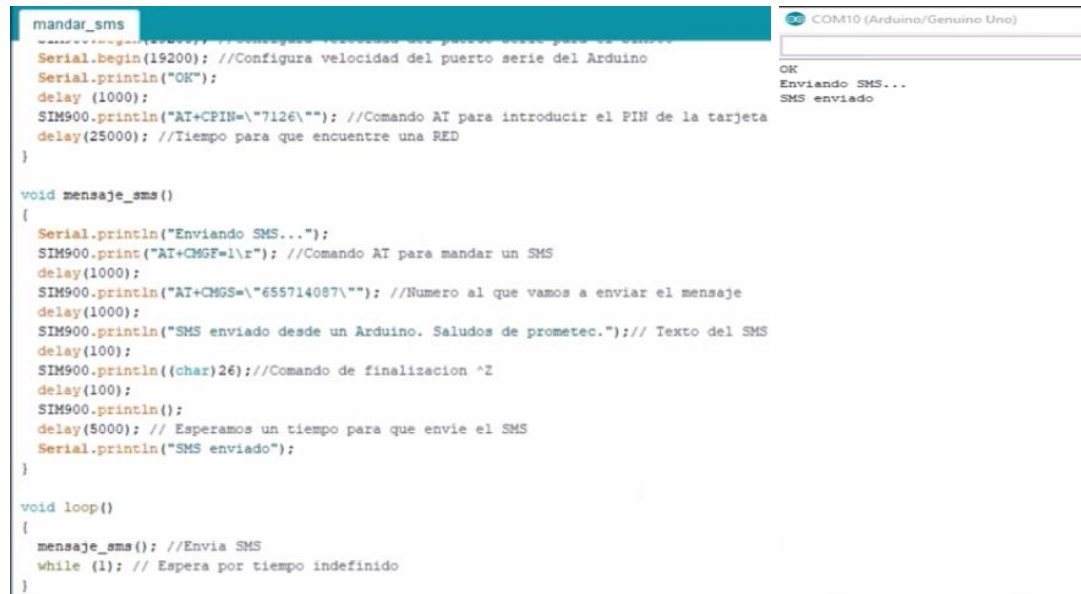
OK
Realizando llamada...
Llamada finalizada
  
```

Figura 148: Sketch básico para realizar llamada

En la figura 148 se presenta un sketch para que sea el microcontrolador ATMEGA el encargado de ejecutar directamente los comandos AT necesarios para ordenar a la tarjeta de comunicaciones que realice y cuelgue una llamada después de pasados unos segundos.

6.9. SKETCH BÁSICO PARA REALIZAR PRUEBA DE ENVIO DE SMS CON ARDUINO Y COMANDOS AT

Para ordenar a la tarjeta de comunicaciones SIM900 que envíe un SMS a un número de teléfono determinado es bastante parecido al de cómo realizar una llamada. En la figura 149 se puede apreciar el código fuente utilizado para enviar SMS.



```
mandar_sms
//Configura velocidad del puerto serie para el Arduino
Serial.begin(19200); //Configura velocidad del puerto serie del Arduino
Serial.println("OK");
delay(1000);
SIM900.println("AT+CPIN=\"7126\""); //Comando AT para introducir el PIN de la tarjeta
delay(25000); //Tiempo para que encuentre una RED
}

void mensaje_sms()
{
  Serial.println("Enviando SMS...");
  SIM900.print("AT+CMGF=1\r"); //Comando AT para mandar un SMS
  delay(1000);
  SIM900.println("AT+CMGS=\"655714087\""); //Numero al que vamos a enviar el mensaje
  delay(1000);
  SIM900.println("SMS enviado desde un Arduino. Saludos de prometec."); // Texto del SMS
  delay(100);
  SIM900.println((char)26); //Comando de finalizacion ^Z
  delay(100);
  SIM900.println();
  delay(5000); // Esperamos un tiempo para que envíe el SMS
  Serial.println("SMS enviado");
}

void loop()
{
  mensaje_sms(); //Envia SMS
  while (1); // Espera por tiempo indefinido
}
```

COM10 (Arduino/Genuino Uno)

OK
Enviando SMS...
SMS enviado

Figura 149: Sketch básico para enviar SMS

En la figura 149 se presenta un Sketch para que sea el microcontrolador ATMEGA de la tarjeta Arduino el encargado de ejecutar directamente los comandos AT necesarios para ordenar a la tarjeta de comunicaciones que realice un envío de SMS después de pasados unos segundos.

6.10. SOFTWARE DE COMUNICACIONES

Se presenta en este apartado el prototipo y Sketch que se diseñó para Arduino, el cual permite que el sistema se comunique con el servidor y la base de datos en la nube (Internet) utilizando la red GPRS de la compañía telefónica movistar el Salvador. Para ello se realizó una comprobación previa con un pequeño sketch cargado a la tarjeta Arduino al cual se le incorporan en su cuerpo todos los comandos necesarios para lograr dicha conexión. Ver tabla 13. De comando AT.

6.11. ESQUEMA DE FUNCIONAMIENTO DEL SISTEMA

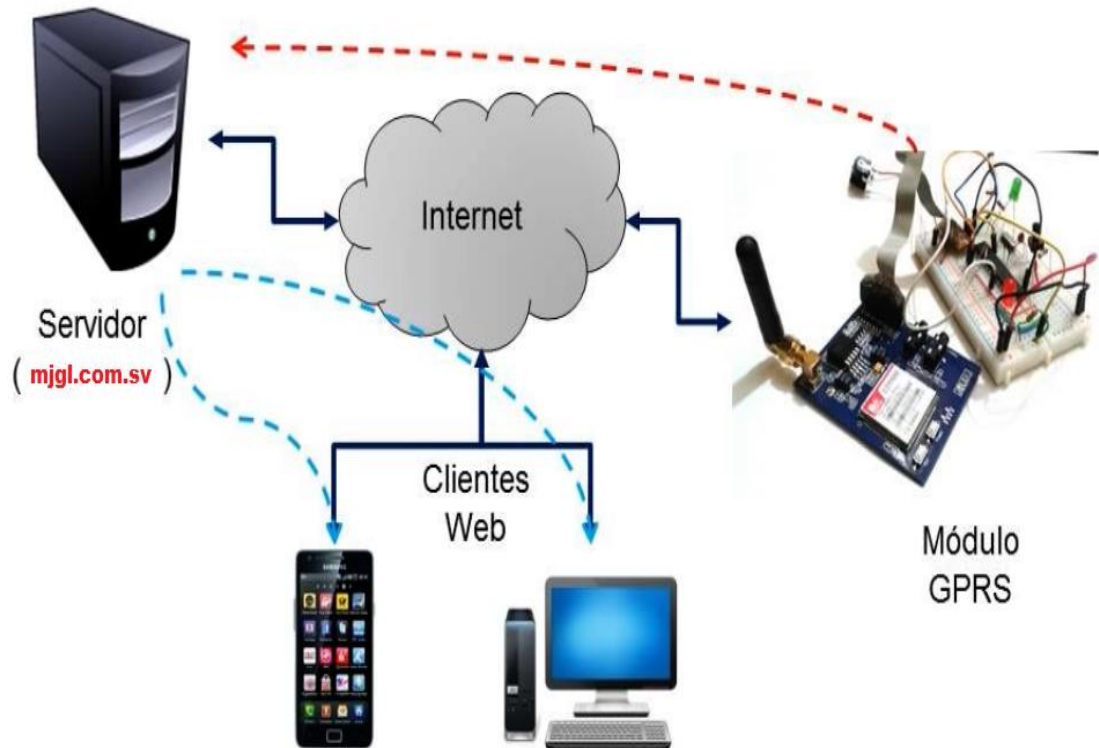


Figura 150: Comunicación mediante GPRS e Internet (Server – BD)

En la figura 150 se muestra los elementos que son parte directa e indirecta para la conexión con el servidor en Internet y se describen los principales partes a continuación:

1. **Módulo GRPS:** Este módulo es el encargado de la conexión con el servidor. Una vez lograda la conexión con el servidor se tuvo que pensar en la data o paquete de información a enviar a la base de datos en el servidor en Internet y esta fue tomada y generada por los sensores que son parte del sistema.
2. **Servidor:** Equipo apegado a un dominio de internet en el cual se corre el servidor web, servidor de aplicaciones y la base de datos. Para acceder a este se hace a través de una URL (nombre de dominio).

3. **Internet:** La red de redes que nos permite acceder a los servicios brindados por el servidor. Por medio de internet será posible la conexión desde diversos dispositivos inteligentes a la información almacenada en la base de datos del servidor.
4. **Cientes Web:** Los clientes web son todos los host o dispositivos que se apegarán al sistema para la consulta de información almacenada en la base de datos a través de un navegador web desde cualquier dispositivo inteligente. Para el caso particular de los dispositivos móviles Android se podrá acceder a dicha información a través de las opciones incorporadas a la App que se ha desarrollado como parte del resultado de esta investigación. Ver capítulo II.
5. **Web Service:** Servicio web activo en el servidor, el cual está atento a información que envía el sistema electrónico GPRS para procesar y enviar dicha información a la base de datos.
6. **Base de datos:** Entidad encargada de almacenar la información recibida del sistema.

6.12. **PROTOTIPO Y SKETCH**

Se presenta el prototipo que se diseñó para comprobar la correcta funcionalidad del sistema de comunicaciones GPRS. En la figura 153 se puede apreciar el montaje, en el cual se han añadido otros elementos que son necesarios para la monitorización de los parámetros y el control ON/OFF de las luminarias. Así mismo, se han incorporado una serie de LEDs como indicadores de activación de cargas y de estado de los procesos a realizar por el sistema. Además, se presenta un Sketch básico que permite dar las órdenes adecuadas al escudo de comunicaciones GPRS para tener comunicación con el exterior desde el sistema electrónico diseñado como parte de este trabajo de investigación.

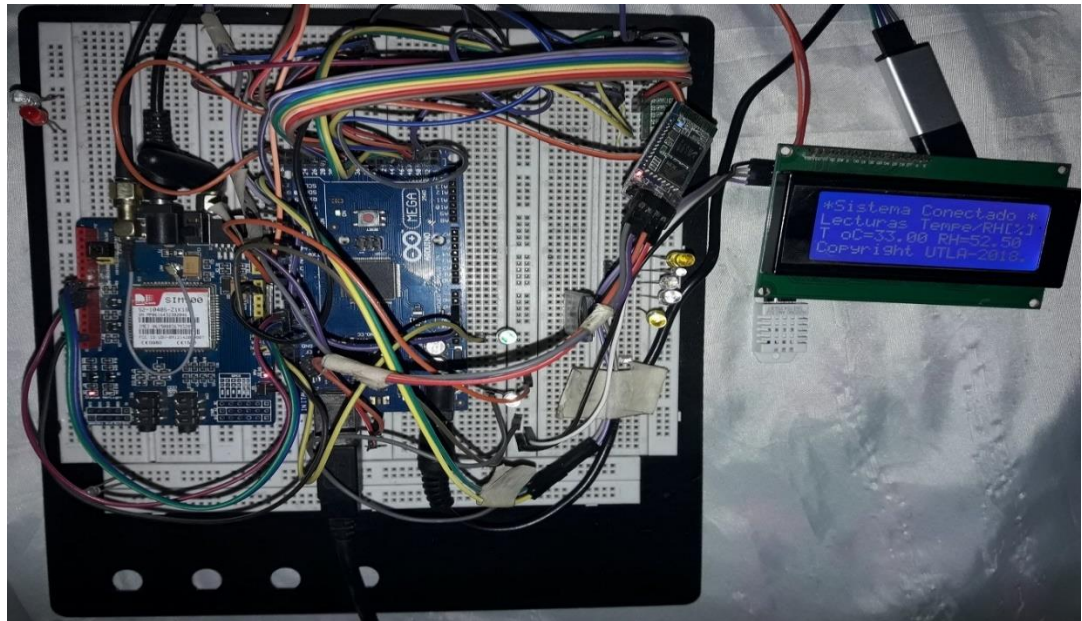
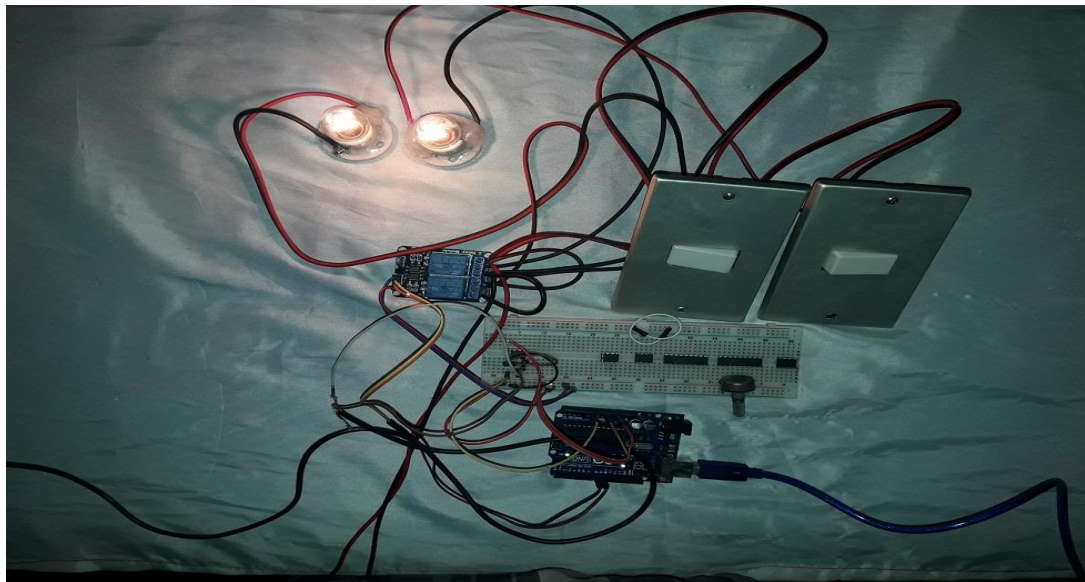


Figura 151: Prototipo físico del sistema y comunicaciones



*Figura 152: Prototipo para control de luminarias por interruptor y electrónicamente con
Arduino*

Se describen las conexiones principales que se han realizado en el prototipo.

- ✚ Pines 10 y 11 se hace la conexión entre Arduino y la Shield SIM900 GSM/GPRS. Por estos terminales se establece la comunicación de datos y comandos AT para dar órdenes al sistema.
- ✚ Conexión de pin 9 con pin 9 de Arduino sirve para iniciar la Shield SIM900 por software.
- ✚ El pin GND de Shield SIM900 y el pin GND de Arduino.
- ✚ El pin 2 de Arduino con el pin data del sensor DHT22.
- ✚ Pines 46-53 de Arduino son los LED para visualizar estado de cargas (Luminarias).
- ✚ Pines A0-A5 de Arduino son las entradas de los sensores para detección del estado de las luminarias.
- ✚ Pines 20 SDA, 21 SCL de Arduino está la conexión de la pantalla LCD 20X4.
- ✚ Conexiones de cables de alimentación (5vdc y GND).
- ✚ La alimentación de la Shield SIM900 se realiza mediante un adaptador que suministra una intensidad de corriente máxima de 2 A 5VDC.
- ✚ La alimentación de Arduino y comunicación serie con el PC se realiza mediante el puerto USB tipo B que incorpora la tarjeta Arduino, el cual corresponde al puerto serie por Hardware (Pines 0 y 1 de Arduino)

6.13. SKETCH BÁSICO PARA REALIZAR PRUEBA DE CONEXIÓN A INTERNET POR GPRS A TRAVÉS DE ARDUINO Y COMANDOS AT

En la figura 154 se presenta un Sketch para que sea el microcontrolador ATMEGA de la tarjeta Arduino el encargado de ejecutar directamente los comandos AT necesarios para ordenar a la tarjeta de comunicaciones que se realice la conexión con la base de datos en el servidor de Internet. El encargado de recibir y procesar esta información antes de llegar a la base de datos en el servidor es el Web Service almacenado en el servidor, el cual está escrito en el lenguaje de programación PHP (Pre-procesador de HyperTexto). Este mismo es el encargo de controlar la comunicación entre la App Android y el servidor en internet.

```

GPRS_Internet1 Arduino 1.8.1
Archivo Editar Programa Herramientas Ayuda

GPRS_Internet1 $
1 // Programa para controlar el módulo SIM900 a través de un
2 // módulo GSM/GPRS
3 #include <SoftwareSerial.h>
4 SoftwareSerial SIM900(10, 11); // Configura el puerto serial para el SIM900. Para el Arduino MEGA u
5
6 int respuesta;
7 char aux_str[50];
8
9 char direccion[] = "GET /hola.php HTTP/1.1\r\nHost: servidor.com.sv\r\nConnection: close\r\n\r\n \x
10
11 void setup()
12 {
13     SIM900.begin(19200); //Configura velocidad del puerto serie para el SIM900
14     Serial.begin(19200); //Configura velocidad del puerto serie del Arduino
15     delay(1000);
16     Serial.println("Iniciando...");
17
18     quitGPRS();
19
20     power_on();
21     iniciar();
22 }
23
24 void loop()
25 {
26     if (Serial.available())
27         switch (Serial.read())
28         {
29             case 'h':
30                 PeticionHttp();
31                 break;
32             case 'q':
33                 quitGPRS();
34                 break;
35         }
36
37     if (SIM900.available()) {
38         Serial.write(SIM900.read());
39     }
40 }
41
42 void quitGPRS() {
43     // Closes the socket
44     enviarAT("AT+CIPCLOSE", "CLOSE OK", 10000); //Cerramos la conexion
45     enviarAT("AT+CIPSHUT", "OK", 10000); //Cierra el contexto PDP del GPRS
46     delay(10000);
47 }
48
49 void power_on()
50 {
51     int respuesta = 0;
52
53     // Comprueba que el modulo SIM900 esta arrancado
54     if (enviarAT("AT", "OK", 2000) == 0)
55     {
56         Serial.println("Encendiendo el GPRS...");
57
58         pinMode(9, OUTPUT);
59         digitalWrite(9, HIGH);
60         delay(1000);
61         digitalWrite(9, LOW);
62         delay(1000);
63
64         // Espera la respuesta del modulo SIM900

```

```

98 while (respuesta == 0) {
99     // Envía un comando AT cada 2 segundos y espera la respuesta
100     respuesta = enviarAT("AT", "OK", 2000);
101     SIM900.println(respuesta);
102 }
103 }
104 }
105
106 void power_off()
107 {
108     digitalWrite(9, HIGH);
109     delay(1000);
110     digitalWrite(9, LOW);
111     delay(1000);
112 }

122 void iniciar()
123 {
124     enviarAT("AT+CPIN=\"1234\"", "OK", 1000); //Introducimos el PIN de la SIM
125     Serial.println("Conectando a la red...");
126     delay (5000);
127
128     //Espera hasta estar conectado a la red movil
129     while ( enviarAT("AT+CREG?", "+CREG: 0,1", 1000) == 0 )
130     {
131     }
132
133     Serial.println("Conectado a la red.");
134     enviarAT("AT+CGATT=1", "OK", 1000); //Iniciamos la conexión GPRS
135     enviarAT("AT+CSTT=\"internet.movistar.sv\", \"xxxxx\", \"xxxxx\", \"OK\", 3000); //Definimos el APN, usu
136     enviarAT("AT+CIICR", "OK", 3000); //Activamos el perfil de datos inalámbrico
137     enviarAT("AT+CIFSR", "", 3000); //Activamos el perfil de datos inalámbrico

139     //enviarAT("AT+CLIP=1", "OK", 1000); // Activamos la identificacion de llamadas
140     //enviarAT("AT+CMGF=1", "OK", 1000); //Configura el modo texto para enviar o recibir mensajes
141     //enviarAT("AT+CNMI=2,2,0,0,0", "OK", 1000); //Configuramos el modulo para que nos muestre los SMS recibi
142     //Serial.println("Listo!!!");
143 }

145 void PeticionHttp()
146 {
149     enviarAT("AT+CIPSTART=\"TCP\", \"servidor.com.sv\", \"80\", \"CONNECT OK\", 5000); //Inicia una conexión TCP
150     // Enviamos datos a través del TCP
151     sprintf(aux_str, "AT+CIPSEND=%d", strlen(direccion));
152     if (enviarAT(aux_str, ">", 10000) == 1)
153     {
154         enviarAT(direccion, "OK", 10000);
155     }
156 }
157 else
158 {
159     reiniciar();
160     iniciar();
161 }
162 }
163

```

Figura 153: Sketch básico para enviar datos al servidor en Internet por GPRS.

Los Sketch (código fuente Arduino) que se presentan en este capítulo no son los oficiales o completos, debido a que el código que se genera como resultado de la integración de todas las etapas o módulos de funcionalidad del sistema, hace que se genere un Sketch bastante grande. Por tanto, este se incorporará ya sea en los anexos de este documento o como fichero en el CD/DVD que se entregue de esta investigación. Ver capítulo II para mayor detalle respecto a la funcionalidad del sistema.

6.14. RESUMÉN DEL CAPÍTULO

A lo largo de este capítulo se ha desarrollado y explicado el proceso necesario para que el sistema puede comunicarse con el exterior, permitiendo al usuario interactuar de forma remota mediante su terminal de telefonía móvil y la red GSM / GPRS.

En primer lugar, se ha realizado una pequeña introducción a la tecnología GSM/GPRS, detallando algunos aspectos técnicos, con la finalidad de contextualizar el entorno en el que se realizarán las comunicaciones. Dado que no se requiere una tasa de transferencia de información elevada, se ha considerado que es más que suficiente esta tecnología frente a la posibilidad de haber empleado otras alternativas antes mencionadas en el capítulo I.

En segundo lugar, se ha realizado una descripción básica del módulo de comunicaciones que se empleará en este proyecto, tanto a nivel de Hardware como de Software que se requiere para su control. Así, se ha explicado cómo es físicamente la tarjeta y su pin out para conocer cómo se puede conectar a Arduino. También, se ha explicado que son los comandos AT y se han listado algunos de los que se requieren para poder interactuar con el módulo de comunicaciones. Se recomienda leer el manual de comandos AT del MODEM que se posee.

En tercer lugar, se ha presentado el montaje del prototipo completo del sistema y las partes principales del Sketch que permite que el sistema sea operativo y realice las funciones básicas que se persiguen en el proyecto. Para ello, ha sido necesario entender y después demostrar cómo se puede interactuar con el módulo de comunicaciones mediante los comandos AT. Por un lado, se ha demostrado como se puede enviar y recibir SMS y llamadas conectando el módulo de comunicaciones directamente a la PC. Esto ha permitido conocer que sucede cuando se envía un comando AT (Que información devuelve la tarjeta de comunicaciones). Por otro lado, se ha realizado un primer Sketch para verificar que una vez cargado en Arduino y

alimentando al módulo de comunicaciones GSM/GPRS, el sistema por si solo es capaz de enviar y recibir SMS. Así, se ha conseguido integrar dicho Sketch con el de la parte de monitoreo y automatización para lograr un Sketch definitivo que permite que el sistema cumpla o realice las funcionalidades deseadas.

Finalmente, luego de mucha dedicación y bastante esfuerzo de investigación y muchas pruebas, se ha logrado obtener un sistema de comunicación funcional dejándonos como riqueza el conocimiento adquirido tras la realización de diferentes pruebas y la satisfacción de haber logrado alcanzar los objetivos planteados desde un inicio.

CONCLUSIONES

En este trabajo se presentan los datos teóricos y prácticos, así como los esquemas y demás diseños involucrados en el desarrollo de un dispositivo para un sistema de control y monitoreo de variables físicas a través de la instrumentación electrónica adecuada y un controlador Arduino, más un escudo electrónico de comunicación para redes GSM y GPRS. Además, del monitoreo de variables físicas ambientales como la temperatura y humedad relativa se monitoreo el estado de las luminarias. Las cuales pueden ser administradas de manera local y remota, utilizando un dispositivo móvil Smartphone y realizando la interconexión por medio de las redes Bluetooth y de telefonía celular GSM/GPRS.

Los dispositivos móviles se han convertido en herramientas indispensables para la economía mundial del último siglo. Una de sus recientes facetas es su potencial como herramientas de apoyo en la monitorización y control, debido a su capacidad de procesamiento, incluso en dispositivos de bajo costo, y a su fácil operación al alcance de cualquier usuario. En este proyecto se desarrolló una aplicación móvil Android para el control y monitoreo de variables físicas, a partir del controlador Arduino y la tarjeta de comunicaciones GSM/GPRS SIM900, haciendo uso de los sensores correspondientes para tal caso.

La aplicación fue probada con un prototipo montado e implementado en el Laboratorio de Electrónica de la UTLA, procesando registros de las variables controladas y monitoreadas por el sistema en la base de datos en el servidor online.

Es posible la creación de sistemas de telemetría con Software y Hardware libre abaratando costos en comparación a otros sistemas existentes y además se añade valor agregado haciendo una integración con otras tecnologías (Aplicación móvil Android) para la visualización de la información obtenida del ambiente por los diferentes sensores y el control de variables TODO/NADA del sistema.

RECOMENDACIONES

Dentro del proceso de investigación y pruebas efectuadas se identificó una diversidad de situaciones que obstaculizaban alcanzar los objetivos planteados. Con fines de que sirva a investigaciones futuras con propósitos similares al de este trabajo se listan los principales inconvenientes presentados y la forma de solventar dicha situación.

- ❖ Para el escudo o Shield GSM/GPRS SIM900, fue necesario utilizar una Antena diferente a la que se incluye al adquirir dicha tarjeta, debido a que esta Antena solo funciona muy bien estando en exteriores, pero en interiores presenta demasiados fallos de conexión a la red.
- ❖ Para el escudo o Shield Ethernet, fue necesario adicionar un disipador térmico al chip principal encargado de la comunicación (W5100), debido a que se detectó en las pruebas efectuadas que después de estar transmitiendo datos por más 24 horas el chip llega a excesos de temperatura provocando con ello interrumpir el envío de la información al servidor online y así también prevenir el daño del escudo completo.
- ❖ Se considera ideal que al Shield Ethernet se le incorpore un sistema de enfriamiento para mayor fiabilidad y seguridad del funcionamiento. El sistema que se recomienda y se considera el más adecuado, es el sistema de enfriamiento que incorpora una computadora en su CPU, el cual consiste en: Microprocesador, pasta térmica, disipador térmico, Cooler o fan y finalmente otros Cooler en los laterales del chasis.
- ❖ Al escudo o Shield Ethernet es necesario asegurarse de asignarle una dirección MAC única para su identificación en la red, pero principalmente para que a través de este identificador único se puede realizar una asignación de dirección IP de manera automática por un servidor DHCP (Sistema operativa de red o Router) como una reserva, para lograr que el dispositivo encargado de la comunicación por Ethernet siempre posea la misma dirección IP.
- ❖ Dar permisos a la dirección IP asignada al dispositivo electrónico en el Router para que puede ser accesible desde cualquier lugar (Red interna o externa).

BIBLIOGRAFÍA

Bernardo, A. (2013). E-Health: Tecnología móvil que revoluciona la medicina. Recuperado en Marzo de 2015 de: <http://blogthinkbig.com/ehealthtecnologia-movil-medicina/>

Tomás Gironés, J., (2015), Android: Introducción a la programación. Recuperado en marzo de 2015 de: <http://cursointroduccionandroid.upvx.es/course>
[19] Dispositivos móviles (2012). Recuperado en marzo de 2015 de: <http://dispositivosmobilesits.blogspot.com.es/2012/02/evolucion-de-moviles.html>

Poder Pda (2014). Recuperado en marzo de 2015 de: <http://www.poderpda.com/plataformas/android/android-supera-los-mil-millonesde-smartphones-en-2014/>

Ounae (2014). Recuperado en marzo de 2015 de: <http://ounae.com/ventajas-inconvenientes-android-google/>

Blog de WordPress, Sistema Android (2015). Recuperado en marzo de 2015 de: <https://scoello12.wordpress.com/ventajas-y-desventajas/>

Gigatecno (2015). Recuperado en marzo de 2015 de: <http://gigatecno.blogspot.com.es/2014/05/ventajas-y-desventajas-de-android.html>

Php (2014), ¿Que es PHP? Recuperado en marzo de 2015 de: <http://php.net/manual/es/intro-what-is.php>

Álvarez M.A (2001, 09 de mayo). Qué es PHP. Recuperado en marzo de 2015 de: <http://www.desarrolloweb.com/articulos/392.php>

Van Der Henst S C. (2001, 23de mayo). ¿Qué es PHP? Recuperado en Marzo de 2015 de: <http://www.maestrosdelweb.com/phpintro/>

Welling L. & Thomson L. (2003). Desarrollo web con PHP y MySQL. Madrid: Anaya

Rivero-Cornelio E., Martínez-Fuentes L. & Alonso-Martínez I. (2005). Bases de datos relacionales: fundamentos y diseño lógico. Madrid: Universidad Pontificia Comillas.

Martín-Escofet C. El lenguaje SQL. Recuperado en marzo de 2015 de: http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-dedatos/bases-de-datos/P06_M2109_02149.pdf

Ramakrishnan R. & Gehrke J. (2007). Sistemas de gestión de bases de

datos (3ª Ed.). Madrid: McGraw-Hill.

MySQL (2014), WhyMySQL? Recuperado en marzo de 2015 de:
<http://php.net/manual/es/intro-what-is.php>

LukeWelling y Laura Thomson, Desarrollo web con PHP y MySQL,
Anaya Multimedia, 2003.

Paul Dubois, MySQL, Prentice Hall, 2001

Hugh E. Williams y David Lane, Web Database Applications with
PHP and MySQL, O'Reilly, 2002

AndroidDevMx (2011). Recuperado en marzo de 2015 de:
<http://www.androiddevmx.net/java-y-la-maquina-virtual-dalvik>

TodoAndroid (2015). Recuperado en marzo de 2015 de:
<http://www.todoandroid.es/index.php/faq-de-android/65-versiones/1698-androidstudio-o-eclipse-opinion-de-un-desarrollador-de-aplicaciones.html>

Catalan, M. (2014). ¿Es el momento de saltar a Android Studio?
Recuperado en Marzo de 2015 de: <https://geekytheory.com/es-el-momento-desaltar-a-android-studio/>

Datasheet Chip ethernet IPv6:

-  <http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en022889>
-  <http://ww1.microchip.com/downloads/en/DeviceDoc/39662c.pdf>

Más información sobre IPv6:

-  <https://sites.google.com/site/ghoelzl/ipv6>
-  https://sites.google.com/site/ghoelzl/ipv6ethershield/ipv6_http_server
-  <http://www.tweaking4all.com/hardware/arduino/arduino-enc28j60-ethernet/>
-  http://www.fut-electronics.com/wp-content/plugins/fe_downloads/Uploads/Ethernet-Module-ENC28J60-Arduino.pdf

Librerías:

-  https://github.com/ntruchsess/arduino_uip
-  <https://github.com/jcw/ethercard>
-  <https://github.com/ghoelzl/IPv6EtherShield>
-  <https://sites.google.com/site/ghoelzl/ipv6ethershield>

KEN SCHWABER Y JEFF SUTHERLAND, La Guía Definitiva de Scrum: Las Reglas del Juego, Scrum.Org and Scrumming, 2014. [7] ALBERT GARRIGA
Metodología de gestión de proyectos

<http://www.rekursosenprojectmanagement.com/metodologia-de-gestion-de-proyectos/>

[https://es.wikipedia.org/wiki/Anexo:Historial de versiones de Android](https://es.wikipedia.org/wiki/Anexo:Historial_de_versiones_de_Android)

GLOSARIO

AUTOMATIZACIÓN: (automatización; del griego antiguo auto: guiado por uno mismo) es el uso de sistemas o elementos computarizados y electromecánicos para controlar maquinarias y/o procesos industriales sustituyendo a operadores humanos.

GSM: (Global System for Mobiles) Estándar paneuropeo para la constitución de redes telefónicas móviles celulares, creado por la CEPT y que utiliza el estándar ETSI en la banda 900 MHz

RTU: Proviene de las siglas "Supervisory Control And Data Acquisition". Es un sistema basado en computadores que permite supervisar y controlar variables de proceso a distancia, proporcionando comunicación con los dispositivos de campo (controladores autónomos) y controlando el proceso de forma automática por medio de un software especializado. También provee de toda la información que se genera en el proceso productivo a diversos usuarios, tanto del mismo nivel como de otros usuarios supervisores dentro de la empresa (supervisión, control calidad, control de producción, almacenamiento de datos, etc.).

GPRS: General Packet Radio Service. Tecnología de transmisión de voz y datos en terminales móviles. Algo más avanzado que el GSM y que funciona mediante la conmutación de paquetes entre el terminal (móvil) y la antena.

RF: El término radiofrecuencia, también denominado espectro de radiofrecuencia o RF, se aplica a la porción menos energética del espectro electromagnético, situada entre unos 3 Hz y unos 300 GHz.

ACTUADOR: Se denominan actuadores aquellos elementos que pueden provocar un efecto sobre un proceso automatizado, el actuador recibe la orden de un controlador y da una salida necesaria para activar a un elemento final de control como lo son motores, lámparas o válvulas.

CONTROL: Es tener bajo supervisión una variable de un determinado proceso, la cual al final arroja un resultado, dicho resultado llega a un controlador y compara el resultado obtenido al final del proceso con un valor predeterminado y así lograr corregir el margen de error al final del proceso.

CPU: Unidad central de procesamiento (Central Processing Unit).

EMISOR: Es aquel objeto que codifica un mensaje y lo transmite por medio de un canal o medio hasta un receptor.

ESTÁNDAR: Es un sistema de reglas preestablecidas, condicionales, o requisitos referentes a definiciones de términos.

HERTZ: Es la unidad de medida de la frecuencia de las ondas radioeléctricas y correspondientes a un ciclo por segundo. Las ondas electromagnéticas de esta región del espectro se pueden transmitir aplicando la corriente alterna originada en un generador a una antena.

IEEE: Instituto de Ingenieros Electrónicos y Electricistas.

IEEE 802.15.4: Es un estándar que define el nivel físico y el control de acceso al medio de redes inalámbricas de área personal con tasas bajas de transmisión de datos (low-rate Wireless personal area network, LR-WPAN).

MODULO: Es un componente auto controlado de un sistema, dicho componente posee una interfaz bien definida hacia otros componentes; algo es modular si está construido de manera tal que se facilite su ensamble, acomodamiento flexible y reparación de sus componentes.

SENSOR: Es un dispositivo capaz de medir magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas. Las variables de instrumentación pueden ser, por ejemplo: temperatura, humedad relativa, intensidad,

luminosa, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, ph, etc.

TRANSDUCTOR: Convierte una magnitud física como la temperatura, presión o vibraciones en una señal eléctrica correspondiente, que es transmitida a una distancia a efectos de medición y registro.

RADIOFRECUENCIA: También denominado espectro de radiofrecuencia o RF, se aplica a la porción menos energética de espectro electromagnético, situada entre unos 3 Hz y unos 300 GHz.

RECEPTOR: Es aquel encargado de recibir una señal o código emitido por un transmisor o emisor.

TOPOLOGÍA DE RED: La topología de una red es el arreglo físico o lógico en el cual los dispositivos o nodos de una red (computadores, impresoras, servidores, switches, enrutadores, etc.) Se interconectan entre sí sobre un medio de comunicación. Las topologías de red básicas son (estrella, árbol y malla).

ARDUINO: Microcontrolador ensamblado en una placa conectada con pines para configuraciones de entrada/salida con el objetivo de ejecutar funciones específicas mediante la carga de un programa.

BIT: Unidad binaria que puede tomar el valor de cero ó uno lógico en el sistema digital.

CHIP: Siglas en inglés para Circuito Híbrido Integrado Encapsulado.

COM: Puerto standard utilizado para la comunicación de datos en forma serial.

COMUNICACIÓN SERIE: Forma de comunicación de datos usando una sola línea.

HARDWARE: Nombre que reciben los dispositivos físicos que componen un sistema electrónico ó informático.

I/O: Representación para Input/Output, es decir Entradas/Salidas por un mismo pin.

INTERFAZ: Dispositivos que usan un medio para la conversión de un sistema a otro.

TELEMETRÍA: Es una tecnología que permite la medición remota de magnitudes físicas y el posterior envío de la información hacia el operador del sistema. La palabra telemetría procede de las palabras griegas tele ("lejos") y metrón ("medida").

La **TELEMETRÍA** es una de las áreas de la ingeniería que está orientada a la medición de cualquier cantidad física, utilizando interfaces electrónicas que conectadas a través de alguna línea de transmisión ya sea un medio guiado o no guiado permiten enviar la información a un centro de gestión.

INSTRUMENTACIÓN: Es un campo de la ingeniería desarrollado para que todos los procesos, automatizados o no, funcionen de acuerdo con parametrizaciones, las cuales se basan en máquinas diseñadas por el hombre; para entender la variación de los distintos fenómenos físicos dentro de un proceso, y de acuerdo con ello tomar la posición preventiva o correctiva dentro de un modelo de gestión.

INTERNET: Red global compuesta por redes de ordenadores interconectadas mediante el protocolo TCP/IP, para facilitar la comunicación y el intercambio de datos. Abarca todo el mundo y permiten el acceso a millones de usuarios a información prácticamente sobre cualquier tema.

RED INALÁMBRICA: Conjunto de dispositivos que transmiten y reciben datos vía ondas electromagnéticas. Generalmente se emplea entre equipos situados en un radio de acción

reducido. Los sistemas de comunicaciones remotas disponen de características de seguridad como el cifrado y el salto entre frecuencias.

AP (Access Point): Punto de acceso. Es un dispositivo que conecta dispositivos inalámbricos a otra red. Por ejemplo, a una red local inalámbrica, un módem de Internet, etc.

BLUETOOTH: Es una especificación tecnológica para redes inalámbricas que permite la transmisión de voz y datos entre distintos dispositivos mediante una radiofrecuencia segura 2,4 GHz.

AMBIENTE: Con respecto a la temperatura, el ambiente se refiere a la temperatura del aire circundante.

AUTENTICACIÓN: Verifica la identidad de un usuario que inicia sesión en la red. Para comprobar la identidad del cliente en la red se utilizan contraseñas, certificados digitales, tarjetas inteligentes y biométricas. Las contraseñas y los certificados digitales se utilizan también para que el cliente identifique la red.

FIRMWARE: Programa informático dentro de un dispositivo de hardware.

ACTIVITY: Representan el componente principal de la interfaz gráfica de una aplicación Android. Se puede pensar en una actividad como el elemento análogo a una ventana en cualquier otro lenguaje visual.

APP: Tipo de programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de trabajos para dispositivos móviles o tabletas.

HANDSHAKING: La traducción de "handshaking" es "apretón de manos" y viene a significar que el procesador y los periféricos intercambian señales de control que les permiten sincronizar sus acciones y "colaborar" conjuntamente en la transferencia de información.

HTTP: Hypertext Transfer Protocol o HTTP (en español protocolo de transferencia de hipertexto) es el protocolo usado en cada transacción de la World Wide Web. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor.

IGU: Interfaz gráfica de usuario. Está construida mediante una jerarquía de View y objetos ViewGroup.

INTERRUPCIÓN: Señal recibida por el procesador de una computadora, para indicarle que debe «interrumpir» el curso de ejecución actual y pasar a ejecutar código específico para tratar esta situación.

IU: Interfaz de usuario. Todo aquello que el usuario puede ver e interactuar.

JSON: acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

MICROCONTROLADOR: Circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica. Un microcontrolador incluye en su interior las tres principales unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada/salida.

MySQL: Es un sistema de gestión de bases de datos relacional, multihilo y multiusuario.

OPEN SOURCE: Código abierto es la expresión con la que se conoce al software o hardware distribuido y desarrollado libremente. Se focaliza más en los beneficios prácticos (acceso al código fuente) que en cuestiones éticas o de libertad que tanto se destacan en el software libre.

PUERTO SERIE: Interfaz de comunicaciones de datos digitales, frecuentemente utilizado por computadoras y periféricos, donde la información es transmitida bit a bit enviando un solo bit a la vez, en contraste con el puerto paralelo que envía varios bits simultáneamente.

SMARTPHONE: El teléfono inteligente o “teléfono” (en inglés: smartphone) es un tipo teléfono móvil construido sobre una plataforma informática móvil, con una mayor capacidad de almacenar datos y realizar actividades, semejante a la de una minicomputadora, y con una mayor conectividad que un teléfono móvil convencional.

APÉNDICES

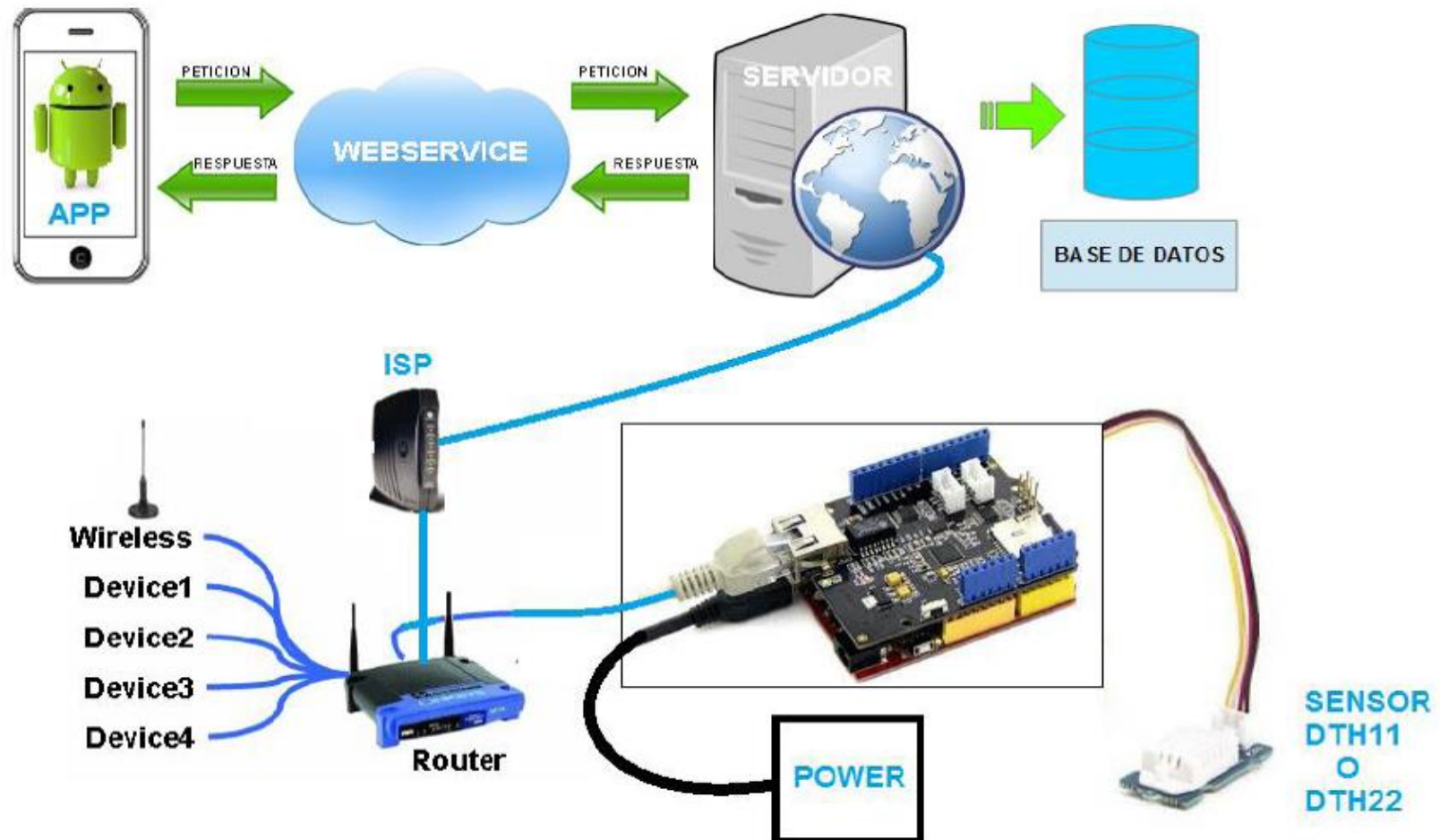


Figura 154: Esquemático representativo del circuito de monitoreo de temperatura y humedad relativa por Ethernet

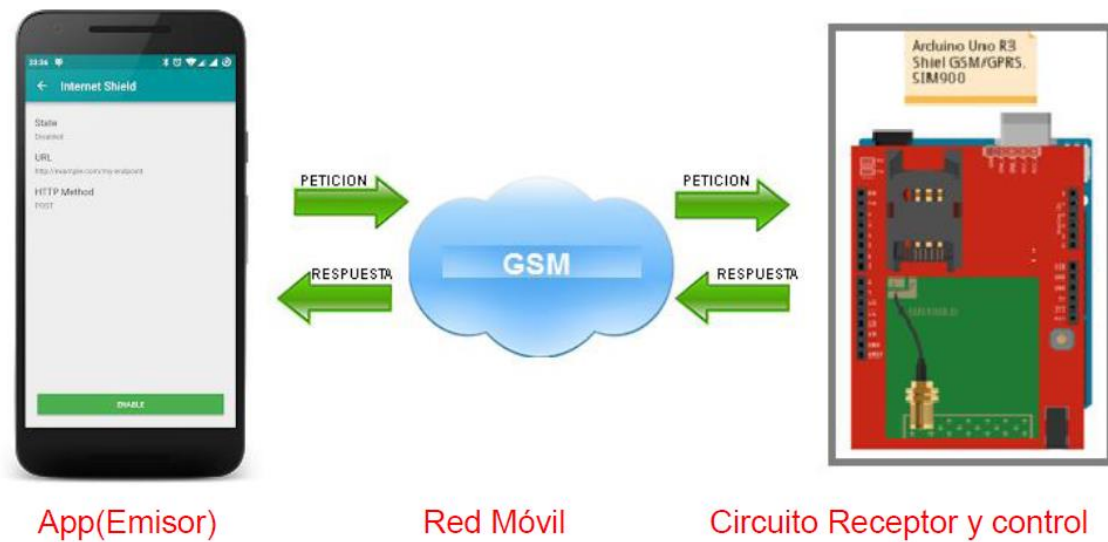


Figura 155: Esquemático representativo del circuito de control de luces remotamente con Shield GSM/GPRS SIM900

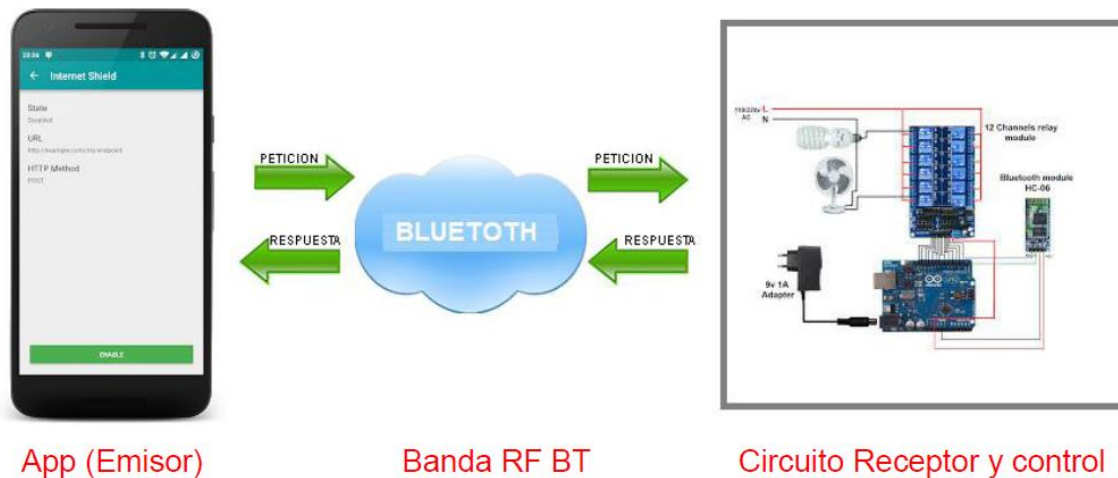


Figura 156: Esquemático representativo del circuito de control de luces localmente con Bluetooth

Nombre	Microcontrolador	Voltaje del Sistema	Frecuencia de Reloj	Digital I/O	Entradas Analógicas	Memoria Flash	Interfaz de programación
Arduino Due	AT91SAM3X8E	3,3V	84MHz	54	12	512Kb	Nativa USB
Arduino Leonardo	ATmega32U4	5V	16MHz	20	12	32Kb	Nativa USB
Arduino Uno – R3	ATmega328	5V	16MHz	14	6	32Kb	USB vía ATmega16U2
Arduino Mega	ATmega2560	5V	16MHz	54	16	256Kb	USB vía ATmega16U2
Arduino Mini	ATmega328	5V	16MHz	14	6	32Kb	Cabecera Serial

Tabla 10. Características de placas Arduino

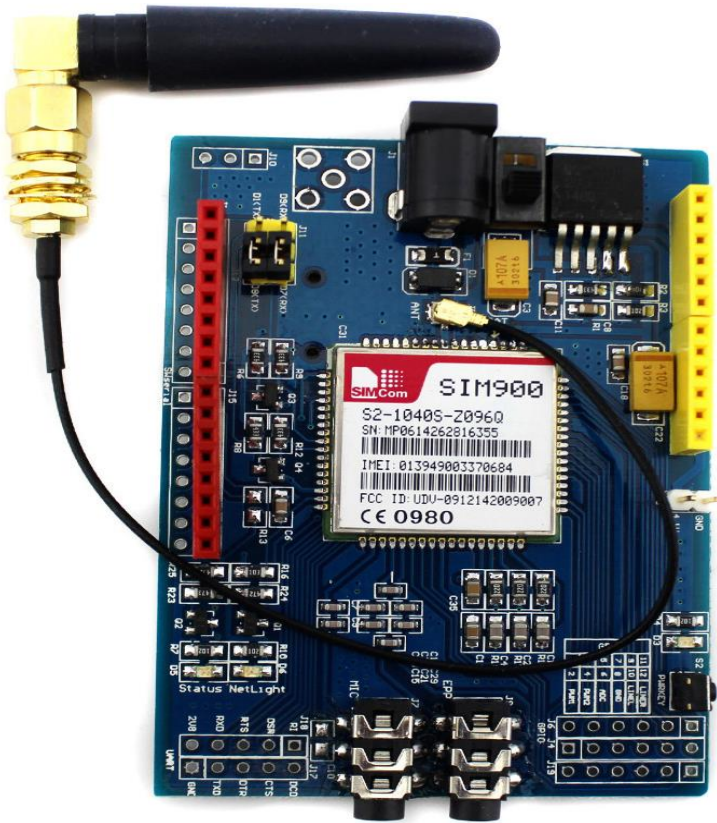


Figura 157: Tarjeta Shield GSM/GPRS SIM900

El GPRS SIMCOM SIM900 GSM Quad band GSM shield es una tarjeta ultra compacta de comunicación inalámbrica. Además, es compatible con todos los modelos de Arduino con el formato UNO, es decir, se puede controlar con otros microcontroladores también. Por último, la tarjeta está basada en el módulo SIM900 GSM 4. El GPRS está configurado y controlada por vía UART usando comandos AT. Por lo tanto, sólo conecta la tarjeta al microcontrolador, Arduino y se comienza a comunicarte a través de comandos AT. Ideal para sistemas remotos, comunicación recursiva, puntos de control, mandar mensajes de texto a celulares, etc.

ESPECIFICACIONES:

-  Totalmente compatible con Arduino.
-  Conexión con el puerto serial
-  Band 850/ 900/ 1800/ 1900 Mhz
-  GPRS multi-slot clase 10/8
-  GPRS mobile station clase B
-  Compatible GSM fase 2/2+
-  Clase 4 (2 W (AT) 850 / 900 MHz)
-  Clase 1 (1 W (AT) 1800 / 1900MHz)
-  TCP/UDP embebido
-  Soporta RTC
-  Consumo de 1.5 mA (susp)

AT	Comprueba estado del módulo.
AT+CPIN="XXXX"	Introducir el PIN de la SIM. Cambiar XXXX por el PIN.
AT+CREG?	Comprueba la conexión a la red.
ATDXXXXXXXX;	Realiza una llamada. Sustituir XXXXXXXXXX por el nº al que queramos llamar.
ATA	Descuelga una llamada.
ATH	Finaliza la llamada.
AT+CMGF=1	Configura el modo texto para enviar o recibir mensajes. Devuelve ">" como inductor.
AT+CMGS="XXXXXXXXXX"	Nº al que vamos a enviar el mensaje.
AT+CLIP=1	Activamos la identificación de llamada.
AT+CNMI=2,2,0,0,0	Configuramos el módulo para que muestre los SMS por el puerto serie.
AT+CGATT=1	Conectamos a la red GPRS.
AT+CSTT="APN","usuario","contraseña"	Definimos APN. usuario y contraseña
AT+CIICR	Activamos el perfil de datos inalámbrico
AT+CIFSR	Obtenemos nuestra IP

AT+CIPSTART="TCP","direccionIP","puerto"	Indicamos el tipo de conexión, dirección IP y puerto al que realizamos la conexión
AT+CIPSEND	Preparamos el envío de datos. Devuelve ">" como inductor.
AT+CIPCLOSE	Cerramos la conexión.
AT+CIPSHUT	Cierra el contexto PDP del GPRS
AT+CGPSPWR=1	Activar el GPS.
AT+CGPSSTATUS?	Comprueba que el GPS ha encontrado la red.
AT+CGPSINF=0	Obtiene los datos del GPS: Modo, Latitud, Longitud, Altitud, Horario UTC, Tiempo de respuesta, Número de satélites, Velocidad, Curso

Tabla 11. Comandos AT más usados en los módulos GSM/GPRS y GPS SIM900 y SIM808

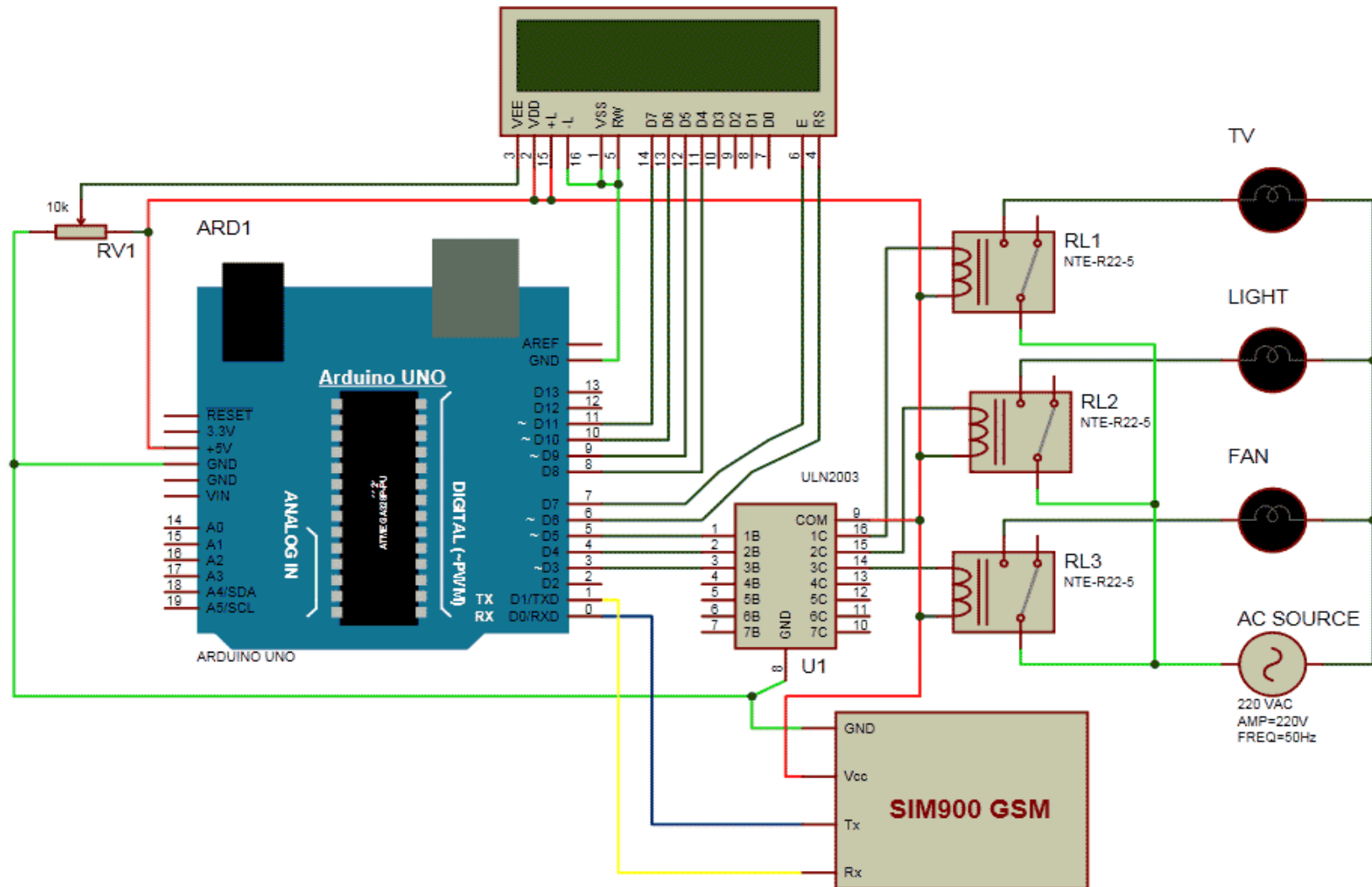


Figura 158: Simulación en software Proteus del sistema de control de cargas (Luminarias)

N°	Herramientas de Software Utilizadas
1	IDE de Arduino
2	IDE de Android Studio
3	XAMPP Control Panel
4	Servidor de Aplicaciones PHP
5	Servidor Web Apache
6	Base de datos MYSQL y SQLite
7	Dominio de pruebas: mjgl.com.sv
8	JSON, JAVASCRIPT, CSS, HTML5
9	Framework Bootstrap
10	Editores de texto: Visual Studio Code, Atom, Sublime Text 3, Notepad++, Dreamweaver
11	Postman
12	

Tabla 12. Herramientas de Software Utilizadas

N°	Electrónica Utilizada
1	Tarjeta Shield Ethernet V2 sin PoE
2	Tarjeta Shield GSM/GPRS SIM900
3	Tarjeta Arduino UNO R3
4	Tarjeta Arduino MEGA
5	Sensor DHT22 y dependiente de luz o LDR
6	Sensor Corriente SCT013
7	ROUTER
8	Bluetooth HC-05
9	Antena GSM con conector SMA, cable RG174 3 Mtrs
10	Cable RG174 10mts conectores SMA macho hembra

11	Antena - GSM con conector SMA, cable RG174 5 Mtrs
11	Power Adapter - Cargador Ard. DC 9V 3 A
12	Power Adapter - Cargador Ard. DC 5V 2 A
13	Mini Cooler Fan
14	Mini Disipador de aluminio
15	Tarjeta de Relés

Tabla 13. Principales componentes electrónicos del sistema

Parámetro	DHT11	DHT22
Alimentación	$3Vdc \leq Vcc \leq 5Vdc$	$3.3Vdc \leq Vcc \leq 6Vdc$
Señal de Salida	Digital	Digital
Rango de medida de Temperatura	De 0 a 50 °C	De -40°C a 80 °C
Precisión Temperatura	± 2 °C	$<\pm 0.5$ °C
Resolución Temperatura	0.1°C	0.1°C
Rango de medida Humedad	De 20% a 90% RH	De 0 a 100% RH
Precisión Humedad	4% RH	2% RH
Resolución Humedad	1%RH	0.1%RH
Tiempo de sensado	1s	2s
Tamaño	12 x 15.5 x 5.5mm	14 x 18 x 5.5mm

Tabla 14. Características de sensores DHT

ANEXOS

Evidencia de imágenes del equipo implementado en el laboratorio de electrónica de la Universidad Técnica Latinoamericana.











Data sheet de W5500:

https://www.sos.sk/productdata/15/26/12/152612/W5500_datasheet_v1.0.2_1.pdf

Más información sobre esta tarjeta en:

<http://www.arduino.org/products/shields/arduino-ethernet-shield-2>

Este Shield usa la librería ethernet2:

<http://www.arduino.org/learning/reference/Ethernet-two-Library>

Librería SD:

<http://arduino.cc/en/Reference/SD>

Mejoras de W5500:

<https://feilipu.me/2014/11/16/wiznet-w5500-ioshield-a/>

Data sheet de Power Over Ethernet (POE):

<http://arduino.cc/en/uploads/Main/PoE-datasheet.pdf>

Más información sobre el Power Over Ethernet (POE):

http://es.wikipedia.org/wiki/Power_over_Ethernet

http://en.wikipedia.org/wiki/Power_over_Ethernet

Datasheet Chip e información sobre ethernet IPv6:

- ❖ <http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en022889>
- ❖ <http://ww1.microchip.com/downloads/en/DeviceDoc/39662c.pdf>
- ❖ <https://sites.google.com/site/ghoelzl/ipv6>

IPv6WebServer:

- ❖ https://sites.google.com/site/ghoelzl/ipv6ethershield/ipv6_http_server
- ❖ <http://www.tweaking4all.com/hardware/arduino/arduino-enc28j60-ethernet/>
- ❖ http://www.fut-electronics.com/wp-content/plugins/fe_downloads/Uploads/Ethernet-Module-ENC28J60-Arduino.pdf

Librerías:

- ❖ https://github.com/ntruchsess/arduino_uip
- ❖ <https://github.com/jcw/ethercard>
- ❖ <https://github.com/ghoelzl/IPv6EtherShield>
- ❖ <https://sites.google.com/site/ghoelzl/ipv6ethershield>