



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO
CENTRO UNIVERSITARIO UAEM TEXCOCO



AUTOMATIZACIÓN DE UN SISTEMA DE INMERSIÓN
TEMPORAL CON CONTROL MULTIPLATAFORMA PARA LA
REGENERACIÓN DE ESPECIES DE FLORA *IN VITRO*

T E S I S

PARA OBTENER EL GRADO DE
MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA

DIEGO HERNÁNDEZ COTE

TUTOR ACADÉMICO

D. EN C. OZIEL LUGO ESPINOSA

TUTORES ADJUNTOS

DR. EN C. ALFONSO ZARCO HIDALGO

DR. EN E. JOEL AYALA DE LA VEGA

TEXCOCO. ESTADO DE MÉXICO, NOVIEMBRE DE 2016



DICTAMEN DE AUTORIZACIÓN Y OBTENCIÓN DE GRADO DE MAESTRÍA

Texcoco, Méx. , a 11 de julio del 2016

Título del proyecto:

Automatización de un Sistema de Inmersión Temporal con control multiplataforma para la regeneración de especies de flora *in Vitro*

Tesista:

Diego Hernández Cote

Dictamen:

No. de revisión: 5

- Rechazado
- Sujeto a modificaciones
- Aceptado, condicionado
- Aceptado

**POSGRADO
TITULACION
RECIBIDO**
Por Adriana Arevalo
Texcoco, Méx., a 11 de VII del 2016

Observaciones generales:

Acceptado para la impresión

Acceptado para la defensa de grado

Tutor Adjunto	Tutor Académico	Tutor Adjunto
Dr. Joel Ayala de la Vega	Dr. Oziel Lugo Espinosa	Dr. Alfonso Zarco Hidalgo





Contenido

Tabla de ilustraciones	6
Índice de tablas	9
Resumen.....	10
Introducción	10
Planteamiento del problema	12
Objetivos	13
Objetivo general	13
Objetivos específicos.....	13
Hipótesis.....	13
Justificación	14
Metodología.....	15
Antecedentes	19
Software	20
Hardware.....	22
Arduino.....	22
Control del clima en SITs.....	24
Campo de estudio: COLPOS	25
Capítulo I: Marco Teórico	28
1.1 Cultivo in vitro	28
1.2 SIT: Sistema de Inmersión Temporal.....	29
1.3 Automatización.....	31
1.3.1 Monitorización.....	32
1.4 Control multiplataforma.....	32
Capítulo II: Materiales en la automatización.....	35

2.1 Plataforma de hardware libre Arduino	35
2.1.1 Tipos de placas Arduino.....	36
2.1.2 Placa Arduino empleada para la automatización del SIT	39
2.1.3 Programación en Arduino.....	40
2.1.4 Entorno de desarrollo integrado de Arduino	44
2.2 Sensores	46
2.3 Dispositivos electrónicos controlados	48
2.3.1 Tira de led.....	48
2.3.2 Relevadores	50
2.4 Comunicación GSM	51
Capítulo III: Desarrollo	54
Prototipo 1	54
Desarrollo del Hardware.....	55
Desarrollo del software.....	60
Implementación y resultados.....	79
Prototipo 2	83
Desarrollo del Hardware.....	84
Desarrollo del software.....	86
Implementación y resultados.....	101
Prototipo final	106
Desarrollo de hardware	107
Desarrollo de software	108
Implementación y resultados.....	113
Capítulo IV: Análisis y resultados	116
Fase de prueba: 18 de diciembre de 2014	116
Día 19 de diciembre de 2014	117
Día 21 de diciembre de 2014	118
Día 22 de diciembre de 2014	119
Día 25 de diciembre de 2014	120
Día 27 de diciembre de 2014	120



Día 28 de diciembre de 2014.....	121
Día 29 de diciembre de 2014.....	121
Día 31 de diciembre de 2014.....	122
Día 2 de enero de 2015.....	122
Día 4 de enero de 2015.....	123
Día 7 de enero de 2015.....	124
Día 9 de enero de 2015.....	124
Día 13 de enero 2015.....	125
Día 16 de enero de 2015.....	125
Día 19 de enero de 2015.....	126
Resultados obtenidos.....	127
Capítulo V: Conclusiones	133
Bibliografía.....	135
Anexo I.....	139

Tabla de ilustraciones

Ilustración 1: Parte de un SIT cortesía del COLPOS	12
Ilustración 2. Metodología Buttom Up	15
Ilustración 3. Metodología Kanban	16
Ilustración 4 Diagrama de la metodología	16
Ilustración 5. Ubicación del COLPOS.....	25
Ilustración 6. COLPOS edificio principal.....	26
Ilustración 7. Departamento de Biotecnología de frutales.....	27
Ilustración 8 Biorreactores en algunos SIT, cortesía del COLPOS	30
Ilustración 9 Diagrama de bloques del sistema de control abierto	33
Ilustración 10 Diagrama de bloques del sistema de control cerrado	33
Ilustración 11. Arduino UNO R3	40
Ilustración 12. Funciones void() y setup().....	41
Ilustración 13. Función de retorno con una variable.....	42
Ilustración 14. Funciones de Arduino y consola serial.....	44
Ilustración 15. IDE de Arduino.....	46
Ilustración 16. Fotorresistencia	47
Ilustración 17. Sensor de temperatura	48
Ilustración 18. Esquema de conexión de la tira de led	50
Ilustración 19 Tira de relevadores para Arduino.....	51
Ilustración 20 Shield GSM utilizada en el SIT	53
Ilustración 21. Diagrama UML del prototipo 1	55
Ilustración 22 Conexión del sensor de temperatura a Arduino Mega 2560	57
Ilustración 23 Conexión de fotorresistencias a la placa Arduino Mega 2560	58
Ilustración 24 Conexión de relevadores a la placa Arduino.....	59
Ilustración 25 Conexión física de una tira de led.....	59
Ilustración 26. Diseño de la base de datos. Prototipo 1	68
Ilustración 27 Alerta de conexión a Arduino	70
Ilustración 28 Ventana principal de la aplicación Java	71
Ilustración 29 Ventana de monitoreo por sector	72



Ilustración 30 Ventana de cambio de nombre.....	73
Ilustración 31 Ventana de horarios de bombeos.....	73
Ilustración 32 Ventana para guardar un nuevo horario	74
Ilustración 33 Advertencia de temperatura baja.....	75
Ilustración 34 Advertencia de temperatura alta.....	75
Ilustración 35 Advertencia de falta de luz	76
Ilustración 36 Advertencia de existencia de luz	76
Ilustración 37 Diagrama del hilo de temperatura	77
Ilustración 38 Diagrama del proceso de fotorresistencias.....	78
Ilustración 39 Diagrama del proceso de detección de horas para el bombeo.....	78
Ilustración 40 Diagrama del hilo activador de relevadores.....	79
Ilustración 41 Prototipo I colocado en el SIT del COLPOS.....	80
Ilustración 42 Biorreactores en el SIT automatizado.....	80
Ilustración 43 Biorreactores fuera del SIT automatizado	81
Ilustración 44 Mejor biorreactor con la automatización	81
Ilustración 45 El mejor biorreactor sin la automatización	82
Ilustración 46. UML del prototipo 2	84
Ilustración 47 Elementos de hardware en Arduino UNO.....	85
Ilustración 48 Shield GSM en Arduino UNO	86
Ilustración 49. Diagrama de base de datos, prototipo 2.....	89
Ilustración 50 Ventana principal en el prototipo II	90
Ilustración 51 Ventana de usuarios.....	91
Ilustración 52 Ventana de nuevo usuario.....	92
Ilustración 53 Ventana de monitoreo por sector en el prototipo II.....	93
Ilustración 54 Ventana de horarios de iluminación.....	94
Ilustración 55 Ventana de nuevo horario de iluminación.....	95
Ilustración 56 Interfaz principal de la aplicación Android	96
Ilustración 57 Layout del número telefónico.....	96
Ilustración 58 Toast con el texto "Petición enviada".....	97
Ilustración 59 Diagrama del hilo de iluminación.....	98
Ilustración 60 Diagrama de casos de uso de la aplicación Android.....	99

Ilustración 61 Maqueta de pruebas del prototipo II	101
Ilustración 62 Anillo de ferrita conectado al cable de comunicación	102
Ilustración 63 Cambio de luz en un sector del SIT	103
Ilustración 64 Cambio simultaneo de colores en el SIT.....	103
Ilustración 65 Recepción en Android del mensaje de reinicio	104
Ilustración 66 Estado del SIT cuando se solicitó la información desde Android.....	105
Ilustración 67 Respuesta enviada desde el SIT	105
Ilustración 68 Shield GSM separada en otra placa Arduino	108
Ilustración 69 Layout principal del prototipo final.....	110
Ilustración 70 Notificación sin respuesta del SIT	113
Ilustración 71 Arduinos Uno del prototipo final en funcionamiento	114
Ilustración 72. Evidencia del desarrollo de explantes.....	116
Ilustración 73. Elementos de investigación	117
Ilustración 74. Sistema automatizado instalado en el SIT	118
Ilustración 75. 21 de diciembre de 2014.....	119
Ilustración 76. 22 de diciembre de 2014.....	119
Ilustración 77. 25 de diciembre de 2014.....	120
Ilustración 78. 27 de diciembre de 2014.....	120
Ilustración 79. 28 de diciembre de 2014.....	121
Ilustración 80. 29 de diciembre de 2014.....	121
Ilustración 81. 31 de diciembre de 2014.....	122
Ilustración 82. 2 de enero de 2015.....	123
Ilustración 83. 4 de enero de 2015.....	123
Ilustración 84. 7 de enero de 2015.....	124
Ilustración 85. 9 de enero de 2015.....	124
Ilustración 86. 13 de enero de 2015.....	125
Ilustración 87. 16 de enero de 2015.....	125
Ilustración 88. Corte, medición y conteo de brotes	126
Ilustración 89. Explantes en los biorreactores destapados	126
Ilustración 90. Frascos donde se depositaron los brotes	127
Ilustración 91. Gráfico del porcentaje de propagación	128



Ilustración 92. Gráfico del periodo de los bombeos	129
Ilustración 93. Gráfico de la duración de bombeos	130
Ilustración 94. Tiempo de iluminación y tipo de color en los biorreactores	131
Ilustración 95. Total de brotes por biorreactor	132
Ilustración 96. Brote de tamaño máximo y mínimo por biorreactor	132

Índice de tablas

Tabla 1. Especificaciones Diecimilia. Fuente: www.arduino.cc	36
Tabla 2. Especificaciones Duemilanove. Fuente: www.arduino.cc/	37
Tabla 3. Especificaciones Mega 2560. Fuente: www.arduino.cc	38
Tabla 4. Especificaciones DUE. Fuente: www.android.cc	38
Tabla 5. Especificaciones UNO. Fuente: www.arduino.cc	39
Tabla 6. Puertos de Arduino Mega para el cambio de luz	60
Tabla 7. Rutas donde para los archivos de RXTX	69
Tabla 8. Peticiones de la App al SIT	111
Tabla 9. Mensajes de alerta en la App.....	112
Tabla 10. Mensajes del SIT a la App	112
Tabla 11. Resultados de la implementación en ambos sectores	128
Tabla 12. Total de brotes y tamaños.....	132
Tabla 13. Costos de la automatización	134

Resumen

El presente trabajo condensa los puntos principales del desarrollo del proyecto de automatización de un Sistema de Inmersión Temporal, cuyo principal objetivo consiste en el perfeccionamiento de las condiciones en la producción de flora que se reproduce en los bioreactores. Para la realización del proyecto se desarrolló tanto el hardware que permitiera la automatización, como el software necesario para controlarlos, por medio de una computadora y mediante un dispositivo móvil, lo que facilitará su monitorización y manejo.

Introducción

En los últimos años el desarrollo tecnológico ha impactado en diversos ámbitos de nuestras vidas, por lo que la innovación en este campo se ha convertido en un factor indispensable que permite satisfacer necesidades y facilita el trabajo de la población. El desarrollo tecnológico es constante y permanente por lo que es fundamental emplearlo para beneficiar a la sociedad en general. Teniendo en cuenta que la agricultura, así como la producción y reproducción de variedades vegetales es necesaria para diversos fines, desde la alimentación hasta la conservación ecológica de algunas especies, estos se han convertido factores de interés en el aspecto tecnológico.

La producción de un cultivo es de suma importancia en cualquier país y abarca desde la producción de frutas o vegetales, hasta la producción y reproducción de plantas en peligro de extinción. Actualmente los cultivos expuestos al ambiente han tenido menor producción que aquellos cultivados mediante la implementación de tecnologías y sistemas de micropropagación, que aceleran la producción y mejoran la calidad del producto al tener un entorno regulado de acuerdo con las *características de desarrollo* propias del cultivo.

El SIT es una de las técnicas empleadas para la producción de semillas y consiste en un sistema empleado para producir plantas difíciles de cultivar y de las que se necesita una mayor producción o que están en peligro de extinción, pues a partir de una hoja pueden extraerse cientos de embriones, elementos sumamente delicados



que requieren de cuidados continuos y periódicos, durante el día y la noche. Desde el inicio de su desarrollo hasta que son capaces de resistir en invernaderos, el cuidado y mantenimiento de estos sistemas es un proceso complicado y requiere personal dedicado día y noche a la regulación y cambios de luz, temperatura y bombes de nutrientes, por esta razón es de suma importancia automatizar algunos procesos para mejorar las condiciones que existen dentro de un SIT.

El control de sistemas automatizados por medio de software es común hoy en día, ya que ofrece al usuario la opción de manipular el sistema conforme a sus necesidades; sin embargo, la posibilidad de controlar el sistema de forma remota resulta más provechoso para las personas, que pueden supervisar, manipular y controlar las condiciones del sistema en tiempo real.

La importancia de controlar el ambiente en un SIT es el beneficio al desarrollo y el crecimiento que se presenta en el cultivo, por lo que se hizo factible el desarrollar una automatización que mantenga la iluminación y las inmersiones si necesidad de que una persona lo opere, además agregar un control remoto en dispositivos móviles da la posibilidad de alterar y monitorear los elementos de la automatización para el beneficio del cultivo desde fuera del SIT.

En el presente documento se describe el desarrollo de toda la automatización, incluyendo el hardware y el software creado para el control y monitoreo del SIT. La exploración de campo, las necesidades que se tienen en un SIT y su automatización son explicados en el *Capítulo I*, posteriormente en el *Capítulo II* se describen los materiales inmersos en el hardware y la razón por las que fueron usados, a continuación, en el *Capítulo III* se desarrollan los prototipos creados, así como sus mejoras y finalmente obteniendo un prototipo final, finalmente en el *Capítulo IV* se muestran las conclusiones obtenidas después del análisis de resultados.

Planteamiento del problema

Las condiciones climáticas actuales han complicado la producción agrícola en condiciones naturales, por lo que se han tomado medidas para mejorar y aumentar la producción vegetal. En este sentido, existen diversas técnicas como la colocación de luz artificial, la regulación de la temperatura y el bombeo de nutrientes al birreactor, para mejorar la resistencia, el crecimiento y la propagación de las especies. Cabe mencionar que el uso, control y manejo de estos elementos es complicado y requiere de la preparación del personal, así como de su supervisión constante; por ello, es necesario implementar métodos de automatización que faciliten el monitoreo, control y regulación de estos elementos con el fin de mejorar y aumentar la producción vegetal.

La mayoría de estos sistemas cuentan con calefactores y luces para mantener un ambiente regulado; sin embargo, el personal no es suficiente para supervisarlos 24 horas al día, por lo que el entorno en los biorreactores puede ser inestable mientras no se encuentre bajo supervisión. En la *ilustración 1* se muestra un sector de un SIT, cortesía del Colegio de Posgraduados (COLPOS).

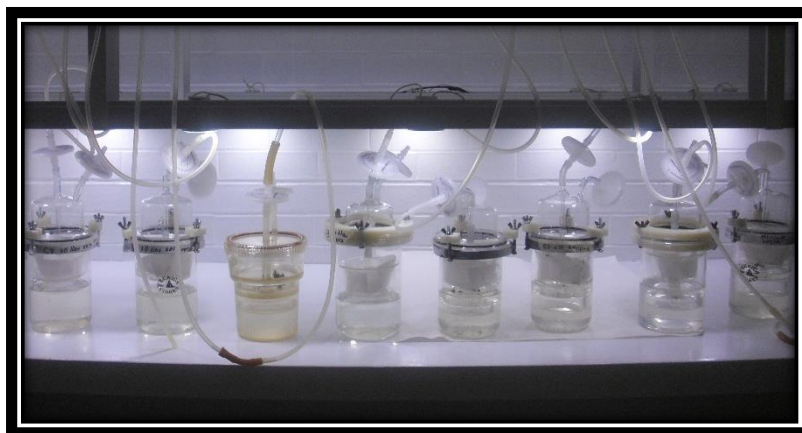
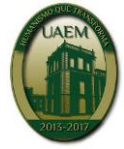


ILUSTRACIÓN 1: PARTE DE UN SIT CORTESÍA DEL COLPOS



Objetivos

Objetivo general

- Realizar un proyecto de automatización, control y monitoreo de un SIT, para mejorar la producción y calidad del cultivo.

Objetivos específicos

- Investigación de las propiedades de un SIT para determinar sus necesidades.
- Revisión documental respecto a las temáticas a desarrollar en el proyecto: sistemas de inmersión temporal, automatización y desarrollo de software.
- Análisis de requerimientos para la automatización del sistema.
- Elaboración de un diseño de automatización.
- Diseño de programas de control de dispositivos necesarios para llevar a cabo la automatización.
- Prueba y reparación de errores.
- Implementación del sistema.

Hipótesis

Si se automatiza un SIT, entonces mejorará el estado del cultivo y su desarrollo, aumentando de esta manera su producción.

Justificación

Debido a las grandes pérdidas de cultivos y a las numerosas especies botánicas en peligro de extinción, es necesario preservar y producir el mayor número posible de ejemplares, con la intención de mantener el material genético además de aprovechar, controlar y aumentar la producción de especies específicas. En este sentido, la automatización de un SIT, controlada por medio de un software elaborado específicamente para la automatización del sistema y una aplicación para la manipulación remota, proporciona al usuario la posibilidad de regular y supervisar en tiempo real, el ambiente en el que se desarrolla el cultivo de los biorreactores. El monitoreo permite notar la pertinencia de las variables y en caso negativo modificarlas para lograr el ambiente preciso que el cultivo requiere, incrementando de esta manera la producción de plantas.

La automatización del SIT permite solucionar problemáticas relacionadas con la falta de conocimiento de los usuarios que intervienen en el sistema de inmersión temporal, así como la dificultad que representa el control y manejo de los dispositivos empleados para mantener las condiciones ambientales idóneas para la producción de cualquier cultivo, específicamente sí este se encuentra en la etapa embrionaria de su desarrollo.



Metodología

Para el desarrollo del hardware, se empleó el uso de la metodología Bottom UP, donde las partes individuales se diseñan con detalle y luego se enlazan para formar componentes más grandes, que a su vez se enlazan hasta que se forma el sistema completo. Una de las peculiaridades del trabajo con esta metodología es la cuestión del ensayo y error para la mejora del producto, lo que implica que el desarrollo a partir de una idea principal y su posterior mejora permite enfrentar el hardware y software elaborados a situaciones y problemáticas reales, por lo que es posible concluir el proyecto y reparar o solucionar los conflictos que surgen durante su desarrollo (Berón, 2006). El diagrama de la metodología se presenta en la *Ilustración 2*.

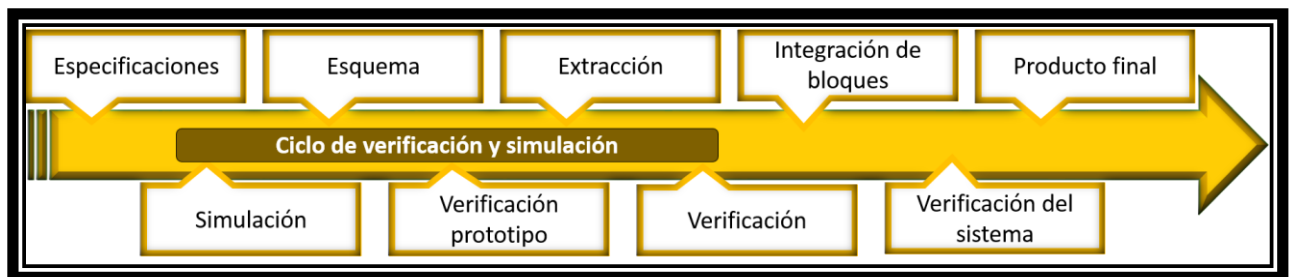


ILUSTRACIÓN 2. METODOLOGÍA BUTTOM UP

Para el desarrollo del software, se empleó el uso de la metodología Kanban (*Ilustración 3*), que consiste en la implementación de un modo de trabajo eficiente y efectivo, al tener en cuenta la producción y logística en todo momento, de manera que es posible lograr el desarrollo de un plan de trabajo específico que permita el desarrollo del objetivo de manera óptima.

Cumplir con estas reglas permite una adecuada gestión de recursos humanos, materiales e intelectuales, así como la organización del plan de trabajo, lo que conlleva a un trabajo bien logrado (Núñez, 2002).



De acuerdo con el diagrama la metodología incluye la elaboración de una investigación documental, definida por Bernal (2006) como “un análisis de la información escrita sobre un determinado tema, con el propósito de establecer relaciones, diferencias, etapas, posturas o estado actual del conocimiento respecto del tema objeto de estudio”.

Asimismo, se llevó a cabo una investigación de campo, que García (2006) define como “la recolección de datos directamente de los sujetos investigados, o de la realidad donde ocurren los hechos”, lo que implica la recolección de datos y elaboración del proyecto *in situ*.

Como parte de la investigación documental, es necesario realizar una búsqueda teórica de los elementos con el fin de conocer sus funciones, definiciones y características, para comprender el marco general de los elementos necesarios para desarrollar el proyecto, en este sentido, éste tipo de investigación sirvió como un referente que permitió la elección de los componentes ideales para el proyecto. Por otro lado, la investigación de campo caracterizada por una recolección de datos en el lugar de estudio, en este caso en el COLPOS, se permitió conocer las necesidades del campo, para elegir de esta manera los componentes que cuentan con las características óptimas para satisfacer las necesidades del campo considerando como base la teoría.

La investigación de campo y documental, fueron empleadas durante todo el proceso de indagación, pues desde la revisión teórica, exploración de campo, el desarrollo del proyecto en general e implementación, fue necesaria la revisión bibliográfica que permitió establecer una relación de concordancia entre el campo, la teoría y el proyecto. Con base en los resultados obtenidos de la revisión teórica-documental y de la exploración de campo, se obtuvieron una serie de datos que permitieron realizar un análisis de requerimientos para la automatización del sistema, contemplando las necesidades del usuario y las condiciones específicas del SIT.

Tras la elaboración de un análisis relativo a las necesidades y requerimientos del SIT, fue necesario desarrollar diagramas, planos y esquemas donde se plasmó la

ubicación de los dispositivos electrónicos, conocidos como hardware, que se utilizaron para la automatización del SIT ya que es de suma importancia para el funcionamiento óptimo del sistema automatizado, ubicar los componentes de manera correcta.

Cuando se realizó la automatización del SIT, se creó el software por medio del cual, el usuario puede monitorear y controlar el SIT automatizado de acuerdo con las necesidades del cultivo, para lo que se creó una aplicación de escritorio que permite al usuario monitorear y verificar las condiciones del SIT de manera sencilla y rápida.

La aplicación de escritorio está orientada a PC u ordenadores. Tras el conocimiento obtenido en la teoría-documentación se eligió el lenguaje de programación orientada a objetos Java, ya que ofrece ventajas tanto con el hardware como con el software que son ideales para el control del SIT automatizado.

La aplicación remota está orientada a teléfonos móviles con sistema operativo Android ya que ofrece las ventajas de estar presente en casi todos los Smartphones del mercado, además de ofrecer los medios necesarios para una comunicación remota con el hardware que se implementó en la automatización del SIT.

Después de realizar la automatización del SIT y de la creación del software que controla y monitorea el sistema, se realizaron pruebas en las que se instaló el SIT en un sector con biorreactores para comprobar su funcionamiento.

Finalmente, cuando se comprobó el correcto funcionamiento de la automatización y del software, este se implementó en forma de prototipos utilizando una maqueta con biorreactores caseros. Posteriormente se evaluaron los resultados y se realizó el análisis correspondiente, que permitió comprobar la hipótesis satisfactoriamente.



Antecedentes

Muñoz y Núñez (2012) mencionan en su tesis *Automatización De Invernadero En Clima Templado* que el monitoreo automatizado de los factores ambientales en un cultivo, garantiza el cumplimiento de las condiciones de operación del invernadero, optimizando de esta manera el desarrollo del cultivo. La consulta del estado de un SIT es indispensable para que el usuario avale el entorno que tiene su cultivo o bioreactores.

En la tesis *Control y automatización de un sistema de bombeo de un invernadero para el desarrollo, cultivo e investigación de flora* (Díaz, García y Espinosa, 2011) se menciona que “un invernadero de investigación controlado y automatizado, facilita el desarrollo de nuevos cultivos, previniendo plagas o enfermedades que puedan adquirir las plantas, obteniendo de manera significativa mayor producción y calidad en cultivos. [...] Establecer un control eficiente permite contrarrestar el efecto climático que ocasiona una baja producción”, lo que permite notar la importancia de establecer un ambiente controlado para obtener mejores resultados respecto al cultivo.

En el informe *Evaluación de dos métodos de micropropagación masal en piña (Ananas comosus L. Merr.) variedad golden* (Ángel, González, Orellana, & Arévalo, 2013) se expresa que “es necesario contar con un sistema automatizado que realice frecuencias y tiempo de inmersión programada para obtener una mejor eficiencia del Sistema de Inmersión Temporal en las fases de multiplicación, desarrollo y enraizamiento”. En este sentido, no sólo es importante contar con un SIT, pues el sistema no asegura por sí mismo la mejora en cuanto al desarrollo y producción de cultivo, sino que debe acompañarse de la automatización que permita controlar firmemente las condiciones ambientales del cultivo.

En este sentido, tenemos una investigación anterior al año 2000 en la que podemos notar la importancia que existía desde hace más de 14 años de lograr la automatización de un SIT, esta es una investigación del Centro Nacional de

Electromagnetismo Aplicado (CNEA) de la Universidad de Oriente titulado: *Sistema de inmersión temporal automatizado* en el que se menciona que “La eficiencia tecnológica de un proceso, dada la naturaleza físico-química y biológica del mismo, depende en gran medida de la calidad del control de las variables del proceso... y así mismo, que ... el principio de funcionamiento de los sistemas de inmersión intermitente se basa en la posibilidad de absorción de nutrientes y otras sustancias por las plantas "in vitro", durante períodos alternos de inmersión en el medio de cultivo líquido y posterior permanencia en el recipiente sin el medio de cultivo (aunque sí con una atmósfera de una elevada humedad relativa), este proceso de nutrición alternativa suministra los elementos necesarios al material vegetal e incrementa notablemente la oxigenación del medio interno, lo cual influye positivamente en el crecimiento y multiplicación de las plantas” (Mulet, et. Al, 1999).

La investigación antes mencionada es de suma importancia para el presente trabajo, pues muestra que existía la necesidad de la automatización de los SITs desde hace 14 años, sin embargo, el trabajo mencionado se concentra en la automatización de electro valvulas, elaborado mediante dispositivos electronicos y valvulas electronicas, un manometro de contacto electrico, etc., por lo que es evidente que aunque la automatización que se hizo fue sumamente avanzada para su momento, fue básica comparada con las necesidades del SIT.

Software

Como se muestra en el artículo *Interfaz Gráfica de Usuario para Pruebas de Marcha en un Robot Bípedo*; “La interfaz gráfica... tiene como propósito ser una herramienta intuitiva y fácil de usar, mediante la que el usuario podrá interactuar con el robot de una manera rápida y efectiva, facilitándole la realización de pruebas y experimentos” (López García, Narváez-A roche, & Rocha Cózatl, 2011), las interfaces deben ser lógicas para el usuario; es decir que el usuario pueda comprender de manera fácil, rápida y sin preparación especializada el trabajo que logran hacer los sistemas electrónicos.



Por su parte, en la tesis *Monitoreo de una Estación Meteorológica Vía Internet* “el pronóstico del tiempo es uno de los factores más importantes en nuestra vida diaria ya que permite realizar ciertas tareas sabiendo que no serán impedidas, arruinadas o complicadas por el tiempo, cuando se pueden tomar precauciones para evitar sus consecuencias o suspenderlas hasta un momento más propicio” (Guardado & Palacios, 2009).

Sin embargo, Medina menciona que “dado que el clima es uno de los componentes ambientales más determinantes en la adaptación, distribución y productividad de los seres vivos, la información del estado del tiempo es parte fundamental para la toma de decisiones en la agricultura moderna que requiere información meteorológica actualizada para orientar los procesos de producción” (Medina García, Grageda Grageda, Ruiz Corral, & Báez González, 2008).

La obtención de datos ambientales y su visualización proveerá a los usuarios de información útil que determinará la realización de determinadas actividades en el espacio que se encuentra y que necesiten un ambiente adecuado. La información del entorno es un factor importante para favorecer la producción en el cultivo, permitiendo que el usuario tome las medidas necesarias para la protección de sus biorreactores y la mejora en su producción.

Así mismo, existen otros escritos en los que se expresa la importancia de los SITs en cuanto a la producción de embriones y hasta clones de ciertas especies, es el caso del texto *Multiplicación en sistema de inmersión temporal del clon de malanga “Viequera” (Xanthosoma spp.)*, artículo científico escrito por Santos *et.al.* (2011), quien menciona que esta especie puede producirse con menores índices de variación genética, importante para la conservación de especies nativas, y puede producirse libre de agentes contaminantes, mostrando el cultivo en SITs como una alternativa viable de micro propagación a corto plazo.

Hardware

Dentro del área, existen diferentes opciones por las cuales una automatización puede llevarse a cabo, por ejemplo, un PLC se enfoca al control de eventos o procesos en tiempo real, sin embargo, debe ser programado en un software especializado, sus costos altos e incluye dispositivos de alto costo o difíciles de conseguir (Vallejo, 2005). El PLC acorde a las necesidades del proyecto es el modelo ELC-PA10AADR que tiene un costo actual de \$ 10,800.00 MN¹.

Otra alternativa de automatización son los FPGA, dispositivos versátiles de fácil programación y capaces de realizar funciones lógicas (Castillo, Ortegón & Rodríguez, 2008); sin embargo, contiene componentes agregados que no son útiles para proyectos sencillos, creando un desperdicio de recursos que impedirían un diseño óptimo del circuito. Un FPGA acorde a las necesidades del proyecto tiene un costo promedio actual de \$ 3530.52 MN², sin contar los componentes de control.

Tras el análisis de las opciones presentadas anteriormente, se llegó a la conclusión de utilizar una plataforma de hardware libre, el cual es compatible con los dispositivos electrónicos a usar en el presente proyecto; dicha plataforma es llamada Arduino.

Arduino

En el artículo *Lab Kits Using the Arduino Prototyping Platform* se menciona que “basado en un microcontrolador de una placa Arduino y el hardware abierto (hardware libre) que permite a estudiantes usarlo con un bajo costo, manejando hardware específico para completar ejercicios de laboratorio en casa, la plataforma está diseñada para ser accesible a un rango de estudiantes y de fácil adaptación para otras aplicaciones” (Sarik & Kymissis, 2010).

¹ http://mx.omega.com/pptst/ELC_PLC.html; consultado el 06 de junio de 2016

² Precio del dólar el 06 de junio de 2016 es de \$ 18.68 MN



Por su parte, Herrador menciona en su libro *Guía de usuario de Arduino* que “Arduino es una plataforma de prototipos electrónica de código abierto (open-source) basada en hardware y software libres y fáciles de usar. Las placas se pueden ensamblar a mano o encargadas preensambladas, el software se puede descargar gratuitamente. Los diseños de referencia del hardware están disponibles bajo licencia open-source, por lo que eres libre de adaptarlas a tus necesidades” (Enríquez Herrador, 2009).

El entorno de Arduino ha sido diseñado para ser fácil de manejar y ha sido utilizado para crear una sorprendente variedad de cosas, incluyendo instrumentos musicales, robots, juegos, etc. La gente experta que trabaja con microcontroladores se ha sentido atraída por Arduino, gracias a la agilidad de desarrollo, la facilidad de implementación y la gran potencia que ofrece (Margolis, 2011).

Con los trabajos citados podemos concluir que las ventajas de las plataformas libres se distinguen por el bajo costo, la gran potencia que tienen para realizar proyectos electrónicos; las interfaces son lógicas, fáciles de manejar y la simplicidad de programación, por lo que la realización de proyectos de automatización en placas Arduino nos ahorran costos y agilizarán su desarrollo.

La plataforma Arduino es uno de los open-source más usados para proyectos electrónicos, ya que poseen más experiencia y gran adaptabilidad/compatibilidad con hardware externo además de la facilidad de desarrollar las rutinas aplicadas a los elementos de la automatización.

Control del clima en SITs

El control del clima en el cultivo es de vital importancia para la producción y calidad en las plantas. Los parametros del ambiente que se automatizaron para los bioreactores son el monitoreo de temperatura e intensidad de luz; cabe destacar que la luz puede variar dependiendo el tipo de planta, por lo que la implementación de sectores es un punto vital, ya que los SITs albergan mas de un tipo de cultivo.

Dentro de los SITs, se ha intentado automatizar solo el funcionamiento de los bioreactores, Del Rivero (2004) menciona en el artículo *Empleo de sistemas de inmersión temporal para la multiplicación in vitro de brotes de Anthurium andraeanum* que la presión del aire en los bioreactores eran controladas por un manómetro y el tiempo de inmersión era regulado por un software *Biosys*, el cual controla 2 válvulas eléctricas de tres pasos que permiten la circulación del aire en el bioreactor.

Además de regular el ambiente en el que se encuentra el cultivo, fue necesario tomar otros parametros como el tamaño, el color, e inclusive el aspecto del bioreactor para supervisar y monitorear el estado del cultivo, por ello, se pretende construir un sistema de monitoreo en tiempo real que pueda ser consultado por medio de una imagen; de esta forma, el usuario podrá deducir si las condiciones del cultivo son idoneas o si es necesario regular algun elemento del SIT.

Cabe mencionar que durante la revisión bibliográfica de los antecedentes no se encontró ningun proyecto similar al que se propone en este escrito, que plantea todo un sistema de automatización y monitoreo de un SIT, lo que representa que este un proyecto novedoso e ideal para realizar.



Campo de estudio: COLPOS

El COLPOS es una institución educativa, dedicada a la enseñanza e investigación en Ciencias Agrícolas³, que está ubicada a un costado de la carretera México- Texcoco km 36.5, Montecillo, Texcoco en el Estado de México, con código postal 56230. En la *ilustración 5* se muestra la ubicación del COLPOS en un mapa.

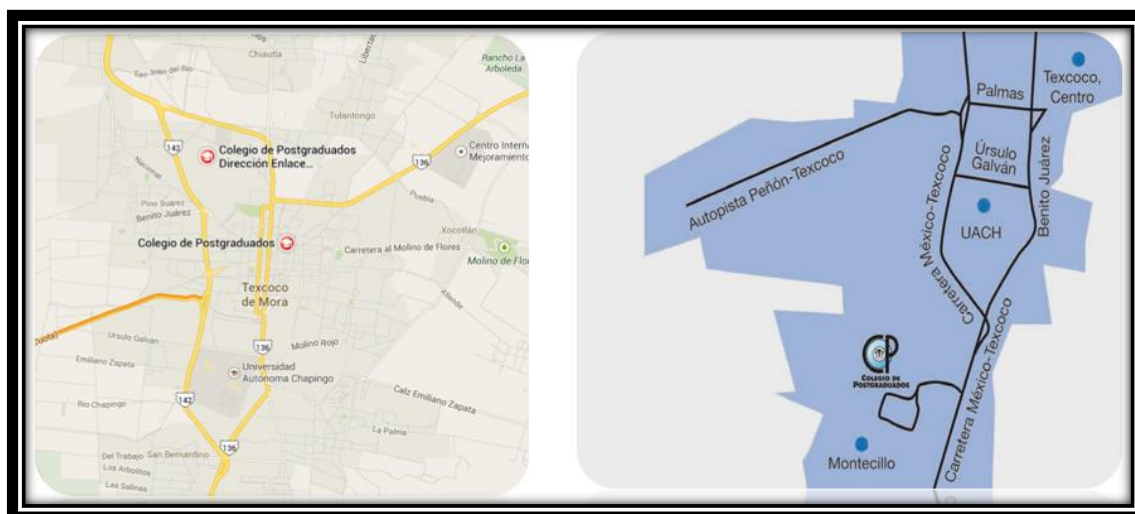


ILUSTRACIÓN 5. UBICACIÓN DEL COLPOS

El COLPOS es considerada una de las instituciones mexicanas más importantes en materia de educación, investigación y vinculación en ciencias agrícolas con un número alto de estudiantes inscritos a nivel maestría con un 71% de los alumnos que ingresan respecto al doctorado que cuenta con un 28.6%, institución dedicada a estudios avanzados y a una parte importante de la investigación relativa a agronomía en el país. Fundada en el año 1959 como parte de la Escuela Nacional para Agricultura, conocida el día de hoy como Universidad Autónoma Chapingo (COLPOS, 2012).

³ Las ciencias agrícolas son definidas como un conjunto de conocimientos aplicables al cultivo de la tierra, además de comprenderse como un derivado de las ciencias exactas, físicas y económicas (Samper, 1969).

En el año 1979 el Colegio se convirtió, por decreto presidencial, en un organismo público y descentralizado del gobierno federal con una personalidad jurídica y un patrimonio propio sectorizado en la Secretaría de Agricultura, Ganadería, Desarrollo Rural, Pesca y Alimentación (SAGARPA).

Hacia el año de 1979, el decreto de creación del mismo año confirmó al Colegio la tarea de impartir enseñanza a nivel postgrado, así como de realizar investigaciones y llevar a cabo servicios, así como asistencia técnica en materia agropecuaria y forestal.

De esta manera el Colegio “enseña a investigar e investiga para enseñar en un contexto de vinculación integral con su entorno bajo un sistema de Campus



ILUSTRACIÓN 6. COLPOS EDIFICIO PRINCIPAL

localizados en Campeche, Campeche; Montecillo, Estado de México; Puebla, Puebla; San Luis Potosí, SLP; Córdoba, Veracruz; Cárdenas, Tabasco, y Tepetates, Veracruz” (COLPOS, 2012).

En el año 2001 el Colegio fue reconocido por la SAGARPA y por el Consejo Nacional de Ciencia y Tecnología (CONACYT) como un Centro Público de Investigación y realizó un Convenio de Desempeño con la Administración Pública Federal, cuyas



metas multianuales están relacionadas con sus actividades sustantivas calificadas por un Comité de Evaluación Externa.

Cabe destacar que con la finalidad de hacer del Colegio de Postgraduados una institución innovadora y de progreso evidente, capaz de desarrollarse y transformarse con los cambios sociales actuales, se aprobó en el año 2004 una Reestructuración Integral para la Modernización de la Institución, que contempla la innovación y renovación de la institución en materia agronómica, económica, social y cultural, entre otros aspectos. En este sentido, el centro educativo ha desarrollado, investigaciones y proyectos acordes a este fin, priorizando el desarrollo tecnológico y la innovación para llevar a cabo la modernización del Colegio.



ILUSTRACIÓN 7. DEPARTAMENTO DE BIOTECNOLOGÍA DE FRUTALES

Con la finalidad de cumplir con la misión, la visión y los objetivos estratégicos institucionales, el COLPOS desarrolla estrategias y proyectos como la creación de diversos sectores de biorreactores que contemplan el uso de SITs para cumplir el objetivo de producción y reproducción de especies, mediante el impulso de la tecnología para mejorar los resultados de las investigaciones.

Capítulo I: Marco Teórico

1.1 Cultivo *in vitro*

La reproducción *in vitro* o micro propagación vegetal es empleada en cultivos de importancia económica, de resguardo de especies nativas o en peligro de extinción, aportando grandes logros en materia de agricultura. El término *in vitro* refiere a las técnicas que se llevan a cabo dentro de laboratorios, que permiten la unión de las células sexuales, óvulos y espermatozoides en recipientes de vidrio, de ahí *in vitro*, en vidrio.

En el año 1902 se realizaron los primeros experimentos relacionados con el cultivo de tejidos vegetales dentro de recipientes de vidrio, *in vitro*, y fue en el año 1922 que se llevó a cabo con éxito el primer experimento de germinación *in vitro* de semillas de orquídeas. Después de la germinación las plántulas⁴ obtenidas fueron transferidas a un medio de cultivo aséptico, para ser protegidas hasta convertirse en plantas adultas (Frid, 2009).

En este sentido el cultivo *in vitro*, incluye diversas técnicas y métodos diseñados con la intención de multiplicar material vegetal, que consiste en tomar una porción de la planta, que se nombra *explante* y se le proporciona de manera artificial un medio de cultivo nutritivo y esterilizado que permite regenerar un número variado de plantas.

Existen diversos motivos por los que la reproducción *in vitro* se convierte en un factor importante como son la propagación masiva de plantas en peligro de extinción o difíciles de propagar, clonación de especies con características importantes como: especies nativas o especies con características deseables como: mejores frutos, resistentes a las sequias, mayor producción de frutos, resistentes a ciertas plagas,

⁴ Nombre que se otorga a la parte del desarrollo de la planta en la que la semilla sale de su dormancia y germina y termina cuando el esporófito desarrolla sus primeras hojas (Philippi, 2012).



etc., así como para obtener plantas libres de virus y la conservación de la diversidad genéticas de una población (Frid, 2009).

1.2 SIT: Sistema de Inmersión Temporal

Las técnicas biotecnológicas han tomado un mayor auge en la actualidad debido a que, hasta el momento, los sistemas de propagación tradicional no satisfacen las diversas demandas para muchas especies vegetales. Así mismo, existen especies endémicas y en peligro de extinción que es necesario resguardar y reproducir, empleando estas técnicas biotecnológicas como vía alternativa para su rescate y conservación. Sin embargo, todo el avance tecnológico y biotecnológico desarrollado en esa área todavía no ha cumplido con las expectativas existentes para algunos cultivos, por lo que se han buscado variantes que permitan la creación de una herramienta completa para lograr reproducir un mayor número de plantas sanas y con capacidad de desarrollarse en condiciones de campo.

En este sentido, las técnicas biotecnológicas y en específico el cultivo *in vitro* como un método efectivo para lograr el objetivo mencionado, en cuanto a que permite lograr un mayor número de plantas con menos costos, sin embargo esta técnica aún puede mejorarse para obtener mejores resultados, reduciendo costos y simplificando el cuidado y supervisión de los biorreactores, por lo que se ha trabajado a lo largo de los años en la automatización de la micro propagación, ya sea de forma completa o parcial, lo que significa un gran avance en cuanto al mantenimiento control y producción de cultivos *in vitro* (Rosales, 2009).

Una alternativa en cuanto a la micropropagación⁵ de plantas, es el uso del SIT, basado en contacto temporal del medio de cultivo con los explantes, lo que genera un aumento de la productividad del material propagado. En este sentido el cultivo en inmersión temporal puede constituir una alternativa de reproducción de embriones

⁵ Nombre que recibe la técnica de propagación vegetativa de plantas por medio de condiciones *in vitro* (Paz, 1985).

en un periodo corto de tiempo y a largo plazo es posible obtener plantas y embriones sanos y con un buen desarrollo (Santos *et al.*, 2011).

Este sistema está formado por (Cabrera, 2004) dos frascos de vidrio o biorreactores, uno empleado para el crecimiento de explantes y uno más dedicado a mantener el medio del cultivo líquido. Estos frascos están conectados a través de orificios en las tapas con una manguera de silicón auto clavable. En el interior se coloca otra tubería similar con puntas de micro pipeta que sirven para que el medio de cultivo circule de un frasco a otro, dependiendo de la apertura o cierre de las electroválvulas encaradas de controlar el flujo de aire, en la *ilustración 8* se muestran los biorreactores utilizados en algunos SIT, cortesía del COLPOS.



ILUSTRACIÓN 8 BIORREACTORES EN ALGUNOS SIT, CORTESÍA DEL COLPOS

Para llevar a cabo un método de micro propagación eficiente en sistemas de inmersión es de suma importancia la optimización de los parámetros técnicos en cuanto al desarrollo de cada cultivo. Los parámetros cuya importancia es mayor son el tiempo y la frecuencia de inmersión y es debido a su repercusión en la calidad y desarrollo del cultivo que la optimización de estas variables se convierte en un factor fundamental en cuanto a la producción en SITs (Vilchez, et al., 2011).

Cabe destacar que (Mulet, 1999) la eficiencia de un proceso, independientemente de su naturaleza, depende del control de sus variables fundamentales, las cuales se



definen como condiciones que están sujetas a cambios, ya sea en los materiales del proceso o en los equipos, y por esto el mantenimiento del control del mismo en su totalidad es parte muy importante del diseño, en este caso el control de las variables permitirá que el SIT funcione en las mejores condiciones.

Tras notar la importancia del manejo y control de las variables del SIT, resulta importante el desarrollo de una herramienta que permita dicho control. Como se ha mencionado a lo largo del presente escrito, esto se pretende por medio de la automatización del bombeo de nutrientes, así como el control de intensidad y color de la tira de *diodos emisor de luz* (led, por sus siglas en inglés) que proporciona luz a los biorreactores. Así como el resguardo de los datos relativos a la temperatura, cambios de luz y tiempos de bombeo, lo que permitirá un conocimiento general y profundo del desarrollo de los embriones dentro del SIT, facilitando de esta manera el reporte de resultados sobre los mismos.

1.3 Automatización

La automatización se define como “el uso de sistemas o elementos computarizados y electromecánicos para controlar maquinarias o procesos”. En este sentido, se comprende al proceso de automatizar como aquel que permite controlar determinadas funciones de maquinarias o procesos específicos, mediante la aplicación de sistemas mecánicos, electrónicos y de bases computacionales, logrando así facilitar la tarea (Martínez, 2001).

En el caso del presente trabajo, la automatización se realizó para controlar la función de bombeo de nutrientes, el cambio e intensidad de luz y el resguardo de datos de manera autónoma mediante el empleo de una computadora, la autonomía de la automatización en este caso, está relacionada con el empleo de un equipo electrónico que permite que el programa regule las funciones mencionadas del SIT y que se resguarden los datos en el mismo equipo.

1.3.1 Monitorización

La monitorización se refiere a la evaluación continua de una acción específica que se está llevando a cabo, es decir, de una acción que se encuentra en desarrollo. Existe determinados programas u organizaciones avanzadas que requieren de sistemas de monitoreo, en el caso de que la información o datos recabados sean de una importancia específica. La monitorización de un proceso permite la recolección, almacenamiento y documentación de los datos con fines de análisis o evaluación.

El monitoreo tanto de los datos como del equipo de automatización que se implementó en el SIT, no sólo sirve para verificar el correcto funcionamiento de las variables de temperatura, luminosidad, bombeo de nutrientes, etc., sino que se mantendrá el equipo de automatización en buenas condiciones, lo que conlleva a la correcta recolección de datos y un sano desarrollo del material vegetativo contenido en el SIT.

Las aplicaciones móviles son importantes hoy en día, por lo que el sistema de monitoreo para celulares Android es de gran ayuda ya que se puede lograr una monitorización remota, así mismo, pedir una imagen en tiempo real del SIT ayuda al usuario verificar su cultivo sin necesidad de estar presente en el laboratorio.

1.4 Control multiplataforma

Dentro de este proyecto se diseñaron dos aplicaciones que permiten el control de dispositivos usados en la automatización del SIT, por ejemplo, encendido o apagado de una bomba de aire, encendido o apagado de luces led, etc. Un sistema de control es “un arreglo de componentes físicos conectados de tal manera que el arreglo pueda comandar, dirigir o regular, asimismo o a otro sistema. Estos sistemas comandan, dirigen o controlan dinámicamente” (Alberto, Perez y Perez, 2007).

Un sistema de control se compone de objetivos de control (entradas o señales actuantes), componentes del sistema de control y de resultados (salidas o variables

controladas); en general, el objetivo de un sistema de control es regular las salidas en alguna forma prescrita mediante las entradas a través de los elementos del sistema (Kuo, 1996).

Como mencionan Ramírez y Rosero (2007) existen dos tipos de sistema de control:

- De control en lazo o bucle abierto: acción de control independiente de la salida; para su óptimo desempeño se requiere de una correcta calibración; si el proceso a controlar es estable no hay riesgo de mal funcionamiento, en la *ilustración 9* se muestra un diagrama de bloques representando este tipo se

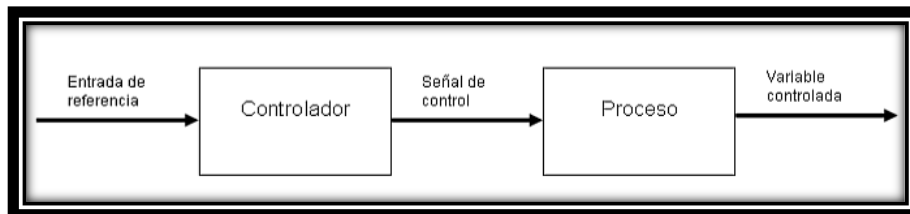


ILUSTRACIÓN 9 DIAGRAMA DE BLOQUES DEL SISTEMA DE CONTROL ABIERTO

sistema.

- De control en lazo o bucle cerrado: se compara la entrada y la salida, empleando la diferencia o error como acción de control; se requiere por tanto de una realimentación que genera posibilidad de inestabilidad, en la *ilustración 10* se muestra un diagrama de bloques representando este tipo se sistema.

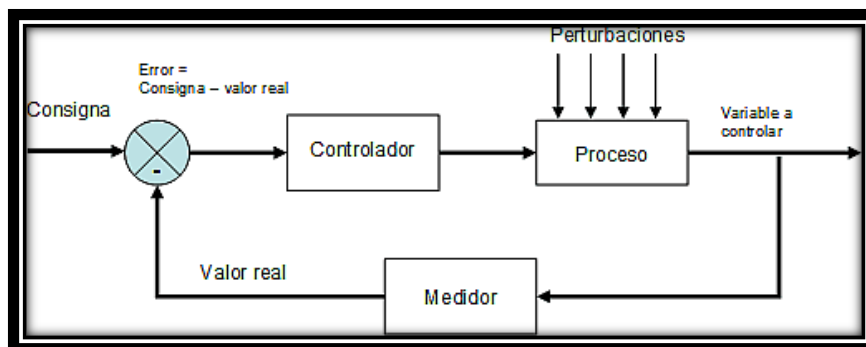


ILUSTRACIÓN 10 DIAGRAMA DE BLOQUES DEL SISTEMA DE CONTROL CERRADO

Para el presente proyecto de investigación se diseñaron dos aplicaciones: una para computadora personal (PC por sus siglas en inglés) y otra para dispositivos celulares con sistema operativo Android, las cuales envían una entrada a los componentes de la automatización haciendo que se produzca una salida. Las aplicaciones usan el control en lazo abierto.



Capítulo II: Materiales en la automatización

En el presente proyecto se utilizaron varios elementos para realizar la automatización, se implementaron herramientas y materiales de hardware, así como de software para la creación y desarrollo de dispositivos y programas que llevan a cabo el funcionamiento de la automatización realizada.

2.1 Plataforma de hardware libre Arduino

En este proyecto se utilizó la plataforma de hardware libre Arduino que es una placa de hardware libre que incorpora un microcontrolador reprogramable y una serie de pines hembra (unidos internamente a las patillas de entrada y salida del microcontrolador) que permiten conectar de forma muy sencilla y cómoda diferentes sensores y actuadores (Torrente, 2013).

Como lo explica Enríquez Herrador (Enríquez Herrador, 2009) existen muchos otros controladores y plataformas de hardware libre (como Raspberry o Beagle Bone) que encierran la programación de microcontroladores en un paquete fácil de usar; sin embargo, Arduino también simplifica este proceso ofreciendo ventajas sobre las otras plataformas por lo que se ha elegido para el desarrollo de la de la automatización del SIT. A continuación, se muestran las ventajas de Arduino:

- Bajo costo: La creación de una placa Arduino a mano propia o pre ensamblada tiene un costo menor a \$500.
- Multiplataforma: El software de Arduino es ejecutable en diferentes sistemas operativos como lo son Windows, Macintosh OSX y GNU/Linux.
- Entorno de programación: Es simple y claro además que contiene gran flexibilidad de aprovechamiento.

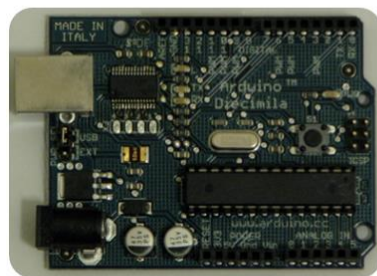
- Open-source y software extensible: El lenguaje del software de Arduino puede ser expandido mediante librerías en lenguaje C++⁶, ya que se encuentra basado en C⁷, ofreciendo la opción de añadir código C directamente en el programa.
- Hardware extensible: Arduino está basado en microcontroladores ATmega de Atmel, son microcontroladores de bajo costo y gran potencia; al ser Hardware abierto, puede ser creado en él una versión propia del módulo, extendiéndolo o mejorándolo, ahorrando dinero y logrando entender cómo funciona la placa Arduino.

2.1.1 Tipos de placas Arduino

ARDUINO DIECIMILIA

Arduino Diecimilia es una placa de tipo Arduino USB estándar. En la *tabla 1* se muestra la placa Diecimilia y sus características.

TABLA 1. ESPECIFICACIONES DIECIMILIA. FUENTE: WWW.ARDUINO.CC



Características	Arduino Diecimilia
Micro controlador	ATMega168
Voltaje operativo	5 V
Voltaje de entrada	7 V – 12 V
Entradas digitales	14
Entradas análogas	6
Memoria Flash	16 KB – 2 KB de bootloader
SRAM ⁸	1 KB
EEPROM	512 bytes
Velocidad de reloj	16 MHz

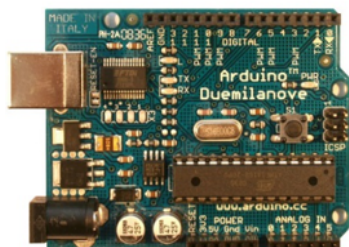
⁶ Lenguaje de programación orientado a objetos; es la evolución del lenguaje C más clases. Está formado por instrucciones explícitas y corta (Osorio Rivera, 2007).

⁷ Lenguaje de programación de propósito general. Su lenguaje es de medio nivel, es decir, muy cercano al lenguaje máquina (Martínez Gil & Martín Quetglás, 2003).

ARDUINO DUEMILANOVE

Esta placa de Arduino puesta a disposición en el 2009 pertenece a la versión Arduino USB estándar y es la evolución de la placa Diecimilia, esta placa Arduino fue la primera placa en la que el microcontrolador podía ser intercambiado entre un ATmega168 y un ATmega328, siendo esta última usada para proyectos de mayor potencia. En la *tabla 2* se muestra la placa Duemilanove así como una tabla mostrando sus características.

TABLA 2. ESPECIFICACIONES DUEMILANOVE. FUENTE: WWW.ARDUINO.CC/



Características	Arduino Duemilanove
Micro controlador	ATMega168 – ATMega328
Voltaje operativo	5 V
Voltaje de entrada	7 V – 12 V
Entradas digitales	14
Entradas análogas	6
Memoria Flash	16 KB (ATMega168) 32 KB (ATMega328) – 2 KB de bootloader
SRAM	1 KB (ATMega 168) 2 KB (ATMega 328)
EEPROM	512 bytes (ATMega 168) 1 KB (ATMega 328)
Velocidad de reloj	16 MHz

ARDUINO MEGA 2560

<https://www.arduino.cc/en/Main/ArduinoBoardDuemilanove> Arduino Mega 2560 es una placa de tipo Arduino Mega, teniendo mayor espacio en memoria, mayor cantidad de entradas/salidas y mayor capacidad de procesamiento que una Arduino USB estándar. En la *tabla 3* se muestra a la placa Mega 2560 y su tabla de característica

TABLA 3. ESPECIFICACIONES MEGA 2560. FUENTE: WWW.ARDUINO.CC



Características	Arduino Mega 2560
Micro controlador	ATMega2560
Voltaje operativo	5 V
Voltaje de entrada	7 V – 12 V
Entradas digitales	54
Entradas análogas	16
Memoria Flash	256 KB – 8 KB de bootloader
SRAM	8 KB
EEPROM	4KB
Velocidad de reloj	16 MHz

ARDUINO DUE

Arduino DUE es una placa de tipo Arduino Mega, es la primera placa Arduino basada en una arquitectura de procesamiento de 32 bits del microcontrolador. En la *tabla 4* se muestra a la placa DUE, y su tabla de características.

TABLA 4. ESPECIFICACIONES DUE. FUENTE: WWW.ANDROID.CC



Características	Arduino DUE
Micro controlador	AT91SAM3X8E
Voltaje operativo	3.3 V
Voltaje de entrada	7 V – 12 V
Entradas digitales	54
Entradas análogas	14
Memoria Flash	512 KB
SRAM	96 KB (64 KB y 32 KB)
Velocidad de reloj	84 MHz



2.1.2 Placa Arduino empleada para la automatización del SIT

Para llevar a cabo el proyecto de automatización de SIT, fue necesario usar dos placas Arduino que fueron capaces de soportar todos los procesos de automatización relacionados con los dispositivos a controlar y con el sistema de comunicación remota. Arduino ofrece distintas placas con diferentes capacidades, la necesidad de mayor memoria y más SRAM para las librerías que se integraran en el proyecto nos llevó a la opción de adquirir dos placas del modelo Arduino UNO.

La elección de la placa Arduino modelo UNO se basa en las características que se muestran en la *tabla 5*.

TABLA 5. ESPECIFICACIONES UNO. FUENTE: WWW.ADUINO.CC

Características	Arduino UNO
Micro controlador	ATmega328
Voltaje operativo	5 V
Voltaje de entrada	7 V – 12 V
Entradas digitales	14
Entradas análogas	6
Memoria Flash	32 KB – 0.5 KB de bootloader
SRAM	2 KB
EEPROM	1 KB
Velocidad de reloj	16 MHz

Este modelo de Arduino, al igual que sus predecesoras, cuenta con un plug de alimentación externo de 7 V a 12 V. En la *ilustración 11* se muestra la placa Arduino UNO, así como los elementos que lo componen.

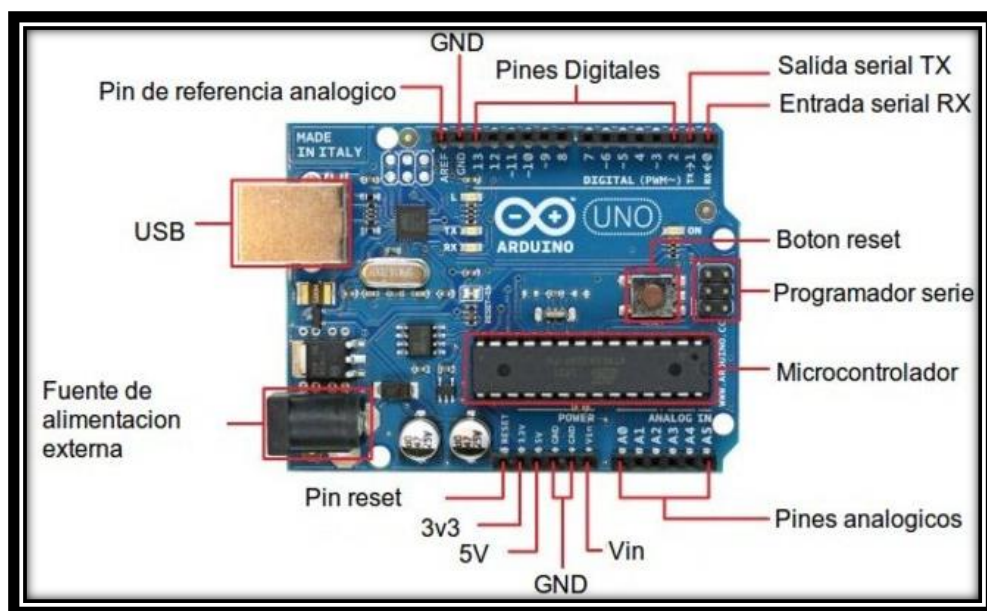


ILUSTRACIÓN 11. ARDUINO UNO R3

2.1.3 Programación en Arduino

En la página oficial de Arduino⁸ indica que los programas hechos en esta plataforma se dividen en tres partes principales: estructura, valores (variables y constantes), y funciones. El Lenguaje de programación Arduino se basa en C/C++ que son lenguajes de medio nivel, usualmente utilizados en la programación de microcontroladores por medio de un entorno de desarrollo.

La estructura de un programa en Arduino (Pomares Baeza, 2009) es bastante simple y divide la ejecución en dos partes: `setup()` (inicialización) y `loop()` (bucle); `setup()` constituye la preparación del programa donde se incluye la declaración de variables

⁸ Página WEB oficial de Arduino: <http://www.arduino.cc/es/>



y es la primera función que se ejecuta en un programa, esta función se ejecuta una sola vez y es empleada para indicar si un pin (digital o analógico) es de entrada o salida; la función `loop()` incluye el código a ser ejecutado continuamente. En la *Ilustración 12* se muestra un ejemplo de estas funciones.

```
int ledPin=13; //LED en PIN digital 13

void setup(){
  pinMode(ledPin, OUTPUT); //El pin 13 del LED se decalra como salida
}

void loop(){
  digitalWrite(ledPin, HIGH); //Activa el pin 13 o LED
  delay(1000); //Pausa de 1 segundo
  digitalWrite(ledPin,LOW); //Desactiva el pin 13 o LED
  delay(500); //Pausa de 0.5 segundos
}
```

ILUSTRACIÓN 12. FUNCIONES VOID() Y SETUP()

Una función se describe como “un bloque de código identificado por un nombre y que es ejecutado cuando la función es llamada” (Martínez Gil & Martín Quetglás, 2003), en la declaración de la función incluye el tipo de variable que devuelve la función y los parámetros de la misma. En la *ilustración 13* se muestra una función que en la declaración indica que retorno una variable entera.

Las variables de un programa deben ser declaradas y opcionalmente asignadas a un determinado valor, en la declaración de la variable se indica el tipo de dato que almacenará. Los tipos de datos para las variables en Arduino (Banzi, 2009) son:

- Byte: Almacena un valor numérico de 8 bits. Tienen un rango de 0-255.
- Int: Almacena un valor entero de 16 bits con un rango de 2,147,483,647 a -2,147,483,648.
- Long: Valor entero almacenado en 32 bits.

- Float: Tipo coma flotante almacenado en 32 bits, usado para números con punto decimal.
- Array[]: Colección de valores que pueden ser accedidos con un número de índice (el primer valor es 0).
- String: tipo de dato que almacena una cadena de caracteres, como números o palabras.

```

int ledPin=13; //LED en PIN digital 13
int f=0; //Se declara la variable f como entero con valor 0
int x=0; //Se declara la variable x como entero con valor 0
void setup(){
  pinMode(ledPin, OUTPUT); //El pin 13 del LED se declara como salida
}

void loop(){
  x=funcion() //Llamada a funcion(), el dato retornado se guarda en x
  if(x==1){ //Condición
    digitalWrite(ledPin, HIGH); //El pin 13 o LED se activa
  }
  else{ //Si no se cumple la condición
    digitalWrite(ledPin, LOW); //El pin 13 o LOD se desactiva
  }
  delay(1000); //Espera de 1 segundo
}

int funcion(){ //Declaración de la función
  if (digitalRead(ledPin==LOW)){ //Se lee el estado del pin 13 o LED
    f=1; //f toma el valor de 1
  }
  else{ //Si no se cumple la condición
    f=0; //f toma el valor de 0
  }
  return f; //Variable que retorna la función
}

```

ILUSTRACIÓN 13. FUNCIÓN DE RETORNO CON UNA VARIABLE

Los operadores aritméticos usados en Arduino se utilizan de forma similar a las funciones correspondientes en el lenguaje C, sin embargo, Arduino presenta las siguientes constantes predefinidas:



- HIGH/LOW: Estas constantes definen los niveles de los pines, donde HIGH se define como el nivel lógico 1 (encendido) o 5 V y LOW es nivel lógico 0 (apagado) o 0 V.
- INPUT/OUTPUT: Constantes empleadas con la función pinMode() para definir el tipo de un pin digital usando como entrada INPUT o salida OUTPUT, ejemplo:
 - pinMode (13, OUTPUT): El pin digital 13 es declarado como pin de salida.

Las sentencias condicionales que se manejan en Arduino son usados de manera idéntica al lenguaje C y son: if (), else if (), else, for (), while (), do () while.

Arduino maneja funciones predefinidas para el uso de las entradas de la placa, las más importantes son:

- ✓ pinMode (pin, modo): Función usada en setup () para configurar un pin dado para comportarse como INPUT (entrada) u OUTPUT (salida).
- ✓ digitalRead (pin): lee el valor de un pin digital específico. Devuelve un valor HIGH o LOW.
- ✓ digitalWrite (pin, valor): puede activar (HIGH) o desactivar (LOW) el pin indicado.
- ✓ analogRead (pin): Lee el valor desde el pin analógico especificado en una resolución de 10 bits. El valor resultante es un entero de 0 a 1023.

En la *ilustración 14* se muestra un programa utilizando algunas funciones de Arduino, a la derecha se ubica la consola serial, el cual proyecta 1 (encendido) y 0 (apagado) del led.

El lenguaje de desarrollo de Arduino (Enríquez Herrador, 2009) maneja una función de tiempo llamada delay(ms), el cual realiza una pausa en el programa la cantidad de

tiempo en milisegundos, así mismo, Arduino maneja funciones de puerto serial de la placa y estas son:

- `Serial.begin (rate)`. Abre un puerto serie y especifica la velocidad de transmisión. La velocidad típica para la comunicación con el ordenador es de 9600.
- `Serial.println (dato)`: Imprime datos al puerto serie, seguido por un retorno de línea automático, dichos datos pueden ser visualizados en un monitor serial del IDE donde se esté programando.
- `Serial.read ()`: lee o captura un byte (un carácter) desde el puerto serie.
- `Serial.available ()`. Devuelve el número de caracteres disponibles para leer desde el puerto serie.

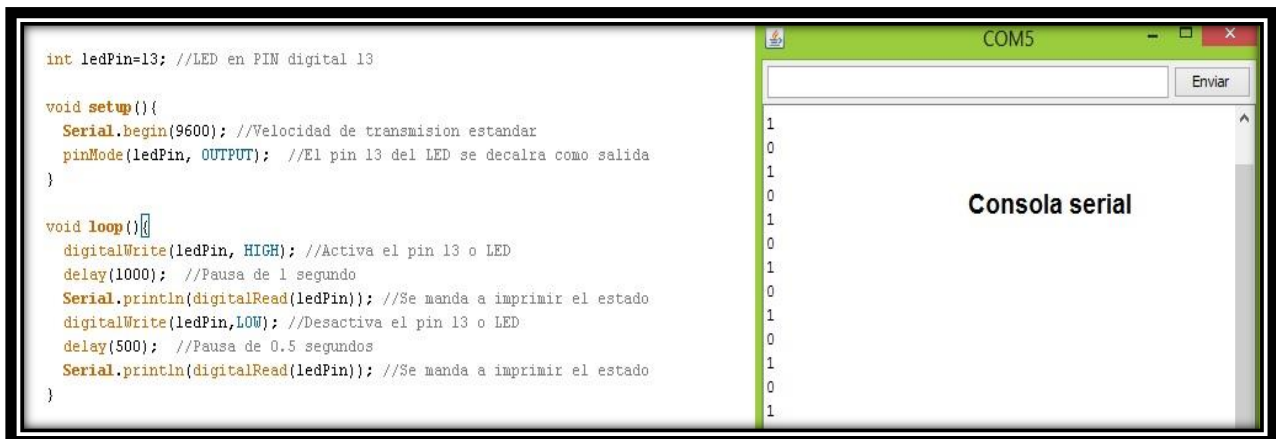


ILUSTRACIÓN 14. FUNCIONES DE ARDUINO Y CONSOLA SERIAL

2.1.4 Entorno de desarrollo integrado de Arduino

Los programas en Arduino llamados sketches, son creados en una computadora usando un entorno de desarrollo integrado de Arduino (IDE), que permite escribir y editar un código y convertirlo en instrucciones que el hardware de Arduino entiende y



transfiere las instrucciones del sketch a la placa Arduino, en un proceso llamado “cargar” o “uploading” (Margolis, 2011).

El entorno de desarrollo integrado (IDE) (Banzi, 2009), es un programa especial que se ejecuta en la computadora que permite escribir Sketches para la placa Arduino en un simple lenguaje modelado mediante procesamiento (Processing). El código que se escribe en el IDE es traducido al lenguaje C y es pasado a un compilador avr-gcc el cual convierte el sketch de lenguaje C a un código binario que solo es entendible por el microcontrolador.

El ciclo de programación de Arduino es básicamente el siguiente:

- Conecta la placa Arduino al puerto USB de la computadora.
- Escribir el Sketch que llevará la placa Arduino.
- Carga el sketch a la placa mediante la conexión USB y espera un par de segundos para el reinicio del Arduino.
- La placa ejecuta el sketch que se ha escrito.

El IDE de Arduino es creado en Java⁹, la integración de una placa de Arduino al IDE se hace mediante la instalación del driver y configurando un puerto COM (de comunicación) en la computadora y está disponible de forma gratuita en su página oficial: <http://arduino.cc/es/Main/Software>.

En la *ilustración 15* se muestra el entorno de desarrollo, así como los elementos con los que cuenta.

⁹ Lenguaje de alto nivel orientado a objetos, de propósito general, concurrente y basado en clases, es sencillo, fácil de usar y potente (Osorio Rivera, 2007).

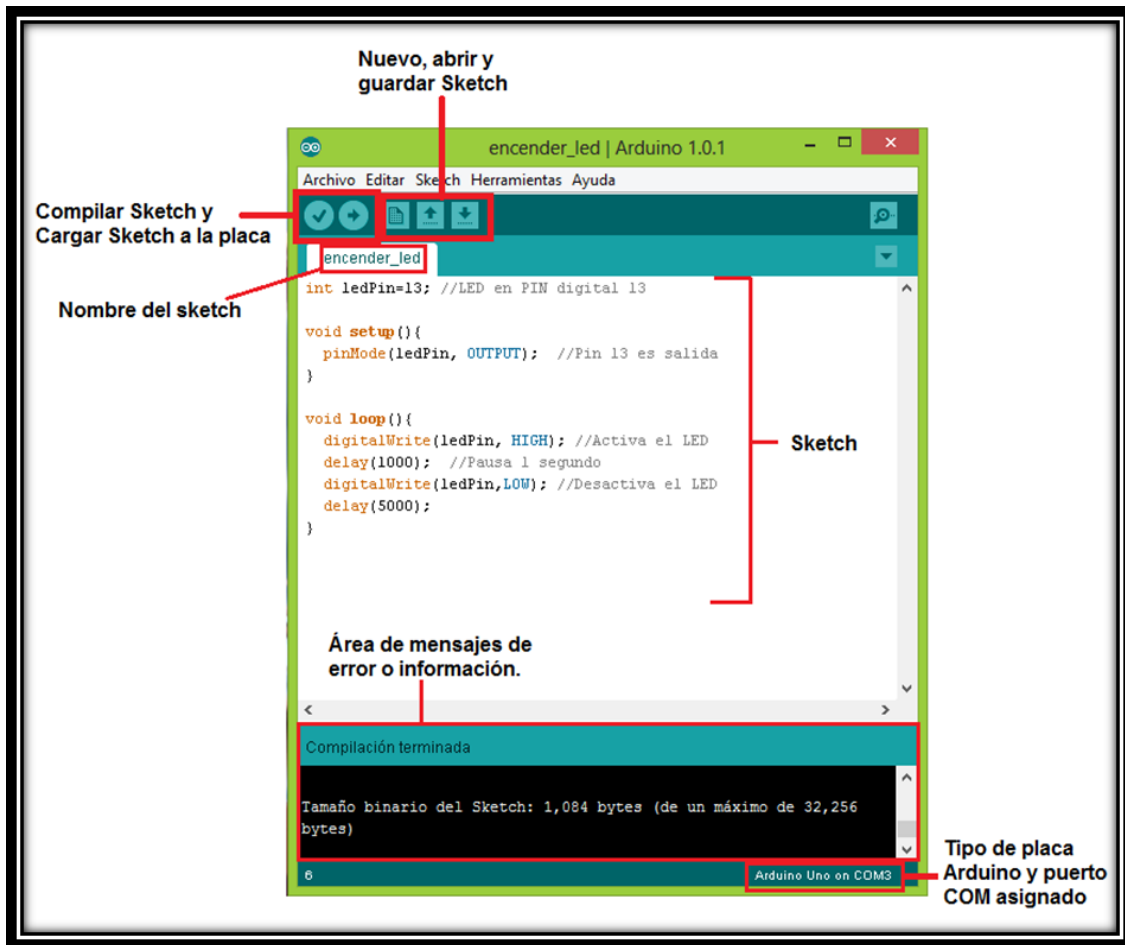


ILUSTRACIÓN 15. IDE DE ARDUINO

2.2 Sensores

Un sensor es “el componente de una instalación o sistema, que se encarga de recibir el valor de una magnitud preferentemente no eléctrica para convertirla en una señal eléctrica de fácil aplicación en un sistema más complejo” (Sobrevila & Sobrevila, 1998), es decir, un componente electrónico que permite la detección y obtención de datos, los transforma en otra magnitud, que por lo general es eléctrica y se convierte en una serie de datos posibles de cuantificar y manipular.



Los sensores que fueron empleados en el presente proyecto se enlistan a continuación:

- Fotorresistencia: Es un elemento resistivo cuya característica principal es que la magnitud de su resistencia es una función de la luz incidente, es decir, es un transductor entre la luz incidente en este y la resistencia eléctrica que presenta a la salida (Corona Ramírez, Abarca Jiménez, & Mares Carreño, 2014); en la *ilustración 16* se muestra la resistencia que se implementó en la automatización del SIT.



ILUSTRACIÓN 16. FOTORRESISTENCIA

En la automatización del SIT se implementaron fotorresistencias (una en cada sector) para monitorear el estado de la luz, es decir si existe o no iluminación cuando es requerido por el usuario ya sea ocasionada por la tira de led o por iluminación externa.

- Sensor de temperatura: Estos sensores transforman los cambios de temperatura en cambios de señales eléctricas que son procesados por un equipo electrónico o eléctrico; típicamente suele estar formado por el elemento sensor, la vaina que lo envuelve y que está rellena de un material conductor de la temperatura.

El modelo del sensor digital de temperatura que se implementó en este proyecto es el DS18B20, el cual detecta temperaturas desde los $-55\text{ }^{\circ}\text{C}$ a $125\text{ }^{\circ}\text{C}$ con un margen de error de $\pm 0.5\text{ }^{\circ}\text{C}$, su consumo eléctrico es de 5 V y posee solo 3 pines de conexión (voltaje, tierra y datos); por estas características, este sensor es ideal para

detectar la temperatura del SIT y ofrecer mediante software el valor obtenido del dispositivo. En la *ilustración 17* se muestra el sensor de temperatura DS18B20.



ILUSTRACIÓN 17. SENSOR DE TEMPERATURA

2.3 Dispositivos electrónicos controlados

Además del uso de sensores en la automatización del SIT, también se implementaron otros dispositivos electrónicos, los cuales son controlados por las aplicaciones diseñadas para este proyecto. A continuación, se enlistan los dispositivos que son controlados por medio del software.

2.3.1 Tira de led

En el SIT se requiere de cambios de luz entre colores, encendido y apagado para el beneficio del cultivo, para esto se usaron tiras de led RGB (Red, Green and Blue) modelo 5050. Como lo mencionan Gago y Fraile (2012) el led es un dispositivo semiconductor que emite luz con una longitud de onda monocromática específica muy



bien definida cuando se polariza de forma directa, permitiendo el paso de una corriente eléctrica entre sus dos extremos. Las características de la tira de led 5050 son:

- ✓ Alimentación de 12 V y 2 A (cada 5 metros)
- ✓ 30 ledes por metro
- ✓ Consumo de energía total por metro de 4.8 Watts
- ✓ Lúmenes: 2160 Lm (en blanco)
- ✓ Horas de vida: 50 000 horas aproximadamente
- ✓ Tamaño de led: 5mm x 5mm
- ✓ Puede iluminar en diferentes colores

Las propiedades de la tira de led RGB 5050 son ideales para ser implementadas en la automatización del SIT, ya que ofrecen una alta iluminación, un bajo consumo eléctrico y puede ser controlado el color de luz que emite la tira, siendo que las plantas se benefician de los colores blanco, azul y rojo.

Dado que una tira de led se alimenta de 12 V y 2 A es importante el uso de una fuente eléctrica que genere el poder necesario para su activación y en el caso de necesitar más de una tira de led, que la fuente proporcione la corriente necesitada; para este proyecto se usó una fuente de poder que genera dos salidas, cada una de 12 V (regulables) y 10 A por lo que esta fuente puede soportar hasta un máximo de 10 tiras de led (50 m).

El cambio de color en una tira de led es importante para el desarrollo del cultivo, sin embargo, una tira de led tiene 4 entradas, una que representa el color rojo, 1 el color verde, 1 el color azul y una entrada de voltaje a 12 V; para que los ledes enciendan un color, uno o más de los primeros 3 pines deben ser enviados a la señal de tierra, por lo tanto fue necesario de un dispositivo llamado transistor, con el que se logró enviar a tierra un pin a partir de una señal de entrada enviada desde un Arduino.

Un transistor es (Tipler & Mosca, 2005) un dispositivo semiconductor utilizado para producir una señal de salida en respuesta a otra señal de entrada, consta de tres

regiones semiconductoras distintas llamadas emisor, base y colector. Para que la tira de led cambie de color fue necesario el uso de transistores BD135, la base del transistor está conectada a la entrada de color de la tira led, el emisor a la señal de la placa Arduino y el colector está conectada a la fuente de poder. En la *ilustración 18* se muestra el esquema de conexión de la tira de led.

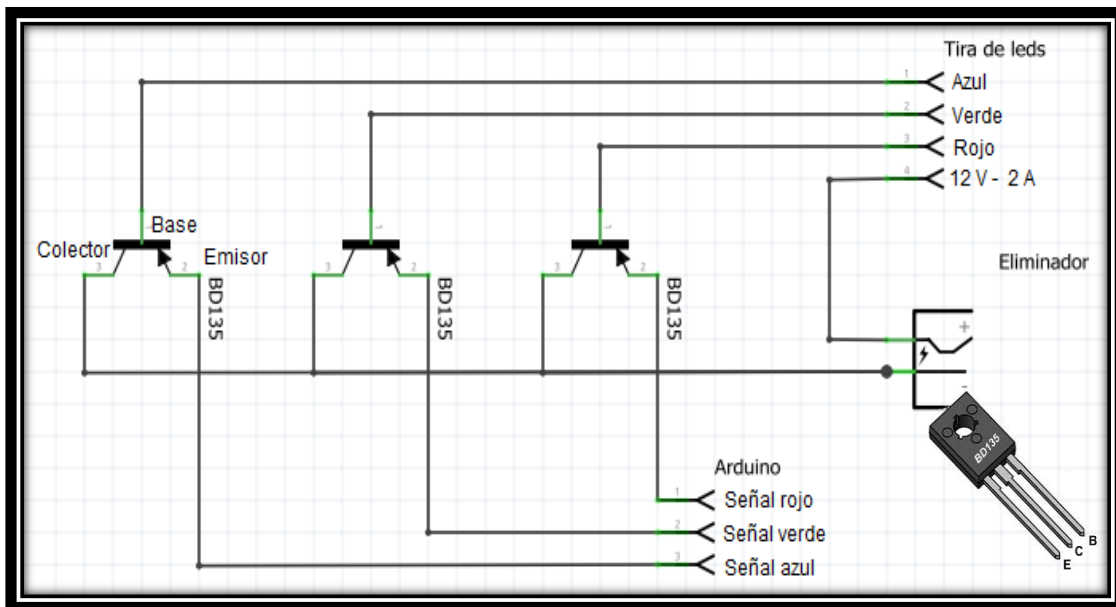


ILUSTRACIÓN 18. ESQUEMA DE CONEXIÓN DE LA TIRA DE LED

2.3.2 Relevadores

Un relevador es un interruptor controlado por un circuito eléctrico capaz de manipular un circuito de salida de mayor potencia que el de entrada, en el que, por medio de una bobina y un electroimán, se acciona un juego de uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes.

En la automatización del SIT es necesario la activación de una bomba de aire, así como de electroválvulas, las cuales necesitan de una cantidad de energía elevada para funcionar, por lo que fue necesario implementar relevadores con los que se les logró activar utilizando una señal de entrada mínima, los cuales permitirán el paso de la energía eléctrica a los dispositivos a partir de una señal enviada desde Arduino. En

se buscó una shield con la cual se logró establecer una conexión de comunicación remota; las *shields* que se analizaron se muestran a continuación.

- Ethernet shield: En la documentación oficial (Arduino, 2015) menciona que el Ethernet shield se conecta a internet en cuestión de minutos por medio un cable tipo RJ45, se conecta desde la placa hasta el modem de red; para funcionar necesita de una placa Arduino y una librería de programación para poder configurar la shield, opera a 5 V que son suministrados desde Arduino y tiene una velocidad de conexión de 10/100 Mb.

Hoy en día internet es una herramienta muy importante, sin embargo, se debe considerar que el costo de su acceso en México es elevado y aún más en dispositivos móviles, otra desventaja es que es necesario la contratación de un servidor para almacenar datos y crear un puente de comunicación entre Arduino y el dispositivo móvil aumentando el costo en la aplicación, a la vez se estaría obligando a contratar un servicio de internet en el SIT para que se pueda realizar la comunicación, por lo tanto no es viable la utilización de esta placa.

- Bluetooth Shield: Esta shield utiliza un módulo que soporta la comunicación serial inalámbrica por medio de la tecnología bluetooth, para usarse es necesario de una placa Arduino que le suministra los 5 V necesarios para su operación, sin embargo la última versión de esta tecnología (v 4.0) tiene un alcance máximo de 100 m en espacio abierto por lo que no permite un control remoto desde lugares lejanos del SIT, otro punto negativo de esta tecnología es que los dispositivos móviles gastan una gran cantidad de batería en la operación del bluetooth por lo que es no es conveniente para ningún usuario, por lo tanto no es viable el uso de esta shield.
- GSM Shield: La tecnología GSM (por sus siglas en ingles *Global System for Mobile communications*) es un estándar de comunicación para dispositivos celulares aceptado globalmente, por lo tanto todo teléfono móvil contiene la tecnología GSM. La placa GSM necesita de una placa Arduino, se alimenta

con 9V y posee una ranura para colocar una tarjeta inteligente SIM (acrónimo en inglés de *Subscriber Identity Module*) de la compañía celular que se desee.

El uso de la tecnología GSM tiene la ventaja de tener una cobertura mundial y al ser un estándar de comunicación está presente en todos los celulares, por lo tanto la distancia que exista entre el SIT y el usuario no tiene importancia para el control remoto, otra ventaja es su costo, puesto que hoy en día el envío de mensajes SMS (por sus siglas en inglés *Short Message Service*) es de bajo costo; con estas ventajas,

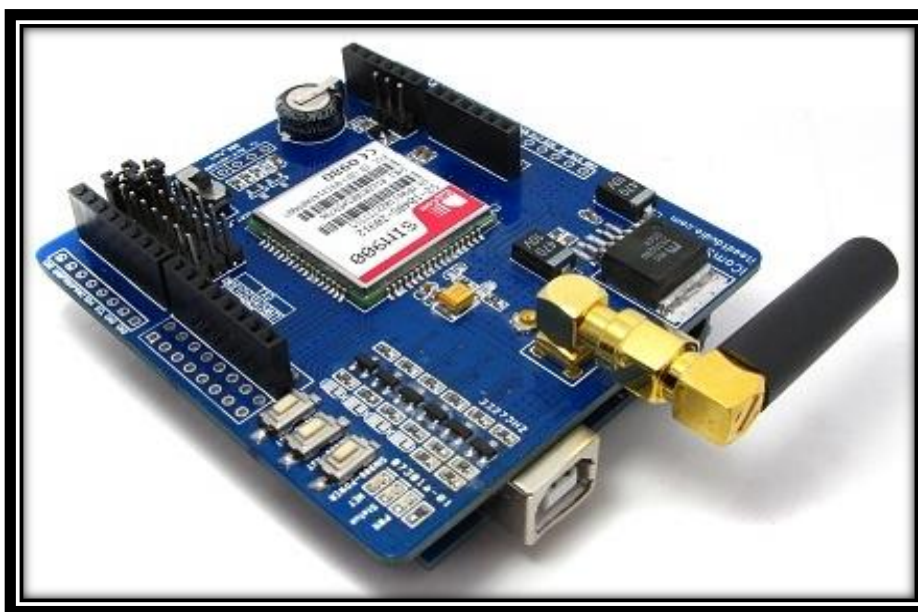


ILUSTRACIÓN 20 SHIELD GSM UTILIZADA EN EL SIT

esta tecnología es la mejor opción para aplicarse en el control remoto del SIT. En la *ilustración 20* se muestra una imagen de la placa GSM que se aplicó al proyecto de automatización.

Capítulo III: Desarrollo

Durante el proceso de desarrollo en la automatización del SIT, se crearon prototipos los cuales eran sometidos a pruebas y después del análisis de resultados se detectaron los errores presentes en sus ejecuciones, posteriormente se desarrolló un nuevo prototipo que solucionaba los problemas que evitaban su buen desempeño hasta lograr un producto final.

En este capítulo se demuestra el desarrollo e implementación de cada uno de los prototipos, también se describen los problemas y las actualizaciones realizadas para lograr el desarrollo de un siguiente prototipo o en su defecto, llegar al prototipo final que soluciona todos los problemas a resolver en el SIT.

Prototipo 1

El desarrollo del prototipo 1 se enfocó hacia el desarrollo de un software de control; el cual se comunica con el usuario por medio de una interfaz y la vez con la placa Arduino; este último con el uso de un cable USB. La conexión entre la placa Arduino y los actuadores o dispositivos electrónicos son también punto clave durante el proceso en la creación del presente prototipo; para simplificar la comprensión y alcance del desarrollo, en la *ilustración 21* se muestra el diagrama UML de los procesos que intervienen en el sistema de prueba.

En el prototipo I se orientó al desarrollo del hardware para la activación de los dispositivos involucrados en la automatización del SIT, también se programó un software de escritorio con el lenguaje de programación Java para la computadora que controla los procesos de activación en los elementos físicos del hardware a partir de horarios asignados por un usuario mediante el uso de una GUI (por su acrónimo en inglés *Graphical User Interface*) diseñado también en el programa Java.

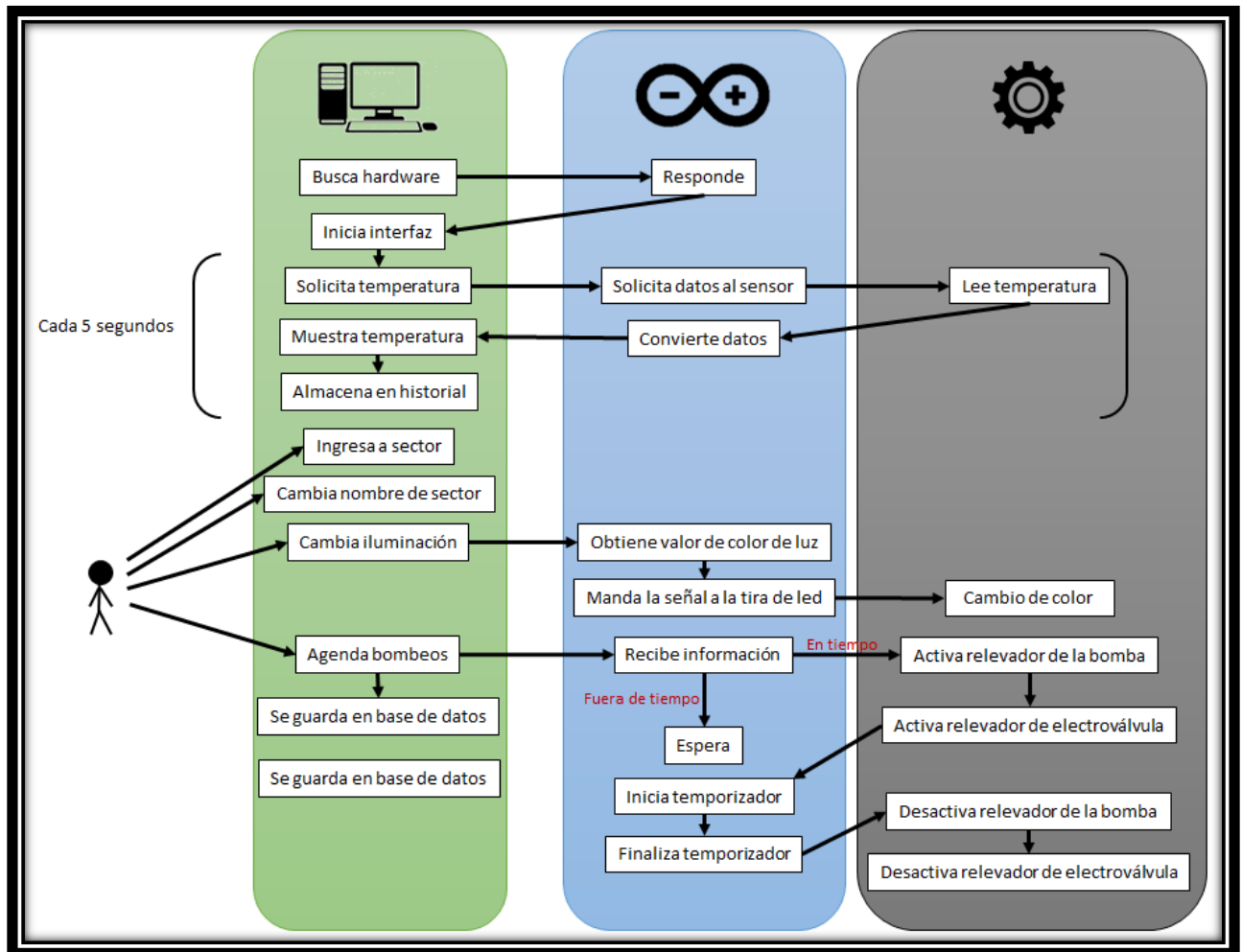


ILUSTRACIÓN 21. DIAGRAMA UML DEL PROTOTIPO 1

Desarrollo del Hardware

Para el desarrollo del hardware se crearon diversos módulos en el que cada uno realiza una acción en la automatización, a lo largo de este tema se describen cada uno de los módulos creados en el prototipo 1 así como su unión a la placa Arduino, el cual envía las señales de activación. A continuación, se enlistan los módulos que se desarrollaron.

- Monitoreo de temperatura: en este módulo se configuró el sensor de temperatura que monitorea el SIT, para su implementación en Arduino se descargó una librería especial el cual convierte el valor analógico del sensor en grados Centígrados (°C), Fahrenheit (°F) o Kelvin (°K). En el caso de la presente investigación, el valor de la temperatura se mostrará en grados centígrados.

- Monitoreo de luz: la monitorización de iluminación se obtiene mediante fotorresistencias conectadas a la placa Arduino; el valor que obtienen los dispositivos es de forma análoga, por lo que se determinó un tono de iluminación para establecer cuando se considera falta de iluminación o cuando hay demasiada luz dependiendo el caso que el usuario haya asignado a los sectores del SIT.

- Relevadores: modulo donde se conectaron los relevadores y se aplicaron al encendido y apagado de dispositivos que interactúan con el SIT, como electroválvulas y bomba de aire.

- Iluminación led: en este apartado se conectaron la tira de ledes a cada sector del SIT, y se le aplicaron los transistores que permiten la activación y cambios de color desde la placa Arduino.

Para este prototipo se había propuesto el uso de una placa Arduino Mega 2560 (véase apartado 2.1.1.3) ya que su estructura ofrecía lo necesario para la automatización y el número de puertos eran más de los necesitados por los dispositivos a controlar.

El primer módulo que se realizó tiene la tarea de monitorear la temperatura, por lo que se tuvo la necesidad de buscar un sensor digital que satisficiera esta necesidad.

El sensor usado es el DS18B20, el cual tiene 3 puertos o salidas (voltaje, tierra y datos), el sensor de temperatura puede ser conectado directamente a la placa Arduino, y se alimenta desde los puertos de 3.3V o 5V; a la salida de datos del sensor se le asignó la entrada digital número 46. En la *ilustración 22* se muestra la conexión del sensor de temperatura a la placa Arduino.

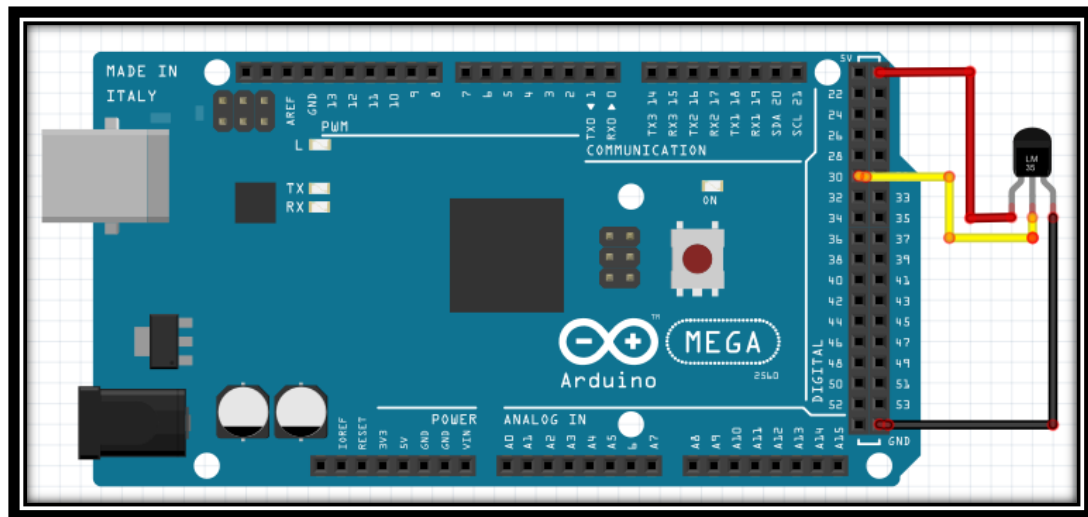


ILUSTRACIÓN 22 CONEXIÓN DEL SENSOR DE TEMPERATURA A ARDUINO MEGA 2560

El segundo módulo en desarrollarse fue el de la monitorización de presencia de luz, para esto se usaron fotorresistencias, una para cada sector; para su aplicación fue necesario de conectarles una resistencia de 1 KOhm a cada una para evitar daños generados por la corriente eléctrica. Las fotorresistencias son conectadas al puerto de 3.3V y a las entradas de la placa Arduino, que reciben la señal de las fotorresistencias con el puerto analógico 0 (A0) y el puerto analógico 1 (A1). En la *ilustración 23* se muestra la imagen de la conexión de las fotorresistencias a la placa Arduino.

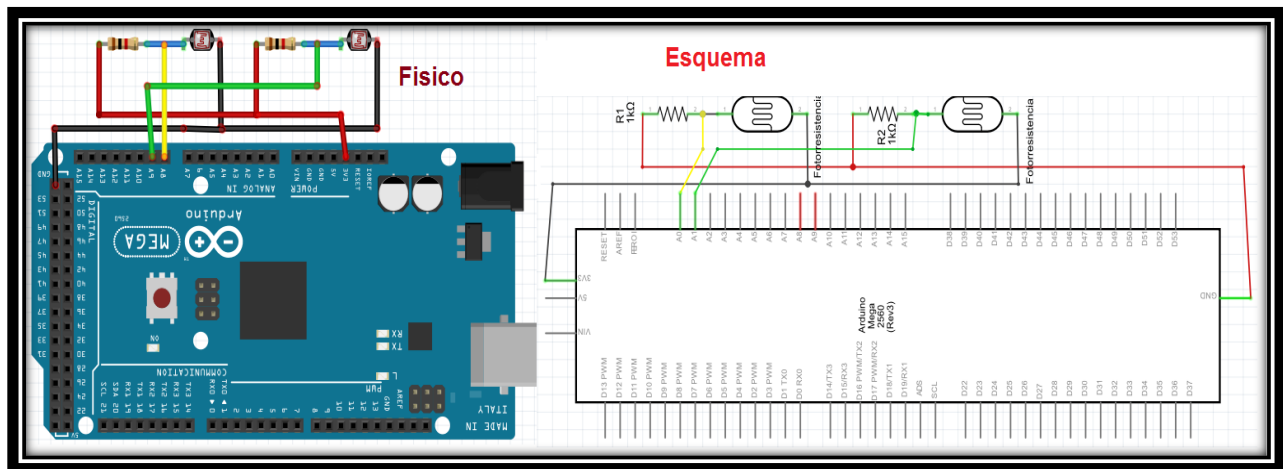


ILUSTRACIÓN 23 CONEXIÓN DE FOTORRESISTENCIAS A LA PLACA ARDUINO MEGA 2560

El tercer módulo es el que activa la bomba y electroválvulas del SIT a partir de la activación de relevadores por la placa Arduino. Para el presente proyecto se decidió adquirir una placa creada específicamente para la tecnología de Arduino, por lo que su conexión fue similar al del sensor de temperatura. La placa contiene 4 relevadores los cuales tienen sus puertos para conectar los dispositivos de la inmersión, también contiene de 8 puertos para Arduino: voltaje a 5V, tierra, señal de relevador 1, señal de relevador 2, señal de relevador 3, señal de relevador 4 y dos puertos de comunicación que deben estar unidos entre sí.

En el presente proyecto de automatización se decidió trabajar sobre un SIT que tuviera 2 sectores por lo tanto, solo se necesitaran 3 relevadores para poder operar el sistema de aire; el relevador 1 conectado al puerto digital 45 de la placa Arduino controla la bomba de aire, se debe aclarar que el relevador soporta una bomba de máximo ¼ hp, por lo que si se decide colocar una bomba de aire de más fuerza se tendrá que adquirir un relevador más potente, el relevador 2 y 3 conectados a los puertos digitales 43 y 42 respectivamente, activarán electroválvulas que permitirán el acceso del aire al sector que necesite realizar una inmersión. En la *Ilustración 24* se muestra la conexión de la tira de relevadores a la placa Arduino.

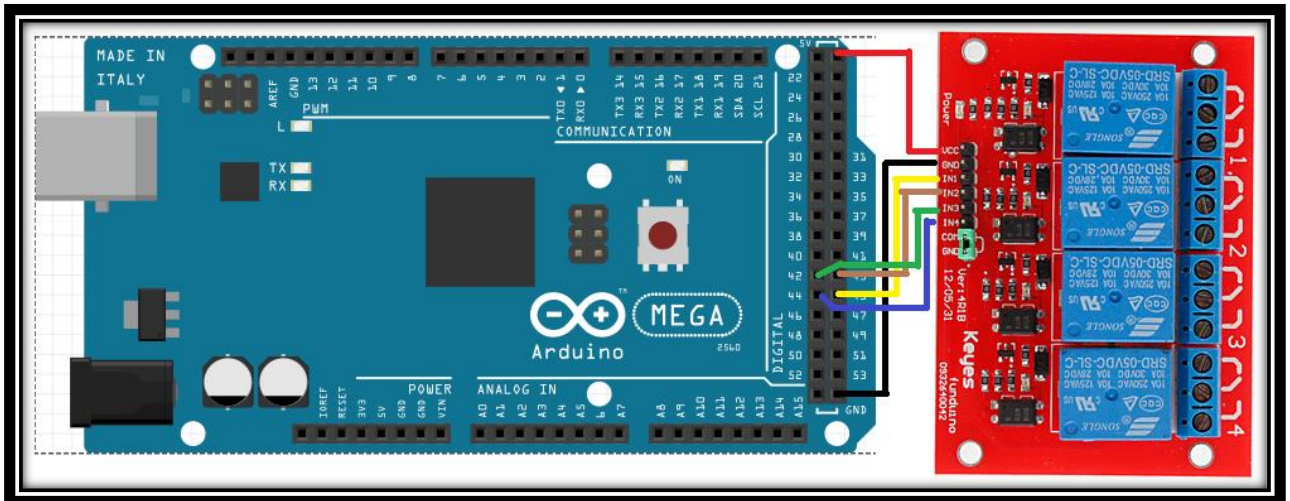


ILUSTRACIÓN 24 CONEXIÓN DE RELEVADORES A LA PLACA ARDUINO

El último módulo desarrollado fue el de las luces led, el cual se conforma de una tira de led RGB 5050 para la generación de iluminación en el SIT, esta tira debe combinarse con transistores BD135 para recibir la señal de activación de la placa Arduino, la conexión de la tira de led por sector puede verse en el capítulo 2.3.1. En la *ilustración 25* se muestra la conexión de una tira de led físicamente.

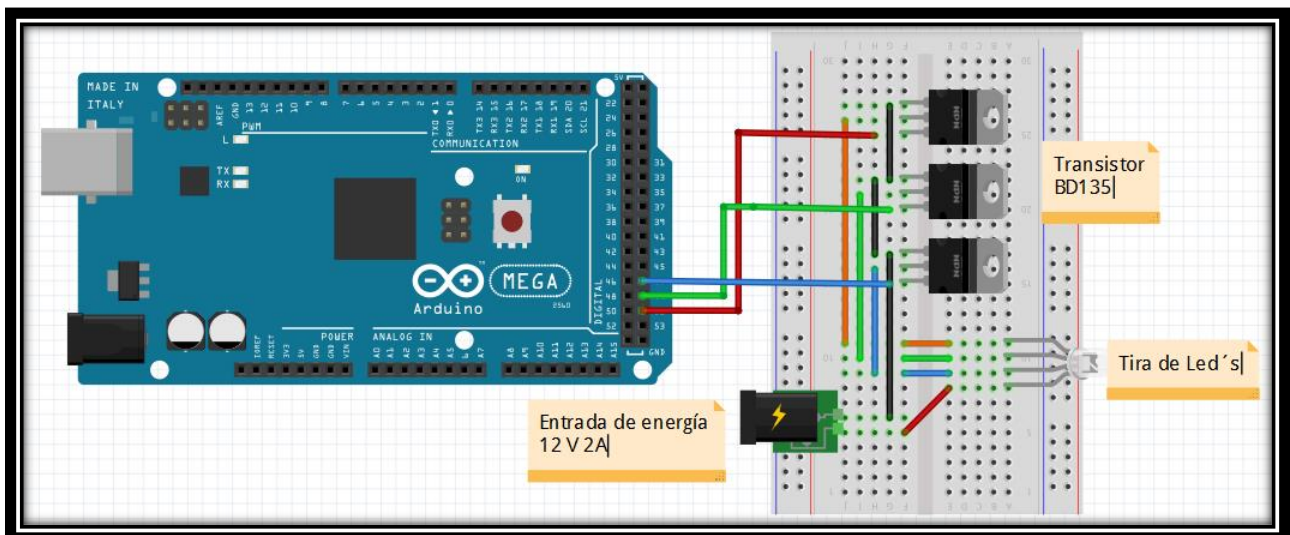


ILUSTRACIÓN 25 CONEXIÓN FÍSICA DE UNA TIRA DE LED

Al ser dos sectores en el SIT propuesto para esta investigación, es necesario de 6 puertos digitales para controlar los cambios de luces por medio de la placa Arduino, los cuales son mostrados en la *tabla 6*.

TABLA 6. PUERTOS DE ARDUINO MEGA PARA EL CAMBIO DE LUZ

Colores	Arduino Sector 1	Arduino Sector 2
Rojo	52	53
Verde	50	49
Azul	48	49

Desarrollo del software

El software de la automatización se caracteriza por llevar a cabo la mayor carga de trabajo, aprovechando los recursos que se obtienen de las computadoras actuales; gracias al poder de procesamiento de una computadora se pudo lograr la ejecución de hilos¹⁰, partes importantes del presente software. El programa creado para el SIT contiene una GUI simple donde el usuario podrá realizar el registro de los horarios para la activación tanto de los relevadores como de la tira de led, así mismo se desarrolló una base de datos para el almacenamiento de la información, donde se registran los datos más importantes que intervienen en la automatización del SIT.

¹⁰ Parte que puede ejecutarse de manera concurrente al resto de una aplicación y establece un camino de ejecución independiente. Es un tipo de multitarea (Osorio Rivera, 2007)



Base de datos

Para el tratado de la información como los horarios, el color de las luces, la duración de activación en los relevadores, los registros de temperatura y los de las fotorresistencias fue necesario implementar una base de datos en la que se pudiera guardar, modificar e inclusive eliminar información a juicio del usuario. Para el presente proyecto de investigación se decidió crear una base de datos con el gestor SQLite ya que este presenta muchas ventajas a diferencia de otros sistemas de almacenamiento de información.

SQLite es (Kreibich, 2010) un paquete de software de dominio público que provee de un sistema de administración de base de datos relacionales, que son usados para almacenar información en largas tablas y que puede procesar comandos SQL (por su acrónimo en inglés Structured Query Language) complejos que combinan datos desde múltiples tablas para generar reportes y resúmenes. SQLite es definido por las siguientes características:

- No necesita servidor
- No necesita configurarse
- Es multiplataforma
- Ocupa muy bajos recursos
- Alto rendimiento

En general, SQLite provee de un entorno de base de datos relacionales flexible y muy funcional que consume mínimos recursos y crea mínima molestia a desarrolladores y usuarios.

A continuación se describe la base de datos que se desarrolló para el almacenamiento de los datos del SIT en el prototipo I, es decir, se mostrarán los nombres de las tablas y los campos que lo conforman:

AdvertenciaFoto: cuando la aplicación de escritorio solicita la información del valor de las fotorresistencias en la placa Arduino, estas se comparan con la iluminación

presente en el sistema y determina si la intensidad lumínica es alta o baja dependiendo el estado que debería tener el SIT. En esta tabla se guarda la información cuando las fotorresistencias detectan que la iluminación no es adecuada en el sistema de micropropagación, almacenando los datos que representan una falla de iluminación ya sea por la tira de ledes o por un factor ajeno a la automatización.

- **IdAdvertencia:** campo donde se almacena un identificador único de registro de valor numérico, este valor es el atributo llave de la tabla, así mismo se configura como auto incrementable, es decir que cada vez que se ingresa un nuevo registro, este adquiere automáticamente un identificador único.

- **IdSector:** en este campo se guarda el id del sector como un valor numérico, si es el primer sector el que presenta la falla se escribe 1 y si es el segundo sector donde se encuentra la incongruencia de iluminación se almacena un 2, esto para diferenciar el sector donde se haya presentado la alteración lumínica.

- **Fotorresistencia:** cuando el programa de escritorio Java tiene la información de las fotorresistencias y es procesada dando como resultado una falla lumínica, entonces se almacena el valor de la fotorresistencia del sector correspondiente al error de luz, siendo este de valor numérico.

- **Fecha:** campo donde se registra la fecha, proporcionada por la computadora, en que se realizó la lectura de la fotorresistencia y se haya detectado el error de la intensidad de luz, en el formato YYYY-MM-DD, es decir, año – mes – día.



- Hora: almacenamiento de la hora, proporcionada por la computadora, en la que fue realizada la lectura de la fotorresistencia y determinado la presencia de error en la iluminación, en el formato de 24h.

Bombeo: cuando el usuario ingresa la información de los bombeos en la aplicación de escritorio Java, se usa esta tabla para registrar los horarios y duraciones de los bombeos en los sectores del SIT que el usuario haya asignado para cada sector, de esta forma la información siempre está presente y es almacenada de forma segura para su aplicación en la automatización.

- IdTabla: identificador único de cada registro de la tabla de valor numérico y a la vez es autoincrementable, es decir que cada registro ingresado obtiene automáticamente un identificador, el cual ayuda a las búsquedas de información.
- IdSector: en este campo se guarda el id del sector como un valor numérico, si es el primer sector donde el usuario ingresa un nuevo horario de bombeo se escribe 1 y si es el segundo sector donde se guarda información se almacena un 2, esto para diferenciar el sector donde se haya guardado el horario asignado para efectuar los bombeos en el SIT.
- HoraBombeo: en este campo se almacena la hora en el formato de 24h, que el usuario haya designado al nuevo bombeo registrado, este valor se usa en la aplicación Java para la activación de los rel evadores que encienden la bomba de aire así como la electroválvula del sector

correspondiente al comparar la hora guardada con la hora de la computadora.

- Duración: se guarda la duración que tendrá el bombeo, este valor se guarda como un valor numérico y se toma como los segundos que están activos los relevadores, que encienden los elementos que intervienen en la inmersión.

HistorialBombeos: cuando en el SIT se lleva a cabo una inmersión, se guarda la información del bombeo en la presente tabla, esto es para que el usuario pueda verificar que las inmersiones se están llevando a cabo en el horario asignado. Si un bombeo no se realiza, este no se registra en la tabla, por lo que el usuario puede consultar la información y determinar si no se ha llevado a cabo una inmersión por factores externos a la automatización.

- IdHisBombeo: identificador único de cada registro de la tabla de valor numérico y a la vez es autoincrementable, es decir que cada registro ingresado obtiene automáticamente un identificador, el cual ayuda a las búsquedas de información.
- IdSector: en este campo se guarda el id del sector como un valor numérico, si es el primer sector donde se realizó el bombeo se escribe 1 y si es el segundo sector donde existió una inmersión se almacena un 2, esto para diferenciar el sector donde se haya presentado actividad de los relevadores en horario asignado por él usuario.



- Fecha: campo donde se registra la fecha cuando se ha llevado a cabo la inmersión, proporcionada por la computadora en el formato YYYY-MM-DD, es decir, año – mes – día.
- Hora: almacenamiento de la hora, proporcionada por la computadora, en la que fue realizada la lectura de la fotorresistencia y determinado la presencia de error en la iluminación, en el formato de 24h.
- Duración: en este campo se guarda la duración del bombeo que se realizó, con un valor numérico indicando los segundos que se realiza la inmersión, esto es útil cuando el usuario cambia la duración del bombeo, teniendo un registro confiable de las inmersiones realizadas con las distintas duraciones que se han almacenado.

HistorialIluminacion: de forma similar a la tabla *HistorialBobeos*, en esta tabla se almacena la información de los cambios de color en la iluminación del SIT, con esta información, el usuario puede verificar el correcto funcionamiento del sistema de iluminación presente en la automatización.

- IdHistorial: identificador único de cada registro de la tabla de valor numérico y a la vez es autoincrementable, es decir que cada registro ingresado obtiene automáticamente un identificador, el cual ayuda a las búsquedas de información.
- IdSector: en este campo se guarda el id del sector como un valor numérico, si es el primer sector donde se realizó el cambio de

iluminación se escribe 1 y si es el segundo sector donde existió una alteración en el sistema de luz se almacena un 2, esto para diferenciar el sector donde se haya presentado actividad de cambio en la tira de ledes.

- Tipo: para la obtención de datos de forma más rápida, se almacena un valor numérico entero del 0 al 4, a los cuales se les asigna un color, es decir con este número se identifica el color de la iluminación después del cambio de luz, donde 0 indica que las luces están apagadas, 1 indica iluminación blanca, 2 significa iluminación roja, 3 cambio a iluminación azul y 4 que indica iluminación verde.

- Fecha: campo donde se registra la fecha cuando se llevo a cabo el cambio de iluminación en algún sector del SIT, proporcionada por la computadora en el formato YYYY-MM-DD, es decir, año – mes – día.
- Hora: almacenamiento de la hora, proporcionada por la computadora, en la que fue realizada la lectura de la fotorresistencia y determinado la presencia de error en la iluminación, en el formato de 24h.

Sector: en la presente tabla, se almacena la información básica de los sectores del SIT, así mismo esta tabla puede ser editada en caso de tener más de dos sectores ya que con el atributo llave presente en la tabla, permite la integración e interacción con las demás tablas presentes en la base de datos.

- IdSector: en este campo se guarda el id del sector como un valor numérico, si es el primer sector donde se realizó el cambio de iluminación se escribe 1 y si es el segundo sector donde existió una alteración en el sistema de luz se almacena un 2, esto para diferenciar



el sector donde se haya presentado actividad de cambio en la tira de ledes.

- Nombre: nombre asignado a cada sector, este nombre puede ser editado desde el software de escritorio Java, esto ayuda al usuario a identificar los sectores de manera más fácil.
- ColorLuz: para la obtención de datos de forma más rápida, se almacena un valor numérico entero del 0 al 4, a los cuales se les asigna un color, es decir con este número se identifica el color de la iluminación actual, donde 0 indica que las luces están apagadas, 1 indica iluminación blanca, 2 significa iluminación roja, 3 indica a iluminación azul y 4 que indica iluminación verde.

Temperatura: en esta tabla se registra la temperatura que el programa de escritorio Java pide al sensor de temperatura ubicada en el SIT, este valor es decimal y el registro se hace cada 5 segundos.

- IdTemperatura: identificador único de cada registro en la tabla de valor numérico y a la vez es autoincrementable, es decir que cada registro ingresado obtiene automáticamente un identificador, el cual ayuda a las búsquedas de información.
- Estado: para determinar el estado de la temperatura, el valor obtenido del sensor es comparado con el rango mínimo (15 °C) y el rango máximo (30 °C), si la temperatura se encuentra dentro del rango, entonces se guarda la cadena de caracteres "Estable", si el valor de la temperatura es menor al rango mínimo, entonces se almacena la

cadena “Bajo” y finalmente, si el valor de la temperatura es mayor al rango máximo, se guarda la cadena “Alto”.

- Fecha: campo donde se registra la fecha, proporcionada por la computadora, cuando se solicitó el valor de la temperatura desde el programa de escritorio Java, en el formato YYYY-MM-DD, es decir, año – mes – día.
- Hora: almacenamiento de la hora, proporcionada por la computadora, en la que fue realizada la lectura de la fotorresistencia y determinado la presencia de error en la iluminación, en el formato de 24h.

Después de la creación de la base de datos para almacenar la información del SIT, se procedió a realizar el programa de control en Java, donde se utilizó una librería de código abierto para poder vincular la base de datos con la aplicación, de esta manera el usuario podrá leer, escribir, eliminar o modificar información de la base de datos. En la *ilustración 26* se muestra el diseño de la base para el prototipo 1.

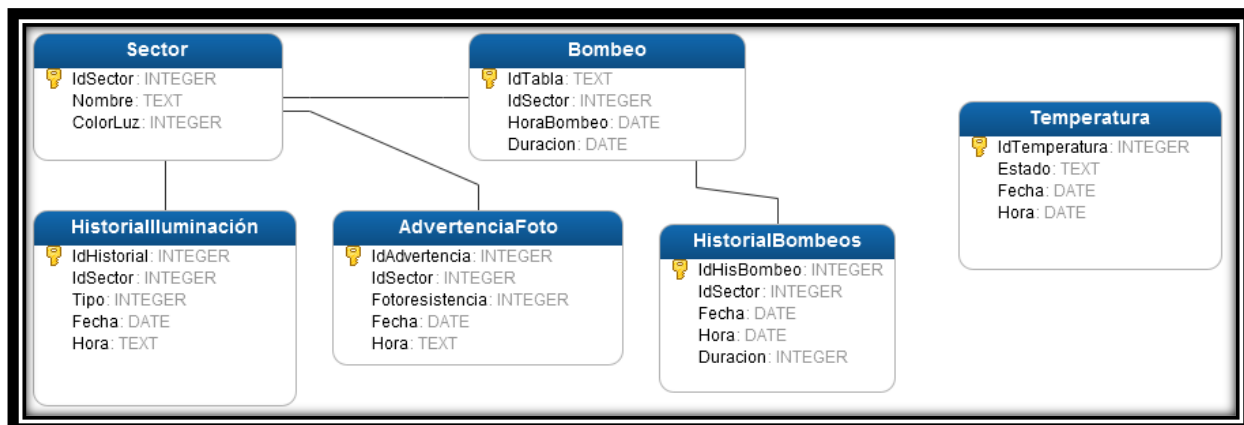


ILUSTRACIÓN 26. DISEÑO DE LA BASE DE DATOS. PROTOTIPO 1



GUI

Tomando en cuenta a los usuarios, se desarrolló una interfaz en lenguaje Java fácil de usar donde se ingresa de forma manual los horarios para los bombeos, también se despliega la información necesaria de los sensores que monitorean al SIT, así como las advertencias que aparecerán en la pantalla en caso de una incorrecta temperatura o iluminación.

Para que el programa pueda ejecutarse correctamente es necesario tener conectada la placa Arduino a la computadora; Arduino se conecta a un puerto USB (por sus siglas en inglés *Universal Serial Bus*) y se configura como un puerto COM (también conocido como puerto de comunicación serial). Para comunicar la placa Arduino a Java fue necesaria la descarga de una librería especial llamada RXTX el cual puede descargarse¹¹ gratuitamente desde internet.

La librería RXTX contiene un conjunto de archivos los cuales deben copiarse a las carpetas correspondientes para lograr la detección y comunicación de la placa Arduino en Java. A continuación, se enlistan los archivos y las rutas donde deben ser copiadas:

TABLA 7. RUTAS DONDE PARA LOS ARCHIVOS DE RXTX

Archivo	Ruta
RXTXcomm.jar	C:\Program Files\Java\jreX.X.X \lib\ext
rxtxSerial.dll	C:\Program Files\Java\jreX.X.X \bin
rxtxParallel.dll	C:\Program Files\Java\jreX.X.X \bin

¹¹ Librería RXTX para Java puede descargarse en http://www.jcontrol.org/download/rxtx_en.html

NOTA: en las rutas, la *jreX.X.X* corresponden a la versión de Java que se tenga instalado en la computadora, en el presente proyecto la versión de Java usada fue la *jre1.8.0_31*.

Al tener la comunicación establecida entre Java y Arduino, se logró la creación de ventanas emergentes que alertan al usuario si la placa Arduino está conectada. En la *ilustración 27* se muestra la ventana emergente cuando la aplicación de escritorio Java no detecta el puerto COM de la placa Arduino.



ILUSTRACIÓN 27 ALERTA DE CONEXIÓN A ARDUINO

Si el hardware o Arduino está conectado al puerto USB de la computadora, aparece la ventana principal de la aplicación, donde se muestra la lectura del sensor de temperatura que es actualizada cada 10 segundos, a la vez, el dato de la temperatura es guardada en la base de datos con la hora y fecha del ordenador; también aparecen en la ventana tres botones, como se había mencionado anteriormente el prototipo del SIT se orientó a dos sectores, por esta razón hay un botón para cada sector y el de salir de la aplicación. En la *ilustración 28* se muestra la venta principal de la aplicación Java.



ILUSTRACIÓN 28 VENTANA PRINCIPAL DE LA APLICACIÓN JAVA

Para que el usuario pueda interactuar con un sector del SIT es necesario que en la ventana principal seleccione el botón del sector que se desea ver, posteriormente se abrirá el marco de monitoreo, en la cual se especifica el nombre asignado por el usuario, un panel de iluminación donde el usuario elige el color de cambio de luz, así como una imagen representativa y un botón para agendar los horarios de los bombeos. En la *ilustración 29* se muestra la ventana de monitoreo por sector desarrollada para el prototipo I.



ILUSTRACIÓN 29 VENTANA DE MONITOREO POR SECTOR

Al cambiar el nombre del sector se abre una ventana con un área de texto donde el usuario podrá editar el alias para una mayor comodidad; cuando se acepta el cambio se obtiene el texto y se almacena en la base de datos, esto ayuda al usuario a identificar los sectores del SIT. En la *ilustración 30* se muestra la ventana de cambio de nombre.

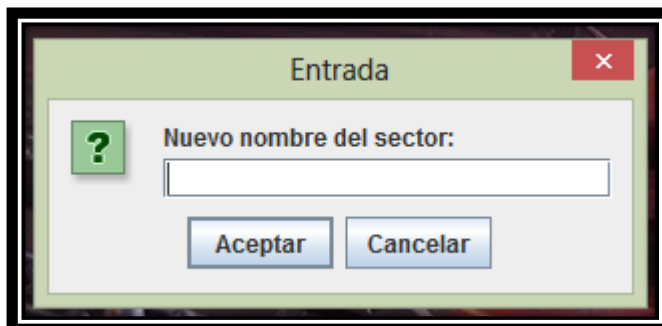


ILUSTRACIÓN 30 VENTANA DE CAMBIO DE NOMBRE

Para la entrada de los horarios por parte del usuario, se creó una ventana que es accesible desde el marco de monitoreo por sector, a continuación, se despliega una tabla con los horarios almacenados actualmente y la duración en segundos que va a ejecutarse los bombeos, en la parte izquierda de la ventana se puede acceder a registrar un nuevo horario o a eliminarlo, según sea el caso del botón que se elija. En la *ilustración 31* se muestra la ventana de los bombeos.

Para ingresar un nuevo horario para el bombeo de un sector, se tiene que seleccionar el botón “Agregar” de la ventana de horarios, posteriormente se abre un nuevo marco donde se le pedirá al usuario ingrese la hora (formato 24h) y los minutos para



ILUSTRACIÓN 31 VENTANA DE HORARIOS DE BOMBEO

almacenarlo en la base de datos, así mismo se pide la duración en segundos del bombeo, cuando el usuario presiona el botón *Aceptar*, entonces a información es guardada en la base de datos. En la *ilustración 32* se muestra la ventana para guardar un nuevo horario. La ventana tiene programación que valida los datos que ingresa el usuario:

- ✓ El usuario no puede ingresar caracteres alfabéticos
- ✓ El campo de horas, no puede ser mayor a 24 ni menor a 0
- ✓ En el campo de minutos, no puede ser mayor a 60 ni menor a 0

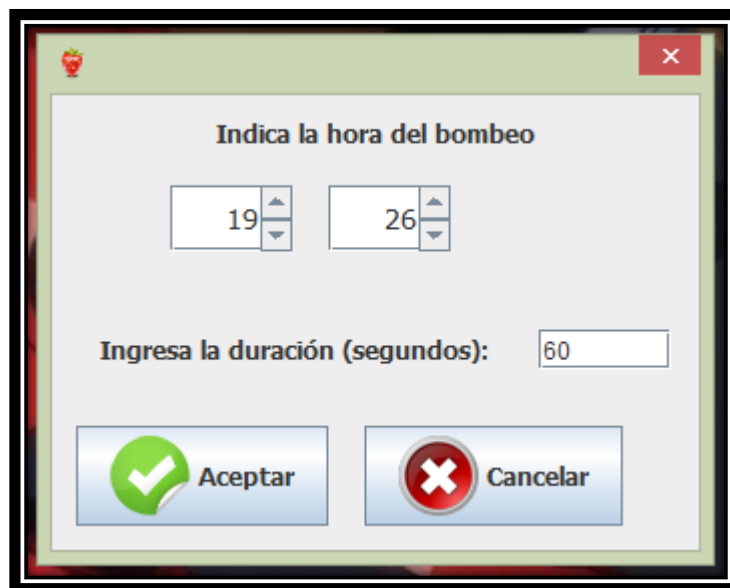
The image shows a software dialog box with a title bar containing a small red icon and a close button. The main content area has a title "Indica la hora del bombeo". Below the title are two spinners: the first shows "19" and the second shows "26". Underneath these is a label "Ingresa la duración (segundos):" followed by a text input field containing "60". At the bottom, there are two buttons: "Aceptar" with a green checkmark icon and "Cancelar" with a red X icon.

ILUSTRACIÓN 32 VENTANA PARA GUARDAR UN NUEVO HORARIO

Una necesidad importante para el usuario, es que el sistema advierta cuando el SIT sufre de alteraciones como la temperatura, en el caso de los explantes, el rango de temperatura estable oscila de entre 15°C a 30°C, por lo tanto, es indispensable que el software emita una notificación de alerta cuando la temperatura que detecta el sensor no está dentro del rango. En la *ilustración 33* se muestra la ventana de advertencia que emite el programa cuando la temperatura es menor a 15°C.

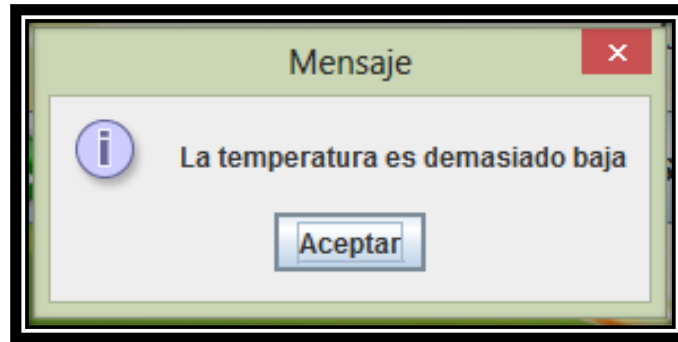


ILUSTRACIÓN 33 ADVERTENCIA DE TEMPERATURA BAJA

En el caso que la temperatura exceda el rango máximo permitido (30°C), se emite una ventana de alerta como la que se muestra en la *ilustración 34*. Cuando la temperatura está dentro del rango, no se solicita ninguna notificación.

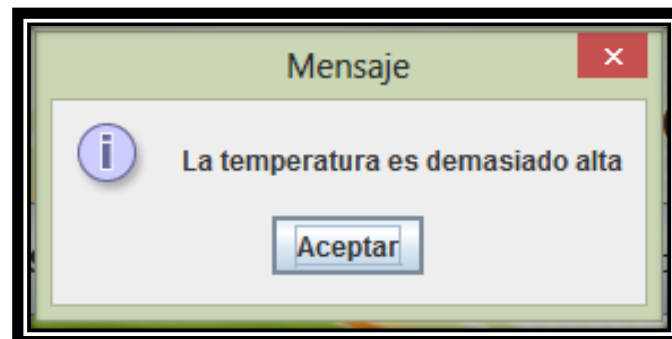


ILUSTRACIÓN 34 ADVERTENCIA DE TEMPERATURA ALTA

En los tres casos de temperatura, la información es almacenada en sus respectivas tablas en la base de datos, guardando una cadena de texto dependiendo la situación, estas palabras son: *Estable*, indica que la temperatura está dentro del rango asignado, *Baja*, indica que la temperatura está por debajo del rango establecido y *Alta* que determina al valor de la temperatura por encima del rango.

El software cuenta también con notificaciones en relación a las fotorresistencias, en la *ilustración 35* se muestra la advertencia cuando hace falta iluminación, por el contrario, la *ilustración 36* muestra la advertencia cuando hay iluminación cuando no debería existir.

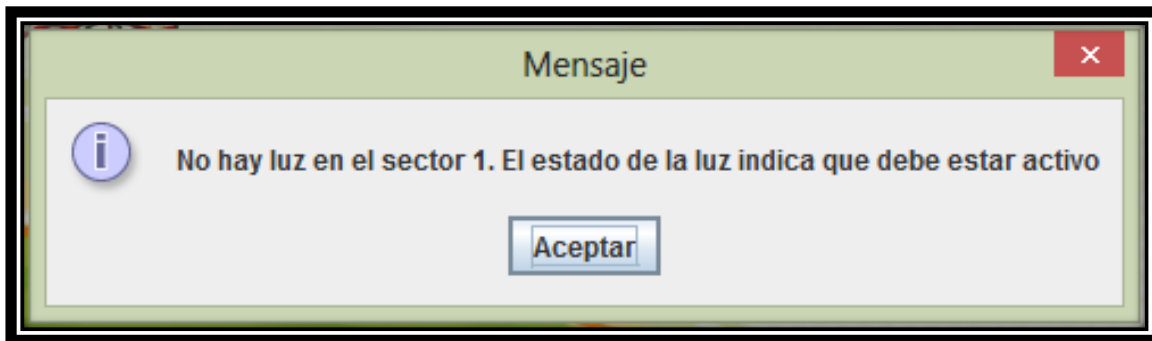


ILUSTRACIÓN 35 ADVERTENCIA DE FALTA DE LUZ

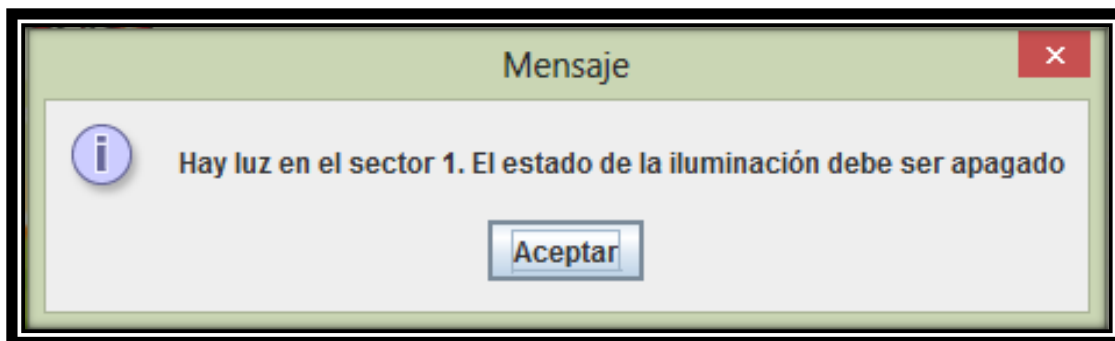


ILUSTRACIÓN 36 ADVERTENCIA DE EXISTENCIA DE LUZ

Procesos del software

El cuerpo del software se compone de varios métodos que realizan diferentes acciones, una característica principal de este programa Java es el uso de hilos que se ocupan de distintos procesos en la aplicación, llevándolos a cabo al mismo tiempo y de forma independiente entre ellos.

En el monitoreo de la temperatura, un hilo se ocupa del proceso de solicitar la información a Arduino el valor del sensor cada cinco segundos. En la *ilustración 37* se muestra el diagrama de flujo donde se muestra el proceso de la temperatura;



después de obtener el valor del sensor desde Arduino es almacenado en la base de datos, el siguiente paso es proyectar el valor en la aplicación y después el valor se somete a una condición: si no está en el rango, se emite una notificación y pasa a esperar 5 segundos antes de repetir el proceso; si la temperatura esta en rango, el proceso pasa a la espera de 5 segundos para el reinicio del hilo de forma directa.

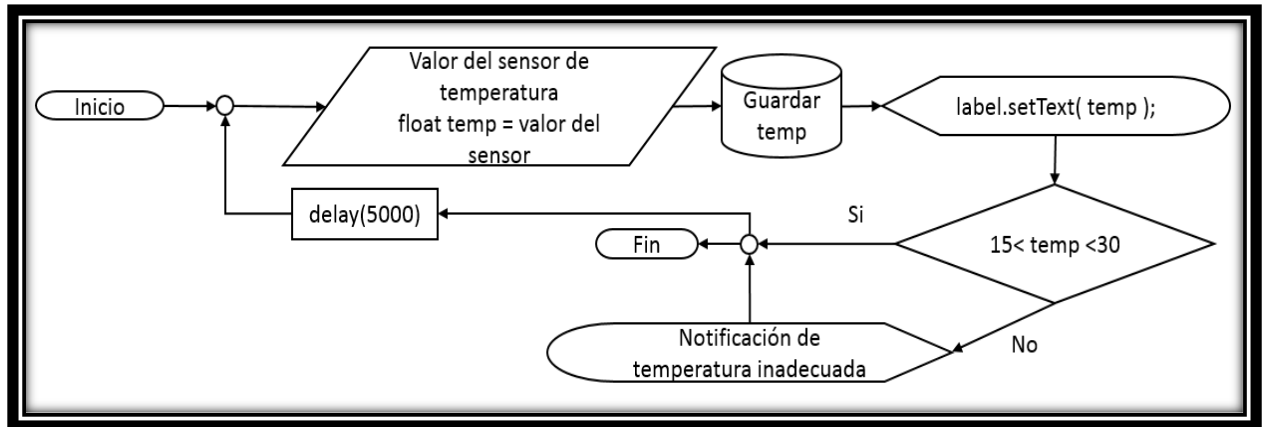


ILUSTRACIÓN 37 DIAGRAMA DEL HILO DE TEMPERATURA

El monitoreo de temperatura es importante, ya que el usuario puede regularla a las necesidades que se tengan con el cultivo y puede consultar su estado para interferir o no en la alteración de la temperatura monitoreada.

En el caso del monitoreo de la presencia de luz, se sigue un proceso similar al de la temperatura, en la *ilustración 38* se muestra el diagrama del hilo, el cual inicia obteniendo el valor de la fotorresistencia desde Arduino, posteriormente se despliega en la ventana del monitoreo de sector, a continuación se solicita por medio de un archivo llamado *luz.txt* el valor que tiene almacenado, donde 0 indica que las luces deben estar apagadas y donde un número mayor a 0 indica que las luces deben estar encendidas, luego se somete a condiciones, se guarda la información en la base de datos y el hilo reinicia el proceso cada 5 segundos.

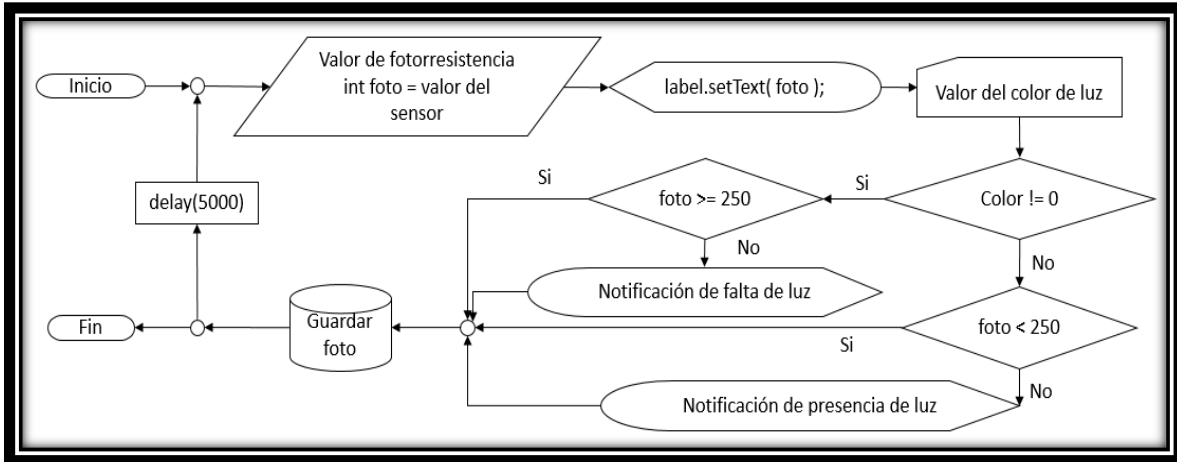


ILUSTRACIÓN 38 DIAGRAMA DEL PROCESO DE FOTORRESISTENCIAS

Para llevar a cabo el bombeo de aire o activación de los relevadores se desarrollaron dos hilos, uno para la detección de la hora de bombeo y otro para la duración del bombeo. En la *ilustración 39* se muestra el diagrama del hilo que detecta la hora del sistema y la compara con las de los horarios de bombeos de los sectores, cuando coinciden las horas, se llama al hilo de activación de relevadores; el reinicio de este proceso es de 45 segundos.

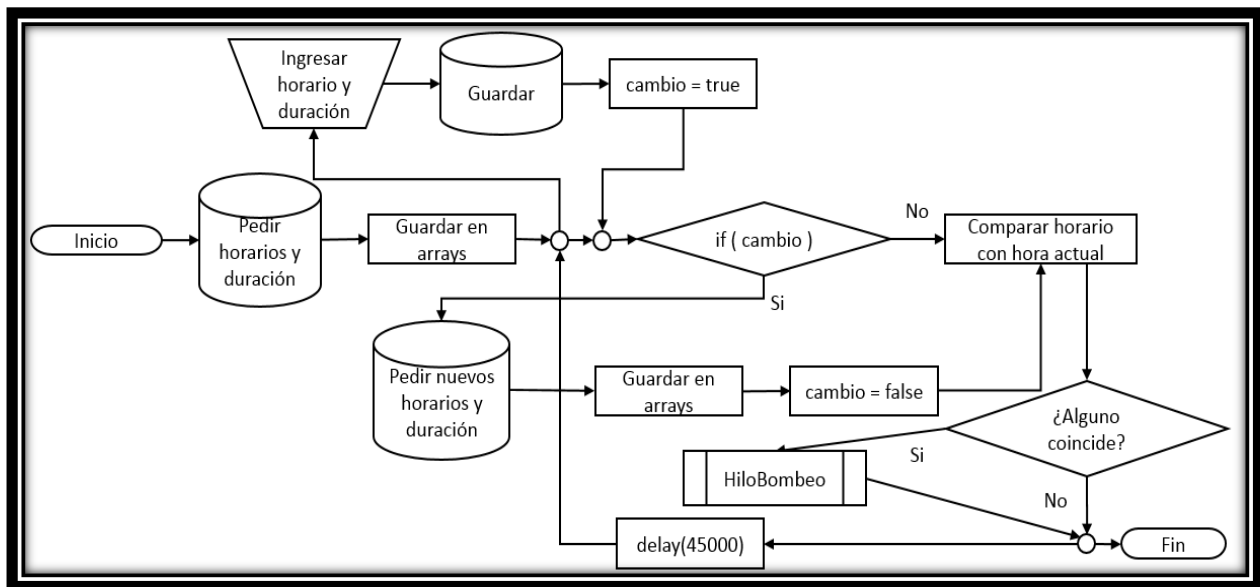


ILUSTRACIÓN 39 DIAGRAMA DEL PROCESO DE DETECCIÓN DE HORAS PARA EL BOMBEO



El proceso del hilo que activa los relevadores se muestra en la *ilustración 40*. El proceso enciende dos relevadores (uno para la bomba de aire y otro para la electroválvula) durante el periodo de tiempo en segundos que el usuario había

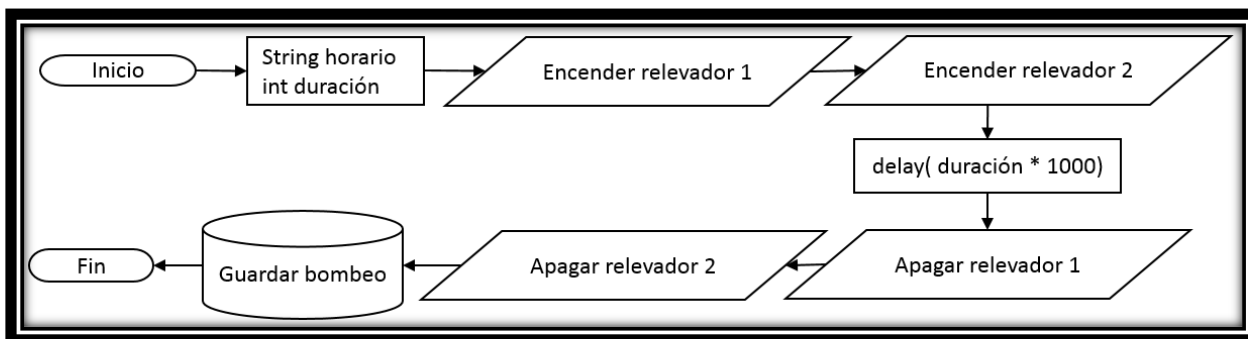


ILUSTRACIÓN 40 DIAGRAMA DEL HILO ACTIVADOR DE RELEVADORES

designado al bombeo.

Implementación y resultados

Tras el desarrollo del primer prototipo, los elementos fueron sometidos a pruebas de funcionamiento, realizando bombeos calendarizados, por lo que se comprobó el funcionamiento de los relevadores, así mismo se realizaron cambios en la iluminación, verificando que la tira de ledes encendiera y realizaran los cambios configurados; también se monitoreó los valores obtenidos por el sensor de temperatura y de las fotorresistencias, confirmando su configuración y funcionamiento.

Por otro lado, se buscó un lugar para poder aplicar el prototipo I, por lo que se acudió al COLPOS quienes nos permitieron hacer una prueba por un mes usando biorreactores reales sin embargo, fue necesario el crear una maqueta donde se colocaran los sensores y dispositivos inmersos en la automatización.

Para la implementación de la automatización se acudió al SIT del COLPOS (colegio de posgraduados) en el departamento de fitotecnia y se instaló el prototipo durante

un periodo de tiempo en una maqueta donde se colocó la tira de led y 4 biorreactores (2 en cada sector) con explantes de fresa. En la *ilustración 41* se muestra el prototipo montado en el SIT del COLPOS.



ILUSTRACIÓN 41 PROTOTIPO I COLOCADO EN EL SIT DEL COLPOS

Se conectó una bomba de vacío de $\frac{1}{4}$ hp y dos electroválvulas, se colocaron horarios de bombeos que se diferenciaban por 6 horas y todos con una duración de 60s para cada sector, así mismo se asignó la iluminación, en el sector uno tenía 4h color azul, 6h blanco y 8h apagado, por el otro lado el sector dos se le programó 4h blanco, 6h azul y 8h apagado; todo esto durante un mes. En la *ilustración 42* se muestran los

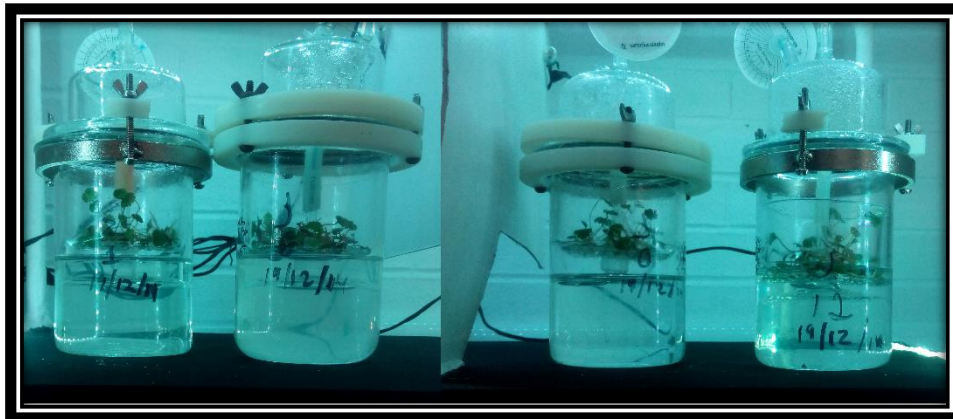


ILUSTRACIÓN 42 BIORREACTORES EN EL SIT AUTOMATIZADO

explantes de los biorreactores puestos en el sistema automatizado y en la *ilustración 43* se colocó la imagen de biorreactores que no estaban presentes en la automatización puestos el mismo día.



ILUSTRACIÓN 43 BIORREACTORES FUERA DEL SIT AUTOMATIZADO

Después de un mes se obtuvo el mejor biorreactor (ver capítulo IV: Análisis y resultados) del sistema automatizado donde se obtuvo 131 brotes a partir de 10 explantes; este biorreactor estuvo colocado en el sector 2 donde se expuso un mayor tiempo a la iluminación azul, en la *ilustración 44* se muestra el mejor biorreactor en el

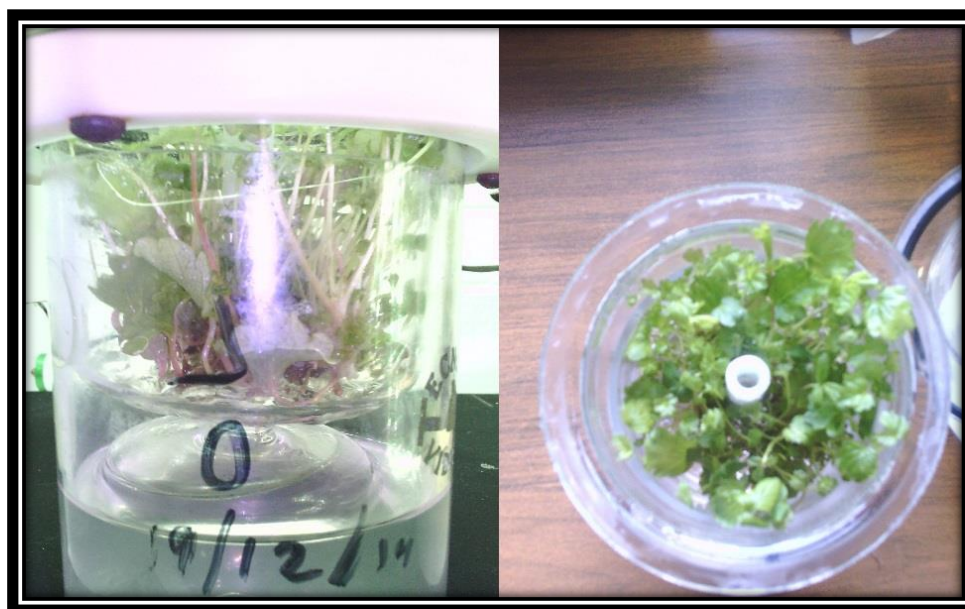


ILUSTRACIÓN 44 MEJOR BIORREACTOR CON LA AUTOMATIZACIÓN

SIT automatizado, por otro lado el mejor biorreactor del sistema sin automatizar mostrado en la *ilustración 45* no logra un desarrollo similar, siendo posible a simple vista determinar que el número de brotes es menor al biorreactor en la automatización.



ILUSTRACIÓN 45 EL MEJOR BIORREACTOR SIN LA AUTOMATIZACIÓN

La implementación y los resultados obtenidos del prototipo I fueron satisfactorios al notar que la automatización optimiza el desarrollo de los explantes, proporcionando un mayor número de brotes de mayor calidad, por lo tanto, se puede determinar que el sistema de automatización sirve y que mejora la producción a diferencia de la forma tradicional.

A lo largo del mes de prueba del prototipo I se presentaron varios puntos que se sometieron a arreglar o mejorar; los problemas presentados en la implementación de la automatización fueron:

1. Falta de energía eléctrica: Si bien todos los sistemas digitales deben de abastecerse de energía eléctrica, es ideal que el usuario pueda saber el estado de la automatización en caso de este tipo de percances o en cualquier



- momento, por lo que se propuso desarrollar un medio de alertas y monitoreo remoto.
2. Parada inesperada del programa de control: Un problema presentado en la implementación fue el bloqueo del sistema cuando en un par de ocasiones la activación de la bomba bloqueaba la placa Arduino, ocasionando que se perdieran bombeos cuando el usuario estaba ausente, nuevamente surge la necesidad de un sistema de monitoreo remoto.
 3. Ingresar horarios a la iluminación: Otro aspecto importante que no se tomó en cuenta en el prototipo I fue la calendarización de horarios de iluminación accesible al usuario, si bien el prototipo I tenía horarios programados, la edición, creación y eliminación de estos no estaban accesibles al usuario, por lo que se vio la necesidad de implementar un registro similar a los bombeos donde el usuario almacenara los horarios y color de luz del SIT.

Los problemas mencionados anteriormente se resuelven en el desarrollo del prototipo II.

Prototipo 2

El desarrollo del prototipo 2 se enfocó hacia la mejora y optimización del software de control; el cual se comunica con el usuario por medio de una interfaz y la vez con la placa Arduino; este último con el uso de un cable USB; así mismo, se desarrolló una aplicación para dispositivos Android, con la finalidad de conocer el estado del cultivo en tiempo real aun sin estar presente en el SIT. Para simplificar la comprensión y alcance del desarrollo, en la *ilustración 46* se muestra el diagrama UML de los procesos que intervienen en el sistema de prueba.

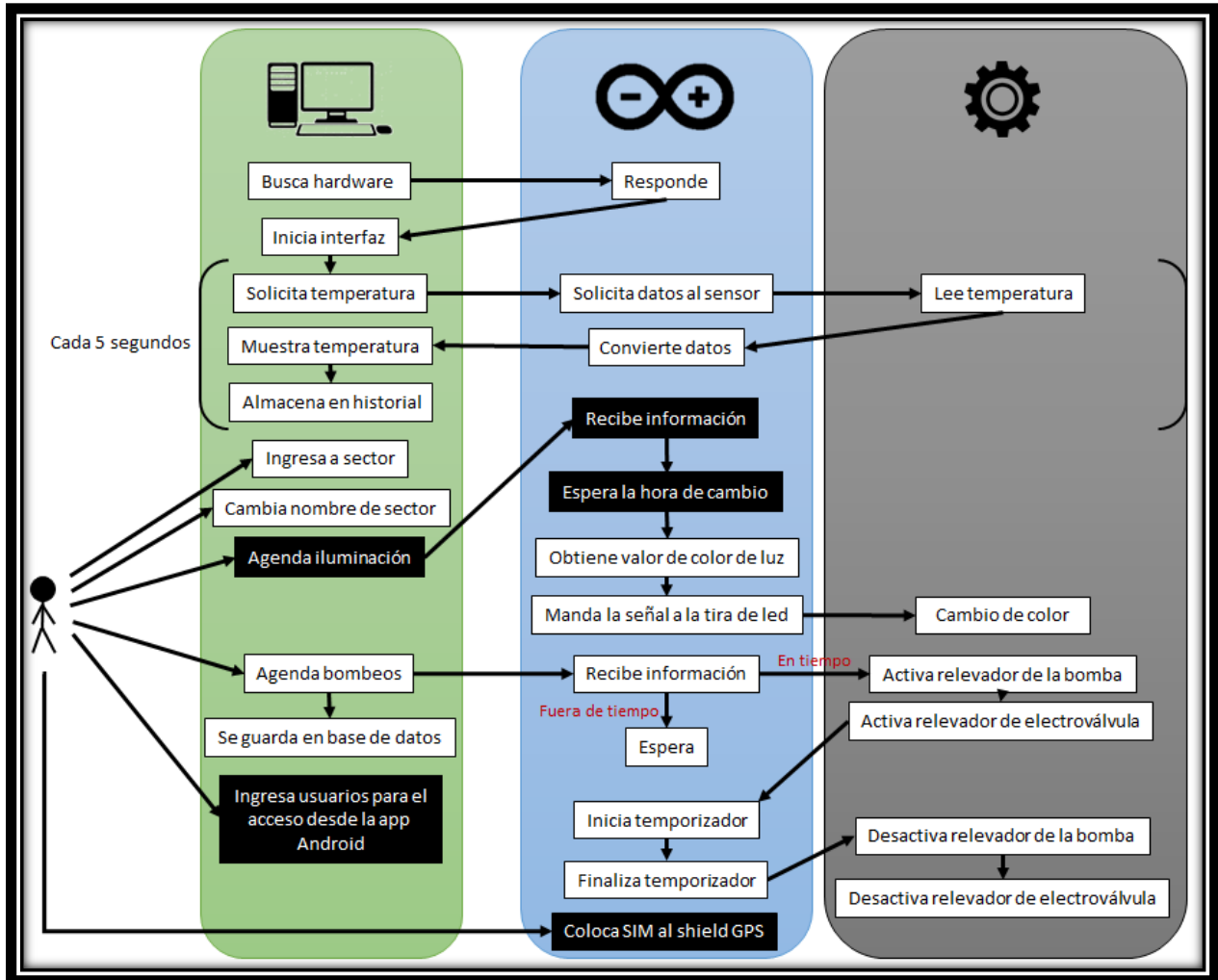


ILUSTRACIÓN 46. UML DEL PROTOTIPO 2

Desarrollo del Hardware

Al tener las conexiones completas y necesarias para automatizar el SIT del prototipo I, se observó que los recursos ofrecidos por la placa Arduino Mega 2560 eran más de lo que se necesitaba, por lo que se decidió cambiar de modelo siendo este el Arduino UNO. El cambio de placa Arduino no afectó el funcionamiento de los sensores y dispositivos electrónicos, sin embargo, se logró un mejor costo, ya que una placa Arduino UNO es más económica que una Mega 2560, además se ocupó un menor espacio y los puertos en la nueva placa son los suficientes para la automatización,

evitando el desperdicio de recursos que se tenía en el Arduino Mega 2560. En la *ilustración 47* se muestra las conexiones de los elementos para la automatización en la placa Arduino UNO.

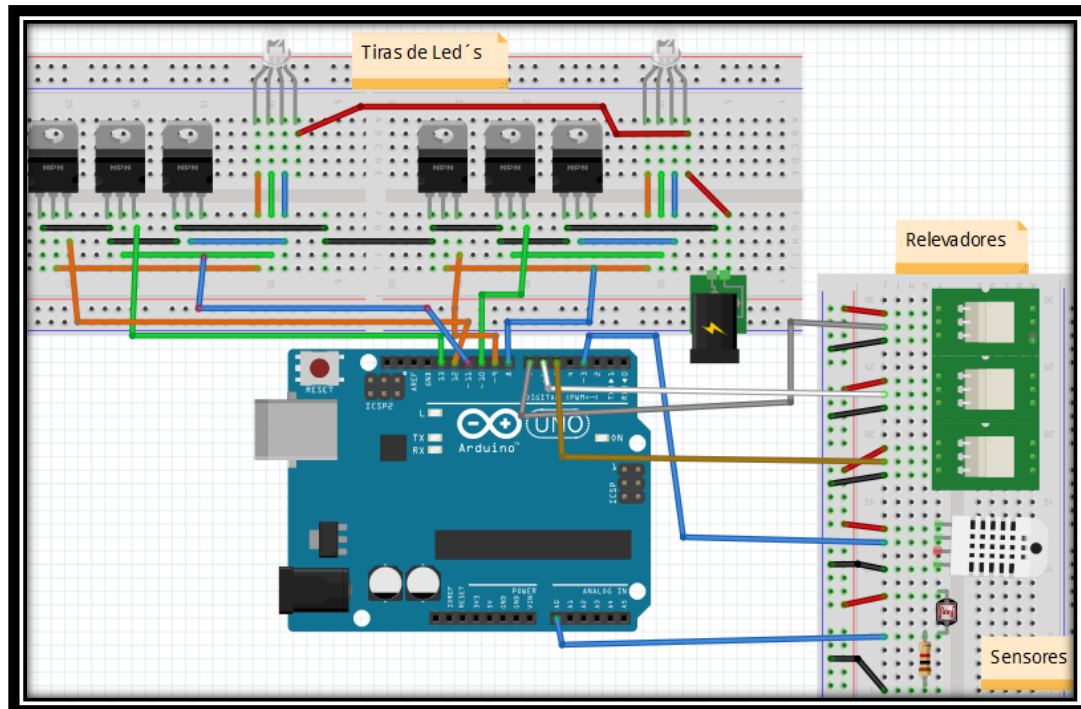


ILUSTRACIÓN 47 ELEMENTOS DE HARDWARE EN ARDUINO UNO

Para la creación de un software de control remoto, se usó la tecnología GSM, por lo que se adquirió una shield GSM en la que se colocó una tarjeta SIM, la cual fue montada en el Arduino UNO ya que tiene compatibilidad directa con el dispositivo de comunicación, en la *ilustración 48* se muestra la placa GSM montada en el Arduino UNO.

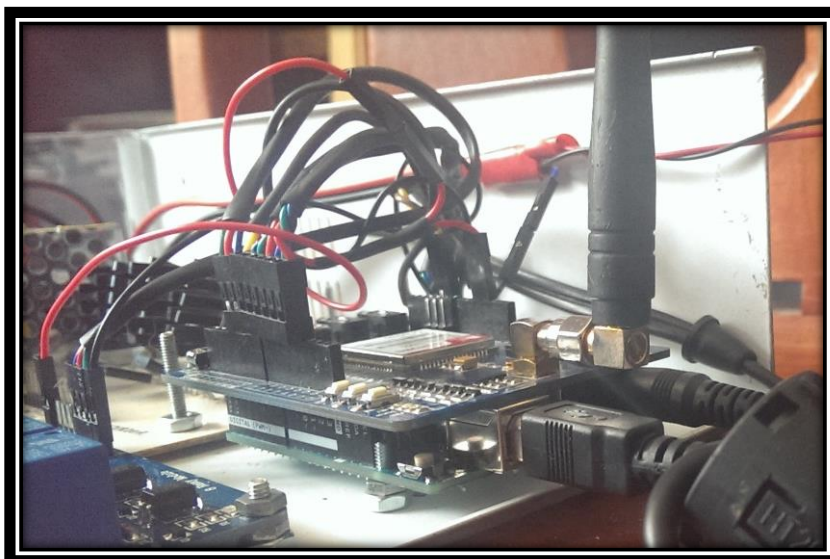


ILUSTRACIÓN 48 SHIELD GSM EN ARDUINO UNO

Un problema presentado en la implementación del prototipo I es el detenimiento del sistema cuando en un par de ocasiones la bomba se activaba, investigando se encontró que al activar la bomba, este crea un campo electromagnético que genera interferencias al Arduino, por lo que se buscó un método para evitar la reacción mencionada. Como mencionan Balcells y Daura (1992) los núcleos de ferrita se emplean como elementos individuales para la supresión de interferencias, por ejemplo, si un conductor por el que circula una corriente de alta frecuencia que se desea suprimir, se rodea una zona de su trayecto por un núcleo de ferrita, creando una impedancia sin que se atenúen las frecuencias más bajas o deseadas; por lo tanto se colocó un anillo de ferrita en el cable de comunicación entre Arduino y la computadora.

Desarrollo del software

En el desarrollo del software para el prototipo II se hicieron correcciones y mejoras en el programa Java, posteriormente se desarrolló una aplicación para celulares con



sistema operativo Android, siendo este último el software de control remoto que interactuara con los elementos del SIT por medio de la tecnología GSM.

Base de datos

Una mejora a aplicar a la automatización del SIT es la creación de un sistema de horarios para el cambio de iluminación, por lo que se tuvo que editar la base de datos para almacenar la información que el usuario ingrese, para esto se creó una nueva tabla con sus campos para guardar los datos; la tabla se muestra a continuación:

Iluminación: en esta tabla se almacenan los horarios de cambio de iluminación que el usuario ingrese por medio de la GUI en la aplicación de escritorio Java. Con esta información se activa o se cambian los colores de las luces en la tira led.

- IdTabla: campo donde se almacena un identificador único de registro de valor numérico, este valor es el atributo llave de la tabla, así mismo se configuro como auto incrementable, es decir que cada vez que se ingresa un nuevo registro, este adquiere automáticamente un identificador único.

- IdSector: en este campo se guarda el id del sector como un valor numérico, si es el primer sector en el que se guarda e horario se escribe 1 y si es el segundo sector donde se configura la iluminación se almacena un 2, esto para diferenciar el sector donde se haya presentado la alteración lumínica.

- Hora: almacenamiento de la hora, proporcionada por el usuario, en la que se realiza el cambio de iluminación, en el formato de 24h.

- Tipo: se almacena un valor entero del 0 al 4 donde 0 indica que las luces están apagadas, 1 iluminación blanca, 2 indica iluminación roja, 3 iluminación azul y 4 que indica iluminación verde.

Para la creación de un software remoto es necesario de métodos de seguridad que no permitan a personas no autorizadas, por lo que se optó hacer un registro de usuario, donde el encargado de la aplicación Java ingresara la información de los usuarios que tienen permiso de manipular el SIT a partir de un control remoto. A continuación se muestra la tabla creada en la base de datos del SIT automatizado para el almacenamiento de la información de usuarios:

Usuario: tabla donde se guarda la información de los usuarios que pueden acceder al SIT de forma remota usando la aplicación para dispositivos móviles Android, usando esta información se desarrollaron métodos de validación de usuario.

- Id: campo donde se almacena un identificador único de registro de valor numérico, este valor es el atributo llave de la tabla, así mismo se configuro como auto incrementable, es decir que cada vez que se ingresa un nuevo registro, este adquiere automáticamente un identificador único.
- Nombre: nombre del usuario quien podrá controlar de forma remota la automatización del SIT, por medio de la aplicación móvil.
- Apellidos: apellidos del usuario quien podrá controlar de forma remota la automatización del SIT, por medio de la aplicación móvil.



- Teléfono: número telefónico celular con sistema operativo Android, donde se instalara el control remoto.
- Correo: correo electrónico de usuario donde se manda una fotografía vía internet sobre el estado del SIT.

En la *ilustración 49* se muestra el diagrama y las relaciones de la base de datos usada para almacenar la información del prototipo 2.

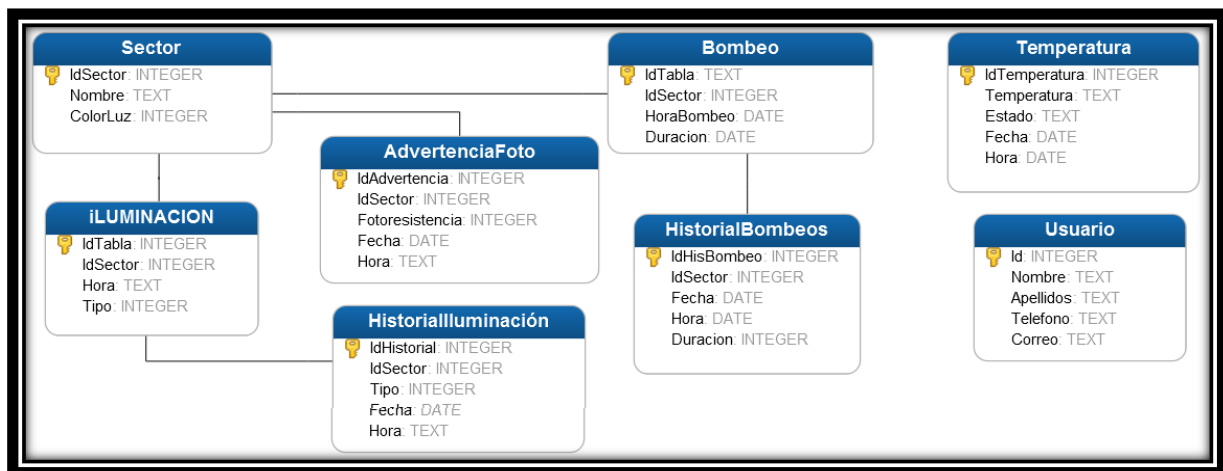


ILUSTRACIÓN 49. DIAGRAMA DE BASE DE DATOS, PROTOTIPO 2

GUI

Para el prototipo 2 se modificó la ventana principal con la suma de un botón, el cual accede a una nueva ventana para la consulta de usuarios que están permitidos a interactuar con el SIT usando la aplicación remota. En la *ilustración 50* se muestra la ventana principal actualizada.



ILUSTRACIÓN 50 VENTANA PRINCIPAL EN EL PROTOTIPO II

La ventana para la consulta de usuarios despliega en una tabla la información almacenada en la base de datos, donde la primera columna muestra un id único de registro, seguido del nombre de usuario, en la tercera columna se muestran los apellidos, en la cuarta el número telefónico y finalmente el correo electrónico de usuario, además, en la ventana existe un botón para agregar un nuevo usuario, otro para eliminar un usuario seleccionándolo en la tabla, otro para actualizar la tabla y finalmente un botón para cerrar la ventana. En la *ilustración 51* se muestra la ventana de usuarios registrados en el SIT.

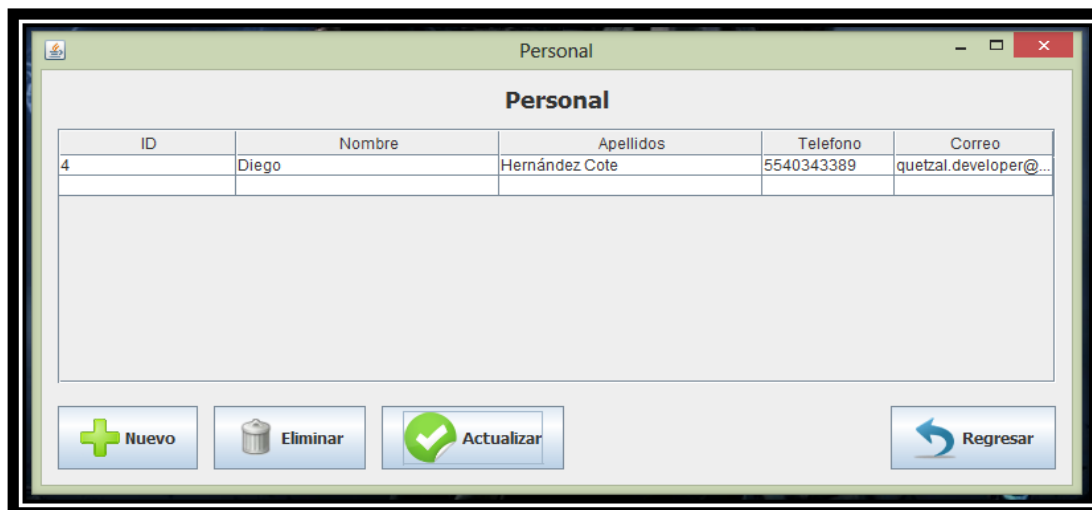


ILUSTRACIÓN 51 VENTANA DE USUARIOS

Cuando se desea agregar un nuevo usuario se debe acceder a la ventana de nuevo usuario desde el botón de la ventana personal. En la ventana nuevo usuario mostrada en la *ilustración 52* se solicita al interesado la información de la persona a ser registrada, iniciando por el nombre, apellidos, teléfono y correo; esta ventana cuenta con los métodos de validación listados a continuación:

- El nombre debe contener únicamente caracteres alfabéticos
- El apellido debe contener únicamente caracteres alfabéticos
- El teléfono debe contener únicamente caracteres numéricos
- El teléfono solo debe contar con 10 dígitos
- El correo debe tener un símbolo arroba (@)
- El correo debe terminar en .com
- El correo solo puede contener caracteres permitidos (a-z, A-Z,0-9,.,-,_) antes del arroba
- El correo solo puede contener caracteres permitidos (a-z, A-Z,0-9) después del símbolo arroba

- El correo no permite espacios

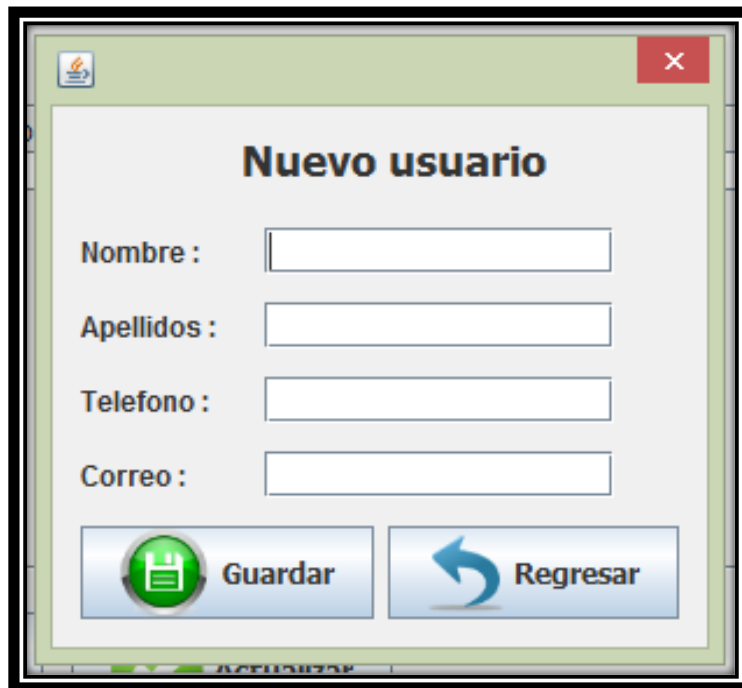


ILUSTRACIÓN 52 VENTANA DE NUEVO USUARIO

Por otra parte, en la ventana de monitoreo de sector se le agregó un botón de iluminación y otro botón de bombeo instantáneo. En algunas ocasiones en el SIT del COLPOS se solicitaba un bombeo inmediato en los birreactores, por lo que tenía que ser registrado y eliminado ya que solo era necesitado una sola vez, por lo que se vio la necesidad de implementar un medio por el cual el usuario pudiera realizar una inmersión de forma inmediata sin necesidad de registrar una hora o duración, para esto se colocó un botón de bombeo instantáneo que al oprimirlo se activarán los relevadores y al recibir una segunda pulsación la inmersión cesará. En la *ilustración 53* se muestra la ventana de monitoreo de sector con las modificaciones realizadas.



ILUSTRACIÓN 53 VENTANA DE MONITOREO POR SECTOR EN EL PROTOTIPO II

Para el registro de horarios de la iluminación es necesario abrir la ventana de horarios, el cual es accesible al presionar el botón de iluminación en la ventana de monitoreo por sector. En la *ilustración 54* se muestra la ventana de horarios orientada a la iluminación, donde aparece una tabla con los horarios actuales mostrando en la primera columna el id de registro, en la segunda columna se despliega la hora de cambio y finalmente en la tercera columna se muestra el id del color a cambiar, también existen tres botones, uno para agregar un nuevo horario, otro para eliminar un bombeo seleccionándolo de la tabla y un botón para cerrar la ventana.

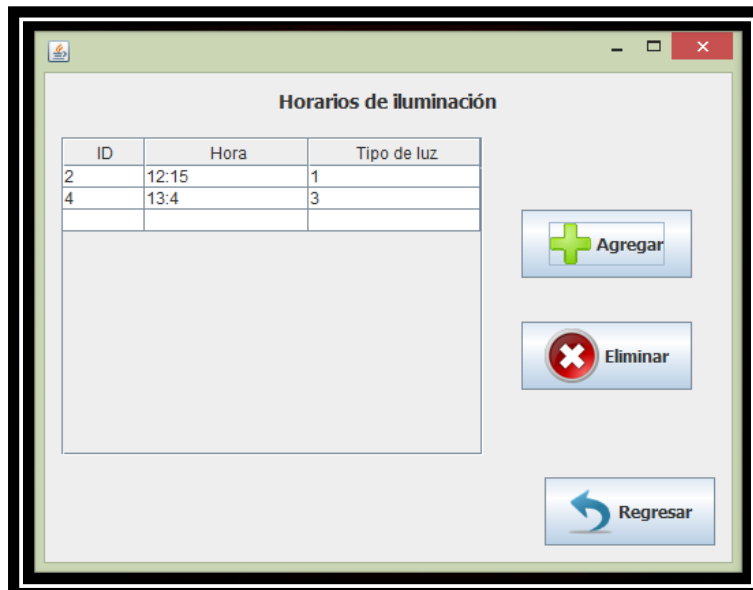


ILUSTRACIÓN 54 VENTANA DE HORARIOS DE ILUMINACIÓN

Cuando se presiona el botón agregar de la ventana de horarios de iluminación, se accede a una ventana que solicita los datos necesarios para almacenar un nuevo registro, los datos solicitados son la hora, minutos y la selección del color que se desea aplicar, cabe mencionar que los métodos de validación de ingreso de hora y minutos son los mismos a los presentados en la ventana de nuevo horario de bombeo realizado en el prototipo I; finalmente se encuentran dos botones, uno para guardar la información en la base de datos y otro para cancelar la operación. En la *ilustración 55* se muestra la ventana para agregar un nuevo horario.



ILUSTRACIÓN 55 VENTANA DE NUEVO HORARIO DE ILUMINACIÓN

Una aplicación importante en el presente proyecto de investigación es el desarrollo de una herramienta que permita la comunicación e interacción remota con la automatización del SIT, para esto se creó una aplicación para dispositivo celulares con sistema operativo Android y aprovechar el acceso de la tecnología GSM que se tiene en la plataforma.

Para la aplicación Android en el presente prototipo se creó un layout¹² principal donde se mostrara la información del SIT cuando sea solicitado por el usuario, los datos mostrados son la temperatura actual, el color de la iluminación en el sector uno y el color de iluminación en el sector dos, además se muestran dos imageButton (botones en Android que tienen como forma una imagen seleccionada por el desarrollador) uno para pedir la información al SIT y otro para editar el número telefónico de la tarjeta SIM ubicada en la shield GSM de la placa Arduino.

Para una mejor apariencia de la aplicación, cuando se recibe la respuesta de información acerca del color de luces, el texto cambia de color respecto al estado del SIT y además se le agrego un menú deslizante para acceder a los botones y evitar

¹² Un layout define como una estructura visual para una interfaz de usuario, como puede ser la GUI para una actividad o un widget (Android, 2015).

una saturación de elementos en la pantalla principal. En la *ilustración 56* se muestra la interfaz principal de la aplicación Android.



ILUSTRACIÓN 56 INTERFAZ PRINCIPAL DE LA APLICACIÓN ANDROID

Cuando se elige el botón para el cambio de número se abre un nuevo layout que se muestra en la *ilustración 57*, donde se solicita el número telefónico del SIT, esta ventana tiene métodos de validación, donde se verifica que el teléfono cumpla con las siguientes características:



ILUSTRACIÓN 57 LAYOUT DEL NÚMERO TELEFÓNICO



- Solo son aceptados números
- El numero debe ser de 10 dígitos

Cuando el usuario presiona el botón de información del layout principal se le advierte que se ha mandado el mensaje de texto al SIT automatizado para que este último le regrese los datos pedidos, la advertencia se realiza empleando una función nativa de Android llamado *toast* que es (Kurniawan, 2016) una diminuta ventana emergente que imprime en pantalla un mensaje de retroalimentación para el usuario y no reemplaza la aplicación, ocupando solo el espacio necesario que necesita el mensaje. En la *ilustración 58* se muestra un *toast* con el texto "Petición enviada" en la aplicación Arduino.



ILUSTRACIÓN 58 TOAST CON EL TEXTO "PETICIÓN ENVIADA"

Procesos del software

En la aplicación de control Java se le integró un nuevo hilo que se encarga de realizar el cambio de iluminación en el SIT, comparando los horarios establecidos por el usuario y la hora actual, cuando las horas coinciden se cambia la iluminación y el valor del color de luz es guardado en un archivo de texto llamado "luz.txt" para reiniciar el hilo, si no coinciden los horarios el hilo se ejecuta inmediatamente. Cada vez que el hilo de iluminación reinicia se busca el valor del color de luz en el archivo "luz.txt" para reenviarlo a la placa Arduino. En la *ilustración 59* se muestra el diagrama del hilo de iluminación.

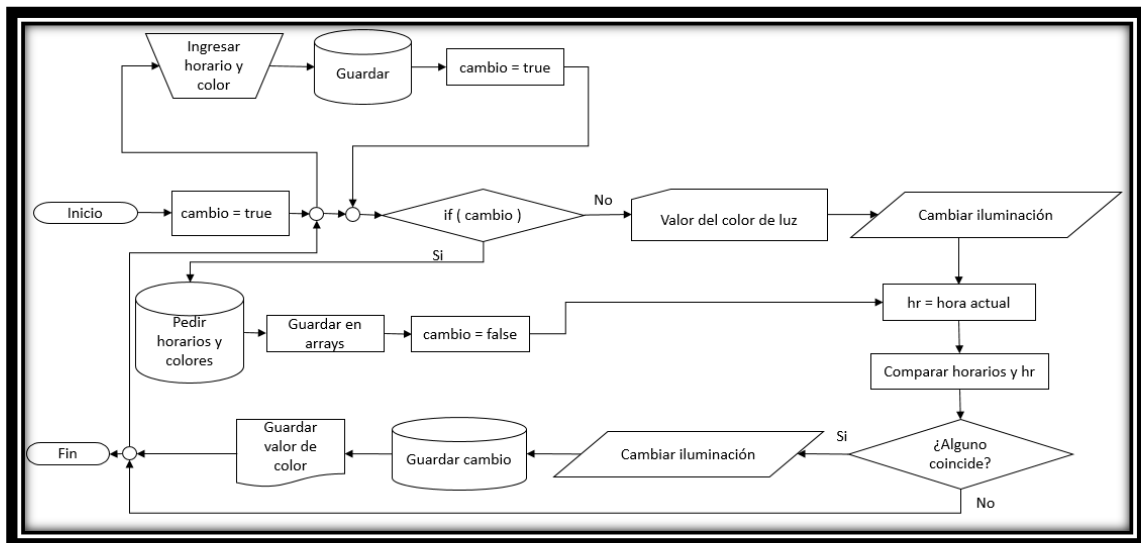


ILUSTRACIÓN 59 DIAGRAMA DEL HILO DE ILUMINACIÓN

Para la aplicación de control remoto en Android se usó la librería SmsMessage el cual permite el envío de mensajes de texto desde la aplicación y a la vez, permite la lectura de los mensajes; para la lectura de los mensajes se limitó a solo permitir la manipulación de los SMS enviados desde el número del SIT, de esta forma, se evita que la aplicación se abra cada vez que un mensaje llegue al dispositivo móvil. En la

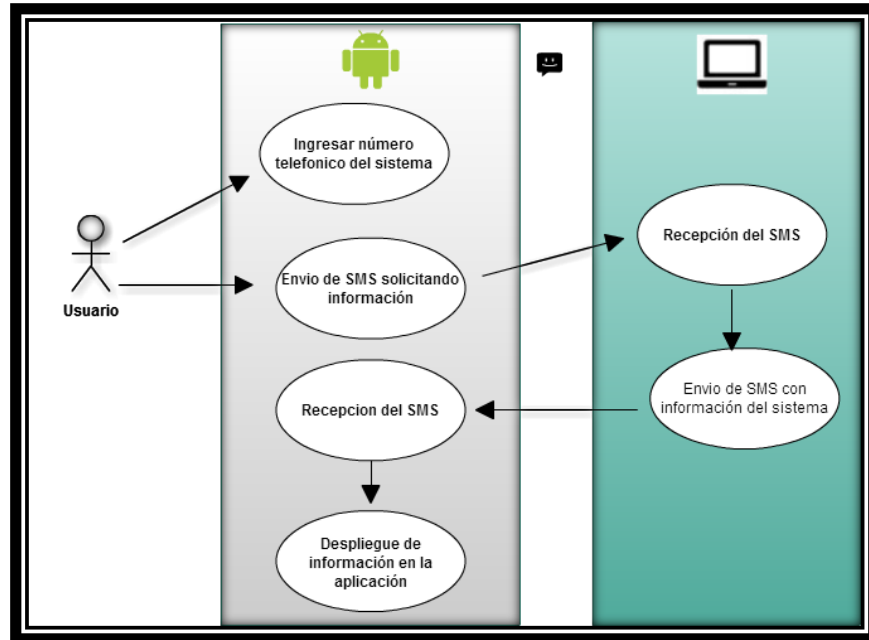


ILUSTRACIÓN 60 DIAGRAMA DE CASOS DE USO DE LA APLICACIÓN ANDROID

ilustración 60 se muestra un diagrama de casos de uso del funcionamiento del control remoto.

Cuando el usuario manda el mensaje de texto solicitando información, se envía la cadena “Info”, cuando el shield de la placa Arduino del SIT recibe el mensaje se obtiene el número telefónico del dispositivo móvil y el cuerpo del mensaje, el número se envía al programa de control Java el cual verifica su existencia en la base de datos de usuarios, si el número telefónico existe en la base, entonces le indica a Arduino que puede enviar una respuesta al dispositivo móvil. La respuesta del SIT al usuario cuando solicita información se compone de dos partes, la primera es una imagen del SIT que es obtenida desde una cámara WEB y la otra es una cadena de texto que se envía vía SMS.

La primera parte de la respuesta se crea en el programa de control en Java; cuando el número telefónico es consultado en la base de datos, si existe en los registros entonces se solicita su correo electrónico, posteriormente se toma una fotografía con

la cámara WEB predeterminada en la computadora utilizando una herramienta llamada OpenCv, el cual es (Bradski & Kaebler, 2008) una librería de código abierto para visión computacional, diseñada para la eficiencia en las aplicaciones de tiempo real que interactúa con cámaras conectadas en el sistema, logrando así activar una cámara WEB y tomar una captura de pantalla en tiempo real; esta imagen es almacenada en una carpeta ubicada en el directorio raíz de la aplicación Java y es enviada por correo electrónico con el asunto SIT. El nombre de la imagen se compone de la fecha y hora de su captura con el siguiente formato:

Año-mes-dia-hora-minutos-segundos.jpg

La parte de la respuesta que se envía al dispositivo móvil se crea en la placa Arduino después de recibir la confirmación de la existencia del número telefónico en la base de datos, este se compone de una cadena de texto que es enviada vía SMS en el siguiente formato de ejemplo:

A27.44/3#2\$

Cuando la aplicación Android recibe el mensaje, verifica que el número sea del SIT, si lo es, entonces abre la aplicación y se recorre carácter por carácter el cuerpo del mensaje de la siguiente forma: si la cadena de texto empieza con la letra “A” entonces significa que pertenece a una respuesta de información, posteriormente se leen los siguientes números, los cuales serán el valor de la temperatura, en este ejemplo es 27.44; cuando se detecta el carácter “/”, el siguiente dígito es el valor del color de la iluminación en el sector 1, en este caso el número 3 pertenece al color azul; el siguiente carácter es “#” por lo que su siguiente carácter es un dígito correspondiente al color del sector 2, en este ejemplo el número 2 equivale al color rojo, finalmente se obtiene el símbolo “\$” que indica el fin del texto.

Al tener los valores de la temperatura y los colores de los sectores, esta información es desplegada en la ventana inicial de la aplicación, así mismo se coloca el texto “*Foto enviada a correo*” para indicarle al usuario que se ha tomado una captura y ha sido enviada a la dirección del correo electrónico que se ha registrado en la base de datos en el programa de control Java.



Cada vez que el SIT es reiniciado o la placa Arduino es encendida, se envía un mensaje a la aplicación remota de Android indicándole que el sistema ha sido reiniciado, haciéndole saber al usuario que en caso de no haber programado una suspensión o apagado la automatización, ha ocurrido algo ajeno al sistema que ha provocado un reinicio del SIT automatizado; en este caso, el mensaje enviado lleva como cuerpo el carácter “B”, de esta forma, al ser procesado el mensaje en la aplicación Android se desplegara el texto “Sistema reiniciado” en la pantalla.

Implementación y resultados

Las mejoras hechas en el prototipo II fueron puestas a prueba programando horarios de cambios de luz y horarios de activación de relevadores, así mismo se puso en prueba la aplicación para dispositivos móviles Android para la solicitud de información y monitoreo del SIT automatizado.

Tras los resultados del primer prototipo se pudo constar que la automatización optimiza el crecimiento de las plantas, por lo que no hubo la necesidad de implementarlo nuevamente en el SIT del COLPOS. Para realizar las pruebas de funcionamiento del sistema se empleó la misma maqueta hecha para el prototipo uno y crear unos biorreactores caseros, los cuales al activar un bombeo daban constancia que el sistema hacia sus tareas adecuadamente. En la ilustración 61 se muestra la maqueta montada para las pruebas.

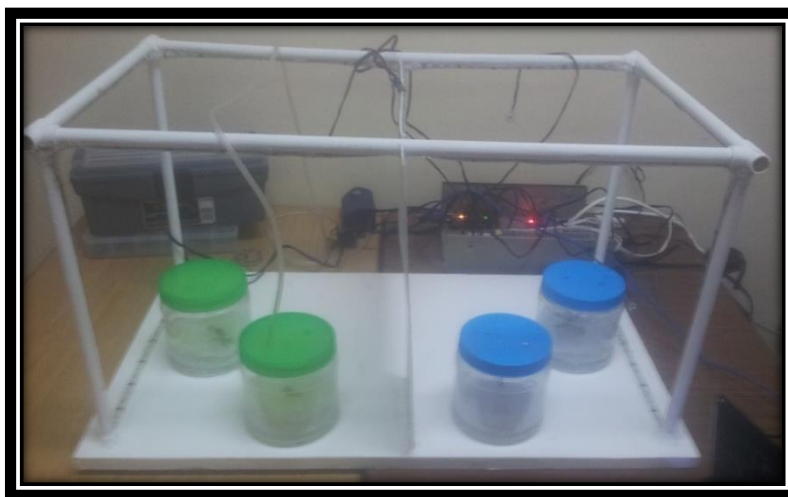


ILUSTRACIÓN 61 MAQUETA DE PRUEBAS DEL PROTOTIPO II

La primer prueba realizada fue el encendido y apagado de una bomba de aire varias veces al día de forma calendarizada e instantánea, esto para verificar que el campo electromagnético generado por la bomba no interfiriera con los procesos de la placa Arduino y se bloqueara el sistema, como resultado se logró la eliminación del bloqueo y un correcto funcionamiento del sistema de inmersión, por lo que el anillo de ferrita aplicado al cable de comunicación Arduino – computadora tuvo un efecto positivo al consumir la interferencia proveniente de la bomba de aire. En la *ilustración 62* se muestra el anillo de ferrita cubriendo el cable de comunicación.

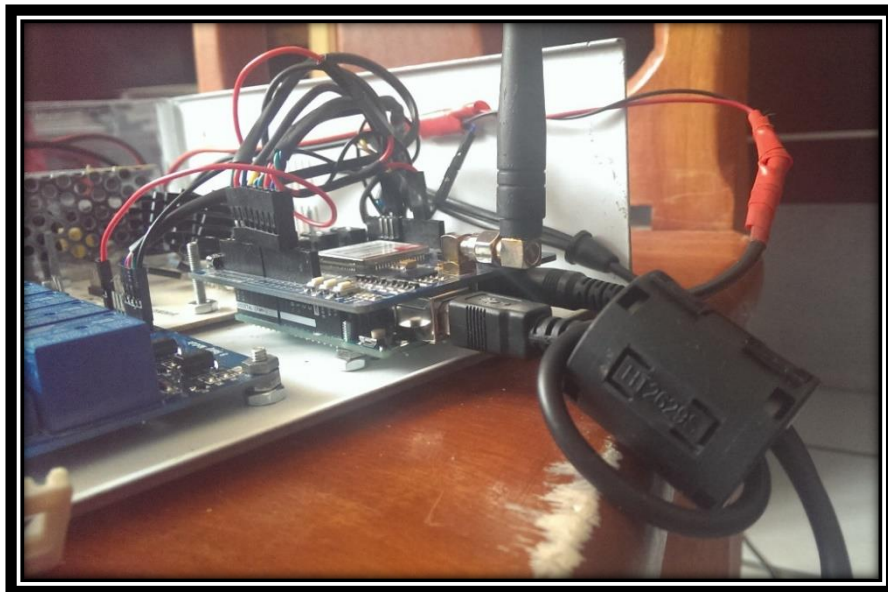


ILUSTRACIÓN 62 ANILLO DE FERRITA CONECTADO AL CABLE DE COMUNICACIÓN

Estas pruebas realizadas en el encendido y apagado de la bomba de aire lograron eliminar el problema numero dos mencionado en el prototipo I, logrando una mejor estabilidad en el sistema de automatización.

La segunda prueba realizada fue la aplicación de horarios en el programa de control Java para el cambio de iluminación, con esto se completó el control de los dispositivos

de hardware del SIT, ya que se realizaron las ediciones de color en la luz de forma exitosa a las horas que fueron programadas. En la *ilustración 63* se muestra el cambio de luz de un sector de verde a blanco aplicando el uso de horarios ingresados en el sistema de control Java y en la *ilustración 64* se muestra el cambio simultáneo de luz de los dos sectores.

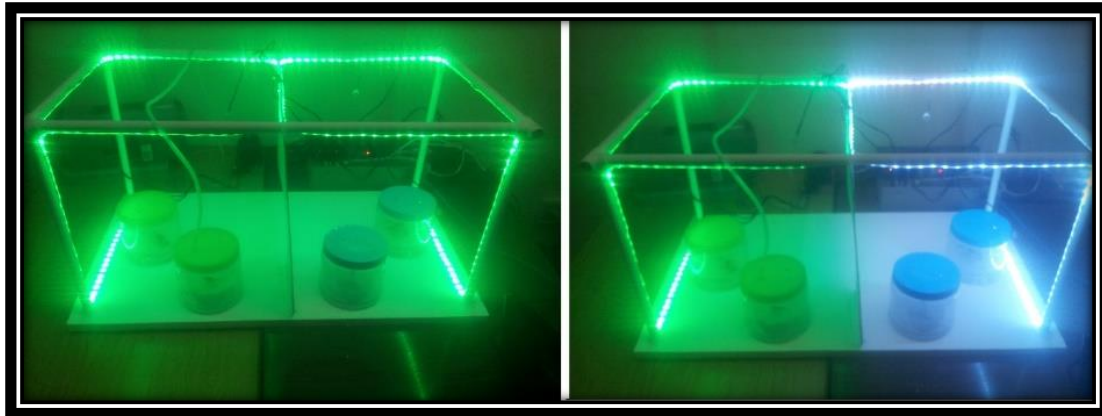


ILUSTRACIÓN 63 CAMBIO DE LUZ EN UN SECTOR DEL SIT

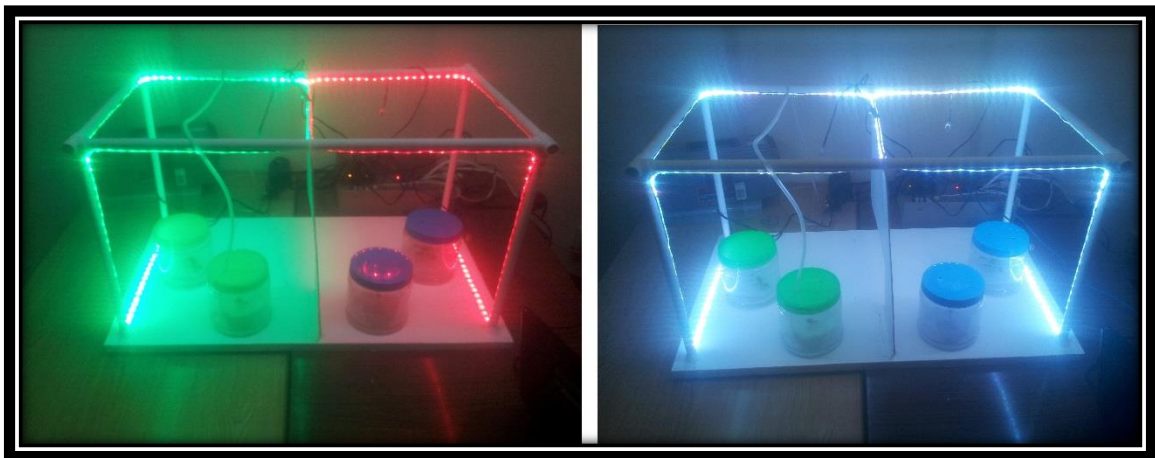


ILUSTRACIÓN 64 CAMBIO SIMULTANEO DE COLORES EN EL SIT

Con la segunda prueba que se realizó a la automatización del SIT, se logró solucionar el tercer problema encontrado en el prototipo I, con la aplicación de una interfaz efectiva para el usuario y con el desarrollo de un hilo que logra de forma exitosa la ejecución de los cambios de iluminación.

La tercera prueba se basó en la interacción de la aplicación Android de forma remota con el SIT, cuando se encendió la placa Arduino este como primera operación envió el mensaje de texto que indica el reinicio del sistema. En la *ilustración 65* se muestra la pantalla de la aplicación cuando se recibe el mensaje de reinicio del SIT automatizado.

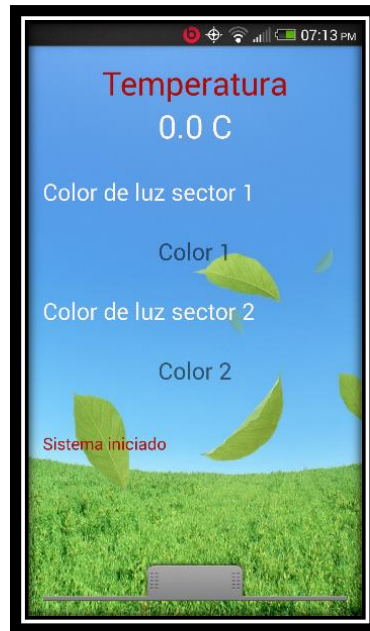


ILUSTRACIÓN 65 RECEPCIÓN EN ANDROID DEL MENSAJE DE REINICIO

Después de recibir el mensaje de reinicio, se solicitó la información del estado actual del SIT, por lo que se oprimió el botón de información en la aplicación Android para que Arduino regresara la respuesta de la petición enviada; en primer lugar, se recibió a la aplicación móvil el estado del SIT, posteriormente se recibió por correo la imagen del sistema. En la *ilustración 66* se muestra el estado del SIT cuando se solicitó la información desde la aplicación Android y en la *ilustración 67* se muestra lo que se recibió de la petición, del lado izquierdo en la aplicación móvil y del lado derecho la imagen recibida por correo electrónico de la cámara WEB.



ILUSTRACIÓN 66 ESTADO DEL SIT CUANDO SE SOLICITÓ LA INFORMACIÓN DESDE ANDROID



ILUSTRACIÓN 67 RESPUESTA ENVIADA DESDE EL SIT

Esta última prueba, aunque no resuelve el problema número uno del prototipo I, se implementó un mensaje de alerta de reinicio que avisa al usuario de cualquier corto de energía, ya que al regresar la electricidad activa el Arduino y envía el mensaje de inicio.

Durante la implementación del prototipo II se detectaron situaciones que son necesarios de cubrir por lo que es factible la realización de un nuevo prototipo, los ajustes necesarios son:

1. La aplicación Android no se puede considerar de control, ya que solo pide información, pero no altera el SIT de forma remota.
2. Se debe implementar métodos de validación cuando se envía un SMS desde la aplicación móvil, ya que, si no aparece el mensaje de *petición enviada*, no se sabe que problemas hay que impiden la comunicación.
3. En el caso de que la aplicación Android no reciba respuesta durante un periodo de tiempo, se alerte al usuario ya que puede ser desde una falla en el servicio celular como una falla en la automatización, como por ejemplo un corte eléctrico.

En el presente prototipo se encontró un problema muy grave que se necesita corregir: la aplicación de la shield GSM en Arduino hace que tome bastante tiempo el revisar si hay un mensaje de llegada en la SIM, por lo que impidió en varias ocasiones la ejecución de cambios de luz y de bombeos ya que se pasaba del horario establecido y no se ejecutaba la acción; por lo tanto, es necesario encontrar una vía por la cual el tiempo que toma al Arduino el procesar las tareas GSM, no interfieran con los procesos de activación de bomba o de cambio de luz.

Prototipo final

En el desarrollo del presente prototipo se logró resolver los problemas mencionados en el prototipo 2, dando lugar a una automatización eficiente por medio de un control Java y un control remoto para dispositivos móviles, esto beneficiará al usuario ya que no necesitará estar en el SIT para manipular los bombeos o el cambio de las luces, además se puede monitorear el estado del sistema desde la aplicación Android.



Desarrollo de hardware

Un problema grave presentado en el prototipo II es la tardanza que conlleva el procesamiento de los SMS en la shield GSM en la placa Arduino, por lo que se buscó un método en el cual el tiempo asignado a la lectura de los mensajes de entrada no interfiriera con los cambios de iluminación o activación de los relevadores.

El tiempo que usa la shield GSM abarca desde que verifica que exista señal celular, posteriormente se busca si existe actividad en el servicio de la compañía celular del que se haya adquirido la tarjeta SIM y finalmente busca si hay mensajes de entrada, si existen entonces se procesa el cuerpo del mensaje para conocer la petición que se ha recibido, todos estos procesos consumen mucho tiempo y recursos de la placa Arduino y estos no pueden ser editados, así mismo, el puerto serial de Arduino se ocupa de la señal GSM, por lo que también existe señales perdidas cuando la computadora envía un cambio a Arduino, por lo tanto se deduce que el mejor camino es separar la shield GSM a una placa Arduino independiente para que no interfiera con las señales de la automatización.

La separación de la shield llevó a la implementación de otro Arduino UNO, siendo una de estas exclusivamente para los elementos de automatización del SIT y otra para uso exclusivo del GSM, la interacción entre las dos placas Arduino se lleva a cabo por el software de control Java. En la *ilustración 68* se muestran los dos Arduino uno.



ILUSTRACIÓN 68 SHIELD GSM SEPARADA EN OTRA PLACA ARDUINO

Al tener una segunda placa Arduino que se comunica con la computadora, fue necesario de implementar un segundo anillo de ferrita al cable de comunicación USB, esto se realizó para evitar que la frecuencia del campo electromagnético, hecha por la bomba de aire, afectara la comunicación entre el programa Java y la placa Arduino.

Desarrollo de software

Para el presente prototipo, la actualización del software fue en mayor medida para la aplicación móvil, ya que no cumplía con la propiedad de ser un control remoto, además de implementar un sistema de alertas para cada acción que se realizara desde la aplicación celular. Por otra parte, el software de control Java fue editado solo para comunicarse con la placa Arduino GSM implementada.



Base de datos

Para este prototipo no se necesitó de ninguna actualización en la base de datos, ya que se contaba con los campos necesarios para almacenar la información pertinente para el usuario y el sistema.

GUI

En la aplicación de control Java no se tuvo la necesidad de implementar o crear nuevas ventanas, ya que los cambios se realizaron en los procesos de software. Por otro lado, la aplicación Android sufrió de ajustes visuales para lograr la integración de los métodos de control que harían de esta aplicación un medio remoto de alterar el SIT.

Para tener un completo control del SIT de forma remota, se necesita poder alterar el estado de los componentes de la automatización, como son los relevadores (inmersión) o la iluminación led, por lo tanto, se agregaron nuevos `imageButtons` a la interfaz de la aplicación Android. En la *ilustración 69* se muestra el layout principal de la aplicación 6Android, donde se agregaron 10 nuevos botones con sus imágenes representativas; donde los dos últimos botones de la primera fila activan la bomba de aire por 30 segundos del sector que se elija, la segunda fila cambia el color de la luz led en el primer sector y los botones de la última fila cambian el color de la luz led del segundo sector.



ILUSTRACIÓN 69 LAYOUT PRINCIPAL DEL PROTOTIPO FINAL

Procesos del software

Al tener otra placa Arduino comunicándose con la aplicación de control Java, fue necesario de crear un hilo que consultara el estado de la shield GSM, por lo tanto, este hilo pide la información de los mensajes que incluye: el número telefónico de envió y tipo de petición. Dependiendo el tipo de petición recibida, el programa Java realizara los cambios y enviara una respuesta al Arduino con la shield GSM.

Cuando el Arduino con el shield GSM recibe un mensaje de la aplicación Android, separa el número telefónico y el cuerpo del mensaje, estos datos son enviados por puerto serial a la aplicación Java, el cual verifica la existencia del número telefónico en la base de datos, si existe entonces se compara el cuerpo del mensaje para obtener el tipo de petición solicitada desde Android. Las peticiones que realiza la aplicación móvil al SIT se muestran en la *tabla 8*.



TABLA 8. PETICIONES DE LA APP AL SIT

Petición	Cuerpo del mensaje
Solicitud de información	Info
Inmersión de 30s en el sector 1	B1
Inmersión de 30s en el sector 1	B2
Apagar la luz en el sector 1	L11
Luz blanca en el sector 1	L12
Luz roja en el sector 1	L13
Luz azul en el sector 1	L14
Apagar la luz en el sector 2	L21
Luz blanca en el sector 2	L22
Luz roja en el sector 2	L23
Luz azul en el sector 2	L24

Cuando la aplicación Java determina la petición, envía señales a la placa Arduino de la automatización para cambiar el estado del SIT; posteriormente se envía un mensaje de confirmación al usuario, haciéndole saber que la petición se ha procesado correctamente, para esto el programa Java envía al Arduino con GSM un carácter de confirmación, el cual es enviado como SMS a la aplicación Android solicitante.

Para el usuario es importante conocer el estado de su petición, por lo que la aplicación móvil tiene un sistema de alertas que abarca desde él envió de mensaje hasta la confirmación de la recepción del mensaje en el SIT. En el caso de envió, cuando el usuario oprime un botón en la app de Android puede desplegarse en un *toast*, a continuación, en la *tabla 9* se muestran los mensajes de alerta que pueden aparecer al enviar un mensaje:

TABLA 9. MENSAJES DE ALERTA EN LA APP

Mensaje	Significado
Petición enviada	El SMS se ha enviado correctamente
Sin comunicación	La señal celular no está presente
Sin servicio	No hay servicio de red
Envió fallido	Problema de envío no específico
La compañía celular no permite la comunicación	Bloqueo del SMS por la compañía celular, normalmente cuando no hay saldo.
PDU no definido	El mensaje no cuenta con el protocolo del SMS

Cuando él envió es exitoso y es procesado por el SIT, el sistema automatizado envía una respuesta de confirmación, en caso que el usuario no reciba la respuesta del SIT puede ser por falta de saldo en la SIM del sistema, falla del servicio celular, que el SIT este sin energía o que se esté presentando una falla. Si el usuario recibe un mensaje del SIT, la aplicación Android puede desplegar en el layout principal uno de los mensajes mostrados en la *tabla 10*.

TABLA 10. MENSAJES DEL SIT A LA APP

Mensaje	SMS recibido
Sistema reiniciado	B
Teléfono no registrado en el sistema	C
Bombeo de 30s en el sector 1 activado	D
Bombeo de 30s en el sector 2 activado	E
Foto enviada a correo	F
Luz en sector 1 cambiada	G
Luz en sector 2 cambiada	H



En el caso que no se reciba respuesta del SIT, se desarrolló una notificación en la aplicación Android que en el transcurso de 6 minutos se activará para avisarle al usuario que no se ha recibido un mensaje de respuesta del SIT. En la *ilustración 70* se muestra la notificación en la aplicación Android cuando no se recibe respuesta del SIT.

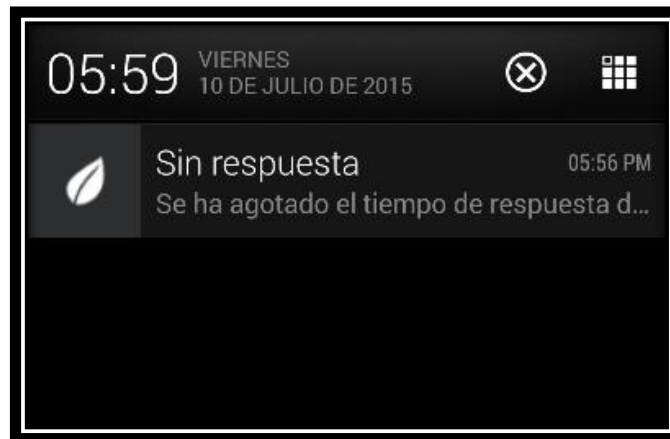


ILUSTRACIÓN 70 NOTIFICACIÓN SIN RESPUESTA DEL SIT

Implementación y resultados

Se realizaron las pruebas volviendo a montar la maqueta del SIT para verificar el funcionamiento de la automatización; la primera prueba realizada fue el funcionamiento de los dispositivos y la aplicación Java en usar dos Arduinos resultando en una buena comunicación entre ellos, se volvió a encender la bomba de aire varias veces para verificar el funcionamiento de los anillos de ferrita en los cables de comunicación resultando en ningún bloqueo. En la *ilustración 71* se muestra los dos Arduinos Uno trabajando de manera conjunta.

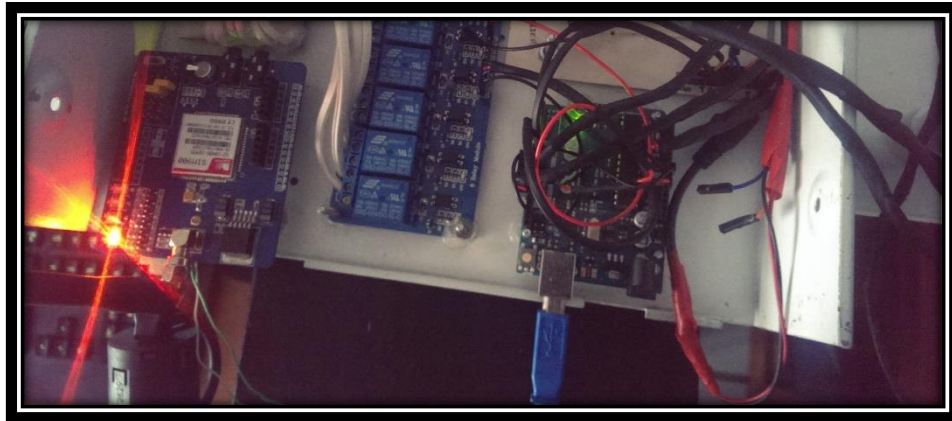


ILUSTRACIÓN 71 ARDUINOS UNO DEL PROTOTIPO FINAL EN FUNCIONAMIENTO

En esta primera prueba se eliminó el problema más importante mostrado en el prototipo 2, ya que los elementos de la automatización corresponden a un Arduino separado a la shield GSM, evitando un desfase de tiempo en los horarios de inmersión y cambios de luz, así mismo dejaron de existir las señales perdidas presentadas en la implementación del pasado prototipo.

Como segunda prueba se hicieron peticiones desde la aplicación Android al SIT automatizado, donde se probaron cada uno de los botones agregados en el presente prototipo y verificando las acciones en el SIT, teniendo como resultado una comunicación completa entre el dispositivo móvil y Arduino, ya que todas las peticiones que se solicitaron fueron realizadas.

Con la segunda prueba se resolvió el primer problema presentado en el prototipo II, ya que se desarrolló un control remoto por medio de mensajes SMS mandados vía GSM desde una aplicación para dispositivos móviles con sistema operativo Android.

Cuando las peticiones enviadas desde Android se cumplían en el SIT, se recibían respuestas de confirmación, resolviendo el problema dos del prototipo II, donde surgía la necesidad de implementar métodos de validación tanto para el envío de mensajes como de confirmación de la recepción en la automatización en el SIT, así mismo se comprobó las alertas cuando no hay señal de red, cuando no hay saldo y



cuando no hay servicio, resultando en una correcta aplicación de los mensajes de alerta.

Finalmente, se apagó el sistema automatizado y se solicitó información desde la aplicación Android, como era de esperarse la aplicación conto los 6 minutos de espera y lanzo la alerta en la barra de notificaciones del dispositivo móvil, resolviendo el problema número tres de la implementación del prototipo II.

Tras la implementación y resultados del presente prototipo no se percató de ningún problema o fallo en el sistema ya que la automatización funciona de la forma correcta tanto con el software de control Java como con la aplicación de control remoto Android., por lo tanto, este prototipo es el último y el que funciona totalmente.

Capítulo IV: Análisis y resultados

La implementación del proyecto se llevó a cabo en el departamento de Biotecnología de Frutales del COLPOS con el prototipo 1, del día 19 de diciembre al 19 de enero, con una duración de 31 días, ciclo que corresponde al periodo de tiempo establecido para la propagación de brotes en el SIT. Por lo que la automatización del sistema se llevó a cabo desde la colocación de los brotes en el SIT, el día 19 de diciembre, hasta que fueron removidos para su conteo el día 19 de enero. A continuación, en la *ilustración 72* se muestra evidencia fotográfica del desarrollo de los explantes durante el periodo de propagación.



ILUSTRACIÓN 72. EVIDENCIA DEL DESARROLLO DE EXPLANTES

Fase de prueba: 18 de diciembre de 2014

Aplicación del sistema automatizado en el SIT, conectando todos los dispositivos involucrados en la automatización, que son: luces, bomba de vacío, y sensores de luminosidad y temperatura. Así mismo se colocó el hardware necesario, conectado al software elaborado para su control, de esta manera, a partir de la programación de periodos y duración de bombeos y cambios de luz, se logró la automatización del sistema conectado a una laptop, empleada para comunicar la aplicación con los

dispositivos, permitiendo de esta manera la activación y cambios de luz y de bombeo, así como el reguardo de los datos relacionados con la automatización. En la *ilustración 73* se muestran los elementos involucrados en la instalación del sistema automatizado.



ILUSTRACIÓN 73. ELEMENTOS DE INVESTIGACIÓN

Día 19 de diciembre de 2014

El día 19 de diciembre, se colocaron los biorreactores con explantes dentro del SIT tras obtener excelentes resultados tras la fase de prueba realizada el día anterior. Programando los periodos de cambio de luz: para el sector 1 seis horas color azul, diez horas color blanco y ocho horas ausencia de luz y para el sector 2 diez horas color azul, seis horas color blanco y ocho horas con ausencia de luz. Por otra parte, los bombeos e programaron para ambos sectores cada cuatro horas con una duración de 60 segundos para cada bombeo. A partir de este momento comenzó la fase de propagación que duró 31 días.

A partir de este día, las visitas al COLPOS se realizaron con la intención de verificar los cambios de luces y la inmersión de aire a los biorreactores por medio de los bombes, para comprobar el correcto funcionamiento del programa y que efectivamente funcionaria la automatización. En la *ilustración 74* se muestra el sistema automatizado ya instalado con 4 biorreactores.



ILUSTRACIÓN 74. SISTEMA AUTOMATIZADO INSTALADO EN EL SIT

Día 21 de diciembre de 2014

Se realizó la verificación del funcionamiento del sistema; en la *ilustración 75* se observa el programa funcionando y las luces encendidas, indicando que ha funcionado correctamente las primeras 48 horas de activación; así mismo se

registraron los bombeos agendados que se realizaron y coinciden con las necesidades del cultivo.



ILUSTRACIÓN 75. 21 DE DICIEMBRE DE 2014

Día 22 de diciembre de 2014



ILUSTRACIÓN 76. 22 DE DICIEMBRE DE 2014

Día 25 de diciembre de 2014

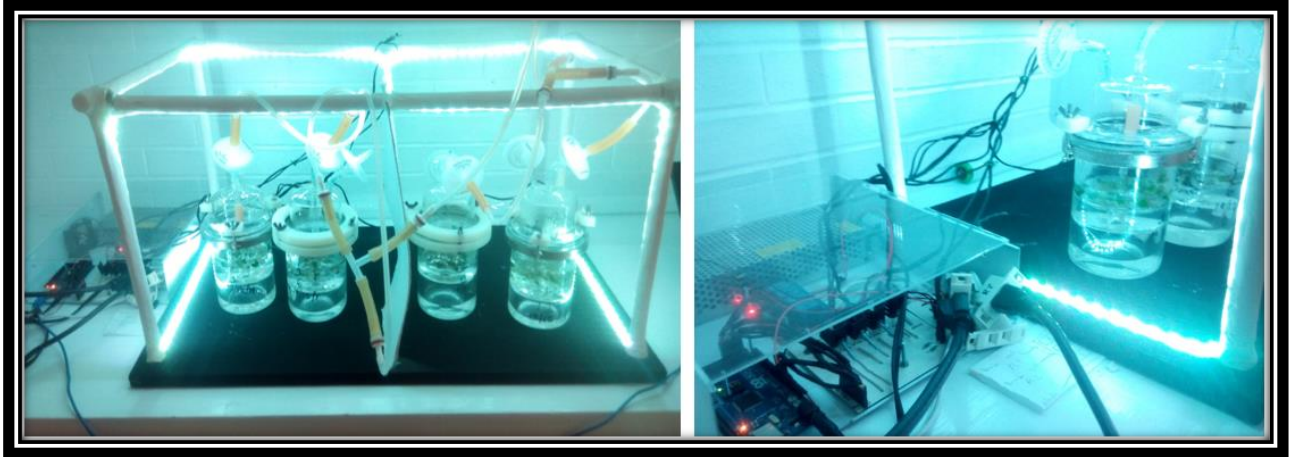


ILUSTRACIÓN 77. 25 DE DICIEMBRE DE 2014

Día 27 de diciembre de 2014

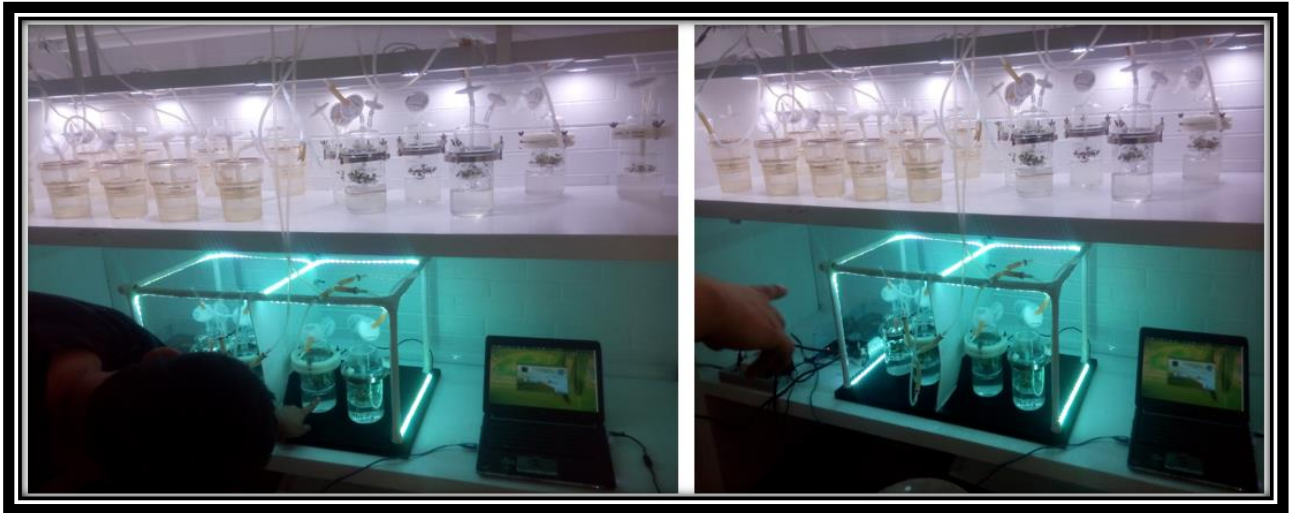


ILUSTRACIÓN 78. 27 DE DICIEMBRE DE 2014

Día 28 de diciembre de 2014



ILUSTRACIÓN 79. 28 DE DICIEMBRE DE 2014

Día 29 de diciembre de 2014



ILUSTRACIÓN 80. 29 DE DICIEMBRE DE 2014

Día 31 de diciembre de 2014



ILUSTRACIÓN 81. 31 DE DICIEMBRE DE 2014

Día 2 de enero de 2015

En este día el sistema sufrió una reactivación tras un corte de luz por parte de la institución debido a días festivos (31 de diciembre de 2014 por la tarde hasta 2 de enero de 2015 por la madrugada); la computadora donde se almacenan los datos tiene una duración de batería de 2.5 h, por lo que el sistema quedo sin energía eléctrica por festividades de fin de año durante 2 días. Para la reactivación solo basto encender la computadora y ejecutar la aplicación, ya que gracias a la base de datos la información de los horarios de bombeo y luz quedo almacenada, por lo que no existió la necesidad de realizar algo más técnico.



ILUSTRACIÓN 82. 2 DE ENERO DE 2015

Día 4 de enero de 2015

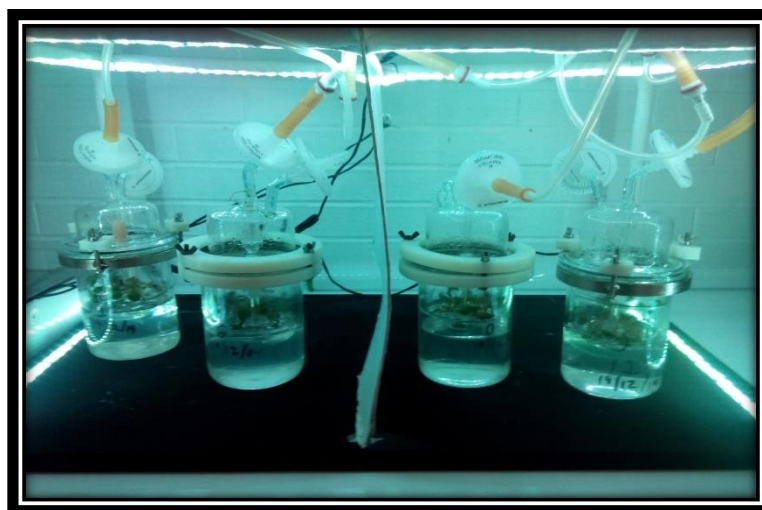


ILUSTRACIÓN 83. 4 DE ENERO DE 2015

Día 7 de enero de 2015

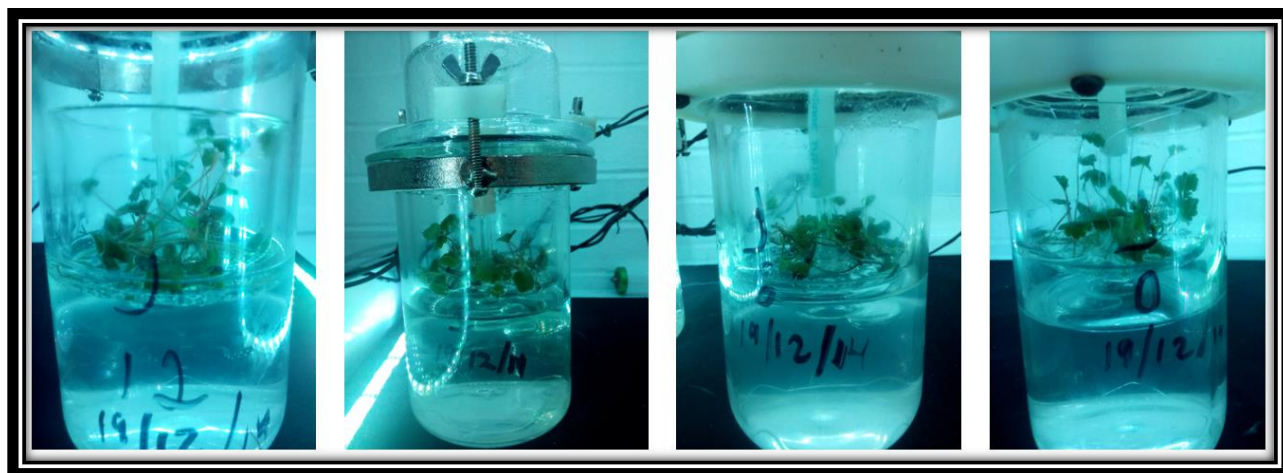


ILUSTRACIÓN 84. 7 DE ENERO DE 2015

Día 9 de enero de 2015

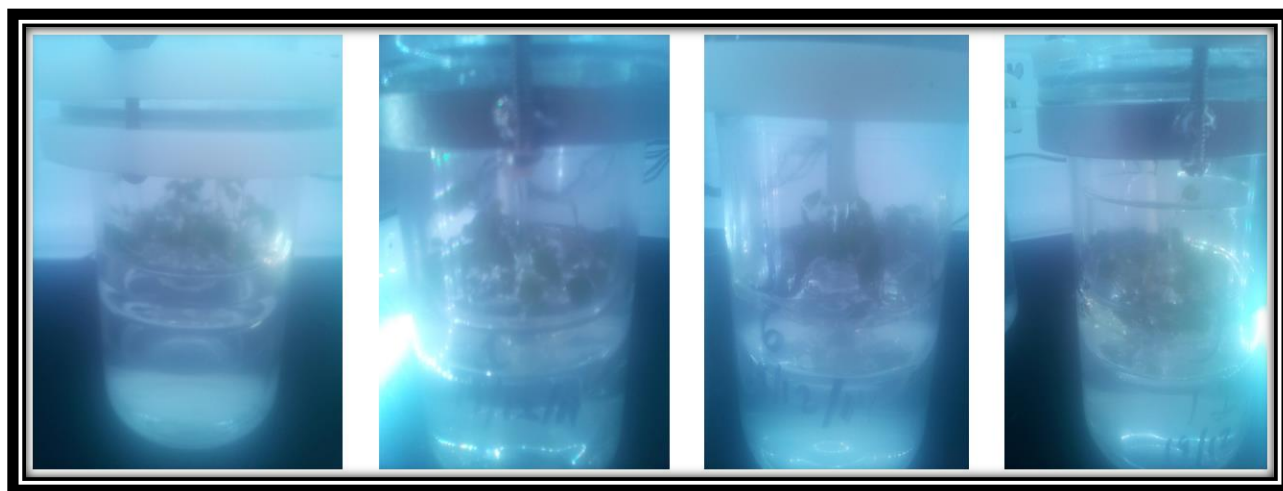


ILUSTRACIÓN 85. 9 DE ENERO DE 2015

Día 13 de enero 2015



ILUSTRACIÓN 86. 13 DE ENERO DE 2015

Día 16 de enero de 2015

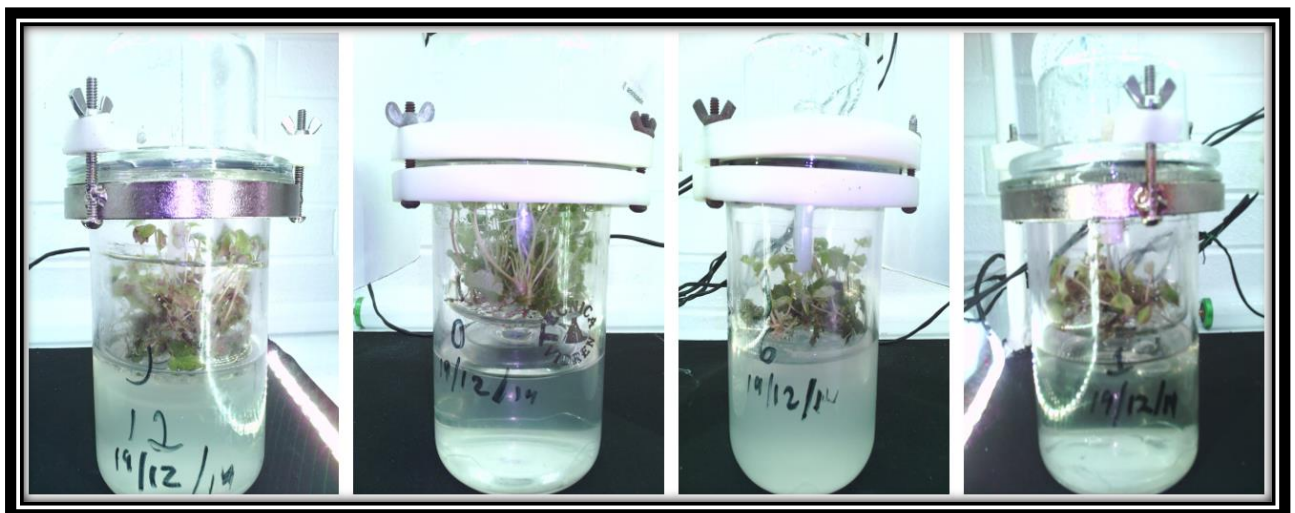


ILUSTRACIÓN 87. 16 DE ENERO DE 2015

Día 19 de enero de 2015

Este día concluyó el proyecto en el COLPOS, con la sustracción de los brotes desarrollados en los biorreactores automatizado para su conteo.



ILUSTRACIÓN 88. CORTE, MEDICIÓN Y CONTEO DE BROTES

Se desconectó un biorreactor por vez para el conteo y medición de los brotes, cada biorreactor fue destapado cuidadosamente, sacando explante por explante como el mostrado en la *ilustración 88* para realizar el corte y medición de cada uno de los brotes, esta operación se repitió por explantes y biorreactores, verificando la existencia de hiper hidricidad, la cantidad de brotes y comprobando su tamaño como se muestra en la *ilustración 89*.



ILUSTRACIÓN 89. EXPLANTES EN LOS BIORREACTORES DESTAPADOS

Los brotes resultantes se colocaron en recipientes de plástico mostrados en la *ilustración 90* con etiquetas en las que se escribió la variedad de fresa, la fecha en que se colocaron y retiraron de los biorreactores, se colocó el horario de luz del biorreactor y su tiempo de inmersión, de acuerdo con instrucciones del responsable del laboratorio.

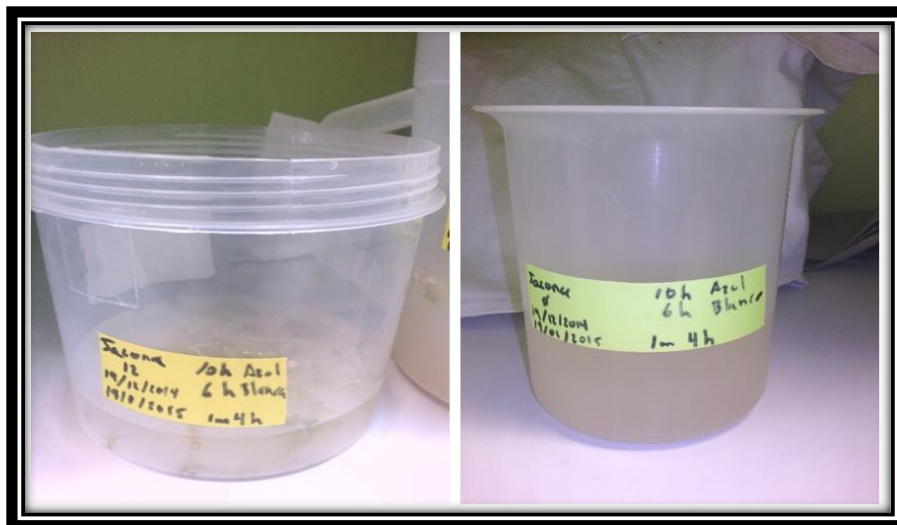


ILUSTRACIÓN 90. FRASCOS DONDE SE DEPOSITARON LOS BROTES

Resultados obtenidos

Los resultados obtenidos a partir de la implementación del proyecto durante un ciclo de propagación, en ambos sectores se muestran en la *tabla 11*.

TABLA 11. RESULTADOS DE LA IMPLEMENTACIÓN EN AMBOS SECTORES

Sector 1									
Variedad	Estado del Biorreactor	Bombeo		Iluminación		N. de brotes Total de brotes	% de propagación %	Tamaño de brote (cm)	
		Periodo (h)	Duración (s)	Azul (h)	Blanco (h)			Mínimo	Máximo
Jacona (6)	Contaminado	4	60	6	10	39	390	0.3	4
Zamorana (1)	Estable	4	60	6	10	11	110	0.3	2.1
Sector 2									
Variedad	Estado del Biorreactor	Bombeo		Iluminación		N. de brotes Total de brotes	% de propagación %	Tamaño de brote (cm)	
		Periodo (h)	Duración (s)	Azul (h)	Blanco (h)			Mínimo	Máximo
Jacona (12)	Contaminado	4	60	10	6	50	500	0.3	5
Jacona (0)	Estable	4	60	10	6	131	1310	0.3	7.6

Se trabajaron dos variedades vegetales de fresa en cuatro biorreactores diferentes, colocando 10 explantes en cada uno, por lo que a partir del número de explantes se obtuvo el porcentaje de propagación mostrado en la *ilustración 91*, que forma parte fundamental del presente trabajo.

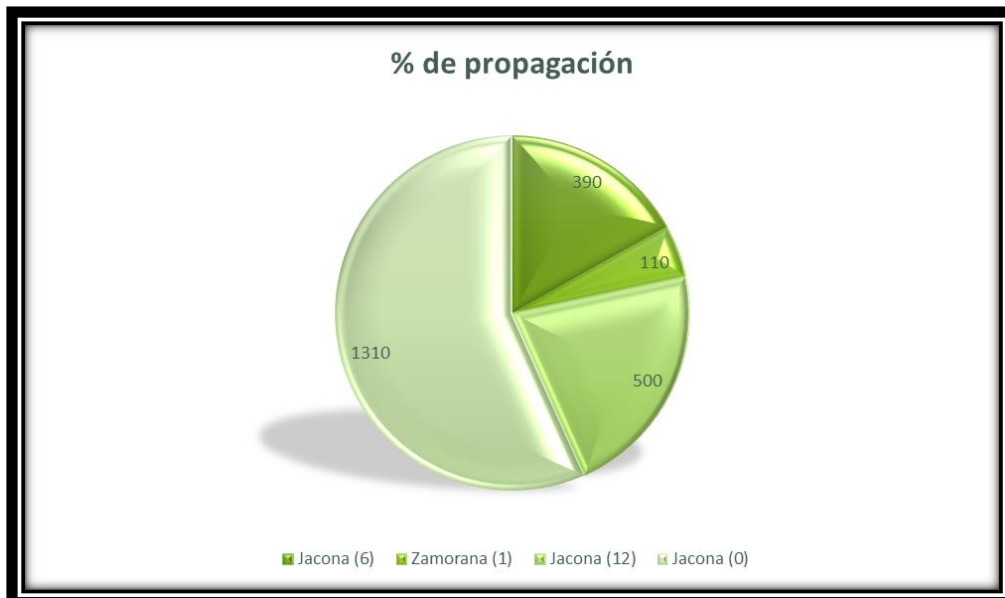


ILUSTRACIÓN 91. GRÁFICO DEL PORCENTAJE DE PROPAGACIÓN

El porcentaje de propagación depende de la calidad del ambiente que existe dentro del biorreactor, por lo que las condiciones de limpieza afectan directamente y de manera significativa el número de brotes obtenidos a partir del proceso de propagación. En este sentido se contó con dos recipientes sin contaminación, de los que podemos observar el mayor porcentaje, por lo que podemos concluir que en



condiciones óptimas de propagación, la automatización ofrece la posibilidad de una mayor reproducción de especies de flora *in vitro*.

Es importante mencionar que en el texto “Shoot regeneration from leaf explants of five strawberry (*Fragaria x Ananassa*) cultivars in temporary immersion bioreactor system“, la autora Hanhieva (2005), menciona que las frecuencias de propagación fueron de 70 ± 8 a $94\pm 2\%$ en biorreactores de inmersión temporal, TIB por sus siglas en inglés, por lo que un porcentaje de propagación de 1310% contra un porcentaje de propagación *normal* de 94% en otros cultivares de fresa, representa una gran mejora en cuanto a la propagación se refiere. según Hanhieva (2005)¹³.

El periodo de los bombeos fue de cuatro horas, lo que significa que se realizaba un bombeo con una duración de 60 segundos cada vez que este periodo se cumplía, como se muestra en la *ilustración 92*. De esta manera se realizaron seis bombeos al día durante 29 días¹⁴.

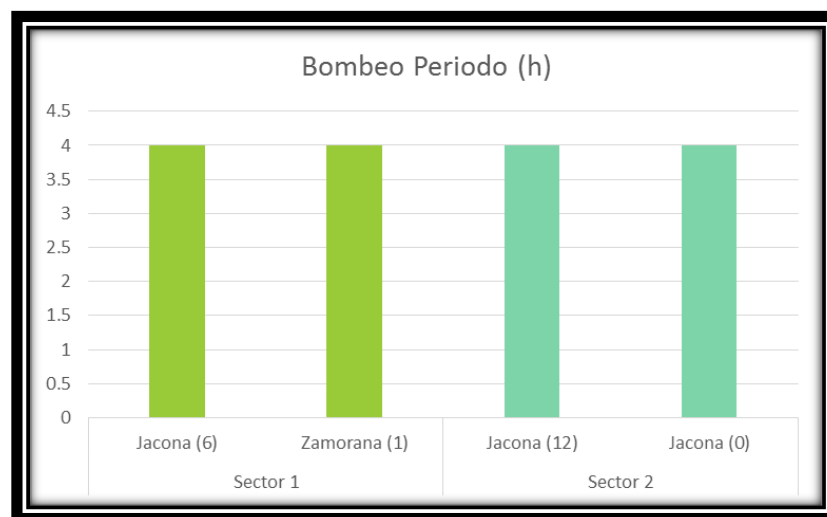


ILUSTRACIÓN 92. GRÁFICO DEL PERIODO DE LOS BOMBEOS

¹³ Hanhieva, K. Kokko, H; Karenlampi S. (2005) Shoot regeneration from leaf explants of five strawberry (*Fragaria x Ananassa*) cultivars in temporary immersion bioreactor system. *In vitro. Cell Dev-Plant.* 41 (&), 826-831.

¹⁴ El día 14 del periodo de propagación, el COLPOS sufrió una pérdida de luz. Evidentemente el sistema que funciona conectado a la energía eléctrica sufrió una falta de energía, quedando inutilizable hasta su reconexión dos días después.

Es importante señalar que el bombeo de nutrientes forma parte fundamental para la propagación *in vitro* pues mediante este proceso se proporciona a los explantes los nutrientes necesarios para su correcto desarrollo, así mismo el bombeo periódico que no implique la permanencia de los explantes en el liquido evita de esta manera la hiperhidratación de los brotes. En la *ilustración 93* se muestra la gráfica de la duración de los bombeos por sector y tipo de fresa.

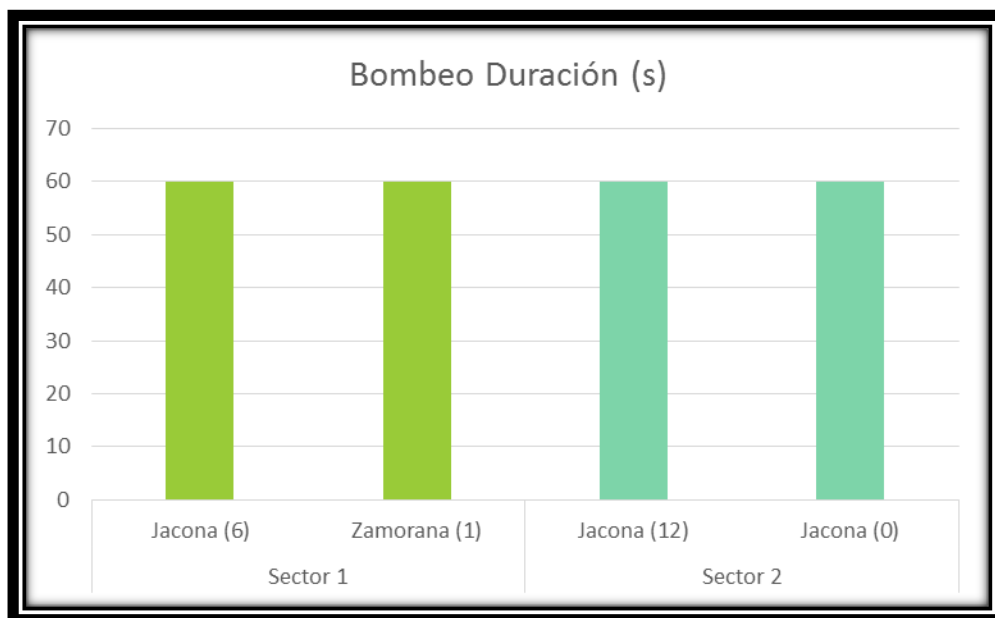


ILUSTRACIÓN 93. GRÁFICO DE LA DURACIÓN DE BOMBEOS

Los tiempo de iluminacion en los biorreactores constaron de ciclos de 16 horas de encendido continuo y 8 horas de inactividad por dia, alternando el color de luz en las horas de actividad entre blanco y azul, como se muestra en la gráfica de la *ilustración 94*.

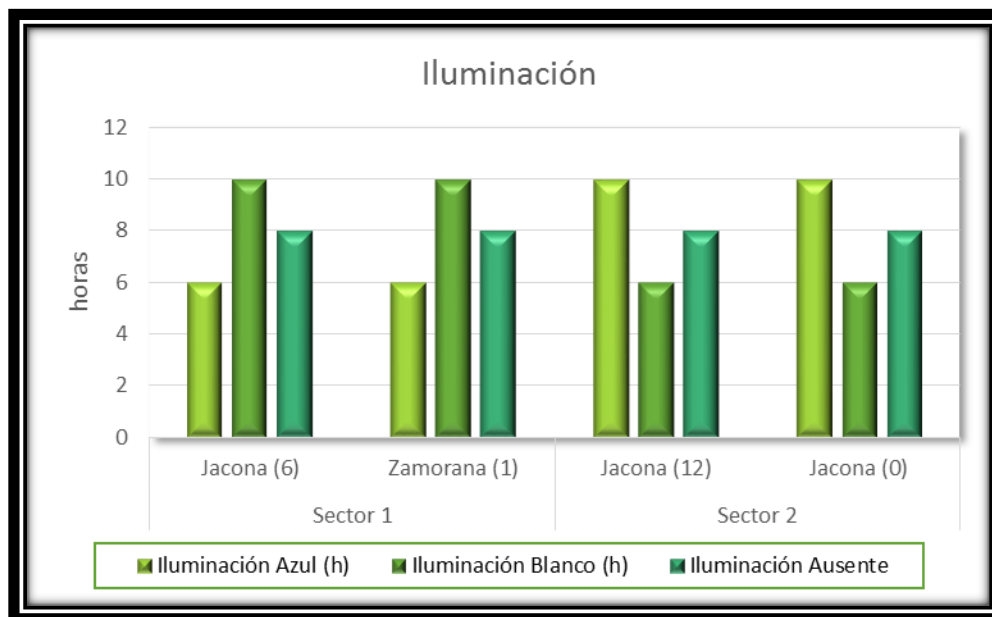


ILUSTRACIÓN 94. TIEMPO DE ILUMINACIÓN Y TIPO DE COLOR EN LOS BIORREACTORES

En condiciones de laboratorio la luz blanca tiene los mismos efectos que la luz solar sobre las plantulas por lo que es indispensable su uso para lograr el desarrollo normal de las plantas. Por otro lado el empleo de luz azul tiene un efecto de inhibición respecto al crecimiento de las plantas convirtiendose en una estrategia de producción de plantas con un crecimiento compacto con la intención de favorecer la micropropagación de plántulas.

De esta manera, los ciclos de bombeo y cambios de iluminación favorecieron la propagación de dos variedades de fresa, dando como resultado un total de 131 brotes en el biorreactor más productivo y sin contaminación y de 11 en el biorreactor que presentó necrosis y oxidación debido al alto nivel de contaminación. En la *tabla 12* se muestra el total de brotes obtenidos; así como el tamaño mínimo y máximo; así mismo en la *ilustración 95* se muestra el gráfico del total de brotes generados en cada uno de los biorreactores, finalmente en la *ilustración 96* se observa la gráfica del tamaño mínimo y máximo de los brotes por biorreactor.

TABLA 12. TOTAL DE BROTES Y TAMAÑOS

Total de brotes en ambos sectores	231
Tamaño mínimo de brote (cm)	0.3
Tamaño máximo de brote (cm)	7.6



ILUSTRACIÓN 95. TOTAL DE BROTES POR BIORREACTOR

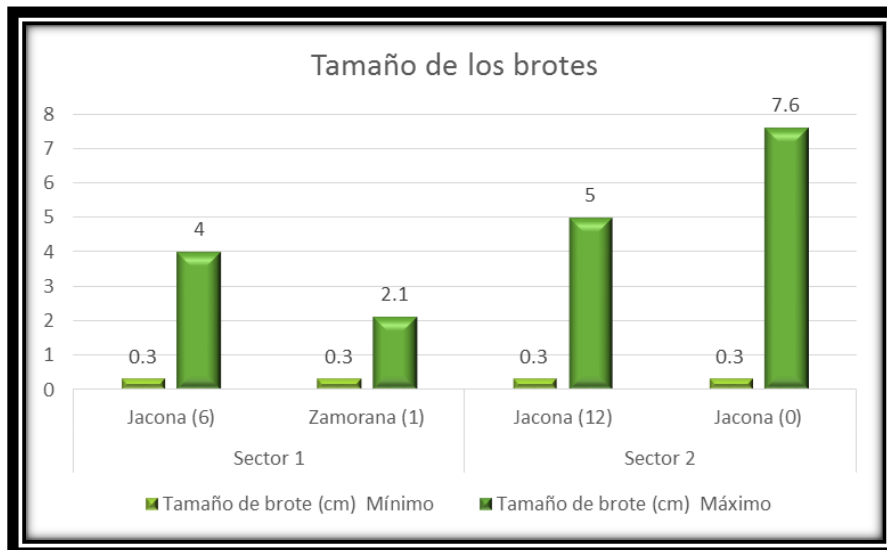


ILUSTRACIÓN 96. BROTE DE TAMAÑO MÁXIMO Y MÍNIMO POR BIORREACTOR



Capítulo V: Conclusiones

Ante las problemáticas actuales de escasas de producción agrícola, es necesario desarrollar herramientas que permitan aumentar la cantidad y calidad de diferentes tipos de cultivo. La automatización de un SIT permite el control del ambiente dentro del sistema, lo que mejora los resultados de la reproducción *in vitro*.

Por tal motivo, la automatización del sistema se convierte en una herramienta necesaria para el aumento de producción agrícola al mejorar los atributos de los brotes. Tras la implementación del primer prototipo en el colegio de postgraduados fue posible corroborar que el proyecto mejoró no sólo el ambiente del cultivo, sino su micro propagación, que se vio beneficiada en cantidad y calidad de la producción de nuevos brotes. Por lo que la automatización del SIT no es sólo un factor de actualización de la manera de cultivo *in vitro*, sino que se convierte en una necesidad en la producción de especies de flora por este medio.

Asimismo, el software elaborado en el proyecto, se convirtió en una herramienta que permite al usuario comunicarse de manera sencilla con el hardware elaborado para la automatización, de manera que esta herramienta facilita el trabajo para el usuario del sistema y el control del SIT.

Las tecnologías móviles actuales permiten el desarrollo de herramientas que permitan el control remoto de dispositivos, de manera que el desarrollo de una aplicación Android que permite el control remoto del SIT, fue uno de los puntos fuertes del presente proyecto, pues permite al usuario el monitoreo y control del hardware involucrado en la automatización del SIT aun sin estar presente en el lugar.

De manera que la automatización en el SIT, permite mejorar la calidad y cantidad de los brotes necesarios para el aumento de la producción agrícola. Mientras que el desarrollo de la aplicación móvil del SIT permite el control y monitoreo de las condiciones del mismo, convirtiéndose el software elaborado en el proyecto en la herramienta fundamental de comunicación entre el hardware, el software y el usuario.

De esta forma, es posible concluir que la automatización obtuvo los resultados esperados, comprobando tanto la hipótesis *si se automatiza un SIT, entonces se mejorará tanto el estado del cultivo como su desarrollo, aumentando de esta manera su producción*, como la importancia de la automatización de este tipo de sistemas en vías de una mejor calidad de vida.

Costos

Uno de los objetivos del prototipo es la disminución de costos por lo que a continuación, la *tabla 13* presenta los costos necesarios para la implementación del prototipo.

TABLA 13. COSTOS DE LA AUTOMATIZACIÓN

Material	Cantidad	Costo unitario	Costo
Arduino Uno R3	2	\$ 420	\$ 840
Tira de relevadores	1	\$ 149	\$ 149
Electroválvulas	2	\$ 195	\$ 290
Shield GSM	1	\$ 1020	\$ 1020
Sensor DS18B20	1	\$ 123	\$ 123
Fotorresistencias	2	\$ 5	\$ 10
Total			\$ 2432

Dado que se trabajó con plataformas abiertas de hardware, es posible incluir sensores comerciales o de desarrollo propio para medir alguna condición en particular del sistema,

Todo el software que se utilizó es de licencia gratuita por lo que es posible generar herramientas robustas que pueden adaptarse a las necesidades de cada problema.



Bibliografía

- Alberto Perez, M., Perez Hidalgo, A., & Perez Berenguer, E. (2007). *Introducción a los Sistemas de Control y Modelo Matemático para Sistemas Lineales Invariantes en el Tiempo*. San Juan: Universidad Nacional de San Juan.
- Android. (08 de 07 de 2015). *Android developers*. Obtenido de Layouts: <http://developer.android.com/guide/topics/ui/declaring-layout.html>
- Ángel , J., González, J., Orellana, M., & Arévalo, C. (2013). *Evaluación de dos métodos de micropropagación masal en piña (Ananas comosus L. Merr.) variedad Golden*. San Salvador: Universidad de El Salvador.
- Arduino. (01 de Julio de 2015). *Arduino Ethernet Shield*. Obtenido de Arduino: <https://www.arduino.cc/en/pmwiki.php?n=Main/ArduinoEthernetShield>
- Balcells, J., & Daura, F. (1992). *Interferencias electromagnéticas en sistemas electrónicos*. Barcelona: Marcombo.
- Banzi, M. (2009). *Getting Started with Arduino* (1 ed.). Sebastopol, Estados Unidos: O'Reilly.
- Bello, J., & Iglesias, L. (2012). Desarrolla el INBIOTECA sistemas de biorreactores para la micropropagación de especies vegetales de importancia para el estado de Veracruz. *Agro Entorno*, 7.
- Bernal, C. (2006). *Metodología de la Investigación*. México: Pearson Educación.
- Berón, M., Henriques, P., Varanda, M., & Uzal, R. (2006). Herramientas para la comprensión de programas. *VIII Workshop de Investigadores en Ciencias de la Computación*.
- Bradski, G., & Kaebler, A. (2008). *Learning OpenCV*. California: O'Reilly.

- Castillo, A., Vázquez, J., Ortegón, J., & Rodríguez, C. (2008). Prácticas de laboratorio para estudiantes de ingeniería con FPGA. *IEEE Latin America Transactions*, 130-136.
- Corona Ramírez, L. G., Abarca Jiménez, G. S., & Mares Carreño, J. (2014). *Sensores y actuadores. Aplicaciones con Arduino*. México D.F.: Grupo Editorial Patria, S.A. de C.V.
- Del Rivero, N. (2004). Empleo de sistemas de inmersión temporal para la multiplicación in vitro de brotes de *Anthurium andraeanum*. *Biotecnología Vegetal*, 97.
- Díaz, R., García, L., & Espinosa, D. (2011). *Control y automatización de un sistema de bombeo de un invernadero para el desarrollo, cultivo e investigación de flora*. Zacatenco: Instituto Politécnico Nacional.
- Enríquez Herrador, R. (2009). *Guía de Usuario de Arduino* (primera ed.). Córdoba: Creative Commons.
- Gago Alderón, A., & Fraile Vilarrasa, J. (2012). *Iluminación con tecnología led*. España: Paraninfo.
- García, P. (2006). *Introducción a la investigación bioantropológica en actividad física, deporte y salud*. Venezuela: Colección Monografías.
- Guardado, I., & Palacios, I. (2009). *Monitoreo de una Estación Meteorológica Vía Internet*. Zacatecas: UAZ.
- Kreibich, J. A. (2010). *Using SQLite*. California: O`Reilly Media, Inc.
- Kuo, B. C. (1996). *Sistemas de Control Automatico*. Pearson Educación.
- Kurniawan, B. (2016). *Android Application Development: A Beginner's Tutorial*. California: BrainySoftware.



- López García, R., Narváez-A roche, O., & Rocha Cózatl, E. (Septiembre de 2011). Interfaz Gráfica de Usuario para Pruebas de Marcha en un Robot Bípedo. *Memorias del XVII Congreso Internacional Anual de la SOMIM(17)*, 357-364.
- Margolis, M. (2011). *Arduino Cookbook* (Primera ed.). Sebastopol, Estados Unidos de America: O'Reilly.
- Martínez Gil, F., & Martín Quetglás, G. (2003). *Introducción a la programación estructurada en C* (Segunda ed.). Valencia: Quiles Artes Gráficas.
- Medina García, G., Grageda Grageda, J., Ruiz Corral, J. A., & Báez González, A. D. (2008). *Uso de Estaciones meteorológicas en la agricultura* (Primera ed.). México: INIFAP.
- Muñoz, J., & Núñez, D. (2012). *AUTOMATIZACIÓN DE INVERNADERO EN CLIMA TEMPLADO*. Cali: Universidad de San Buenaventura.
- Núñez, H. (2002). *Los sistemas just-in-time/Kanban, un paradigma productivo*. México: Política y cultura.
- Osorio Rivera, F. L. (2007). *Introducción a la programación en Java. Un enfoque práctico*. Medellín: Fondo editorial ITM.
- Pomares Baeza, J. (2009). *Manual de Arduino: Control por computador* (Primera ed.). Alicante: Grupo de Innovación Educativa en Automática.
- Ramírez S., J. M., & Rosero García, E. E. (2007). *Sistemas de Control I*. Santiago de Cali: Universidad del Valle.
- Sarik, J., & Kymissis, I. (27 de Octubre de 2010). Lab Kits Using the Arduino Prototyping Plataforma. *40th ASEE/IEEE Frontiers in Education Conference(40)*, 1-5.
- Tipler, P. A., & Mosca, G. (2005). *Física para la ciencia y la tecnología*. España: Reverté.
- Torrente, Ó. (2013). *ARDUINO. Curso práctico de formación*. Madrid: RC Libros.

Vallejo, H. (6 de Junio de 2016). *Todo PIC*. Obtenido de Los controladores lógicos programables: <http://www.todopic.com.ar/utiles/PLC>



Anexo I



TERRA
Latinoamericana

[Inicio](#) | [ACERCA DE](#) | [ÁREA PERSONAL](#) | [BUSCAR](#) | [ACTUAL](#) | [NÚMEROS ANTERIORES](#) | [AVISOS](#) | [INFORMACIÓN LEGAL](#) | [GUÍA DE AUTORES](#)

#193 Resumen

RESUMEN
REVISIÓN
EDICIÓN

ENVÍO

Autores/as	Oziel Lugo Espinosa, Gregorio Arellano Ostoa, Diego Hernández Cote
Título	Automatización de un sistema de inmersión temporal con base en plataformas abiertas de hardware y software
Archivo original	193-645-1-SM.DOCX 2016-11-08
Archivos adicionales	Ninguno/a Añadir un archivo complementario
Emisor/a	Dr. Oziel Lugo Espinosa
Fecha de envío	noviembre 8, 2016 - 10:14
Sección	Nota de Investigación
Editor/a	Ninguno/a asignado/a

[Ayuda de la revista](#)

IDIOMA

Español

TAMAÑO DE FUENTE

A⁺ A⁻ A

USUARIO/A

Ha iniciado sesión como ozieluz

[Mi perfil](#)

[Cerrar sesión](#)

ESTADO

Estado	Asignación en espera
Iniciado	2016-11-08
Modificado por última vez	2016-11-08

METADATOS DEL ENVÍO

[Editar metadatos](#)

AUTORES/AS

Nombre	Oziel Lugo Espinosa
Empresa	Universidad Autónoma del Estado de México
País	Mexico
Biografía	Ingeniería en computación
	Centro Universitario UAEM Texcoco
	Universidad Autónoma del Estado de México

Contacto principal para la correspondencia editorial.

Nombre	Gregorio Arellano Ostoa
Empresa	Colegio de Postgraduados
País	Mexico
Biografía	POSDOCTORADO EN BIOTECNOLOGIA, MEX, 2009 DOCTORADO EN CIENCIAS, UNIVERSIDAD DE SANTIAGO DE COMPOSTELA, ESP, 2005 MESTRIA EN CIENCIAS, COLEGIO DE POSTGRADUADOS, MEX, 2004 INGENIERO AGRICOLA, UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO, MEX, 1985

Nombre	Diego Hernández Cote
Empresa	Maestría en Ciencias de la computación
	Centro Universitario UAEM Texcoco
	Universidad Autónoma del Estado de México
País	Mexico
Biografía	Ingeniería en computación

CREATIVE COMMONS

AUTOR/A

Envíos

Activos (1)

Archivados (0)

[Nuevo envío](#)

NOTIFICACIONES

[Ver](#)

[Administrar](#)

CONTENIDO DE LA REVISTA

Buscar

Todos

Buscar

Examinar

[Por número](#)

[Por autor/a](#)

[Por título](#)

NÚMERO ACTUAL

193 1.0
192 1.0
191 1.0

[OPEN JOURNAL SYSTEMS](#)

TÍTULO Y RESUMEN

Título	Automatización de un sistema de inmersión temporal con base en plataformas abiertas de hardware y software
Resumen	Un Sistema de Inmersión Temporal (SIT) permite la micropropagación de plantas en ambientes esterilizados y herméticos; sin embargo, es una tecnología complicada y costosa. Se presenta una alternativa de automatización con base en plataformas abiertas de hardware y software lo que se traduce en la reducción de costos frente a sistemas comerciales, reconfiguración de rutinas de aplicación de nutrientes mediante software, monitoreo y control a distancia del sistema.

INDEXACIÓN

Palabras clave	java; reproducción de especies in vitro; arduino; biorreactores; software libre
Idioma	es

ORGANISMOS COLABORADORES

Agencias	Universidad Autónoma del Estado de México, Colegio de Postgraduados
----------	---

REFERENCIAS

Referencias —

TERRA LATINOAMERICANA, Volumen 34, número 4, Octubre-Diciembre 2016, es una publicación trimestral editada y publicada por la Sociedad Mexicana de la Ciencia del Suelo A. C., Apartado Postal 45, 56230 Chapingo Estado de México, México. Edificio del Departamento de Suelos, Universidad Autónoma Chapingo, 56230 Chapingo, Estado de México. Teléfono 01 (595) 95 2 17 21. <http://www.terralatinoamericana.org.mx>. Editora responsable: Dra. Mariela Hada Fuentes Ponce. ISSN 2395 - 8030. Reserva de Derechos al Uso Exclusivo No. 04-2012-092017263600-203. Ambos otorgados por el Instituto Nacional del Derecho de Autor.

Responsable de la última actualización de este Número: Dra. Mariela Hada Fuentes Ponce, Apartado Postal 45, 56230 Chapingo Estado de México, México. Fecha de última actualización, 3 de octubre de 2016. Los artículos publicados son responsabilidad absoluta de los autores. Se autoriza la reproducción parcial o total de esta revista, citándola como fuente de información.

Sistema de Clasificación de Revistas Mexicanas de Ciencia y Tecnología

Buscar revista (Por Título, ISSN, E-ISSN, Institución Editora o Editores)


[Inicio](#) / [Terra Latinoamericana \(México, D.F.\)](#)

Terra Latinoamericana (México, D.F.)

ISSN: 1870-9982

[Descargar PDF](#)

Dimensión	Puntaje Dimensión	Criterio	Puntaje Criterio
Política y gestión editorial 10.00 % / 15.00 %	18 pts / 27 pts	1.1 Disponibilidad de título paralelo y abstracts en inglés	3 pts
		1.2 Nivel de conveniencia de la política editorial	2 pts
		1.3 Tipo de revisión por pares	4 pts
		1.4 Aplicación de criterios éticos	2 pts
		1.5 Internacionalidad de los editores y del comité científico	2 pts
		1.6 Internacionalidad de los autores	1 pts
		1.7 Porcentaje de autores de la propia institución	3 pts
		1.8 Número de artículos de producción citables publicados	1 pts
Calidad del contenido 7.50 % / 25.00 %	6 pts / 20 pts	2.1 Claridad de los abstracts	3 pts
		2.2 Calidad y conformidad con el alcance (scope) de la revista	1 pts
		2.3 Legibilidad de los artículos	2 pts
Nivel de citación 4.17 % / 25.00 %	2 pts / 12 pts	3.1 Citación de los artículos de la revista en Scopus	0 pts
		3.2 Citación de los artículos de la revista en Google Scholar	2 pts
		3.3 Citación de los editores de la revista en Scopus	0 pts
Cumplimiento de la frecuencia de publicación 8.75 % / 10.00 %	7 pts / 8 pts	4.1 Publicación al inicio del período programado	4 pts
		4.2 Tiempo entre aceptación, revisión y aprobación	1 pts
		4.3 Uso de plataforma de edición en línea	2 pts
Accesibilidad 6.15 % / 10.00 %	8 pts / 13 pts	5.1 Contenido disponible on-line	3 pts
		5.2 Disponibilidad en inglés del home page de la revista	1 pts
		5.3 Calidad del home page de la revista	4 pts
		5.4 Edición de artículos en formato XML	0 pts
Visibilidad internacional 12.00 % / 15.00 %	16 pts / 20 pts	6.1 Indización en SciELO	6 pts
		6.2 Indización en Redalyc	6 pts
		6.4 H-Index Google Scholar	1 pts
		6.3 Puntaje según características editoriales de Latindex Catálogo	3 pts
		6.5 La revista está incluida en el Directory of Open Access Journals (DOAJ)	0 pts

La revista obtuvo 48.57 pts.
Se considera:

De 53.63 pts. a 68.55 pts.	Competencia Internacional
De 45.62 pts. a 53.62 pts.	Competencia Nacional
De 37.57 pts. a 45.61 pts.	En Proceso de Consolidación
De 4.42 pts. a 37.56 pts.	En Desarrollo