



Universidad de Jaén

Escuela Politécnica Superior de Jaén

Trabajo Fin de Grado

# **DESARROLLO E IMPLEMENTACIÓN DE UN INVERNADERO INTELIGENTE CONTROLADO CON ARDUINO**

**Alumno: Antonio Jesús Mañas Torres**

**Tutor: Prof. D. Diego Manuel Martínez Gila**

**Dpto: Automática y Electrónica**





Universidad de Jaén  
Escuela Politécnica Superior de Jaén  
Departamento de Electrónica y Automática

Don Diego Manuel Martínez Gila , tutor del Proyecto Fin de Grado titulado: Desarrollo e implementación de un invernadero inteligente controlado con Arduino, que presenta Antonio Jesús Mañas Torres, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Junio de 2019

El alumno:

Los tutores:

Una firma manuscrita en tinta azul, que parece ser la del tutor, escrita sobre una línea horizontal.

Antonio Jesús Mañas Torres

Diego Manuel Martínez Gila



## RESUMEN

Los invernaderos permiten tener un mayor control del entorno donde crecen las plantas. Siendo útiles para aumentar la calidad y rendimiento de las plantaciones en algunas ubicaciones que, por motivos de localización geográfica, tiempo, temperatura no permiten la ubicación y plantación de estos cultivos. En este informe se recoge el desarrollo y la implementación de un proyecto clasificado en el mundo del control por computador. Consiste en una planta a escala de un invernadero, en la cual podemos tanto monitorizar variables como actuar sobre actuadores. Este proceso se realiza a través de Arduino. Se incluye también, el diseño de una interfaz en java, con la que interactuamos, haciendo posible el encendido o apagado de actuadores (bombilla, tres ventiladores y bomba de agua), así como visualizar variables de temperatura, humedad del suelo, luminosidad y nivel del tanque de agua, además de poder hacer un control todo/nada, un control P, PI, PID o una captura de datos (tanto automática, como de forma manual).

## ABSTRACT

Greenhouses allow for greater control of the environment in which plants grow. Being useful to increase the quality and yield of plantations in some locations that, for reasons of geographical location, time, temperature do not allow the location and planting of these crops. This report covers the development and implementation of a project classified in the world of computer control. It consists of a plant at the scale of a greenhouse, in which we can monitor variables as well as act on actuators. This process is done through Arduino. It also includes the design of a java interface, with which we interact, making it possible to turn on or off actuators (bulb, three fans and water pump), as well as display temperature variables, soil moisture, luminosity and water tank level, in addition to being able to make an all / nothing control, a P, PI, PID control or a data capture (both automatic, as well as manually).



## ÍNDICE GENERAL

<b>1. INTRODUCCIÓN.....</b>	<b>11</b>
<b>1.1 Motivación.....</b>	<b>11</b>
<b>1.2 Estado del arte.....</b>	<b>12</b>
<b>1.2.1 Invernaderos de Grupo de Investigación .....</b>	<b>12</b>
1.2.1.1 Domotic GreenHouse .....	12
<b>1.2.2 IOGreenhouse.....</b>	<b>14</b>
<b>1.2.3 Novagric.....</b>	<b>16</b>
<b>1.3 Objetivos.....</b>	<b>18</b>
<b>1.3.1 Objetivo General .....</b>	<b>18</b>
<b>1.3.2 Objetivos Específicos.....</b>	<b>19</b>
<b>2. METODOLOGÍA.....</b>	<b>20</b>
<b>2.1 Desarrollo Hardware.....</b>	<b>20</b>
<b>2.1.1 Sensores .....</b>	<b>20</b>
2.1.1.1 Sensor de temperatura tipo SHT31 .....	20
2.1.1.2 Sensor de temperatura tipo DHT11 .....	22
2.1.1.3 Sensor de humedad del suelo tipo SKU SEN0114 .....	24
2.1.1.4 Sensor de luminosidad tipo LDR.....	25
2.1.1.5 Sensor de nivel de agua tipo LS02.....	27
<b>2.1.2 Actuadores .....</b>	<b>29</b>
2.1.2.1 Bombilla incandescente 60W 24Vdc .....	29
2.1.2.2 Ventilador 12Vdc .....	31
2.1.2.3 Bomba de Agua 12Vdc 10l/min 0.5bar .....	32
<b>2.1.3 Placa de Arduino .....</b>	<b>33</b>
<b>2.1.4 Otros componentes .....</b>	<b>35</b>
2.1.4.1 Mosfet IRF520.....	35
2.1.4.2 Fuente de Alimentación .....	36
2.1.4.3 PCB .....	37
<b>2.1.5 Circuito Eléctrico .....</b>	<b>39</b>
<b>2.2 Desarrollo Software.....</b>	<b>41</b>
<b>2.2.1 Arduino.....</b>	<b>41</b>
2.2.1.1 Librerías y Funciones.....	41
2.2.1.2 Diagramas de Flujo .....	43
<b>2.2.2 JAVA .....</b>	<b>48</b>
2.2.2.1 Librerías y Clases.....	48
2.2.2.2 Diagramas de Flujo .....	49
2.2.2.3 Interfaz Gráfica .....	55

<b>3. RESULTADOS</b>	58
3.1 Solución Final	59
3.2 Ensayos	63
3.2.1 Ensayo Tipo 1	63
3.2.2 Ensayo Tipo 2	68
<b>4. CONCLUSIÓN Y TRABAJOS FUTUROS</b>	72
<b>5. BIBLIOGRAFÍA</b>	75
<b>6. ANEXOS</b>	77
6.1 Manual NetBeans	77
6.2 Manual Arduino	83

## ÍNDICE ILUSTRACIONES

Ilustración 1. Automatización	11
Ilustración 2. Motor paso a paso	13
Ilustración 3. Sensor humedad	13
Ilustración 4. Esquema del proceso	14
Ilustración 5. Materiales IOGreenhouse	15
Ilustración 6. Conexiones IOGreenhouse	15
Ilustración 7. Carros de riego	16
Ilustración 8. Xilema	16
Ilustración 9. Ventilación forzada	17
Ilustración 10. Calefacción Agua	17
Ilustración 11. Cooling	17
Ilustración 12. Fog System/Fogger	18
Ilustración 13. Objetivo General	19
Ilustración 14. Sensor SHT31	21
Ilustración 15. División Mensaje I2C	22
Ilustración 16. Conexiones sensor DHT11	23
Ilustración 17. División Mensaje DHT11	24
Ilustración 18. Conexiones sensor SEN0114	24
Ilustración 19. Componentes sensor LDR	25
Ilustración 20. Fotodiodo	26
Ilustración 21. Fotorresistencia no lineal	26
Ilustración 22. Esquema Conexión LDR 1	27
Ilustración 23. Esquema Conexión LDR 2	27
Ilustración 24. Esquema Funcionamiento LS02	28
Ilustración 25. Esquema Posiciones Flotador LS02	28
Ilustración 26. Conexiones LS02	29
Ilustración 27. Partes de una bombilla	30
Ilustración 28. Conexiones Mosfet Bombilla	30
Ilustración 29. Conexiones Mosfet Ventiladores	31
Ilustración 30. Funcionamiento Bomba de Agua	32
Ilustración 31. Conexión Bomba de Agua	33
Ilustración 32. Placa Arduino Mega	34



Ilustración 33. Pines Arduino Mega .....	35
Ilustración 34. Tipos de Mosfets.....	35
Ilustración 35. Conexión IRF520 .....	36
Ilustración 36. Fuente de alimentación.....	37
Ilustración 37. PCB Orcad Capture.....	38
Ilustración 38. Diseño PCB .....	39
Ilustración 39. Esquema Eléctrico .....	40
Ilustración 40. Diagrama de Flujo Arduino General.....	43
Ilustración 41. Modo Manual Arduino .....	44
Ilustración 42. Modo Automático Arduino .....	45
Ilustración 43. Modo PID1 Arduino .....	46
Ilustración 44. Modo PID2 Arduino .....	47
Ilustración 45. Diagrama de Flujo Java General.....	50
Ilustración 46. Diagramas de Flujo de los diferentes Modos .....	51
Ilustración 47. Botón Enviar Umbrales .....	52
Ilustración 48. Botón Start Captura de datos Automático.....	53
Ilustración 49. Botón Enviar PID.....	54
Ilustración 50. Estado Actuadores.....	55
Ilustración 51. Estado Sensores.....	55
Ilustración 52. Modo Manual.....	56
Ilustración 53. Modo Automático .....	56
Ilustración 54. Modos PID .....	57
Ilustración 55. Capturas de Datos .....	57
Ilustración 56. Interfaz Gráfica.....	58
Ilustración 57. Primer Modelo Invernadero.....	59
Ilustración 58. Bomba de Agua y Sensor de Nivel .....	60
Ilustración 59. Sensores de Humedad y Sistema de Riego.....	60
Ilustración 60. Grupo Mosfet Final .....	61
Ilustración 61. Versión Fase Tres.....	61
Ilustración 62. PCB Final .....	61
Ilustración 63. Ventiladores Finales .....	62
Ilustración 64. Conjunto Mosfets, PCB, Sensor SHT31 .....	62
Ilustración 65. Invernadero Inteligente Final.....	62
Ilustración 66. Ensayo1 T=30.5°C.....	63
Ilustración 67. Ensayo1 Temperatura.....	64
Ilustración 68. Ensayo1 Actuador.....	64
Ilustración 69. Ensayo1 Actuador con Perturbación .....	64
Ilustración 70. Ensayo1 Temperatura con Perturbación.....	64
Ilustración 71. Ensayo2 T=35.5°C.....	65
Ilustración 72. Ensayo2 Actuador.....	65
Ilustración 73. Ensayo2 Temperatura.....	65
Ilustración 74. Ensayo2 Actuador con Perturbación .....	66
Ilustración 75. Ensayo2 Temperatura con Perturbación.....	66
Ilustración 76. Ensayo3 T=38°C.....	66
Ilustración 77. Ensayo3 Actuador.....	67
Ilustración 78. Ensayo3 Temperatura.....	67
Ilustración 79. Ensayo3 Actuador con Perturbación .....	67
Ilustración 80. Ensayo3 Temperatura con Perturbación.....	67
Ilustración 81. Constantes Ensayo1 Tipo2 .....	68
Ilustración 82. Ensayo1 Tipo2 Temperatura.....	69
Ilustración 83. Ensayo1 Tipo2 Actuador.....	69

Ilustración 84. Constantes Ensayo2 Tipo2 .....	69
Ilustración 85. Ensayo2 Tipo2 Actuador.....	70
Ilustración 86. Ensayo2 Tipo2 Temperatura.....	70
Ilustración 87. Constantes Ensayo3 Tipo2 .....	70
Ilustración 88. Ensayo3 Tipo2 Actuador.....	71
Ilustración 89. Ensayo3 Tipo2 Temperatura.....	71
Ilustración 90. Constantes Ensayo4 Tipo2 .....	71
Ilustración 91. Ensayo4 Tipo2 Actuador.....	72
Ilustración 92. Ensayo4 Tipo2 Temperatura.....	72
Ilustración 93. Selección de Idioma y Sistema Operativo .....	77
Ilustración 94. Selección de Paquetes a descargar .....	78
Ilustración 95. Descarga JDK .....	78
Ilustración 96. Segunda Pantalla de Instalación .....	79
Ilustración 97. Primera Pantalla de Instalación .....	79
Ilustración 98. Cuarta Pantalla de Instalación .....	79
Ilustración 99. Tercera Pantalla de Instalación .....	79
Ilustración 100. Primera Pantalla NetBeans.....	80
Ilustración 101. Segunda Pantalla NetBeans.....	80
Ilustración 102. Tercera Pantalla NetBeans .....	81
Ilustración 103. Cuarta Pantalla NetBeans.....	81
Ilustración 104. Quinta Pantalla NetBeans .....	81
Ilustración 105. Sexta Pantalla NetBeans.....	82
Ilustración 106. Séptima Pantalla NetBeans .....	82
Ilustración 107. Octava Pantalla NetBeans .....	82
Ilustración 108. Descarga Arduino.....	83
Ilustración 109. Primera Pantalla Arduino.....	83
Ilustración 110. Segunda Pantalla Arduino.....	83

## ÍNDICE TABLAS

Tabla 1. Especificaciones Técnicas SHT31 .....	21
Tabla 2. Características I2C.....	22
Tabla 3. Características técnicas DHT11.....	23
Tabla 4. Características técnicas SEN0114 .....	25
Tabla 5. Características Bomba de Agua .....	32
Tabla 6. Características Arduino Mega .....	34
Tabla 7. Características IRF520.....	36
Tabla 8. Constantes Ensayo Tipo 1.....	63
Tabla 9. Temperatura Inicial y Consigna Ensayo Tipo2 .....	68
Tabla 10. Constantes PID Ensayo1 Tipo2.....	68
Tabla 11. Constantes PID Ensayo2 Tipo2.....	69
Tabla 12. Constantes PID Ensayo3 Tipo2.....	70
Tabla 13. Constantes PID Ensayo4 Tipo2.....	71

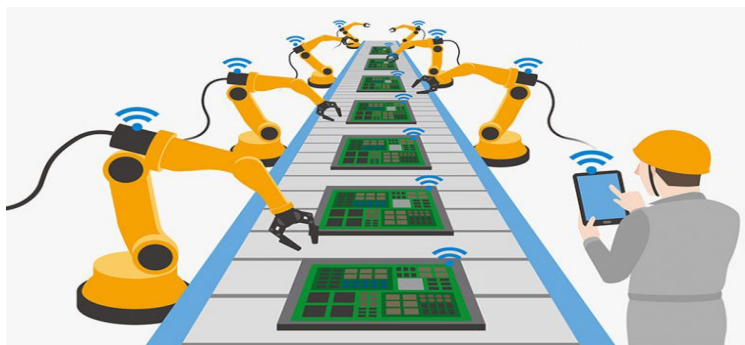
## 1. INTRODUCCIÓN.

En este apartado se procede a mencionar y desarrollar las razones por las que se ha llevado a cabo este proyecto, cómo ha surgido, cuáles son sus objetivos y las diferentes opciones que puedes encontrar hoy en día en el mercado con las mismas funcionalidades o similares, para realizar proyectos con las mismas características.

### 1.1 Motivación.

En la actualidad, la automatización está teniendo mucha importancia, tanto es así que en la mayoría de sectores de trabajo, desde el sector primario (agricultura, ganadería...), pasando por el sector secundario (industria, construcción...), hasta el sector terciario (sanidad, comunicación...), están renovando sus equipos, haciéndolos más cómodos para el usuario y simplificando la producción.

Podemos definir la automatización como un sistema donde se transfieren tareas de producción, realizadas habitualmente por operadores humanos a un conjunto de elementos tecnológicos. Un sistema automatizado consta de dos partes principales: Parte de Mando y Parte Operativa. La parte operativa son los actuadores, es decir, la parte del sistema que lleva a cabo las órdenes realizadas por el controlador. La parte de mando es la que está ligada al controlador, es decir el centro del sistema, el que realiza las órdenes y gestiona las señales de los sensores, en nuestro caso una placa de Arduino.



*Ilustración 1. Automatización.*

Este proyecto se ha llevado a cabo con la intención de representar a escala un invernadero inteligente, consiguiendo así la posibilidad de poner en práctica tanto conocimientos prácticos (montaje de circuitos, análisis de intensidades, potencias y tensiones de los componentes, realización de una PCB, etc.), como conocimientos teóricos (programación, creación de circuitos, cálculos y selección de los componentes, etc.).

El invernadero inteligente, surgió también como una manera práctica y atractiva de enseñar y mostrar, en una clase, la funcionalidad y la representación de automática y control por computador. Haciendo posible la interacción entre el invernadero y el usuario.

## **1.2 Estado del arte.**

Este apartado recoge algunas de las diferentes opciones que existen hoy en día en el mercado, desde invernaderos inteligentes desarrollados por grupos de investigación (maquetas), pasando por invernaderos inteligentes desarrollados para casas particulares, hasta invernaderos inteligentes desarrollados por compañías con estándares de producción. Se indicarán también, sus componentes, tanto sensores como actuadores.

### **1.2.1 Invernaderos de Grupo de Investigación**

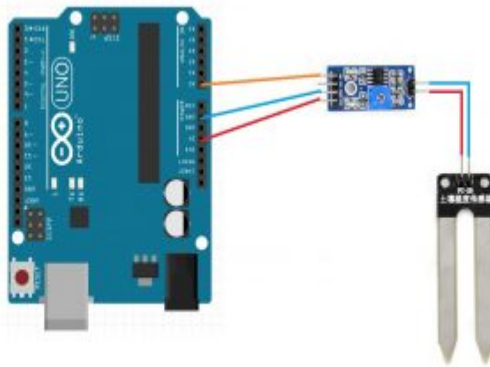
En este apartado, se exponen las diferentes maquetas que han desarrollado grupos de institutos y grupos de trabajo de la universidad. Al haber tanta oferta, de diferentes tipos, comentaré el más completo.

#### **1.2.1.1 Domotic GreenHouse**

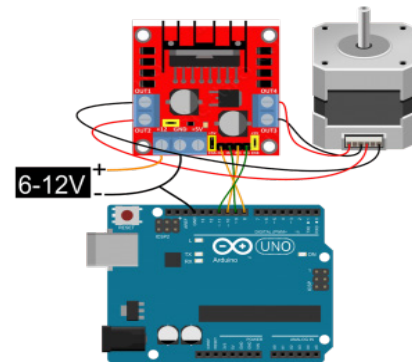
Este proyecto está desarrollado por un grupo de 2º de Bachiller de un instituto. Su idea se basa en una maqueta en la que a través de un sitio web puedan ver y controlar gráficamente 4 parámetros o sensores: temperatura, humedad del aire, humedad del suelo y apertura/clausura de la puerta.

El sistema está compuesto por una Raspberry Pi 3, un Arduino UNO R3, un sensor DHT22, un sensor de humedad del suelo, un controlador del motor paso a paso con L298N, un motor paso a paso 17HS2408 4-lead Nema 17, una bomba de agua y un relé electrónico.

El funcionamiento es el siguiente: el Arduino, obtiene las variables de los sensores (humedad del aire y del suelo y temperatura) a través de sus pines digitales. También estarán conectados a los pines digitales tanto el relé con la bomba como el controlador del motor paso a paso. Arduino compara por medio de un programa los valores de los sensores con unos valores preestablecidos, y según estas comparaciones se activará/desactivará cada uno de los actuadores (siendo su control todo/nada en los dos casos).

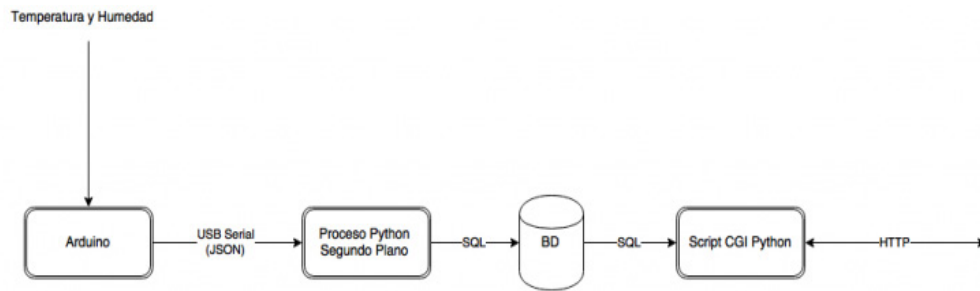


*Ilustración 3. Sensor humedad*



*Ilustración 2. Motor paso a paso*

Por otro lado, proceden a la conexión de la Raspberry, utilizando un software de base de datos llamado Mysql, además de un programa realizado en Python y un servidor web Apache. La Raspberry está en constante intercambio de datos con Arduino mediante una comunicación serial programada en Python y se encarga de la gestión de la base de datos y de mostrar los datos en el servidor Apache.

*Ilustración 4. Esquema del proceso*

### 1.2.2 IOGreenhouse

Corresponde con el primer sistema de invernadero inteligente para el hogar. Controla los principales parámetros de crecimiento de los cultivos ayudado por sensores que facilitan un cultivo guiado para profanos en la huerta.

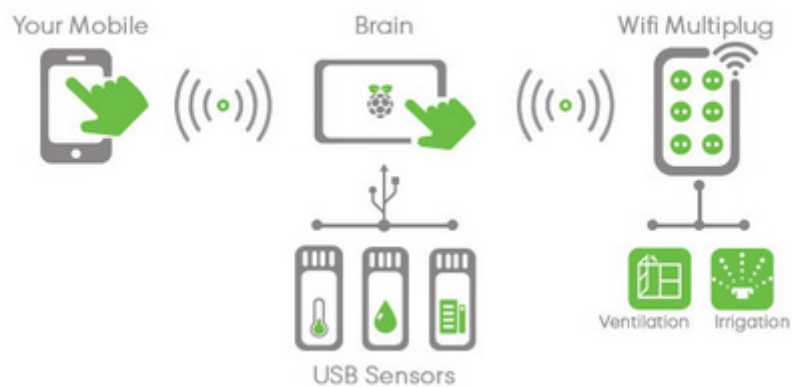
IOGreenhouse (intelligent organic greenhouse), es una solución para personas que quieren cultivar y obtener alimentos ecológicos en su propia casa. Este invernadero se basa en la comunicación instantánea con el usuario vía Wifi, haciendo posible que el usuario controle las variables de su invernadero tanto desde su smartphone, Tablet u ordenador.

Existen diferentes tamaños de invernaderos, el estándar sería de 3x2 metros. El invernadero está equipado con un procesador, que se encarga de transmitir la información a la app y de gestionar los actuadores y los sensores. Está equipado con sensores de humedad, temperatura, luminosidad y ventilación. Se basa también en una tecnología Plug&Play, haciéndolo un sistema sencillo y adaptable para cualquier persona. Además, se encarga de reutilizar agua de lluvia y emplear una placa fotovoltaica para la alimentación de sus actuadores y de su controlador.



*Ilustración 5. Materiales IOGreenhouse.*

IOGreenhouse, no es solo un invernadero inteligente, es una comunidad, ya que dispone de intercambio bidireccional entre usuarios, haciendo posible compartir recetas de platos, mejoras de los actuadores y sensores, recomendación de semillas, etc.



*Ilustración 6. Conexiones IOGreenhouse.*

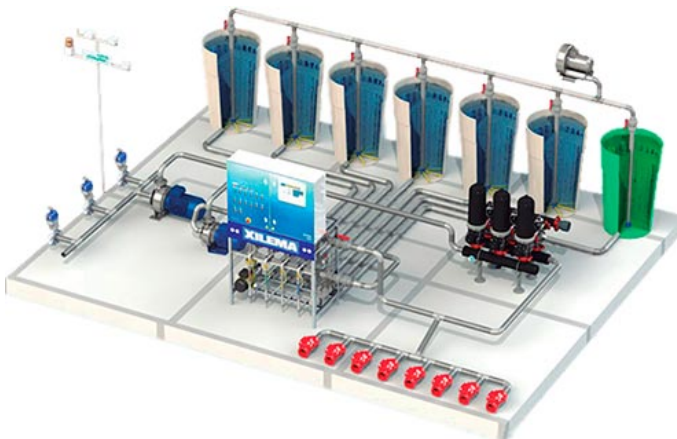
### 1.2.3 Novagric

Novagric es una empresa que se encarga, entre otros, de la producción de invernaderos a nivel comercial, es decir, grandes producciones. Se basan en el empleo de un ordenador central al que se conectan un conjunto de sensores, que recogen las variaciones de los distintos parámetros respecto a unos valores programados inicialmente. Estos sistemas están a su vez conectados a los sistemas de riego y de climatización. Los sensores o automatismos se distribuyen en diferentes sectores y pueden funcionar de forma autónoma.

Tienen distintos parámetros de control, cada uno con su conjunto de sensores, los cuales les ofrecen una orientación cualitativa de las condiciones ambientales del cultivo.

- **Control del Riego:**

- Xilema: Es un sistema de fertirrigación controlada en los cultivos de alto rendimiento. Se basa en una plataforma a través de internet que permite programar y visualizar el estado de los riegos. Tiene distintos tipos de programación, según el criterio del agricultor: Programación por tiempos, por volúmenes medidos, en base a sensores del estado hídrico del suelo y la planta, basada en datos meteorológicos.
- Carros de riego: Empleado para semilleros que permiten automatizar el riego, permitiéndonos así hacer un reparto uniforme de los aportes de agua y fertilizantes.



*Ilustración 8. Xilema.*



*Ilustración 7. Carros de riego*



- **Climatización:**



*Ilustración 9. Ventilación forzada.*



*Ilustración 10. Calefacción Agua.*



*Ilustración 11. Cooling.*

- **Climatec:** Es un controlador climático que es capaz de adelantarse a las necesidades climatológicas de cualquier tipo de plantación, así como controlar los diferentes factores climáticos (la ventilación, la calefacción, las pantallas térmicas o de sombreo, la humidificación, los destratificadores), las alarmas y los diferentes programas estándar.
- **Ventilación:** Se basa en dos tipos: Forzada, basada en ventiladores haciendo posible el flujo y la regulación de la temperatura del aire. Natural, con aperturas y cierres de las ventanas del invernadero mediante motores y brazos.
- **Calefacción:** Hay dos tipos: Por agua, mediante el calentamiento de agua en tanques, se distribuye al invernadero mediante una serie de motores conectados a unos tubos, teniendo un flujo de ida y otro de vuelta. La apertura se realiza a través de electroválvulas. Y por aire, mismo método de transmisión, pero en vez de tanques usa un generador de aire caliente.
- **Cooling:** Es un sistema de paneles de refrigeración evaporativos, que están destinados a proveer la temperatura y la humedad más adecuada. Están dispuestos de: Ventiladores, son helicoidal y tienen un funcionamiento en modo extractor. Panel evaporativo, fabricado con hojas de celulosa

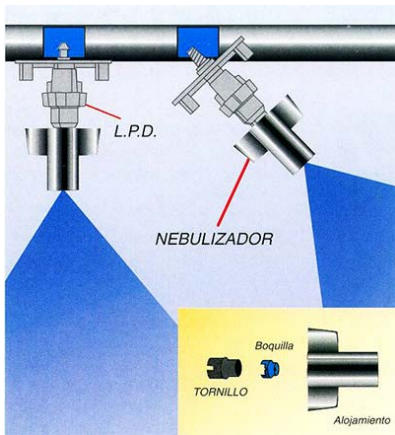


Ilustración 12. Fog System/Fogger

plegadas en distintos ángulos. Pared húmeda de paneles evaporativos.

- Fog System: Sistema que se basa en la creación de niebla a alta presión. Este sistema está constituido por: Equipo de pretratamiento físico-químico, equipo de presurización, redes de distribución y boquillas de pulverización.
- Fogger: Mismo sistema que el Fog System, pero a alta presión.

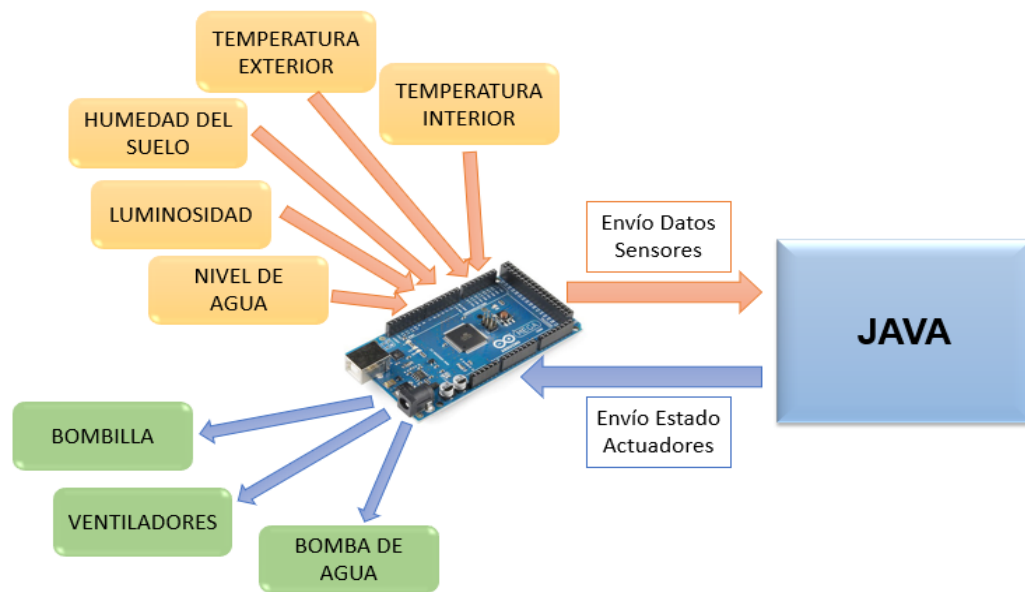
### 1.3 Objetivos.

En este apartado comentaremos tanto del objetivo general como de los objetivos específicos de este proyecto. Estos objetivos vienen dados por unos conocimientos previos, en este caso, conocimientos de programación en Arduino, programación en Java, conocimientos básicos en control y en automática, así como conocimientos en circuitos electrónicos y en sus componentes.

#### 1.3.1 Objetivo General

Nuestro objetivo general es la monitorización de temperaturas (interior y exterior), humedad del suelo, luminosidad y nivel de agua. Así como la acción sobre los actuadores de la bombilla, ventiladores y bomba de agua. Todo esto mediante un microcontrolador, en este caso Arduino, que lee las variables recibidas por los sensores y comunicándose por puerto serie, las muestra en un programa realizado en Java. De igual forma, pero en sentido inverso, desde la interacción del usuario con Java, se activan o desactivan las salidas de los actuadores.

Este proceso está reflejado en la ilustración13:

*Ilustración 13. Objetivo General*

### 1.3.2 Objetivos Específicos

En este apartado explicamos cada uno de los objetivos del proyecto. Cabe destacar que los apartados van de forma progresiva, es decir, sin el apartado anterior ejecutado, el consecuente no se puede realizar. Tenemos los siguientes objetivos:

- 1) Selección de componentes.
- 2) Desarrollar un programa en Arduino para la gestión de sensores y actuadores.
- 3) Desarrollar una aplicación para poder controlar desde PC los parámetros del invernadero (Java).
- 4) Implementar a la aplicación (Java) una captura de datos independiente a la actuación sobre actuadores o la monitorización de los sensores.
- 5) Implementación de un modelo PID.
- 6) Realización de una PCB. Diseño y ejecución en OrCad (Capture y PCB editor). Elaboración de la PCB en una shield (prefabricada) de Arduino.

7) Obtención de resultados y comparación con los resultados teóricos.

## 2. METODOLOGÍA

En este apartado se abordarán todas las herramientas hardware y software empleadas para la realización de este proyecto. Se explicará también el funcionamiento de cada una de ellas y se justificará el porqué de su utilización.

### 2.1 Desarrollo Hardware

A continuación, se desarrollarán los dispositivos que forman la parte relativa al hardware del sistema desarrollado.

#### 2.1.1 Sensores

Los sensores son unos dispositivos empleados para adquirir de la forma más aproximada variables medibles o cuantificables. Los sensores instalados en este proyecto son, sensor de temperatura tipo SHT31 (para la temperatura interior), sensor de temperatura tipo DHT11 (para la temperatura exterior), sensor tipo SKU SEN0114 (para la humedad del suelo), sensor tipo LDR (para la luminosidad) y sensor tipo LS02 (para el nivel de agua del tanque).

##### 2.1.1.1 Sensor de temperatura tipo SHT31

El sensor digital de temperatura y humedad relativa SHT31 permite realizar mediciones muy precisas a bajo coste. Posee mejores prestaciones respecto a los sensores DHT11 y DHT22, entre ellas la resolución, mayor precisión y un empaquetamiento más compacto. Está muy empleado en aplicaciones de control automático ya que, entre otros, sus tiempos de muestreo son rápidos.

En la tabla1 podemos observar las especificaciones técnicas de este sensor:

<b>Voltaje de Operación</b>	2.4V a 5.5V DC
<b>Interfaz de comunicación</b>	I2C
<b>Dirección I2C</b>	0x44
<b>Rango de trabajo de temperatura</b>	-40°C a 125°C
<b>Resolución temperatura</b>	0.015°C
<b>Precisión temperatura</b>	0.2°C
<b>Rango de trabajo de humedad</b>	0 a 100% RH
<b>Resolución humedad</b>	0.01% RH
<b>Precisión humedad</b>	2% RH

*Tabla 1. Especificaciones Técnicas SHT31*

Posee una integración muy sencilla tanto a nivel de software como de hardware. A nivel de software, hay disponibilidad de librerías para Arduino con soporte para el protocolo I2C. En lo relacionado con el hardware, son necesarios cuatro cables, dos de alimentación (Vcc y Gnd) y dos para el bus de datos I2C, como podemos ver en la ilustración14.



*Ilustración 14. Sensor SHT31*

El protocolo de datos I2C se basa en la comunicación varios maestros y uno o múltiples esclavos. Toma e integra lo mejor de los protocolos SPI y UART. Emplea dos vías de comunicación, SDA y SCL. SDA (Serial Data), es la vía de comunicación entre el maestro y el esclavo. SCL (Serial Clock), es la vía por donde se transmite la señal de reloj.

En la tabla2, podemos ver algunas especificaciones o características de este tipo de comunicación:

<b>Número de vías o cables</b>	2
<b>Velocidad máxima</b>	Modo estándar (Sm) = 100kbps
	Modo rápido (Fm) = 400kbps
	Modo High Speed (Fm+) = 3.4Mbps
	Modo Ultra Fast (Hs-mode) = 5Mbps
<b>Síncrono o Asíncrono</b>	Síncrono
<b>Paralelo o Serial</b>	Serial
<b>Número máximo de maestros</b>	Ilimitado
<b>Número máximo de esclavos</b>	1008

Tabla 2. Características I2C

El funcionamiento del protocolo I2C es muy sencilla, la información viaja en mensajes. Los mensajes están divididos en tramas de datos. Cada mensaje lleva una trama con una dirección la cuál transporta la dirección binaria del esclavo al que va dirigido el mensaje, y una o más tramas que llevan la información del mensaje.

El mensaje también contiene condiciones de inicio y paro, lectura y escritura de bits, y los bits ACK y NACK. Todo esto va entre cada sección de datos. La ilustración15 muestra la división del mensaje.



Ilustración 15. División Mensaje I2C

#### 2.1.1.2 Sensor de temperatura tipo DHT11

El sensor DHT11 presume de ser un sensor con alta fiabilidad y estabilidad debido a señal digital calibrada. Gracias a que su transmisión de datos es mediante una vía digital, podemos obtener una señal exenta de ruido.

Los DHT11 se componen de un sensor capacitivo para medir la humedad y de un termistor. Existen tres tipos de encapsulados en el mercado:



- Sensor suelto: con cuatro pines disponibles para conectar. En este caso es necesario añadir una resistencia pull-up.
- El sensor con una placa soldada, con tres pines disponibles para conectar y una resistencia pull-up integrada (entre 4.7-10k $\Omega$ ).
- El sensor con una placa soldada, pero con un condensador de filtrado (normalmente de 100nF).

La tabla3, recoge las principales características técnicas del sensor DHT11:

DHT11	
<b>Alimentación</b>	3.5V - 5V
<b>Consumo</b>	2.5mA
<b>Señal de salida</b>	Digital
Temperatura	
<b>Rango</b>	0°C - 50°C
<b>Precisión</b>	$\pm 2^{\circ}\text{C}$
<b>Resolución</b>	1°C
Humedad	
<b>Rango</b>	20% - 90% RH
<b>Precisión</b>	$\pm 5\% \text{ RH}$
<b>Resolución</b>	1% RH

Tabla 3. Características técnicas DHT11

El conexionado es muy sencillo, en nuestro caso al tener PCB soldada, tenemos solo tres pines para conectar. La ilustración16 nos lo muestra.

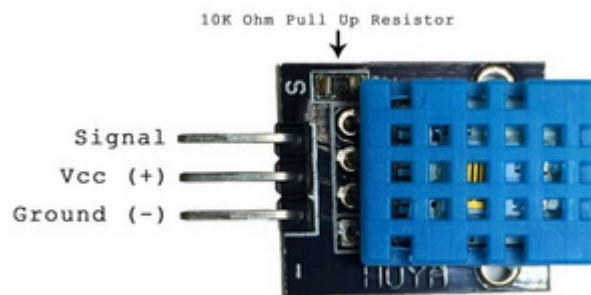


Ilustración 16. Conexiones sensor DHT11

Se trata de un dispositivo analógico, pero dentro del propio dispositivo se hace la conversión entre analógico y digital. Por tanto, partimos de una señal analógica que luego es convertida en formato digital y que se enviará al

microcontrolador. La trama de datos es de 40 bits, divididos en dos informaciones la de temperatura y la de humedad, como está representado en la ilustración 17.

0011 0101      0000 0000      0001 1000      0000 0000      0100 1001  
 8 bits humedad    8 bits humedad    8 bits temperatura    8 bits temperatura    bits de paridad

*Ilustración 17. División Mensaje DHT11*

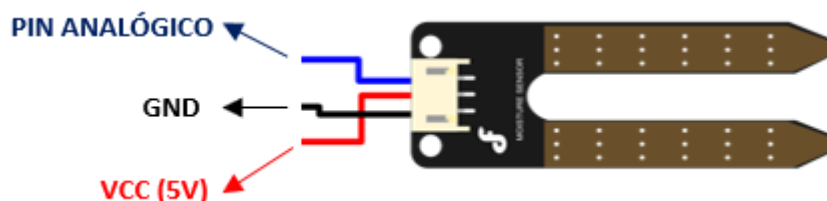
El primer grupo de 8 bits, corresponde a la parte entera de la humedad y el segundo grupo a la parte decimal. Lo mismo ocurre con el tercer y cuarto grupo, pero en este caso de temperatura. El último grupo corresponde con los bits de paridad, empleados para confirmar que no hay datos corruptos. Suma los 4 primeros grupos de 8 bits, y esta suma debe ser igual a los bits de paridad.

#### 2.1.1.3 Sensor de humedad del suelo tipo SKU SEN0114

Este sensor de humedad puede leer la cantidad de humedad presente en el suelo que lo rodea. Es un sensor de baja tecnología, pero ideal para monitorizar pequeños jardines o maquetas.

Este sensor utiliza las dos sondas para pasar una corriente a través del suelo, luego lee la resistencia que hay entre ellas y así obtiene el nivel de humedad del suelo. Un suelo más húmedo, conducirá electricidad más fácilmente, por tanto, tendrá una menor resistencia. Un suelo más seco, conducirá menos electricidad, por tanto, tendrá una resistencia mayor.

Este valor de resistencia se envía a un pin analógico del Arduino, donde se procede a leerlo. La ilustración 18, nos muestra las conexiones de este sensor



*Ilustración 18. Conexiones sensor SEN0114*



En la siguiente tabla, se muestran las características más importantes del sensor SEN0114.

<b>Alimentación</b>	3.3V – 5V
<b>Tensión de salida</b>	0V – 4.2V
<b>Corriente de entrada</b>	35mA
<b>Tamaño</b>	60x20x5mm
<b>Rango</b>	0 – 300 suelo seco
	300 – 700 suelo húmedo
	700 – 950 en agua

Tabla 4. Características técnicas SEN0114

#### 2.1.1.4 Sensor de luminosidad tipo LDR

LDR proviene de sus siglas en inglés (Light Dependent Resistor) o fotorresistor, es una resistencia la cual varía su valor en función de la cantidad de luz que incide sobre su superficie. Cuanto mayor sea la intensidad de luz que incide en la superficie del LDR menor será su resistencia y cuanto menor sea la luz que incida sobre este mayor será su resistencia.

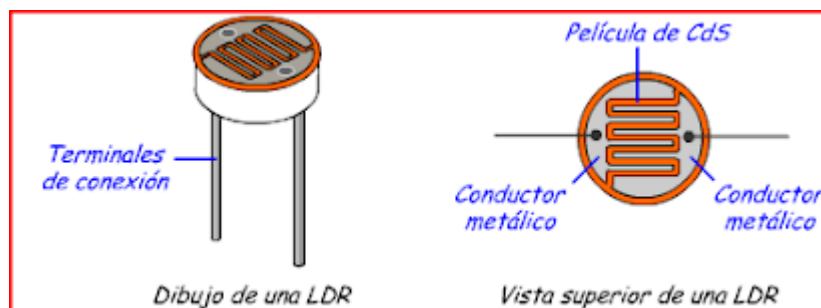


Ilustración 19. Componentes sensor LDR

Su funcionamiento es el siguiente, cuando el LDR no está expuesto a radiaciones luminosas, los electrones están unidos en los átomos que lo conforman, pero cuando incide radiación luminosa sobre él, esta energía libera a los electrones y el material se hace más conductor y de esta manera se reduce su resistencia.

Hay dos tipos de clasificación de los LDR, según su material fotosensible o según su linealidad. Los materiales de cristal semiconductor fotosensible más empleados para su fabricación son el sulfuro de talio, el seleniuro de cadmio, el sulfuro de cadmio y el sulfuro de plomo (estos dos últimos son los más comunes):

- Sulfuro de cadmio: son extremadamente sensibles a todo tipo de radiaciones luminosas que son visibles en el espectro del ser humano
- Sulfuro de plomo: son especialmente sensibles a las radiaciones infrarrojas.

Según la linealidad, obtenemos dos tipos:

- Fotorresistencia lineal: más conocidos como fotodiodos, aunque en algunas aplicaciones es posible utilizarlos como fotorresistores debido al comportamiento lineal que presentan. Se polariza de manera inversa. Su funcionamiento está reflejado en la ilustración 20.

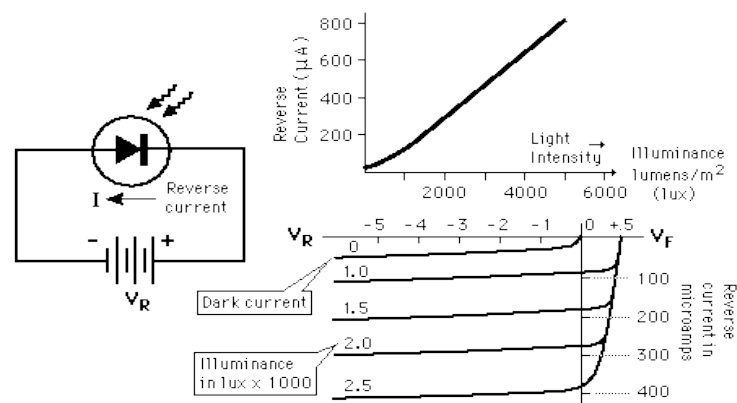


Ilustración 20. Fotodiodo

- Fotorresistencia no lineal: son las más comunes y son aquellas cuyo comportamiento no depende de la polaridad con la que se conecte. Su funcionamiento está representado en la ilustración 21.

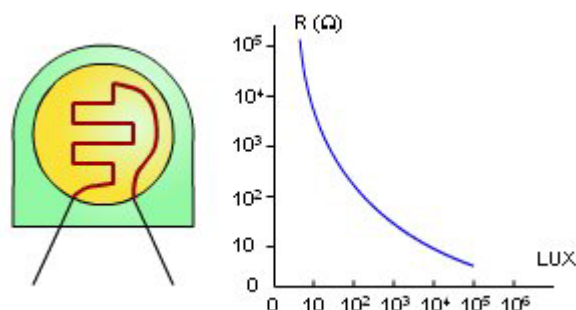
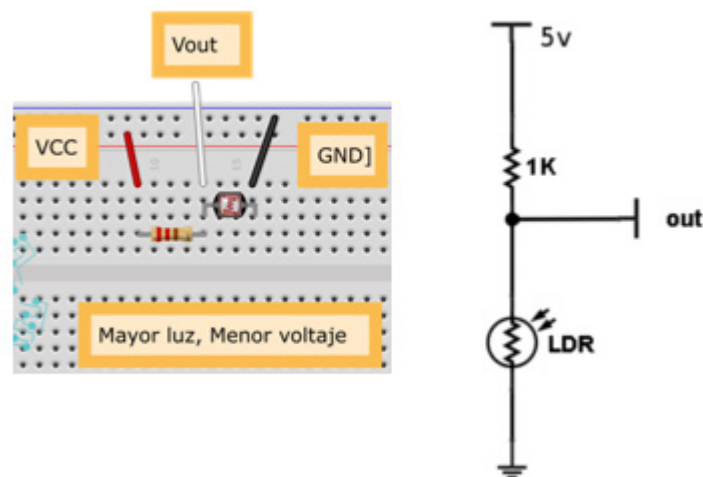
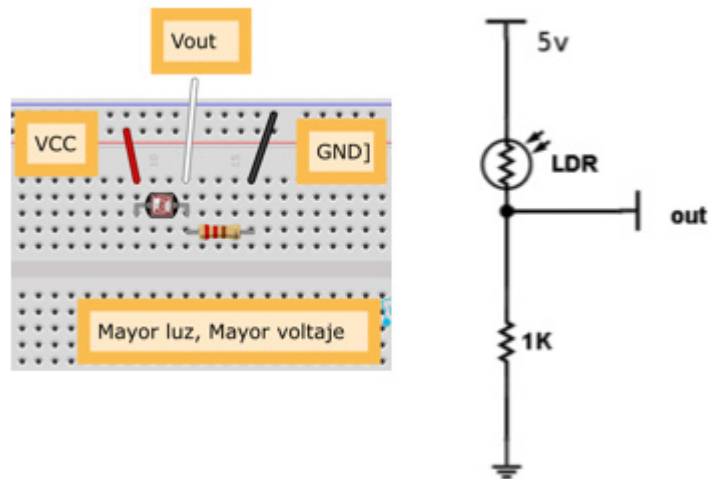


Ilustración 21. Fotorresistencia no lineal

Para obtener una lectura analógica de un LDR, debemos hacer un montaje de un divisor de tensión, de donde obtendremos una señal que leeremos por uno de los pines analógicos del Arduino. Hay dos posibles montajes, según como queramos realizar la lectura. Las ilustraciones, ilustración22 e ilustración23, muestran estos montajes.



#### 2.1.1.5 Sensor de nivel de agua tipo LS02

Los sensores de nivel de agua o interruptores de flotador, es un dispositivo electrónico que mide la altura del material, generalmente líquido, dentro de un tanque u otro recipiente.

Los contactos reed están formados por 2 láminas de material ferromagnético, herméticamente selladas dentro de una cápsula de vidrio. Dichas láminas están ligeramente separadas entre sí y se ponen en contacto en presencia de un campo magnético.

Obtenemos así dos casos, normalmente abierto y normalmente cerrado. Actuando el flotador como un interruptor (véase en la Ilustración 24).

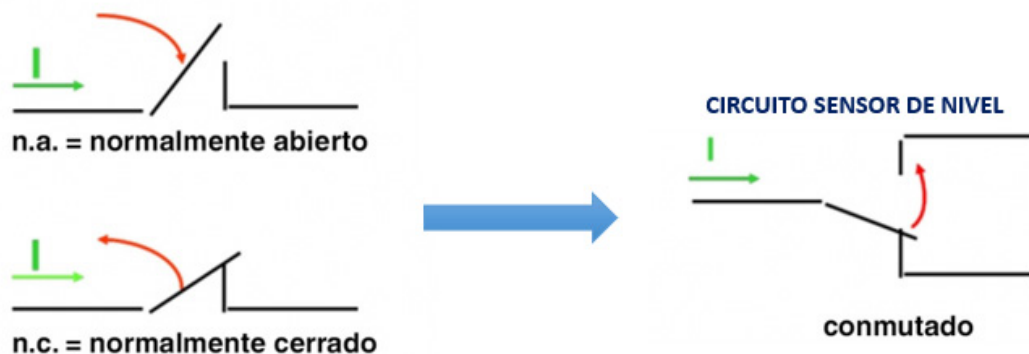


Ilustración 24. Esquema Funcionamiento LS02

Debido a que las láminas se encuentran bajo una precarga mecánica, se debe aplicar un campo magnético para que el contacto se cierre o abra para generar la conmutación deseada y eso se logra gracias al imán interno del flotador. Los collarines de ajuste instalados actúan como topes para situar el flotador en la posición correcta.

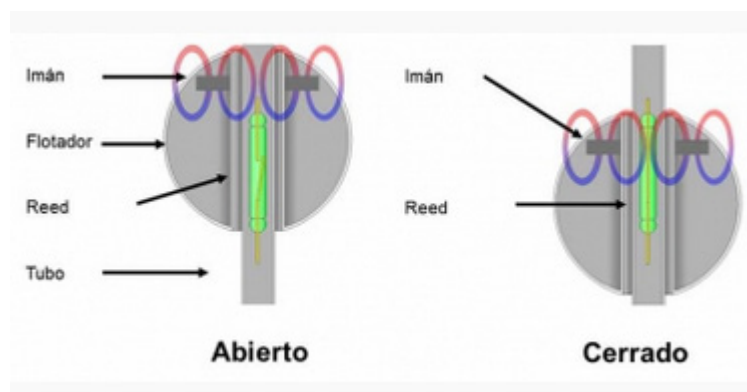
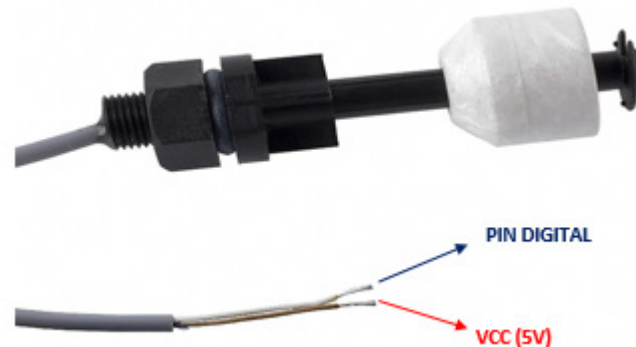


Ilustración 25. Esquema Posiciones Flotador LS02

En cuanto a conexión, este dispositivo contiene dos polos (o cables), uno de ellos dirigido a alimentación (5V) y el otro al pin digital. Cada vez que se cierre

el circuito del flotador, obtendremos el pulso de alimentación en el pin digital. En la figura siguiente, se muestra la conexión de este sensor:



*Ilustración 26. Conexiones LS02*

### 2.1.2 Actuadores

Son los encargados de llevar a cabo los cambios de estado del invernadero, desde incrementar la temperatura, consiguiendo disminuir la humedad del suelo, pasando por crear flujos de aire de forma que seamos capaces de evacuar el exceso de temperatura, hasta incrementar la humedad del suelo por medio de un sistema de riego. Los actuadores empleados en este proyecto han sido una bombilla incandescente 60W 24Vdc, un sistema de tres ventiladores (uno de entrada de aire, dos de salida de aire) 12Vdc y una bomba de agua 12Vdc 10l/min 0.5bar.

#### 2.1.2.1 Bombilla incandescente 60W 24Vdc

Se denomina bombilla incandescente al dispositivo que produce luz mediante el calentamiento por Efecto Joule de un filamento metálico, mediante el paso de una corriente eléctrica.

Las lámparas incandescentes están formadas por un hilo de wolframio que se calienta por efecto Joule alcanzando temperaturas tan elevadas que empieza a emitir luz visible. Para evitar que el filamento se queme en contacto con el aire, se rodea con una esfera de vidrio a la que se le ha hecho el vacío o se ha rellenado con un gas. Se completa el conjunto con una selección de elementos con funciones de soporte y conducción de corriente eléctrica, así como un

casquillo normalizado que sirve para conectar la lámpara (E27). La ilustración 27 nos muestra las partes de la bombilla.



Ilustración 27. Partes de una bombilla

En nuestro caso, la conexión de la bombilla se muestra en la figura, donde el V+ del Mosfet, proviene directamente de la fuente de alimentación y V- proviene de la puerta del mosfet.

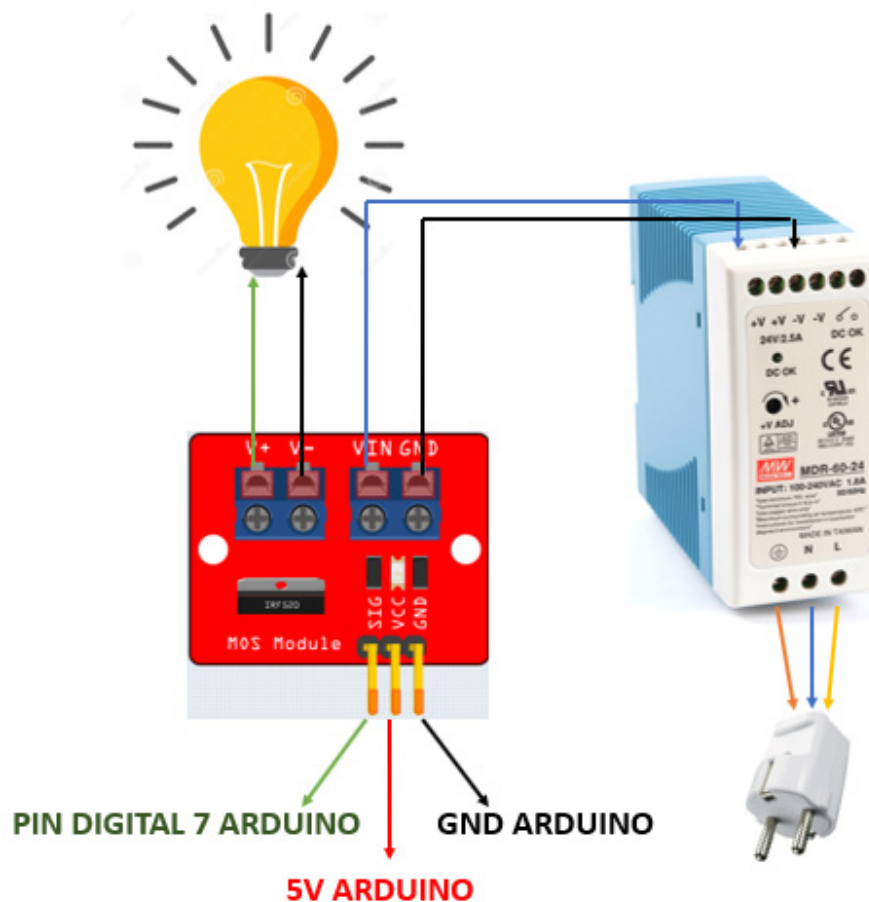


Ilustración 28. Conexiones Mosfet Bombilla

La elección del empleo de una bombilla de 60W 24Vdc, proviene de la limitación del mosfet (el cual comentaremos más adelante), ya que el módulo solo admite tensiones de hasta 24V. Esta bombilla nos da la capacidad de poder cambiar la temperatura en un corto periodo de tiempo, haciendo posible reflejar su actuación sobre la variable temperatura en las gráficas.

#### 2.1.2.2 Ventilador 12Vdc

Un ventilador, es un dispositivo empleado para refrigerar o enfriar un emplazamiento o lugar, impulsando aire a una presión moderada mediante un motor que hace girar unas aspas.

En este caso hemos usado ventiladores que su utilización está destinada para la refrigeración de ordenadores, pero al ser un invernadero a escala, nos reproduce la función a la perfección. Al tener tres ventiladores, empleamos uno de ellos (el más grande) para la insuflación de aire, mientras que los otros dos, se dedican a la expulsión del aire, creando así un flujo de aire constante y en una dirección.

La conexión de los ventiladores es una conexión en paralelo, es decir, un solo mosfet nos alimenta a los tres ventiladores.

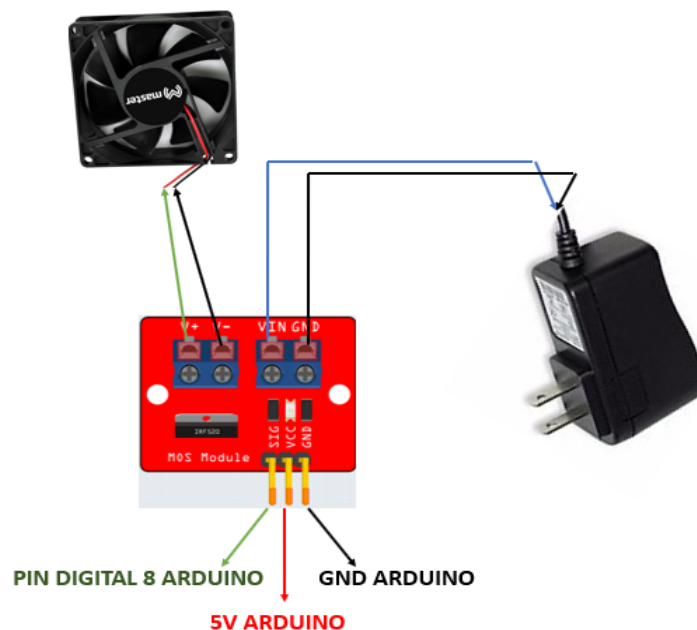
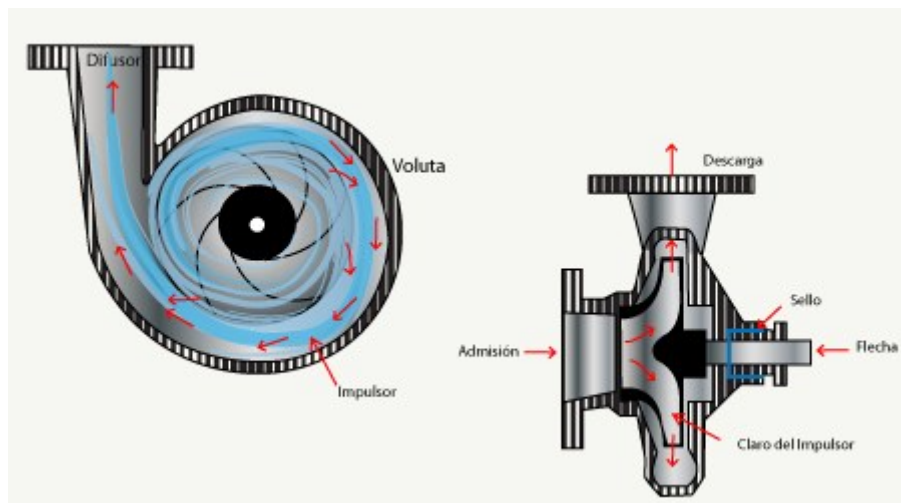


Ilustración 29. Conexiones Mosfet Ventiladores

### 2.1.2.3 Bomba de Agua 12Vdc 10l/min 0.5bar

Una bomba de agua es un dispositivo que se encarga de hacer fluir el agua por un medio. El agua es aspirada para luego ser impulsada por un motor que utiliza bobinas e imanes para crear un campo magnético y así lograr que el impulsor gire de una manera continua.

La ilustración30 nos muestra un pequeño esquema de su funcionamiento.



*Ilustración 30. Funcionamiento Bomba de Agua*

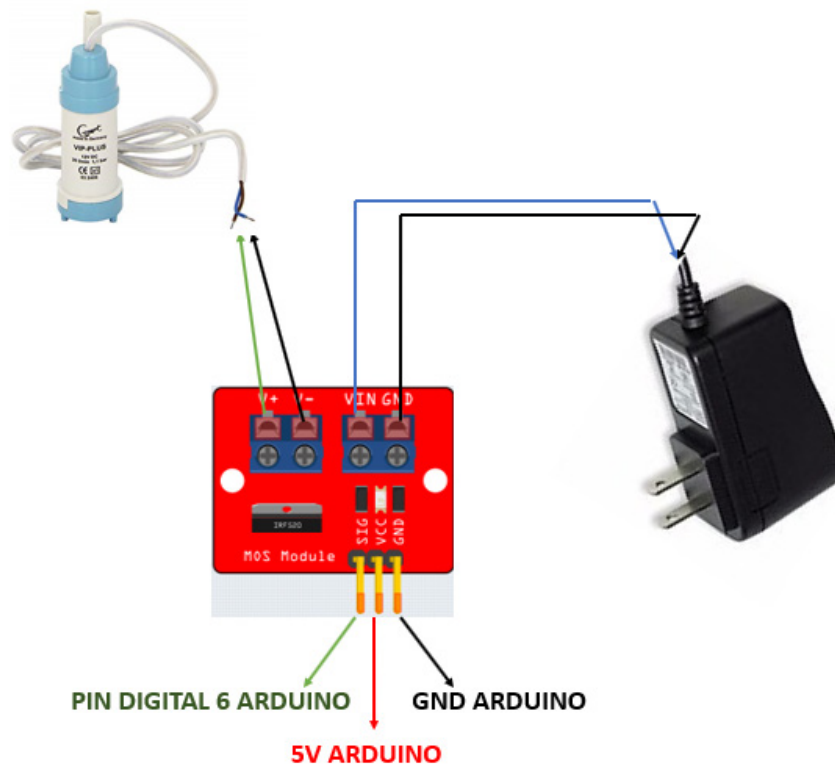
En nuestro caso, hemos utilizado una bomba que trabaja con una tensión continua de 12V, ya que con sus características se adapta a la perfección a nuestro invernadero a escala. En la siguiente tabla se muestran algunas de las características más importantes de nuestra bomba de agua.

<b>Caudal</b>	10l
<b>Presión</b>	0.5bar
<b>Altura max</b>	7m
<b>Consumo</b>	15W – 25W

*Tabla 5. Características Bomba de Agua*

La conexión es igual a la de los ventiladores, solo que ahora en vez de ventiladores como cargar, tenemos una bomba de agua. La figura nos muestra esta conexión.



*Ilustración 31. Conexión Bomba de Agua*

### 2.1.3 Placa de Arduino

Arduino es una plataforma de software y hardware libre. Las placas de desarrollo de Arduino son las más utilizadas debido a su reducido precio. También se debe a su facilidad de uso y, sobre todo, a la gran comunidad que posee.

Existen diferentes tipos de Arduino, los cuales incorporan un microcontrolador Atmel y se programan en un lenguaje basado en C, llamado Wiring, a través de su IDE de código abierto.

En nuestro caso nos centraremos en el Arduino Mega, el cual hemos empleado para la gestión y el desarrollo de nuestro invernadero inteligente. Se escogió esta placa por la cantidad de pines tanto digitales como analógicos que tiene, teniendo como opción la implementación de más sensores o actuadores en un futuro.

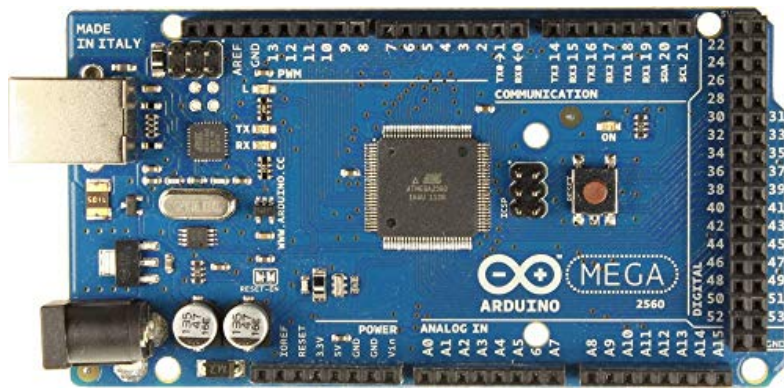


Ilustración 32. Placa Arduino Mega

En la siguiente tabla se recogen las características más importantes de esta placa de Arduino.

<b>Microcontrolador</b>	ATmega2560
<b>Tensión Operativa</b>	5V
<b>Tensión de Entrada</b>	7V – 12V
<b>Límites tensión de entrada</b>	6V – 20V
<b>Pines digitales</b>	54 (15 con salida PWM)
<b>Pines analógicos</b>	16
<b>Corriente por cada Pin</b>	40mA
<b>Corriente entregada en el pin 3.3V</b>	50mA
<b>Memoria flash</b>	259Kb (8Kb bootloader)
<b>SRAM</b>	8Kb
<b>EEPROM</b>	4Kb
<b>Clock speed</b>	16MHz

Tabla 6. Características Arduino Mega

Las entradas analógicas del modelo Mega son las correspondientes a los pines de A0 a A15. Se caracterizan por leer valores de tensión de 0 a 5 voltios con una resolución de 1024 (10bits). Si dividimos 5V entre 1024, obtenemos las variaciones que somos capaces de detectar en el nivel de la señal de entrada, siendo este de casi 5 mV.

En cuanto a los pines digitales, tenemos que destacar que las entradas son las mismas que las salidas, es decir van del 0 al 53. Se diferencian de las analógicas porque éstas son capaces de detectar sólo dos niveles de señal, LOW para valores cercanos a 0V y HIGH para valores cercanos a 5V. Otra

Reset  
SCK  
MISO

GND  
MOSI  
Vcc

MADE IN ITALY

ICSP for 16U2  
USB interface

(I2C) SCL  
(I2C) SDA

SPI is not here

Interrupt 1  
Interrupt 0

Interrupt 5  
Interrupt 4  
Interrupt 3  
Interrupt 2

MISO  
SCK  
Reset

Vcc  
MOSI  
GND

5V pins  
23  
25  
27  
29

PWM on  
44,45,46

(SPI) MISO  
(SPI) SCK

(SPI) SS

use for digital ground

7 to 12V  
DC input,  
center positive

use for analog ground

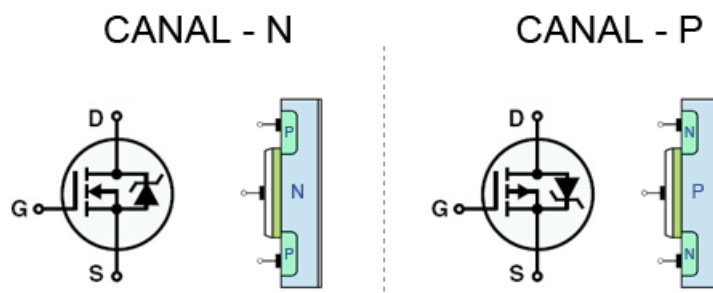
not I2C  
not I2C  
UNO shields  
can cover up

reset

ICSP for ATmega2560

### 2.1.4 Otros componentes

Un mosfet es un dispositivo semiconductor utilizado para la conmutación y amplificación de señales. Estos dispositivos poseen 3 terminales: Gate, Drain y Source, y a su vez, se subdividen en 2 tipos: los mosfet canal N y los mosfet canal P.



35

En nuestro caso, hemos empleado el Mosfet IRF520, el cual está insertado en una placa o PCB, la cual mediante unos sockets nos permite conectar nuestras cargas o dispositivos al módulo. Asimismo, el módulo consta de unas resistencias, de encapsulado SMD, que nos protegen el módulo de corrientes parásitas, además de un led de funcionamiento. Este Mosfet nos permite manejar cargas de más de 2A, en concreto de hasta 9A. Además de soportar control por PWM. En nuestro caso, las cargas que controla son: una bombilla de 24V 60W, tres ventiladores de 12V y una bomba de agua de 12V. Los transistores de tipo Mosfet, presentan mejores características que los BJT en aplicaciones de encendido/apagado de cargas de alto amperaje. Para activar el Mosfet se debe enviar 5V al Gate del Mosfet, esto permitirá que la corriente fluya a través de la carga y esta se active.

<b>Tensión de salida</b>	0 – 24Vdc
<b>Tensión de control</b>	5V TTL
<b>Corriente máxima</b>	9A
<b>Corriente nominal</b>	6A

Tabla 7. Características IRF520

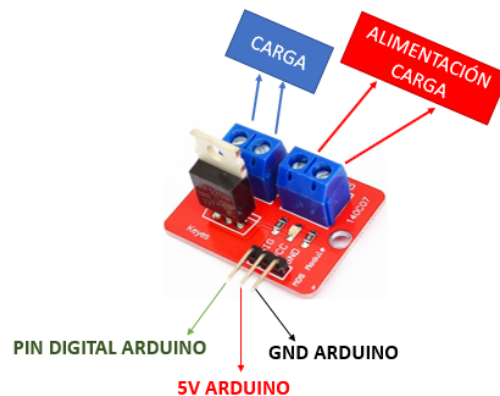


Ilustración 35. Conexión IRF520

#### 2.1.4.2 Fuente de Alimentación

Una fuente de alimentación es un dispositivo encargado de convertir la corriente alterna AC en una o varias corrientes continuas DC, las cuales son utilizadas para alimentar el sistema.

Las funciones que realiza una fuente de alimentación son las siguientes:

- **Transformación:** se consigue reducir la tensión de entrada de la fuente (220V – 240V), que son las que suministra la red eléctrica. Se emplea para tal proceso un transformador en bobina. La salida de este proceso generará desde 5V hasta 12V en alterna.

- Rectificación: en esta fase se produce la conversión de corriente alterna a corriente continua, por medio de un puente rectificador o de Graetz, con el objetivo de asegurar que no se produzcan oscilaciones de voltaje en el tiempo. Esto permite que el voltaje no baje de 0V y siempre se mantenga por encima de esta cifra.
- Filtrado: en esta etapa se nivela al máximo la señal. Esto se consigue al utilizar uno o varios condensadores, que retienen la corriente y la dejan pasar lentamente
- Estabilización: una vez obtenida la señal continua y completamente plana, quedaría estabilizarla por completo.

La fuente de alimentación empleada en este proyecto transforma la corriente alterna de 220V en alterna a 24V en continua, que es la tensión necesaria para alimentar a la carga (la bombilla) que empleamos para incrementar la temperatura de nuestro sistema.



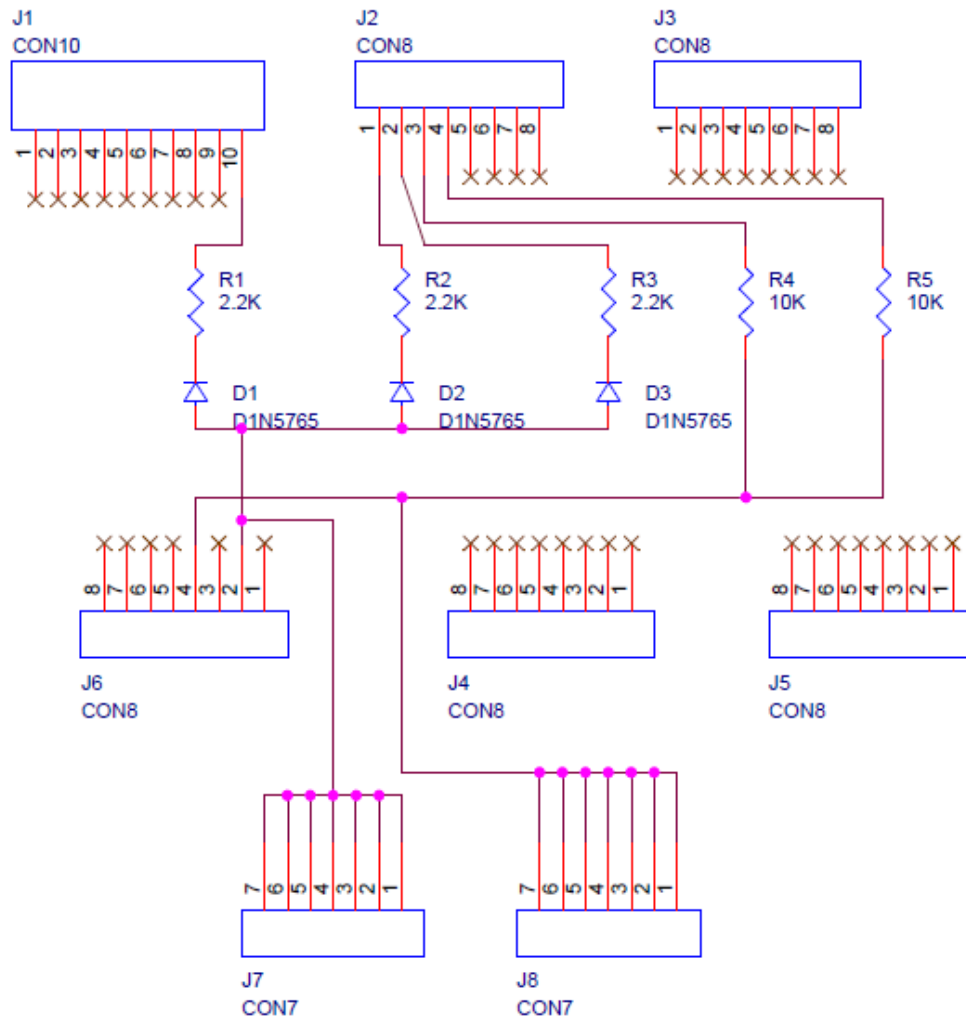
*Ilustración 36. Fuente de alimentación*

#### 2.1.4.3 PCB

Un circuito impreso es una placa de material aislante (plástico, vidrio, etc.), provista de unas pistas de cobre que sirven para interconectar los diversos componentes que constituyen el circuito en cuestión.

En nuestro caso, se diseñó un circuito impreso que sustituyera a la mayor parte del cableado, evitando así problemas externos al proyecto (el cableado no estuviera expuesto a tirones, rupturas...). Nuestra PCB consta de varias partes, la primera son los sockets, destinados a conectarse a los pines digitales y analógicos del Arduino, además de otros destinados a la alimentación de los sensores y Mosfets. La segunda parte son las resistencias de protección, destinadas a proteger la entrada de los pines digitales de posibles sobreintensidades. Y, por último, los leds, destinados a avisar al usuario de la activación o desactivación de esa salida digital.

El diseño se realizó con la herramienta de software de Orcad Capture, para más tarde, crear el circuito impreso en PCB Editor. El resultado se muestra en las siguientes dos figuras.



*Ilustración 37. PCB Orcad Capture*

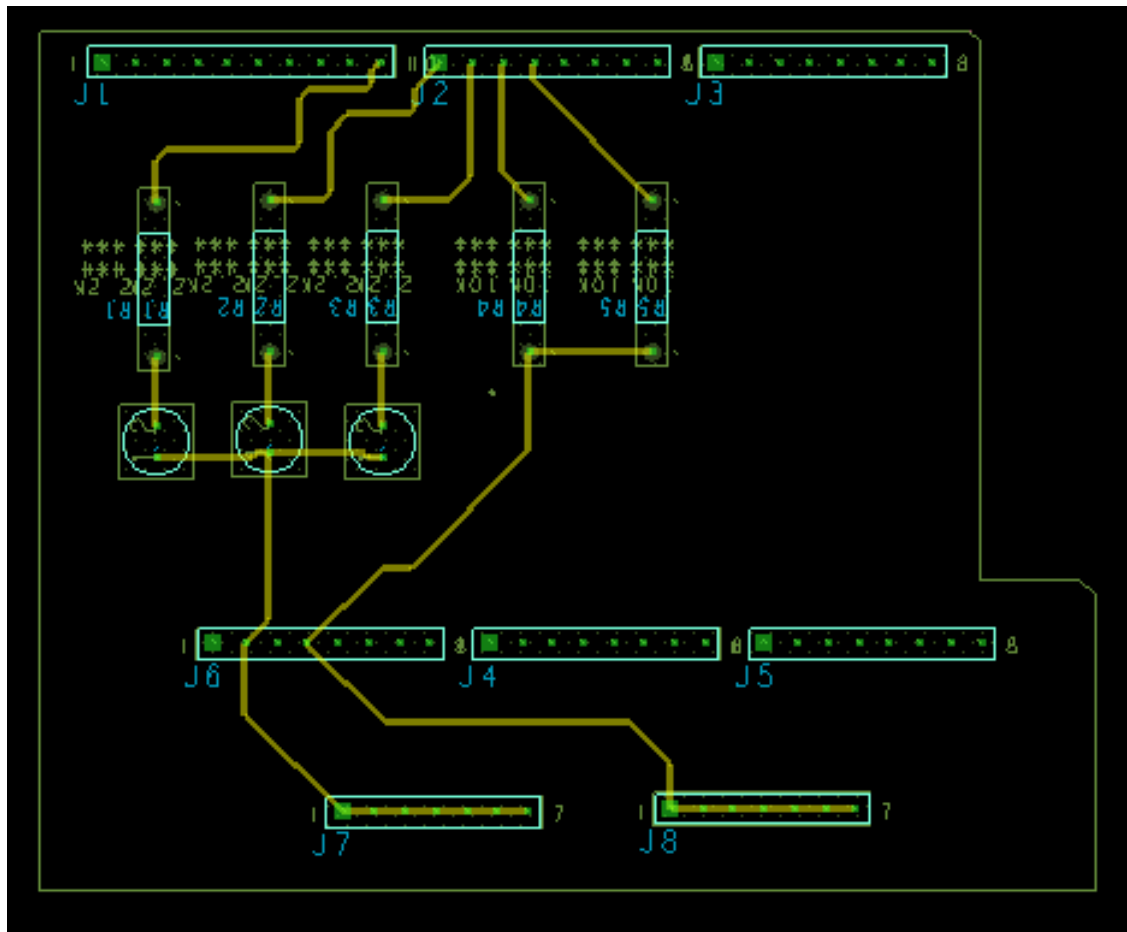


Ilustración 38. Diseño PCB

### 2.1.5 Circuito Eléctrico

El circuito eléctrico general, se ha realizado con la herramienta Fritzing, de licencia libre. Esta herramienta nos permite realizar circuitos o representaciones de ellos y nos ofrece una idea clara de lo que se quiere conseguir. Contiene librerías de la mayoría de sensores utilizados en este proyecto.

La siguiente figura nos muestra la aproximación de la representación de lo que es nuestro circuito general del invernadero a escala. Contiene los sensores utilizados, el Arduino, el circuito impreso y los mosfets de los actuadores.



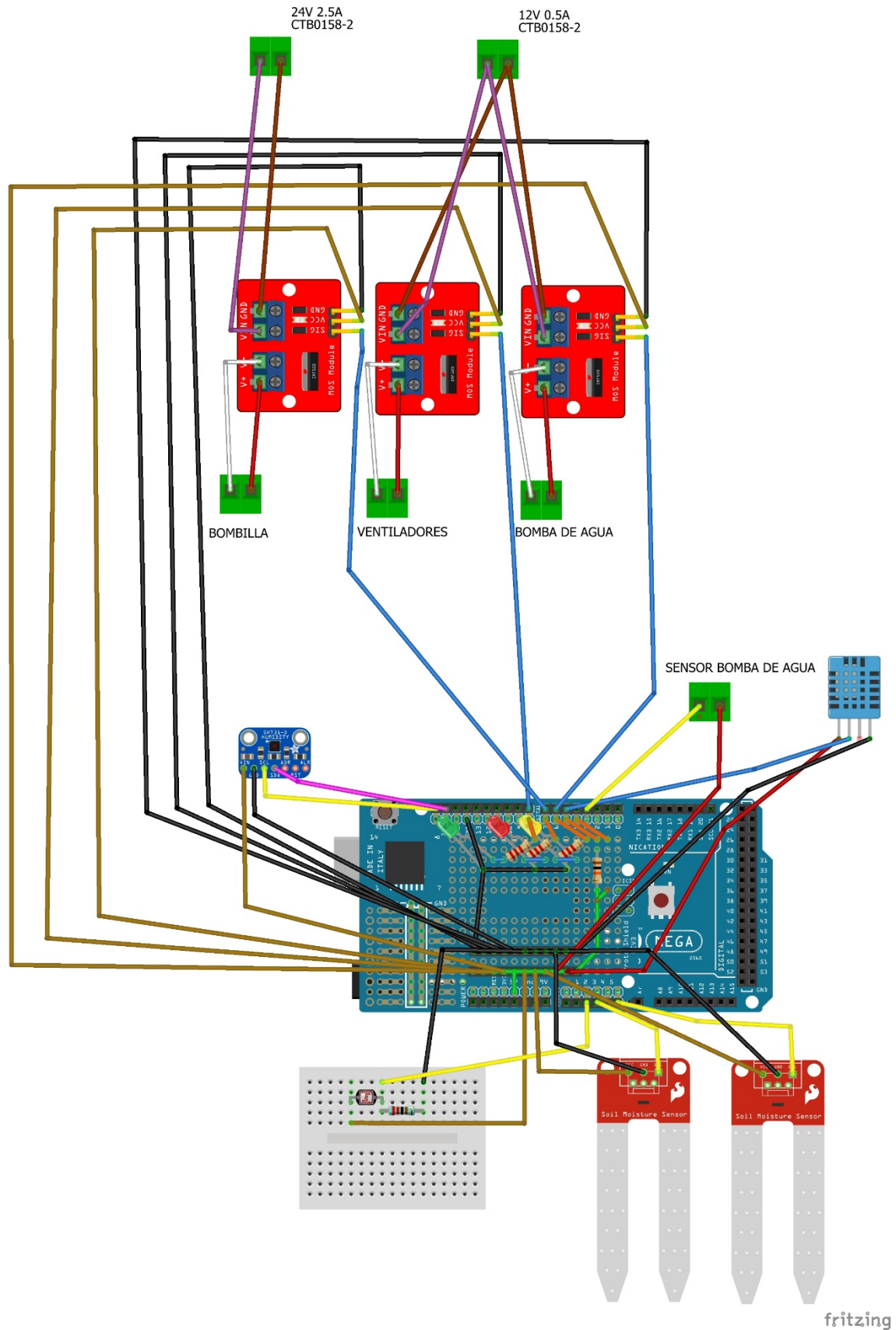


Ilustración 39. Esquema Eléctrico



## 2.2 Desarrollo Software

Este apartado engloba la parte del software, tanto el software utilizado como los flujogramas de las funciones (más importantes) empleadas. También se explicarán algunas funciones propias de los lenguajes utilizados. Para este proyecto se ha empleado en primera instancia Arduino, para el control y gestión de las variables, y en segundo lugar Java, para la interfaz gráfica y la interacción del usuario con los actuadores y sensores.

### 2.2.1 Arduino

Es un software libre, con una comunidad muy extensa, y con un lenguaje fácil e intuitivo. Dispone librerías para la lectura, escritura y gestión de sensores, funciones, Shields...

#### 2.2.1.1 Librerías y Funciones

En este proyecto se han empleado las siguientes librerías:

- **Adafruit\_SHT31.h** → Empleada para la lectura del sensor de temperatura (interior) SHT31. Necesita la declaración del sensor (sht31), así como su registro (0x44), para más tarde realizar la lectura mediante la función *"sht31.readTemperature()"* (esta función nos devuelve una variable de tipo Float).
- **Wire.h** → Empleada para la gestión del protocolo I2C. Con esta librería, entre otros, podemos imponer una frecuencia de lectura del protocolo I2C con la función *"Wire.setClock(frecuencia)"*.
- **DHT.h** → Implementada para la lectura del sensor de temperatura (exterior) DHT11. Necesita la declaración del tipo de sensor y en qué pin se ubica su lectura (dht1). A continuación, se procedería a la lectura con la función *"dht1.readTemperature()"* (esta función nos devuelve una variable de tipo Int).
- **PID\_v1.h** → Esta librería es la encargada del cálculo y realización del PID de nuestro sistema. Haciendo uso de su función *"PID (input, output, setpoint, kp, ki, kd, DIRECT)"*, siendo input (variable controlada), setpoint

(consigna),  $k_p$ ,  $k_i$ ,  $k_d$ , variables de entrada y output (variable que tenemos que implementar en nuestro PWM), una variable de salida. Además, es necesario usar la función *“pid.SetMode (AUTOMATIC)”* para que nuestro sistema sea dinámico y no estanco. Y por último para poder realizar los cálculos del PID, es necesaria la función *“pid.Compute()”*.

También debemos destacar algunas de las funciones empleadas en este proyecto, ya que son propias del lenguaje de Arduino (no necesitan librería):

- **Serial.begin ()** → Función empleada para definir la velocidad de la comunicación serie.
- **PinMode ()** → Nos permite definir la naturaleza del pin digital, entrada o salida.
- **analogRead ()** → Función empleada para la lectura de los pines analógicos.
- **analogWrite ()** → Función empleada para la escritura de los pines digitales pero que funcionan con PWM.
- **digitalRead ()** → Función utilizada para la lectura de los pines digitales de entrada.
- **digitalWrite ()** → Nos permite escribir un valor alto o bajo en los pines digitales de salida.
- **Serial.println ()** → Nos permite enviar datos por el puerto Serie, los datos se envían como una cadena de caracteres (String).
- **Serial.available ()** → Función que se encarga de reconocer si hay datos esperando a ser leídos en el puerto serie.
- **Serial.readStringUntil ()** → Esta función nos permite diferenciar mensajes dentro del puerto serie, por medio de un indicador.
- **Length ()** → Función que nos indica la longitud de una variable.
- **toInt ()** → Nos permite convertir una cadena de caracteres de tipo String a un número entero.

## 2.2.1.2 Diagramas de Flujo

En este apartado se representan los flujogramas tanto general, como de los distintos modos empleados.

En esta figura representa el diagrama de flujo del proceso general que sigue Arduino.

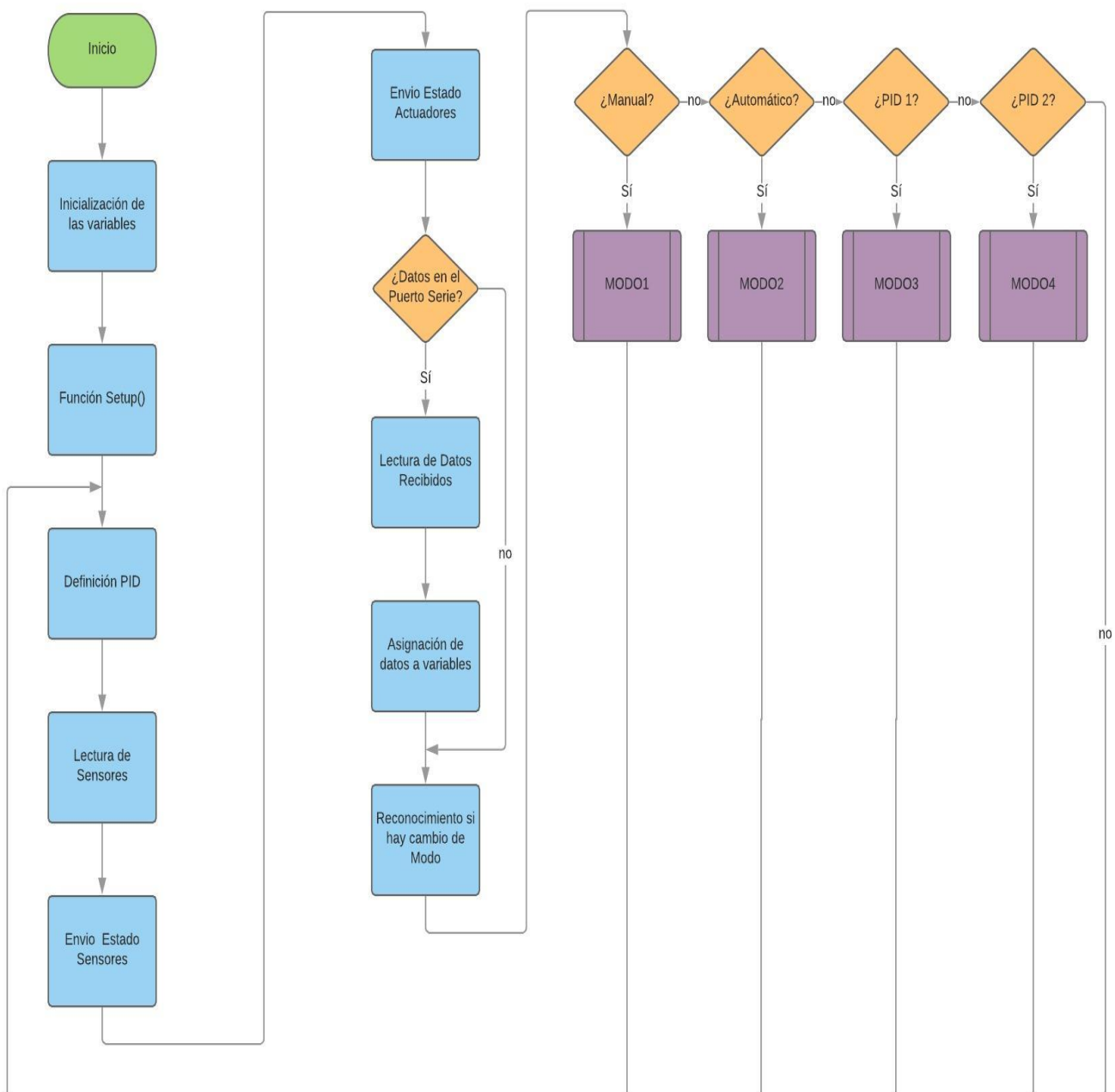


Ilustración 40. Diagrama de Flujo Arduino General

Como hemos visto en la figura anterior, tenemos 4 modos de operación: manual, automático, PID1 y PID2. Están representados como macroetapas. En las siguientes cuatro figuras, se muestra cómo funcionan estos cuatro modos de operación por separado.

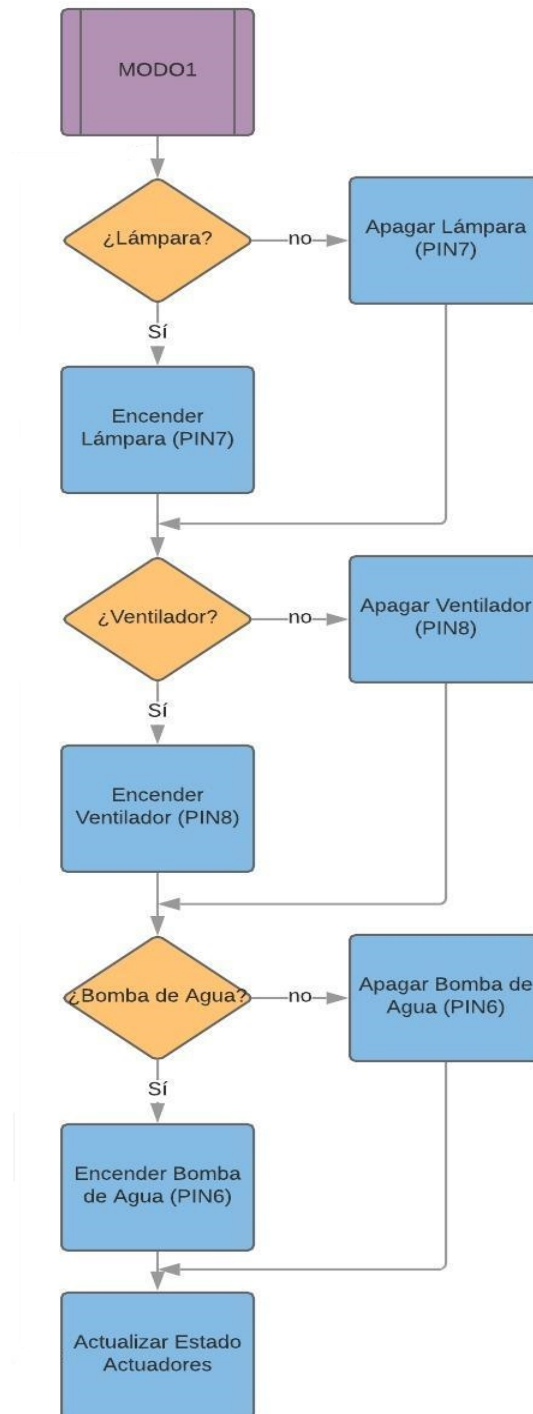


Ilustración 41. Modo Manual Arduino

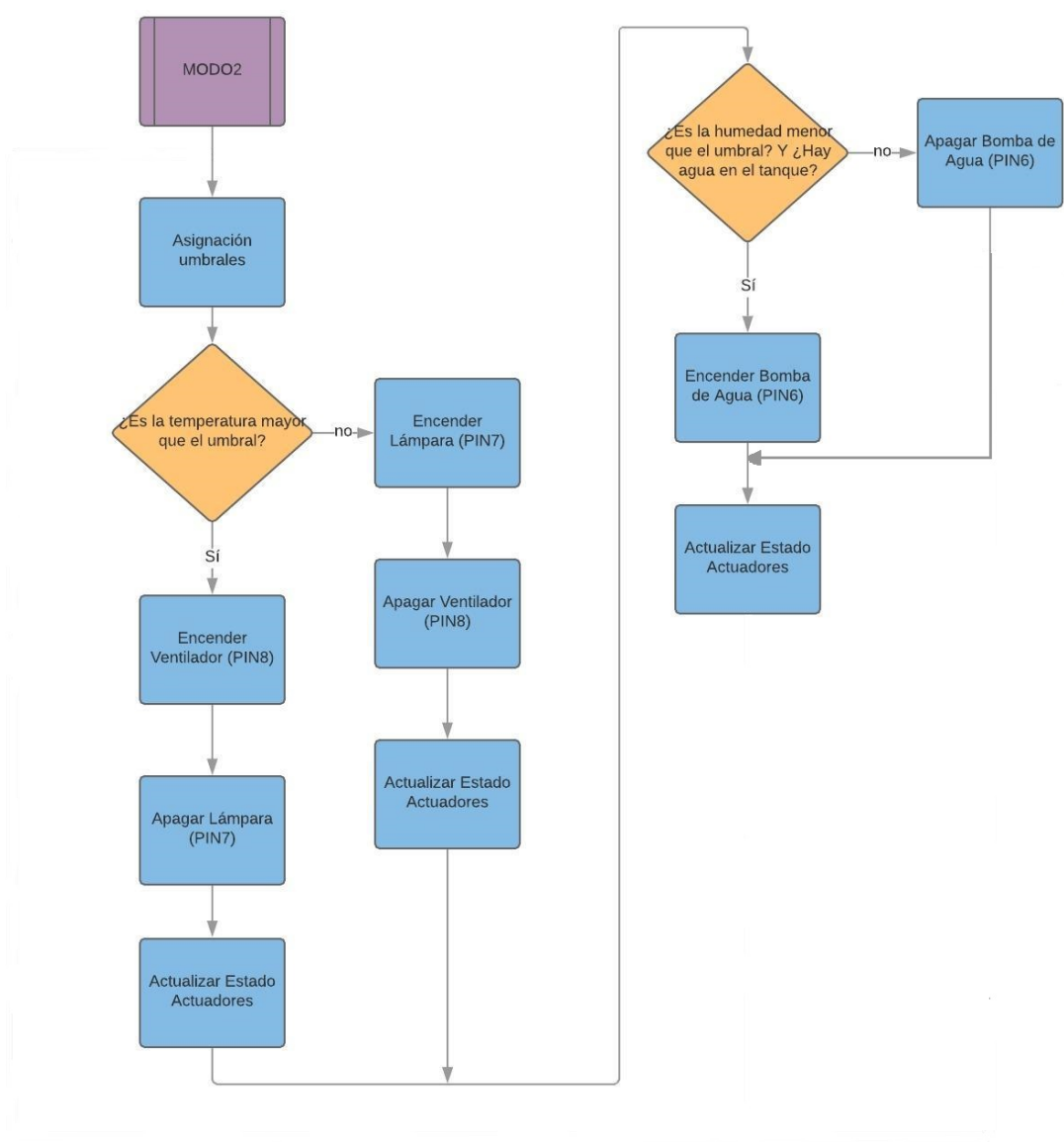
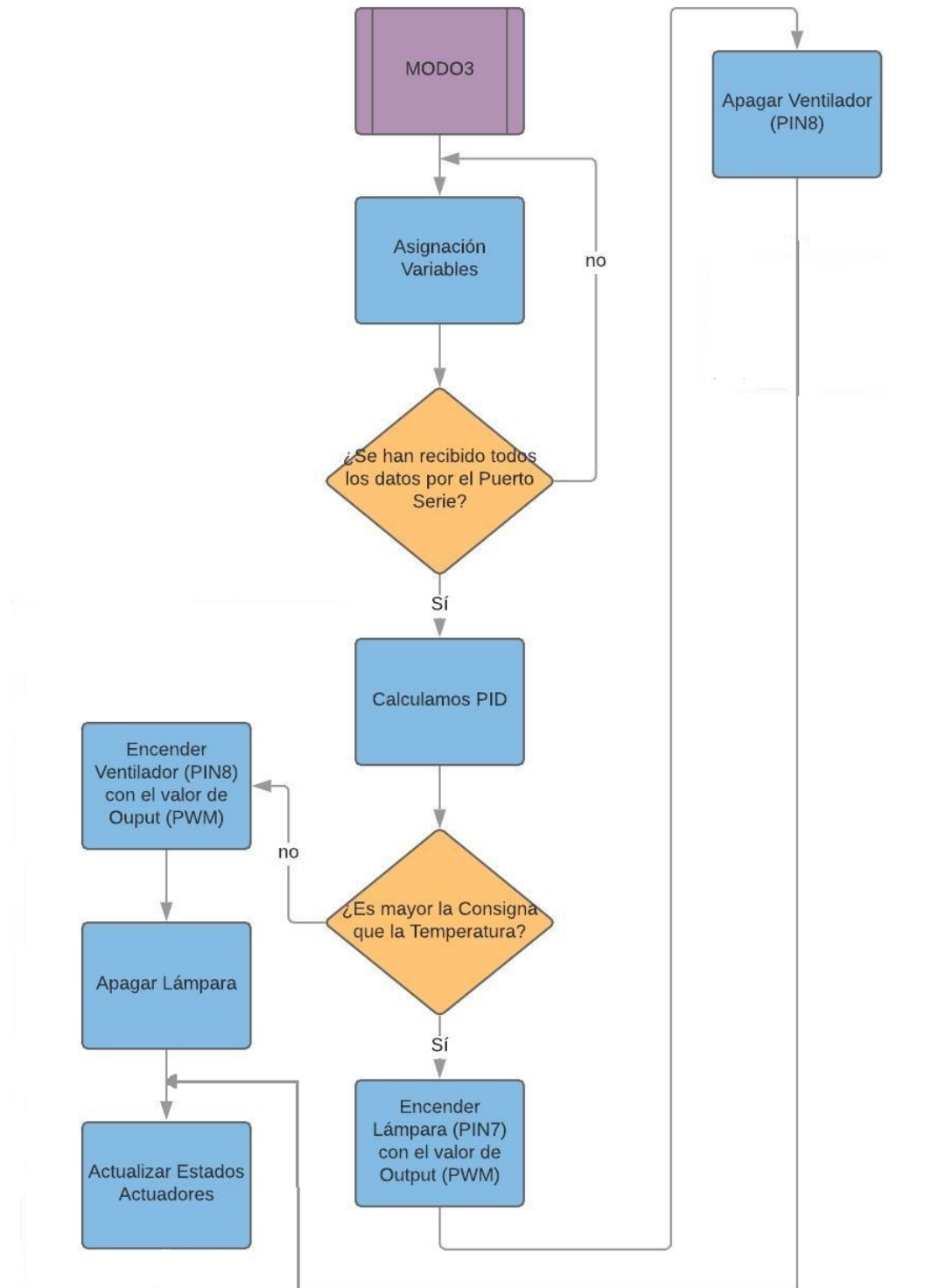


Ilustración 42. Modo Automático Arduino

*Ilustración 43. Modo PID1 Arduino*

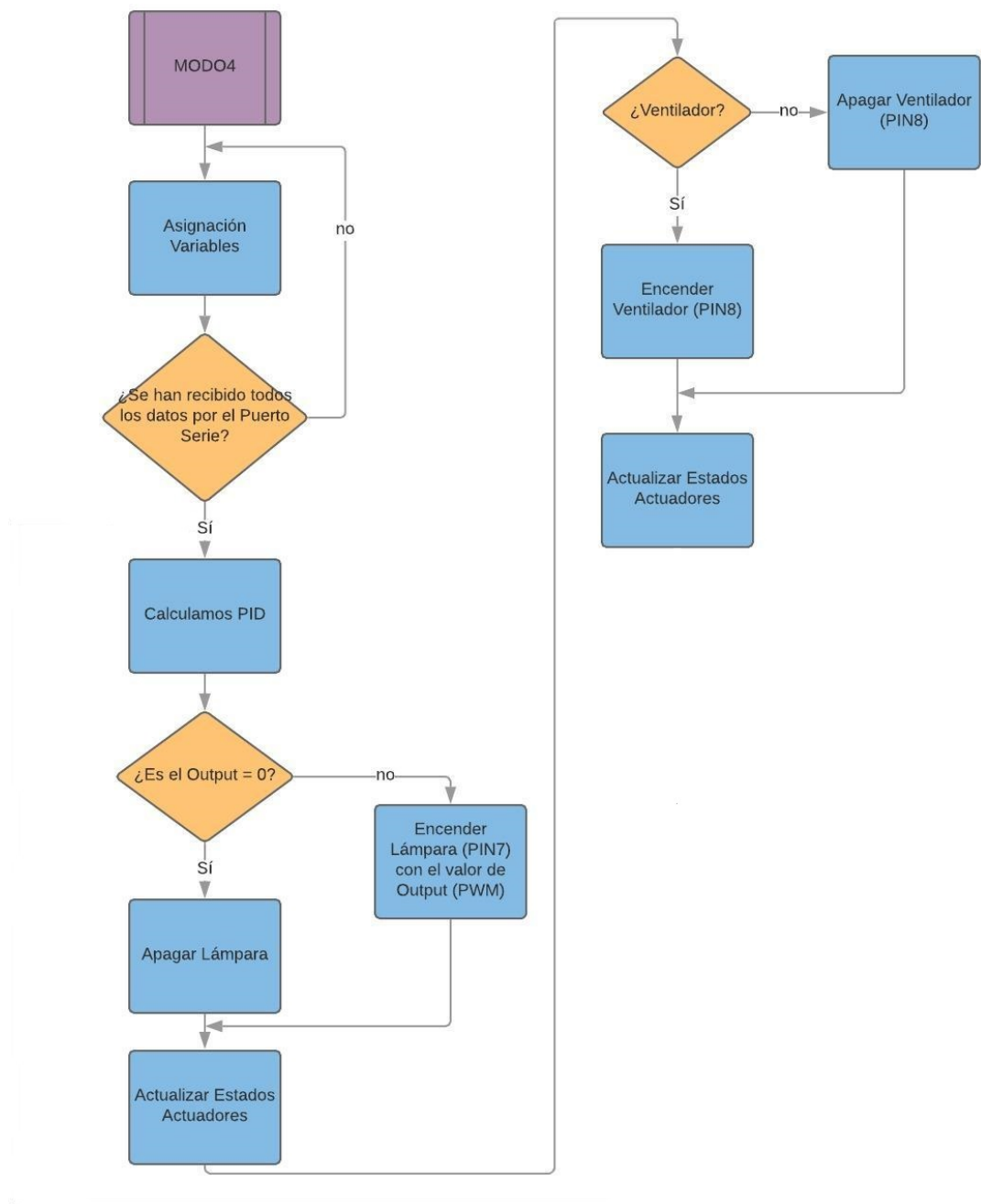


Ilustración 44. Modo PID2 Arduino

### 2.2.2 JAVA

Java es un lenguaje de programación enfocado en la programación de objetos. La idea de Java es que puedan realizarse programas con la posibilidad de ejecutarse en cualquier contexto y en cualquier entorno, siendo la portabilidad una de sus principales características. De desarrollo libre, cualquier usuario puede acceder a su software de programación y crear un programa. En nuestro caso hemos empleado NetBeans, ya que en cuanto a librerías, funciones y estructura de programación es muy intuitivo.

#### 2.2.2.1 Librerías y Clases

En este caso, cuando hablamos de librerías también estamos hablando de clases, esto significa que una librería puede englobar varias clases, por tanto, hay que especificar la librería, así como la clase que se está empleando. En este proyecto se han empleado las siguientes librerías, así como clases:

- **PanamaHitek** → librería necesaria para la comunicación serie con Arduino. Hemos empleado tres clases dentro de esta librería, “*PanamaHitek\_Arduino*” utilizada para la definición de la comunicación en serie, es decir, nos crea un objeto que es el encargado de recibir los datos por el puerto serie. “*PanamaHitek\_MultiMessage*” encargada de diferenciar los datos recibidos en el puerto serie, definiendo previamente el número de datos que se van a recibir. Y por último “*ArduinoException*” empleada para las excepciones que puede crear el envío de datos por el puerto serie.
- **Awt** → Librería que nos permite crear eventos, en nuestro caso, “*ActionEvent*”, empleado para generar interrupciones en nuestro código cuando, por ejemplo, pulsamos un botón. También hemos utilizado “*ActionListener*”, evento necesario para poder recibir datos por el puerto serie.
- **Io** → Empleada para escribir y guardar datos, en nuestro caso, en un formato de texto. Las clases utilizadas de esta librería han sido “*File*”, nos crea el archivo dónde se va a guardar los datos. “*FileWriter*” encargada



de escribir nuestros datos en nuestro archivo y de guardarlos. Y, por último, “*IOException*” empleada para las excepciones que esta librería nos pueda generar.

- **Math** → Librería empleada para realizar operaciones matemáticas. En nuestro caso la clase utilizada ha sido “*RoundingMode*”, necesaria para declarar el tipo de redondeo que queremos aplicar al número.
- **Text** → Empleada para dar un formato específico al texto, entre sus clases, hemos utilizado “*SimpleDateFormat*”, que unida a “*DateFormat*”, nos dan un formato simple a nuestra fecha.
- **Swing** → Encargada de las interfaces de usuario, botones, etiquetas, sliders, timers... En este proyecto se han empleado, “*JFileChooser*”, que nos permite la lectura de archivos o apertura de directorios. “*JFrame*” para crear frames en la pantalla, “*JOptionPane*” es la encargada de los mensajes por pantalla y, por último, “*Timer*” que nos gestiona y crea timers.
- **SerialPort** → Librería que nos permite crear eventos en el puerto Serie. Las clases empleadas en este proyecto han sido “*SerialPortEvent*”, nos crea un evento cuando queramos enviar algún dato por el puerto serie. “*SerialPortEventListener*”, cada vez que reciba un dato por el puerto serie, nos crea un evento para recibirlo. “*SerialPortException*” nos permite gestionar las excepciones de esta librería.
- **Jfree** → Esta librería se encarga de la creación y representación de las gráficas de las variables monitorizadas. Entre otras clases, hemos empleado “*ChartFactory*”, “*JFreeChart*”, “*ChartPanel*”, “*PlotOrientation*”, “*XYSeries*” y “*XYSeriesCollection*”.

#### 2.2.2.2 Diagramas de Flujo

En este apartado se representan los flujogramas de nuestra programación en Java. En primer lugar, se expone un diagrama de flujo del bucle general, para más tarde explicar las funciones o interrupciones más importantes. Cabe destacar, que este programa sólo se encarga de la interacción con el usuario, es

decir, la gestión de los modos, la hace Arduino. La función de Java es avisar a Arduino del modo y de los parámetros que se van a utilizar, así como de la creación y gestión de la captura de datos y su monitorización.

La siguiente figura nos muestra el diagrama de flujo del proceso general que sigue.

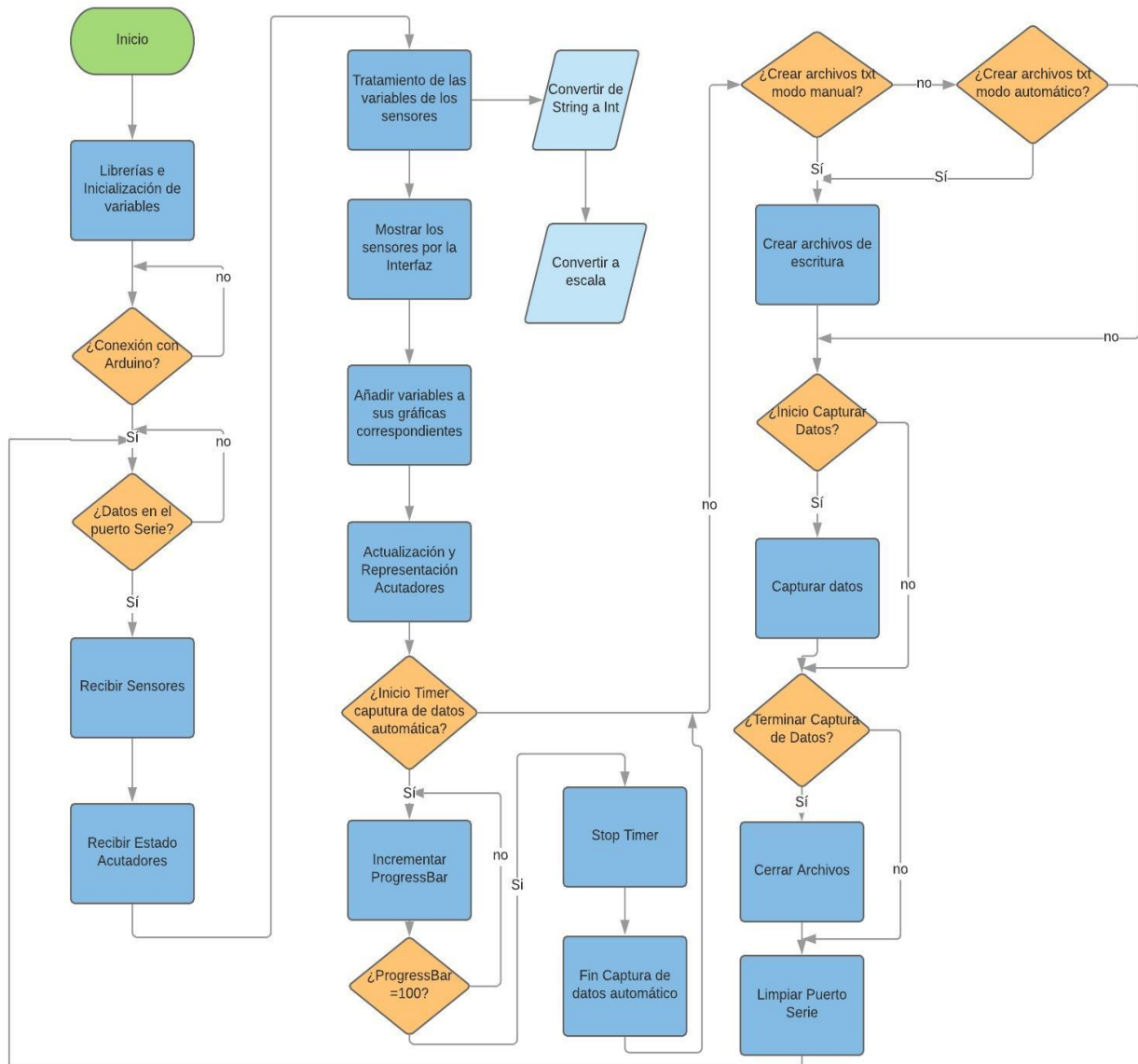
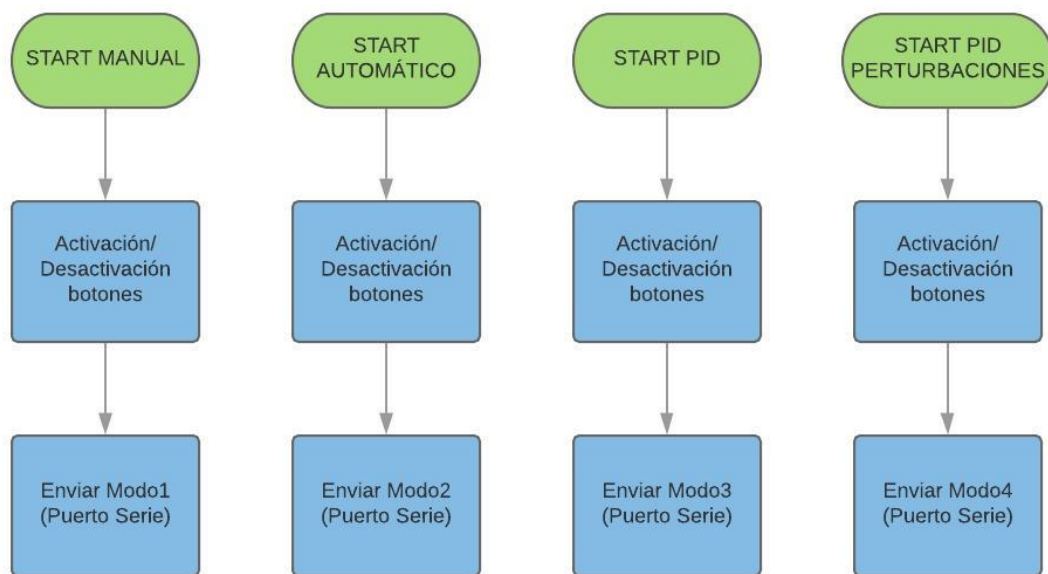


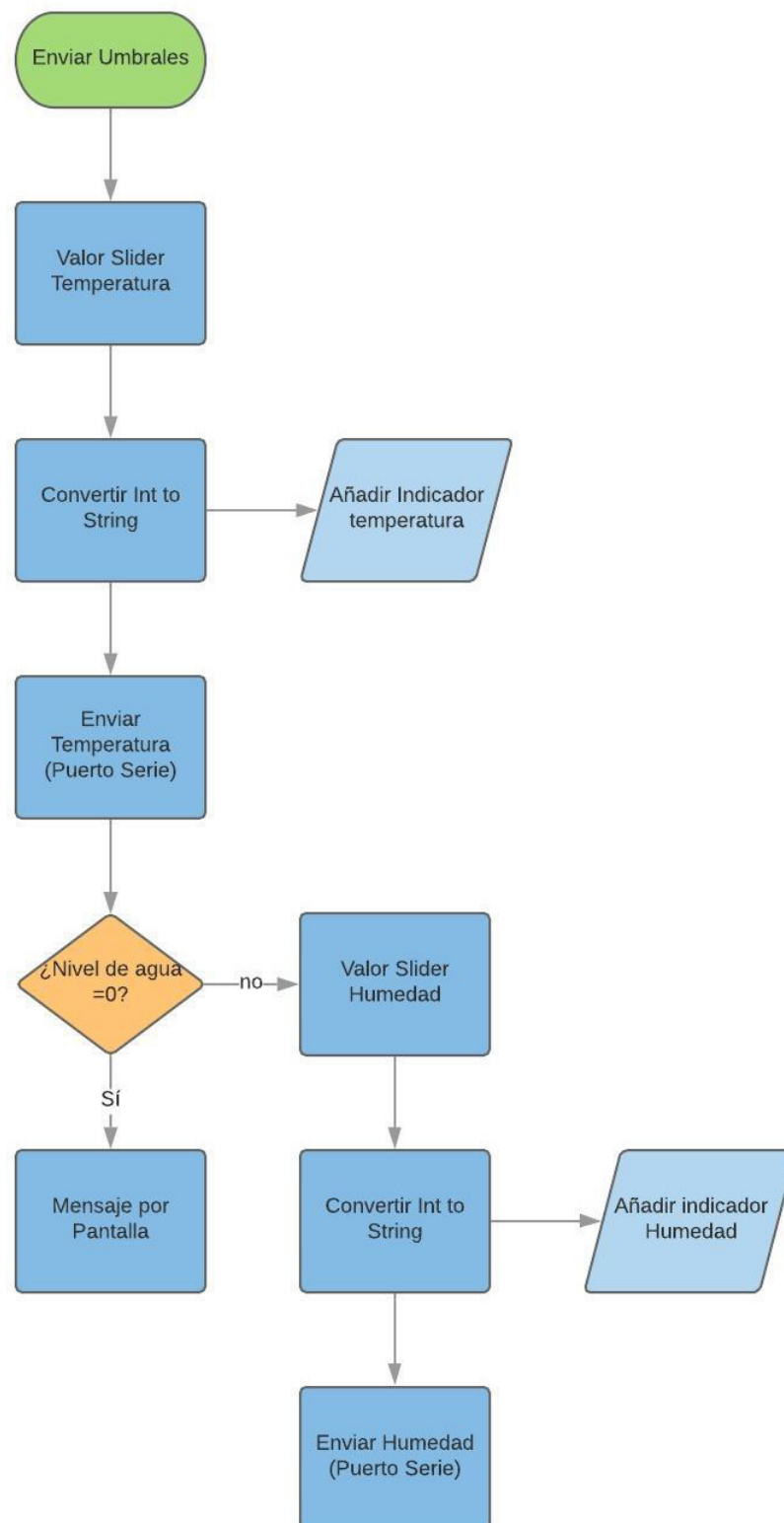
Ilustración 45. Diagrama de Flujo Java General

En cuanto a los botones, actúan como interrupciones, es decir, crean un evento cuando se pulsan. Este evento ejecuta un código, interrumpiendo el flujo normal del programa, y en la mayoría de casos o se envían datos por el puerto Serie (caso de los botones de modo manual), o se activan variables banderas (caso de la captura de datos), que nos advierten en el programa principal que se han pulsado y por tanto se ejecutará un código u otro.



*Ilustración 46. Diagramas de Flujo de los diferentes Modos*

En las siguientes figuras se representan los botones o eventos más complejos o que tienen más líneas de código. La primera figura representa el botón de “Enviar Umbrales” del modo Automático. La segunda figura define el evento que produce el botón “Start” en la captura de datos Automática. Y, para terminar, la última figura representa el botón “Enviar” del modo PID (para ambos, el funcionamiento es el mismo).

*Ilustración 47. Botón Enviar Umbrales*

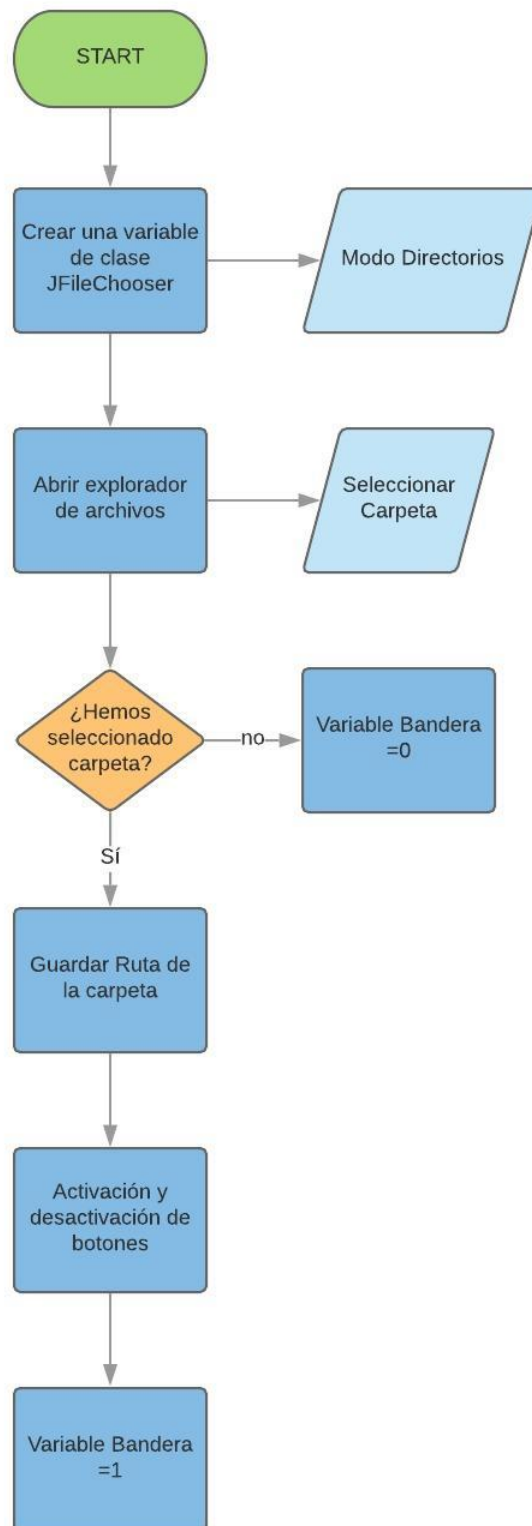


Ilustración 48. Botón Start Captura de datos Automático

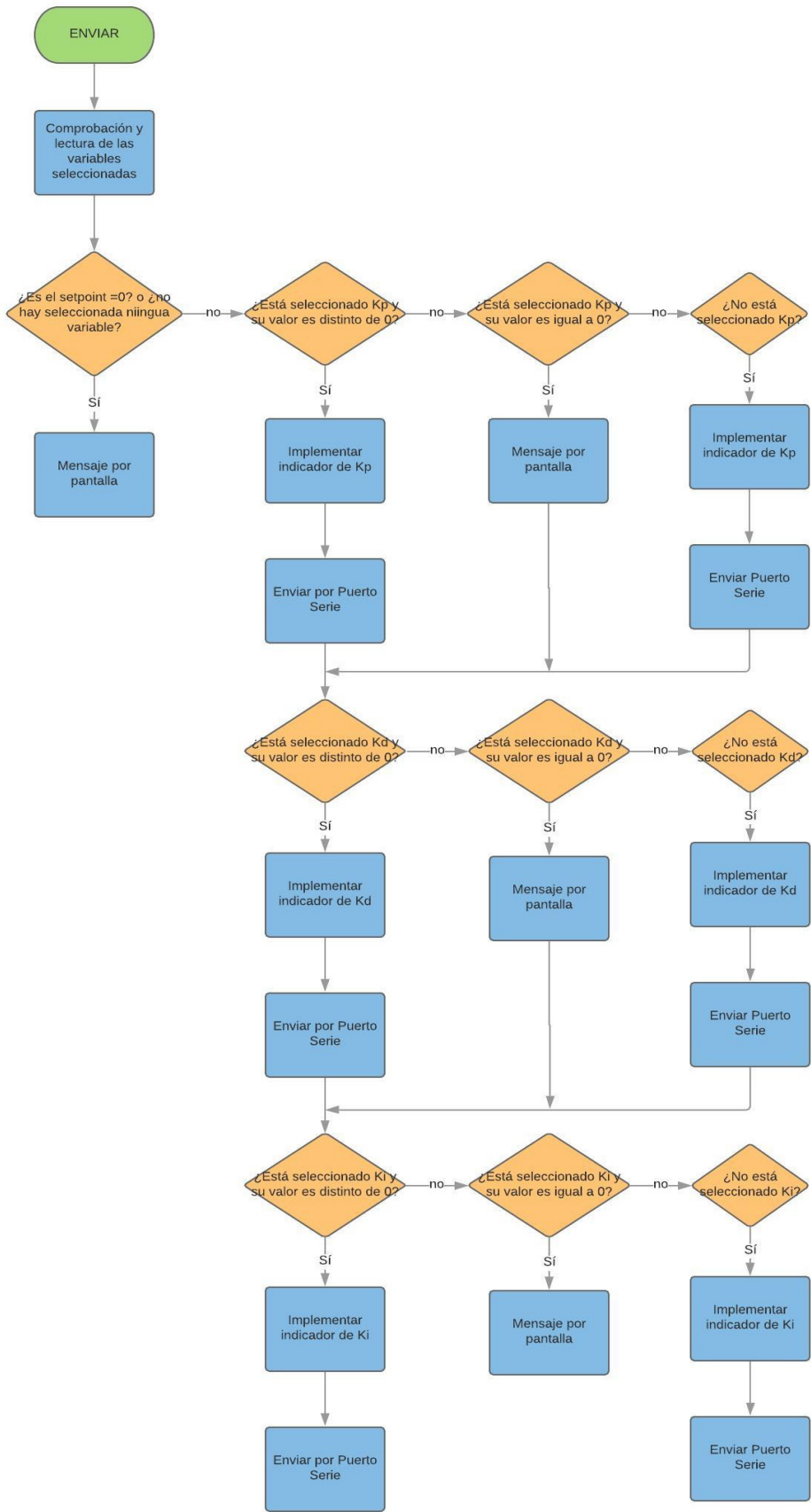


Ilustración 49. Botón Enviar PID

### 2.2.2.3 Interfaz Gráfica

En este apartado se aborda el tema de la interfaz gráfica. Esta interfaz ha sido creada en NetBeans con Java Swing. La característica más importante de la interfaz es su facilidad de manejo, es decir, es una interfaz atractiva para el usuario y no compleja, a la vez que intuitiva. Posee siete bloques, de los cuales cuatro están interconectados entre ellos (los diferentes modos), haciendo imposible utilizar dos o más bloques a la vez.

El primer bloque, bloque de la monitorización, es el relacionado con los sensores. Nos permite visualizar los datos captados por los sensores en tiempo real, además de ver su cambio en el tiempo por medio de gráficas.

El segundo bloque, nos permite visualizar el estado de los actuadores, por medio de dos sencillas marcas. Siendo rojo apagado, azul encendido.

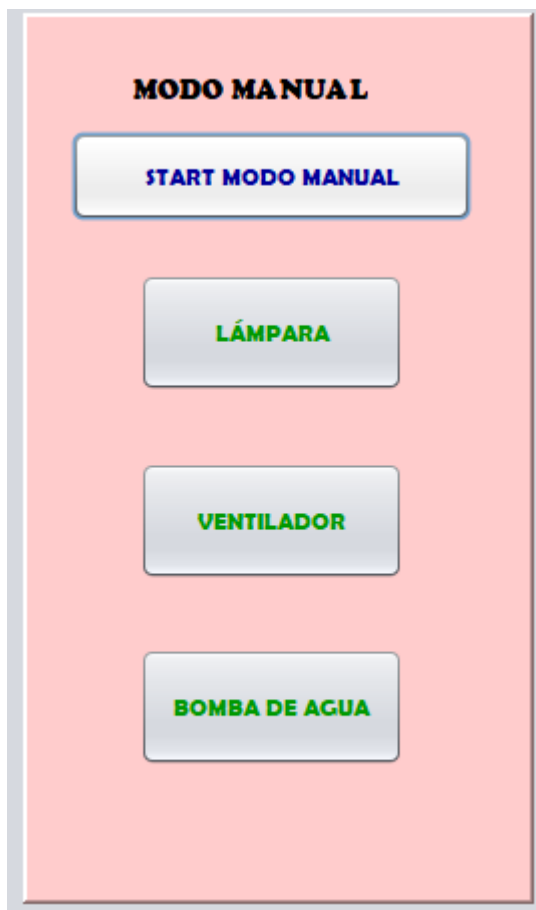


Ilustración 51. Estado Sensores



Ilustración 50. Estado Actuadores

Los bloques tres, cuatro y cinco, se corresponden con los modos manual, automático y PID. El modo manual consta de tres botones de acción de doble efecto, encendido/apagado. El modo automático consta de dos sliders, que nos permiten situar la temperatura interior y la humedad en un valor, además de un botón para hacer posible el envío de estos valores. Por último, el modo PID, que en este caso tenemos dos modos, nos permiten seleccionar las variables que conforman el PID, así como su valor, también nos permite seleccionar un valor de consigna. El modo PID con perturbaciones, además nos permite actuar directamente sobre el ventilador (apagado/encendido). Las siguientes figuras nos muestran la interfaz de los diferentes modos.

*Ilustración 52. Modo Manual**Ilustración 53. Modo Automático*



**CONTROLADOR PID**

**START MODO PID**

**VENTILADOR Y LÁMPARA**

Temperatura Interior (Set Point)

☐ Kp  °C

☐ Kd

☐ Ki

**START PID PERTURBACIONES**

**PID LÁMPARA CON PERTURBACIONES**

Temperatura Interior (Set Point)

☐ Kp  °C

☐ Kd

☐ Ki

Ilustración 54. Modos PID

Los últimos dos bloques son los relacionados con la captura de datos, haciendo posible la captura de datos manual o automática (nunca las dos a la vez). En la captura manual da la opción de empezar y acabar cuando el usuario disponga. En la automática le da al usuario la opción de programar durante cuantos minutos quiere que se capture datos.

**CAPTURAR DATOS MANUAL**

**CAPTURAR DATOS AUTOMÁTICO**

**NÚMERO DE MINUTOS**

Ilustración 55. Capturas de Datos

En la siguiente figura, se muestra como es la versión final de la interfaz gráfica y como se compone de todas las partes mostradas anteriormente

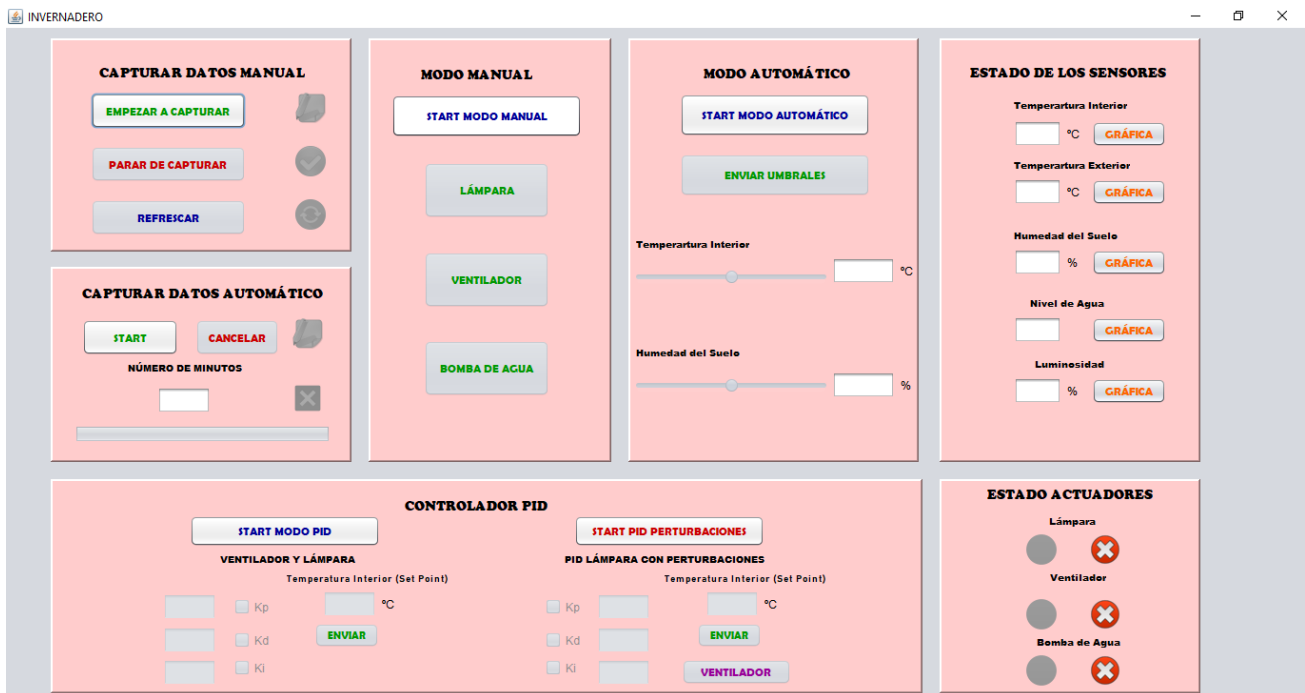


Ilustración 56. Interfaz Gráfica

### 3. RESULTADOS

Una vez llevados a cabo el diseño y la implementación del sistema, se procedieron a realizar una serie de ensayos para comprobar su funcionamiento.

En primer lugar, que la comunicación establecida fuera correcta y bidireccional, que fuéramos capaces de ver en nuestra interfaz los registros de los sensores obtenidos por Arduino y que la acción sobre los actuadores desde la interfaz gráfica se viera reflejado en nuestro sistema (en vivo).

A continuación, comprobamos que la captura de datos también funcionaba acorde a lo previsto, guardando las variables en archivos separados, además de su fecha, con una precisión de milisegundos, de cada uno de los registros guardados.

Y, por último, se realizaron una serie de ensayos, de objeto de estudio, para comprobar el correcto funcionamiento de nuestro sistema PID, obteniendo unos resultados bastante buenos.

### 3.1 Solución Final

En este apartado se muestra el progreso del invernadero inteligente, desde su comienzo hasta su solución final.

Al principio, el sistema constaba de dos actuadores, una bombilla y un ventilador, cuatro sensores, dos sensores de temperatura tipo DHT11, un sensor de humedad del suelo y un sensor de luminosidad tipo LDR. El control de los actuadores se realizaba mediante dos relés electrónicos (tipo Todo/Nada). Y todo estaba interconectado en formato prototipo, es decir, con una placa de desarrollo ProtoBoard, como se muestra en la siguiente figura:

*Ilustración 57. Primer Modelo Invernadero*

En una segunda fase, se implementaron nuevos sensores, otro sensor más de humedad del suelo (para conseguir una medición en dos puntos distintos y hacer la media), un sensor de nivel de agua de tipo LS02 y un actuador, en este

caso, una bomba de agua, así como su sistema de riego por goteo. Las siguientes ilustraciones muestran la implementación de estas mejoras.



*Ilustración 59. Sensores de Humedad y Sistema de Riego*



*Ilustración 58. Bomba de Agua y Sensor de Nivel*

En la tercera fase del proyecto, se realizó el cambio de los relés de potencia por Mosfets de potencia. Esto conllevaba el acoplamiento de una fuente de alimentación para poder alimentar el Mosfet de la bombilla, y además el implantar una bombilla de 60W que trabajase en continua, es decir, que consiguiera encenderse con 24V dc. Además, se llevó acabo el diseño y la fabricación de circuito impreso o PCB. Consiguiendo eliminar cableado y la placa de prototipado. Las siguientes ilustraciones reflejan la implementación de estos dispositivos, así como una versión cerca de la versión final.

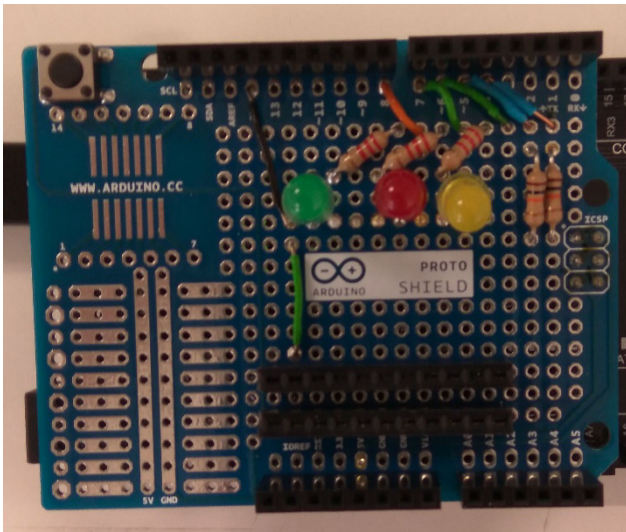


Ilustración 62. PCB Final

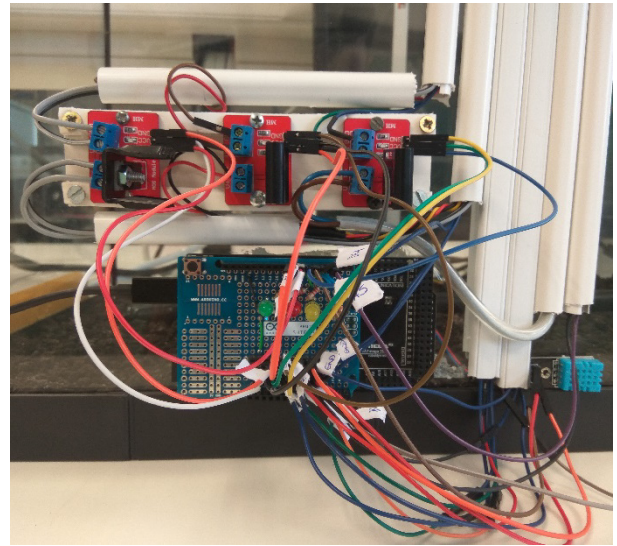


Ilustración 60. Grupo Mosfet Final

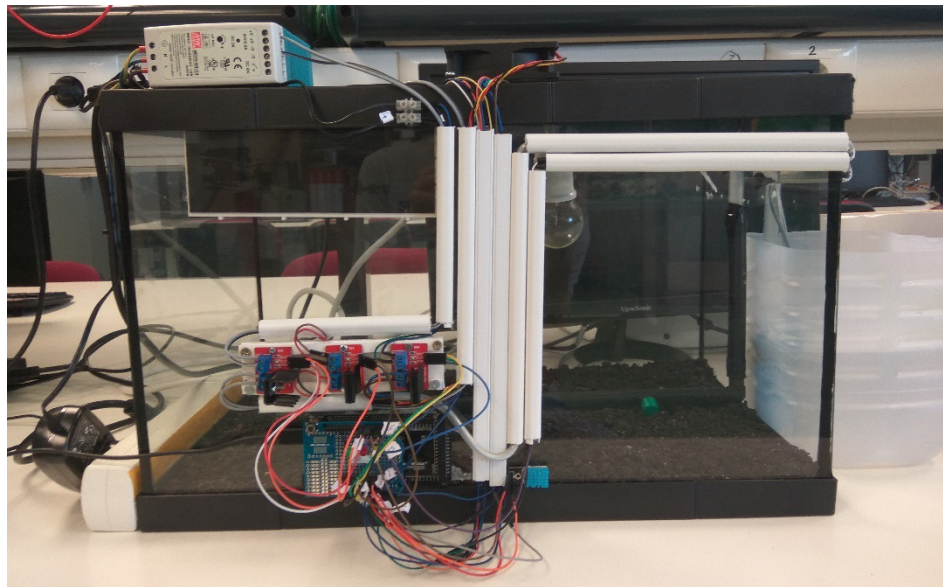
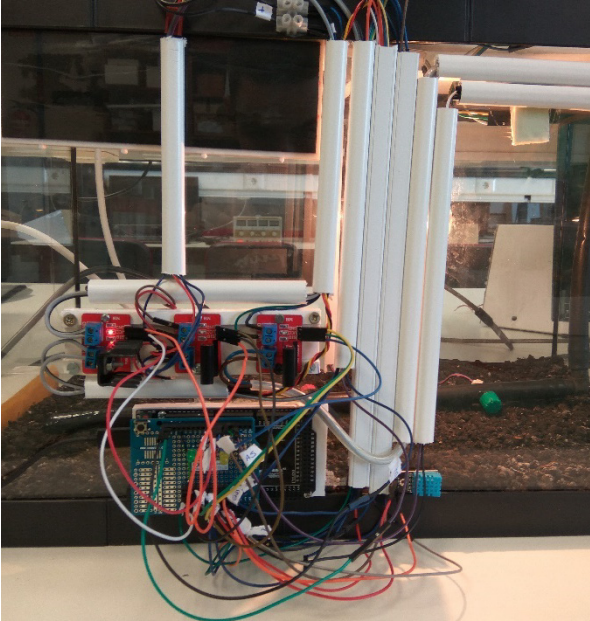


Ilustración 61. Versión Fase Tres

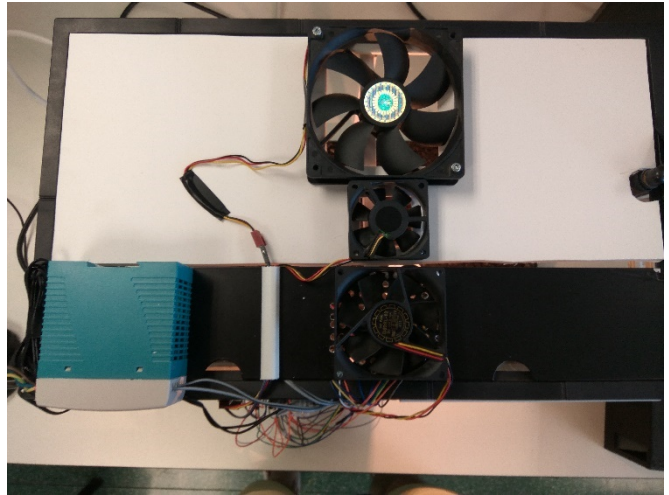
En la fase final, se le realizaron mejoras tanto de protección como de dispositivos. En primer lugar, se le hizo una nueva tapa, a la que se le incluyeron dos ventiladores, que contando con el que había, se encargan de crear un flujo de aire (el de mayor capacidad inyecta aire, los dos de menor capacidad expulsan aire), haciendo posible una notoria influencia de las perturbaciones en nuestro sistema PID, esto se muestra en la ilustración.... También se implementó un nuevo sensor de temperatura (SHT31), dándonos la capacidad de crear gráficas próximas a un PID. En cuanto a la mejora para una mejor protección del



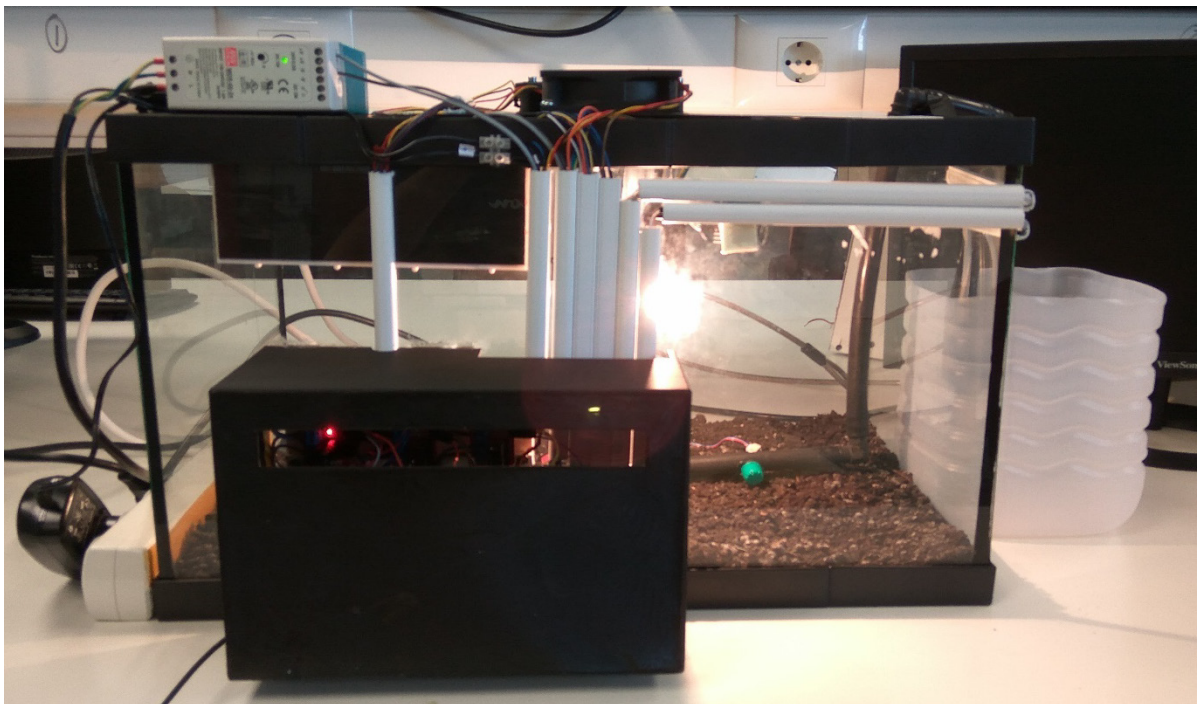
sistema, se diseñó una carcasa, y se imprimió con una impresora 3D, para proteger al sistema de posibles rupturas, tirones... Obteniendo una versión final de nuestro sistema, como se muestra en la figura...



*Ilustración 64. Conjunto Mosfets,  
PCB, Sensor SHT31*



*Ilustración 63. Ventiladores  
Finales*



*Ilustración 65. Invernadero Inteligente Final*

### 3.2 Ensayos

Como se menciona anteriormente, este proyecto tiene como objetivo la implantación de un sistema PID, por tanto, es necesario hacer ensayos y obtener resultados.

En este apartado se centra en esa sección, en el PID. Se han realizado dos diferentes tipos de ensayos, en los cuales se han recogido capturas de la consigna, de la gráfica del actuador y la gráfica de la variable controlada, que, en nuestro caso, es la temperatura interior del invernadero.

#### 3.2.1 Ensayo Tipo 1

En este tipo de ensayo, lo que se procede a realizar es emplear los mismos parámetros del PID, pero variando nuestro Set Point, es decir,  $K_p$ ,  $K_d$  y  $K_i$  permanecerán iguales, mientras que nuestra consigna variará. Más adelante se activará la perturbación, en nuestro caso, encendido de ventiladores (que expulsarán el calor). Se realizarán tres ensayos. Las constantes de nuestro sistema en los tres ensayos serán las mostradas en la siguiente tabla.

<b><math>K_p</math></b>	3
<b><math>K_d</math></b>	10
<b><math>K_i</math></b>	1
<b>Temperatura Inicial</b>	25.5°C

Tabla 8. Constantes Ensayo Tipo 1

El primer ensayo nuestra Consigna, se establecerá en  $T=30.5^\circ\text{C}$ . Obteniendo las siguientes gráficas.

Ilustración 66. Ensayo1  $T=30.5^\circ\text{C}$

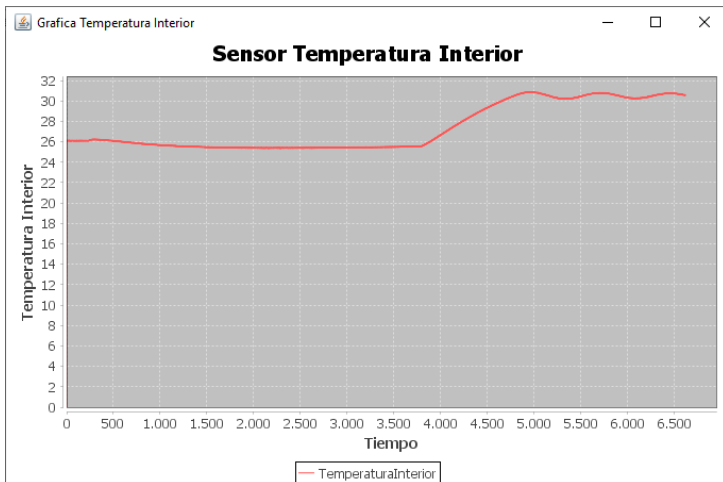


Ilustración 67. Ensayo1 Temperatura

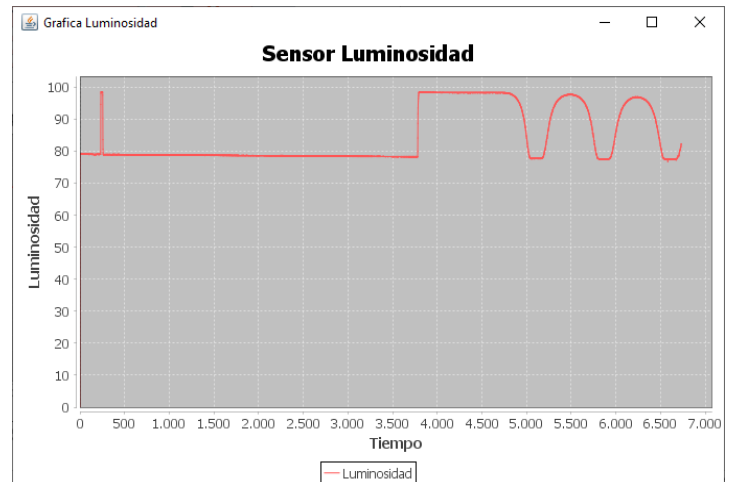


Ilustración 68. Ensayo1 Actuador

Como podemos observar el sistema es estable, y ha empleado poco tiempo en alcanzar nuestra consigna. El actuador, por otro lado, se comporta de forma oscilatoria, lo que nos indica que nuestro sistema no es muy estable. Comprobamos ahora el comportamiento del sistema frente a la influencia de una perturbación.

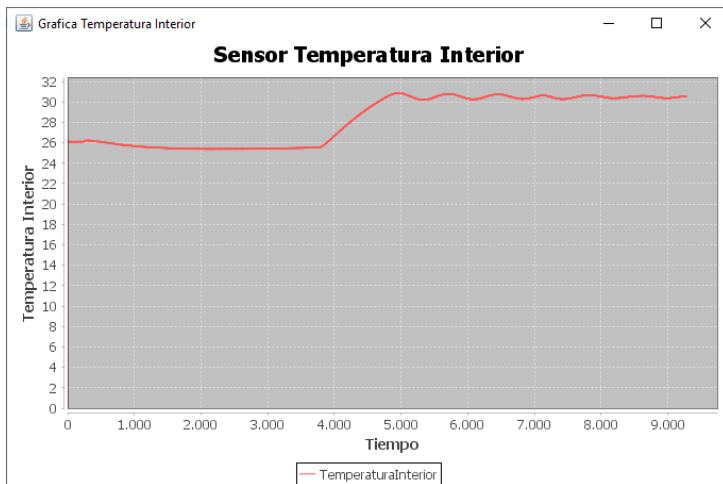


Ilustración 70. Ensayo1 Temperatura con Perturbación

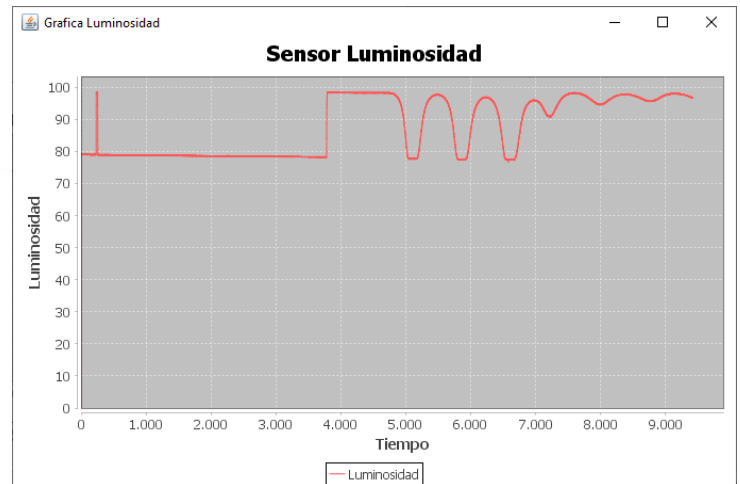


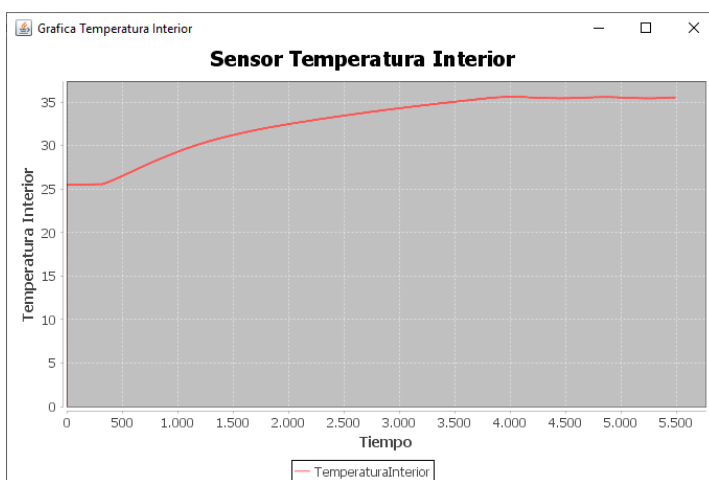
Ilustración 69. Ensayo1 Actuador con Perturbación

Nuestro sistema se ha comportado de una manera eficiente, ya que tanto el actuador, como la variable controlada, han sido capaces de remontar y conseguir continuar en nuestra consigna.

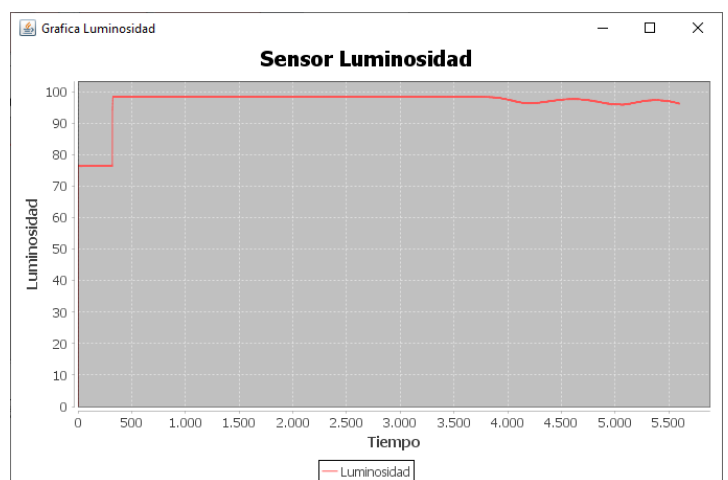


El segundo ensayo nuestra Consigna, se establecerá en  $T=35.5^{\circ}\text{C}$ .  
Obteniendo las siguientes gráficas.

*Ilustración 71. Ensayo2  $T=35.5^{\circ}\text{C}$*

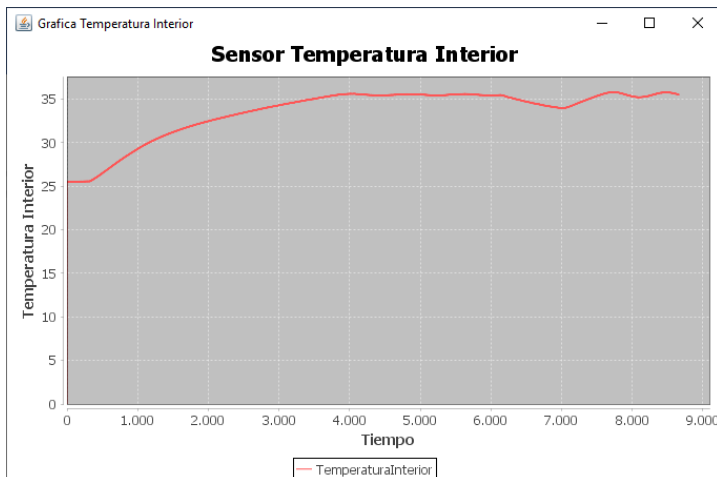


*Ilustración 73. Ensayo2 Temperatura*

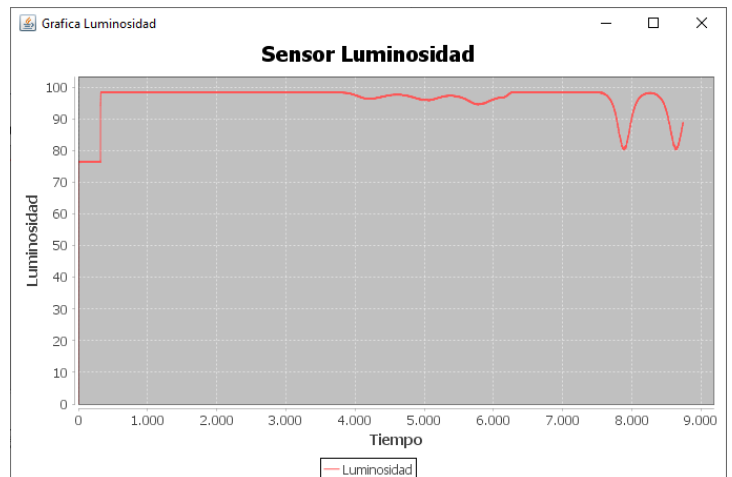


*Ilustración 72. Ensayo2 Actuador*

El sistema, como podemos observar, es estable. Ha alcanzado la temperatura de consigna en un breve periodo de tiempo. Por otro lado, el actuador, oscila, pero no tanto como en el ensayo anterior. Comprobamos ahora su respuesta ante una perturbación.



*Ilustración 75. Ensayo2 Temperatura con Perturbación*



*Ilustración 74. Ensayo2 Actuador con Perturbación*

El sistema ha tenido una respuesta óptima, ya que como se puede observar, en el minuto siete, ha hecho efecto la perturbación, obteniendo un pico de temperatura. Nuestro sistema se ha sobrepuesto a este pico consiguiendo llegar a la temperatura consigna. También podemos ver, como nuestro sistema se vuelve inestable a causa de esta perturbación (tanto en la variable controlada como en el actuador).

En el último ensayo, tenemos de valor de Consigna  $T=38^{\circ}\text{C}$ . Obteniendo las siguientes gráficas.

*Ilustración 76. Ensayo3  $T=38^{\circ}\text{C}$*

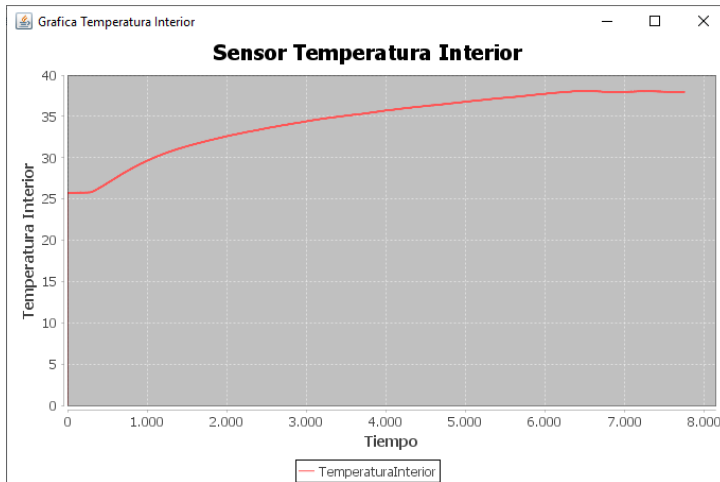


Ilustración 78. Ensayo3 Temperatura

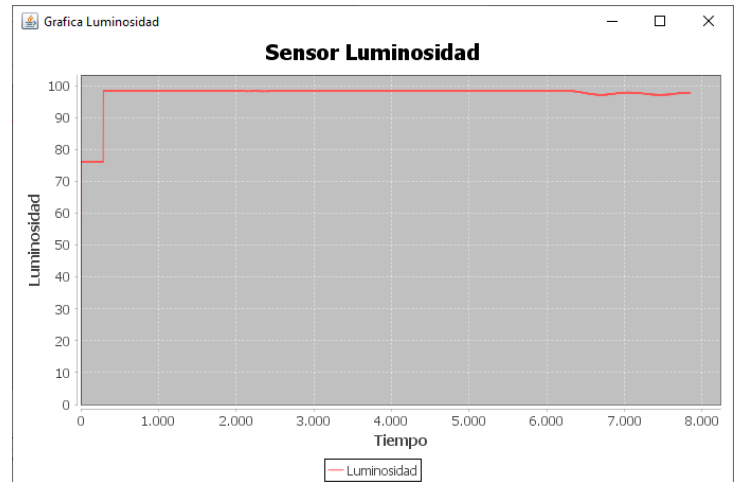
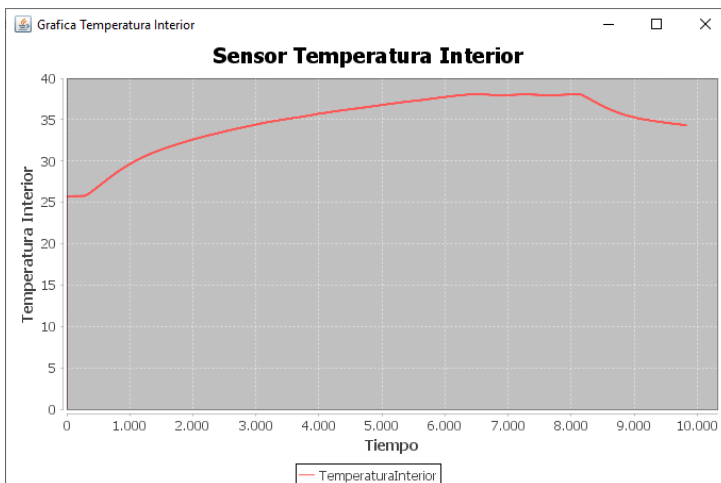
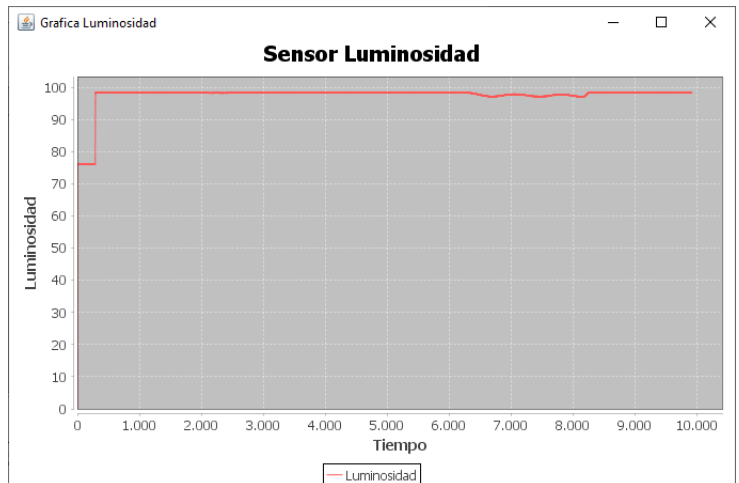


Ilustración 77. Ensayo3 Actuador

El sistema es estable, pero ha requerido mucho tiempo para alcanzar la Consigna. Como podemos observar el actuador está al máximo de su capacidad, esto se debe a que estamos cercanos a los límites de temperatura de nuestro sistema (40°C). Ahora veamos como se comporta el sistema frente a una perturbación.

Ilustración 80. Ensayo3 Temperatura  
con PerturbaciónIlustración 79. Ensayo3 Actuador con  
Perturbación

Al estar en los límites de temperatura, nuestro sistema es incapaz de enaltecer la temperatura, aún teniendo el actuador en su valor máximo.

### 3.2.2 Ensayo Tipo 2

En este tipo de ensayos, se procede a realizar un estudio con los parámetros del PID variantes y el Set Point invariante, es decir, nuestras constantes del sistema  $K_p$ ,  $K_d$  y  $K_i$ , se irán cambiando en cada uno de los ensayos, pero lo único que permanecerá constante es la Consigna. Se realizarán cuatro ensayos. Las temperaturas de inicio y de Set Point se ven reflejadas en la siguiente tabla.

<b>Consigna</b>	35°C
<b>Temperatura Inicial</b>	26°C

Tabla 9. Temperatura Inicial y  
Consigna Ensayo Tipo2

En el primer ensayo, impondremos las constantes reflejadas en la siguiente tabla. Obteniendo las siguientes gráficas.

<b><math>K_p</math></b>	3
<b><math>K_d</math></b>	10
<b><math>K_i</math></b>	1

Tabla 10. Constantes PID Ensayo1  
Tipo2

The screenshot displays the 'CONTROLADOR PID' interface. It is divided into two main sections for different test modes. The left section, 'VENTILADOR Y LÁMPARA', has a 'START MODO PID' button and input fields for 'Temperatura Interior (Set Point)',  $K_p$ ,  $K_d$ , and  $K_i$ , with an 'ENVIAR' button. The right section, 'PID LÁMPARA CON PERTURBACIONES', has a 'START PID PERTURBACIONES' button and similar input fields, but with checked checkboxes for  $K_p$ ,  $K_d$ , and  $K_i$ , and a 'VENTILADOR' button. On the far right, the 'ESTADO ACTUADORES' section shows the status of three actuators: 'Lámpara' (blue circle), 'Ventilador' (grey circle), and 'Bomba de Agua' (red circle with an 'X').

Ilustración 81. Constantes Ensayo1 Tipo2

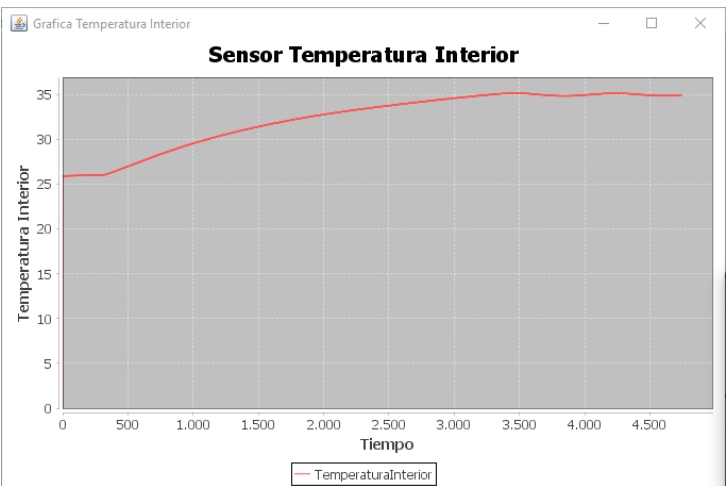


Ilustración 82. Ensayo1 Tipo2 Temperatura

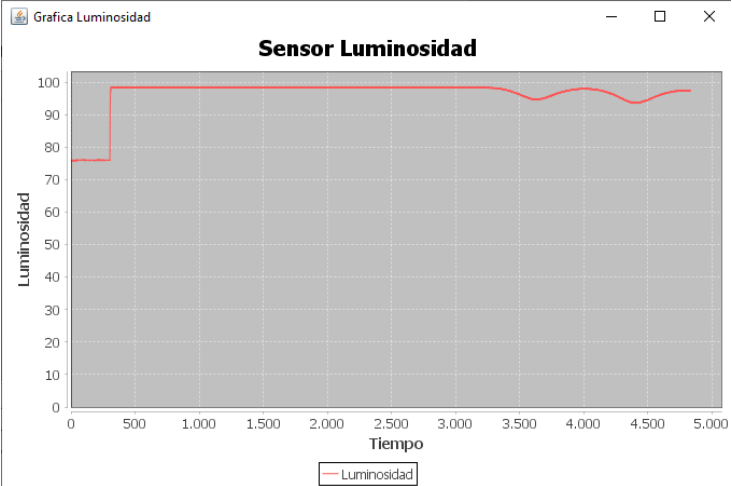


Ilustración 83. Ensayo1 Tipo2 Actuador

El sistema es estable, como se puede observar, aunque el actuador oscile. Tiene un buen tiempo de respuesta, ya que, ha alcanzado la consigna en un breve periodo de tiempo.

En el segundo ensayo, asignaremos las siguientes constantes. Consiguiendo las consecuentes gráficas.

<b>Kp</b>	2
<b>Kd</b>	15
<b>Ki</b>	5

Tabla 11. Constantes PID Ensayo2  
Tipo2



Ilustración 84. Constantes Ensayo2 Tipo2

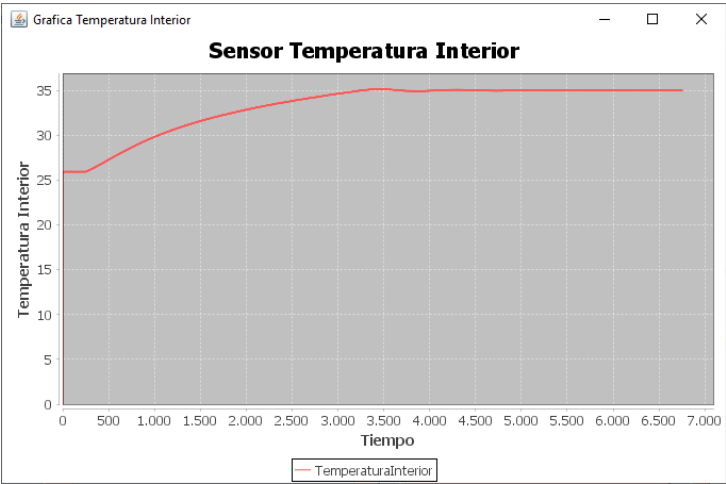


Ilustración 86. Ensayo2 Tipo2 Temperatura

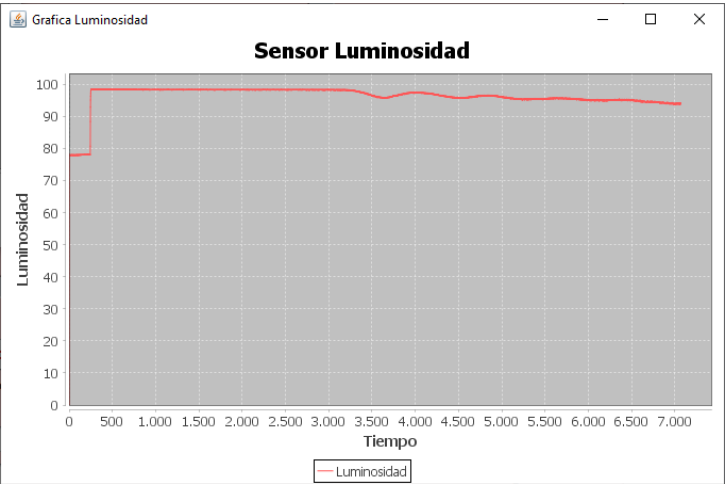


Ilustración 85. Ensayo2 Tipo2 Actuador

El ensayo con resultados más estables hasta ahora, como podemos ver, tiene un tiempo de respuesta corto, con un pequeño sobreimpulso (debido a una acción integral de valor alto). También podemos observar que el actuador se mantiene de forma estable, sin oscilaciones.

En el tercer ensayo, impondremos las constantes de la siguiente tabla. Obtenemos las siguientes gráficas.

Kp	10
Kd	2
Ki	3

Tabla 12. Constantes PID Ensayo3  
Tipo2



Ilustración 87. Constantes Ensayo3 Tipo2

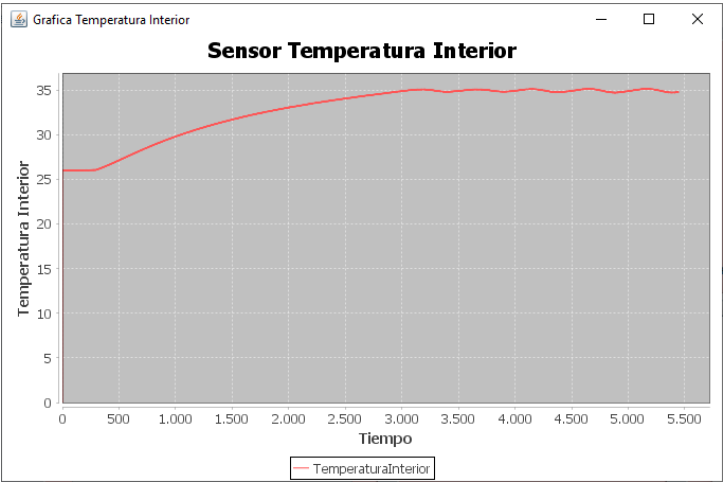


Ilustración 89. Ensayo3 Tipo2 Temperatura

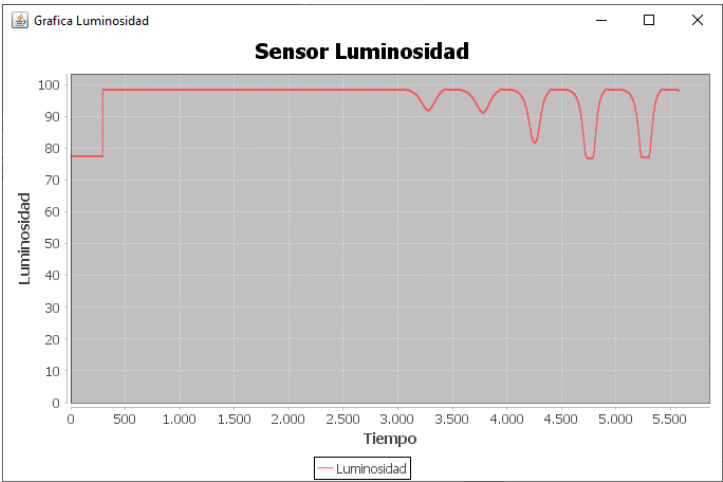


Ilustración 88. Ensayo3 Tipo2 Actuador

Como se puede observar, este sistema se vuelve inestable, primeramente, en nuestra variable controlada, donde se mantiene en la consigna, pero oscilando y en segunda instancia nuestro actuador, el cual, se vuelve inestable al intentar regular la temperatura. Esto se debe a la alta acción proporcional y la baja acción derivativa.

Por último, el cuarto ensayo, en el cual asignaremos unas constantes altas para comprobar como actúa el sistema. Obtendremos de resultado las gráficas siguientes.

<b>Kp</b>	20
<b>Kd</b>	50
<b>Ki</b>	6

Tabla 13. Constantes PID Ensayo4  
Tipo2

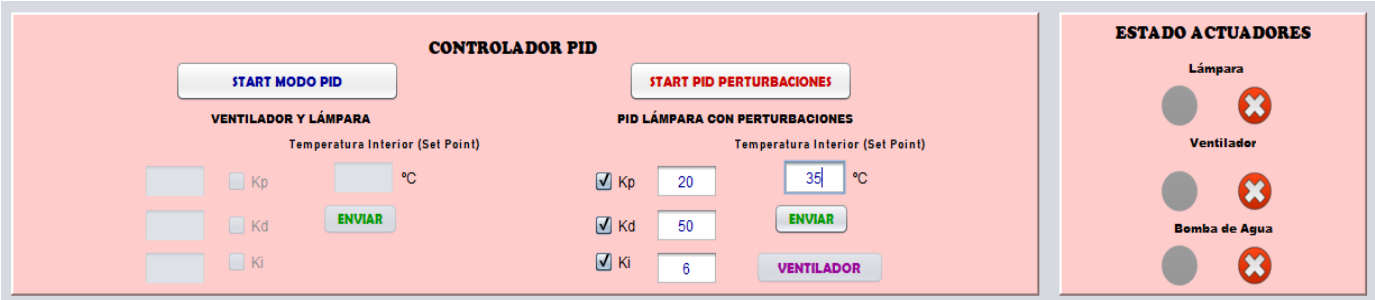


Ilustración 90. Constantes Ensayo4 Tipo2

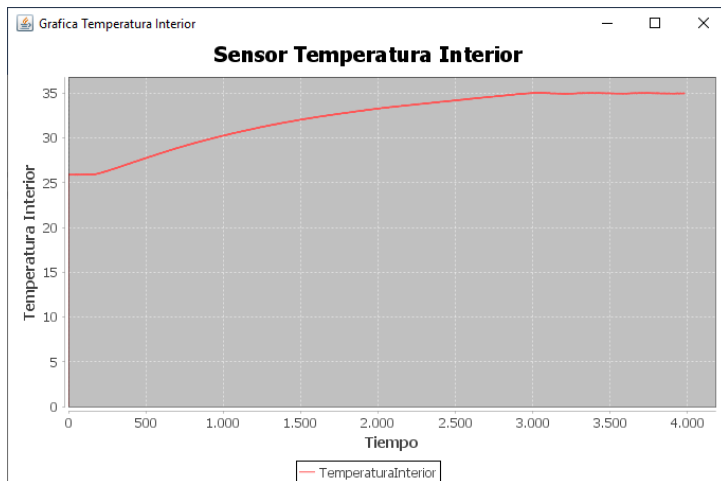


Ilustración 92. Ensayo4 Tipo2 Temperatura

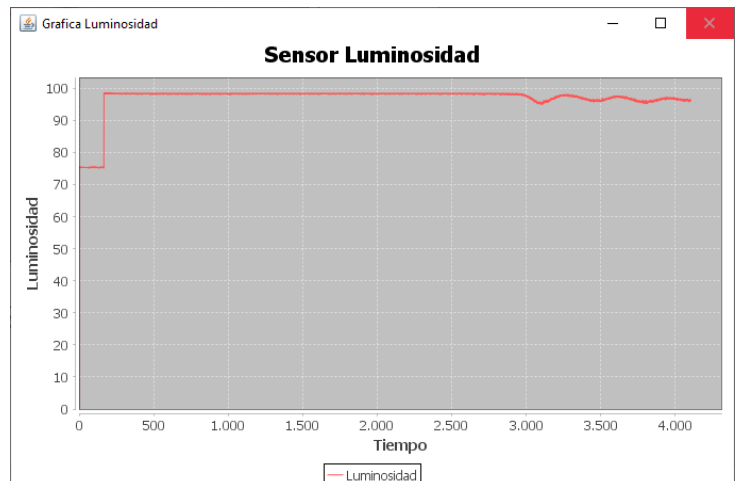


Ilustración 91. Ensayo4 Tipo2 Actuador

El sistema es estable, la variable controlada se mantiene constante en la consigna. El actuador, por otra parte, tiende a oscilar, pero no se vuelve inestable en ningún momento.

## 4. CONCLUSIÓN Y TRABAJOS FUTUROS

En este apartado de la memoria, se describirán posibles propuestas o mejoras para la evolución de este proyecto. Además, se expondrán unas conclusiones, así como hasta que punto se han cumplido los objetivos.

Para empezar, este proyecto ha sido desarrollado para ser útil en un ámbito de enseñanza, es decir, para formar parte de prácticas de las asignaturas de control, control por computador y automática industrial. Por tanto, sería de gran utilidad incluirlo en un laboratorio.

Además, gracias a que el desarrollo de la interfaz se ha realizado en java, sería posible realizarlo como parte de un laboratorio virtual, enfocado en la enseñanza a distancia. Dando la posibilidad de simular nuestra planta por medio de JavaScript.

Otro posible trabajo futuro e interesante, sería la creación de una base de datos en una plataforma Online, esto significaría que podríamos tener nuestro



sistema monitorizado siempre. Gracias a esto, podríamos hacer tratamientos de datos desde otros lugares, sin necesidad de estar conectados (físicamente) al invernadero.

También sería interesante la eliminación del ordenador, ya que ahora mismo es necesario un ordenador para que el sistema funcione. La implementación de un HMI, nos daría autonomía, en cuanto a software se refiere, haciendo el sistema mucho más cerrado y finalizado.

Como última propuesta futura, para que el sistema fuera autónomo, es decir, sin necesidad de estar conectados a la red eléctrica, sería de gran utilidad y empleabilidad, instalarle una pequeña placa fotovoltaica, además de una batería, consiguiendo así un además de un invernadero inteligente, un invernadero ecológico. Se podría así hacer también un estudio (a pequeña escala) del funcionamiento de la placa fotovoltaica, así como el tratamiento de su tensión de salida.

Llegados a este punto, podemos afirmar que se han cumplido todos los objetivos impuestos en un primer momento.

Se han seleccionado todos los componentes hardware y software necesarios. Primeramente, en la parte de hardware, empleando una placa de desarrollo Arduino Mega, un sensor de temperatura DHT11, un sensor de temperatura SHT31, dos sensores de humedad del suelo de tipo SKU SEN0114, un sensor de luminosidad LDR, un sensor de nivel de agua de tipo LS02, tres mosfets de potencia, una bombilla de 60W 24Vdc, tres ventiladores de 12V y una bomba de agua de 12V 0.5A. Respecto a la parte de software, se ha desarrollado un programa en Arduino para la gestión de los sensores y actuadores. Así como, una aplicación en Java, con interfaz gráfica, para hacer posible el control de los parámetros del invernadero, además de la implementación de una captura de datos independiente a la actuación sobre los actuadores o la monitorización de los sensores.

Se ha implementado con éxito, un modelo PID en el invernadero, tanto a nivel de hardware como a nivel de software. Haciendo posible la simulación y la realización de ensayos en esta planta.

También, se ha realizado el diseño y desarrollo de un circuito impreso o PCB. Reduciendo así la cantidad de cableado exterior y eliminando la placa de prototipado. Además de llevarla acabo en una shield de Arduino prefabricada.

En cuanto a la obtención de resultados y comparación con los resultados teóricos. Se han realizado ensayos, de distintos tipos, y se ha explicado el contexto teórico de porque se han producido esos resultados. Obteniendo como conclusión final que la mejor temperatura de Consigna es alrededor de 35°C, así como la elección de las constantes de nuestro modelo, siendo las más óptimas  $K_p=2$ ,  $K_d=15$ ,  $K_i=5$ .

Por último, y para finalizar, para mi la realización de este proyecto ha sido todo un reto. Este proyecto ha supuesto, la implicación de todos mis conocimientos tanto prácticos, como teóricos adquiridos en el Grado de Ingeniería Industrial Electrónica, además de la necesidad de aprender nuevos conocimientos en este campo, como pueden ser la selección de componentes o la programación en Java.

## 5. BIBLIOGRAFÍA

- *Universidad de Gipuzkoa- Definición de Automatización.*  
Disponible en: <http://www.sc.ehu.es>
- *Ecoinventos- IOGreenHouse. 2019.*  
Disponible en: <https://ecoinventos.com/>
- *Novedades Agrícolas (Novagric)- Invernaderos Automatizados. 2016.*  
Disponible en: <http://www.novagric.com/>
- *IES Camp de Morvedre- Domotic Green House.*  
Disponible en: <http://www.iescamp.es/>
- *Teslabem- Fundamentos del protocolo I2C. 2017.*  
Disponible en: <https://teslabem.com/>
- *NaylampMechatronics- Sensor de humedad relativa y temperatura SHT31.*  
Disponible en: <https://naylampmechatronics.com/>
- *Wika- ¿Qué es y como funciona un interruptor de flotador? 2017.*  
Disponible en : <https://www.bloginstrumentacion.com/>
- *Omega- Sensor de nivel.*  
Disponible en: <https://es.omega.com/>
- *Mecatrónica Latam- Fotorresistencia LDR. 2018.*  
Disponible en: <https://www.mecatronicalatam.com/>
- *Ecured- Lámpara Incandescente.*  
Disponible en: <https://www.ecured.cu/>
- *OpenWebinars- Entradas analógicas y digitales. 2015*  
Disponible en: <https://openwebinars.net/>

- *NaylampMechatronics- Driver Mosfet IRF520.*

Disponible en: <https://naylampmechatronics.com/>

- *PanamaHitek- ¿Qué es y cómo funciona un mosfet? 2016.*

Disponible en: <http://panamahitek.com/>

- *Definición de Java.*

Disponible en: <https://definicion.mx/java/>

- *Thebigwebdeveloper- Instalación de Netbeans paso a paso.*

Disponible en: <https://thebigwebdeveloper.blogspot.com/>

## 6. ANEXOS

### 6.1 Manual NetBeans

En este manual se explicará la instalación de NetBeans, así como el plugin necesario para hacer posible la utilización del software.

Comenzaremos con la instalación de NetBeans. Primero, debes ir al siguiente enlace y descargar la última versión de NetBeans.

<https://netbeans.apache.org/>

Seleccionaremos el idioma que queremos que tenga nuestro software. Además del sistema operativo donde vamos a instalar nuestro programa.

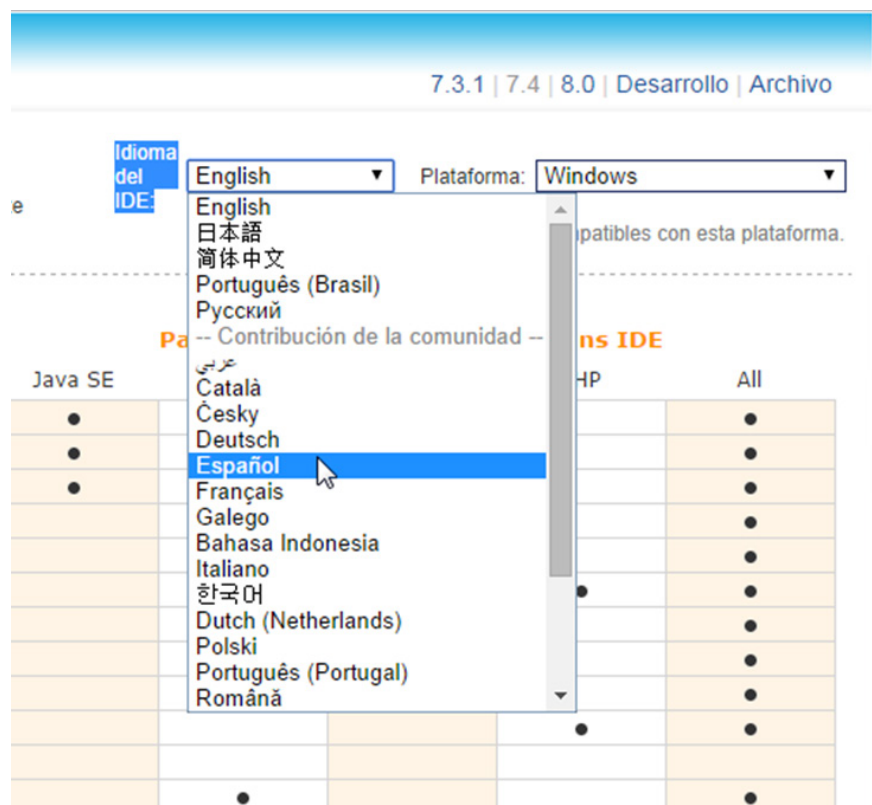
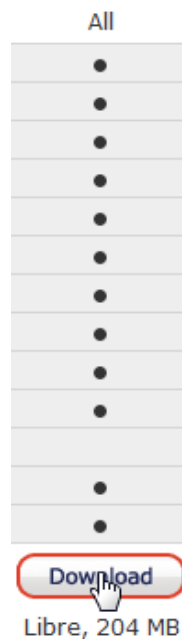


Ilustración 93. Selección de Idioma y Sistema Operativo

Ahora es muy importante seleccionar “Download All” en la selección de paquetes, ya que necesitaremos varios paquetes a la hora de programar.



*Ilustración 94. Selección de Paquetes a descargar*

Una vez tengamos el archivo ejecutable del IDE, procedemos a descargar el Java Development Kit, necesario par instalar Java. Ve al siguiente enlace:

<https://www.oracle.com/>

El siguiente paso, será ubicar el archivo a descargar y que sea compatible con nuestro sistema operativo.

Product / File Description	File Size	Download
Linux x86	119.67 MB	<a href="#">jdk-7u67-linux-i586.rpm</a>
Linux x86	136.94 MB	<a href="#">jdk-7u67-linux-i586.tar.gz</a>
Linux x64	120.98 MB	<a href="#">jdk-7u67-linux-x64.rpm</a>
Linux x64	135.78 MB	<a href="#">jdk-7u67-linux-x64.tar.gz</a>
Mac OS X x64	186.01 MB	<a href="#">jdk-7u67-macosx-x64.dmg</a>
Solaris SPARC (SVR4 package)	138.77 MB	<a href="#">jdk-7u67-solaris-sparc.tar.Z</a>
Solaris SPARC	98.61 MB	<a href="#">jdk-7u67-solaris-sparc.tar.gz</a>
Solaris SPARC 64-bit (SVR4 package)	23.99 MB	<a href="#">jdk-7u67-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	18.39 MB	<a href="#">jdk-7u67-solaris-sparcv9.tar.gz</a>
Solaris x64 (SVR4 package)	24.74 MB	<a href="#">jdk-7u67-solaris-x64.tar.Z</a>
Solaris x64	16.35 MB	<a href="#">jdk-7u67-solaris-x64.tar.gz</a>
Solaris x86 (SVR4 package)	139.39 MB	<a href="#">jdk-7u67-solaris-i586.tar.Z</a>
Solaris x86	95.47 MB	<a href="#">jdk-7u67-solaris-i586.tar.gz</a>
Windows x86	127.98 MB	<a href="#">jdk-7u67-windows-i586.exe</a>
Windows x64	129.7 MB	<a href="#">jdk-7u67-windows-x64.exe</a>

*Ilustración 95. Descarga JDK*

Una vez tengamos el JDK, procedemos a su instalación. Se nos abrirá una ventana de instalación, y daremos clic en “Next” a las tres ventanas y por último en “Close”.

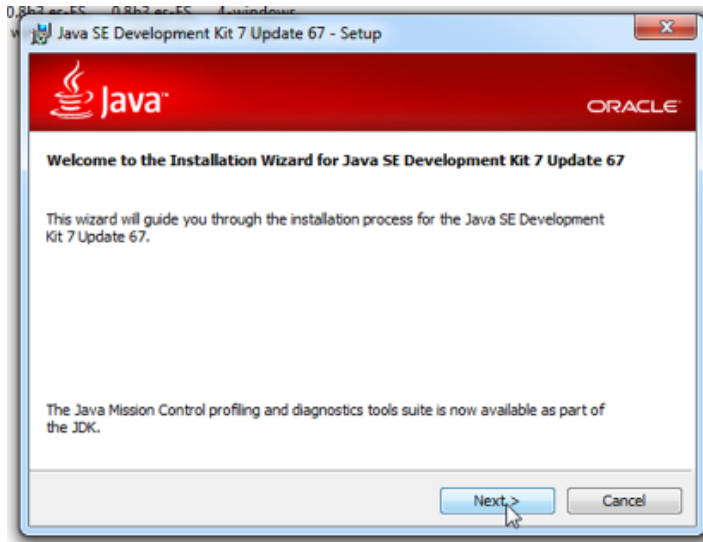


Ilustración 97. Primera Pantalla de Instalación

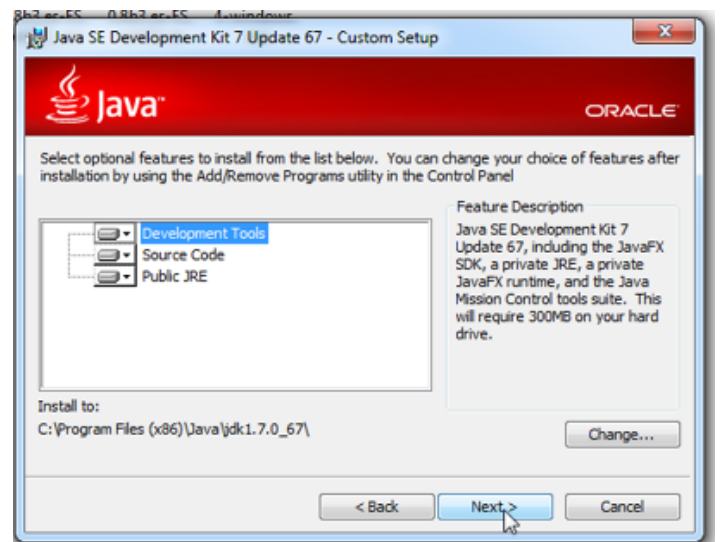


Ilustración 96. Segunda Pantalla de Instalación

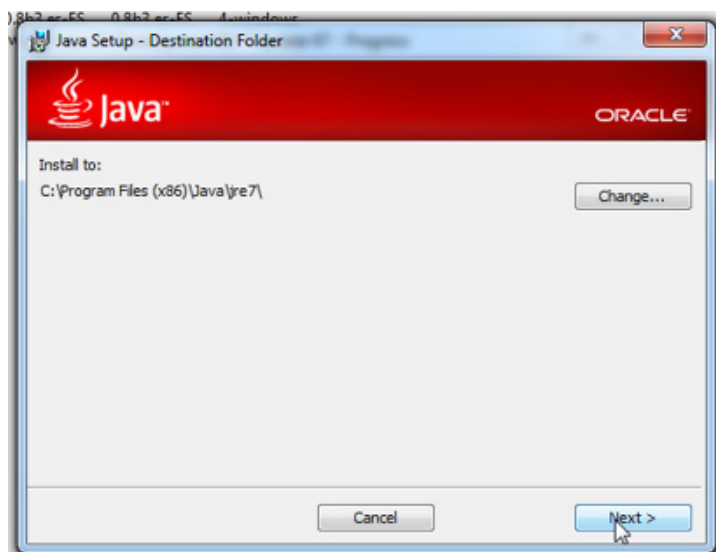


Ilustración 99. Tercera Pantalla de Instalación



Ilustración 98. Cuarta Pantalla de Instalación

Ahora ubicaremos el archivo ejecutable de NetBeans y daremos doble clic, para proceder a su instalación. Se nos abrirá una ventana, en la cual podremos ver los componentes y características que se van a proceder a instalar. Podemos

personalizar estos componentes utilizando el botón “Personalizar”, que nos aparece en la pantalla.

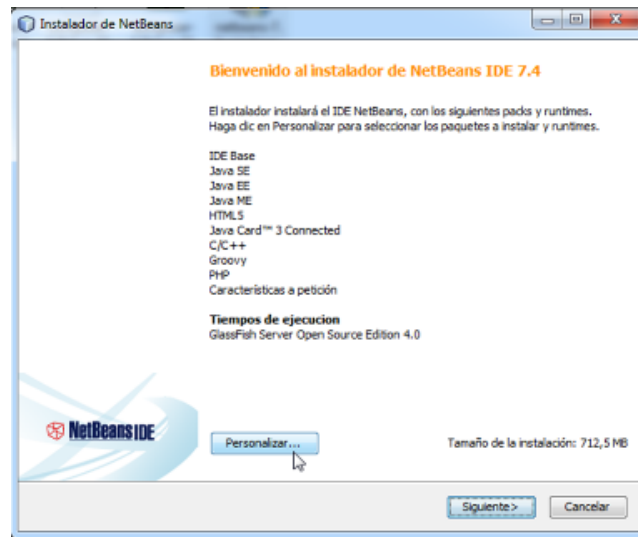


Ilustración 100. Primera Pantalla NetBeans

Una vez escogidos los paquetes que queremos instalar, nos aparecerá una ventana de acepto de términos de uso, la cual deberemos marcar y dar al botón “Siguiente”.

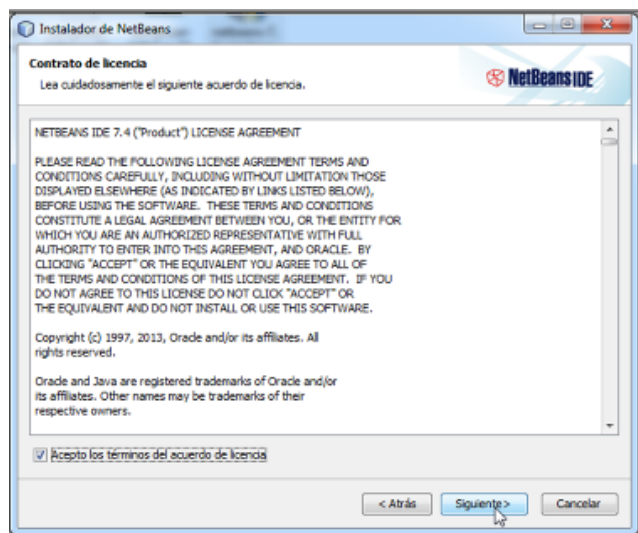
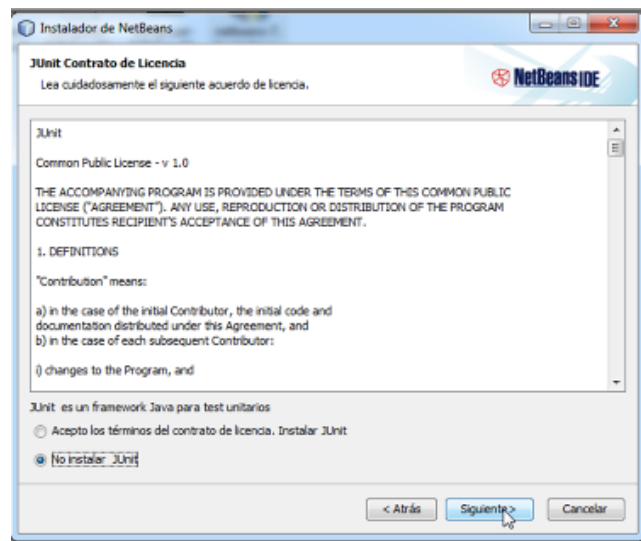


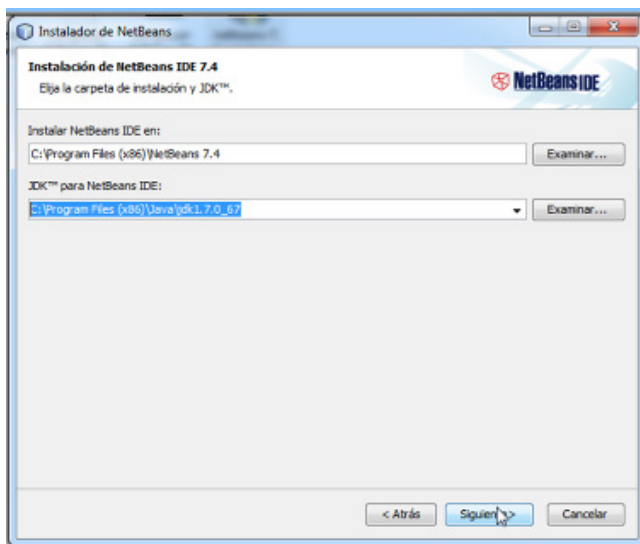
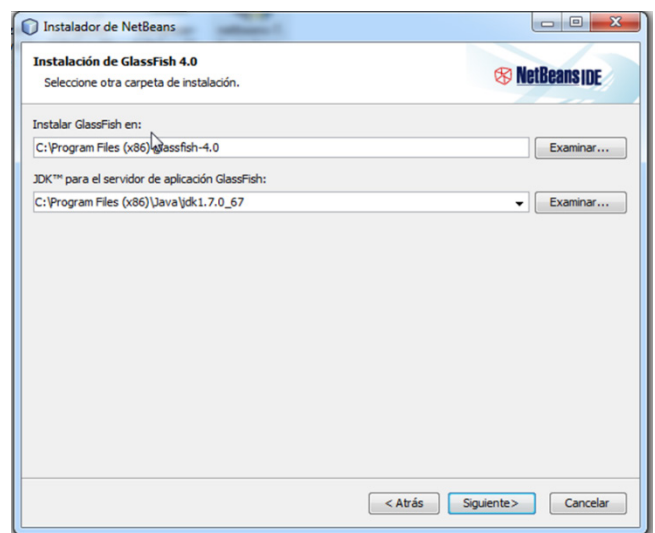
Ilustración 101. Segunda Pantalla NetBeans

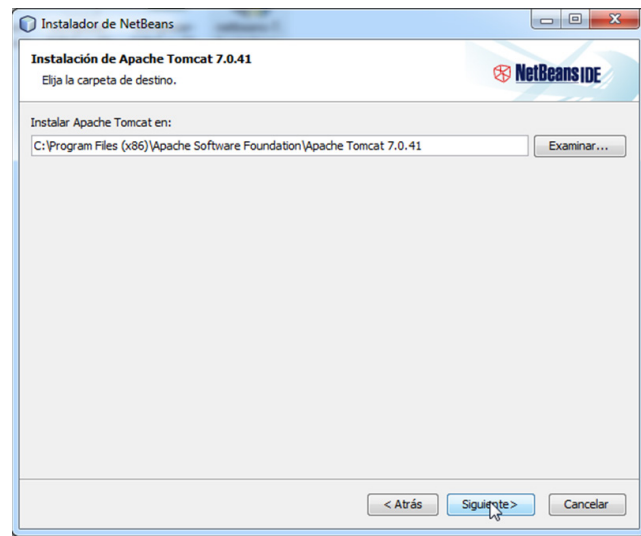
Se nos abrirá otra ventana, la cual indicaremos que no queremos instalar Junit y volveremos a pulsar el botón “Siguiente”.



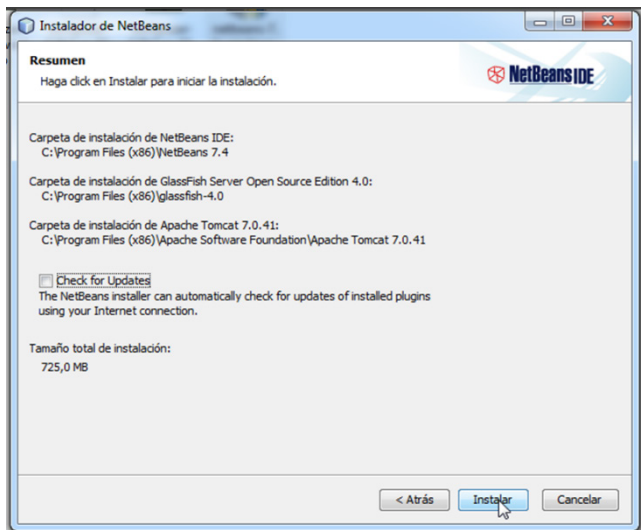
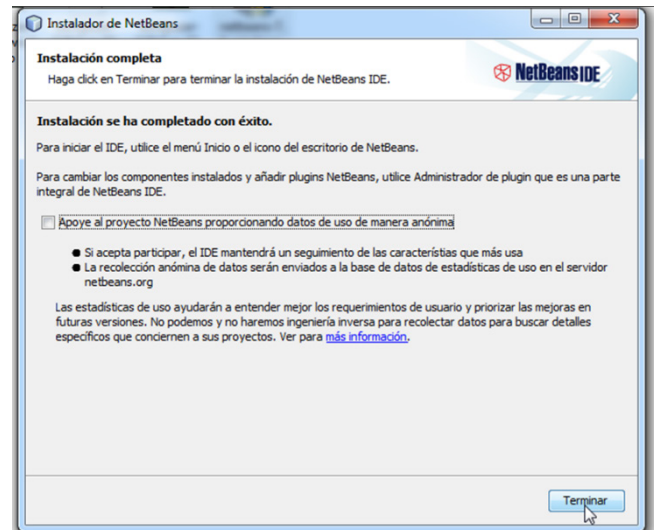
*Ilustración 102. Tercera Pantalla NetBeans*

Las tres siguientes ventanas son para definir las rutas de instalación de NetBeans, Glassfish y Apache. En este caso, no las vamos a cambiar y por tanto pulsaremos el botón “Siguiendo” en los tres casos.

*Ilustración 103. Cuarta Pantalla NetBeans**Ilustración 104. Quinta Pantalla NetBeans*

*Ilustración 105. Sexta Pantalla NetBeans*

Por último, nos aparecerá una ventana que nos indicará un botón para proceder a la instalación. Pulsaremos “Instalar” y esperaremos unos minutos a que termine la instalación y pulsaremos “Terminar”.

*Ilustración 106. Séptima Pantalla NetBeans**Ilustración 107. Octava Pantalla NetBeans*

## 6.2 Manual Arduino

Antes de comenzar la instalación del IDE de Arduino, debes descargar su última versión en el siguiente enlace:

<https://www.arduino.cc/en/Main/Software>

Debes seleccionar ahora, el instalador del sistema operativo en el cual se va a instalar el software. En nuestro caso Windows.

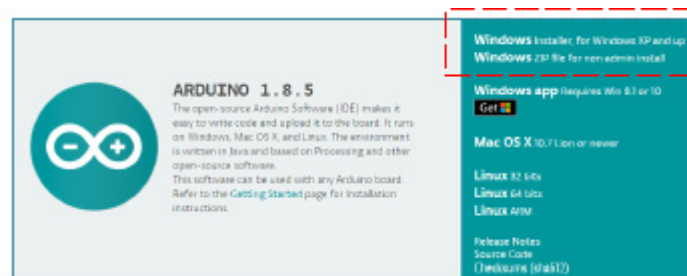


Ilustración 108. Descarga Arduino

Para iniciar la instalación, debes aceptar el acuerdo de licencia y a continuación marcar todos los componentes que ofrece para incluir en la instalación.

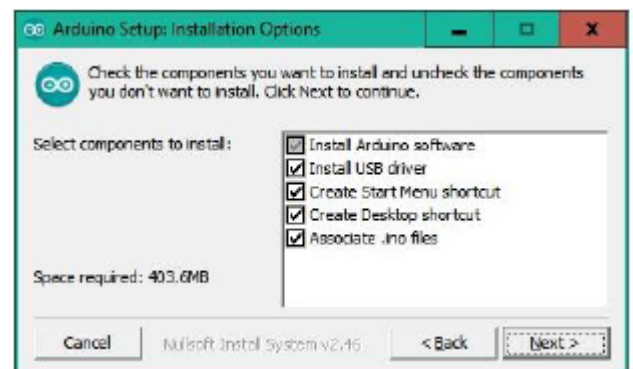


Ilustración 109. Primera Pantalla Arduino

A continuación, selecciona el directorio donde se realizará la instalación de Arduino, y pulsaremos el botón "Install". Esperaremos unos minutos a que haya finalizado la instalación y ya podremos empezar a utilizar nuestra herramienta de Arduino.

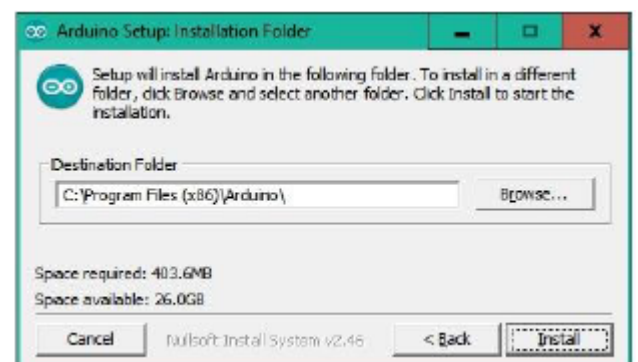


Ilustración 110. Segunda Pantalla Arduino