



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSIDAD POLITÉCNICA DE VALENCIA  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE DISEÑO  
GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

---

# **Diseño y control de un Invernadero automatizado de bajo coste adaptado para su desarrollo en Educación Primaria.**

## **Design and control of a low-cost automated Greenhouse adapted for its development in Primary Education.**

---

TRABAJO FIN DE GRADO

**AUTOR:** ALBERTO DELGADO ROMERO

**TUTOR:** LEOPOLDO ARMESTO ÁNGEL

VALENCIA, JUNIO 2019



## Agradecimientos

*“A mis padres, a mi hermano, a mis abuelos y al resto de mi familia,  
por apoyarme en todo momento y por brindarme  
la oportunidad de estudiar fuera de casa.”*

*“A Claudia por estar siempre  
atenta y animarme en todo lo que me proponga.”*

*“A Leopoldo, por confiar en mí y guiarme en el camino.”*



## Resumen

El propósito principal de este Trabajo Fin de Grado es el diseño y control de un invernadero automatizado para su desarrollo en Educación Primaria, atendiendo a necesidades de bajo coste y facilidad de uso por parte de los alumnos y el profesorado.

En el grueso de este proyecto, se encontrará el estudio de la placa microcontroladora a utilizar junto con los módulos necesarios para su automatización y la tecnología Easy-Plug utilizada para su conexión, así como los materiales sostenibles escogidos para la fabricación de la base del invernadero.

Además, se desarrollarán los módulos de programación por bloques para su integración en Facilino Junior, una herramienta de programación por bloques para Arduino basada en la librería Blockly de Google.

También se incluye el marco de plan de estudio en el que se engloba dicho proyecto y las mejoras en la educación, pensamiento y concienciación que aporta a los alumnos la realización de este.

Finalmente, se mostrará el diseño y la fabricación en 3D de un estuche para el alojamiento y protección de la electrónica.

**Palabras clave:** Electrónica; Programación por bloques; Easy-Plug; Impresión 3D; Microcontrolador; Robótica Educativa.



## Resum

El propòsit principal d'aquest treball final de grau és el disseny i control d'un hivernacle automatitzat per a desenvolupar-lo en educació primària, tenint en compte necessitats de baix cost i facilitat d'ús per part de l'alumnat i el professorat.

En el gruix d'aquest projecte, es troba l'estudi de la placa microcontroladora que cal utilitzar, juntament amb els mòduls necessaris per a l'automatització i la tecnologia *easy-plug* utilitzada per a la connexió, així com els materials sostenibles seleccionats per a la fabricació de la base de l'hivernacle.

A més, es desenvolupen els mòduls de programació per blocs per a integrar-los en Facilino Junior, una eina de programació per blocs per a Arduino basada en la biblioteca (*library*) Blockly de Google.

També s'inclou el marc del pla d'estudi en el qual s'engloba aquest projecte, i les millores en l'educació, pensament i conscienciació que aporta a l'alumnat la realització d'aquest.

Finalment, es mostra el disseny i la fabricació en 3D d'un estoig per a l'allotjament i la protecció de l'electrònica.

**Paraules clau:** programació per blocs; *easy-plug*; impressió en 3D; microcontrolador; robòtica educativa;



## Abstract

The main purpose of this Final Degree Project is the design and control of an automated greenhouse for its development in Primary Education, attending to needs of low cost and ease of use by students and faculty.

In the bulk of this project, you will find the study of the microcontroller board to be used together with the modules needed for its automation and the Easy-Plug technology used for its connection, as well as the sustainable materials chosen for the manufacture of the base of the greenhouse .

In addition, block programming modules will be developed for integration into Facilino Junior, a block programming tool for Arduino based on Google's Blockly library.

It also includes the study plan framework in which this project is included and the improvements in education, thought and awareness that students bring to the realization of this.

Finally, it includes the design and 3D fabrication of a case for the housing and protection of the electronics.

**Keywords:** Electronics; Block programming; Easy-Plug; 3D print; Microcontroller; Educational Robotics;



## Contenidos

|                     |    |
|---------------------|----|
| 1. MEMORIA.....     | 1  |
| 2. PLANOS .....     | 83 |
| 3. PRESUPUESTO..... | 93 |
| 4. ANEXOS .....     | 99 |





UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSIDAD POLITÉCNICA DE VALENCIA  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE DISEÑO  
GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

---

# **Diseño y control de un Invernadero automatizado de bajo coste adaptado para su desarrollo en educación primaria.**

## **Design and control of a low-cost automated Greenhouse adapted for its development in primary education.**

---

### **1. MEMORIA**

TRABAJO DE FIN DE GRADO

**AUTOR:** ALBERTO DELGADO ROMERO

**TUTOR:** LEOPOLDO ARMESTO ÁNGEL

VALENCIA, JUNIO 2019



## Índice

|   |    |
|---|----|
| 1. OBJETIVO .....   | 5  |
| 2. INTRODUCCIÓN .....                                       | 7  |
| 2.1. Importancia de la tecnología en la educación.....      | 7  |
| 2.2. Robótica en la educación .....                         | 8  |
| 2.3. Robótica educativa: Software .....                     | 9  |
| 2.4. Robótica Educativa: Hardware .....                     | 10 |
| 2.5. Antecedentes .....                                     | 12 |
| 2.6. Motivación.....  | 15 |
| 2.7. Justificación .....                                    | 16 |
| 3. NORMATIVA DE IMPLEMETACIÓN .....                         | 17 |
| 4. DESARROLLO DEL PROYECTO .....                            | 21 |
| 4.1. Hardware. Easy-Plug KeyStudio .....                    | 21 |
| 4.1.1. Placa principal.....                                 | 21 |
| 4.1.2. Conexiones .....                                     | 26 |
| 4.1.3. Sensores y módulos de comunicación externos .....    | 27 |
| 4.1.4. Actuadores externos .....                            | 32 |
| 4.1.5. Alimentación de la placa y actuadores externos.....  | 34 |
| 4.2. Software. Facilino Junior .....                        | 35 |
| 4.2.1. Estructura-Base. Blockly de Google.....              | 35 |
| 4.2.2. Interfaz de usuario. ....                            | 38 |
| 4.2.3. Microcontroladores compatibles. ....                 | 39 |
| 4.2.4. Idiomas disponibles.....                             | 40 |
| 4.2.5. Programación de Facilino Junior con JavaScript. .... | 40 |
| 4.2.6. Categorías existentes y bloques creados. ....        | 45 |
| 4.3. Invernadero .....                                      | 52 |
| 4.4. Montaje del proyecto.....                              | 56 |
| 4.4.1. Montaje del hardware de control. ....                | 56 |
| 4.4.2. Diseño y desarrollo del software de control. ....    | 58 |
| 4.4.3. Montaje del invernadero.....                         | 68 |
| 4.5. Diseño guion del proyecto.....                         | 72 |
| 5. RESULTADOS DEL PROYECTO .....                            | 73 |
| 5.1. Consideraciones a tener en cuenta.....                 | 73 |
| 5.2. Funcionamiento del proyecto .....                      | 73 |

|                               |    |
|-------------------------------|----|
| 5.3. Acceso a documentos..... | 78 |
| 6. CONCLUSIÓN Y MEJORAS.....  | 79 |
| 7. BIBLIOGRAFÍA .....         | 81 |

## 1. OBJETIVO

El objetivo de este Trabajo Fin de Grado es la realización de un invernadero totalmente automatizado y autosuficiente para su fabricación y programación por alumnos de Educación Primaria.

Este invernadero destaca por ser de bajo coste, para su posible su implementación en la educación y robusto, para que los componentes utilizados duren en el tiempo y el proyecto pueda elaborarse año tras año.

El proceso de programación y montaje del invernadero se han simplificado al máximo para que los alumnos, en la medida de lo posible, puedan desarrollar el proyecto de forma autónoma.

- Para el montaje del automatismo del invernadero se ha escogido el uso de la tecnología de conexión Easy-Plug, que facilita las conexiones entre los módulos de control del proyecto, eliminando de esta manera las principales barreras de entrada a las que este tipo de proyectos se enfrentan cuando se plantea su desarrollo en Educación Primaria.
- Para la programación de las instrucciones se ha elegido el software Facilino Junior, para el cual se desarrollarán los módulos de programación por bloques necesarios, con la ventaja de la personalización de estos para la realización del proyecto en cuestión.

En el presente Trabajo Fin de Grado también se estudiará el plan de estudios y el método a seguir para la introducción de este trabajo en el último ciclo de Educación Primaria, además de los nuevos pensamientos, conceptos, competencias y capacidades que los alumnos pueden adquirir a raíz de la realización de este, teniendo en cuenta que abarca dos materias fundamentalmente:

- Robótica Educativa, como campo para la introducción conceptos más concretos de tecnología en la Educación
- Educación Medioambiental, como concepto para la concienciación y el cambio de comportamientos erróneos hacia el medio ambiente en las nuevas generaciones

Por último, también se desarrollará y diseñará un estuche en 3D para el alojamiento y protección de todos los componentes y aparatos electrónicos del proyecto así como la configuración de su impresión en 3D.



## 2. INTRODUCCIÓN

### 2.1. Importancia de la tecnología en la educación

Actualmente, el ámbito de la enseñanza es un medio cuya progresión no cesa. Una constante actualización de las fuentes de aprendizaje junto con una modificación insistente de las leyes que la regulan hace que en algunos casos, la enseñanza pierda cierta estructura y coherencia.

Los jóvenes de hoy en día pertenecen a la llamada generación de los nativos digitales por haber nacido después de la invención de internet. Por ello, la incorporación de la tecnología a la educación es fundamental y aporta al alumno una gran cantidad de beneficios para su aprendizaje mejorando así su productividad e incorporando pensamientos nuevos más flexibles y creativos como el Pensamiento Computacional. También aporta otros beneficios como la mejora de la capacidad de trabajo en grupo o el desarrollo de relaciones interpersonales con otros alumnos gracias a las redes sociales.

La tecnología lleva ya años siendo utilizada en el proceso educativo. Desde la aparición de las primeras calculadoras, los primeros ordenadores para el uso de procesadores de texto hasta el uso de las impresoras. Algunos factores como el auge de internet y de dispositivos móviles ha propiciado que la tecnología avance de ser un complemento en la educación a ser el pilar fundamental de esta, revolucionando la forma de educar y aprender.

En la Educación Primaria, la tecnología está presente en varios ámbitos: por un lado en las aulas, gracias a equipos como los proyectores, o las pizarras interactivas, que ayudan a los alumnos a tomar la iniciativa de participar activamente en las clases, y por otro lado, en el trabajo personal de cada estudiante, gracias a las aulas virtuales que permiten el intercambio de archivos entre el profesor y el alumno y viceversa.

Gracias a estas aulas virtuales, se está viendo reemplazado el uso de los libros físicos a libros virtuales, permitiendo a los alumnos obtener resultados y correcciones inmediatas de sus ejercicios, la capacidad de personalizar y adaptar los contenidos, así como indirectamente, una mejora en la salud al no tener que soportar diariamente el peso de los libros tradicionales.

Es importante que los alumnos comiencen a desarrollar conocimientos más concretos sobre tecnología, adquiriéndolos en asignaturas dedicadas a ello e igualando el aprendizaje en este campo a las materias tradicionales como Lengua Castellana, Matemáticas, Ciencias Naturales y Ciencias Sociales, Educación física o Artística. En el presente Trabajo Fin de Grado este es uno de los objetivos que se pretende abordar.

La robótica es uno de los campos tecnológicos más utilizados para acercar al alumno a conocimientos más concretos y especializados de tecnología. Es una de las disciplinas tecnológicas más extendidas y utilizada por el ser humano, ayudando a este en todo tipo de ámbitos, desde el doméstico, la medicina, la industria, el entretenimiento y la educación entre otros. Es por ello, que este último debe ser el encargado de formar a las nuevas generaciones en este terreno, permitiéndoles realizar un buen uso y conocer las ventajas y desventajas que puede aportar a la sociedad este campo de la tecnología.

Según un artículo publicado por la Revista Educación, el aprendizaje de la robótica puede lograrse desde dos puntos: a partir de la conceptualización sobre robótica o a partir del diseño y construcción de aparatos robóticos, ambos serán discutidos a lo largo de esta Trabajo Fin de Grado [1].

## 2.2. Robótica en la educación

En la enseñanza, como se ha indicado anteriormente, uno de los métodos más sencillos y dinámicos para introducir aspectos más concretos de tecnología en las aulas es la robótica.

La Robótica Educativa es una disciplina basada en el desarrollo de una pequeña construcción robótica mediante la cual los alumnos aprenden lenguaje de programación por ensayo-error logrando que el robot realice tareas no muy complejas.

El método más sencillo y común para la introducción de conceptos de Robótica Educativa en las aulas es a través del lenguaje de programación.

El lenguaje de programación aporta una nueva forma de expresión para los alumnos, un lenguaje nuevo como el lenguaje hablado, el matemático o el musical y permite al alumno utilizarlo para afrontar tareas y problemas del día a día. Lo más difícil para el alumno es ser consciente de que el robot va a cumplir las órdenes que se le indiquen.

En la educación, la programación siempre se plantea mediante el uso de bloques. Este tipo de programación facilita la tarea de comprensión a los alumnos. Los bloques son piezas de tipo puzle que enlazándose unas con otras crean las instrucciones que conformaran el programa. Es un lenguaje modular que permite apilar las instrucciones.

El desarrollo de esta habilidad fomenta la creatividad y el pensamiento computacional en los alumnos. Dicho pensamiento computacional, ayuda al alumno a resolver un problema mediante la división de este en objetivos más simples, estableciendo una secuencia o algoritmo con todos ellos. Además, logra desarrollar habilidades de forma sistemática, como el pensamiento crítico y fomenta la atención y el trabajo en equipo.

Según un estudio realizado por el Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado, el pensamiento computacional se divide en cinco habilidades [2]:

- Abstracción: proceso mediante el cual se reducen los detalles innecesarios de un artefacto para hacerlo más comprensible.
- Pensamiento algorítmico: camino para llegar a la solución de un problema gracias a una clara definición de los pasos.
- Automatización: proceso de ahorro de trabajo gracias al cual, un ordenador es programado para realizar un conjunto de tareas repetitivas rápida y eficientemente.
- Descomposición: consiste en desglosar los artefactos en sus partes y componentes para comprenderse por separado.
- Depuración: aplicación sistemática de las habilidades de análisis y evaluación utilizando el pensamiento lógico para predecir y verificar los resultados.
- Generalización: consiste en la identificación de patrones, similitudes y conexiones, y la explotación de las características.

Con el desarrollo de actividades que incluyan estos conceptos de robótica educativa, programación y pensamiento computacional, se fomentan en los alumnos el desarrollo de estas habilidades y conocimientos, con el objetivo principal de formar, preparar e integrar a los alumnos en los nuevos avances tecnológicos que están aún por llegar.

Para la aplicación de la Robótica en la educación, se subdivide en sus dos campos fundamentales: Software y Hardware que serán explicados a continuación.

### 2.3. Robótica educativa: Software

El concepto de software en la Robótica Educativa corresponde con la programación de las acciones del robot. Dicha programación se lleva a cabo utilizando un método especial, atractivo y motivador para el alumnado, buscando la máxima simplificación de esta tarea. Este tipo de programación es modular, basada en una lógica de bloques de colores que encajan o no entre sí completando una secuencia de operaciones e instrucciones.

En la Figura 1, se muestra un ejemplo de combinación de bloques donde a cada uno de ellos le corresponde un tipo de instrucción.



Figura 1. Ejemplo de programación por bloques. Código en Scratch

En el caso de la combinación de bloques de la Figura 1 , el color amarillo identifica los bloques que crean una instrucción de un evento, el color morado identifica los bloques que crean instrucciones de acción y el color naranja corresponde con los bloques que crean una instrucción de control.

El alumno contará con la capacidad de analizar la tarea de forma global, dividiéndola y planificándola en objetivos parciales en los que el problema ha de descomponerse. Es el propio alumno quien debe ir programando y probando cada objetivo por separado, y cuando el funcionamiento individualizado de los mismos sea satisfactorio, se agruparán formando el programa principal. Si una tarea no funciona correctamente o no da el resultado esperado, se debe rectificar y volver a programar hasta que su funcionamiento sea el deseado, avanzando así en la secuencia global.

El alumno deberá ser capaz de escribir los programas pero también de leerlos y comprender las instrucciones que está programando, por ello, las actividades planteadas son de dificultad creciente, comenzando por el uso de los bloques más simples como por ejemplo bloques de sensores, hasta los bloques más complejos como los de control, lógica u operadores, que contienen conceptos más propios de código y programación general.

Para facilitar esta labor a priori compleja, en el presente Trabajo Fin de Grado se propone la utilización de un software que permite la simplificación de este trabajo, Facilino Junior, que será desarrollado a lo largo de esta memoria.



El término *Easy-Plug* hace referencia a conexiones fáciles de realizar entre dos componentes electrónicos, permitiendo su conexión y desconexión sin deterioros ni desperfectos en ninguno de los dos componentes, y unificando el conjunto de cables que forman la conexión en un solo conector. Este método es utilizado para la conexión de equipos de telecomunicaciones o de datos, como por ejemplo los mostrados en la Figura 3.



*Figura 3. De izquierda a derecha, conector RJ45 y conector RJ11*

El uso principal de este tipo de conexiones son las redes de telefonía (RJ11), o cableado de Ethernet (RJ45) debido a la ventaja que supone por la gran cantidad de cables que son necesarios para la conexión entre los equipos.

Otro de los problemas que plantea resolver el actual Trabajo Fin de grado es la falta de robustez de las placas microcontroladoras utilizadas con normalidad. Las placas con esta finalidad van a ser sometidas a un fuerte desgaste por parte de los alumnos, por lo que deben ser resistentes y estar dotadas con protecciones para evitar la rotura de estas en caso de caídas o golpes. Esta placa *Easy-Plug* escogida cuenta con todas estas mejoras importantes a la hora de la incorporación de un proyecto de este ámbito en educación.

## 2.5. Antecedentes

A mediados de los años noventa, se comienzan a utilizar por primera vez los robots para apoyar el aprendizaje y la oferta de cursos en colegios y universidades aumenta a la vez que se comienza a investigar sobre un nuevo campo de desarrollo conocido con el término “Robótica Educativa”. Asimismo, algunas empresas comienzan a fabricar material como base de esta nueva vertiente educativa.

En 2001, se lleva a cabo un estudio sobre el uso de la robótica en la educación clasificándola en dos tipos: “*Robótica en educación y robótica para la educación*”, entendiéndolo primero como una materia más dentro de la enseñanza de los alumnos, y entendiéndolo segundo como una herramienta para enseñar otros campos [3].

En el ámbito de la educación, se han llevado a cabo medidas de inserción de robótica en diversos centros buscando una combinación de ambos tipos para el desarrollo en los alumnos de nuevas habilidades a través de nuevas estrategias de aprendizaje.

Siguiendo el principio de división de aprendizaje de robótica en dos puntos mencionado con anterioridad en el apartado 2.1 pero con una adaptación a la educación durante los primeros cursos de primaria, se exponen los diferentes intentos de introducción de robótica educativa agrupados en aprendizaje a partir de conceptualización y aprendizaje a partir de diseño y construcción:

### 1. Aprendizaje de robótica a partir de conceptualización o manejo de robots:

Este primer intento del aprendizaje contempla el estudio de conocimientos sobre los elementos que forman los robots, los materiales, las funciones que pueden realizar, así como las órdenes que son capaces de ejecutar. Para este método de aprendizaje de robótica, se suele hacer uso de kits y componentes compactos, para centrarse más en el aprendizaje conceptual.

Algunos ejemplos ya existentes de este aprendizaje conceptual son:

- Proyecto Beebot: Es un robot con forma de abeja programable muy fácil de utilizar ya que no necesita ningún montaje previo. Es recomendado para niños mayores de tres años, y hasta alumnos de primeros cursos de primaria. Permite programación de secuencias básicas de movimiento (“adelante”, “atrás”, “giro 90º” entre otras). El propio robot contiene batería incorporada y botones para realizar acciones de forma manual como podemos observar en la Figura 4.



Figura 4. Proyecto Beebot.

La programación de dicho robot se puede realizar a través de dispositivos móviles y además, este proyecto dispone de material complementario como alfombras o cartas para ampliar conocimientos en otras materias.

- Proyecto Cubetto: Es un robot de madera, que permite su programación sin pantallas mediante el juego sensorial. Está diseñado para niños que no poseen aun conocimientos ni de lectura ni de escritura, a partir de tres años. Consta de tres partes como se puede observar en la Figura 5:
  - o Robot principal. Realizará los movimientos establecidos dependiendo de la colocación de las fichas de programación en el tablero de control.
  - o Tablero de control, donde se colocan las fichas de colores en un cierto orden, programando así la secuencia que el robot ejecutará.
  - o Fichas de programación. Cada una de ellas representa una acción, permitiendo todo tipo de combinaciones entre sí para formar secuencias únicas.



Figura 5. Proyecto Cubetto. De izquierda a derecha, robot, tablero y fichas. [4]

Sendos proyectos inculcan en los alumnos un primer contacto con este tipo de educación, así como unos conocimientos fundamentales sobre robótica, muy útiles en su desarrollo y para su progreso atendiendo a esta nueva forma de aprender y crecer.

## 2. Aprendizaje a partir del diseño y construcción:

Este segundo caso, dirigido a alumnos de edades más avanzadas, inculca en los estudiantes conocimientos algo más profundos sobre la materia, permitiendo el desarrollo de sus propios robots, fomentando en ellos habilidades como imaginación, el trabajo en equipo y el aprendizaje práctico. Con este tipo de aprendizaje, se crea en los alumnos interés por la búsqueda en su robot de una función válida y útil para ayudar al ser humano en sus labores diarias.

- Proyecto Lego MindStorm: es una línea de productos de la empresa Lego fundamentada en la construcción de modelos siguiendo el principio de Lego de unir piezas pero incorporando el factor de movilidad a estas, lo que permite a los usuarios simular modelos robóticos de la vida real, como por ejemplo elevadores o plantas de producción automatizadas entre otros. El usuario construye el robot deseado y gracias al software incluido en el paquete, indica a la construcción las órdenes a ejecutar, controlando todo el proceso desde una interfaz de control remoto a través de un dispositivo móvil smartphone o Tablet. En la Figura 6 se muestra el módulo principal de control y diversos módulos conectados a él:



*Figura 6. Proyecto Lego Mindstorm. En el centro módulo principal de control y alrededor diversos módulos*

Es la alternativa que más similitudes posee con el proyecto que se desarrolla en este Trabajo Fin de Grado, ya que sigue los mismos principios de conexión y programación del robot, con la gran diferencia del coste económico y de la poca compatibilidad con módulos de sensores ya existentes en el mercado. Está recomendado para niños a partir de 9 años, por lo que no requiere conocimientos previos de robótica.

- Proyecto Scratch: Es un lenguaje de programación que permite el desarrollo de habilidades mentales enfocadas a la programación sin tener conocimientos profundos sobre código. Es muy difundido en la educación de niños, adolescentes y adultos debido a la facilidad de uso y la rapidez a la hora de visualizar el resultado final. Suele ser el primer contacto con la programación en los alumnos ya que además de programar, les permite crear historias animadas. Este proyecto es utilizado con fines pedagógicos como apoyo para conceptos complejos de otras materias.

Como se ha mencionado anteriormente, Scratch es muy similar al método de programación que se empleará en el proyecto que se aborda en este trabajo.

- Proyecto Arduino: Arduino es el hardware por excelencia utilizado para la introducir a los alumnos de todas las edades e incluso a adultos en el mundo de la robótica debido a su reducido coste económico, su facilidad de manejo y programación. Es el proyecto más utilizado en las aulas, debido a la gran comunidad que abriga esta plataforma.

Arduino es utilizado para el desarrollo de proyectos más técnicos y avanzados que los casos planteados con anterioridad. Gracias a su gran versatilidad y compatibilidad con módulos de sensores y actuadores disponibles en el mercado, Arduino posibilita a los usuarios un mayor abanico y libertad para crear su proyecto, con la desventaja de requerir un mayor conocimiento de electrónica y robótica. Este proyecto está enfocado para alumnos a partir de 12 años.

Todos los proyectos mencionados con anterioridad se planean siguiendo una metodología común. A continuación en la Figura 7 se muestra un diagrama de flujo con dicha metodología establecida por un equipo de la Universidad de Texas [5]:

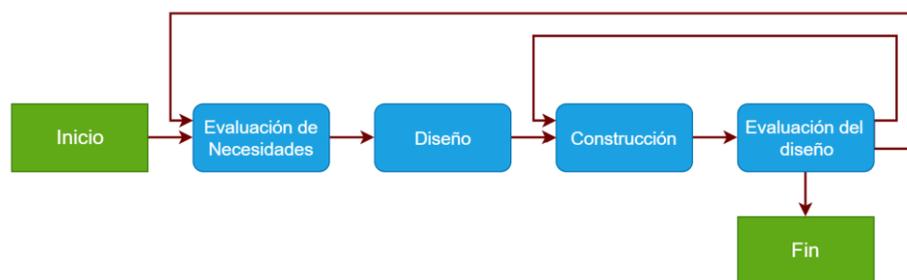


Figura 7. Diagrama de flujo de la metodología a seguir para la construcción de un robot.

Este diagrama de flujo nos muestra las diferentes partes en el proceso de planteamiento de cualquier proyecto.

En primer lugar, el proyecto se lleva a cabo para cubrir unas necesidades que es preciso tener presentes durante todo el proceso. En segundo lugar, se avanza a la etapa de diseño, donde el alumno en este caso debe desarrollar a partir de los conocimientos que él posee como construir y fabricar el proyecto en cuestión. A continuación se lleva a cabo la construcción de este usando como base la etapa de diseño anterior. Y por último se evalúa dicho proyecto. Dependiendo si cubre o no las necesidades de la primera etapa, se da por finalizado el proyecto, o se rectifica su construcción.

## 2.6. Motivación

La robótica en la educación es un campo que tiene un gran recorrido en el futuro. Teniendo en cuenta la evolución que está propiciando la tecnología en todos los ámbitos diarios es fundamental introducir su enseñanza en el plan de estudios de las nuevas generaciones. Si a este hecho le añadimos la situación medioambiental en la que se encuentra nuestro planeta y su constante y negativa evolución, podemos ser conscientes también de la necesidad de educar a las nuevas generaciones en este ámbito.

Por ello, este proyecto reúne dos disciplinas, Robótica Educativa y Educación Medioambiental, que en la actualidad pasan desapercibidas en los planes de estudio, pero que en próximos años deben ser primordiales en la educación de los alumnos. Por un lado, los alumnos adquieren conocimientos de programación y robótica, es decir, el aprendizaje de un nuevo lenguaje, como el matemático o el musical, que pueden y deben utilizar para resolver problemas del día a día.

Por otro lado, adquieren un conocimiento de sensibilización, responsabilidad y concienciación que permita a las nuevas generaciones comenzar a modificar sus hábitos diarios con la finalidad de frenar al cambio climático que actualmente sufre nuestro planeta.

La principal diferencia de este proyecto con otros similares se basa en las mejoras que posee, que a su vez son la superación de las barreras de entrada que tiene impuestas la robótica educativa en la actualidad:

- Ahorro económico.
- Facilidad de aplicación, formación y uso.
- Conjugación de dos disciplinas.

## 2.7. Justificación

Como se ha comentado anteriormente, la tecnología forma parte de la vida de las nuevas generaciones desde que comienzan a tener uso de razón y a veces incluso antes. Por ello el desarrollo de habilidades relacionadas con este campo en la educación es primordial.

No obstante, los centros educativos no son capaces de evolucionar a la hora de impartir estos conocimientos debido a las elevadas barreras de entrada:

- Falta de presupuesto económico en la educación: la mayoría de los proyectos de instauración de la robótica educativa son costosos tanto los bienes materiales, como la formación para el profesorado y los alumnos.
- Falta de personal cualificado: el profesorado presente en los centros escolares no dispone de la formación necesaria para inculcar en los alumnos este tipo de conocimientos, por lo que la introducción de ellos en las aulas conlleva una formación previa para el profesorado.
- Falta directrices sobre la materia a impartir: no existen guías ni materias concretas con las que poder introducir esta metodología.
- Desinformación acerca de las ventajas que puede implicar la enseñanza de estos conocimientos.
- Inseguridad debido al uso de infraestructuras electrónicas ya existentes por el alumnado debido a los elevados costes económicos y de reparación de estas.

Por ello, se ha desarrollado este proyecto, con la idea facilitar en los centros educativos la posibilidad de evolucionar en la enseñanza del ámbito tecnológico a través de la Robótica Educativa, intentando eliminar todas estas barreras de entrada.

En cuanto a la temática secundaria del proyecto, el medio ambiente y su cuidado es uno de los temas de mayor repercusión en la actualidad. La idea de este Trabajo Fin Grado relacionada con esta temática es acercar a los alumnos conceptos simples como el ciclo de crecimiento de las plantas y la importancia de su cuidado para conseguir un crecimiento controlado, seguro y sano. Con este proyecto, se pretende concienciar entre los más pequeños y cada vez antes para corregir malos hábitos y comportamientos que están corrompiendo el planeta en que vivimos.

### 3. NORMATIVA DE IMPLMETACIÓN

El actual Trabajo Fin de Grado estudia el ámbito de aplicación de la enseñanza de dichos conocimientos siguiendo el *Decreto 108/2014, de 4 de Julio, del Consell, por el que establece el currículo y desarrolla la ordenación general de la educación primaria en la Comunitat Valenciana* [6].

La Ley Orgánica 8/2013 establece en su artículo 6.bis la estructura de las asignaturas en tres bloques, asignaturas troncales, asignaturas específicas, y asignaturas de libre configuración autonómica.

Según el artículo 8.1 del *Real Decreto 126/2015*, la etapa de educación primaria comprende seis cursos cursados entre los 6 y los 12 años, organizados en áreas con un carácter global e integrador.

El presente proyecto está enfocado para alumnos de los dos últimos años de estos cursos académicos, es decir, de 10 a 12 años, lo que corresponde con el último ciclo de Educación Primaria.

Esta enseñanza comprende como asignaturas troncales que deben cursarse en los 6 cursos las siguientes:

- Ciencias de la Naturaleza.
- Ciencias Sociales.
- Lengua Castellana y Literatura.
- Matemáticas.
- Primera Lengua extranjera.

Bloque de asignaturas específicas:

- Educación Física.
- Religión o Valores Sociales y Cívicos a elección de los padres, madres o tutores legales.
- Educación Artística, cada centro gestiona el tiempo lectivo para la educación musical y la educación plástica.

Bloque de asignaturas de libre configuración autonómica, en la comunidad valenciana, conforme al artículo 8.4 del Real decreto 126/3024, los alumnos deben cursar el área de Valenciano: Lengua y Literatura, y en quinto curso, el área de Cultura Valenciana [6].

A continuación se explica la forma de introducir los conocimientos básicos que engloba este proyecto en el plan de estudios de los dos últimos cursos de Educación Primaria:

- Educación Medioambiental: la parte de este proyecto correspondiente al ámbito de la naturaleza, englobada dentro de la Educación Medioambiental es un conocimiento que en la actualidad se está inculcando en los alumnos de forma leve gracias a la asignatura Ciencias de la Naturaleza.

Esta asignatura se divide en 5 bloques, siendo el tercero de ellos “*Los seres vivos*”. Durante el desarrollo de este bloque el alumno refuerza y amplía conocimientos que ya adquirió en los anteriores 4 cursos relacionados con los ecosistemas, tanto terrestres como acuáticos. Adquiere conciencia sobre las principales causas de extinción de especies en ambos ecosistemas así como conocimientos sobre el medio ambiente y su conservación desde una perspectiva más práctica, realizando excursiones y salidas a la naturaleza.

Introduciendo estos conocimientos a los alumnos, estos aprenden a identificar los diferentes tipos de ecosistema y toman conciencia de la importancia del cuidado de las especies para evitar la extinción de estas.

En lo referente a este proyecto, en la asignatura de Ciencias de la Naturaleza, los alumnos adquieren valores para la conservación y mejora del medio ambiente, identificando las acciones realizadas por el ser humano positivas o negativas para este.

El desarrollo de este proyecto crea en los centros la posibilidad de experimentar en primera persona los valores que se le han inculcado a los alumnos. De forma tangible, estos, pueden desarrollar un proyecto a través del cual realizan acciones positivas para el cuidado del medio ambiente y los seres vivos enfatizando así los conocimientos vistos de forma teórica en el aula.

- Robótica Educativa: la parte principal de este proyecto, correspondiente al desarrollo de la programación y el montaje de hardware, son conocimientos que hoy en día no están englobados dentro del plan de estudios en estos cursos de Educación Primaria. Por ello, se proponen las siguientes soluciones:
  - o En el primer curso de instauración (5º Educación Primaria), el proyecto se desarrollará dentro del temario de Ciencias de la Naturaleza durante la Semana Europea de la robótica. En este curso, el bloque 5 de esta asignatura, denominado *“La tecnología, objetos y máquinas”* incluye contenidos sobre *“clasificación y resolución de problemas técnicos de aplicaciones y dispositivos de las Tecnologías de la Información y la Comunicación (TIC) sencillos de forma guiada”*, así como *“los beneficios y la importancia de la tecnología en la sociedad, en la escuela, en el hogar, en el trabajo, en la cultura y en el ocio”*. Dentro de este bloque se desarrollará la introducción de los primeros conocimientos sobre programación y robótica educativa. Los alumnos aprenden las posibilidades que las TIC ofrecen para afrontar problemas del día a día, en este caso el cuidado del medio ambiente de forma guiada y comienzan a ser conscientes de los beneficios de estas para en la sociedad, y los factores positivos que estas aportan en su aplicación en el medio ambiente.
  - o En el segundo curso en el que se instaura este proyecto (6º Educación Primaria), los alumnos adquieren los conocimientos relacionados con este proyectos a través de la asignatura Ciencias de la Naturaleza. Este curso contiene el mismo bloque que el anterior pero en mucha más profundidad (*“La tecnología objetos y máquinas”*). Se subdividen las TIC, profundizando en conceptos más concretos y cercanos, acercándose más a los necesarios para el desarrollo del proyecto en cuestión. Los alumnos toman contacto por primera vez con los siguientes temas: *“Circuitos electrónicos y sus componentes”*, *“Planificación, diseño y montaje de un circuito eléctrico y de un electroimán”*, *“Relación entre electricidad y magnetismo”*, *“Conductores y aislantes”*, *“Electricidad: la corriente eléctrica. Electroimán”*, *“Efectos de la electricidad”*. El conocimiento sobre estos temas es muy ventajoso para el correcto desarrollo del proyecto. A su vez, dicho proyecto complementará todos los conocimientos teóricos impartidos en el aula.

Todos estos temas podrán ser complementados y reforzados con la introducción de la robótica educativa a través de las siguientes ideas:

*“Historia de la robótica”*, *“Pensamiento computacional”*, *“Robots y las Tecnologías de la Información y la comunicación”*.

Tanto los conceptos sobre Educación Medioambiental como los conceptos de Robótica Educativa están presentes en la asignatura Ciencias de la Naturaleza en los dos cursos del último ciclo de Educación Primaria. Por ello este proyecto es válido en los dos ámbitos: Por un lado refuerza los contenidos del bloque 3 de la asignatura *“Los seres vivos”* gracias a la elección del invernadero como ámbito de aplicación del proyecto, y por otro lado, ayuda a desarrollar los conceptos pertenecientes al tema 5 *“La tecnología, objetos y máquinas”*, a través de la implementación de la parte de programación y hardware en el proyecto.

Como vía alternativa para la implementación del concepto Robótica Educativa en las aulas existe la posibilidad de impartir estos conocimientos en asignaturas pertenecientes al bloque de libre configuración autonómica.

Actualmente, existen más de 25 colegios de otras comunidades autónomas, como por ejemplo Madrid que poseen los permisos necesarios para impartir una asignatura que engloba todos los conocimientos de Robótica Educativa además de trabajar y ser útil de forma transversal en las demás asignaturas. Esta asignatura denominada *“Tecnología y recursos digitales para la mejora del aprendizaje”*, forma parte del bloque de asignaturas de libre configuración autonómica [7].

El horario establecido para las asignaturas de libre configuración autonómica es de 5 sesiones semanales de 45 minutos cada una en todos los cursos de Educación Primaria. Los contenidos de esta asignatura permiten al alumno adquirir conocimientos tecnológicos necesarios para las nuevas generaciones: por un lado, la asignatura incluye temas de recogida y búsqueda de información con el uso de las TIC, además de conocimiento sobre el correcto uso del correo electrónico. Por otro lado, englobando los contenidos del proyecto que se desarrolla en este Trabajo Fin de Grado, en el plan de estudios se engloban temas de *“Fundamentos de programación y creación de pequeños programas informáticos”* y de *“Utilización de equipos electrónicos e informáticos”* [6].

Gracias a esta asignatura, los conocimientos necesarios para la introducción de este proyecto en la enseñanza primaria figuran englobados dentro del plan de estudios de este nivel de enseñanza, haciendo posible su implementación.

Por otro lado, todas las competencias englobadas en el conjunto de *“Competencias matemáticas y competencias básicas en ciencia y tecnología”* también se ven reforzadas gracias a la realización de este proyecto.



## 4. DESARROLLO DEL PROYECTO

Partiendo del planteamiento anteriormente expuesto, se comienza a desarrollar el proyecto en su grueso, dividiendo el proyecto en 6 grupos diferenciados atendiendo al “Diseño y Control de un Invernadero Automatizado de bajo coste adaptado para su uso en Educación Primaria”.

### 4.1. Hardware. Easy-Plug KeyStudio

#### 4.1.1. Placa principal

El proyecto en cuestión se ha comenzado a desarrollar utilizando una placa ya existente *MainBoard Easy-Plug V2.0* del fabricante KeyStudio mostrada en la Figura 8.

La elección de esta placa se ha llevado a cabo tras un estudio de las posibilidades y ventajas que ofrece respecto a otros competidores, entre las que se encuentran la facilidad de montaje y conexión de los módulos externos, la robustez, el coste económico, el microcontrolador que contiene y por consiguiente el programa utilizado para programar el microcontrolador.

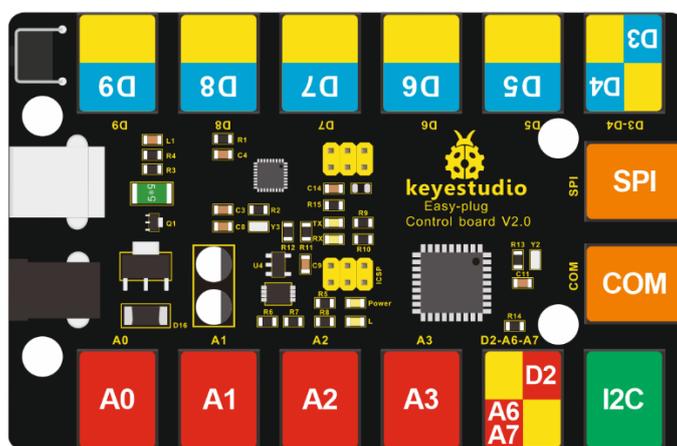


Figura 8. Placa Principal Easy-Plug V2.0 de KeyStudio.

La placa *MainBoard Easy-Plug* de KeyStudio, está constituida por un microcontrolador ATmega328, puertos de entrada analógica y de salida y entrada digital, puertos de comunicación, conector USB, un Jack de alimentación y un botón de reseteo.

- Microcontrolador ATmega328: dicho procesador es el mismo que incluye la placa Arduino UNO. Está fabricado por la empresa Atmel y pertenece a la serie megaAVR. Entre sus especificaciones destaca que es un circuito integrado de 8 bits con una memoria flash de 32 Kbyte y 2 Kbyte de SRAM. La frecuencia máxima de operación es 20 MHz. La tensión de operación es de 1,8 a 5 Voltios. Es un microcontrolador simple, de bajo coste y de bajo consumo<sup>1</sup>.

Este microcontrolador cuenta con 23 líneas de entrada salida de propósito general, aunque en su implementación en la placa *Easy-Plug* solo se utilizan 14 de ellos.

A continuación se muestra en la Figura 9 el *Pin-Out* de dicho microcontrolador con la posición de cada una de las líneas así como los pines de alimentación, comunicación y reseteo.

<sup>1</sup> Datasheet Summary Atmega328/P. Atmel Corporation, 11/2016.  
[http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf)

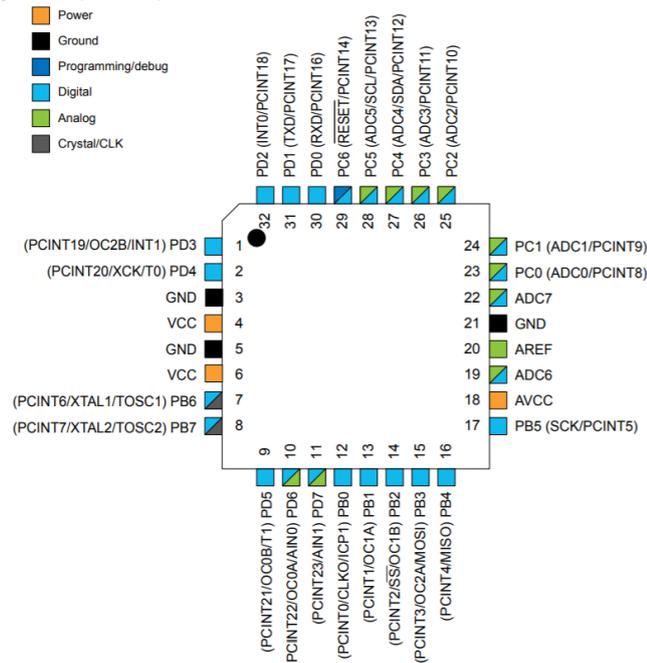


Figura 9. Pin-Out del microcontrolador ATmega328 extraída de su datasheet<sup>2</sup>

- Puertos de entrada/salida: dichos puertos, se caracterizan por que las conexiones a ellos se realizan mediante tecnología de conexión *Easy Plug*.  
Cuenta con un total de catorce puertos con las siguientes características:
  - o Cinco puertos simples de entrada/salida digital, capaces de leer o actuar con señales digitales. Las señales digitales son señales booleanas, es decir, que solo pueden tomar dos valores o estados, abierto o cerrado, encendido o apagado, activo o inactivo, alto o bajo o 1 ó 0, lo que correspondería con valores binarios. En la Figura 10 se muestra la representación de estos puertos en la placa *Easy-Plug*.



Figura 10. Puertos simples digitales. Extraídos de una imagen de la placa principal

Como se observa en la anterior figura, los puertos están marcados con etiquetas de color amarillo (indicando que son puertos de entrada digital) y azul (indicando que son puertos de salida digital) y nombrados desde D5 hasta D9.

Tres de estos cinco puertos, D5, D6 y D9 son de señal de modulación por ancho de pulsos o PWM, es una señal que permite modificar el ciclo de trabajo para transmitir información o para controlar la cantidad de energía enviada. Algunas de sus aplicaciones reales que permiten desarrollar estos puertos son el control de la velocidad de giro de un motor o la comunicación analógica con sistemas digitales.

<sup>2</sup> Datasheet Summary Atmega328P. Atmel Corporation, 11/2016. [http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf)

- Un puerto doble de entrada/salida digital, con las mismas características que los puertos anteriormente explicados, pero destinado para la conexión de la placa a módulos que requieran dos puertos digitales, como por ejemplo un módulo display de cuatro dígitos de siete segmentos. En la Figura 11 se muestra la representación de estos puertos en la placa *Easy-Plug*.



Figura 11. Puerto doble digital. Extraído de una imagen de la placa principal

Este puerto está marcado con una etiqueta que muestra el nombre de los pines, D3 y D4, siguiendo el mismo código de colores que en los puertos simples digitales.

- Cuatro puertos de entrada analógicos, utilizados para la lectura de sensores que captan señales analógicas, es decir, aquellas que captan variaciones de valores de tensión. Para poder trabajar con dichas señales, se convierte digital utilizando la resolución de dichos pines. En el caso de este microcontrolador, la resolución de los canales analógicos es de 10 bits, y como las señales varían entre 0 y 5V, cuando el sensor capte una tensión de 0 Voltios, corresponderá con el valor digital 0, y cuando el sensor capte una tensión de 5 Voltios, corresponderá con el valor digital 1023. En la Figura 12 se muestra la representación de estos puertos en la placa *Easy-Plug*.

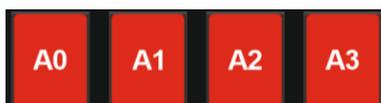


Figura 12. Puertos de entrada analógicos. Extraídos de una imagen de la placa principal

Como se puede observar en la anterior figura, dichos están marcados con una etiqueta de color rojo que indica que son puertos de entrada analógica, y están nombrados desde el A0 hasta el A3.

- Un puerto con dos pines analógicos y uno digital, para interfaz Joystick. Los pines analógicos permiten el control de las coordenadas X e Y, y el pin digital es utilizado para leer el valor de un interruptor controlado desde este mismo módulo. En la placa *Easy-Plug*, este puerto tiene el aspecto mostrado en la Figura 13.



Figura 13. Puerto para interfaz Joystick. Extraídos de una imagen de la placa principal.

Los canales analógicos etiquetados en la placa son A6 y A7, y el canal digital es D2. La etiqueta es de color rojo, indicando que son puertos analógicos y de color amarillo, indicando que el puerto digital es de entrada.

- Puertos de comunicación:

- Un puerto de comunicación SPI. Es un bus maestro-esclavo, donde la comunicación se realiza utilizando dos líneas independientes, lo que permite al maestro el envío y la recepción simultánea de datos. En la Figura 14 se muestra el aspecto de este puerto en la placa *Easy-Plug*.



*Figura 14. Puerto de comunicación SPI. Extraído de una imagen de la placa principal.*

Este puerto está identificado con una etiqueta de color naranja indicando que es un puerto de comunicación y el rótulo del nombre SPI.

- Un puerto COM Puerto serie: Este puerto engloba los pines 13, 12 y 11 del microcontrolador junto con el pin Reset y alimentación y masa. Una de las finalidades de este puerto es la programación del bootloader del microcontrolador a través del puerto serie sin necesidad de un programador externo. El bootloader de Arduino será explicado más adelante (apartado 39 del actual Trabajo Fin de Grado). El aspecto de este puerto se muestra en la Figura 15. Figura 16. Puerto de comunicación I2C. Extraído de una imagen de la placa principal.



*Figura 15. Puerto de Comunicación serie. Extraído de una imagen de la placa principal.*

- Un puerto de comunicación I2C, el cual solo requiere de dos pines, uno para la señal de reloj y otro para el envío de datos. Es un bus de comunicación que tiene una arquitectura maestro-esclavo, pero no permite el envío y recepción simultanea de datos. Este bus de comunicación es síncrono, por lo que todos los dispositivos se mantienen sincronizados. Algunas de sus aplicaciones son la conexión de la placa a módulos Bluetooth o a pantallas OLED. En la Figura 16 se muestra al aspecto de este puerto en la placa *Easy-Plug*.



*Figura 16. Puerto de comunicación I2C. Extraído de una imagen de la placa principal.*

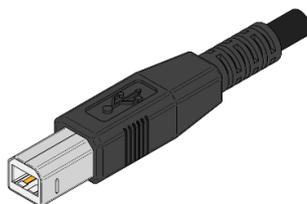
Este puerto se identificado con una etiqueta de color verde rotulada con el texto I2C para su identificación.

Todos los puertos anteriormente mencionados comunican la placa a través de un conector RJ-11 desplazado con los módulos externos. Estos conectores, se encuentran soldados a la placa mediante los 6 pines que los conforman.

En el caso de los puertos de entradas/salidas digitales y las entradas analógicas, de los seis pines que contiene el conector, solo se usan 3 de ellos. Un pin es la alimentación del módulo externo, otro pin la masa y el tercer pin es el de comunicación mediante el cual se envía y recibe la información.

En el caso del resto de puertos, todos tienen un pin para la alimentación y otro para la masa; los demás pines varían dependiendo de la función de cada puerto.

- Conector USB, el cual permite crear un puerto serie virtual para conectar el ordenador con el microprocesador. También realiza la función de alimentación. Este conector es de tipo B, por lo que cuenta con 4 pines, dos de ellos para la transmisión de datos y los otros dos para alimentación (5 Voltios Corriente Continua y masa). Este conector se muestra en la Figura 17.



*Figura 17. Conector USB tipo B utilizado para la conexión y alimentación de la placa principal.*

Dicho conector se puede utilizar con dos funcionalidades diferentes:

- o Conector USB para comunicación Placa-Ordenador: Los pines de datos transfieren la información utilizando lenguaje binario de 0 y 1. Estos dos valores son identificados por el microprocesador como una diferencia de voltaje,  $D+$  o  $D-$  en función de la tensión (0,4 V o -0,4 V).
  - o Conector USB como alimentación: si se utiliza este conector para alimentar la placa, hay que tener en cuenta que la entrada de tensión máxima son 5 Voltios. La corriente máxima de entrada queda limitada a 500 mA por la acción del fusible presente en el puerto USB.
- Conector tipo Jack de alimentación, permite alimentar la placa a través de un adaptador de corriente alterna a corriente continua. El polo positivo corresponde con el pin central del conector. Esta entrada acepta tensiones de 7 a 12 voltios. La placa incluye un diodo de protección para el caso en que se produzca un error de polaridad al alimentarla. Este conector tiene el aspecto que se muestra en la Figura 18.



*Figura 18. Conector tipo Jack para alimentación de la placa principal entre 7 y 12 Voltios.*

- Botón de reseteo, tiene la función de ejecutar el Bootloader de Arduino, eliminando el programa que esté alojado en la memoria flash de este, dejando espacio para albergar el nuevo programa. En el caso en que el microprocesador no detecte ningún programa nuevo, es decir, que la placa no se encuentre conectada al ordenador a través del puerto USB, este botón simplemente reiniciará el programa que el microcontrolador contenga en la memoria Flash en ese instante. En la Figura 19 se muestra una imagen del botón de reseteo en la placa *Easy-Plug*.



Figura 19. Botón de reseteo de la placa principal

#### 4.1.2. Conexiones

Todos estos puertos, permiten su conexión a los módulos a través de conectores *Easy-Plug*. Estos conectores, son puertos de conexión RJ-11, normalmente utilizados en las redes de telefonía, pero en su variante 6P6C para tres líneas, es decir, 6 posiciones 6 conectores. Las posiciones son el número máximo de contactos que pueden utilizarse y los conectores son el número de contactos utilizados. En la Figura 20 y Figura 21 se muestra el cabezal de dicho cable y el aspecto completo del mismo.

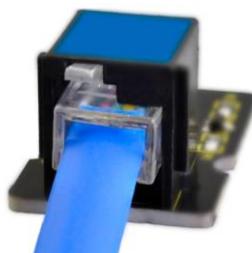


Figura 20. Conector RJ11 6P6C Easy-Plug con clavija desplazada.



Figura 21. Cable de conexión Easy-Plug con conector RJ11.

El cabezal del conector posee un pequeño clip de plástico en la parte superior, desplazado ligeramente hacia la izquierda, evitando así cualquier confusión que pueda surgir en los alumnos sobre la posición de conexión.

El uso de esta tecnología de conexión puede evitar algunos de los principales problemas que surgen a la hora de introducir este tipo de hardware en la robótica educativa:

- Soldaduras con estaño: Es el principal problema de los montajes de hardware en proyectos de robótica educativa, siendo uno de los inconvenientes por los que en muchos casos se rechaza la formación de dichos conocimientos en las aulas:
  - Su grado de peligrosidad derivado de las elevadas temperaturas que alcanzan los soldadores en sus extremos, entre los 200° C y los 400° C.
  - Gases nocivos fácilmente inhalables por los alumnos durante el proceso de soldadura, producidos al fundir dicho metal.
  - Complejidad para su uso en alumnos con conocimientos nulos y edades tempranas, debido a la precisión necesaria para conseguir una soldadura fiable y no acarrear errores en el montaje que puedan ocasionar problemas en las siguientes fases del proyecto.
  - Coste económico, se precisa a los centros de disponer de estos equipos de soldadura y en su defecto, si concierne, disponer de extractores de humo encargados de absorber los gases nocivos expulsados durante el proceso de soldadura.

- Placas auxiliares. Los proyectos en los que se utilicen placas con tecnología de conexión común, es necesario y aconsejable en el ámbito educativo, disponer de placas auxiliares que faciliten las conexiones entre la placa principal y los módulos externos. La necesidad de disponer de estas placas acarrea costes económicos, al tener que disponer de ellas, así como pérdida de tiempo al tener que formar a los alumnos sobre su correcto uso.
- Falta de tiempo en los planes de estudio. La robótica educativa en la actualidad no es una materia contemplada en el plan de estudios, por lo que si se logra la formación a los alumnos en este campo, será en jornadas de tiempo muy reducida, en las que el uso de placas que no dispongan de la tecnología *Easy-Plug* retrasaría mucho el desarrollo del proyecto.
- Poco conocimiento en la materia, al ser un campo que está aún por instaurar en el plan de estudios como se ha dicho anteriormente, existe poco conocimiento por parte del profesorado y de los alumnos, por lo que el uso de esta tecnología de conexión presente en la placa facilitaría mucho la labor de formación del profesorado y por consiguiente de los alumnos.
- Conexiones erróneas o de mala calidad. Al utilizar placas auxiliares, con facilidad se producen conexiones erróneas que pueden dañar componentes de la placa, módulos exteriores o el microprocesador. Son frecuentes también las conexiones de mala calidad, ya que con el uso prolongado de las placas auxiliares, las conexiones no son tan fuertes, dando lugar a desconexiones frecuentes o intermitentes de los módulos.

Esta tecnología de conexión *Easy-Plug* además de evitar problemas durante la introducción de este proyecto en la robótica educativa, crea algunas ventajas en el proyecto final. La conexión de todos los módulos de comunicación y de los sensores a través de estas conexiones, forman un hardware completo compacto, permitiendo la movilidad del proyecto sin necesidad de realizar la desconexión de los módulos para su transporte.

#### 4.1.3. Sensores y módulos de comunicación externos

Para el desarrollo del proyecto de Invernadero Automatizado, se han empleado módulos de sensores y actuadores compatibles con la placa principal siguiendo el mismo principio de conexión *Easy-Plug* y del mismo fabricante, KeyStudio. Dicho módulos se explican a continuación:

- Módulos de entrada digital. Todos ellos están identificados con etiquetas de color amarillo para facilitar la conexión de estos a los puertos correctos de la placa *Easy-Plug*:
  - o Botón: Este módulo de entrada digital permite introducir un valor alto o 1 digital en el momento que esté pulsado y un valor bajo o 0 en el caso que no esté pulsado. El módulo en cuestión consta de un botón y una resistencia en configuración pull-Down con un valor de 10 K $\Omega$ . El aspecto del módulo de botón se muestra en la Figura 22.

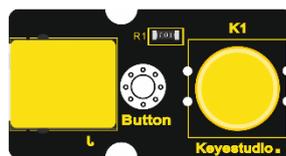


Figura 22. Módulo *Easy-Plug* botón.

- Sensor de humedad y temperatura ambiente: Este módulo es un sensor de entrada digital. En concreto se trata del sensor DHT11, capaz de captar tanto temperatura como humedad del medio. La conexión con la placa principal se realiza a través de tres pines: alimentación, masa y datos.

Para la temperatura, dicho módulo cuenta con una resistencia NTC (*Negative Temperature Coefficient*), basada en la variación de la resistividad de un semiconductor en función de la temperatura. Su principio de funcionamiento es a mayor temperatura menor resistividad.

Para la humedad, el módulo DHT-11 contiene un sensor de humedad HR202. Su principio de funcionamiento se basa en las variaciones de conductividad que experimenta dependiendo del grado de humedad de la zona. A mayor humedad, mayor conductividad.

Este sensor, incorpora una memoria OTP (*one-time-programmable*) con los coeficientes de medición grabados durante el proceso de calibración<sup>3</sup>.

La comunicación de la placa principal con el sensor se lleva a cabo a través de un único pin. Cada periodo de comunicación tiene una duración de 40 milisegundos y un total de 40 bits de información con el formato mostrado en la Figura 23.

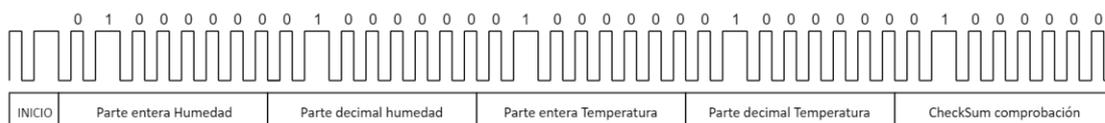


Figura 23. Formato de comunicación sensor DHT11.

La placa modular cuenta con una resistencia en configuración pull-Up con un valor de 4,7 K $\Omega$  cuya finalidad es evitar el ruido y las posibles medidas erróneas. Además, contiene un condensador de filtro para rectificar las variaciones de la tensión de alimentación. Ambos componentes y el conjunto del módulo se pueden observar en la Figura 24.

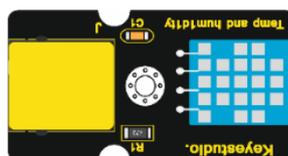


Figura 24. Módulo Easy-Plug sensor de temperatura y humedad ambiente DHT11.

- Módulos de salida digital. Todos ellos están identificados con etiquetas de color azul para facilitar la conexión de estos a los puertos correctos de la placa *Easy-Plug*:
  - Relé: Este módulo consta de un dispositivo electromagnético, permitiendo a la placa principal controlar cargas de una tensión e intensidad muy superiores a las que puede soportar su electrónica.  
Este dispositivo está formado por dos circuitos:
    - Primario: incluye la parte de electrónica de baja tensión, es decir, la placa principal. Es la parte del módulo encargada de recibir la señal de activación o desactivación del propio relé. Contiene la bobina.

<sup>3</sup> Datasheet Summary DHT11, D- Robotics UK. <https://www.micro4you.com/files/sensor/DHT11.pdf>

- Secundario: incluye la parte mecánica, equivalente a un interruptor, encargada de cerrar o abrir el circuito en función de la señal recibida por el circuito primario, encendiendo o apagando la carga. Dispone de tres contactos NO (*Normally Open* o Normalmente Abierto), C (*Common* o Común) y NC (*Normally Close* o Normalmente cerrado)<sup>4</sup>. En la Figura 24 se muestra el módulo de relé con las indicaciones de las diferentes partes que lo componen.

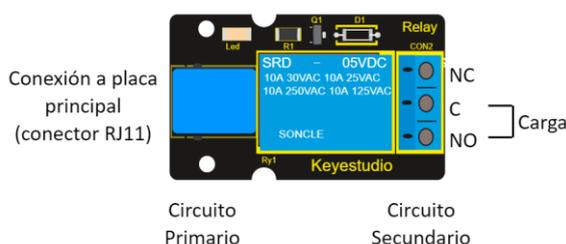


Figura 25. Módulo Easy-Plug relé con indicaciones de conexión.

Su funcionamiento se basa en la recepción de la señal digital de la placa principal. Esta señal crea una corriente en la bobina, generando un campo magnético que desencadena la conmutación de los contactos del circuito secundario. En el momento que el relé recibe una señal digital de nivel alto, los contactos del circuito secundario pivotan creando contacto entre NO-C. Cuando la señal de nivel alto cesa, los contactos del relé vuelven a su posición de contacto (NC-C).

La conexión de este módulo con la placa principal se realiza a través de tres pines, alimentación, masa y pin de datos.

El módulo *Easy-Plug* incluye un transistor debido a la insuficiente corriente de salida de la placa principal, inferior a la necesaria para el funcionamiento del dispositivo. Además, incorpora un optoacoplador y un diodo flyback para proteger a la placa principal de cualquier fallo que pueda producirse en el relé. Destaca por su bajo consumo (0,36W) [8].

- Diodo Led: Este módulo está formado por un diodo led cuya activación se realiza con la recepción de un nivel alto ("HIGH") de la placa principal. Este módulo contiene también una resistencia en configuración pull-Up con un valor de  $470 \Omega$  para proteger el diodo, un condensador de filtro para rectificar las variaciones de la tensión de alimentación y un transistor para controlar la corriente de entrada. En la Figura 26 se puede observar el módulo completo.

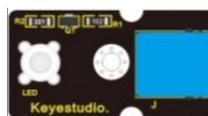


Figura 26. Módulo Easy-Plug Diodo Led.

<sup>4</sup> Datasheet Summary Single Relay SRD-05VDC-SL-C. <http://www.circuitbasics.com/wp-content/uploads/2015/11/SRD-05VDC-SL-C-Datasheet.pdf>

- Módulo de conversión *Easy-Plug*: Este módulo será utilizado de forma opcional en el proyecto. Permite la conexión de módulos convencionales que no dispongan la tecnología *Easy-Plug* a la placa principal.

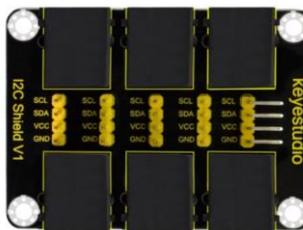


Figura 27. Módulo de conversión *Easy-Plug* a conexión convencional.

- Módulos de entrada analógica. Todos ellos están identificados con etiquetas de color rojo para facilitar la conexión de estos a los puertos correctos de la placa *Easy-Plug*:
  - Sensor de luminosidad: Este módulo está formado por una resistencia variable dependiente de la cantidad de luz del entorno, LDR (*Light Dependent Resistor*). Su principio de funcionamiento es a menor cantidad de luz, mayor resistencia, y a mayor cantidad de luz, menor resistencia. Este componente presenta una resistencia mínima a la luz de 20 K $\Omega$  y máxima de 100 K $\Omega$ . En condiciones de oscuridad, presenta una resistencia mínima de 20 M $\Omega$ <sup>5</sup>.

La conexión a la placa principal es a través de un puerto analógico, por tanto, para ser procesado por el microprocesador, se transforma esta señal analógica a digital (como se ha comentado en la explicación de los puertos analógicos en el apartado 4.1.1), devolviendo así valores entre 0 a 1023.

El módulo está formado por la resistencia variable, que es una célula de sulfuro de cadmio encapsulada con una resina epoxi resistente a la humedad, una resistencia en configuración pull-Up con un valor de 4,7 K $\Omega$  para evitar ruido en las medidas y un condensador de filtro para rectificar las variaciones de la tensión de alimentación. En la Figura 28 se puede observar el módulo completo.

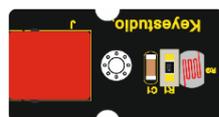


Figura 28. Módulo *Easy-Plug* sensor de luminosidad LDR.

- Higrómetro o sensor de humedad del terreno: permite conocer el nivel de humedad de un suelo en función de la variación de su conductividad. Su principio de funcionamiento es a mayor humedad, es capaz de conducir la electricidad mejor, por tanto ofrece menos resistencia, y a menor humedad, conduce peor la electricidad, por lo que ofrece mayor resistencia. El sensor estará conectado a un puerto analógico, por tanto, para ser procesado por el microprocesador, se transforma esta señal analógica a digital (como se ha comentado en el apartado de los puertos analógicos de la placa principal), devolviendo así valores de 0 a 1023.

<sup>5</sup> Datasheet Summary LDR NSL-19M52. Silonex, Inc. <https://docs-emea.rs-online.com/webdocs/002e/0900766b8002e0d5.pdf>

- La Figura 29 muestra el conjunto de todos los componentes que conforman este módulo.



Figura 29. Módulo Easy-Plug Higrómetro o sensor de humedad del terreno.

- Módulo de comunicación I2C. Este módulo está identificado con una etiqueta de color verde para facilitar su conexión al puerto correcto de la placa Easy-Plug:

- Pantalla OLED: este módulo permite la comunicación y el intercambio de datos entre el usuario y la placa principal. Es una pantalla de 0.96" y 128 x 64 pixels monocromática. Esta pantalla funciona sin luz de fondo, es decir, los pixeles en los que no se muestra información están apagados, lo que dota a esta pantalla de un mayor contraste y un consumo menor que una pantalla LCD.

El uso principal de esta pantalla es mostrar texto al usuario. Para facilitar su programación, la pantalla se divide en 8 filas (*pages*) de 8 pixeles de altura y cada fila tiene 127 columnas. De esta manera, un byte de información configura el estado de una de las 127 columnas de una fila. La distribución de las zonas de la pantalla se puede observar en la Figura 30.

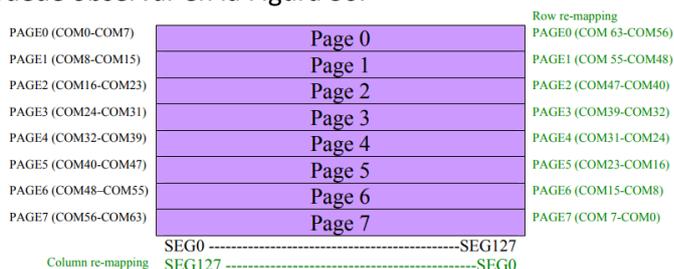


Figura 30. Distribución del módulo de pantalla OLED [9]

Esta pantalla OLED necesita un controlador para convertir los datos recibidos en señales electrónicas para controlar la pantalla. El controlador que incorpora es el SDD 1306, que junto con el uso de las librerías “Adafruit\_SSD1306” y “Adafruit GFX” permiten al usuario representar texto y gráficos de manera mucho más sencilla y rápida<sup>6</sup>.

En la Figura 31 se muestra el aspecto del módulo de pantalla OLED.

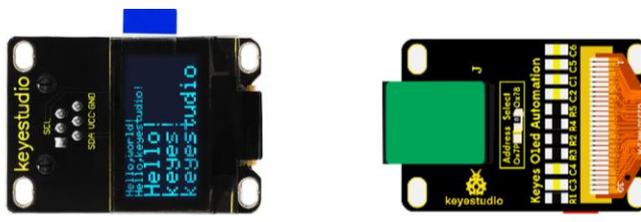


Figura 31. Módulo Easy-Plug pantalla OLED

<sup>6</sup> Datasheet Summary Oled Screen SSD1306. Solomon Systech. <https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf>

#### 4.1.4. Actuadores externos

A continuación se expone la explicación de los actuadores que serán utilizados en este proyecto:

- Bomba de agua sumergible JT-DC3W de Garosa: es un dispositivo de poca potencia, normalmente utilizado en aplicaciones como la recirculación de agua en acuarios o para riego de plantas entre otras. Este tipo de bombas formadas por un pequeño motor de corriente continua, en aplicaciones de poca potencia en las que intervienen placas con microprocesadores, son activadas a través de la conmutación de relés.

Una salida digital de placa principal establece estado alto o 1 digital, accionando el contacto de un relé y cerrando así el circuito de alimentación de la bomba, que al ser de baja potencia, tiene una tensión de funcionamiento entre 3 y 5,5 voltios.

Su funcionamiento se basa en el movimiento de un motor eléctrico de corriente continua, que acciona la rotación de un rodete con hélices transmitiendo así energía al fluido para impulsarlo hasta el destino deseado.

Una de las ventajas de su característica sumergible es que solo posee un orificio, en este caso de salida, lo que permite compactar el montaje. La bomba se coloca en posición vertical en el fondo de un pequeño depósito, y cuando el motor con las hélices comienza a funcionar, el agua es impulsada por el orificio de salida, que normalmente tiene un conducto conectado para facilitar el transporte del fluido hasta el lugar deseado.

En la Figura 32 se muestra una imagen de la bomba de agua.



*Figura 32. Bomba de agua corriente continua.*

- Ventilador corriente continua 40x40 mm de EbmPapst: este dispositivo electrónico es utilizado para la refrigeración en aparatos eléctricos como ordenadores o fuentes de alimentación. Estos pequeños ventiladores están formados por un pequeño motor de corriente continua de escobillas o sin ellas (*Brushless*). Existen diversos tipos con diferentes rangos de tensión de alimentación, 5V, 12V, 24V<sup>7</sup>.

Estos ventiladores presentan la ventaja de su fácil alimentación, su bajo nivel de ruido y principalmente su escaso consumo de potencia (0,9 W).



*Figura 33. Ventilador 5 V corriente continua 40x40mm*

---

<sup>7</sup> Datasheet Summary Fan. EbmPast. <https://docs-emea.rs-online.com/webdocs/16d6/0900766b816d6d56.pdf>

Para este proyecto en concreto se ha utilizado un ventilador que necesita 5V de tensión de alimentación para su correcto funcionamiento, obteniendo una velocidad de giro de 6000 rpm.

Para la activación de este dispositivo se seguirá el mismo procedimiento que en el caso de la bomba de agua. Un relé mediante su activación a través de la placa principal cierra el circuito de alimentación del ventilador.

- Tira de leds. Es un dispositivo de iluminación que permite la activación de los leds rojo, verde o azul de manera independiente o combinándolos unos con otros variando los pines de la tira que se alimenten o puenteen. En este caso, la tira escogida es de la marca Epistar y es recortable, por lo que la longitud se puede ajustar a la necesitada. Esta tira de fabrica tiene una longitud total de 0,5 m, pero en este proyecto, solo utilizaremos unos 10-12 cm. Se alimenta con una tensión de 5 Voltios de corriente continua y tiene un consumo de 2,88 W/m<sup>8</sup>.



Figura 34. tira de leds RGB 5 V.

Para la activación de este dispositivo, se utilizará el módulo de conversión *Easy-Plug*, conectando el pin digital al pin rojo de la tira de leds y el pin de masa con la masa del módulo de conversión.

Una consideración para tener en cuenta para la iluminación del invernadero sería la cantidad y el tipo de luz más propicio para el crecimiento de las plantas. La elección de esta es bastante complicada, pero un estudio realizado por K.J McCree en 1970, muestra la curva en la que se puede observar el mejor tipo de iluminación para la realización de la fotosíntesis de las plantas.

Mientras que para los seres humanos, la parte visible de luz es de los 450 nm a los 700 nm, para las plantas, este rango se amplía desde los 400 nm a los 750 nm, y dentro de este rango, las plantas son más sensibles a la luz roja, por lo que es la luz que se le debería aplicar al invernadero [10].

---

<sup>8</sup> Datasheet Summary Led Strip SMD 5050: <http://www.lineastc.es/new/cont/fichastecnicas/Grupo-LineasTC-Ficha-Tecnica-543241.pdf>

#### 4.1.5. Alimentación de la placa y actuadores externos.

Para la alimentación del conjunto del proyecto se ha decidido utilizar fuentes de corriente continua, posibilitando así el transporte del proyecto y la total autonomía en cuanto a enchufes o fuentes de alimentación de corriente alterna se refiere.

- Alimentación de la placa principal.

Para la placa principal, se ha optado por el uso de una batería portátil recargable para dispositivos móviles (*PowerBank*). La elección de esta fuente de alimentación de corriente continua es debida a la posibilidad de cargarla cuando se agote, y poder reemplazarla por otra durante su recarga.

Este tipo de batería es muy común en la actualidad, siendo accesible para todo el mundo y con un coste económico muy reducido.

Este cargador, proporciona una tensión de alimentación de 5V, con 1 Amperio de corriente, suficiente para el correcto funcionamiento de la placa principal y del resto de módulos. El usado durante el desarrollo de este proyecto tiene una capacidad de 2000 mAh, siendo más que suficiente para el funcionamiento del conjunto del proyecto durante más de 10 horas. Dicha batería se muestra en la Figura 35:



Figura 35. Batería de alimentación de la placa principal (*PowerBank*).

- Alimentación bomba de agua.

La bomba de agua, como se ha indicado durante su explicación en el apartado 4.1.4, necesita una tensión de operación de entre 3 y 5,5 Voltios para su funcionamiento con normalidad. En el caso de este proyecto, la función de esta bomba es el riego por goteo del invernadero. Es por ello por lo que se ha utilizado una batería de 1,5 Voltios para su alimentación. Esta elección conlleva un funcionamiento por debajo del nominal, reduciendo la velocidad de giro del motor y por tanto el caudal de la bomba, siendo más que suficiente la velocidad de circulación del agua a través del conducto para el riego por goteo requerido en esta instalación.

En la Figura 36 se muestra la batería de alimentación de la bomba de agua.



Figura 36. Batería de alimentación de la bomba de agua (*Batería AAA*).

- Alimentación del ventilador de corriente continua.  
En el caso del ventilador, como se ha indicado en el apartado 4.1.4, su tensión de funcionamiento normal son 5V, por ello se ha escogido una batería de 9 Voltios de tensión, consiguiendo de esta manera una velocidad de giro mayor que la indicada en el datasheet del producto. Un uso continuado de esta fuente de alimentación superior a la recomendada podría dañar el dispositivo de ventilación, pero en este caso, su uso va a ser de duraciones cortas por lo que el ventilador no sufrirá daños a corto plazo.  
En la Figura 37 se muestra la batería de alimentación de la bomba de agua.



Figura 37. Batería de alimentación del ventilador (Pila 9V).

## 4.2. Software. Facilino Junior

Para el diseño del software de este proyecto, se ha utilizado la aplicación *Facilino Junior* a través de la cual se desarrolla el código que los alumnos cargarán en el microcontrolador para el control del proyecto.

La elección de esta aplicación se ha llevado a cabo tras un estudio de las ventajas que esta representa a la hora de facilitar la labor de la parte programación tanto para el alumno como para el profesorado expuestas en este apartado.

*Facilino Junior* es una herramienta de programación visual por bloques desarrollada por la compañía Robótica Fácil. Esta aplicación sirve como introducción a los alumnos al mundo de la programación. A su vez, *Facilino Junior* permite transversalmente a los alumnos desarrollar proyectos de las demás asignaturas y afianzar conocimientos a través del refuerzo de los contenidos con proyectos de simulación y experimentos entre otros.

Además, esta herramienta es de código abierto, por lo que es accesible por cualquier usuario a través de la plataforma de desarrollo colaborativo *GitHub*, desde la cual pueden editar y realizar cambios libremente en el programa [11].

### 4.2.1. Estructura-Base. Blockly de Google.

La base de la estructura de esta aplicación es la herramienta *Blockly*. Esta aplicación desarrollada por Google permite a usuarios con conocimientos nulos de código crear programas básicos para comenzar a desarrollar iniciativas y conocimientos de programación.

Esta herramienta busca la máxima simplificación y facilidad en la programación evitando los problemas más comunes que suelen ocurrir en los comienzos del uso de este lenguaje como la falta de caracteres (paréntesis, corchetes, puntos y coma, etcétera), el desconocimiento de las funciones así como la velocidad de programación. *Blockly* surge bajo la influencia de Scratch, la primera herramienta de programación visual por bloques.

*Blockly* posee bloques por defecto de las instrucciones básicas de control, lógica, matemáticas o texto entre otras, pero el desarrollador puede crear instrucciones de control personalizadas para formar aplicaciones específicas para el desarrollo de proyectos concretos con bloques de control únicos.

La interfaz de programa corresponde con el menú manejado por el usuario y consta de las siguientes partes indicadas en la Figura 38.

- (1) Menú de bloques.
- (2) Espacio de trabajo para organizar los bloques. (*WorkSpace*)
- (3) Selección del lenguaje de programación del código generado.
- (4) Código generado
- (5) Herramienta para eliminación de bloques
- (6) Botón de inicio para la ejecución del código creado.

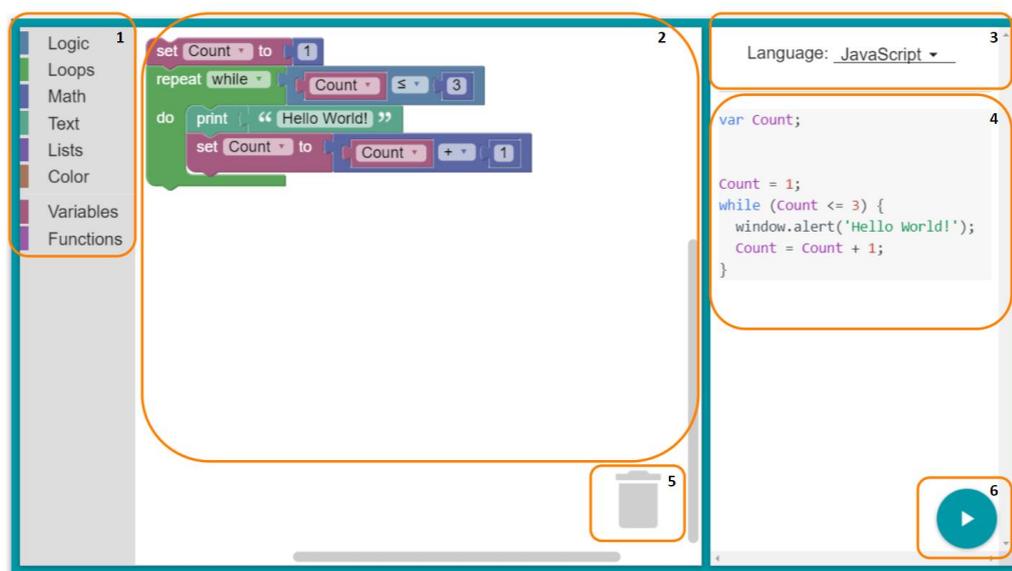


Figura 38. Interfaz de usuario de Blockly [12].

El funcionamiento de esta herramienta es muy simple. El usuario es el encargado de utilizar los comandos existentes y combinarlos con criterio arrastrándolos al espacio de trabajo o *WorkSpace*. Los comandos son los bloques de distintos colores con forma de piezas de puzle, permitiendo la unión de unos con otros. Cada color identifica una categoría de bloques con las mismas características, por ejemplo, el color morado se relaciona con los bloques que contienen instrucciones de lógica, el color verde con la categoría de texto, etc. Las categorías de bloques por defecto de Blockly son control, lógica, operaciones matemáticas, texto, listados, procesos, variables y funciones.

Tras la configuración de este “puzle”, la herramienta genera un script de código sintácticamente correcto en el lenguaje de programación que el usuario desee, JavaScript, Python o PHP, Lua o Dart.

Los desarrolladores pueden utilizar esta herramienta de base para la creación de aplicaciones personalizadas. Google Blockly dispone de una plataforma denominada “*Blockly Developer Tools*” que permite a los desarrolladores personalizar su propio *WorkSpace*, crear sus propios bloques personalizados a través de “*Block Factory Tab*” y crear un menú de bloques y un espacio de trabajo con las necesidades de la aplicación que vaya a desarrollar.

La creación de los bloques puede realizarse también a través de programación usando el lenguaje de programación JavaScript, desarrollando los dos conjuntos de instrucciones que cada bloque necesita:

1) Conjunto de instrucciones de definición del bloque, que describen la apariencia de este. Puede contener información sobre el nombre del bloque en cuestión, el color, el texto o textos que contiene, las entradas que pueden conectarse a él y a que piezas puede conectarse.

En la Figura 39 se muestra un ejemplo del conjunto de instrucciones de la definición de un bloque cuya finalidad es extraer el número de caracteres que contiene una cadena de caracteres:

```
Blockly.Blocks['string length'] = {
  init: function() {
    this.appendValueInput('VALUE')
      .setCheck('String')
      .appendField('length of');
    this.setOutput(true, 'Number');
    this.setColour(160);
  }
};
```

Figura 39. Apariencia del código de definición de un bloque en JavaScript.

2) Conjunto de instrucciones de código del bloque, que contienen el código para el programa en el lenguaje de programación apropiado para cada proyecto. En el caso de este ejemplo, se declara la variable “text” donde se guardará el contenido de la cadena de caracteres de texto que debe enlazar con este bloque. A continuación se generan las instrucciones para que al ejecutar el código, la función “text.length” de JavaScript calcule la cantidad de caracteres que contiene el texto introducido en la función.

Estas instrucciones para el código del bloque se muestran en la Figura 40:

```
Blockly.JavaScript['string length'] = function(block) {
  var text = Blockly.JavaScript.valueToCode(block, 'VALUE',
    Blockly.JavaScript.ORDER_MEMBER) || `''`;
  var code = text + '.length';
  return [code, Blockly.JavaScript.ORDER_MEMBER];
};
```

Figura 40. Apariencia de las instrucciones de código de un bloque en JavaScript.

La apariencia en la interfaz del programa del bloque con las instrucciones explicadas sería la mostrada en la Figura 41.



Figura 41. Apariencia final del bloque con las instrucciones del ejemplo planteado en las anteriores dos figuras.

El código como resultado de la unión del bloque en cuestión con una cadena de texto es el mostrado en la Figura 42.

```
'Hello World!'.length;
```

Figura 42. Código generado a raíz de la unión de una cadena de texto al bloque creado.

En el apartado 4.2.5 se explicará con más detalle el código en JavaScript creado para la personalización de cada bloque en el actual Trabajo Fin de Grado.

#### 4.2.2. Interfaz de usuario.

Como se ha indicado anteriormente, Facilino Junior está construido sobre la herramienta de Google Blockly, pero posee la peculiaridad que está orientada a su uso a estudiantes de edades tempranas. Esta aplicación busca la máxima simplificación de los bloques, de la interfaz y de su funcionalidad permitiendo a los alumnos conocer la función que puede realizar un bloque sin necesidad de texto escrito en él y solo con la ayuda de imágenes e iconos. Siguiendo este principio, se facilita la visualización de los bloques y las partes del programa, consiguiendo que los alumnos relacionen la función de un bloque a primera vista.

Este es el motivo por el que se ha escogido la aplicación Facilino Junior para desarrollar este proyecto, además de la posibilidad que ofrece a la hora de introducir bloques personalizados para cada una de las aplicaciones que se quieran desarrollar con él.

En la Figura 43 se puede visualizar la interfaz de usuario de Facilino Junior similar a la de Blockly pero con algunas particularidades:

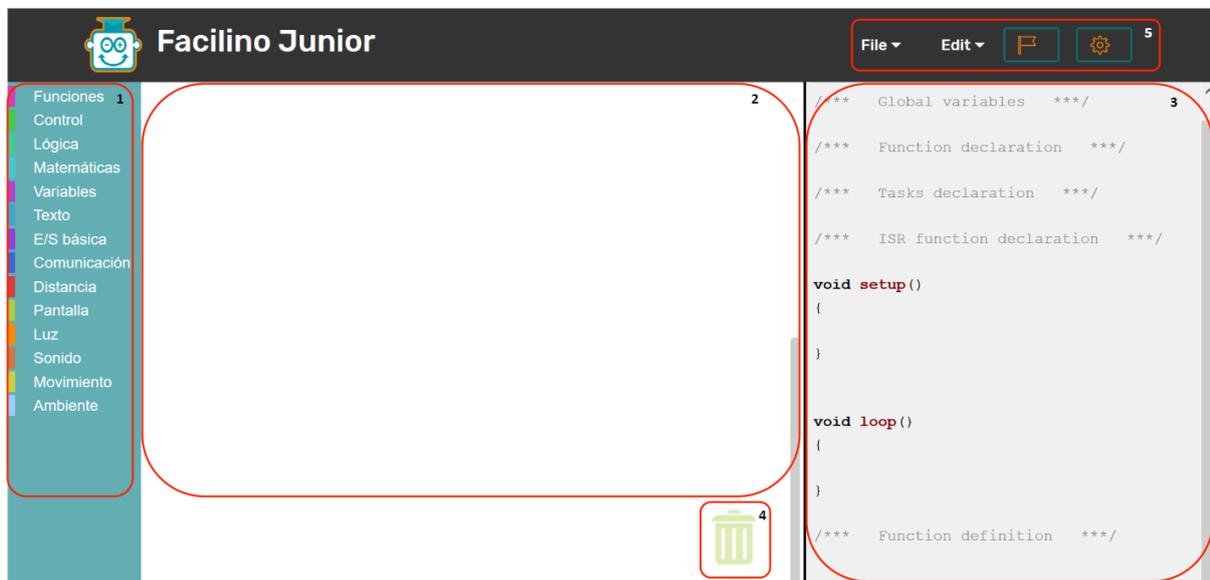


Figura 43. Interfaz de usuario Facilino Junior.

Consta de las siguientes partes:

- (1) Menú de bloques con categorías y bloques personalizados.
- (2) Espacio de trabajo para organizar los bloques. (*Workspace*)
- (3) Selección del lenguaje de programación del código generado.
- (4) Herramienta para eliminación de bloques
- (5) Parámetros de configuración para abrir proyectos ya existentes o descargar el proyecto realizado, selección del lenguaje de la aplicación y selección de la placa que se va a utilizar para cargar el programa.

### 4.2.3. Microcontroladores compatibles.

El código de instrucciones generado por Facilino es compatible con la programación de microprocesadores Arduino y ESP. El botón de ajustes de la interfaz, como se muestra en la Figura 44, permite al usuario seleccionar que placa va a utilizar para desarrollar el proyecto, pudiendo elegir entre Arduino UNO, Arduino Nano o Placa *Easy-Plug* de KeyStudio.

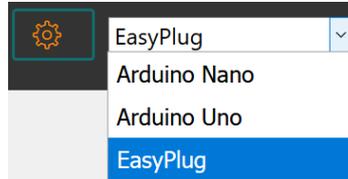


Figura 44. Herramienta de selección de placa en la aplicación Facilino Junior.

Este parámetro de configuración cuenta con la ventaja de que permite que un programa generado en un principio para la placa *Easy-Plug*, se convierte compatible para su uso en una placa de Arduino UNO, simplemente seleccionando la placa a utilizar en el desplegable.

La diferencia entra la elección de una placa u otra son los puertos digitales y analógicos disponibles para su uso. En el caso de la placa *Easy-Plug*, los puertos disponibles explicados en el apartado 4.1.1 son inferiores a los puertos disponibles en un Arduino UNO debido a la falta de espacio en la placa debido a los conectores RJ11 *Easy-Plug*.

En el proyecto que se trata en esta memoria, utilizaremos las instrucciones en lenguaje de programación de Arduino, ya que utilizaremos la placa principal *Easy-Plug* que contiene el microprocesador ATmega328 explicada con anterioridad (4.1.1).

Cargar estas instrucciones creadas por Facilino Junior en el microcontrolador es posible gracias al Bootloader albergado en la memoria flash de este.

Bootloader es un programa de poco tamaño guardado en el microcontrolador que permite la carga del programa del proyecto a través del puerto serie USB.

Este programa solo se ejecuta cuando Arduino se enciende o cuando se pulsa el botón reseteo, hasta que detecta un nuevo programa a través del puerto serie USB. Solo ocupa 0,5 KBytes de los 32 Kbytes de memoria flash disponibles. El Bootloader de Arduino es lo que diferencia al microcontrolador ATmega328/P de otro microcontrolador salido de fábrica.

La función principal del Bootloader es resetear el programa en ejecución para cargar uno nuevo. En el caso de Arduino, este reseteo se hace de forma automática cuando se manda el nuevo programa. Se ejecuta el Bootloader de Arduino para eliminar el antiguo programa, dejando así disponible toda la memoria flash, y después espera hasta que un nuevo programa sea enviado a través del puerto serie USB.

#### 4.2.4. Idiomas disponibles

Esta aplicación permite su uso en 15 idiomas, ampliando de esta manera las fronteras de su utilidad. Para hacer esto posible, toda la información ha sido traducida a estos 15 idiomas que se muestran a continuación en la Figura 45.

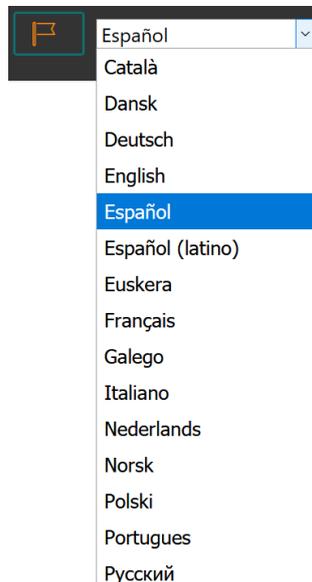


Figura 45. Herramienta de selección de idioma en la aplicación Facilino Junior.

Cuando uno de estos idiomas es seleccionado, todas las instrucciones, así como la información perteneciente a cada bloque, se traduce, facilitando su uso en otros idiomas.

#### 4.2.5. Programación de Facilino Junior con JavaScript.

Para la creación de los bloques personalizados para este proyecto y la adaptación de los bloques ya existentes se ha utilizado tanto la herramienta de creación de bloques de Blockly (*"Blockly Factory Tab"*) como el lenguaje de programación JavaScript. Como se ha indicado en el apartado 4.2.1, cada bloque de Facilino Junior contiene dos conjuntos de instrucciones:

1. Instrucciones de definición de bloque.
2. Instrucciones de código de bloque.

A continuación se explican los comandos los bloques y las instrucciones utilizados para la configuración de cada conjunto

##### 1. Conjunto de instrucciones de definición del bloque.

Conforman el aspecto visual de este. Desde la herramienta *Blockly Factory Tab*, se proporciona un bloque base como el que se muestra en la Figura 46, a partir del cual se combinan los demás y se configuran los diferentes parámetros.

Para la explicación de dichos parámetros, se utilizará la siguiente figura, haciendo uso de la numeración de las distintas partes que esta contiene.

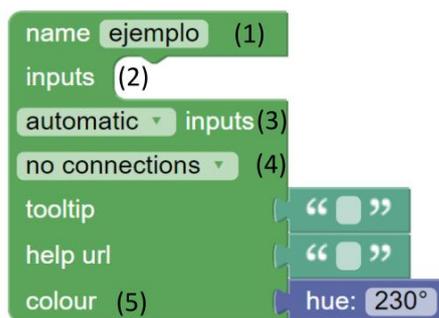


Figura 46. Bloque base para configuración de bloques en Blockly Factory Tab.

- En primer lugar, se introduce el nombre del bloque en el campo “**name**” (1).
- En segundo lugar, se introducen los tipos de entrada deseados en el campo “**input**” (2). Existen tres tipos de entradas posibles para cubrir las necesidades de cada bloque.

a) “Value Input”. Configura el bloque con una entrada hembra, permitiendo la conexión de este con una entrada macho, para introducir bloques de texto, numéricos, lógicos, etc.

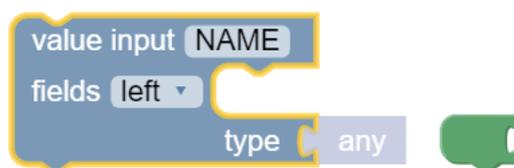


Figura 47. Bloque de configuración “Value Input” y resultado de este.

b) “Statement Input”. Configura el bloque con una entrada que permite su combinación con bloques mayores, de tipo sensores, actuadores, lógicos, etc.

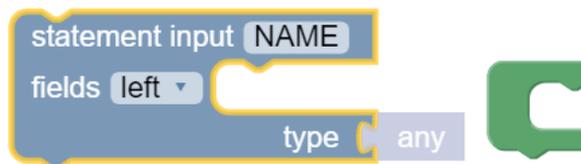


Figura 48. Bloque de configuración “Statement Input” y resultado de este.

c) “Dummy Input”. Permite configurar visualmente un espacio vacío, para la introducción. De los diferentes “fields” que serán explicados a continuación.



Figura 49. Bloque de configuración “Dummy Input” y resultado de este.

Como se pueden observar en los bloques de configuración de tipos de entrada, todos poseen un campo en común denominado "field". Este campo puede ser completado con las siguientes opciones:

- Opción "text". Permite mostrar un texto en el bloque, el cual no realiza ninguna acción, simplemente lo muestra información de forma visual al usuario.



Figura 50. Bloque de configuración para la opción "text" y resultado de este.

- Opción "text input". Crea un campo en el bloque para la introducción de una cadena de caracteres, la cual puede ser utilizada más adelante en el programa. Esta cadena de caracteres se guarda en la variable "NAME", a la cual se hace referencia en el conjunto de instrucciones de código.



Figura 51. Bloque de configuración para la opción "text input" y resultado de este.

- Opción "numeric input". Crea un campo con la misma finalidad que la opción anterior, con la diferencia que en vez de ser para la introducción de una cadena de caracteres es para la introducción de un número. Este bloque permite limitar el número máximo y mínimo, así como los decimales.

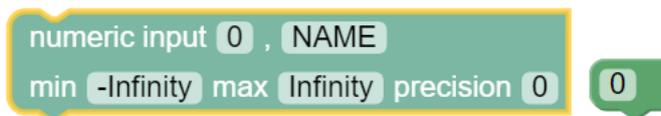


Figura 52. Bloque de configuración para la opción "numeric input" y resultado de este.

- Opción "angle input". Crea un campo en el bloque para la introducción de un valor de ángulo en grados. Esta opción se utiliza por ejemplo para la creación de un bloque que contenga instrucciones para el movimiento de un servomotor.



Figura 53. Bloque de configuración para la opción "angle input" y resultado de este.

- Opción "dropdown". Esta opción crea un desplegable con contenido establecido por el creador, de tal manera que permite al usuario escoger entre las opciones que contenga y hacer referencia a cada una de las variables en el conjunto de instrucciones de código.

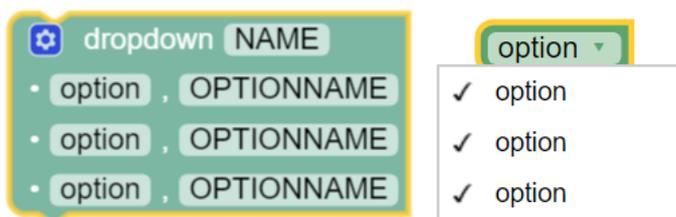


Figura 54. Bloque de configuración para la opción "dropdown" y resultado de este.

- En tercer lugar, se selecciona la posición para el enganche de la entrada en el desplegable “input” (3), interna, externa o automática como se muestra en la Figura 55.



Figura 55. Tipo de entrada interna o externa para combinación de bloques.

- En cuarto lugar, se configura el bloque atendiendo a las conexiones que se le pueden realizar y la posición de estas seleccionándolas en el desplegable “connections” (4), por la izquierda, por arriba y por abajo, solo por arriba o solo por abajo, como se muestra de izquierda a derecha en la Figura 56.



Figura 56. Tipos de combinaciones posibles.

- Por último, se configura el tono de color en el campo “color” (5) de cada bloque con un número de 0 a 360, con la ayuda de la circunferencia que se muestra en la Figura 57 .

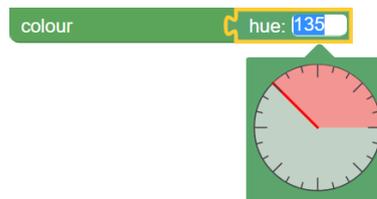


Figura 57. Configuración del tono de color para el bloque.

Para ejemplificar la explicación de toda la parte de configuración del conjunto de instrucciones de definición del bloque, se alude al siguiente ejemplo:

Se desea crear un bloque con un desplegable para la selección del tamaño del texto de una pantalla OLED. El tamaño del texto está restringido a 3 posibles campos: pequeño, mediano o grande.

Para resolver este ejemplo, se utiliza el bloque “dummy input”, ya que no necesitamos introducir ninguna variable a nuestro bloque. A continuación utilizamos el field “text” para escribir la palabra “tamaño” en el bloque, y lo combinamos con el field “dropdown” que contendrá los tres posibles valores para el tamaño del texto, como podemos observar en la Figura 58.

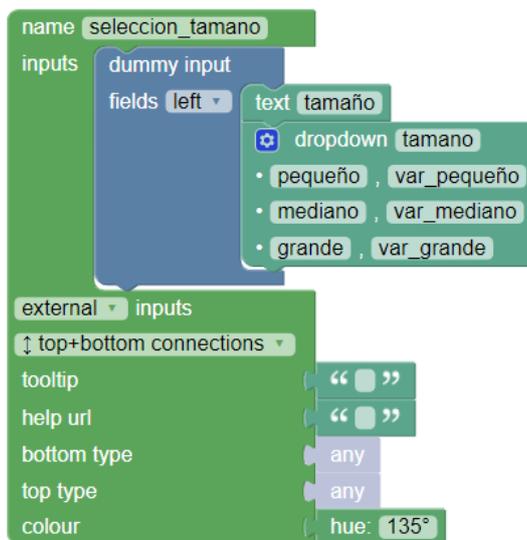


Figura 58. Combinación de bloques en Blockly Factory Tab para obtener un bloque desplegable con tres tamaños.

El resultado visual de las combinaciones de estos bloques de configuración da lugar al siguiente bloque mostrado en la Figura 59:

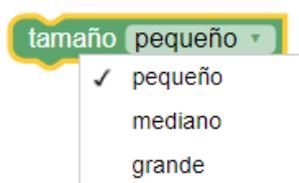


Figura 59. Bloque de selección de tamaño.

La creación de este bloque a través de la herramienta de creación de bloques de Blockly, genera el código de la Figura 60 en lenguaje JavaScript para su incorporación en la aplicación deseada:

```
Blockly.Blocks['seleccion_tamano'] = {
  init: function() {
    this.appendDummyInput()
      .appendField("tamaño")
      .appendField(new Blockly.FieldDropdown([["pequeño", "var_pequeño"], ["mediano", "var_mediano"],
["grande", "var_grande"]]), "tamano");
    this.setInputsInline(false);
    this.setPreviousStatement(true, null);
    this.setNextStatement(true, null);
    this.setColour(135);
    this.setTooltip("");
    this.setHelpUrl("");
  }
};
```

Figura 60. Conjunto de instrucciones de definición del bloque de selección de tamaño en lenguaje JavaScript

## 2. Conjunto de instrucciones de código del bloque.

Utilizando la herramienta “*Bloky Factory Tab*”, se genera tanto el conjunto de instrucciones de definición del bloque como el conjunto de instrucciones de código del bloque. En el caso del ejemplo anterior, la herramienta devuelve el siguiente conjunto de instrucciones para la configuración del código mostrado en la Figura 61:

```
Blockly.JavaScript['seleccion_tamano'] = function(block) {  
  var dropdown_tamano = block.getFieldValue('tamano');  
  // TODO: Assemble JavaScript into code variable.  
  var code = '...;\n';  
  return code;  
};
```

Figura 61. Conjunto de instrucciones de código del bloque de selección de tamaño en lenguaje JavaScript.

La configuración del código se introduce en la variable “code”, programando las acciones que se quieran llevar a cabo en cada caso. Ese código es único y se debe desarrollar dependiendo de la función final de cada bloque.

### 4.2.6. Categorías existentes y bloques creados.

Las categorías en el menú de bloques han aumentado, adaptándose para mejorar la clasificación de todos nuevos bloques, dividiéndose en las siguientes:

- Categoría “*Funciones*”: contiene bloques para la creación de funciones. Todos los bloques incluidos en esta categoría permiten al usuario agrupar código dentro de una función para poder hacer referencia a ella múltiples veces durante el resto del programa. Se incluyen los bloques para la creación de estas funciones así como para la llamada de ellas en el programa. Los bloques de esta categoría se identifican con el color fucsia. En la Figura 62 se muestra uno de los bloques de esta categoría:

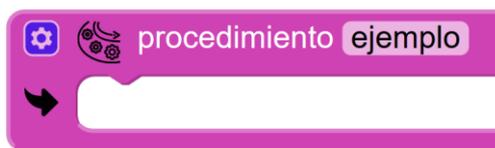


Figura 62. Ejemplo de bloque de la categoría funciones (definición de una función).

- Categoría “*Control*”: está formada por los bloques encargados de estructurar el programa. Esta categoría incluye el bloque que contiene la estructura básica de un programa dividiéndolo en la parte de preparación del programa “*SetUp()*” y la ejecución de este “*loop()*”  
Además incluye bloques para la ejecución de bucles “*while()*” y “*for()*”, y un bloque para condición “*if()*”, “*if(){else{}}*”. Los bloques de esta categoría se identifican con el color verde césped.  
En la Figura 63 se muestra un bloque de ejemplo de esta categoría, en este caso, el bloque de control “*while*”.



Figura 63. Ejemplo de bloque de la categoría control (bucle de control while).

Esta categoría cuenta también con un bloque de tiempo, mostrado en la Figura 64, cuya función es esperar el tiempo indicado por el usuario en milisegundos, a través del comando "delay()".



Figura 64. Bloque de espera de tiempo (en ms) de la categoría control.

- Categoría "Lógica": está constituida por los bloques de operaciones booleanas y estados de las entradas y salidas analógicas y digitales. Los bloques de esta categoría se identifican con el color verde oscuro.
  - o Bloques de estados: Verdadero/falso ("true;" o "false;"), y dos nuevos que contienen las mismas instrucciones en código, pero son utilizados en casos particulares.

Por un lado, un bloque de alto/bajo ("HIGH;" o "LOW;"), y por otro lado un bloque de Encendido/Apagado ("ON;" o "OFF;"), que configuran la activación o desactivación de las salidas digitales.

En la Figura 65 y Figura 66 se muestran estos bloques en sus dos configuraciones posibles:



Figura 65. Bloques de estados "HIGH/LOW" y "ON/OFF" en estado "HIGH" y "ON" de la categoría lógica.



Figura 66. Bloques de estados "HIGH/LOW" y "ON/OFF" en estado "LOW" y "OFF" de la categoría lógica.

- o Bloques de operaciones booleanas, referidas a la comparación de dos variables (iguales o no, mayores, menores, etc.), operaciones de producto o suma lógica ("AND", "OR"), y operaciones de inversión o negación booleanas. Un ejemplo de bloque de esta categoría sería el de operaciones booleanas mostrado en la Figura 67.



Figura 67. Ejemplo de bloque de la categoría lógica (operaciones booleanas).

- Categorías “*Matemáticas*”: contiene bloques matemáticos, tanto numéricos como de operaciones aritméticas, reescalado de valores analógicos o elección al azar entre un rango de números establecido. Los bloques de esta categoría se identifican con el color verde azulado. Un ejemplo de bloque de esta categoría sería el de operaciones aritméticas mostrado en la Figura 68.



Figura 68. Ejemplo de bloque de la categoría matemáticas (Operaciones aritméticas).

- Categoría “*Variables*”: está formada por bloques para crear o definir una variable, o devolver o guardar un valor en una variable. Los bloques de esta categoría se identifican con el color púrpura, como se observa en la Figura 69, donde se muestran algunos bloques de esta categoría.

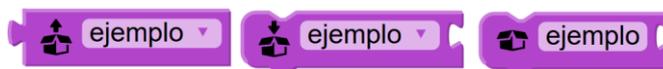


Figura 69. Bloques de la categoría variables.

- Categoría “*Texto*”: contiene bloques destinados a operaciones con cadenas de caracteres: creación de estas, concatenación y cuenta del número de caracteres. Por otro lado, también existen bloques para la comparación o el cambio a mayúsculas o a minúsculas de cadenas de caracteres, o conversión de una cadena a un número. Los bloques de esta categoría se identifican con el color azul oscuro como se puede observar en el bloque de ejemplo de la Figura 70.

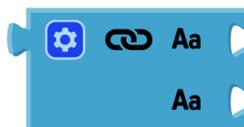


Figura 70. Ejemplo de bloque de la categoría texto (concatenación de cadenas de caracteres).

- Categoría “*Entrada/Salida básica*”: esta categoría contiene los bloques para la lectura y escritura de los pines digitales y analógicos de la placa. Los bloques que contiene son:
  - Lectura y escritura de un pin analógico tomando los posibles valores desde 0 a 255.
  - Lectura y escritura de un pin digital, tomando como posibles valores 0 o 1.
  - Lectura de pin digital para estado de un botón.
  - Escritura en un pin digital para la activación o desactivación de un relé.

Los bloques de esta categoría se identifican con el color morado, como se puede observar en el bloque de ejemplo de la Figura 71.



Figura 71. Ejemplo de bloque de la categoría Entrada/salida básica (Bloque de activación/desactivación de relé).

- Categoría “Comunicación”: contiene los bloques que permiten la comunicación a la placa a través del conector USB o a través del puerto COM. Estos bloques imprimen instrucciones en la consola, leen los datos recibidos por el puerto serie, o configuran el módulo bluetooth para el intercambio de información entre la placa y el dispositivo de manejo. Los módulos de esta categoría se identifican con el color añil. En la Figura 72 se muestra un ejemplo de bloque de esta categoría.



Figura 72. Ejemplo de bloque de la categoría comunicación (Bloque de escritura por puerto USB).

- Categoría “Distancia”: incluye bloques necesarios para la configuración y el uso de un módulo emisor/receptor infrarrojo utilizado en diversas aplicaciones, como por ejemplo la detección de obstáculos. Los bloques de esta categoría se identifican con el color rojo como se puede observar en la Figura 73.



Figura 73. Ejemplo de bloque de la categoría Distancia (Bloque de configuración del módulo infrarrojo).

- Categoría “Pantalla”: está formada por los bloques necesarios para la configuración y el uso de módulos de pantallas a través de la placa principal. Los bloques de esta categoría se identifican con el color verde claro. Incluye bloques para los siguientes tipos de módulos de pantallas:
  - o Módulo de matriz de leds. Contiene bloques para la creación de patrones tanto manuales como predefinidos. En la Figura 74 se muestran dichos bloques.

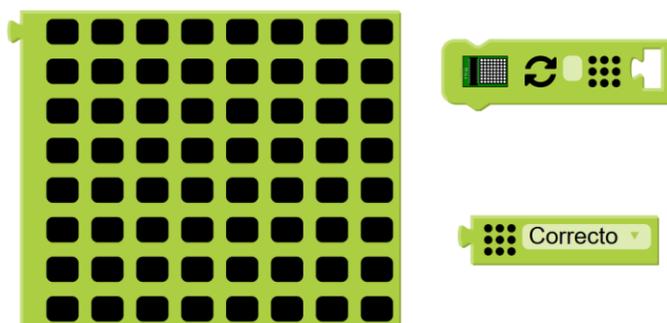


Figura 74. Bloques para módulo de matriz de leds de la categoría pantallas.

- o Módulo pantalla OLED. La categoría “pantallas”, incluye bloques para la configuración y el uso de pantallas OLED de 128x64 píxeles. Estos bloques permiten la escritura en la pantalla OLED tanto de una cadena de caracteres y un número, solo de una cadena de caracteres o solo de un número. Además estos módulos permiten configurar la posición de escritura en la pantalla así como el tamaño del contenido a escribir. También se incluye un bloque que permite el borrado del contenido de la pantalla, dejándola sin contenido a la espera de nuevas instrucciones. En la Figura 75 se observan dichos módulos.



Figura 75. Bloques para módulo OLED de la categoría pantallas.

Cada uno de estos bloques contiene las instrucciones para la configuración de la pantalla OLED, como son la definición de librerías y los parámetros de ajustes para el reseteo de la pantalla.

- Categoría "Luz": contiene todos los bloques relacionados con la activación o desactivación de todo tipo de leds y tiras de leds, permitiendo variar el color en el caso de los leds RGB, o el brillo. Los bloques de esta categoría se identifican con el color naranja. En la Figura 76, se muestra un ejemplo de bloque de esta categoría.



Figura 76. Ejemplo de bloque de la categoría Luz (Bloque de configuración de color Led RGB).

- Categoría "Sonido": está formado por bloques que permiten la reproducción de sonidos a través del uso de buzzers. Los bloques contienen definición de notas de distinta duración y tono así como silencios. Esta categoría se identifica gracias al color marrón de los bloques como se muestra en la Figura 77.



Figura 77. Ejemplo de bloque de categoría Sonido (Bloque de reproducción de sonidos definidos).

- Categoría "Movimiento": contiene los bloques relacionados con la configuración y uso de módulos de servo de rotación continua y de posición o de motor de continua. Los bloques de esta categoría se identifican con el color verde pistacho. En la Figura 78 se muestra uno de los bloques de esta categoría.



Figura 78. Ejemplo de bloque de la categoría Movimiento (Bloque de configuración de la posición de un servomotor).

- Categoría “Ambiente”: Esta formada por los bloques destinados a la configuración y lectura de los valores detectados por sensores ambientales. Estos bloques se identifican con el color azul claro.
  - o Bloques para módulos de temperatura: estos bloques están destinados a leer el valor de temperatura de sensores destinados a este fin. Se incluyen bloques para módulos DHT11 y 18B20. En el caso de los bloques para el sensor DHT11 se ha incorporado un bloque de alarma que permite fijar un valor máximo y uno mínimo de temperatura, pudiendo establecer las instrucciones a ejecutar en cada caso. Este bloque tiene el aspecto que se muestra en la Figura 79.

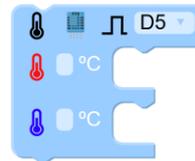


Figura 79. Bloque para lectura de temperatura con sensor DHT11 de la categoría Ambiente

- o Bloques para módulos de humedad: tanto del aire como de un terreno. Para estos módulos también se han desarrollado bloques de alarma con la misma funcionalidad que el bloque DHT11 explicado en el punto anterior, estableciendo en este caso un porcentaje máximo y uno mínimo de humedad. Permiten leer el valor, en este caso analógico (0-1023), de humedad captado por el sensor. Este bloque tiene el aspecto que se muestra en la Figura 80.



Figura 80. Bloque para la lectura de la humedad de la tierra de la categoría Ambiente.

En el caso del bloque de alarma de humedad de tierra, se ha diseñado un bloque más específico para este proyecto mediante el cual se permite el control de la activación o desactivación de una salida digital en el caso en que el valor de la variable humedad de suelo cumpla con las condiciones marcadas en dicho bloque o un botón cuyo pin es seleccionable sea activado por el usuario.

La función principal de este bloque es la activación de la bomba de agua de forma manual, permitiendo así al usuario regar el invernadero cuando lo desee y no solo cuando la humedad sea menor que la mínima programada.

Dicho bloque se muestra en la Figura 81.

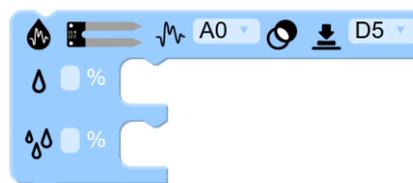


Figura 81. Bloque para la lectura de la humedad de la tierra y de un botón de la categoría Ambiente.

- Bloques para módulos de luminosidad: son bloques diseñados para leer el valor de luminosidad, en este caso es un valor analógico (0-1023), captado por una fotorresistencia. De la misma forma que en los puntos anteriores, se ha desarrollado un bloque de alarma que permite establecer un valor máximo y un valor mínimo de porcentaje de luminosidad. Este bloque tiene el aspecto que se muestra en la Figura 82.

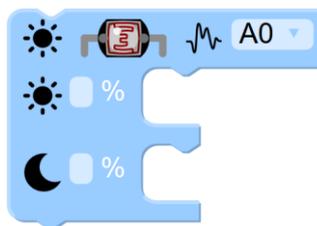


Figura 82. Bloque para la lectura de la luminosidad con una fotorresistencia de la categoría Ambiente.

- Bloques para otros valores ambientales: también se incluyen bloques para la lectura de la cantidad de agua o lluvia, para la lectura de un sensor de gas, para la lectura de temperatura y presión de un sensor barométrico, y para la lectura de altitud. En la Figura 83 se muestra uno de estos bloques a modo de ejemplo.



Figura 83. Ejemplo de bloque para la lectura de otros valores ambientales de la categoría Ambiente (Bloque de lectura de sensor de gas).

Todos estos bloques se han desarrollado siguiendo un principio de máxima simplificación, evitando el uso de textos que puedan dar lugar a confusión o desconocimiento de los conceptos. Es por ello por lo que todos los contenidos de los bloques se han intentado completar mediante el uso de ilustraciones. Todas las ilustraciones empleadas en los módulos han sido extraídas de la biblioteca *Flaticon* [13], siendo todas ellas archivos con extensión “.svg”. Esta extensión incluye imágenes vectoriales. Este formato de imagen permite su escalado sin pérdida de calidad o pixelado, ocupa menor tamaño y permite mayor definición en formatos reducidos. El uso de este formato garantiza que estas ilustraciones serán visualizadas de forma correcta en cualquier ordenador y a cualquier tamaño, evitando problemas a la hora de realizar la programación.

Los bloques que permiten su conexión a otros o que les sean conectados otros, contienen una instrucción que restringen las conexiones sin sentido. Para la explicación de este concepto, se alude a un ejemplo:

Si se quiere comunicar con el ordenador a través del puerto USB y mostrar la unión de dos cadenas de caracteres a través del bloque operaciones aritméticas de la categoría “matemáticas” configurado como suma, el programa restringirá la entrada de cadenas de caracteres a dicho bloque, como podemos observar en la Figura 84



Figura 84. Restricción de entrada de cadenas de caracteres en bloque de categoría “matemáticas”.

En cambio, la creación correcta de los bloques para realizar el programa explicado anteriormente sería mediante la utilización del bloque de concatenación de cadenas de caracteres de la categoría "texto". En la Figura 85 se muestra dicha combinación de bloques.

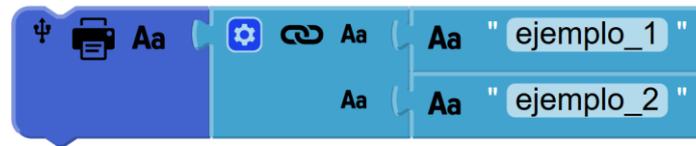


Figura 85. Solución correcta para concatenación de textos mediante el bloque de la categoría "texto".

Otra ventaja significativa para destacar es la incorporación de todas las librerías al código de cada bloque. Es decir, todos bloques que para su utilización impliquen la necesidad del uso de librerías, como por ejemplo los bloques de pantallas o el bloque de temperatura para sensor DHT11, contienen las instrucciones para el acceso a estas librerías, sin necesidad de tener que añadir código tras la realización de la combinación de los bloques, asegurando así el correcto funcionamiento de estos y la compatibilidad con los módulos Easy-Plug de hardware.

### 4.3. Invernadero

Como se ha indicado con anterioridad, el proyecto elegido para poner en práctica los conceptos de robótica educativa utilizando la tecnología *Easy-Plug* explicada es un invernadero automatizado a pequeña escala.

La elección de este proyecto es acorde con la tecnología a utilizar, evaluando las ventajas y los conocimientos que el desarrollo de este proyecto aporta a los alumnos más allá de la robótica educativa y programación.

El estado actual de nuestro planeta plantea una gran cantidad de cuestiones con un importante peso para la sociedad. El cambio climático es uno de los problemas ambientales de mayor gravedad en la actualidad. Su causa principal es el aumento de los gases de efecto invernadero, provocando la interrupción del reflejo de la radiación solar, quedando atrapada dentro de la atmosfera, aumentando así la temperatura del planeta.

La concienciación es una de las armas más poderosas para reducir el cambio climático. Por ello nace el término Educación Medioambiental, que busca en la vida de las personas, impartir conciencia ambiental, conocimiento ecológico, y actitudes que constituyan un uso racional del planeta y sus recursos. Esta rama de educación, como todas, es fundamental comenzar a impartirlas lo antes posible, pero está en concreto más aún, consiguiendo así que las nuevas generaciones, puedan identificar y solucionar problemas ambientales cada vez desde edades más tempranas.

El desarrollo de actividades en contacto directo con la naturaleza, y relacionadas con ella, permite a los alumnos tener una concienciación sobre la importancia de cuidar el planeta, ayudándoles a comprender las ventajas que a nosotros como seres humanos nos proporciona esta, dando a conocer valores, técnicas y métodos para cuidarla.

Por tanto, el desarrollo en los centros educativos de ciertas actividades apoya tanto la adquisición de estos conocimientos como la concienciación y sensibilización en los alumnos.

En el presente Trabajo Fin de grado, la elección de la temática secundaria consta de un invernadero automatizado. La construcción de esta pequeña instalación permite a los alumnos adquirir compromiso y responsabilidad, además de promover el trabajo en equipo y el contacto con la naturaleza.

Este invernadero da a conocer a los alumnos la importancia de la naturaleza, la gran cantidad de recursos que somos capaces de obtener de ella y los cuidados que esta necesita para favorecer su permanencia y no deterioro en el tiempo.

Este es el motivo de la elección del ámbito en el que incorporar la robótica, consiguiendo que este proyecto reúna dos nuevas educaciones fundamentales para las nuevas generaciones: Robótica educativa y Educación Medioambiental.

Cabe destacar que todos los materiales utilizados son en la medida de lo posible reciclados. De esta manera, el proyecto también inculca en los alumnos conocimientos sobre el reciclaje, otro concepto muy necesario para las nuevas generaciones, enseñando a estas, formas de dar una segunda vida a objetos cotidianos, como los envases de los alimentos. Gracias al reciclaje, estos residuos pueden pasar a formar parte de un invernadero, sirviendo de soporte para el crecimiento y el desarrollo de plantas dentro de un espacio controlado.

A continuación, se detallan los materiales y procedimientos para la construcción del invernadero a pequeña escala:

El invernadero se divide en 5 partes principales:

- Base o contenedor.

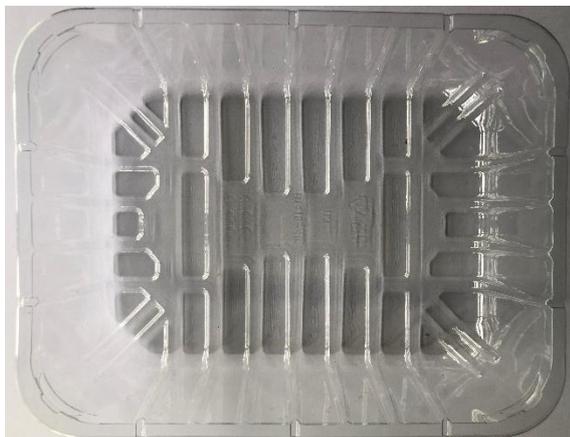
Para la base del invernadero se ha utilizado una bandeja de plástico reciclada opaca. Estas bandejas suelen utilizarse como envase para la venta de verdura y hortalizas (setas, champiñones, judías verdes, etcétera) en los supermercados. Existen de muchos tamaños y formas. En el caso del proyecto en cuestión, se ha utilizado una bandeja de dimensiones 14 cm de ancho, 18 cm de largo y una profundidad de 5 cm. Las dimensiones del ancho y el alto pueden ser variables dependiendo de la cantidad de semillas que se quieran plantar, pero la profundidad debe ser mínimo de 5 cm, dejando espacio suficiente para el próspero crecimiento de la semilla y del desarrollo de las raíces.

En la Figura 86 se muestra una imagen de la bandeja utilizada como base del invernadero:



*Figura 86. Envase utilizado para la base del invernadero.*

- Cubierta o techo.  
Para la cubierta del invernadero se ha utilizado una bandeja también reciclada de las mismas dimensiones que la base. La profundidad de esta debe ser mínimo de 5 cm para permitir el crecimiento natural de los brotes. La principal diferencia con la bandeja de la base es que debe ser transparente. Debe dejar pasar la luz solar y debe tener pequeños orificios de ventilación. Esta bandeja tendrá el aspecto que se muestra en la Figura 87.



*Figura 87. Envase utilizado para la cubierta del invernadero.*

En esta misma cubierta irá instalado el sistema de riego que se explicará en el siguiente punto.

- Conductos de riego.  
Para los conductos de riego se ha utilizado un tubo de PVC transparente de 6 mm de diámetro interior. La longitud de este tubo debe ser acorde con la distancia a la que se encuentre la bomba de agua y el depósito del invernadero. El conducto de riego se implanta en la cubierta superior creando una especie de malla para intentar conseguir un riego uniforme de la superficie de plantación. La parte de conducto que conforma la malla de riego posee unos orificios para la salida por goteo del agua. Este tubo, también puede obtenerse de reciclaje, ya que es un conducto utilizado normalmente en el ámbito médico para la inhalación de oxígeno o medicamentos a través de aerosoles. El tubo escogido para el conducto de riego se muestra en la Figura 88:



*Figura 88. Tubo de 6 mm utilizado como conducto de riego en el invernadero.*

- Depósito de agua.  
El depósito de agua debe encontrarse a un nivel de altura superior o igual que el invernadero, permitiendo la libre y correcta circulación del agua gracias a la ayuda de la fuerza de gravedad. Para este depósito, se ha elegido una botella de agua común de capacidad 1,5 L. Para la elección del tamaño del depósito hay que tener en cuenta dos factores:
  - Frecuencia de llenado del depósito. Dependiendo de la ubicación en la que se realice el proyecto atendiendo a la cantidad de agua que absorberá la plantación.
  - Estación del año en la que se desarrolle el proyecto; en verano será necesario un depósito de agua mayor ya que para mantener el invernadero en un estado óptimo es necesario una mayor cantidad de agua.
  
- Plantación, arena y semillas.  
Para el desarrollo de este proyecto se ha utilizado tierra especial de semillero de la marca Natura como se muestra en la Figura 89 para favorecer y acelerar el proceso de crecimiento de las semillas.



Figura 89. Tierra especial semillero utilizada para la plantación del invernadero. Natura.

Las semillas escogidas son semillas normales, cuyo crecimiento no es demasiado duradero, en este caso de perejil, albahaca y berro, pero pueden seleccionarse cualquier tipo de semillas que permitan ver progreso del proyecto rápidamente. La elección de ellas se ha realizado pensando en que los alumnos sean capaces de identificar los brotes cuando comiencen a crecer, tanto por la forma como por el olor, experimentando de esta manera la utilidad real del proyecto y su ventaja de uso en la vida real. En la Figura 90.

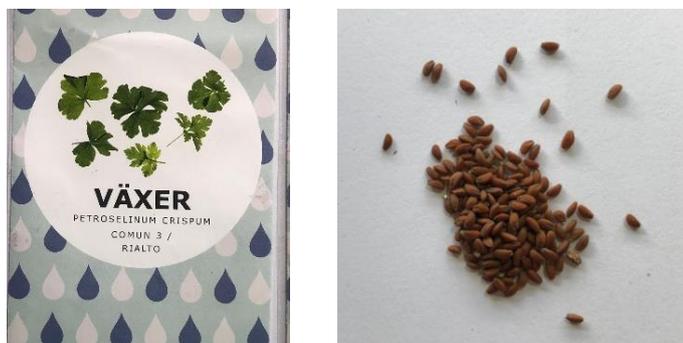


Figura 90. Semillas utilizadas para la plantación del invernadero.

## 4.4. Montaje del proyecto

El montaje del conjunto del proyecto se ha dividido en dos partes. Montaje del invernadero y programación y montaje de la placa principal y sensores y actuadores de control.

### 4.4.1. Montaje del hardware de control.

El hardware de control engloba todos los módulos *Easy-Plug*, la placa principal y los actuadores externos.

El montaje de las conexiones entre la placa principal y los módulos se lleva a cabo con la ayuda de los cables *Easy-Plug*, teniendo en cuenta la facilidad de esta parte del proyecto, contando con que cada módulo se identifica con una etiqueta de un color, que debe coincidir con el color de la etiqueta del puerto en la placa principal al que debe ir conectado. Cada color de etiqueta se relaciona con un tipo de puerto, digital de entrada-salida, analógico, u otro tipo de puerto, como se ha explicado en el apartado 4.1.1.

- Conexión módulos *Easy-Plug*. Se utilizarán un total de siete cables *Easy-Plug* para crear todas las conexiones a la placa como se muestra en la Figura 91.

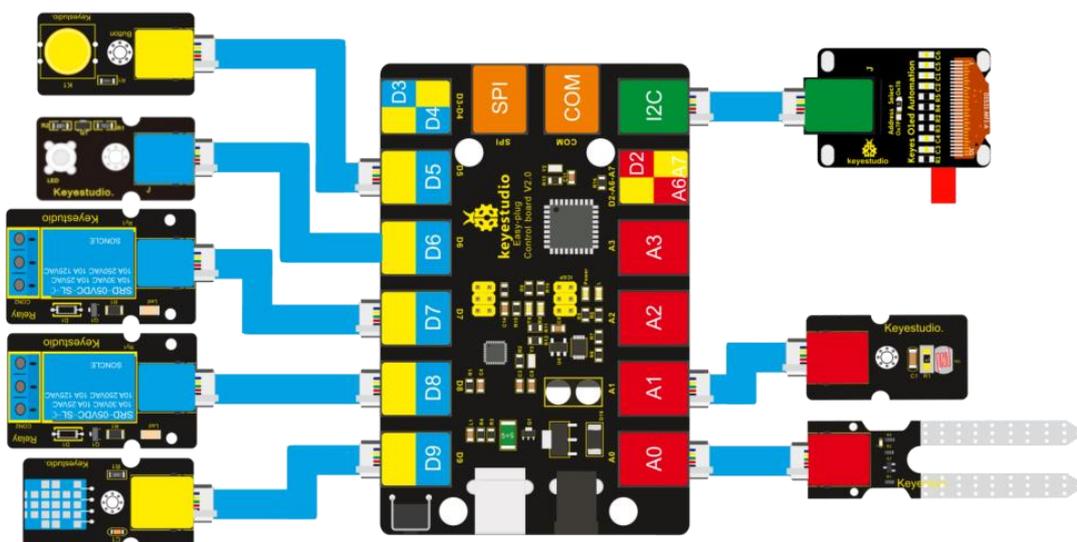


Figura 91. Esquema de conexión de los módulos *Easy-Plug* a la placa principal.

- Conexión de actuadores externos. Para la utilización de estos actuadores utilizaremos los módulos de relé, conectados a los puertos digitales D7 y D8, como se puede observar en la anterior Figura 91.

Por un lado, se realiza la conexión de la bomba de agua, cuyo relé irá conectado al pin D8. Las conexiones, se realizan teniendo en cuenta los contactos del relé. En este caso, realizaremos las conexiones entre los contactos C (Común) y NO (Normalmente abierto). De esta manera, el circuito permanece abierto hasta que la placa principal activa la salida D8 con un 1 digital, cerrando el circuito entre el contacto común y el contacto normalmente abierto, alimentando la bomba de agua y permitiendo el riego del invernadero. En la Figura 92 se muestra el esquema de conexión de este conjunto.

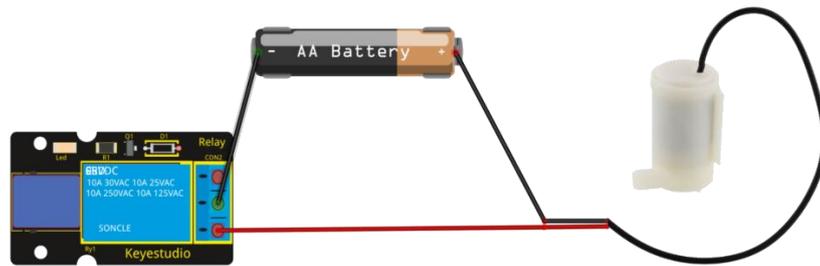


Figura 92. Circuito de montaje Módulo relé Easy-Plug y bomba de agua.

Por otro lado, se realiza la conexión del ventilador con el módulo de relé de la misma forma que la bomba de agua y utilizando los mismo contactos del relé. En la Figura 93 se muestra el esquema de conexión de este conjunto.

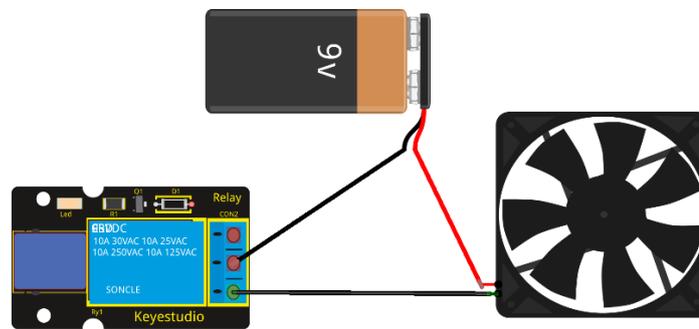


Figura 93. Circuito de montaje Módulo relé Easy-Plug y ventilador 5V DC.

Por último, dependiendo del módulo actuador de luminosidad que se escoja:

- En el caso en que se escoja el módulo de led *Easy-Plug*, simplemente se ha de conectar el pin D5 de la placa principal con el módulo de led.
- En el caso en que se escoja la tira de led para la iluminación, se ha de conectar el pin D5 de la placa principal con el módulo de conversión *Easy-Plug* y a continuación, realizar las conexiones de este con la tira de led. El pin digital de la placa debe ir conectado a la alimentación de la tira de leds y la masa de este con la masa del módulo de conversión.

#### 4.4.2. Diseño y desarrollo del software de control.

Para el desarrollo de este fragmento del proyecto, que engloba toda la parte de programación de este, se utiliza el programa Facilino Junior. Es la parte más exigente para los alumnos, al ser un campo aún por desarrollar en la enseñanza, pero gracias a la utilización de esta herramienta y el desarrollo de los módulos únicos y personalizados para este proyecto, se intenta que sea más lúdico y menos complicado para su desarrollo en las aulas.

Para la explicación de este apartado, se divide en las diferentes partes que forman la programación del programa:

- Programación del módulo OLED y su configuración.

Se utilizan las instrucciones para mostrar por la pantalla OLED los valores de humedad de la tierra, de luminosidad y de temperatura, haciendo uso del bloque de la categoría “pantallas” en tres ocasiones, que permite mostrar una cadena de caracteres y un número simultáneamente.

La cadena de caracteres indicará el factor ambiental en cada caso, y el campo numérico en este caso corresponderá con la variable de cada factor ambiental generada por su respectivo bloque. En la Figura 94 se muestra la combinación de estos tres bloques configurados cada uno para un factor ambiental.



Figura 94. Instrucciones de inicio y configuración de la pantalla OLED.

Estas tres instrucciones se mostrarán a la vez en la pantalla OLED. Para ello, se configura la posición y el tamaño de cada una de ellas.

Todas ellas se configuran con un tamaño pequeño, para que quepan a la vez en la pantalla, y como son tres instrucciones, cada una de ellas se pone a una altura. El resultado final mostrado en la pantalla OLED se muestra en la Figura 95.



Figura 95. Resultado instrucciones de inicio y configuración de la pantalla OLED.

Como se puede observar, los valores de humedad y luz varían entre 0 y 1023 al ser valores analógicos. Para solucionar este problema, opcionalmente se puede hacer uso del bloque de la categoría “matemáticas” que permite reescalar un valor dado entre los extremos indicados. Para este proyecto introducimos como valor mínimo el 0 y como valor máximo el 100, permitiendo así visualizar por la pantalla los valores de humedad y luminosidad en tanto por ciento. La introducción de este bloque en la combinación anterior quedaría como se observa en la Figura 96.



Figura 96. Instrucciones de inicio y configuración de la pantalla OLED con reescalado.

Obteniendo un resultado final en la pantalla OLED como el mostrado en la Figura 97.



Figura 97. Resultado instrucciones de inicio y configuración de la pantalla OLED con reescalado.

En el Código 1 se muestra el resultado del conjunto de instrucciones generadas como combinación de estos bloques.

```
#include <Adafruit_SSD1306.h>
#include <splash.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SPITFT.h>
#include <Adafruit_SPITFT_Macros.h>
#include <gxfont.h>

#include <Wire.h>
/** Global variables */
DHT sensorDHT11_9(9, DHT11);

/** Function declaration */
#define OLED_RESET 4
Adafruit_SSD1306 display(OLED_RESET);

void setup()
{
    display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
    display.clearDisplay();
    delay(250);
void loop()
{

    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.setCursor(1,1);
    display.print("Humedad: ");
    display.println(analogRead(A0));
    display.println(map(analogRead(A0), 0, 1023, 0, 100));

    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.setCursor(1,10);
    display.print("Luz: ");
    display.println(analogRead(A1));
    display.println(map(analogRead(A1), 0, 1023, 0, 100));
```

Código 1. Instrucciones de la programación del módulo OLED del proyecto.

- Programación de activación y desactivación de actuadores.

Seguidamente, se utilizan las instrucciones de alarma de factores ambientales de la categoría “*ambiente*” diseñadas para este proyecto. De esta manera, podemos seleccionar un valor máximo y un valor mínimo a partir de los cuales se producirá la activación o desactivación de los actuadores.

En primer lugar, se procede a la programación del bloque de alarma de humedad de tierra: Este bloque contiene 4 parámetros para configurar y dos campos para la introducción de instrucciones. Se configura el puerto al cual se conectará el módulo del higrómetro, en este proyecto, como indica la Figura 91, en el puerto del pin analógico A0. Se introducen también los valores máximo de humedad y mínimo para obtener el siguiente resultado:

- En el caso en que la humedad sea mayor que la humedad máxima establecida, se introducirán las instrucciones para la desactivación del relé de la bomba de agua.
- En el caso en que la humedad sea menor que la humedad mínima establecida, se introducirán las instrucciones para la activación del relé de la bomba de agua y la escritura del mensaje “BOMBA” en la pantalla OLED.

Con estos dos valores se consigue el funcionamiento de la bomba cuando la humedad sea menor del 10% y hasta que la humedad haya alcanzado al menos el 20%.

En la Figura 98 se muestra la combinación de estos bloques.

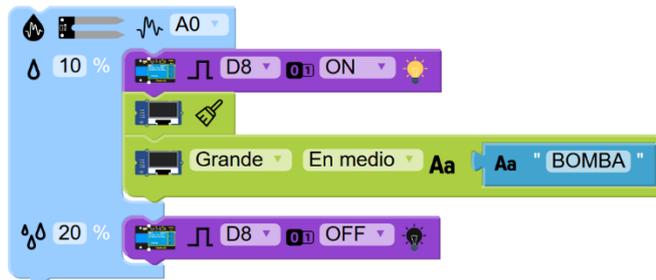


Figura 98. Instrucciones de condición para activación o desactivación de la bomba de agua.

Como se puede observar en la Figura 98, los bloques de activación o desactivación del relé deben ser configurados introduciendo el puerto al cual está conectado el relé de la bomba de agua.

También hay que considerar, que en el momento en el que la bomba se active por falta de humedad, la pantalla OLED contendrá los datos de la Figura 96, por tanto será necesario borrarlos antes de mostrar el texto "BOMBA" en dicha pantalla. Se ha configurado el tamaño del texto en tamaño grande y la posición centrada, para visualizar claramente que la bomba está en funcionamiento.

El resultado de la pantalla OLED sería el mostrado en la Figura 99.



Figura 99. Resultado de las instrucciones de activación de la bomba de agua.

El Código 2 mostrado a continuación corresponde con las instrucciones de la combinación de estos bloques en Arduino:

```
void setup() {  
  pinMode(A0, INPUT);  
}  
  
void loop() {  
  if (analogRead(A0) < 102.3) {  
    while (analogRead(A0) < 204.6) {  
      digitalWrite(8, HIGH);  
  
      display.display();  
      display.clearDisplay();  
      delay(250);  
  
      display.setTextSize(3);  
      display.setTextColor(WHITE);  
      display.setCursor(1, 10);  
      display.print("BOMBA");  
      display.display();  
      display.clearDisplay();  
      delay(250);  
    }  
  } else {  
    digitalWrite(8, LOW);  
  }  
}
```

Código 2. Instrucciones generadas por el bloque alarma de humedad de tierra.

En el caso de la programación de este bloque de alarma, también se puede plantear la posibilidad de utilizar el bloque que contiene a su vez la lectura del estado de un pulsador digital.

La programación de este bloque se realizaría siguiendo el mismo proceso que en la alternativa anterior, pero añadiendo el paso de la configuración del pin de conexión del pulsador, en este paso, el pin digital 5, como se puede observar en la Figura 100.

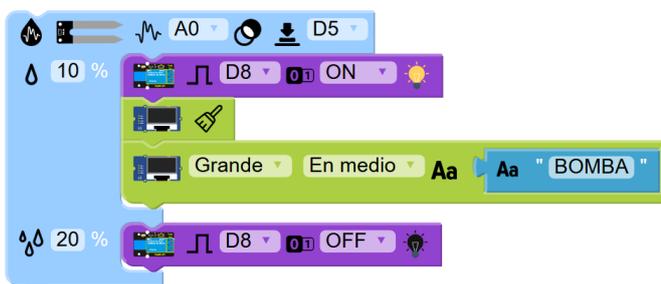


Figura 100. Instrucciones de condición para la activación o desactivación de la bomba de agua con pulsador.

El Código 3 mostrado a continuación corresponde con las instrucciones de la combinación de estos bloques en Arduino:

```
void setup() {
  pinMode(5, INPUT);
  pinMode(A0, INPUT);
}

void loop() {
  if((analogRead(A0)<102.3) || (digitalRead(5)==LOW)) {
    while((analogRead(A0)<204.6) || (digitalRead(5)==LOW)) {
      digitalWrite(8, HIGH);

      display.display();
      display.clearDisplay();
      delay(250);

      display.setTextSize(3);
      display.setTextColor(WHITE);
      display.setCursor(1,10);
      display.print("BOMBA");
      display.display();
      display.clearDisplay();
      delay(250);
    }
  }else{
    digitalWrite(8, LOW);
  }
}
```

*Código 3. Instrucciones generadas por el bloque alarma de humedad de tierra y estado de botón.*

En segundo lugar se programa el bloque de alarma de luminosidad siguiendo los mismos pasos que en el bloque anterior:

Se configuran los tres parámetros del bloque, el puerto en el que estará conectado el módulo de la fotorresistencia, en este caso el puerto del pin analógico A1, y los valores máximo y mínimo de luminosidad que nos permiten realizar unas acciones u otras.

Para la luminosidad, las acciones a realizar serán:

- En el caso en que la luminosidad sea mayor que la máxima, se introducirán las instrucciones para la desactivación del funcionamiento del LED o de la tira de leds.
- En el caso en que la luminosidad sea menor que la mínima, se introducirán las instrucciones para la activación del funcionamiento del led o de la tira de leds y la escritura del mensaje "LED" en la pantalla OLED.

Con estos dos valores se consigue el funcionamiento de la iluminación cuando el nivel de luz sea menor del 20% y hasta que la humedad haya alcanzado al menos el 20%.

En la Figura 101 se muestra la combinación de estos bloques.

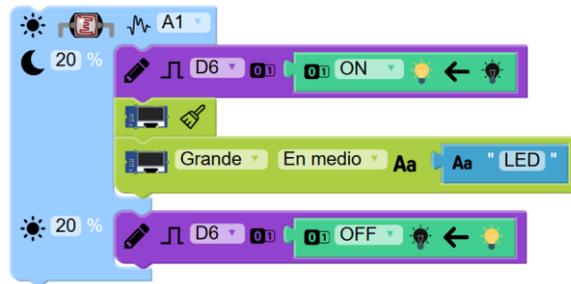


Figura 101. Instrucciones de configuración para la activación o desactivación de los LED's.

Se deben tener en cuenta los mismos conceptos que para la configuración del bloque de humedad de la tierra, como el pin de conexión del relé a la placa principal así como el borrado de la pantalla OLED antes de la escritura de la palabra "LED".

El resultado de la pantalla OLED sería el mostrado en la Figura 102.

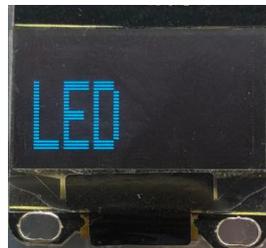


Figura 102. Resultado de las instrucciones de activación de la iluminación con leds.

El Código 4 mostrado a continuación corresponde con las instrucciones de la combinación de estos bloques en Arduino:

```
void setup() {  
  pinMode(A1, INPUT);  
}  
  
void loop() {  
  if(analogRead(A1)<204.6){  
    digitalWrite(6,HIGH);  
  
    display.display();  
    display.clearDisplay();  
    delay(250);  
  
    display.setTextSize(3);  
    display.setTextColor(WHITE);  
    display.setCursor(1,10);  
    display.print("LED");  
    display.display();  
    display.clearDisplay();  
    delay(250);  
  }  
  else{  
    digitalWrite(6,LOW);  
  }  
}
```

Código 4. Instrucciones generadas por el bloque alarma de luminosidad.

Por último, se lleva a cabo la configuración del bloque de alarma de temperatura, de la misma forma que los dos bloques anteriores.

Se configuran los tres parámetros de este bloque, en este caso, el módulo DHT11 de sensor de temperatura estará conectado al puerto del pin digital D9. También se configuran los valores de temperatura máximo y mínimo para la realización de las siguientes acciones:

- En el caso en que la temperatura sea mayor que la temperatura máxima establecida, se introducirán las instrucciones para la activación del relé del ventilador y la escritura de la palabra "FAN" (ventilador en inglés).
- En el caso en que la temperatura sea menor que la temperatura mínima establecida, se introducirán las instrucciones para la desactivación del relé del ventilador.

Con estos dos valores se consigue el funcionamiento de la ventilación cuando la temperatura alcance los 28,5 °C y hasta que esta baje de los 27 °C.

En la Figura 103 se muestra la combinación de estos bloques.

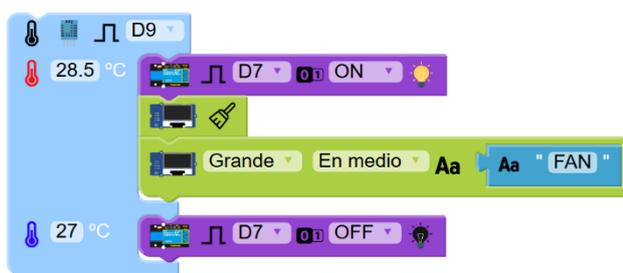


Figura 103. Instrucciones de configuración para la activación o desactivación del ventilador.

Se deben tener en cuenta los mismos conceptos que para la configuración de los bloques de alarma anteriores, como el pin de conexión del relé a la placa principal así como el borrado de la pantalla OLED antes de la escritura de la palabra "FAN".

El resultado de la pantalla OLED sería el mostrado en la Figura 104.



Figura 104. Resultado de las instrucciones de activación del ventilador.

El Código 5 mostrado a continuación corresponde con las instrucciones de la combinación de estos bloques en Arduino:

```
void setup() {
  sensorDHT11_9.begin();
}

void loop() {
  if(sensorDHT11_9.readTemperature() >28.5){
    while(sensorDHT11_9.readTemperature() >27){
      digitalWrite(7,HIGH);

      display.display();
      display.clearDisplay();
      delay(250);

      display.setTextSize(3);
      display.setTextColor(WHITE);|
      display.setCursor(1,10);
      display.print("FAN");
      display.display();
      display.clearDisplay();
      delay(250);
    }
  }else{
    digitalWrite(7,LOW);
  }
}
```

*Código 5. Instrucciones generadas por el bloque alarma de luminosidad.*

El resultado de la combinación de todas estas instrucciones en la interfaz Facilino Junior da lugar a la siguiente imagen mostrada en la Figura 105.

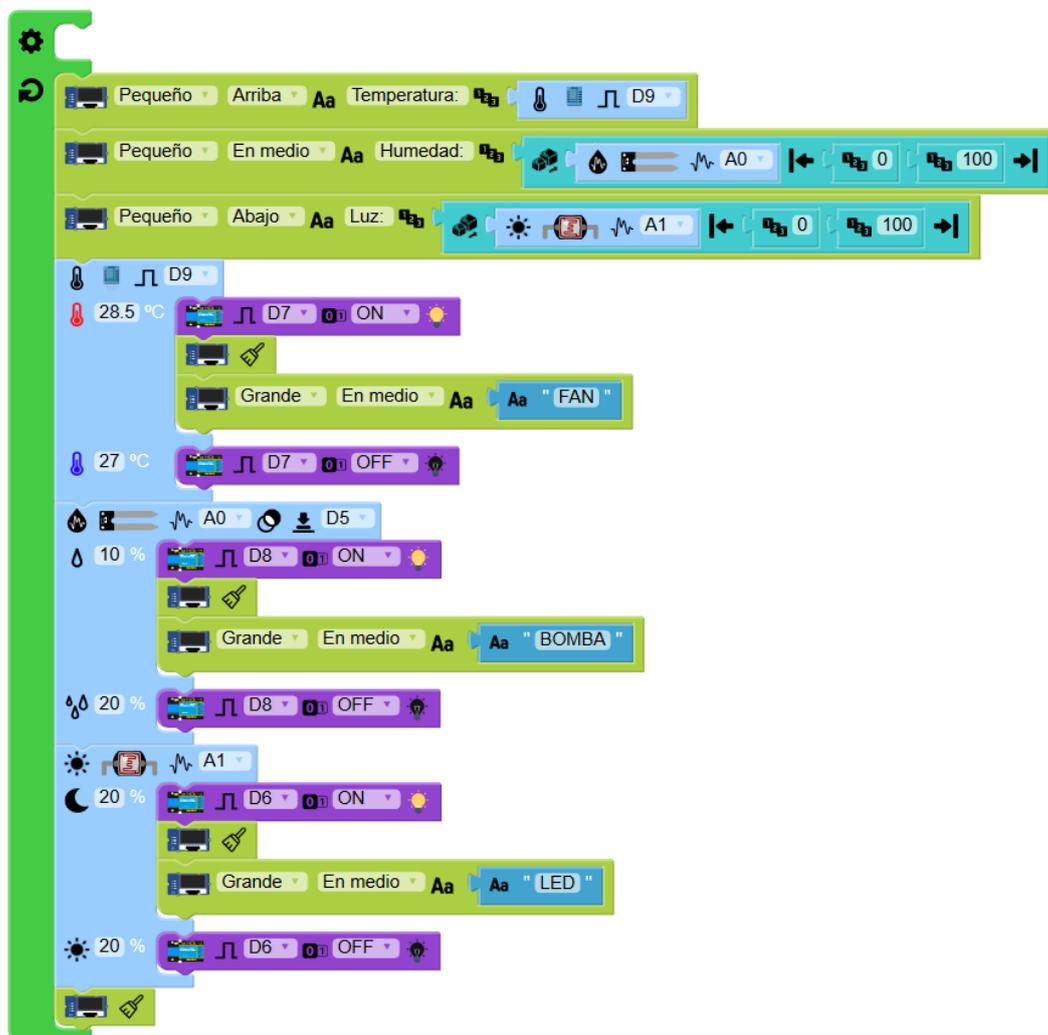


Figura 105. Resultado final del programa tras la combinación de los bloques en Facilino Junior.

Como se observa en la Figura 105, en la parte final del programa, se ha introducido un bloque de borrado de la pantalla OLED, con la finalidad que en el caso en que alguno de los actuadores esté en funcionamiento, y por tanto mostrando mensajes de “BOMBA”, “FAN”, o “LED” en la pantalla OLED, estos desaparezcan al final de la ejecución de las instrucciones, permitiendo observar el estado de las variables en la configuración establecida (Figura 97) durante medio segundo hasta que vuelva aparecer alguna de las anteriores palabras.

El conjunto de todas las instrucciones generadas por Facilino Junior dan lugar al código en lenguaje Arduino contenido en el **ANEXO I**.

#### 4.4.3. Montaje del invernadero.

El montaje del invernadero es muy sencillo, orientado a su fabricación por alumnos de los cursos de aplicación del proyecto y se divide en X etapas:

##### 1. Plantación de las semillas.

En primer lugar, se ha de llevar a cabo la plantación de las semillas. Para ello, se utilizará el envase opaco, que conformará la base del invernadero. Se esparce una fina capa de tierra de semillero de unos dos dedos de altura y se humedece ligeramente como se muestra en la Figura 106.



*Figura 106. Llenado del envase opaco con la tierra de semillero.*

Seguidamente, se realizan orificios en ella para introducir las semillas escogidas como se muestra en la Figura 107.



*Figura 107. Orificios para las semillas.*

Es importante que las semillas no se introduzcan a demasiada profundidad, para que puedan salir fácilmente a la superficie cuando comiencen su crecimiento, ver Figura 108.



*Figura 108. Introducción de las semillas.*

Tras la introducción de las semillas en los orificios, se deben cubrir sin presionar la superficie y regar esta vez de forma un poco más abundante tal y como se observa en la Figura 109.



Figura 109. Tapado de orificios y regado del invernadero.

## 2. Construcción de la cubierta con el conducto de riego y ventilación.

Para la construcción de la cubierta, se utilizará el envase transparente, con la intención que la luz pueda atravesarlo y llegue así a las semillas.

Para comenzar, se realiza un orificio central de 1 cm de radio aproximadamente para la entrada de aire con la ventilación y dos orificios más de unos 3 mm para la fijación del ventilador en uno de los extremos de la cubierta.

A continuación, se realizan otros 3 orificios en cada lateral de la cubierta de unos 8 mm de diámetro y separados 5 cm entre sí. A través de estos orificios será introducido el conducto de 6 mm de diámetro de riego por goteo.

En la Figura 110 se muestra un esquema de la posición exacta de los orificios a realizar en la cubierta transparente.

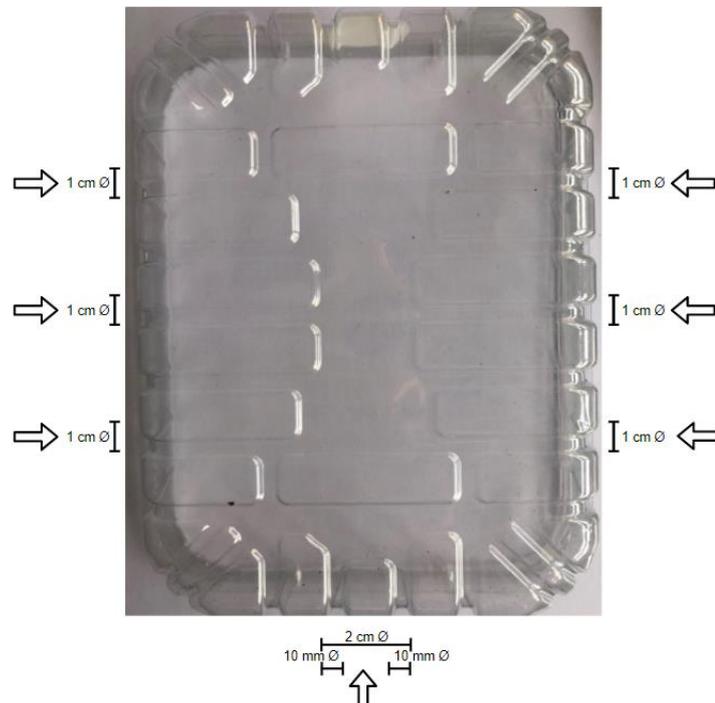
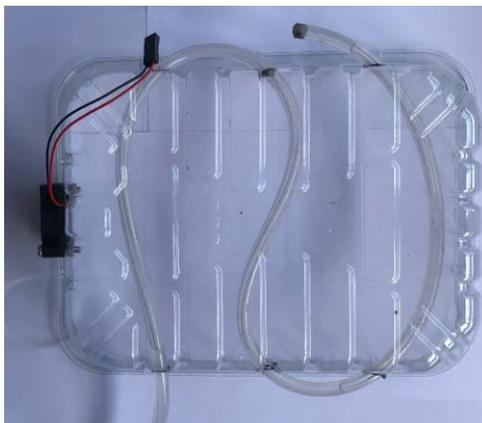


Figura 110. Indicación de posición de los orificios en cubierta superior

Seguidamente, se prepara el conducto de riego, se utilizará 1 metro de conducto dejando en uno de los extremos un cabezal de aerosoles que irá conectado a la bomba. Se introduce el conducto por los orificios de la cubierta superior creados anteriormente para ello, formando una malla como se aprecia en la Figura 111. También se procede al atornillado del ventilador en la pared de la cubierta, utilizando los orificios de 10 mm de diámetro para tal fin.



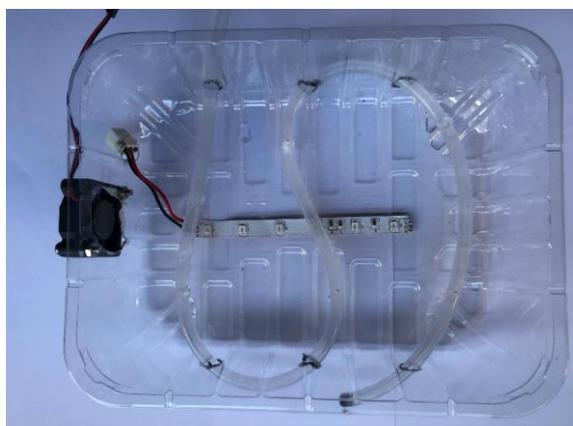
*Figura 111. Imagen del conducto introducido en la cubierta superior.*

En la Figura 112 se muestra el resultado de la colocación del conducto de riego y el ventilador.



*Figura 112. Resultado final de la colocación del conducto y del ventilador en la cubierta superior.*

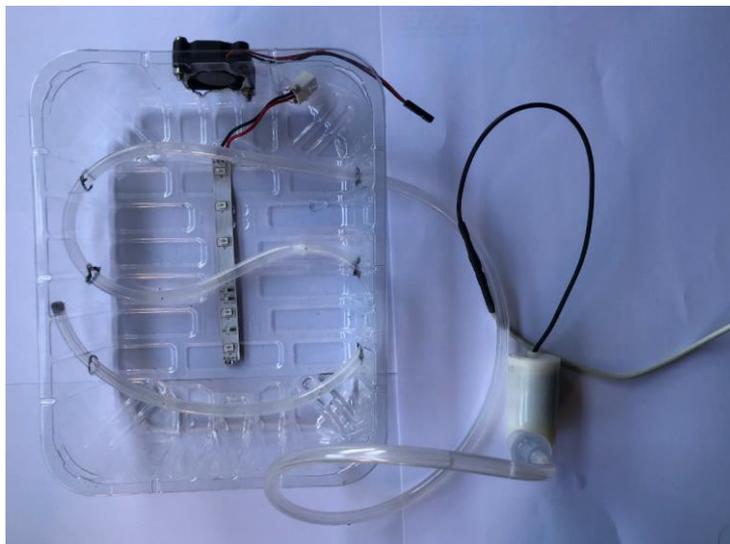
Posteriormente, se corta un trozo de la tira de led de unos 10-12 cm y se coloca en el techo de la cubierta del invernadero anclada con la cinta adhesiva que viene incorporada en la tira, como se muestra en la Figura 113.



*Figura 113. Resultado colocación de la tira de leds.*

Con el conducto de riego, la tira de leds y el ventilador en su posición final, se procede a su incisión del conducto para el riego por gotas. Se realizarán tres orificios en cada línea de la malla con una separación mínima de 3 cm para distribuir uniformemente el riego.

Para finalizar, se procede al cierre del circuito y su conexión a la bomba de agua, como se visualiza en la Figura 114.



*Figura 114. Cubierta finalizada con conexión a bomba de agua.*

Es necesario tener en cuenta algunas consideraciones durante la realización de esta parte de la construcción del invernadero:

- La posición de los orificios puede variar, no es necesaria la máxima precisión en la posición de estos.
- Es necesaria la realización de un orificio para el paso de los cables de alimentación y control de la tira de leds, cuya posición es variable y debe hacerse donde mejor convenga.
- Es necesario el uso de tijeras puntiagudas para la realización de los orificios de la cubierta superior (Figura 110), pero para hacerlos más precisos y más fácilmente, se puede utilizar un soldador viejo para fundir el plástico y perforar la cubierta.
- Es preciso taponar el extremo del conducto de riego que no va a estar conectado a la bomba de manera correcta, para evitar fugas y posibles problemas futuros. Para ello, se puede doblar el extremo una vez introducido el conducto o para mayor seguridad, se puede taponar con cola caliente.
- Al conectar la bomba de agua al circuito completo de riego, se debe cebar el conducto, evitando que entre aire que más adelante, durante el funcionamiento dificulte el empuje de agua a la bomba.

#### 4.5. Diseño guion del proyecto

Con la finalidad de llevar a cabo la implantación de este proyecto en las aulas del último ciclo de Educación Primaria como se ha indicado a lo largo del presente Trabajo Fin de Grado, se ha desarrollado un guion que permite a los alumnos tener una visión global de las partes que conforman este proyecto.

Este guion, junto con el apoyo y una explicación prevista por parte del profesorado son suficientes para la realización del proyecto y la comprensión de todos los contenidos que se potencian durante su implementación.

Para la elaboración de este guion, se han intentado utilizar iconos y la menor cantidad de texto posible, con la finalidad de generar mayor interés en los alumnos, eliminando detalles demasiado complejos que podrían confundir o crear problemas entre los alumnos.

En el **ANEXO II** de este Trabajo Fin de Grado se incluye dicho documento.

## 5. RESULTADOS DEL PROYECTO

### 5.1. Consideraciones a tener en cuenta

En el momento de la implantación del proyecto en las aulas, es necesario tener en cuenta una serie de consejos y consideraciones:

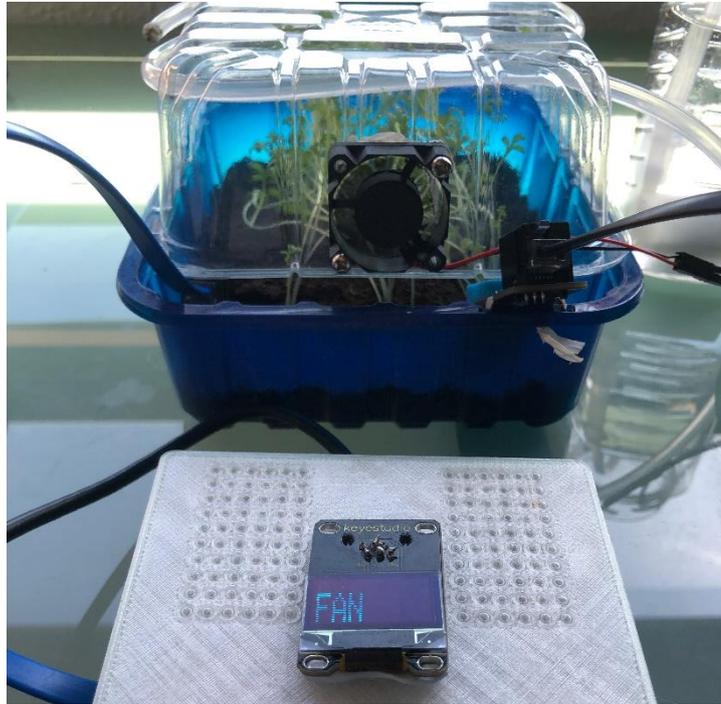
- La práctica de este proyecto se debe realizar en todo momento bajo la supervisión del profesorado.
- La parte de construcción de la cubierta superior de riego del invernadero puede ser opcional para los alumnos dependiendo de su curso y conocimientos. Esta parte de construcción tiene trabajos con cierto grado de peligrosidad o dificultad para ellos. Es recomendable proporcionar al alumno las cubiertas superiores con los orificios para la introducción del conducto y para el anclaje del ventilador ya realizados, asegurándose así un correcto funcionamiento de este sistema y evitando el uso por parte de los alumnos de objetos punzantes (en el caso de realizarse con unas tijeras) o de objetos con mayor peligrosidad (en el caso de realizarse con un soldador).
- En cambio, la parte de plantación de las semillas y llenado de la base con tierra de semillero es recomendable que sea realizada por los alumnos. Esta tarea les ayuda desarrollar las competencias explicadas en el anterior apartado 4.3 Invernadero.
- En la parte del proyecto correspondiente al diseño y desarrollo del software de control, alumno debe tener más libertad para poder impulsar las competencias de programación, lógica, pensamiento computacional, y las demás expuestas en el apartado 2.2 Robótica en la educación.
- El proyecto puede evolucionar con la adición de módulos de sensores y actuadores opcionales, aumentando proporcionalmente la dificultad de este y por tanto las posibilidades de implantación en la Educación en general. Por tanto la implantación de este proyecto en la Educación puede evolucionar desde el último ciclo de Educación Primaria hasta el primer ciclo de Educación Secundaria Obligatoria, es decir, en primero y segundo, enmarcándose en la asignatura optativa Tecnología.

### 5.2. Funcionamiento del proyecto

Tras llevar a cabo todas las partes del montaje del proyecto, se obtiene un conjunto de hardware como el mostrado en la Figura 115.

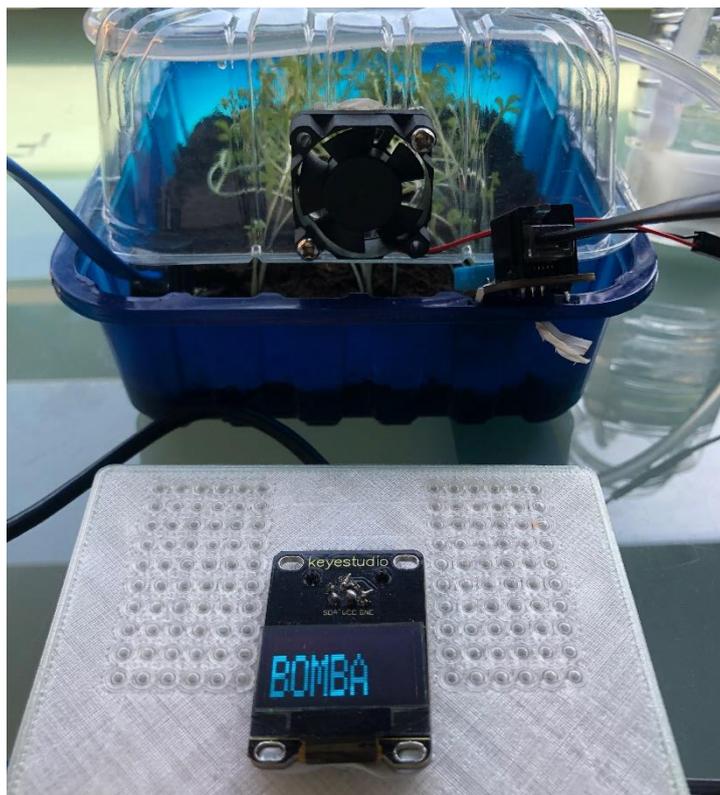


- Para el caso en que la temperatura es mayor que la temperatura máxima programada, el ventilador comienza a funcionar y en la pantalla OLED se muestra el texto "FAN" y el ventilador comienza a funcionar con la finalidad de reducir la temperatura interior del invernadero. La temperatura es captada gracias al sensor DHT11 de temperatura que se encuentra situado en el interior del invernadero, a la derecha del ventilador. En la Figura 116 se pueden observar todos los módulos y partes implicadas para este accionamiento.



*Figura 116. Estado de ventilación del invernadero automatizado.*

- Para el caso de humedad de tierra menor a la programada, la bomba comienza a funcionar regando por goteo el invernadero, y la pantalla OLED muestra la palabra “BOMBA” y la bomba de agua comienza a transportar agua desde el depósito hasta el conducto para que el riego por goteo aumente el índice de humedad de la tierra. Dicho índice es captado por el sensor de humedad de tierra posicionado a la izquierda del ventilador y enterrado superficialmente en la tierra de semillero del invernadero. En la Figura 117 y en la Figura 118 se puede observar con más detalle el montaje y posición de todos los módulos involucrados en este accionamiento, así como el goteo en detalle de la instalación.

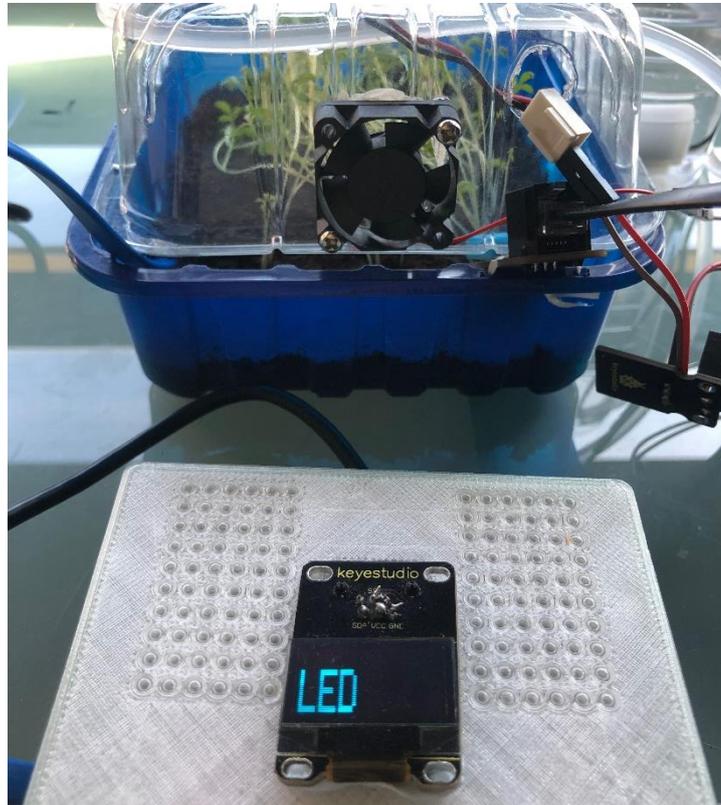


*Figura 117. Estado de riego por goteo del invernadero automatizado.*

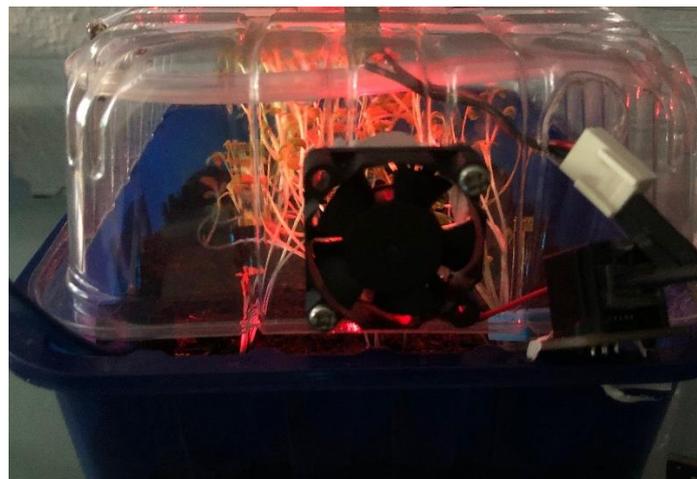


*Figura 118. Detalle de riego por goteo del invernadero automatizado.*

- Para el caso de luminosidad menor que la mínima programada, en la pantalla OLED se muestra la palabra "LED" y se enciende la tira de led fijos hasta que la luminosidad natural vuelva a ser la suficiente para el crecimiento de las plantas. En la Figura 119 y en la Figura 120 donde se muestran todos los módulos necesarios para el accionamiento y detección de luminosidad con más detalle, la tira de leds en funcionamiento.



*Figura 119. Estado de iluminación del invernadero automatizado.*



*Figura 120. Detalle de iluminación del invernadero automatizado.*

Finalmente, en la Figura 121 se muestra la progresión en el crecimiento de las plantas del invernadero tras su cuidado con la automática diseñada y desarrollada en este Trabajo Fin de Grado:



*Figura 121. Evolución del crecimiento del invernadero con la automatización desarrollada en el proyecto.*

### 5.3. Acceso a documentos

Todos los documentos para la implementación de este proyecto en la Educación son abiertos y accesibles para todo el mundo.

El Software Facilino Junior con las instrucciones de los bloques personalizados para este proyecto junto con el guion para la realización del proyecto y los planos están disponibles en la plataforma DIY (*Do It Yourself*) maker.pro<sup>9</sup>. Esta plataforma on-line permite la explicación de proyectos por los usuarios además de datos e información sobre los materiales utilizados, el código a desarrollar o las piezas a imprimir o diseñar como es el caso de este proyecto.

No obstante, en el **ANEXO III** se han incluido las instrucciones para la creación de los diferentes módulos utilizados en el proyecto en lenguaje JavaScript.

---

<sup>9</sup> Proyecto invernadero: <https://maker.pro/arduino/projects/mini-easy-plugprogramming-automated-greenhouse>

## 6. CONCLUSIÓN Y MEJORAS

Tras la realización de este Trabajo Fin de grado y su implementación con el hardware y software indicados, se puede corroborar que es un proyecto fácilmente realizable por los alumnos a los que va destinado.

Este proyecto permite a los alumnos adaptarse a las nuevas tecnologías que día a día van aumentando en importancia. En los próximos años, la tecnología va a ser indispensable en todos los ámbitos de la vida, facilitando tareas a los seres humanos, por lo que el conocimiento sobre este campo en las nuevas generaciones va a ser imprescindible.

Es por ello, que la educación debe evolucionar acorde con las nuevas tendencias tecnológicas. La introducción de estos conocimientos, pensamientos y habilidades en los planes de estudios conforma un papel clave en la formación de las nuevas generaciones.

El desarrollo de este proyecto ha cumplido con todos los objetivos establecidos al inicio de esta memoria, superando las principales barreras de entrada por las que este tipo de proyecto de robótica y tecnológicos no consiguen afianzarse en los planes educativos actuales.

- Barrera económica. El proyecto se ha desarrollado con materiales de bajo coste, y en la medida de lo posible reciclados, abaratando así el precio total de realización de este.
- Sencillez, tanto para los alumnos en el proceso de desarrollo como para el profesorado a la hora de impartir los conceptos y conocimientos necesarios
- Robustez desde la perspectiva de la durabilidad del proyecto gracias al diseño e impresión de la carcasa 3D protectora, consiguiendo alargar la vida útil de este haciéndolo más atractivo aún para los centros educativos.
- Robustez desde el punto de vista del transporte de este durante la realización gracias a la tecnología de conexión *Easy-Plug*.
- Posicionamiento en el plan de estudios, consiguiendo enmarcar el proyecto y sus contenidos en asignaturas pertenecientes al plan de estudios actual.

Indirectamente, gracias a la temática secundaria del proyecto, se consigue inculcar en los alumnos conciencia medioambiental, otro papel fundamental en la educación actual para la corrección de comportamientos y toma de responsabilidades con la finalidad de hacer un uso responsable de los recursos y del medio ambiente.

A continuación, se exponen una serie de posibles mejoras a implementar en el proyecto, aumentando así nivel educativo y las edades de implementación abarcado con este y teniendo en cuenta las posibilidades que la placa es capaz de afrontar gracias a los puertos y pines digitales disponibles que no se han utilizado:

- Tiras Led programables. Como se ha desarrollado en la memoria, el proyecto contiene un bloque de alarma para luminosidad. Este bloque permite al igual que el bloque de humedad de tierra o el bloque de temperatura, activar una salida digital para el funcionamiento de un actuador.

En este proyecto, este bloque de alarma activa un módulo de LED simple que se enciende cuando la luminosidad es menor que el valor mínimo establecido como alarma.

Un proyecto futuro, que a la vez dota al proyecto de algo más de dificultad, consta de la incorporación de una tira de LED RGB completa programable en color e intensidad. La instalación de esta sería en el techo de la cubierta superior del invernadero y su activación sería simplemente a través de un pin digital de la placa *Easy-Plug*.

En el caso de la incorporación de esta tira de LED, el bloque de alarma de luminosidad sería combinado con bloques de la categoría “luz”, para configurar la cantidad de LED, el color y la intensidad de estos. En la Figura 122 se muestra como quedaría este bloque.

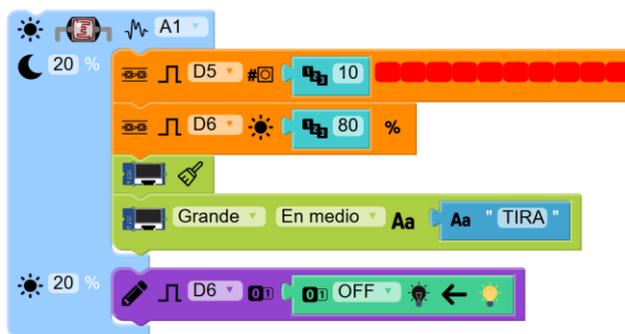


Figura 122. Bloque de alarma de luminosidad para activación y desactivación de tira de LED.

- Placas solares. Una mejora considerable para la alimentación del invernadero sería sustituir la batería de alimentación portátil por unas pequeñas placas solares. La placa solar, debería ser de 5V y 2,5 W de potencia, proporcionando una intensidad de 500 mAh, siendo suficiente para el funcionamiento de la placa *Easy-Plug* y los módulos conectados. La incorporación de esta mejora contribuye a un desarrollo sostenible del proyecto siendo 100% autosuficiente, sin necesidad de recargar las baterías de alimentación. Como aspecto negativo, esta mejora conlleva un aumento considerable en el desarrollo del proyecto, así como un aumento de presupuesto. Estas placas solares necesitan complementariamente un regulador que permita mantener un nivel constante de tensión como por ejemplo el regulador L7805 CV. La incorporación de este componente implica el montaje de este, necesitando más materiales y conocimientos para su incorporación en el proyecto.
- Unificación de las fuentes de alimentación. Un ajuste posible para realizar en el proyecto es la sustitución de la *powerbank* y todas las baterías de alimentación de los actuadores externos por una batería portátil que permita la salida de tensión a 5V y 3V. De esta manera, se podrá alimentar todos los módulos correctamente con una sola fuente de alimentación.
- Un servomotor que accione un pequeño toldo, evitando así la incisión directa de la luz solar en los brotes durante las horas de mayor incidencia. Con esta medida, se puede evitar que la planta sufra daños y pueda llegar a secarse por completo.
- Un sensor de lluvia, encargado de apagar la bomba de agua y accionar otro servomotor que permita la apertura de la cubierta superior del invernadero, que en caso de encontrarse en exterior, sería regado de forma natural.

## 7. BIBLIOGRAFÍA

- [1] P. A. López Ramírez y H. Andreade Sosa, «Aprendizaje de y con robótica, algunas experiencias,» vol. I, nº 37, pp. 43-63, 2013.
- [2] A. C. G. D. A. F. K. E. Stefania Bocconi, «Joint Research Centre,» 2016. [En línea]. Available: [http://publications.jrc.ec.europa.eu/repository/bitstream/JRC104188/jrc104188\\_computhinkreport.pdf](http://publications.jrc.ec.europa.eu/repository/bitstream/JRC104188/jrc104188_computhinkreport.pdf). [Último acceso: 06 05 2019].
- [3] J. Malec, Lund, Suecia, 2001.
- [4] P. Toys, «[www.primotoys.com](http://www.primotoys.com),» Primo HQ, Unit 19, 2019. [En línea]. Available: <https://www.primotoys.com/es/>. [Último acceso: 27 05 2019].
- [5] J. Valvano , «The IEEE Robotics and Automation Society (RAS) at The University of Texas at Austin Proudly: BlastyRAS,» Austin, Texas, 2007.
- [6] Consellería de Educación, Cultura y Deporte y P. d. I. G. Valenciana, «Decreto 108/2014, de 4 de Julio del Consell,» *Diario Oficial de la Generalitat Valenciana*, nº 7311, pp. 16325-16694, 07 07 2014.
- [7] C. d. P. y. P. d. G. C. d. Madrid, «Decreto 89/2014,» vol. Anexo III, pp. 90-91, 24 07 2014.
- [8] S. Relay, *Datasheet SRD-05VDC-SL-C Relay*, Zhejiang, China.
- [9] S. Systech, *Datasheet Pantalla OLED SSD1306*, 2008.
- [10] K. McCree, «The action Spectrum, Absorptance and Quantum Yield of Photosynthesis in crop plants,» Texas, 1970.
- [11] I. GitHub, «GitHub,» Microsoft, [En línea]. Available: [https://github.com/roboticafacil/facilino\\_junior](https://github.com/roboticafacil/facilino_junior).
- [12] G. Developers, «Developers Google Blockly,» Google, 20 09 2018. [En línea]. Available: <https://developers.google.com/blockly/>. [Último acceso: 30 05 2019].
- [13] Freepik, «Flaticon,» Freepik Company, 2013. [En línea]. Available: <https://www.flaticon.es/>. [Último acceso: 2019].
- [14] J. Tomás Palacín, «Innova Spain,» 11 02 2019. [En línea]. Available: <https://www.innovaspain.com/informe-samsung-smart-school-ministerio-de-educacion/>. [Último acceso: 05 05 2019].
- [15] M. B. Industries, «<https://www.thingiverse.com/>,» [En línea].





UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSIDAD POLITÉCNICA DE VALENCIA  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE DISEÑO  
GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

---

**Diseño y control de un Invernadero  
automatizado de bajo coste adaptado para  
su desarrollo en educación primaria.**

**Design and control of a low-cost  
automated Greenhouse adapted for its  
development in primary education.**

---

## **2. PLANOS**

**AUTOR:** ALBERTO DELGADO ROMERO

**TUTOR:** LEOPOLDO ARMESTO ÁNGEL

VALENCIA, JUNIO 2019



Índice

**1. Plano de caja para placa *Easy-Plug***

**2. Plano de tapa de caja para placa *Easy-Plug***

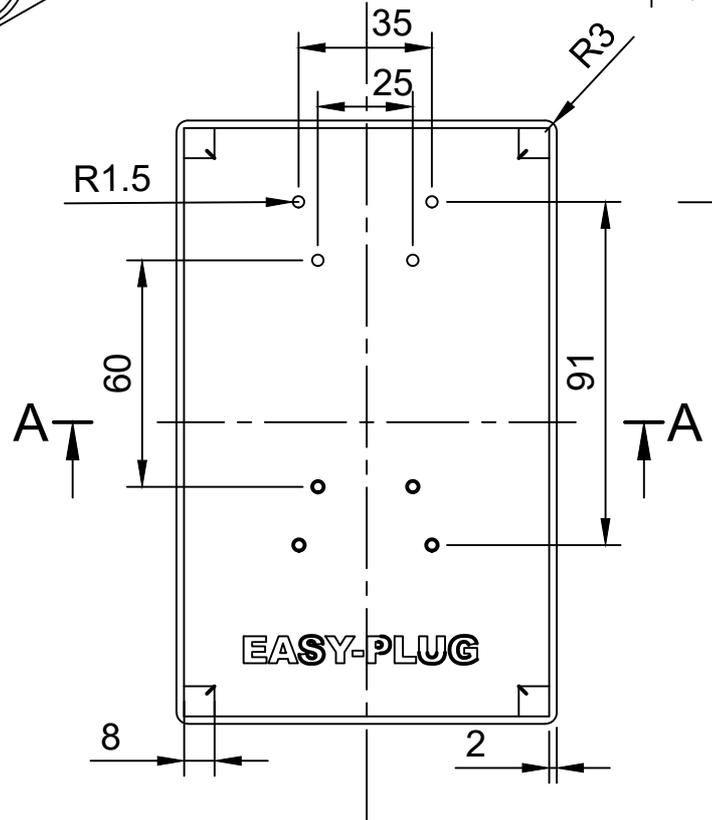
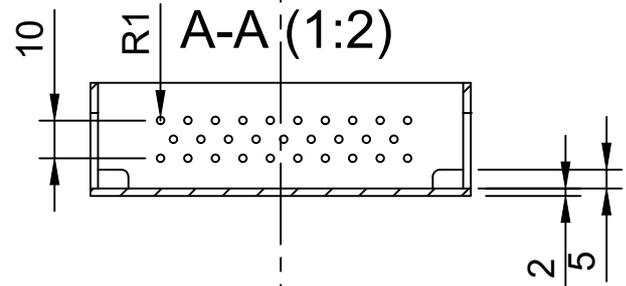
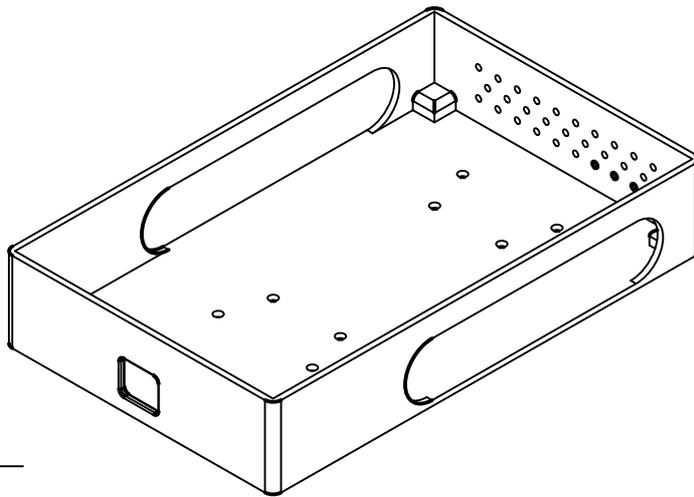
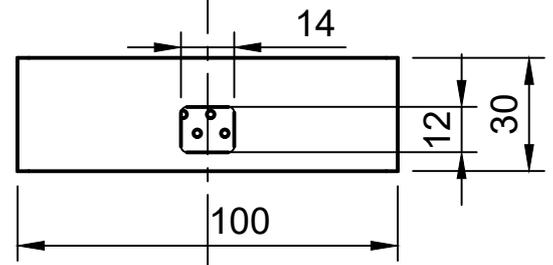
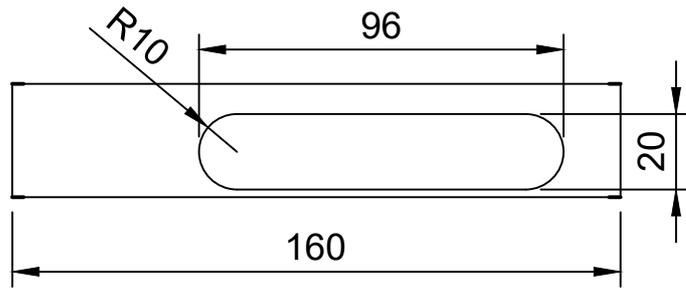
**3. Plano de conjunto**

Para el presente Trabajo Fin de Grado, y para la implementación del proyecto del invernadero en las aulas de Educación Primaria, se han diseñado 2 piezas para la protección y la colocación de la placa principal y los módulos *Easy-Plug*.

Ambas piezas han sido diseñadas con el programa Fusion 360, de la empresa de software *autoDesk*, y están disponibles de forma libre en *Thingiverse*<sup>10</sup>, un sitio web dedicado a la publicación e intercambio de archivos de diseño digital.

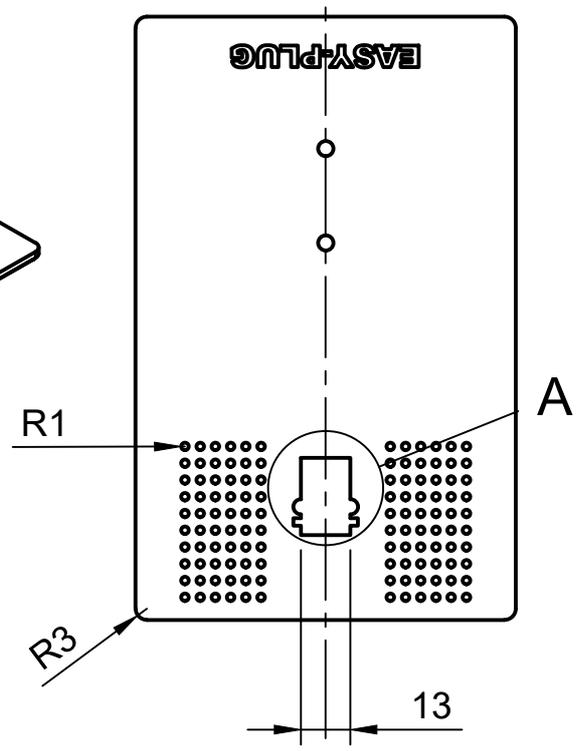
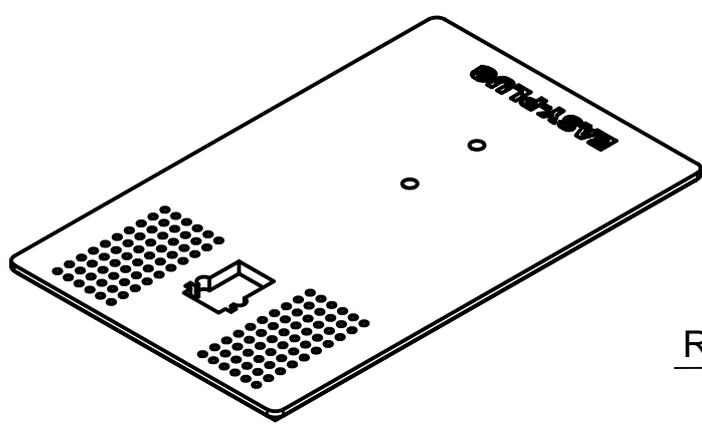
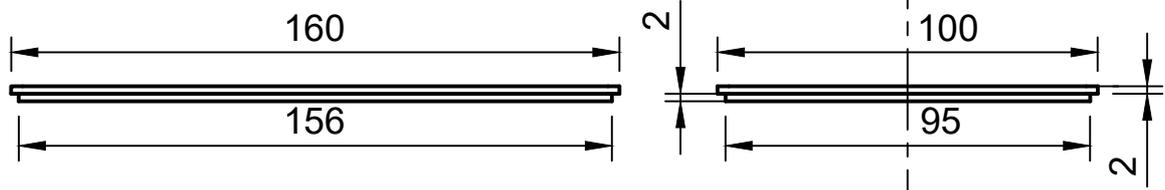
---

<sup>10</sup> Sitio web Thingiverse: <https://www.thingiverse.com/>, piezas del proyecto: <https://www.thingiverse.com/betodr4/designs>

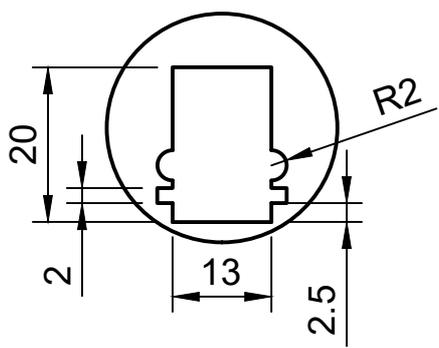


|   |                     |  |                  |              |
|---|---------------------|--|------------------|--------------|
| Scale<br>1:2  | Technical reference | Created by<br>Alberto Delgado 15/06/2019 | Approved by      |              |
|  UNIVERSITAT POLITÈCNICA DE VALÈNCIA<br><br><br>Escuela Técnica Superior de Ingeniería del Diseño |                     | Document type                            | Document status  |              |
|   |                     | Title<br>Caja placa Easy-Plug            | DWG No.<br>ES001 |              |
|   |                     | Rev.                                     | Date of issue    | Sheet<br>1/3 |



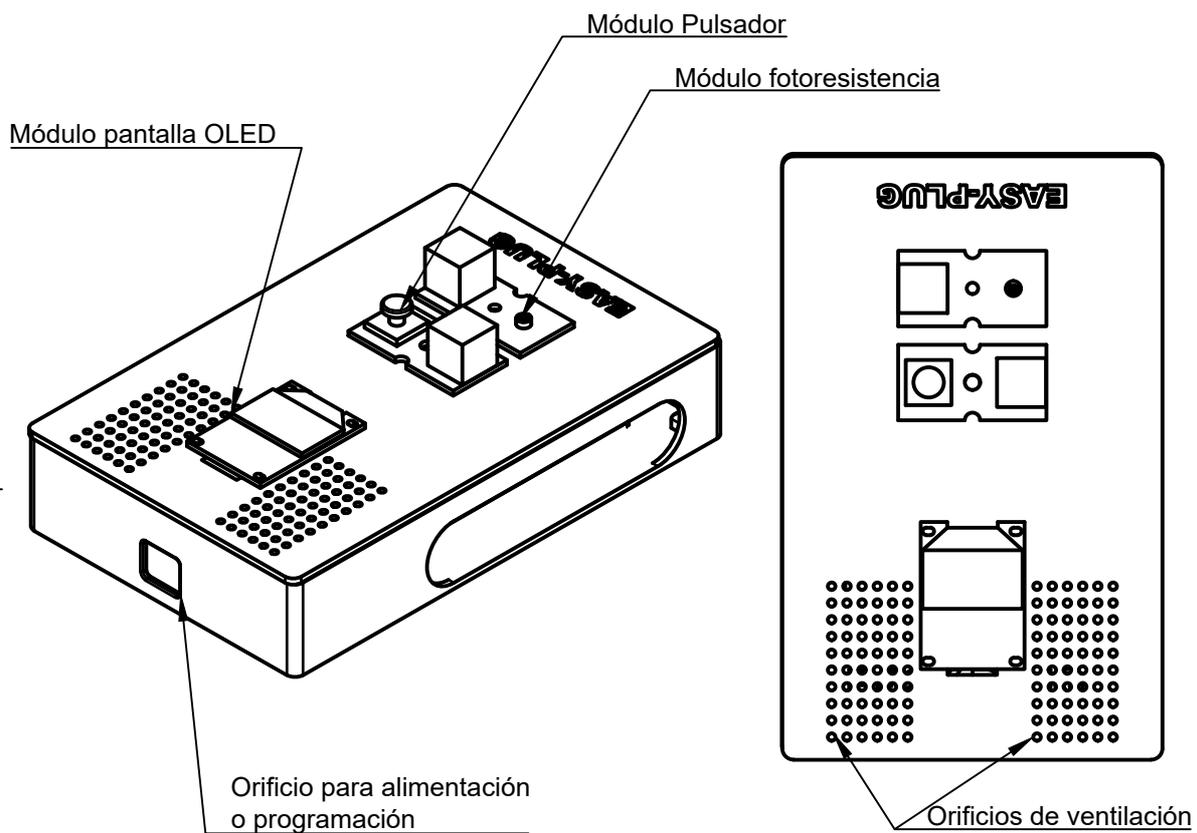
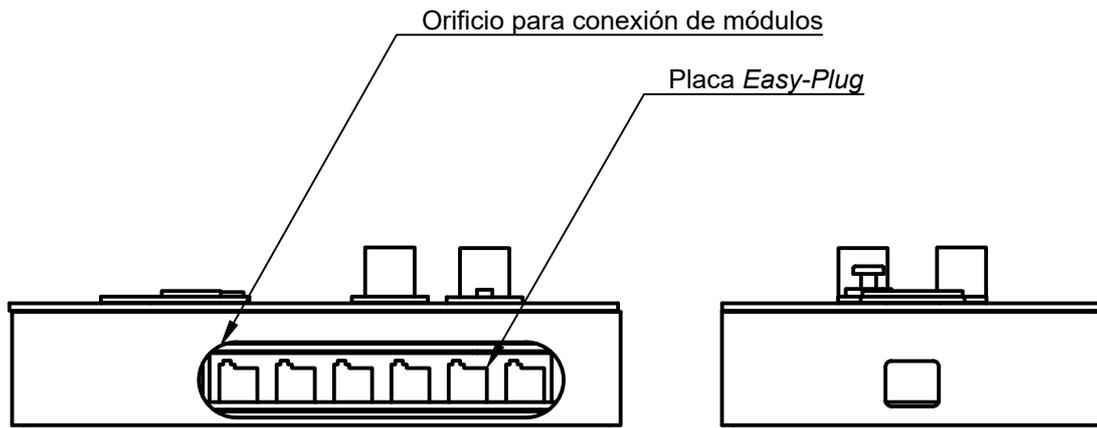


A (1:1)



|   |                     |  |                  |              |
|---|---------------------|--|------------------|--------------|
| Scale<br>1:2  | Technical reference | Created by<br>Alberto Delgado 16/06/2019 | Approved by      |              |
|  UNIVERSITAT POLITÈCNICA DE VALÈNCIA<br><br><br>Escuela Técnica Superior de Ingeniería del Diseño |                     | Document type                            | Document status  |              |
|   |                     | Title<br>Tapa Caja Placa Easy-Plug       | DWG No.<br>ES002 |              |
|   |                     | Rev.                                     | Date of issue    | Sheet<br>2/3 |





|  |                     |   |                         |                     |
|--|---------------------|---|-------------------------|---------------------|
| Scale<br><b>1:2</b>  | Technical reference | Created by<br><b>Alberto Delgado 16/06/2019</b> | Approved by             |                     |
|  <b>UNIVERSITAT POLITÈCNICA DE VALÈNCIA</b><br><br><br>Escuela Técnica Superior de Ingeniería del Diseño |                     | Document type                                   | Document status         |                     |
|  |                     | Title<br><b>TFG</b>                             | DWG No.<br><b>ES003</b> |                     |
|  |                     | Rev.  | Date of issue           | Sheet<br><b>3/3</b> |





UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSIDAD POLITÉCNICA DE VALENCIA  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE DISEÑO  
GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

---

**Diseño y control de un Invernadero  
automatizado de bajo coste adaptado para  
su desarrollo en educación primaria.**

**Design and control of a low-cost  
automated Greenhouse adapted for its  
development in primary education.**

---

### **3. PRESUPUESTO**

**AUTOR:** ALBERTO DELGADO ROMERO

**TUTOR:** LEOPOLDO ARMESTO ÁNGEL

VALENCIA, JUNIO 2019



## Índice

|  |    |
|--|----|
| <b>1. Presupuesto Parcial</b> .....                                  | 97 |
| 1.1 Presupuesto de componentes electrónicos <i>Easy-Plug</i> . ..... | 97 |
| 1.2 Presupuesto actuadores y material de alimentación.....           | 97 |
| 1.3 Presupuesto componentes del invernadero. ....                    | 97 |
| 1.4 Presupuesto herramientas y maquinaria. ....                      | 97 |
| 1.5 Presupuesto mano de obra. ....                                   | 98 |
| <b>2. Presupuesto Total</b> .....                                    | 98 |



Para la comprensión del presupuesto se ha dividido en los diversos grupos de materiales necesarios durante el proceso de desarrollo del proyecto:

## 1. Presupuesto Parcial

### 1.1 Presupuesto de componentes electrónicos *Easy-Plug*.

| Unidad       | Descripción   | Cantidad | Precio Unitario | Importe        |
|--------------|---|----------|-----------------|----------------|
| Ud.          | Easy plug V2.0 Control Board                        | 1        | 12,00 €         | 12,00 €        |
| Ud.          | Easy plug Digital Push Button Module                | 1        | 2,80 €          | 2,80 €         |
| Ud.          | Easy plug 128x64 OLED Module                        | 1        | 7,20 €          | 7,20 €         |
| Ud.          | Easy plug Soil humidity Module                      | 1        | 2,80 €          | 2,80 €         |
| Ud.          | Easy plug DHT11 Temp-Humidity Sensor Module         | 1        | 4,79 €          | 4,79 €         |
| Ud.          | Easy plug Single relay one channel Module           | 2        | 2,80 €          | 5,60 €         |
| Ud.          | Easy plug Photoresistor Sensor Module               | 1        | 2,80 €          | 2,80 €         |
| Ud.          | Easy plug Connection Cable Telephone Cord x5 (RJ11) | 2        | 3,04 €          | 6,08 €         |
| <b>TOTAL</b> | <b>Componentes electrónicos Easy-Plug</b>           |          |                 | <b>44,07 €</b> |

### 1.2 Presupuesto actuadores y material de alimentación.

| Unidad       | Descripción                                | Cantidad | Precio Unitario | Importe        |
|--------------|--|----------|-----------------|----------------|
| Ud.          | Ventilador 5V, 0,2A                        | 1        | 4,99 €          | 4,99 €         |
| m.           | Tira de led                                | 0,5      | 2,24 €          | 1,12 €         |
| Ud.          | Mini bomba de agua sumergible 3-4,5V       | 1        | 5,99 €          | 5,99 €         |
| Ud.          | Batería Externa de alimentación            | 1        | 5,00 €          | 5,00 €         |
| Ud.          | Batería de 9 V para ventilación            | 1        | 2,90 €          | 2,90 €         |
| Ud.          | Pack 4 Batería de 1,5 V para bomba de agua | 1        | 1,37 €          | 1,37 €         |
| <b>TOTAL</b> | <b>Actuadores externos y alimentación</b>  |          |                 | <b>21,37 €</b> |

### 1.3 Presupuesto componentes del invernadero.

| Unidad       | Descripción                             | Cantidad | Precio Unitario | Importe       |
|--------------|---|----------|-----------------|---------------|
| L            | Tierra Semillero Natura                 | 6        | 0,28 €          | 1,65 €        |
| Ud.          | Pack Semillas Albahaca, Berro y Perejil | 1        | 2,99 €          | 2,99 €        |
| Ud.          | Cubierta superior e inferior            | 1        | -               | -             |
| Ud.          | Depósito de agua para la bomba          | 1        | -               | -             |
| <b>TOTAL</b> | <b>Componentes Invernadero</b>          |          |                 | <b>4,64 €</b> |

Los campos del precio unitario de la cubierta superior y de la cubierta inferior están vacíos ya que en este proyecto, estos dos componentes proceden del reciclaje de envases de alimentos.

### 1.4 Presupuesto herramientas y maquinaria.

| Unidad       | Descripción                                     | Cantidad | Precio Unitario | Importe        |
|--------------|---|----------|-----------------|----------------|
| Ud.          | Juego de 6 destornilladores de precisión Alyco  | 1        | 16,59 €         | 16,59 €        |
| Ud.          | Soldador de estaño 60 W Tª ajustable 200-450 °C | 1        | 9,99 €          | 9,99 €         |
| h            | Impresora 3D ANET A8                            | 10       | 0,014 €         | 0,14 €         |
| Kg           | Filamento PLA                                   | 0,05939  | 17,90 €         | 1,06 €         |
| <b>TOTAL</b> | <b>Herramientas y maquinaria</b>                |          |                 | <b>27,77 €</b> |

### 1.5 Presupuesto mano de obra.

| Unidad       | Descripción                   | Cantidad | Precio Unitario | Importe           |
|--------------|-------------------------------|----------|-----------------|-------------------|
| h            | Estudio de proyecto           | 25       | 25,00 €         | 625,00 €          |
| h            | Diseño y montaje de hardware  | 10       | 25,00 €         | 250,00 €          |
| h            | Diseño de Software            | 40       | 25,00 €         | 1.000,00 €        |
| h            | Pruebas, ajustes y depuración | 40       | 25,00 €         | 1.000,00 €        |
| h            | Documentación                 | 100      | 25,00 €         | 2.500,00 €        |
| <b>TOTAL</b> | <b>Mano de obra</b>           |          |                 | <b>5.375,00 €</b> |

### 2. Presupuesto Total

| Ud. Obra     | Presupuestos parciales             | Importe           |
|--------------|------------------------------------|-------------------|
| 1            | Componentes Electrónicos Easy-Plug | 44,07 €           |
| 2            | Actuadores externos y alimentación | 21,37 €           |
| 3            | Componentes Invernadero            | 4,64 €            |
| 4            | Herramientas y maquinaria          | 27,77 €           |
| 5            | Mano de obra                       | 5.375,00 €        |
| <b>TOTAL</b> |                                    | <b>5.472,85 €</b> |



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSIDAD POLITÉCNICA DE VALENCIA  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE DISEÑO  
GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

---

**Diseño y control de un Invernadero  
automatizado de bajo coste adaptado para  
su desarrollo en educación primaria.**

**Design and control of a low-cost  
automated Greenhouse adapted for its  
development in primary education.**

---

## **4. ANEXOS**

**AUTOR:** ALBERTO DELGADO ROMERO

**TUTOR:** LEOPOLDO ARMESTO ÁNGEL

VALENCIA, JUNIO 2019

## ANEXO I

Código del programa final del proyecto en lenguaje Arduino como resultado de la combinación de bloques en Facilino Junior.

```
1. #include <DHT.h>
2. #include <Adafruit_SSD1306.h>
3. #include <splash.h>
4. #include <Adafruit_GFX.h>
5. #include <Adafruit_SPITFT.h>
6. #include <Adafruit_SPITFT_Macros.h>
7. #include <gfxfont.h>
8.
9. #include <Wire.h>
10. #include <SPI.h>
11.
12. /**/ Global variables ***/
13.
14. DHT sensorDHT11_9(9,DHT11);
15.
16. /**/ Function declaration ***/
17. #define OLED_RESET 4
18. Adafruit_SSD1306 display(OLED_RESET);
19.
20. void setup()
21. {
22. display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
23. display.clearDisplay();
24. delay(250);
25. pinMode(A0, INPUT);
26.
27. pinMode(A1, INPUT);
28.
29. pinMode(7, OUTPUT);
30.
31. sensorDHT11_9.begin();
32.
33. pinMode(8, OUTPUT);
34.
35. pinMode(6, OUTPUT);
36. }
37. void loop()
38. {
39. display.setTextSize(1);
40. display.setTextColor(WHITE);
41. display.setCursor(1,1);
42. display.print("Temperatura: ");
43. display.println(sensorDHT11_9.readTemperature());
44.
45. display.setTextSize(1);
46. display.setTextColor(WHITE);
47. display.setCursor(1,10);
48. display.print("Humedad: ");
49. display.println(map(analogRead(A0), 0, 1023, 0, 100));
50.
51. display.setTextSize(1);
52. display.setTextColor(WHITE);
53. display.setCursor(1,20);
54. display.print("Luz: ");
55. display.println(map(analogRead(A1), 0, 1023, 0, 100));
56. if(sensorDHT11_9.readTemperature() >28.5){
57. while(sensorDHT11_9.readTemperature() >27){
58. digitalWrite(7,HIGH);
59.
60. display.display();
61. display.clearDisplay();
62. delay(250);
63.
64. display.setTextSize(3);
65. display.setTextColor(WHITE);
66. display.setCursor(1,10);
67. display.print("FAN");
68. display.display();
69. display.clearDisplay();
70. delay(250);
71. }
72. }else{
73. digitalWrite(7,LOW);
74. }
75. if(analogRead(A0)<102.3){
76. while(analogRead(A0)<204.6){
77. digitalWrite(8,HIGH);
78.
79. display.display();
80. display.clearDisplay();
81. delay(250);
82.
83. display.setTextSize(3);
84. display.setTextColor(WHITE);
85. display.setCursor(1,10);
86. display.print("BOMBA");
87. display.display();
```

```
88.     display.clearDisplay();
89.     delay(250);
90.   }
91.   }else{
92.     digitalWrite(8,LOW);
93.   }
94.   if(analogRead(A1)<204.6){
95.     digitalWrite(6,HIGH);
96.
97.     display.display();
98.     display.clearDisplay();
99.     delay(250);
100.
101.     display.setTextSize(3);
102.     display.setTextColor(WHITE);
103.     display.setCursor(1,10);
104.     display.print("LED");
105.     display.display();
106.     display.clearDisplay();
107.     delay(250);
108.   }
109.   else{
110.     digitalWrite(6,LOW);
111.   }
112.
113.   display.display();
114.   display.clearDisplay();
115.   delay(250);
116. }
```

## ANEXO II

Guion para la realización del proyecto en el último ciclo de Educación Primaria.

# INVERNADERO AUTOMÁTICO EASY-PLUG

## GUIÓN DEL PROYECTO



ALUMNO:  
CURSO:  
CLASE:  
FECHA:



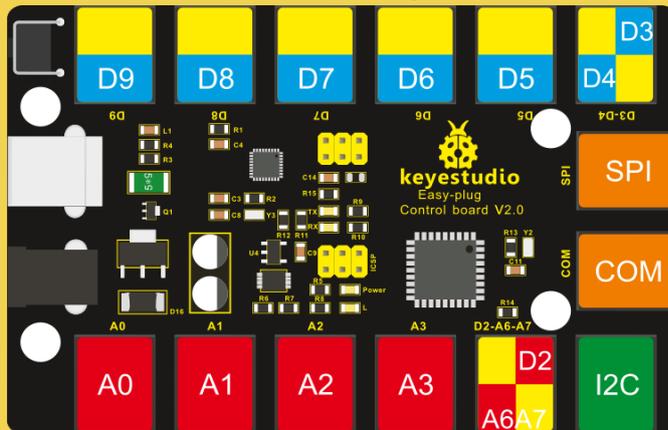
# MATERIALES UTILIZADOS



## MÓDULOS EASY-PLUG



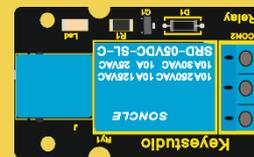
Placa Principal



Módulo humedad tierra



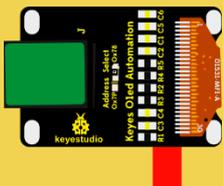
Módulo Relé



Módulo Led



Módulo OLED



Módulo Botón



Módulo Luminosidad



Módulo Temperatura



Cable de conexión Easy-Plug



Bomba de agua



Batería Alimentación



Ventilador 5V





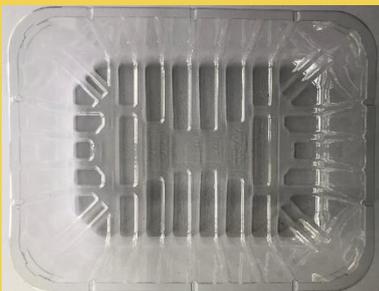
# MATERIALES UTILIZADOS



## MATERIALES PARA INVERNADERO



Envases para base y cubierta



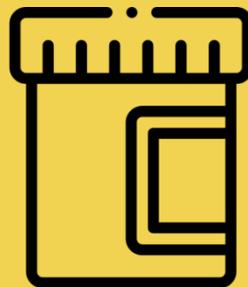
Conducto de riego



Semillas



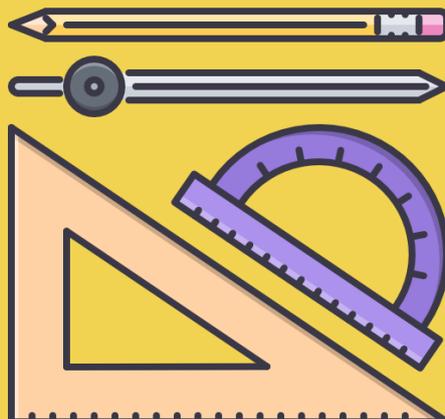
Depósito de agua



Tierra de semillero



## HERRAMIENTAS NECESARIAS

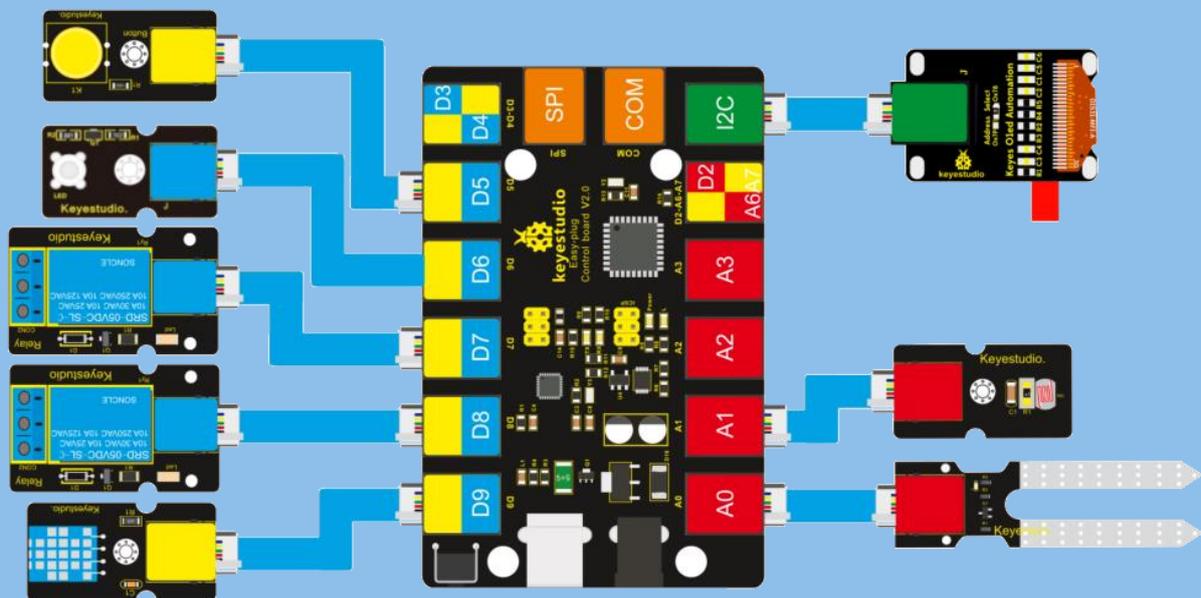




# CONEXIÓN DE LA PLACA



Para la conexión de los módulos a la placa, debemos fijarnos en las etiquetas de colores. Cada módulo debe conectarse en un puerto de la placa principal que sea del mismo color.



## CONEXIONES DE MÓDULOS:

|                             |   |    |                   |   |     |
|-----------------------------|---|----|-------------------|---|-----|
| BOTÓN                       | → | D5 | PANTALLA OLED     | → | I2C |
| LED                         | → | D6 | SENSOR HUM. SUELO | → | A0  |
| RELÉ 1                      | → | D7 | FOTORRESISTENCIA  | → | A1  |
| RELÉ 2                      | → | D8 |                   |   |     |
| SENSOR T <sup>a</sup> DHT11 | → | D9 |                   |   |     |

El color Verde pertenece a los módulos de comunicación, la pantalla OLED.

El color Rojo pertenece a los módulos de entradas analógicas (0-1023).

El color Azul pertenece a los módulos de salidas digitales (0-1).

El color Amarillo pertenece a los módulos de entradas digitales (0-1).



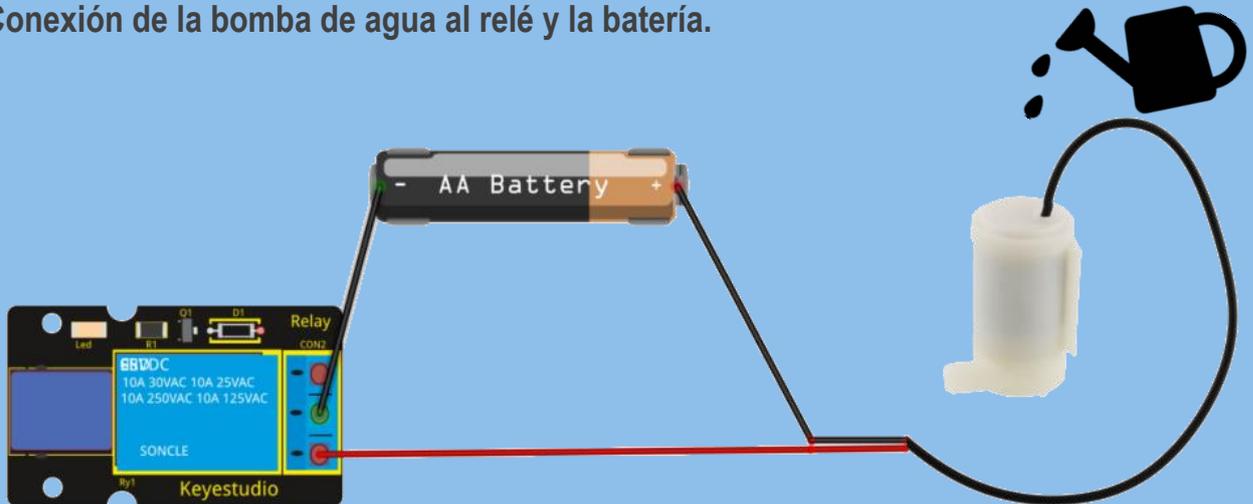
## CONEXIÓN DE LA PLACA



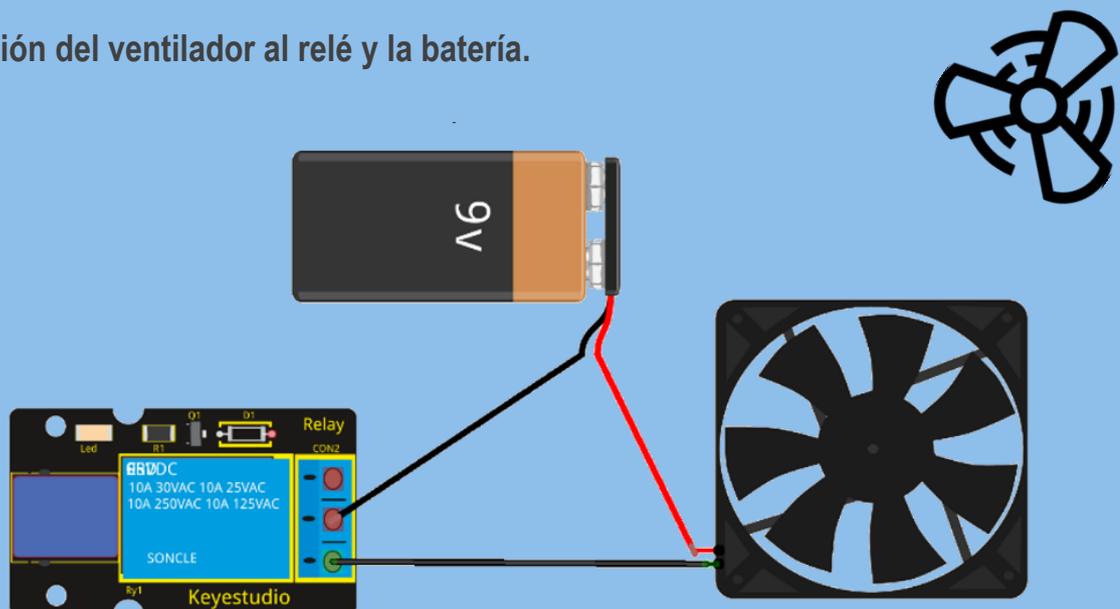
A continuación, se conectan los actuadores a los relés para su activación. En ambas conexiones, se utilizan los puertos C y NO de los relés (Común y Normalmente abierto).

Cuando la placa principal envía información para encender los actuadores al relé, se activa la bobina alimentando la bomba de agua y el ventilador

Conexión de la bomba de agua al relé y la batería.



Conexión del ventilador al relé y la batería.





# MONTAJE INVERNADERO



## PLANTACIÓN DE SEMILLAS

1. LLENADO DE LA BASE



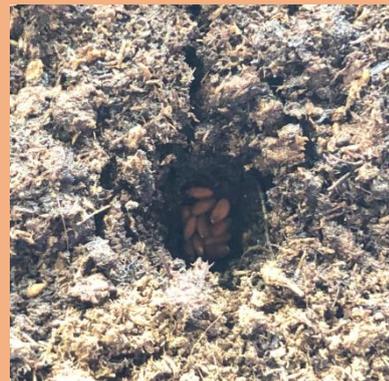
2. ORIFICIOS PARA LAS SEMILLAS



4. TAPADO DE ORIFICIOS Y REGADO



3. INTRODUCCIÓN DE SEMILLAS



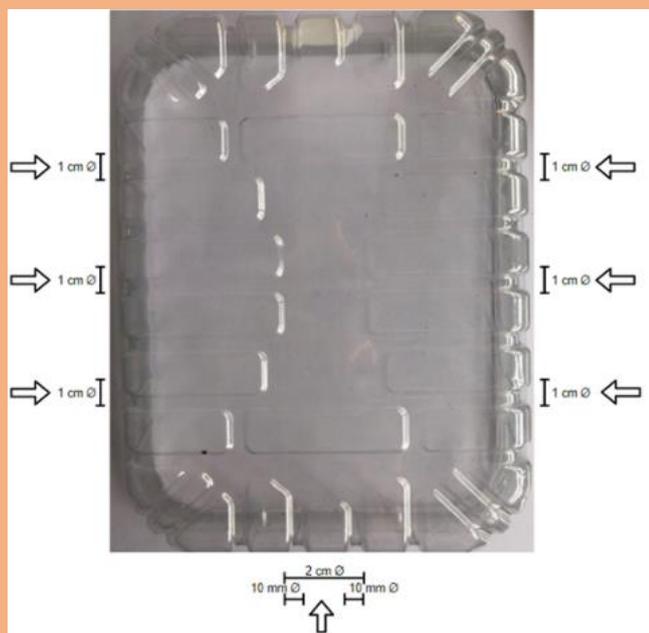


# MONTAJE INVERNADERO



## CONSTRUCCIÓN DE CUBIERTA Y SISTEMA DE RIEGO

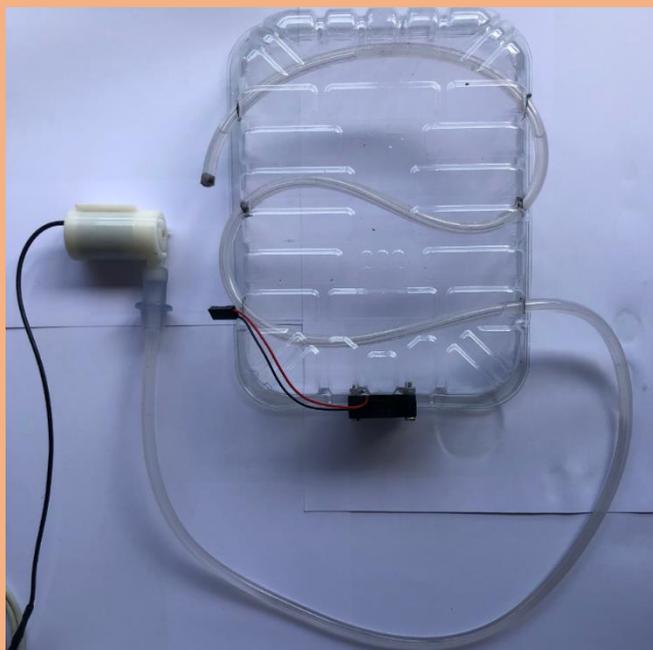
### 1. AGUJEREO DE CUBIERTA



### 2. INTRODUCCIÓN CONDUCTO DE RIEGO Y VENTILADOR



### 3. CONEXIÓN A BOMBA DE AGUA



#### CONSEJOS:

- Utiliza unas tijeras puntiagudas para realizar los agujeros en la cubierta con ayuda del profesor.
- Para evitar que la bomba se rompa se debe introducir agua en el conducto antes de conectarla.



# PROGRAMACIÓN FACILINO JUNIOR



## PROGRAMACIÓN PANTALLA OLED

Utilizaremos la pantalla OLED para mostrar en ella los valores en tiempo real de:

- Humedad de la tierra.
- Temperatura ambiente.
- Luminosidad.



Este es el bloque que usaremos, pertenece a la categoría “PANTALLAS”. En el campo de texto escribiremos las palabras:

Humedad:

Luz:

Temperatura:

En el campo numérico, introduciremos los bloques de la categoría “AMBIENTE”. Debemos configurar la posición y el tamaño de cada bloque. Todos escribirán en el tamaño pequeño, y cada uno de ellos en una posición.



Así se mostrará la información en la pantalla OLED.





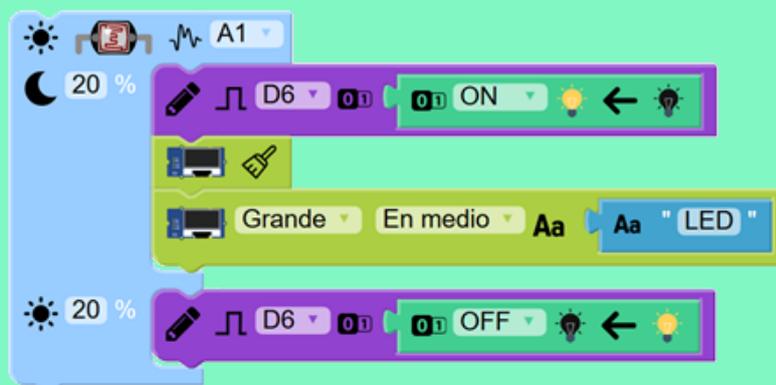
# PROGRAMACIÓN FACILINO JUNIOR



## BLOQUE ALARMA LUMINOSIDAD

Configuramos el puerto de conexión del sensor para el “A1” e introducimos los valores límites de luminosidad, en este caso al 20%. Configuramos las acciones:

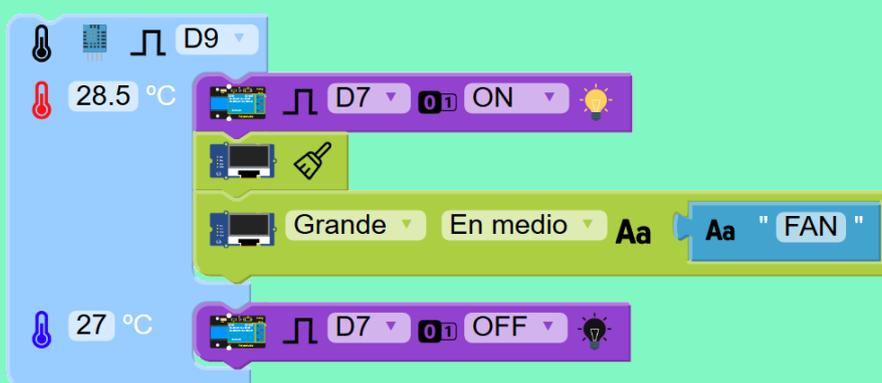
- Si la luminosidad es menor del 20%, queremos que el led proporcione luz al invernadero. Por ello introducimos el bloque del relé (puerto D6) encendido y el bloque de escritura en la pantalla OLED para que muestre la palabra “LED”.
- Si la luminosidad es mayor del 20%, queremos que las luces led se apaguen. Para ello introducimos el bloque del relé (puerto D6) apagado.



## BLOQUE ALARMA TEMPERATURA

Configuramos el puerto de conexión del sensor para el “D9” e introducimos los valores límites de TEMPERATURA, en este caso 28,5 °C. Configuramos las acciones:

- Si la temperatura es mayor de 28,5 °C, queremos que el ventilador funcione, rebajando la temperatura del interior del invernadero. Por ello introducimos el bloque del relé (puerto D7) encendido y el bloque de escritura en la pantalla OLED para que muestre la palabra “FAN”.
- Si la temperatura es menor de 27 °C, no necesitamos el ventilador en funcionamiento. Para ello introducimos el bloque del relé (puerto D8) apagado.





# PROGRAMACIÓN FACILINO JUNIOR



## PROGRAMACIÓN BLOQUES ALARMA

### BLOQUE ALARMA HUMEDAD DE TIERRA

PIN A0  
HUMEDAD MIN → 10%  
HUMEDAD LIM → 20%

### BLOQUE ALARMA LUMINOSIDAD

PIN A1  
LUZ MIN → 20%  
LUZ LIM → 20%

### BLOQUE ALARMA TEMPERATURA

PIN D9  
TEMPERATURA MIN → 28,5 °C  
TEMPERATURA LIM → 27 °C



# PROGRAMACIÓN FACILINO JUNIOR



## CONJUNTO DE BLOQUES FINAL

The final block set consists of the following components:

- Temperature Control:**
  - Sensor: Pequeño Arriba Aa Temperatura: D9
  - Display: 28.5 °C
  - Control: Grande En medio Aa " FAN "
- Humidity Control:**
  - Sensor: Pequeño En medio Aa Humedad: A0
  - Display: 10 %
  - Control: Grande En medio Aa " BOMBA "
- Light Control:**
  - Sensor: Pequeño Abajo Aa Luz: A1
  - Display: 20 %
  - Control: Grande En medio Aa " LED "

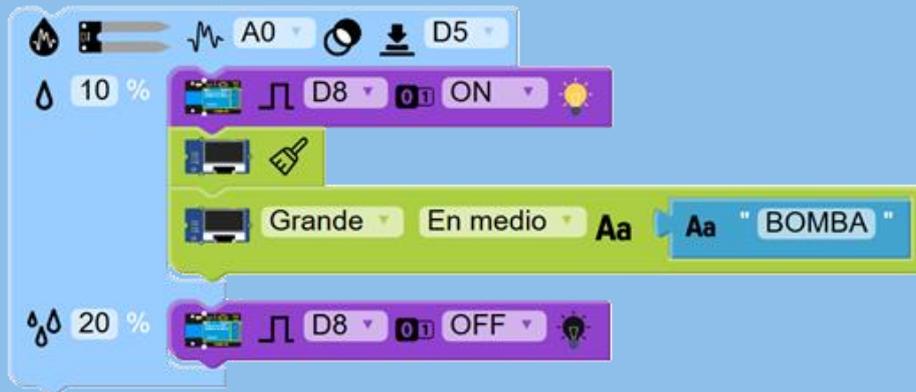


## UN PASO MÁS



### BLOQUE ALARMA HUMEDAD DE TIERRA MÁS PULSADOR

Una solución opcional de ampliación del proyecto es la sustitución del bloque de alarma de humedad por el bloque de alarma de humedad más pulsador:

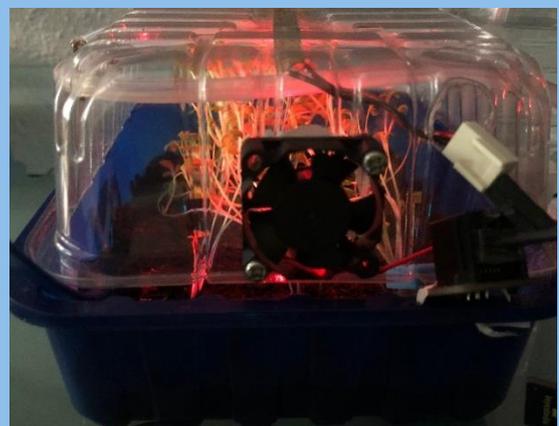
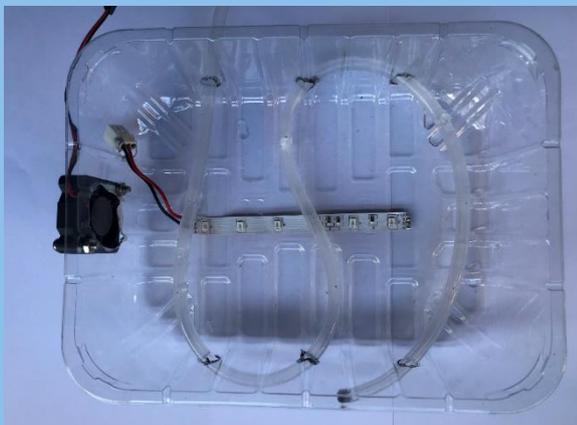


Este bloque nos va a permitir controlar la bomba de forma manual, manejando el riego del invernadero sin necesidad de que la humedad sea menor a la mínima programada.

Para la sustitución de este bloque, solo tenemos que seleccionar el puerto "D5" en el que va a estar conectado el pulsador a la placa, y configurar el resto de bloque de la misma forma que en el caso anterior:

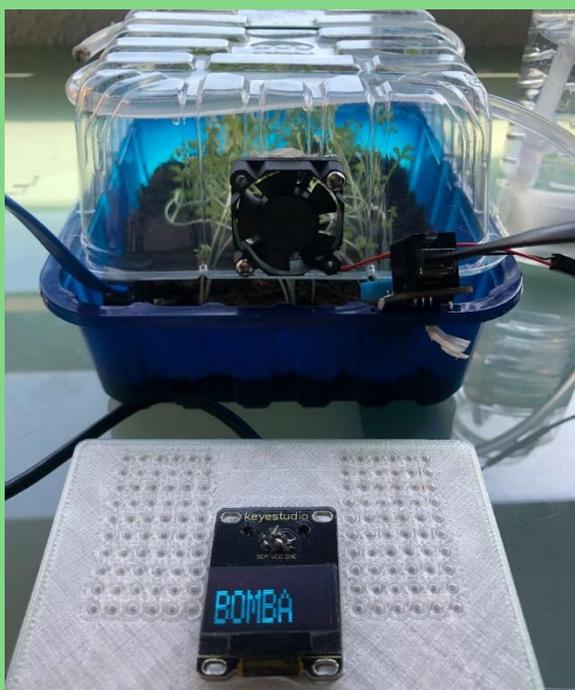
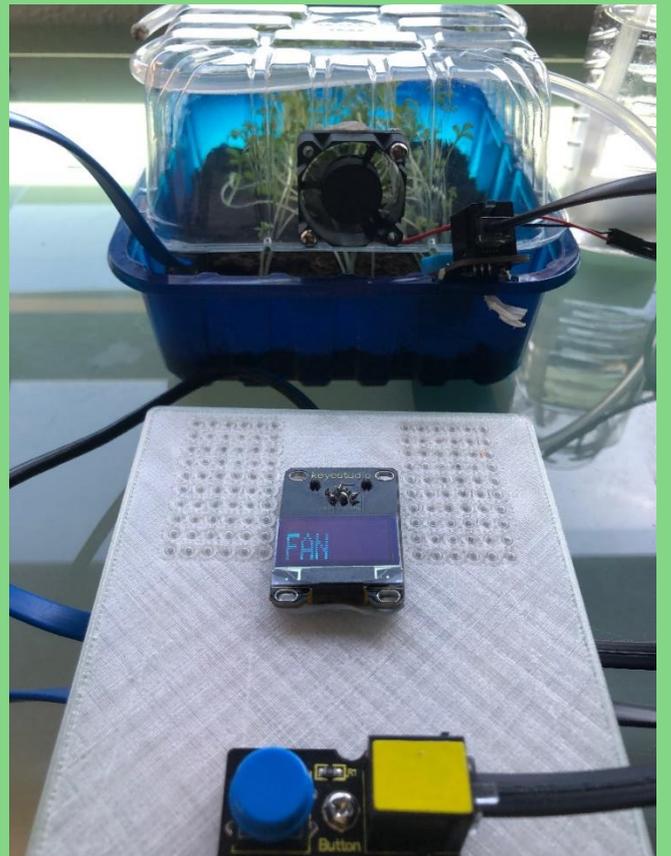
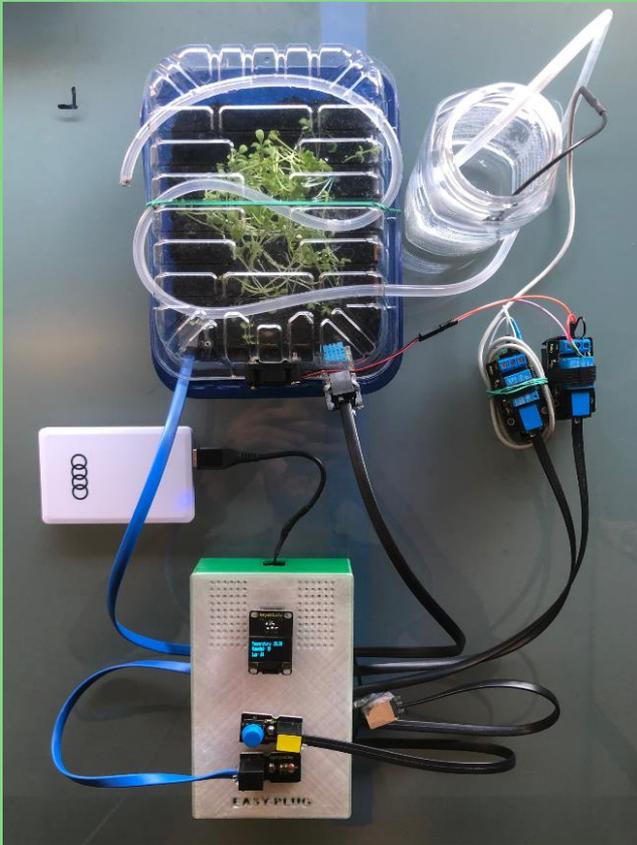
### TIRA DE LEDS

Una mejora al proyecto sería cambiar el LED por una tira de Leds. La programación en Arduino no variaría. Simplemente hay que instalar la tira de leds en la cubierta del invernadero:





# RESULTADO FINAL



Creado por: Alberto Delgado Romero



## ANEXO III

Código en JavaScript de bloques del proyecto incorporados en la aplicación Facilino Junior.

### Bloque estados HIGH-LOW, categoría “Lógica”

```
1.   Blockly.Arduino.logic_state = function() {
2.     var code = (this.getFieldValue('STATE') === 'HIGH') ? 'HIGH' : 'LOW';
3.     return [code, Blockly.Arduino.ORDER_ATOMIC];
4.   };
5.
6.   Blockly.Blocks.logic_state = {
7.     category: Facilino.locales.getKey('LANG_CATEGORY_LOGIC'),
8.     category_colour: Facilino.LANG_COLOUR_LOGIC,
9.     colour: Facilino.LANG_COLOUR_LOGIC,
10.    keys: ['LANG_LOGIC_STATE_HIGH', 'LANG_LOGIC_STATE_LOW', 'LANG_LOGIC_STATE_TOOLTIP'],
11.    output: 'boolean',
12.    init: function() {
13.      this.setColour(Facilino.LANG_COLOUR_LOGIC);
14.      this.setOutput(true, Boolean);
15.      this.appendDummyInput('ENT').appendField(new Blockly.FieldImage('img/blocks/binary.svg', 20*options.zoom, 20*options.zoom))
16.        .appendField(new Blockly.FieldDropdown([
17.          [Facilino.locales.getKey('LANG_LOGIC_STATE_HIGH'), 'HIGH'],
18.          [Facilino.locales.getKey('LANG_LOGIC_STATE_LOW'), 'LOW']
19.        ]), 'STATE')
20.        .appendField(new Blockly.FieldImage('img/blocks/arrow_up.svg', 20*options.zoom, 20*options.zoom), 'HIGH')
21.      this.setTooltip(Facilino.locales.getKey('LANG_LOGIC_STATE_TOOLTIP'));
22.      this.state=this.getFieldValue('STATE');
23.    },
24.    onchange: function()
25.    {
26.      if (this.state!==this.getFieldValue('STATE'))
27.      {
28.        //console.Log('Cambio' + this.getFieldValue('STATE'));
29.        this.state=this.getFieldValue('STATE');
30.        ent=this.getInput('ENT');
31.        ent.removeField('HIGH');
32.        ent.removeField('LOW');
33.        if (this.state==='LOW')
34.        {
35.          ent.appendField(new Blockly.FieldImage('img/blocks/arrow_down.svg', 20*options.zoom, 20*options.zoom), 'LOW')
36.          this.setTooltip(Facilino.locales.getKey('LANG_LOGIC_STATE_TOOLTIP'));
37.        }
38.        else
39.        {
40.          ent.appendField(new Blockly.FieldImage('img/blocks/arrow_up.svg', 20*options.zoom, 20*options.zoom), 'HIGH')
41.          this.setTooltip(Facilino.locales.getKey('LANG_LOGIC_STATE_TOOLTIP'));
42.        }
43.      }
44.    }
45.  };
46.
```

### Bloque estados ON-OFF, categoría “Lógica”

```
1.   Blockly.Arduino.logic_state_2 = function() {
2.     var code = (this.getFieldValue('STATE') === 'HIGH') ? 'HIGH' : 'LOW';
3.     return [code, Blockly.Arduino.ORDER_ATOMIC];
4.   };
5.
6.   Blockly.Blocks.logic_state_2 = {
7.     category: Facilino.locales.getKey('LANG_CATEGORY_LOGIC'),
8.     category_colour: Facilino.LANG_COLOUR_LOGIC,
9.     colour: Facilino.LANG_COLOUR_LOGIC,
10.    keys: ['LANG_LOGIC_STATE_HIGH_2', 'LANG_LOGIC_STATE_LOW_2', 'LANG_LOGIC_STATE_TOOLTIP_2'],
11.    output: 'boolean',
12.    init: function() {
13.      this.setColour(Facilino.LANG_COLOUR_LOGIC);
14.      this.setOutput(true, Boolean);
15.      this.appendDummyInput('INPUT').appendField(new Blockly.FieldImage('img/blocks/binary.svg', 20*options.zoom, 20*options.zoom))
16.        .appendField(new Blockly.FieldDropdown([
17.          [Facilino.locales.getKey('LANG_LOGIC_STATE_HIGH_2'), 'HIGH'],
18.          [Facilino.locales.getKey('LANG_LOGIC_STATE_LOW_2'), 'LOW']
19.        ]), 'STATE')
20.        .appendField(new Blockly.FieldImage('img/blocks/ON.svg', 20*options.zoom, 20*options.zoom), 'LEFT')
21.        .appendField(new Blockly.FieldImage('img/blocks/ARROW.svg', 20*options.zoom, 20*options.zoom), 'ARROW')
22.        .appendField(new Blockly.FieldImage('img/blocks/OFF.svg', 20*options.zoom, 20*options.zoom), 'RIGHT');
23.      this.setTooltip(Facilino.locales.getKey('LANG_LOGIC_STATE_TOOLTIP_2'));
24.      this.state=this.getFieldValue('STATE');
25.    },
26.    onchange: function()
27.    {
28.      if (this.state!==this.getFieldValue('STATE'))
29.      {
30.        //console.Log('Cambio' + this.getFieldValue('STATE'));
31.        this.state=this.getFieldValue('STATE');
32.        var inp=this.getInput('INPUT');
33.        inp.removeField('LEFT');
34.        inp.removeField('ARROW');
35.        inp.removeField('RIGHT');
36.
```

```

37.         if (this.state==='LOW')
38.         {
39.             inp.appendField(new Blockly.FieldImage('img/blocks/OFF.svg',20*options.zoom,20*options.zoom),'LEFT')
40.             .appendField(new Blockly.FieldImage('img/blocks/ARROW.svg',20*options.zoom,20*options.zoom),'ARROW')
41.             .appendField(new Blockly.FieldImage('img/blocks/ON.svg',20*options.zoom,20*options.zoom),'RIGHT');
42.             this.setTooltip(Facilino.locales.getKey('LANG_LOGIC_STATE_TOOLTIP_2'));
43.         }
44.         else
45.         {
46.             inp.appendField(new Blockly.FieldImage('img/blocks/ON.svg',20*options.zoom,20*options.zoom),'LEFT')
47.             .appendField(new Blockly.FieldImage('img/blocks/ARROW.svg',20*options.zoom,20*options.zoom),'ARROW')
48.             .appendField(new Blockly.FieldImage('img/blocks/OFF.svg',20*options.zoom,20*options.zoom),'RIGHT');
49.             this.setTooltip(Facilino.locales.getKey('LANG_LOGIC_STATE_TOOLTIP_2'));
50.         }
51.     }
52. }
53. };

```

### Bloque entrada/salida relé básico, categoría “Entrada/salida básica”

```

1. Blockly.Arduino.relay_write = function() {
2.     var dropdown_pin = this.getFieldValue('PIN');
3.     var dropdown_stat = Blockly.Arduino.valueToCode(this, 'STAT', Blockly.Arduino.ORDER_ATOMIC);
4.     var code = '';
5.     Blockly.Arduino.setups['setup_digital_write_' + dropdown_pin] = JST['inout_digital_write_setups']({'dropdown_pin':dropdown_pin}
6. );
7.     code += JST['inout_digital_write']({'dropdown_pin': dropdown_pin,'dropdown_stat': dropdown_stat});
8.     return code;
9. };
10.
11. Blockly.Blocks.relay_write = {
12.     category: Facilino.locales.getKey('LANG_CATEGORY_ADVANCED'),
13.     category_colour: Facilino.LANG_COLOUR_ADVANCED,
14.     colour: Facilino.LANG_COLOUR_ADVANCED,
15.     keys: ['LANG_RELAY_WRITE_TOOLTIP'],
16.     init: function() {
17.         this.setColour(Facilino.LANG_COLOUR_ADVANCED);
18.         this.appendDummyInput('').appendField(new Blockly.FieldImage('img/blocks/relay.svg',20*options.zoom, 20*options.zoom))
19.         .appendField(new Blockly.FieldImage("img/blocks/digital_signal.svg",20*options.zoom, 20*options.zoom))
20.         .appendField(new Blockly.FieldDropdown(profiles.default.digital), 'PIN');
21.         this.appendValueInput('STAT').appendField(new Blockly.FieldImage('img/blocks/binary.svg',20*options.zoom,20*options.zoom))
22.         .setAlign(Blockly.ALIGN_RIGHT);
23.         this.setPreviousStatement(true,'code');
24.         this.setInputsInline(true);
25.         this.setNextStatement(true,'code');
26.         this.setTooltip(Facilino.locales.getKey('LANG_RELAY_WRITE_TOOLTIP'));
27.     }
28. };

```

### Bloque entrada/salida relé ON-OFF, categoría “Entrada/salida básica”

```

1. Blockly.Arduino.relay_write_2 = function() {
2.     var dropdown_pin = this.getFieldValue('PIN');
3.     var dropdown_stat = (this.getFieldValue('STATE') === 'HIGH') ? 'HIGH' : 'LOW';
4.     var code = '';
5.     Blockly.Arduino.setups['setup_digital_write_' + dropdown_pin] = JST['inout_digital_write_setups']({'dropdown_pin':dropdown_pin}
6. );
7.     code += JST['inout_digital_write']({'dropdown_pin': dropdown_pin,'dropdown_stat':dropdown_stat });
8.     return code;
9. };
10.
11. Blockly.Blocks.relay_write_2 = {
12.     category: Facilino.locales.getKey('LANG_CATEGORY_ADVANCED'),
13.     category_colour: Facilino.LANG_COLOUR_ADVANCED,
14.     colour: Facilino.LANG_COLOUR_ADVANCED,
15.     keys: ['LANG_RELAY_WRITE_TOOLTIP', 'LANG_LOGIC_STATE_HIGH_2', 'LANG_LOGIC_STATE_LOW_2', 'LANG_LOGIC_STATE_TOOLTIP_2' ],
16.     init: function() {
17.         this.setColour(Facilino.LANG_COLOUR_ADVANCED);
18.         this.appendDummyInput('').appendField(new Blockly.FieldImage('img/blocks/relay.svg',30*options.zoom,20*options.zoom))
19.         .appendField(new Blockly.FieldImage("img/blocks/digital_signal.svg",20*options.zoom,20*options.zoom)).setAlign(Blockly.A
20. LIGN_RIGHT)
21.         .appendField(new Blockly.FieldDropdown(profiles.default.digital), 'PIN');
22.         this.appendDummyInput('').appendField(new Blockly.FieldImage('img/blocks/binary.svg',20*options.zoom,20*options.zoom))
23.         this.appendDummyInput('ENT').appendField(new Blockly.FieldDropdown([
24.             [Facilino.locales.getKey('LANG_LOGIC_STATE_HIGH_2'), 'HIGH'],
25.             [Facilino.locales.getKey('LANG_LOGIC_STATE_LOW_2'), 'LOW']
26.         ]), 'STATE')
27.         //this.appendDummyInput('').appendField(new Blockly.FieldImage('img/blocks/ON.svg',20*options.zoom,20*options.zoom), 'HIGH')
28.
29.         this.setPreviousStatement(true,'code');
30.         this.setInputsInline(true);
31.         this.setNextStatement(true,'code');
32.         this.setTooltip(Facilino.locales.getKey('LANG_RELAY_WRITE_TOOLTIP'));
33.     },
34.     onchange: function()
35.     {
36.         if (this.state!==this.getFieldValue('STATE'))
37.         {
38.             //console.Log('Cambio' + this.getFieldValue('STATE'));
39.             this.state=this.getFieldValue('STATE');
40.             ent=this.getInput('ENT');
41.             ent.removeField('HIGH');

```

```

40.         ent.removeField('LOW');
41.         if (this.state==='LOW')
42.         {
43.             ent.appendField(new Blockly.FieldImage('img/blocks/OFF.svg',20*options.zoom,20*options.zoom),'LOW')
44.             this.setTooltip(Facilino.locales.getKey('LANG_LOGIC_STATE_TOOLTIP'));
45.         }
46.         else
47.         {
48.             ent.appendField(new Blockly.FieldImage('img/blocks/ON.svg',20*options.zoom,20*options.zoom),'HIGH')
49.             this.setTooltip(Facilino.locales.getKey('LANG_LOGIC_STATE_TOOLTIP'));
50.         }
51.     }
52. }
53. }
54. };
55.

```

## Bloque escritura cadena de caracteres y numérico o variable pantalla OLED, categoría “Pantallas”

```

1.  Blockly.Arduino.oled_display = function() {
2.      var content1 = this.getFieldValue('CONTENT1');
3.      var content2 = Blockly.Arduino.valueToCode(this, 'CONTENT2', Blockly.Arduino.ORDER_ATOMIC);
4.      var dropdown_pin_size = this.getFieldValue('SIZE');
5.      var dropdown_pin_altura = this.getFieldValue('ALTURA');
6.      var code = '';
7.      Blockly.Arduino.definitions_['oled_include']=JST['oled_definitions_include']({});
8.      Blockly.Arduino.definitions_['oled_define']=JST['oled_definitions_define']({});
9.      Blockly.Arduino.definitions_['wire']=JST['wire_definitions_include']({});
10.     Blockly.Arduino.definitions_['spi']=JST['spi_definitions_include']({});
11.     Blockly.Arduino.setups_['setup_oled']=JST['oled_setups']({});
12.     code +='\ndisplay.setTextSize(' +dropdown_pin_size+');\ndisplay.setTextColor(WHITE);\ndisplay.setCursor(1,'+dropdown_pin_a
ltura+');\ndisplay.print(""+content1+'");\ndisplay.println(""+content2+'");\n'
13.     return code;
14. };
15.
16.  Blockly.Blocks.oled_display = {
17.      category: Facilino.locales.getKey('LANG_CATEGORY_SCREEN'),
18.      category_colour: Facilino.LANG_COLOUR_SCREEN,
19.      colour: Facilino.LANG_COLOUR_SCREEN,
20.      keys:['LANG_OLED_DISPLAY_TOOLTIP','LANG_OLED_SIZE_SMALL','LANG_OLED_SIZE_MEDIUM','LANG_OLED_SIZE_BIG','LANG_OLED_POSIT
ION_UP','LANG_OLED_POSITION_MIDDLE','LANG_OLED_POSITION_DOWN'],
21.      init: function() {
22.          this.setColour(Facilino.LANG_COLOUR_SCREEN);
23.          var size = [[Facilino.locales.getKey('LANG_OLED_SIZE_SMALL'),'1'],[Facilino.locales.getKey('LANG_OLED_SIZE_MEDIUM'
),'2'],[Facilino.locales.getKey('LANG_OLED_SIZE_BIG'),'3']];
24.          var altura = [[Facilino.locales.getKey('LANG_OLED_POSITION_UP'),'1'],[Facilino.locales.getKey('LANG_OLED_POSITION_
MIDDLE'),'10'],[Facilino.locales.getKey('LANG_OLED_POSITION_DOWN'),'20']];
25.          this.appendDummyInput('').appendField(new Blockly.FieldImage('img/blocks/oled_display.svg',36*options.zoom,20*opti
ons.zoom))
26.              .appendField(new Blockly.FieldDropdown(size),'SIZE')
27.              .appendField(new Blockly.FieldDropdown(altura),'ALTURA').setAlign(Blockly.ALIGN_RIGHT)
28.              .appendField(new Blockly.FieldImage('img/blocks/text.svg',24*options.zoom,24*options.zoom))
29.              .appendField(new Blockly.FieldTextInput("    "),'CONTENT1')
30.              .appendField(new Blockly.FieldImage('img/blocks/numbers.svg',20*options.zoom,20*options.zoom));
31.          this.appendValueInput('CONTENT2',String);
32.          this.appendDummyInput('SIZE');
33.          this.setInputsInline(true);
34.          this.setPreviousStatement(true,'code');
35.          this.setNextStatement(true,'code');
36.          this.setTooltip(Facilino.locales.getKey('LANG_OLED_DISPLAY_TOOLTIP'));
37.      }
38.  };

```

## Bloque escritura cadena de caracteres pantalla OLED, categoría “Pantallas”

```

1.  Blockly.Arduino.oled_1_display = function() {
2.      var content = Blockly.Arduino.valueToCode(this, 'CONTENT1', Blockly.Arduino.ORDER_ATOMIC);
3.      var dropdown_pin_size = this.getFieldValue('SIZE');
4.      var dropdown_pin_altura = this.getFieldValue('ALTURA');
5.      var code = '';
6.      Blockly.Arduino.definitions_['oled_include']=JST['oled_definitions_include']({});
7.      Blockly.Arduino.definitions_['oled_define']=JST['oled_definitions_define']({});
8.      Blockly.Arduino.definitions_['wire']=JST['wire_definitions_include']({});
9.      Blockly.Arduino.definitions_['spi']=JST['spi_definitions_include']({});
10.     Blockly.Arduino.setups_['setup_oled']=JST['oled_setups']({});
11.     code +='\ndisplay.setTextSize(' +dropdown_pin_size+');\ndisplay.setTextColor(WHITE);\ndisplay.setCursor(1,'+dropdown_pin_a
ltura+');\ndisplay.print(""+content+'");\ndisplay.display();\ndisplay.clearDisplay();\ndelay(250);\n'
12.     return code;
13. };
14.
15.  Blockly.Blocks.oled_1_display = {
16.      category: Facilino.locales.getKey('LANG_CATEGORY_SCREEN'),
17.      category_colour: Facilino.LANG_COLOUR_SCREEN,
18.      colour: Facilino.LANG_COLOUR_SCREEN,
19.      keys:['LANG_OLED_DISPLAY_TOOLTIP','LANG_OLED_SIZE_SMALL','LANG_OLED_SIZE_MEDIUM','LANG_OLED_SIZE_BIG','LANG_OLED_POSIT
ION_UP','LANG_OLED_POSITION_MIDDLE','LANG_OLED_POSITION_DOWN'],
20.      init: function() {
21.          this.setColour(Facilino.LANG_COLOUR_SCREEN);
22.          var size = [[Facilino.locales.getKey('LANG_OLED_SIZE_SMALL'),'1'],[Facilino.locales.getKey('LANG_OLED_SIZE_MEDIUM'
),'2'],[Facilino.locales.getKey('LANG_OLED_SIZE_BIG'),'3']];
23.          var altura = [[Facilino.locales.getKey('LANG_OLED_POSITION_UP'),'1'],[Facilino.locales.getKey('LANG_OLED_POSITION_
MIDDLE'),'10'],[Facilino.locales.getKey('LANG_OLED_POSITION_DOWN'),'20']];

```

```

24.         this.appendDummyInput('').appendField(new Blockly.FieldImage('img/blocks/oled_display.svg',36*options.zoom,20*opti
25.         ons.zoom))
26.             .appendField(new Blockly.FieldDropdown(size), 'SIZE')
27.             .appendField(new Blockly.FieldDropdown(altura), 'ALTURA').setAlign(Blockly.ALIGN_RIGHT)
28.             .appendField(new Blockly.FieldImage('img/blocks/text.svg', 24*options.zoom, 24*options.zoom))
29.         this.appendValueInput('CONTENT1', String);
30.         this.appendDummyInput('Size');
31.         this.setInputsInline(true);
32.         this.setPreviousStatement(true, 'code');
33.         this.setNextStatement(true, 'code');
34.         this.setTooltip(Facilino.locales.getKey('LANG_OLED_DISPLAY_TOOLTIP'));
35.     };

```

### Bloque de escritura numérico o variable pantalla OLED, categoría “Pantallas”

```

1.  Blockly.Arduino.oled_2_display = function() {
2.      var content2 = Blockly.Arduino.valueToCode(this, 'CONTENT2', Blockly.Arduino.ORDER_ATOMIC);
3.      var dropdown_pin_size = this.getFieldValue('SIZE');
4.      var dropdown_pin_altura = this.getFieldValue('ALTURA');
5.      var code = '';
6.      Blockly.Arduino.definitions_['oled_include']=JST['oled_definitions_include']({});
7.      Blockly.Arduino.definitions_['oled_define']=JST['oled_definitions_define']({});
8.      Blockly.Arduino.definitions_['wire']=JST['wire_definitions_include']({});
9.      Blockly.Arduino.definitions_['spi']=JST['spi_definitions_include']({});
10.     Blockly.Arduino.setups_['oled_setups']=JST['oled_setups']({});
11.     code += '\ndisplay.setTextSize('+dropdown_pin_size+');\ndisplay.setTextColor(WHITE);\ndisplay.setCursor(1, '+dropdown_pin_a
12.     ltura+');\ndisplay.println('+content2+');'
13.     return code;
14. };
15.
16.  Blockly.Blocks.oled_2_display = {
17.      category: Facilino.locales.getKey('LANG_CATEGORY_SCREEN'),
18.      category_colour: Facilino.LANG_COLOUR_SCREEN,
19.      colour: Facilino.LANG_COLOUR_SCREEN,
20.      keys:['LANG_OLED_DISPLAY_TOOLTIP', 'LANG_OLED_SIZE_SMALL', 'LANG_OLED_SIZE_MEDIUM', 'LANG_OLED_SIZE_BIG', 'LANG_OLED_POSIT
21.      ION_UP', 'LANG_OLED_POSITION_MIDDLE', 'LANG_OLED_POSITION_DOWN'],
22.      init: function() {
23.          this.setColour(Facilino.LANG_COLOUR_SCREEN);
24.          var size = [[Facilino.locales.getKey('LANG_OLED_SIZE_SMALL'), '1'], [Facilino.locales.getKey('LANG_OLED_SIZE_MEDIUM'
25.          ), '2'], [Facilino.locales.getKey('LANG_OLED_SIZE_BIG'), '3']];
26.          var altura = [[Facilino.locales.getKey('LANG_OLED_POSITION_UP'), '1'], [Facilino.locales.getKey('LANG_OLED_POSITION_
27.          MIDDLE'), '10'], [Facilino.locales.getKey('LANG_OLED_POSITION_DOWN'), '20']];
28.          this.appendDummyInput('').appendField(new Blockly.FieldImage('img/blocks/oled_display.svg',36*options.zoom,20*opti
29.          ons.zoom))
30.              .appendField(new Blockly.FieldDropdown(size), 'SIZE')
31.              .appendField(new Blockly.FieldDropdown(altura), 'ALTURA').setAlign(Blockly.ALIGN_RIGHT)
32.              .appendField(new Blockly.FieldImage('img/blocks/numbers.svg',20*options.zoom,20*options.zoom));
33.          this.appendValueInput('CONTENT2', String);
34.          this.appendDummyInput('Size');
35.          this.setInputsInline(true);
36.          this.setPreviousStatement(true, 'code');
37.          this.setNextStatement(true, 'code');
38.          this.setTooltip(Facilino.locales.getKey('LANG_OLED_DISPLAY_TOOLTIP'));
39.      };
40.  };

```

### Bloque de borrado pantalla OLED, categoría “Pantallas”

```

1.  Blockly.Arduino.oled_clear = function() {
2.      var content = Blockly.Arduino.valueToCode(this, 'CONTENT', Blockly.Arduino.ORDER_ATOMIC);
3.      var code = '';
4.      code += '\ndisplay.display();\ndisplay.clearDisplay();\ndelay(250);\n'
5.      return code;
6.  };
7.
8.  Blockly.Blocks.oled_clear = {
9.      category: Facilino.locales.getKey('LANG_CATEGORY_SCREEN'),
10.     category_colour: Facilino.LANG_COLOUR_SCREEN,
11.     colour: Facilino.LANG_COLOUR_SCREEN,
12.     keys: ['LANG_OLED_CLEAR_TOOLTIP'],
13.     init: function() {
14.         this.setColour(Facilino.LANG_COLOUR_SCREEN);
15.         this.appendDummyInput('').appendField(new Blockly.FieldImage('img/blocks/oled_display.svg',36*options.zoom,20*opti
16.         ons.zoom)).appendField(new Blockly.FieldImage('img/blocks/escoba.svg',20*options.zoom,20*options.zoom)).setAlign(Blockly.ALIGN_RIGHT);
17.         this.setInputsInline(true);
18.         this.setPreviousStatement(true, 'code');
19.         this.setNextStatement(true, 'code');
20.         this.setTooltip(Facilino.locales.getKey('LANG_OLED_CLEAR_TOOLTIP'));
21.     };

```

## Bloque de alarma de humedad de tierra, categoría “Ambiente”

```
1.     Blockly.Arduino.ambient_humid_humiditySoil_alarm = function() {
2.         var code = '';
3.         var dropdown_pin = this.getFieldValue('PIN');
4.         Blockly.Arduino.setups['setup_analog_read' + dropdown_pin] = JST['inout_analog_read_setups']({dropdown_pin:dropdown_pin
});
5.         var hum_high = this.getFieldValue('hum_high');
6.         var hum_high_c = (hum_high * 1023)/100;
7.         var high = Blockly.Arduino.statementToCode(this,'HIGH') || '';
8.         var hum_low = this.getFieldValue('hum_low');
9.         var hum_low_c = (hum_low * 1023)/100;
10.        var low = Blockly.Arduino.statementToCode(this,'LOW') || '';
11.        var codehigh = 'else{\n'+indentSentences(low)+'\n\n';
12.        var code_low = 'if(+JST['inout_analog_read']({dropdown_pin: dropdown_pin})+'<'+hum_low_c+''){\nwhile('+JST['inout_analog_
read']({dropdown_pin: dropdown_pin})+'<'+hum_high_c+''){\n'+indentSentences(high)+'\n\n';
13.        code += code_low + codehigh;
14.        return [code,Blockly.Arduino.CODE_ATOMIC];
15.    };
16.
17.
18.    Blockly.Blocks.ambient_humid_humiditySoil_alarm = {
19.        category: Facilino.locales.getKey('LANG_CATEGORY_AMBIENT'),
20.        category_colour: Facilino.LANG_COLOUR_AMBIENT,
21.        colour: Facilino.LANG_COLOUR_AMBIENT,
22.        keys: ['LANG_HUMID_READ_HUMID_SOIL_TOOLTIP'],
23.        output: 'number',
24.        init: function() {
25.            this.setColour(Facilino.LANG_COLOUR_AMBIENT);
26.            this.appendDummyInput('').appendField(new Blockly.FieldImage("img/blocks/humidity_analog.svg",20*options.zoom,20*options.zoom)
)
27.                .appendField(new Blockly.FieldImage("img/blocks/soil.svg",48*options.zoom,20*options.zoom)).setAlign(Blockly.ALIGN_RIGHT);
28.            this.appendDummyInput('').appendField(new Blockly.FieldImage("img/blocks/analog_signal.svg", 20*options.zoom, 20*options.zoom)
)
29.                .setAlign(Blockly.ALIGN_RIGHT).appendField(new Blockly.FieldDropdown(profiles.default.analog), 'PIN');
30.            this.appendStatementInput('HIGH').appendField(newBlockly.FieldImage("img/blocks/humidity_low.svg",20*options.zoom,20*options.z
oom))
31.                .appendField(new Blockly.FieldTextInput(""), 'hum_low')
32.                .appendField(new Blockly.FieldLabel("%"))
33.            this.appendStatementInput('LOW').appendField(newBlockly.FieldImage("img/blocks/humidity_high.svg",20*options.zoom,20*options.z
oom))
34.                .appendField(new Blockly.FieldTextInput(""), 'hum_high')
35.                .appendField(new Blockly.FieldLabel("%"))
36.            this.setInputsInline(true);
37.            this.setPreviousStatement(true, null);
38.            this.setNextStatement(true, null);
39.
40.            this.setTooltip(Facilino.locales.getKey('LANG_HUMID_READ_HUMID_SOIL_TOOLTIP'));
41.        }
42.    };
```

## Bloque de alarma temperatura, categoría “Ambiente”

```
1.     Blockly.Arduino.ambient_temp_temperatureDHT_alarm = function() {
2.         var code = '';
3.         var dropdown_pin = this.getFieldValue('PIN');
4.         var temp_high = this.getFieldValue('temp_high');
5.         var high = Blockly.Arduino.statementToCode(this,'HIGH') || '';
6.         var temp_low = this.getFieldValue('temp_low');
7.         var low = Blockly.Arduino.statementToCode(this,'LOW') || '';
8.         var type = 'DHT11';
9.         //var once = this.getFieldValue('ONCE');
10.        Blockly.Arduino.definitions['dht']=JST['dht_definitions_include']({});
11.        Blockly.Arduino.definitions['declare_var_define_dht'+type+dropdown_pin]=JST['dht_definitions_variables']({pin :dropdown_p
in, type: type});
12.        Blockly.Arduino.setups['setup_dht' + dropdown_pin] = JST['dht_setups']({pin: dropdown_pin, type: type});
13.        var code_low='if(sensor'+type+'_'+dropdown_pin+'.readTemperature()
>'+temp_high+''){\nwhile(sensor'+type+'_'+dropdown_pin+'.readTemperature() >'+temp_low+''){\n'+indentSentences(high)+'\n\n'
14.        var codehigh='else{\n'+indentSentences(low)+'\n\n'
15.        code += code_low + codehigh;
16.        return [code,Blockly.Arduino.CODE_ATOMIC];
17.    };
18.
19.
20.    Blockly.Blocks.ambient_temp_temperatureDHT_alarm = {
21.        category: Facilino.locales.getKey('LANG_CATEGORY_AMBIENT'),
22.        category_colour: Facilino.LANG_COLOUR_AMBIENT,
23.        colour: Facilino.LANG_COLOUR_AMBIENT,
24.        keys: ['LANG_TEMP_READ_TEMP_DHT_TOOLTIP'],
25.        output: 'number',
26.        init: function() {
27.            this.setColour(Facilino.LANG_COLOUR_AMBIENT);
28.            this.appendDummyInput('').appendField(new Blockly.FieldImage("img/blocks/thermometer.svg",20*options.zoom,20*options.zoom))
29.                .appendField(new Blockly.FieldImage("img/blocks/dht11.svg",20*options.zoom,20*options.zoom)).setAlign(Blockly.ALIGN_RIGHT)
;
30.            this.appendDummyInput('').appendField(new Blockly.FieldImage("img/blocks/digital_signal.svg", 20*options.zoom, 20*options.zoom)
))
31.                .setAlign(Blockly.ALIGN_RIGHT).appendField(new Blockly.FieldDropdown(profiles.default.digital), 'PIN');
```

```

32.      this.appendStatementInput('HIGH').appendField(newBlockly.FieldImage("img/blocks/thermometer_high.svg",20*options.zoom,20*optio
ns.zoom))
33.          .appendField(new Blockly.FieldTextInput(""), 'temp_high')
34.          .appendField(new Blockly.FieldLabel("°C"))
35.      this.appendStatementInput('LOW').appendField(newBlockly.FieldImage("img/blocks/thermometer_low.svg",20*options.zoom,20*options
.zoom))
36.          .appendField(new Blockly.FieldTextInput(""), 'temp_low')
37.          .appendField(new Blockly.FieldLabel("°C"))
38.      this.setInputsInline(true);
39.      this.setPreviousStatement(true, null);
40.      this.setNextStatement(true, null);
41.
42.      this.setTooltip(Facilino.locales.getKey('LANG_TEMP_READ_TEMP_DHT_TOOLTIP'));
43.    }
44.  };

```

## Bloque de alarma luminosidad, categoría “Ambiente”

```

1.      Blockly.Arduino.ambient_light_sensor_alarm = function() {
2.        var code = '';
3.        var dropdown_pin = this.getFieldValue('PIN');
4.        Blockly.Arduino.setups['setup_analog_read' + dropdown_pin] = JST['inout_analog_read_setups']({'dropdown_pin':dropdown_pin
});
5.        var light_high = this.getFieldValue('light_high');
6.        var light_high_c = (light_high * 1023)/100;
7.        var high = Blockly.Arduino.statementToCode(this, 'HIGH') || '';
8.        var light_low = this.getFieldValue('light_low');
9.        var light_low_c = (light_low * 1023)/100;
10.       var low = Blockly.Arduino.statementToCode(this, 'LOW') || '';
11.       var codehigh = 'else{\n'+indentSentences(low)+'\n'\n'
12.       var codelow = 'if('+JST['inout_analog_read']({'dropdown_pin': dropdown_pin})+'<'+light_low_c+'){\n'+indentSentences(high)+
'\n}\n'
13.       code += codelow + codehigh
14.       return [code,Blockly.Arduino.CODE_ATOMIC];
15.     };
16.
17.     Blockly.Blocks.ambient_light_sensor_alarm = {
18.       category: Facilino.locales.getKey('LANG_CATEGORY_AMBIENT'),
19.       category_colour: Facilino.LANG_COLOUR_AMBIENT,
20.       colour: Facilino.LANG_COLOUR_AMBIENT,
21.       keys: ['LANG_LIGHT_SENSOR_TOOLTIP'],
22.       output: 'number',
23.       init: function() {
24.         this.setColour(Facilino.LANG_COLOUR_AMBIENT);
25.         this.appendDummyInput('').appendField(new Blockly.FieldImage("img/blocks/sun.svg", 20*options.zoom, 20*options.zoom))
26.           .appendField(new Blockly.FieldImage("img/blocks/ldr.svg", 48*options.zoom, 20*options.zoom)).setAlign(Blockly.ALIGN_RIGHT);
27.         this.appendDummyInput('').appendField(new Blockly.FieldImage("img/blocks/analog_signal.svg", 20*options.zoom, 20*options.zoom)
).setAlign(Blockly.ALIGN_RIGHT).appendField(new Blockly.FieldDropdown(profiles.default.analog), 'PIN');
28.         this.appendStatementInput('HIGH').appendField(new Blockly.FieldImage("img/blocks/moon.svg", 20*options.zoom, 20*options.zoom))
.appendField(new Blockly.FieldTextInput(""), 'light_low')
.appendField(new Blockly.FieldLabel("%"))
31.         this.appendStatementInput('LOW').appendField(new Blockly.FieldImage("img/blocks/sun.svg", 20*options.zoom, 20*options.zoom))
.appendField(new Blockly.FieldTextInput(""), 'light_high')
.appendField(new Blockly.FieldLabel("%"))
34.         this.setInputsInline(true);
35.         this.setPreviousStatement(true, null);
36.         this.setNextStatement(true, null);
37.
38.         this.setTooltip(Facilino.locales.getKey('LANG_LIGHT_SENSOR_TOOLTIP'));
39.       }
40.     };
41.   };

```

