



## **ESCUELA UNIVERSITARIA POLITÉCNICA**

**Grado en Ingeniería Electrónica Industrial y Automática**

# **TRABAJO FIN DE GRADO**

**TFG. Nº: 770G01A138**

**TÍTULO: DISEÑO DE UN EQUIPO TIME-LAPSE PARA  
SEGUIMIENTO DE OBRAS**

**AUTOR: YOLANDA PICALLO CAAMAÑO**

**TUTOR: ANTONIO MASDÍAS BONOME**

**FECHA: SEPTIEMBRE DE 2018**

Fdo.: EL AUTOR

Fdo.: EL TUTOR

**TÍTULO: DISEÑO DE UN EQUIPO TIME-LAPSE PARA  
SEGUIMIENTO DE OBRAS**

---

## **INDICE GENERAL**

---

**PETICIONARIO: ESCUELA UNIVERSITARIA POLITÉCNICA**

**AVDA. 19 DE FEBRERO, S/N**

**15405 - FERROL**

**FECHA: SEPTIEMBRE DE 2018**

**AUTOR: EL ALUMNO**

**Fdo.: YOLANDA PICALLO CAAMAÑO**

# 1 ÍNDICE GENERAL

## 1.1 Índice de contenidos

	Páginas
1 ÍNDICE GENERAL .....	3
1.1 Índice de contenidos .....	3
1.2 Índice de figuras .....	9
1.3 Índice de tablas .....	12
2 MEMORIA.....	16
2.1 Objeto.....	16
2.2 Alcance .....	16
2.3 Antecedentes. ....	16
2.4 Normas y referencias. ....	19
2.4.1 Disposiciones legales y normas aplicadas. ....	19
2.4.2 Programas de cálculo. ....	20
2.4.3 Otras referencias. ....	20
2.5 Definiciones y abreviaturas. ....	21
2.6 Análisis de las soluciones. ....	22
2.6.1 Hardware .....	22

2.6.1.1	Raspberry Pi .....	22
2.6.1.2	Arduino.....	26
2.6.1.3	Display LCD con I2D.....	29
2.6.1.4	Teclado matricial 4x3 .....	31
2.6.1.5	Raspberry Pi Camera Rev. 1.3 .....	32
2.6.1.6	Sistema Pan/Tilt con microservomotores.....	33
2.6.2	Software. ....	34
2.6.2.1	Raspbian Debian Jessie .....	35
2.6.2.2	OSMC .....	36
2.6.2.3	LibreELEC.....	37
2.6.2.4	RetroPie .....	38
2.6.2.5	Windows 10 IoT Core.....	39
2.6.2.6	FFmpeg.....	40
2.7	Resultados finales.....	44
2.7.1	Funcionamiento general. ....	44
2.7.2	Modos de funcionamiento.....	45
2.7.2.1	Modo Manual. ....	45
2.7.2.2	Modo Automático. ....	46
2.7.3	Diseño de la caja contenedora. ....	46
3	ANEXOS.....	52

3.1	Documentación de partida. ....	52
3.2	Código fuente.....	55
3.2.1	Código fuente Arduino.....	55
3.2.1.1	Librerías.....	56
3.2.1.2	Declaración e inicialización de variables.....	56
3.2.1.3	Función void setup().....	58
3.2.1.4	Función void loop().....	59
3.2.1.5	Función Num1().....	62
3.2.1.6	Función Num2().....	64
3.2.1.7	Función menu().....	66
3.2.2	Código fuente Python.....	70
3.3	Configuración del hardware.....	72
3.4	Configuración del firmware.....	73
3.4.1	Instalación del software de Arduino.....	73
3.4.2	Instalación del S.O Raspbian.....	75
3.4.3	Instalación de VNC Viewer y configuración de dirección IP Estática. .....	78
3.4.4	Ejecución automática del script de Python.....	84
3.4.5	Instalación de FFmpeg en Raspbian.....	86
3.5	Configuración del menú de usuario.....	87

3.5.1	Pantalla inicial.....	87
3.5.2	Pantalla para modos de funcionamiento.....	87
3.5.3	Primera pantalla para modo manual: Posición eje X. ....	88
3.5.4	Segunda pantalla para modo manual: Posición eje Y. ....	90
3.5.5	Tercera pantalla para modo manual: Intervalo entre capturas.....	91
3.5.6	Pantalla durante captura de fotografías.....	92
3.6	Montaje de la caja. ....	96
3.6.1	Paso 1: Montaje de los componentes electrónicos.....	96
3.6.2	Paso 2: Unión de las partes de la caja. ....	97
3.6.3	Paso 3: Montaje de la tapa. ....	98
3.7	Montaje del vídeo time-lapse. ....	100
3.8	Posibles mejoras para realizar. ....	101
3.8.1	Subida de las fotografías a la nube. ....	101
3.8.2	Mejora de la caja contenedora.....	101
4	PLANOS .....	104
4.1	Ensamblaje de la caja. ....	104
4.2	Partes frontal y trasera de la caja.....	105
4.3	Tapa de la caja.....	106
4.4	Lateral izquierdo de la caja. ....	107
4.5	Lateral derecho de la caja. ....	108

4.6	Base de la caja (Parte 1 de 3).....	109
4.7	Base de la caja (Parte 2 de 3).....	110
4.8	Base de la caja (Parte 3 de 3).....	111
4.9	Esquemas del hardware de Raspberry Pi.....	112
5	PLIEGO DE CONDICIONES .....	115
5.1	Condiciones operativas del sistema.....	115
5.2	Especificación de los materiales. ....	115
5.3	Pruebas de verificación.....	116
5.4	Condiciones de almacenamiento. ....	116
5.5	Manual de uso.....	117
5.5.1	Alimentación del equipo.....	117
5.5.2	Configuración del menú.....	118
5.5.3	Apagado del equipo.....	119
5.5.4	Precauciones.....	120
6	ESTADO DE MEDICIONES .....	123
6.1	Materiales.....	123
6.1.1	Dispositivos electrónicos .....	123
6.1.2	Fungibles .....	126
6.1.3	Elementos de estructura.....	127
6.2	Mano de obra.....	129

7	PRESUPUESTO.....	132
7.1	Presupuesto de materiales.....	132
7.1.1	Dispositivos electrónicos. ....	132
7.1.2	Fungibles. ....	134
7.1.3	Elementos de estructura. ....	135
7.1.4	Presupuesto total de materiales. ....	137
7.2	Presupuesto total de mano de obra. ....	138
7.3	Presupuesto final. ....	139



## 1.2 Índice de figuras

	Páginas
Figura 2.6.1.1.1 - Raspberry Pi 3 Modelo B .....	24
Figura 2.6.1.1.2 - Pines GPIO Raspberry Pi 3 Modelo B .....	25
Figura 2.6.1.2.1 - Universo Arduino.....	26
Figura 2.6.1.3.1 - Display LCD I2D, parte delantera.....	30
Figura 2.6.1.3.2 - Display LCD I2D, parte trasera .....	30
Figura 2.6.1.4.1 - Teclado de membrana .....	31
Figura 2.6.1.5.1 - Raspberry Pi Camera Rev. 1.3 .....	32
Figura 2.6.1.6.1 - Kit Pan-Tilt con microservos.....	33
Figura 2.6.2.1.1 - Escritorio de Raspbian .....	35
Figura 2.6.2.2.1 - Escritorio de OSMC .....	36
Figura 2.6.2.3.1 - Escritorio de LibreELEC.....	37
Figura 2.6.2.4.1 - Escritorio de RetroPie .....	38
Figura 2.6.2.5.1 - Escritorio de Windows 10 IoT Core.....	39
Figura 2.7.3.1 - Caja contenedora (Parte de arriba).....	47
Figura 2.7.3.2 - Caja contenedora (Lateral).....	47
Figura 2.7.3.3 - Caja contenedora (Parte frontal) .....	48
Figura 2.7.3.4 - Caja contenedora (Final).....	48

Figura 3.3.1 - Conexionado del hardware .....	72
Figura 3.4.1.1 - Descarga Arduino IDE 1.8.5.....	73
Figura 3.4.1.2 - Carpeta Libraries Arduino .....	74
Figura 3.4.2.1 - Descarga de NOOBS.....	75
Figura 3.4.2.2 - Programa SD Card Formatter .....	76
Figura 3.4.2.3 - Interfaz de NOOBS .....	77
Figura 3.4.2.4 - Instalación de Raspbian.....	77
Figura 3.4.3.1 - Configuración de Raspberry Pi .....	78
Figura 3.4.3.2 - Comando ifconfig .....	79
Figura 3.4.3.3 - Configuración IP Estática I.....	80
Figura 3.4.3.4 - Configuración IP Estática II.....	81
Figura 3.4.3.5 - Nueva conexión (Ethernet) .....	82
Figura 3.4.3.6 - Nueva conexión (WiFi).....	82
Figura 3.4.3.7 - Autenticación VNC .....	83
Figura 3.4.3.1 - Escritorio Raspbian VNC .....	83
Figura 3.5.1.1 - Menú: Pantalla inicial .....	87
Figura 3.5.2.1 - Menú: Modos de funcionamiento .....	88
Figura 3.5.3.1 - Menú – Posición servo horizontal .....	88
Figura 3.5.3.2 - Menú: Error posición servo horizontal.....	89
Figura 3.5.3.3 - Menú: Posición servo horizontal, dato introducido .....	89

Figura 3.5.4.1 - Menú: Posición servo vertical.....	90
Figura 3.5.4.2 - Menú: Error posición servo vertical .....	90
Figura 3.5.4.3 - Menú: Posición servo vertical, dato introducido .....	91
Figura 3.5.5.1 - Menú: Intervalo entre fotografías .....	92
Figura 3.5.6.1 - Menú: Cuenta regresiva I.....	93
Figura 3.5.6.2 - Menú: Cuenta regresiva II.....	93
Figura 3.5.6.3 - Menú: Cuenta regresiva III.....	94
Figura 3.5.6.4 - Menú: Cuenta regresiva IV .....	94
Figura 3.5.6.5 - Menú: Cuenta regresiva V .....	95
Figura 3.6.1.1 - Taladrado de la base .....	96
Figura 3.6.1.2 - Ensamblado de los componentes a la base.....	97
Figura 3.6.2.1 - Unión de las partes de la caja I.....	97
Figura 3.6.2.2 - Unión de las partes de la caja II.....	98
Figura 3.6.3.1 - Montaje de la tapa.....	99
Figura 5.5.1.1 - Alimentación del equipo .....	117
Figura 5.5.2.1 - LED Rojo cámara Raspberry Pi .....	118
Figura 5.5.3.1 - Apagado de Raspberry Pi (Parte I) .....	119
Figura 5.5.3.2 - Apagado de la Raspberry Pi (Parte II) .....	120

### 1.3 Índice de tablas

	Páginas
Tabla 2.6.1.1.1 - Tabla comparativa de modelos Raspberry Pi 3.....	23
Tabla 2.6.1.2.1 - Tabla comparativa de modelos de Arduino .....	27
Tabla 6.1.1.1 - Tabla de componentes electrónicos (Parte I) .....	123
Tabla 6.1.1.2 - Tabla de componentes electrónicos (Parte II) .....	124
Tabla.6.1.1.3 - Tabla de componentes electrónicos (Parte III) .....	125
Tabla 6.1.3.1 - Tabla de elementos de estructura (Parte I) .....	127
Tabla 6.1.3.2 - Tabla de elementos de estructura (Parte II) .....	128
Tabla 6.2.1 – Tabla de mano de obra .....	129
Tabla 7.1.1.1 - Tabla de presupuesto de dispositivos electrónicos (Parte I) ..	132
Tabla 7.1.1.2 - Tabla de presupuesto de dispositivos electrónicos (Parte II) .	133
Tabla 7.1.2.1 - Tabla de presupuesto de fungibles .....	134
Tabla 7.1.3.1 - Tabla de presupuesto de elementos de estructura (Parte I) ..	135
Tabla 7.1.3.2 - Tabla de presupuesto de elementos de estructura (Parte II) .	136
Tabla 7.1.4.1 - Tabla de presupuesto total de materiales .....	137
Tabla 7.2.1 - Tabla de presupuesto total de mano de obra .....	138
Tabla 7.3.1 - Tabla de presupuesto final .....	139

**TÍTULO: DISEÑO DE UN EQUIPO TIME-LAPSE PARA  
SEGUIMIENTO DE OBRAS**

---

## **MEMORIA**

---

**PETICIONARIO: ESCUELA UNIVERSITARIA POLITÉCNICA**

**AVDA. 19 DE FEBRERO, S/N**

**15405 - FERROL**

**FECHA: SEPTIEMBRE DE 2018**

**AUTOR: EL ALUMNO**

**Fdo.: YOLANDA PICALLO CAAMAÑO**

## INDICE MEMORIA

	Páginas
2 MEMORIA.....	16
2.1 Objeto.....	16
2.2 Alcance .....	16
2.3 Antecedentes. ....	16
2.4 Normas y referencias. ....	19
2.4.1 Disposiciones legales y normas aplicadas. ....	19
2.4.2 Programas de cálculo. ....	20
2.4.3 Otras referencias. ....	20
2.5 Definiciones y abreviaturas. ....	21
2.6 Análisis de las soluciones. ....	22
2.6.1 Hardware .....	22
2.6.1.1 Raspberry Pi .....	22
2.6.1.2 Arduino.....	26
2.6.1.3 Display LCD con I2D.....	29
2.6.1.4 Teclado matricial 4x3 .....	31
2.6.1.5 Raspberry Pi Camera Rev. 1.3 .....	32
2.6.1.6 Sistema Pan/Tilt con microservomotores.....	33

2.6.2	Software. ....	34
2.6.2.1	Raspbian Debian Jessie .....	35
2.6.2.2	OSMC .....	36
2.6.2.3	LibreELEC.....	37
2.6.2.4	RetroPie .....	38
2.6.2.5	Windows 10 IoT Core.....	39
2.6.2.6	FFmpeg.....	40
2.7	Resultados finales.....	44
2.7.1	Funcionamiento general. ....	44
2.7.2	Modos de funcionamiento.....	45
2.7.2.1	Modo Manual. ....	45
2.7.2.2	Modo Automático. ....	46
2.7.3	Diseño de la caja contenedora. ....	46

## **2 MEMORIA**

### **2.1 Objeto**

El presente proyecto tiene como objeto diseñar un equipo compacto capaz de integrarse en cualquier tipo de obra y que permita realizar un time-lapse de dicha obra con el fin de poder realizar un video de su ejecución.

### **2.2 Alcance**

Se deberá diseñar el equipo que tendrá que ser configurado mediante entorno web o mediante display, pudiendo establecer al menos la frecuencia de disparo de cada frame.

El diseño estará basado en open hardware con almacenamiento local y/o remoto, desarrollando el diseño hardware y firmware necesario.

### **2.3 Antecedentes.**

La técnica de cámara rápida o también conocida como time-lapse es una técnica fotográfica que consiste en una secuencia de fotografías (en ocasiones vídeos) tomadas con un cierto lapso de separación entre sí, y que posteriormente son editadas para crear un vídeo que da la sensación de estar grabado en cámara rápida. Por lo general, con esta técnica, se capturan sucesos que se desarrollan a velocidades muy lentas y que el ojo humano es incapaz de percibirlos a simple vista.



Los usos de la técnica time-lapse son muy variados. Algunos casos en los que se utiliza esta técnica fotográfica son los siguientes:

- Para crear vídeos relacionados con toda clase de flora y fauna que habita en la naturaleza. Por ejemplo, el crecimiento de una flor o la creación de un hormiguero.
- Para crear vídeos de paisajes, tales como amaneceres y atardeceres, tormentas y relámpagos, etc.
- Para promoción turística. Un vídeo time-lapse, con la combinación de imágenes adecuada, tiene la capacidad de potenciar y resaltar las peculiaridades de la región en cuestión que se quiere publicitar.
- Para crear vídeos del desarrollo de un evento. Un uso muy extendido es realizar el time-lapse de una feria, ya que se observa ver el montaje y el desmontaje del stand en la feria, así como la cantidad de personas que la visitan.
- Para documentar el tránsito de una calle, la construcción de un edificio, y todo aquello en donde con el paso del tiempo se cree o se transforme algo atractivo.

La utilidad que tendrá esta técnica en este proyecto es la del seguimiento de una obra, recogiendo así, en segundos o minutos, una obra desde el inicio hasta el final, cuando está lista para su entrega o uso.

Un reportaje fotográfico de una obra realizado con cierta periodicidad da una visión muy clara de la evolución de ésta. Las fotografías pueden servir también para ayudar a elaborar informes y son testigos del estado de la obra.

Para su correcto desarrollo, la localización del equipo time-lapse en la construcción es un factor determinante, ya que una vez decidido el lugar de instalación, éste deberá mantenerse a lo largo de toda la obra, lo que puede ser meses, semanas o años.

En un lugar en el que todo cambia a lo largo de los meses, como es en este caso una obra, habrá que prever conscientemente el lugar donde debe colocarse. Debe de estar colocado en un lugar que tenga una buena vista de la obra, pero que al mismo tiempo no se pueda ver comprometida su existencia al cabo de unos meses.

Además, se debe adaptar el equipo a las condiciones particulares de cada obra, de su ambiente y de su entorno, teniendo en cuenta las condiciones meteorológicas en la zona, como pueden ser la lluvia, el viento, la temperatura en invierno o en verano, etc.

Una vez instalado el equipo en el lugar definitivo y adecuado a las singularidades de este, se programan los parámetros del equipo y el intervalo con el que se van a tomar las fotografías durante la obra. Estos parámetros pueden variar en diferentes fases de la obra por lo que es conveniente poder modificarlos a posteriori.

Una vez terminado el proyecto de la toma de fotografías time-lapse se realiza su montaje. La edición del vídeo se hace de forma personalizada para cada proyecto u obra, utilizando distintos programas y software para fotografía y vídeo.

A continuación, a modo de referencia, se indican algunos ejemplos de campos en los que se aplica la técnica de vídeo time-lapse para seguimiento de obras.

- Edificios, casas, naves y centros comerciales.
- Infraestructuras, obras de urbanización.
- Ingeniería civil.
- Construcción de puentes.
- Reconstrucción y restauración de construcciones arquitectónicas protegidas o de relevancia histórica o artística.

## **2.4 Normas y referencias.**

### **2.4.1 Disposiciones legales y normas aplicadas.**

En este apartado se contempla el conjunto de disposiciones legales y normas de obligado cumplimiento para la realización de este proyecto.

- Reglamento del trabajo de fin de grado de la Escuela Universitaria Politécnica.
- UNE-EN-ISO 216. (Papel de escritura y ciertos tipos de impresos. Formatos acabados, series A y B).
- UNE 82100. (Magnitudes y unidades (partes de 0 a 13)).
- UNE 50132. (Numeración de las divisiones y subdivisiones en los documentos escritos).
- UNE 1027. Dibujo Técnico. Plegado de planos.
- UNE 1032. Dibujos Técnicos. Principios generales de representación.
- UNE EN ISO 7200. Documentación técnica de productos. Campos de datos en bloques de títulos y en cabeceras de documentos.
- UNE 1039. Dibujos Técnicos. Acotación. Principios generales, definiciones, métodos de ejecución e indicaciones especiales.
- UNE 1135. Dibujos Técnicos. Lista de elementos.
- UNE-EN-ISO 3098-0. Documentación técnica de productos. Escritura. Requisitos generales. (ISO 3098-0:1997).
- UNE-EN-ISO 5455. Dibujos Técnicos. Escalas. (ISO 5455:1979).

### 2.4.2 Programas de cálculo.

En este apartado se indica la relación de programas utilizados para el correcto desarrollo del proyecto.

- Arduino IDE 1.8.5. Entorno de desarrollo de Arduino.
- VNC Viewer. Software de pantalla compartida que facilita el acceso remoto a cualquier dispositivo.
- SD Card Formatter. Software para realizar el formateo de tarjetas SD.
- WinSCP. Software para facilitar la transferencia segura de archivos entre dos sistemas informáticos, el local y uno remoto que ofrezca servicios SSH.
- IDLE. Entorno de desarrollo de Python.
- Notepad++. Editor de texto y de código fuente libre con soporte para varios lenguajes de programación.
- Fritzing v0.9.3b. Software libre utilizado para realizar el esquema del conexionado del hardware.
- SolidWorks 2016. Software CAD para modelado mecánico en 2D y 3D. Ha sido utilizado para realizar la elaboración de los planos.
- FFmpeg: Software utilizado para realizar vídeos time-lapse a partir de las imágenes capturadas.

### 2.4.3 Otras referencias.

- <https://www.arduino.cc/>
- <https://projects.raspberrypi.org/en/projects/noobs-install>
- <https://www.python.org/>
- <https://docs.python.org/3/>

## 2.5 Definiciones y abreviaturas.

- CAD: Diseño Asistido por Computadora
- SBC: Single Board Computer. Ordenador de placa reducida.
- IoT: Internet of Things. Es un concepto que se refiere a la interconexión digital de objetos cotidianos con Internet.
- PoE: Power over Ethernet. Alimentación a través de Ethernet.
- SDA: Serial Data.
- SCL: Serial Clock.
- CSI: Camera Serial Interface. Interfaz entre una cámara digital y un procesador anfitrión.
- S.O.: Sistema Operativo.
- LXDE: Lightweight X11 Desktop Environment. Entorno de escritorio libre.
- IDLE: Integrated Development Environment for Python. Entorno de desarrollo de Python.
- NOOBS: New Out of Box Software. Aplicación para la instalación fácil de S.O. en Raspberry Pi.
- SSH: Secure Shell. Protocolo de administración remota.
- VNC: Virtual Network Computing. Software libre basado en una arquitectura cliente-servidor que permite observar las acciones del ordenador servidor a través del ordenador cliente.
- OSMC: Open Source Media Center. S.O. orientado a la reproducción de contenido multimedia.
- LibreELEC: Libre Embedded Linux Entertainment Center. S.O. orientado a la reproducción de contenido multimedia.
- SDL: Simple DirectMedia Layer. Es un conjunto de bibliotecas desarrolladas en lenguaje de programación C que proporcionan funciones como gestión de audio y vídeo, así como también carga y gestión de imágenes.

## **2.6 Análisis de las soluciones.**

Para llevar a cabo el diseño del equipo time-lapse, se ha hecho un estudio previo de diferentes soluciones tanto de hardware como de software, en orden de escoger las opciones que mejor se adapten a las necesidades y/o requerimientos del proyecto.

### **2.6.1 Hardware**

En este apartado se realiza una explicación detallada de los componentes utilizados en este proyecto.

#### **2.6.1.1 Raspberry Pi**

Este dispositivo es un ordenador de placa reducida o placa única (SBC) desarrollado en Reino Unido por la Fundación Raspberry Pi.

El software que utiliza es open source, siendo su sistema operativo oficial una versión adaptada de Debian, denominada Raspbian. También permite utilizar otros sistemas operativos, entre éstos una versión de Windows 10. La fundación Raspberry Pi da soporte para la descarga del sistema operativo Raspbian.

A continuación, se muestra una tabla comparativa de los diferentes modelos de Raspberry Pi.







						
<b>Raspberry Pi</b>	1				2	3
<b>Modelo</b>	A	A+	B	B+	B	B
<b>SoC</b>	Broadcom BCM2835				Broadcom BCM2836	Broadcom BCM2837
<b>CPU</b>	700MHz ARM1176JFZ-S				900MHz Quadcore ARM Cortex A7	1,2GHz Quad Cortex A53
<b>RAM</b>	256Mb	512Mb			1Gb	
<b>USB</b>	1		2	4		
<b>Video</b>	RCA, HDMI	Jack, HDMI	RCA, HDMI	Jack, HDMI		
<b>Audio</b>	Jack, HDMI					
<b>Boot</b>	Memoria SD	Memoria MicroSD	Memoria SD	Memoria MicroSD		
<b>Wireless</b>	No					802 11n Bluetooth 4.1
<b>Ethernet</b>	No		Ethernet 10/100			
<b>Alimentación</b>	5V/2V					5V/2.5V
<b>GPIO</b>	26	40	26	40		

Tabla 2.6.1.1.1 - Tabla comparativa de modelos Raspberry Pi 3

En marzo de 2018 se lanzó al mercado la Raspberry Pi 3 Modelo B+, siendo el último modelo que hay en la actualidad. Presenta las siguientes mejoras con respecto al modelo anterior:

- Mejora del procesador ARM Cortex A53 que ahora es de 1,4 GHz frente a los 1,2 GHz del modelo anterior.
- El bluetooth pasa a ser 4.2 frente al 4.1 del modelo anterior.
- Wi-Fi actualizado, pasando a tener banda dual, tanto de 2,4 GHz como de 5 GHz (802.11ac). Con esta nueva característica se trata de mejorar la compatibilidad con diversos dispositivos IoT.
- Se triplica la capacidad de red de la conexión Ethernet vía USB, pasando de 100 Mbits/s a 300 Mbits/s.
- El Ethernet dispone de PoE gracias a lo cual no es necesario conectar un adaptador de corriente para poder encenderla, ya obtiene la suficiente energía a través de ese puerto.

El modelo utilizado en este proyecto ha sido la Raspberry Pi 3 Modelo B, ya que cuando comenzó la fase de desarrollo del proyecto era el modelo más potente y actualizado que había en el mercado.

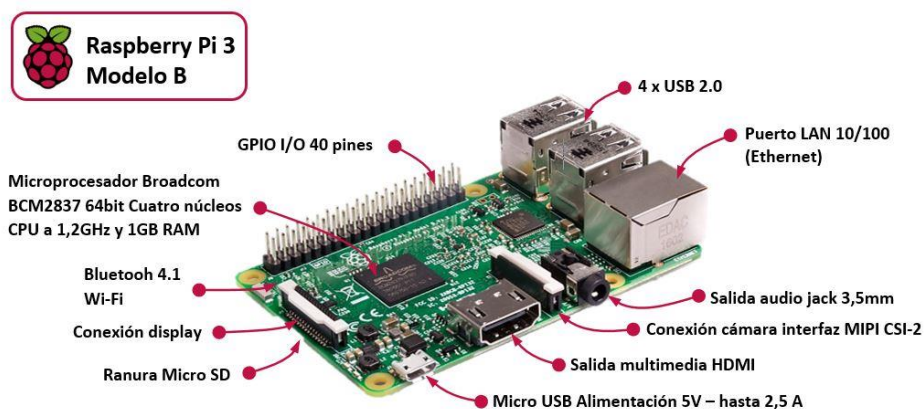


Figura 2.6.1.1.1 - Raspberry Pi 3 Modelo B



En la siguiente imagen, se muestran de forma más detallada los pines GPIO de la Raspberry Pi 3 Model B, de los cuales se hace uso en el proyecto

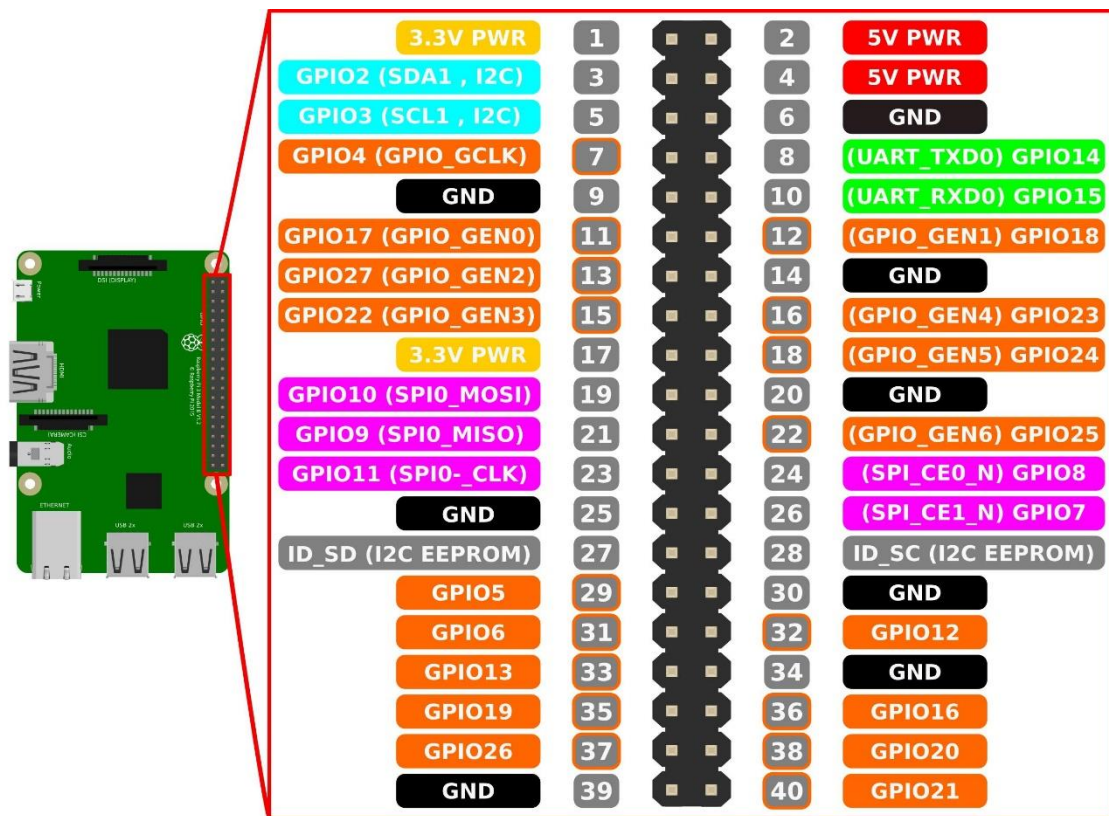
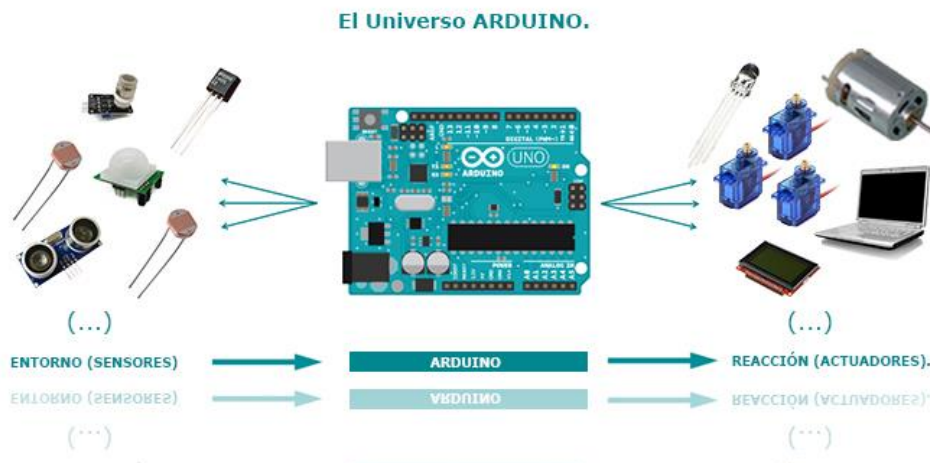


Figura 2.6.1.1.2 - Pines GPIO Raspberry Pi 3 Modelo B

### 2.6.1.2 Arduino

Arduino es una plataforma de hardware y software libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

Esta placa cuenta con infinidad de sensores y periféricos pensados para ser utilizados con Arduino, lo que permite crear una amplia variedad de proyectos.



**Figura 2.6.1.2.1 - Universo Arduino**

Algunas de las ventajas más relevantes de Arduino son las siguientes:

- Es Open Source.
- Utiliza un entorno fácil de programar ya que cuenta con infinidad de librerías libres que se pueden utilizar y/o modificar dependiendo de las necesidades del proyecto.
- No necesita ninguna tarjeta de programación como sucede en la mayoría de las placas de desarrollo, ya que Arduino cuenta con un software conocido

como bootloader que viene cargado en el microprocesador y que se programa a sí mismo de forma automática.

- Bajo coste.
- Se pueden crear una gran variedad de aplicaciones con un coste muy bajo y con poco tiempo de desarrollo.

A continuación, se muestra una tabla con diferentes modelos de placas de Arduino.







						
<b>Modelo</b>	Arduino UNO	Arduino Mega2560	Arduino Leonardo	Arduino Due	Arduino ADK	Arduino Esplora
<b>ucontrolador</b>	ATMega 328	ATMega 2560	ATMega 32u4	AT91SA M3X8E	ATMega 3560	ATMega 32u4
<b>Pines digitales</b>	14	54	20	54	54	-
<b>Pines PWM</b>	6	15	7	12	15	-
<b>Pines analógicos</b>	6	16	12	12	16	-
<b>Memoria</b>	32K	256K	32K	512K	256K	32K
<b>Reloj</b>	16 MHz			84 MHz	16 MHz	
<b>Conexión</b>	USB	Micro USB			USB	Micro USB
<b>Tensión de operación</b>	5V			3,3V	5V	
<b>Corriente Máx. Pines E/S</b>	40 mA			130 mA	40 mA	-
<b>Alimentación</b>	7 – 12 V					5V

Tabla 2.6.1.2.1 - Tabla comparativa de modelos de Arduino

En este proyecto, se ha utilizado la placa Arduino Uno, de la que ya se disponía durante el desarrollo del proyecto y la cual cubre las necesidades de este.

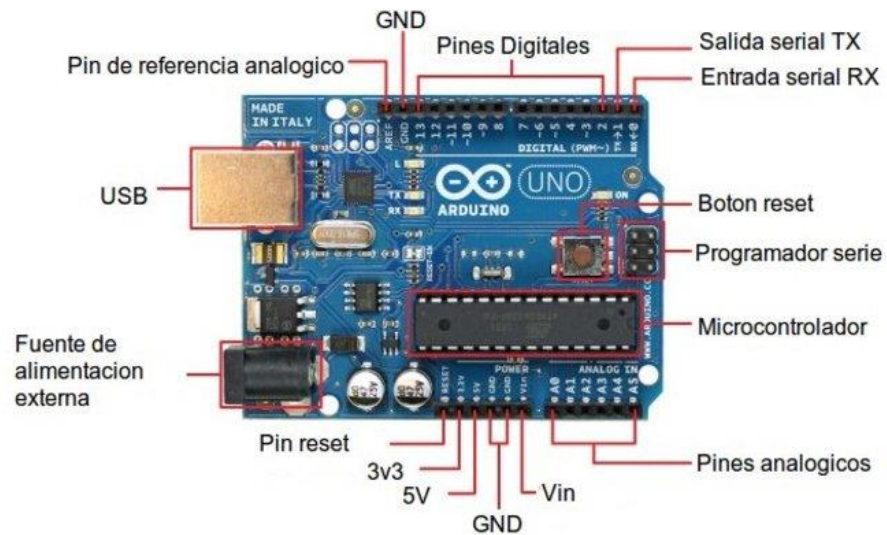


Figura 2.6.1.2.2 – Arduino Uno

### 2.6.1.3 Display LCD con I2D

I2C (Inter-Integrated Circuit) es un bus de comunicaciones serie el cual fue desarrollado por Philips en 1982 para la comunicación interna de dispositivos electrónicos en sus artículos. Posteriormente fue adoptado progresivamente por otros fabricantes hasta convertirse en un estándar del mercado.

Se trata de un protocolo de dos hilos de control, uno para transmitir los datos, SDA y otro, el reloj asíncrono que indica cuando leer los datos SCL. También dispone de un hilo GND y otro VCC.

En el Arduino UNO, que es el modelo que se ha utilizado en este proyecto, los pines I2C están en los pines analógicos A4, que se conecta a SDA, y A5, que se conecta a SCL.

El bus I2C tiene una arquitectura de tipo maestro-esclavo. El dispositivo maestro inicia la comunicación con los esclavos y puede mandar o recibir datos de los esclavos. Los esclavos no pueden iniciar la comunicación, el maestro tiene que preguntarles, ni hablar entre si directamente.

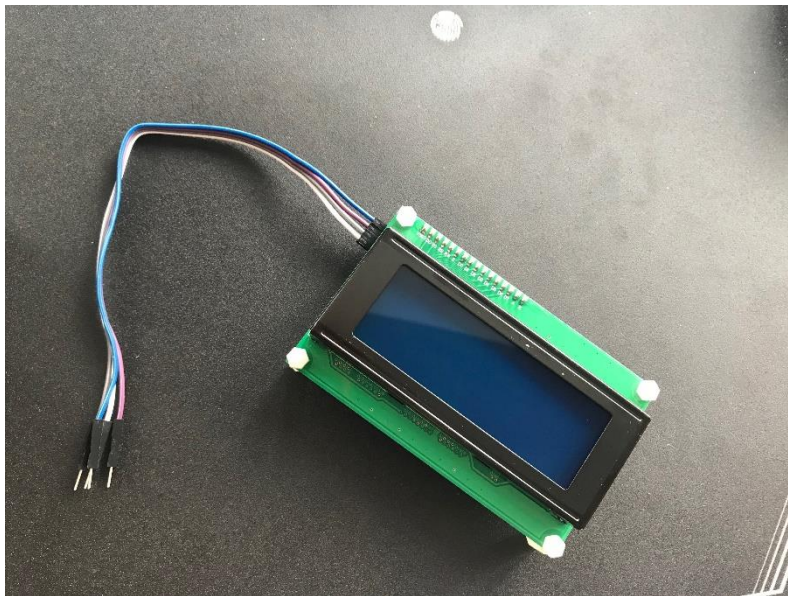
Este bus es síncrono. El maestro proporciona una señal de reloj, que mantiene sincronizados a todos los dispositivos del bus. De esta forma, se evita que cada dispositivo tenga su propio reloj.

Cada dispositivo conectado al bus posee una dirección única. En el caso de este proyecto la dirección I2C por defecto es 0x3F. Para identificar la dirección específica de este módulo se ha utilizado un pequeño sketch llamado I2C Scanner, el cual realiza un barrido por todas las posibles direcciones del bus y muestra el resultado en caso de encontrar un dispositivo en la dirección. Es muy importante identificar correctamente la dirección I2C del módulo ya que de otra forma el programa no funcionaría correctamente.

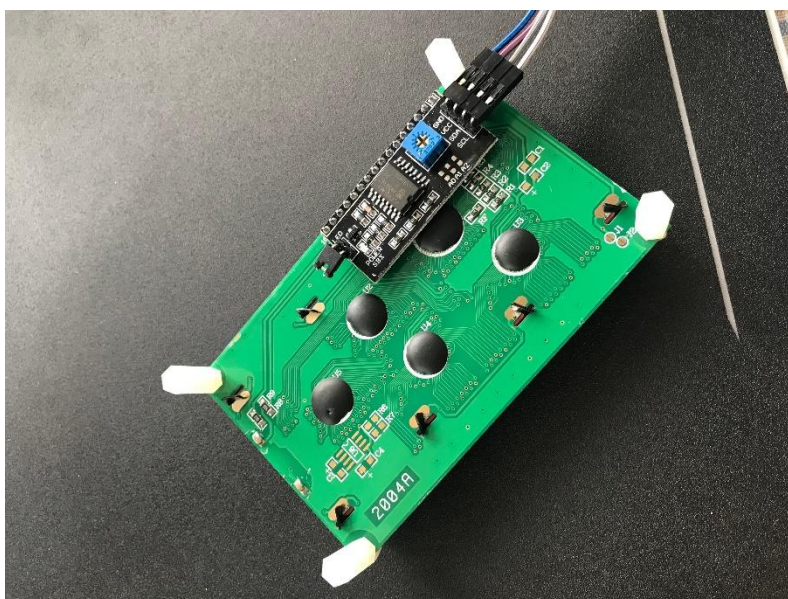
Existe una librería de Arduino, cuyo nombre es Wire, que gestiona el protocolo de comunicaciones completo, lo cual facilita mucho la implementación.

En cuando al LCD, se ha utilizado un display de 20 filas por 4 columnas que ya trae incorporado el módulo I2C.

Para controlar el contraste de los dígitos en el LCD se ha ajustado el potenciómetro que se encuentra en el módulo hasta obtener el resultado deseado.



**Figura 2.6.1.3.1 - Display LCD I2D, parte delantera**



**Figura 2.6.1.3.2 - Display LCD I2D, parte trasera**

### 2.6.1.4 Teclado matricial 4x3

Para poder interactuar con el display LCD e introducir valores y parámetros propios de la configuración del time-lapse, se ha utilizado un teclado matricial de 3 columnas por 4 filas.

Este modelo de teclado es de membrana, ultrafino y dispone de un adhesivo en la parte trasera para poder colocarlo donde sea conveniente, lo cual facilita su montaje en la maqueta del proyecto.

Dispone de 7 pines de conexión con Arduino y para su integración en el proyecto se ha de utilizar la librería keypad.h.



Figura 2.6.1.4.1 - Teclado de membrana

### 2.6.1.5 Raspberry Pi Camera Rev. 1.3

Se trata de un módulo de cámara HD genérico, compatible con los modelos de Raspberry Pi 1, 2 y 3. Especialmente utilizada para aplicaciones móviles, robótica, seguridad y monitoreo.

- Resolución para fotografía de 5 Megapíxeles.
- Soporte para vídeo: 1080p, 720p, 640x480p.
- No cuenta con soporte para audio.
- Utiliza una interface CSI.



Figura 2.6.1.5.1 - Raspberry Pi Camera Rev. 1.3



### 2.6.1.6 Sistema Pan/Tilt con microservomotores

Se ha utilizado este módulo con dos servomotores SG90 para dotar a la cámara de movimiento horizontal (Pan) y vertical (Tilt). El módulo tiene un soporte plástico donde se sujeta la cámara.

Cada servomotor permite hacer un barrido de  $-90^{\circ}$  a  $90^{\circ}$ , lo que quiere decir que el ángulo de giro es de  $180^{\circ}$ .

Estos servomotores funcionan con una señal PWM, con un pulso de trabajo entre 1 ms y 2 ms y con un período de 20 ms.

Para el conexionado con Arduino, cada servomotor dispone de 3 cables. Se ha de incluir la librería servo.h en la programación.

Se ha escogido este modelo de microservo por su pequeño tamaño y peso, lo que hace muy sencilla su integración en el proyecto.



**Figura 2.6.1.6.1 - Kit Pan-Tilt con microservos**

### 2.6.2 Software.

Para la parte software, se ha realizado un estudio de los diferentes sistemas operativos que pueden emplearse en la Raspberry Pi. Posteriormente, se analizan las ventajas e inconvenientes de cada uno de ellos con el objetivo de escoger la opción más adecuada para este proyecto.

Un sistema operativo (S.O.) es el software principal o conjunto de programas de un sistema informático que gestiona todos los recursos del sistema informático, tanto de hardware (partes físicas, disco duro, pantalla, teclado, etc.) como de software (programas e instrucciones), permitiendo así la comunicación entre el usuario y el ordenador.

Las funciones básicas de un S.O. son las siguientes:

- Administrar los recursos del ordenador.
- Coordinar el hardware.
- Organizar archivos y directorios en los dispositivos de almacenamiento del ordenador.
- Determinar qué aplicaciones se deben ejecutar, en qué orden y cuánto tiempo.

En un ordenador, el S.O. puede ser almacenado en un disco duro, y determinadas partes de él son cargadas en la memoria del ordenador (RAM). En el caso de la Raspberry Pi, se instala en una tarjeta microSD.

Actualmente, existen diferentes tipos de S.O. que funcionan sobre la Raspberry Pi. La elección de uno u otro radica en el uso que se le vaya a dar en cada proyecto: Como PC o servidor, como centro multimedia o como emulador de consolas.

### 2.6.2.1 Raspbian Debian Jessie

Raspbian es una distribución del S.O. GNU/Linux. Por lo tanto, se trata de un software libre y de código abierto. Es el S.O. oficial soportado por la Fundación Raspberry Pi. La versión más reciente es la Debian Jessie la cual dispone de una versión reducida llamada Debian Jessie Lite.

La distribución usa LXDE como entorno de escritorio y Chromium como navegador web. Contiene herramientas de desarrollo como IDLE para el lenguaje de programación Python o Scratch. Además, incluye aplicaciones como LibreOffice para el procesamiento de textos, hojas de cálculo, presentaciones con diapositivas, etc.



Figura 2.6.2.1.1 - Escritorio de Raspbian

Se puede instalar a través de la aplicación NOOBS. Esta aplicación consiste en una imagen ISO que facilita la instalación de diversas distribuciones Linux en una Raspberry Pi. Es instalada en una tarjeta SD y no necesita de acceso a internet para ser utilizada.

NOOBS contiene soluciones de S.O. como Raspbian, Raspbian Lite, LibreELEC, OSMC, y en su última versión, Windows IoT 10 Core.

### 2.6.2.2 OSMC

Es un S.O. especialmente diseñado para ser utilizado como Mediacenter. Está basado en Raspbian por lo que se tiene acceso a más de 30.000 paquetes en sus repositorios que permiten adaptar OSMC a las necesidades del usuario.

Este S.O. incluye y ejecuta KODI, una aplicación que permite acceder a una gran cantidad de contenido multimedia. También incluye un servidor Samba, con el cual se puede acceder desde Windows como si fuera una carpeta compartida más, y servidor SSH, para poder acceder a la Raspberry Pi y poder ejecutar órdenes a través de la línea de comandos.

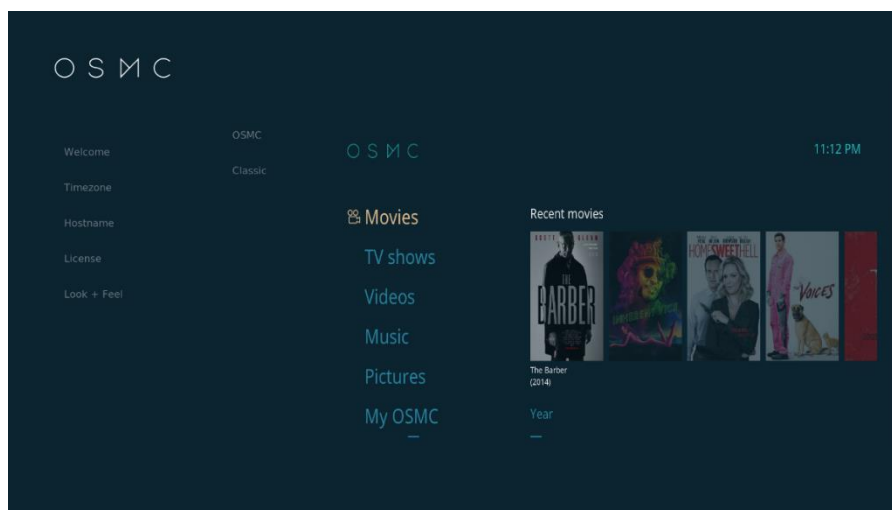


Figura 2.6.2.2.1 - Escritorio de OSMC

### 2.6.2.3 LibreELEC

Es un S.O. orientado específicamente a reproducción de contenido multimedia. Está basado en Linux y como en OSMC, utiliza KODI para la reproducción del contenido.

Se trata de una evolución del S.O. OpenELEC, el cual es un proyecto anterior. Tanto OpenELEC como LibreELEC tienen una configuración mínima que incluye únicamente la capacidad de ejecutar KODI.

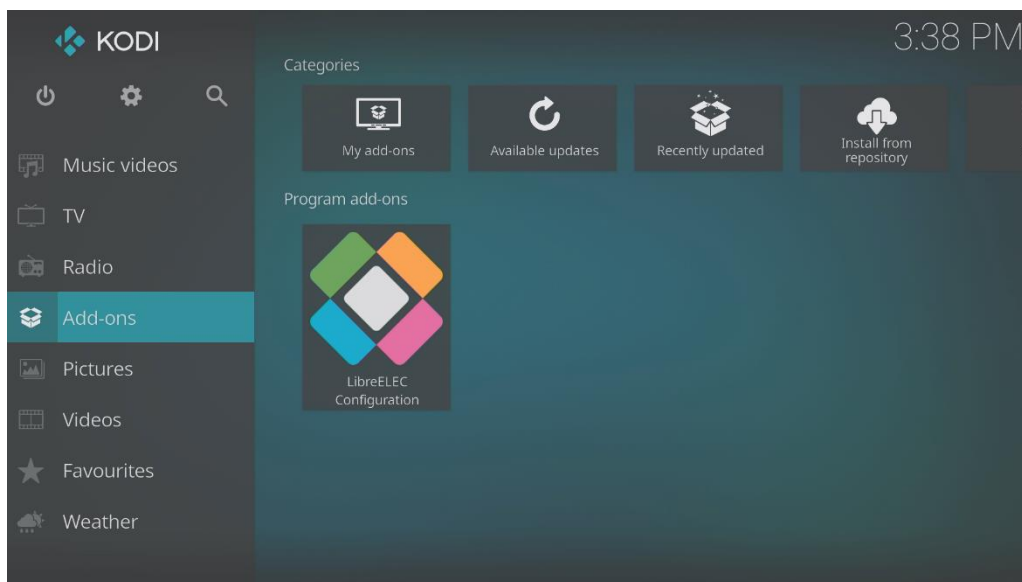


Figura 2.6.2.3.1 - Escritorio de LibreELEC

### 2.6.2.4 RetroPie

Se trata de un S.O. para Raspberry Pi basado en Raspbian que incluye por defecto una capa de personalización con una interfaz desde donde ejecutar los diferentes emuladores de consolas retro.

Toda la interfaz de este S.O., así como los emuladores, son de código abierto por lo que cualquier usuario puede colaborar en el desarrollo y reportar posibles errores detectados durante la instalación o el juego.

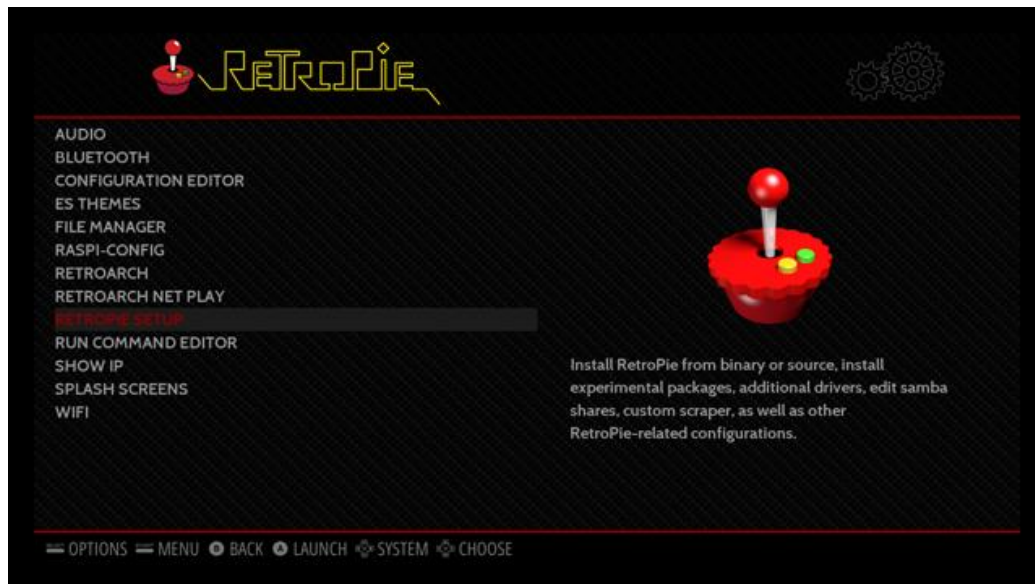


Figura 2.6.2.4.1 - Escritorio de RetroPie

### 2.6.2.5 Windows 10 IoT Core

Este S.O. es una versión del núcleo de Windows 10 adaptada para el IoT.

La gran desventaja respecto a las distribuciones Linux es que no cuenta con un entorno de escritorio ni interfaz gráfico ni tampoco existe la posibilidad de instalar ningún software o ejecutar ninguna aplicación tradicional de Windows 10, por lo que es menos flexible.

Esta plataforma ha sido diseñada principalmente con la intención de invitar a desarrolladores a crear todo tipo de soluciones adaptadas a la domótica, la robótica y el IoT, no centrándose en el segmento de los PCs y portátiles tradicionales.

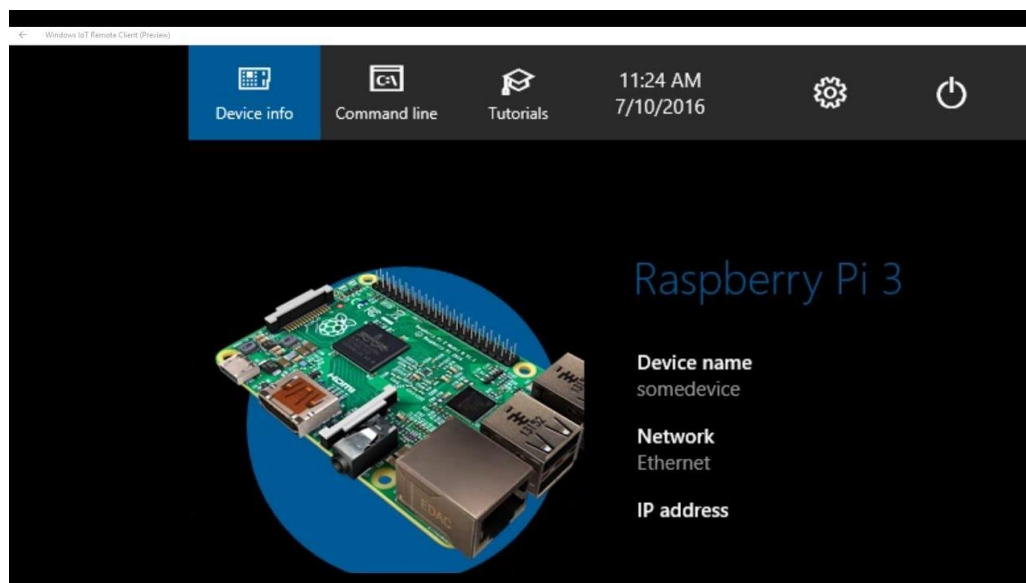


Figura 2.6.2.5.1 - Escritorio de Windows 10 IoT Core

Después de haber estudiado diferentes S.O. para Raspberry Pi, el S.O. elegido para su posterior instalación ha sido Raspbian, ya que en este proyecto se utilizará la Raspberry Pi como PC para realizar parte de la programación del proyecto en Python mediante el entorno de desarrollo IDLE.

Continuando con el software, además del S.O. a escoger, también se ha realizado un estudio previo orientado a la elección del programa que posteriormente transformará las imágenes capturadas por el equipo en un vídeo time-lapse.

#### **2.6.2.6 FFmpeg**

La herramienta FFmpeg se trata de una colección de software libre, requisito muy importante para este proyecto, que puede grabar, convertir y hacer streaming de audio y vídeo. Este programa está desarrollado en GNU/Linux, pero es multiplataforma, por lo que puede ser instalado en la mayoría de S.O.

Algunas de las posibilidades que ofrece este software son las siguientes:

- Cambiar de formato.
- Comprimir y extraer imágenes y vídeo.
- Hacer un vídeo a partir de imágenes.
- Insertar marcas de agua (texto o fotografías) en un vídeo.
- Codificar vídeo para reproducir en cualquier equipo o dispositivo.
- Dividir o unir vídeos.
- Extraer el audio o música de los vídeos.



El proyecto FFmpeg está compuesto por:

- FFmpeg. Es un sistema de conversión de vídeo y audio de un formato a otro. Algunos de estos formatos son, por ejemplo, MP4 o MPEG para vídeo, y mp3 o amr para audio.
- FFserver. Es un servidor de streaming multimedia que permite transmitir vídeo y audio que haya sido convertido previamente a alguno de los formatos soportados por FFmpeg.
- FFplay. Es un reproductor multimedia basado en SDL y en las bibliotecas FFmpeg.
- Libavcodec. Se trata de una biblioteca que contiene todos los códecs de vídeo y audio (codificadores, decodificadores o ambos) de FFmpeg. Entre la lista de códecs que incluye se encuentran MPEG, MJPEG o H.264/MPEG-4 para vídeo y MP2, MP3 o WMA para audio.

FFmpeg es una herramienta de línea de comandos, lo que quiere decir que el trabajo con el programa se realiza a través del Terminal. A continuación, se muestran algunos ejemplos de comandos para manipulación de vídeo y audio en FFmpeg.

- Cambiar formato de vídeo de WMV a FLV.

```
ffmpeg -i entrada.wmv salida.flv
```

- Cambiar el formato de vídeo de WMV a FLV especificando parámetros.

```
ffmpeg -i entrada.wmv -ab 56 -ar 22050 -r 15 -s 320x240  
salida.flv
```

Donde:

-ab: Es el número de bits por segundo para el audio. A mayor número, mayor calidad.

-ar: Es la frecuencia del sonido en Hz.

-r: Indica el número de imágenes por segundo.

-s: Indica la resolución del vídeo.

- Cambiar el formato de vídeo de FLV a MP4 especificando parámetros.

```
ffmpeg -i entrada.flv -vcodec mpeg4 -s 640x360 -ac 2  
salida.mp4
```

Donde:

-vcodec: Es la codificación deseada en el vídeo.

-ac: Indica el canal de audio establecido (1: Mono, 2: Estéreo, 3: Envoltente).

- Extraer 120 segundos de un vídeo a partir del segundo 60.

```
ffmpeg -i entrada.mpeg -t 120 -ss 60 -sameq  
vídeo_salida.mpeg
```

Donde:

-t: Indica la duración, en segundos, del vídeo que se desea extraer.

-ss: Es el tiempo, en segundos, donde se desea cortar el vídeo.

-sameq: Parámetro que se utiliza para que el vídeo de salida mantenga la misma calidad que el de entrada.

- Convertir un vídeo en formato .mpeg en imágenes con formato .jpg.

```
ffmpeg -i video.mpeg imagen%d.jpg
```

- Convertir una secuencia de imágenes con formato .jpg en un vídeo con formato .mpeg.

```
ffmpeg -i imagen%d.jpg video.mpeg
```

- Conocer los códecs asociados a un vídeo.

```
ffmpeg -i video.avi
```

- Conocer los códecs de un audio.

```
ffmpeg -i cancion.wma
```

FFmpeg ha sido el software elegido para su utilización en este proyecto, principalmente porque se trata de un software libre y multiplataforma, el cual se puede instalar en Raspbian. Además, incluye un amplio abanico de posibilidades en cuanto a la manipulación y edición de vídeo y audio y su utilización es muy sencilla.

## **2.7 Resultados finales.**

En este capítulo se describe el equipo time-lapse según las soluciones elegidas en cuanto a software y hardware, indicando sus características finales.

### **2.7.1 Funcionamiento general.**

En cuanto a la funcionalidad de los elementos estudiados en el apartado de “Análisis de las soluciones”, es importante detallar lo siguiente:

- El Arduino controla el movimiento de los servomotores, así como ordena a la Raspberry Pi cuando tomar una fotografía. Esto último se lleva a cabo a través de un switch, obtenido por el relé de 5V conectado a uno de los pines digitales del Arduino.
- La Raspberry Pi espera que se cierre dicho switch, situado entre el pin GPIO 15 y GND.

A continuación, se explica el principio de funcionamiento, en líneas generales, del equipo time-lapse diseñado.

- Una vez alimentado el equipo, a través de una batería portátil, aparecerá el menú de usuario en el display LCD. En dicho menú, el usuario podrá escoger entre dos modos de funcionamiento, manual o automático.
- El usuario introducirá los parámetros requeridos en el menú de configuración mediante un teclado matricial.

- Después de introducir los parámetros, el equipo comenzará a capturar las fotografías.
- En ambos modos de funcionamiento existe la posibilidad de parar el equipo y volver al inicio pulsando cualquier tecla del teclado matricial instalado.
- Las capturas serán guardadas en la tarjeta de memoria microSD de la Raspberry Pi, a la cual tiene acceso el usuario.

## **2.7.2 Modos de funcionamiento.**

El equipo time-lapse diseñado incluye dos modos de funcionamiento, de los cuales se realiza su descripción en este apartado.

### **2.7.2.1 Modo Manual.**

En este modo de funcionamiento, el usuario puede configurar los siguientes parámetros del sistema:

- Ángulo de posición horizontal de la cámara. Este ángulo está delimitado entre 0° y 180°.
- Ángulo de posición vertical de la cámara. Este ángulo está delimitado entre 0° y 180°.
- Tiempo entre capturas de fotos, en segundos. Este valor tiene dos restricciones: Tiene que ser mayor que 0 y menor que 3600.

### **2.7.2.2 Modo Automático.**

Este modo de funcionamiento, los parámetros mencionados anteriormente están configurados por defecto con unos valores determinados. Estos valores son los siguientes:

- Ángulo de posición horizontal de la cámara: 70°.
- Ángulo de posición vertical de la cámara: 35°.
- Tiempo entre capturas de fotos: 3 segundos.

Una vez seleccionado este modo, el equipo comenzará a realizar fotografías de manera automática.

### **2.7.3 Diseño de la caja contenedora.**

En este apartado se muestra el resultado final de la caja contenedora y se indica, en líneas generales, cómo se ha llegado a dicho resultado.

La caja contenedora ha sido creada a partir de dos planchas de metacrilato de 50x25x4 mm, las cuales se han cortado para obtener seis partes, que conforman todas las partes de la caja.

Se ha escogido como material principal el metacrilato, por su resistencia a los golpes y por su apariencia estética, ya que, debido a su transparencia, se puede ver todo el equipo que está en el interior.

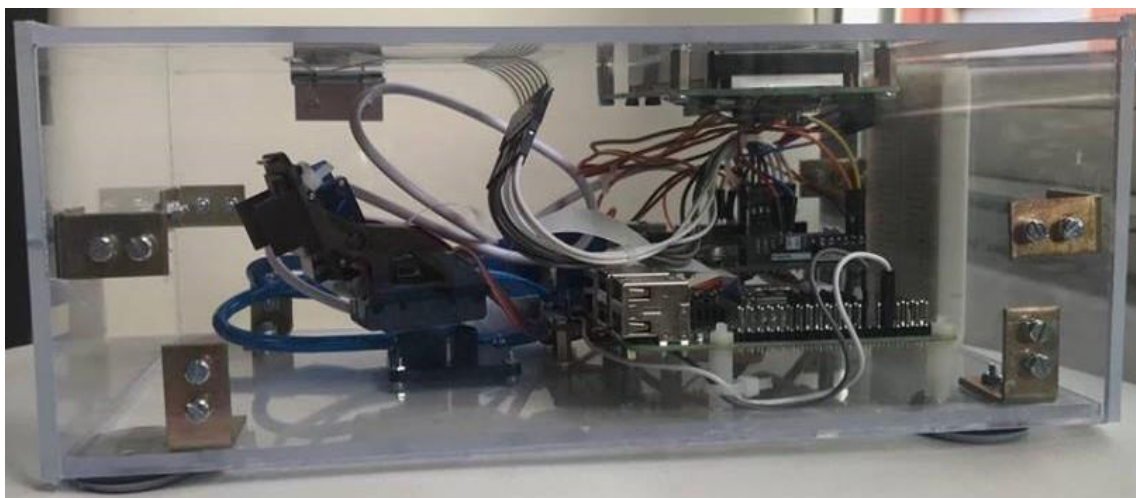
Dichas partes que forman la estructura de la caja han sido ensambladas con escuadras angulares metálicas y tornillería.

Los componentes electrónicos del equipo time-lapse han sido ensamblados a la caja mediante separadores metálicos, para facilitar la disipación del calor que transmiten algunos de ellos, y tornillería.

Se muestra el resultado final en las siguientes imágenes.



**Figura 2.7.3.1 - Caja contenedora (Parte de arriba)**



**Figura 2.7.3.2 - Caja contenedora (Lateral)**



**Figura 2.7.3.3 - Caja contenedora (Parte frontal)**



**Figura 2.7.3.4 - Caja contenedora (Final)**



**TÍTULO: DISEÑO DE UN EQUIPO TIME-LAPSE PARA  
SEGUIMIENTO DE OBRAS**

---

## **ANEXOS**

---

**PETICIONARIO: ESCUELA UNIVERSITARIA POLITÉCNICA**  
**AVDA. 19 DE FEBRERO, S/N**  
**15405 - FERROL**

**FECHA: SEPTIEMBRE DE 2018**

**AUTOR: EL ALUMNO**

**Fdo.: YOLANDA PICALLO CAAMAÑO**

## INDICE ANEXOS

	Páginas
3 ANEXOS.....	52
3.1 Documentación de partida. ....	52
3.2 Código fuente.....	55
3.2.1 Código fuente Arduino .....	55
3.2.1.1 Librerías .....	56
3.2.1.2 Declaración e inicialización de variables .....	56
3.2.1.3 Función void setup().....	58
3.2.1.4 Función void loop().....	59
3.2.1.5 Función Num1() .....	62
3.2.1.6 Función Num2() .....	64
3.2.1.7 Función menu() .....	66
3.2.2 Código fuente Python. ....	70
3.3 Configuración del hardware. ....	72
3.4 Configuración del firmware.....	73
3.4.1 Instalación del software de Arduino. ....	73
3.4.2 Instalación del S.O Raspbian.....	75

3.4.3	Instalación de VNC Viewer y configuración de dirección IP Estática. .....	78
3.4.4	Ejecución automática del script de Python. ....	84
3.4.5	Instalación de FFmpeg en Raspbian. ....	86
3.5	Configuración del menú de usuario.....	87
3.5.1	Pantalla inicial.....	87
3.5.2	Pantalla para modos de funcionamiento.....	87
3.5.3	Primera pantalla para modo manual: Posición eje X. ....	88
3.5.4	Segunda pantalla para modo manual: Posición eje Y. ....	90
3.5.5	Tercera pantalla para modo manual: Intervalo entre capturas.....	91
3.5.6	Pantalla durante captura de fotografías.....	92
3.6	Montaje de la caja. ....	96
3.6.1	Paso 1: Montaje de los componentes electrónicos.....	96
3.6.2	Paso 2: Unión de las partes de la caja. ....	97
3.6.3	Paso 3: Montaje de la tapa. ....	98
3.7	Montaje del vídeo time-lapse. ....	100
3.8	Posibles mejoras para realizar. ....	101
3.8.1	Subida de las fotografías a la nube. ....	101
3.8.2	Mejora de la caja contenedora.....	101

### **3 ANEXOS**

#### **3.1 Documentación de partida.**

En este apartado se incluye el documento de asignación del Trabajo de Fin de Grado. En el citado documento se encuentran datos del autor, así como del tutor y una breve descripción del proyecto y de los objetivos de este.



## ESCUELA UNIVERSITARIA POLITÉCNICA

### ASIGNACIÓN DE TRABAJO FIN DE GRADO

En virtud de la solicitud efectuada por:  
*En virtud da solicitude efectuada por:*

**APELLIDOS, NOMBRE:** Picallo Caamaño, Yolanda  
**APELIDOS E NOME:**

**DNI:** [REDACTED] **Fecha de Solicitud:** Feb2018  
**DNI:** [REDACTED] **Fecha de Solicitude:**

Alumno de esta escuela en la titulación de Grado en Ingeniería en Electrónica Industrial y Automática, se le comunica que la Comisión de Proyectos ha decidido asignarle el siguiente Trabajo Fin de Grado:

*O alumno de esta escola na titulación de Grado en Enxeñería en Electrónica Industrial e Automática, comunicaselle que a Comisión de Proxectos ha decidido asignarlle o seguinte Traballo Fin de Grado:*

**Título T.F.G.:** DISEÑO DE UN EQUIPO TIME-LAPSE PARA SEGUIMIENTO DE OBRAS

**Número TFG:** 770G01A138

**TUTOR:** (Titor) Masdías Bonome, Antonio E.

**COTUTOR/CODIRECTOR:**

**La descripción y objetivos del Trabajo son los que figuran en el reverso de este documento:**

*A descrición e obxectivos do proxecto son os que figuran no reverso deste documento.*

*Ferrol a Miércoles, 18 de Julio del 2018*

Retirei o meu Traballo Fin de Grado o día \_\_\_\_ de \_\_\_\_ do ano \_\_\_\_\_.

*Fdo: Picallo Caamaño, Yolanda*

Documento Generado automáticamente el: 18/07/2018 a las: 17:52:08 desde <https://www.eup.upc.es/trabajosfindegrado/>

**DESCRIPCIÓN Y OBJETIVO:OBJETO:**

Se trata de diseñar un equipo compacto capaz de integrarse en cualquier tipo de obra y que permita realizar un time lapse de dicha obra con el fin de poder realizar un video de su ejecución.

**ALCANCE:**

Se deberá diseñar el equipo que deberá ser configurado mediante entorno web o mediante display, pudiendo establecer al menos la frecuencia de disparo de cada frame.

El diseño estará basado en open hardware con almacenamiento local y/o remoto, desarrollando el diseño Hardware, y Firmware necesario.

## **3.2 Código fuente**

En este apartado se muestra el código principal del proyecto, compuesto por dos programas, uno para Arduino y otro para Raspberry Pi.

### **3.2.1 Código fuente Arduino**

A continuación, se muestra el código fuente para Arduino. En líneas generales, este programa realiza las siguientes funciones:

- Permite al usuario introducir los valores de posición de la cámara y el tiempo entre fotos.
- Controla el movimiento de los servomotores.
- Controla la actividad del relé, el cual ordena a la Raspberry Pi cuando capturar una foto.
- Contiene la configuración del menú que se muestra a través del display LCD.

### 3.2.1.1 Librerías

```
#include <Servo.h> //Libreria para los microservos.
#include <LiquidCrystal_I2C.h> //Libreria del display LCD
I2C.
#include <Keypad.h> //Libreria para el teclado matricial.
#include <Wire.h> //Libreria para comunicaciones I2C.
```

### 3.2.1.2 Declaración e inicialización de variables

```
//Crea el objeto LCD dirección 0x3F y 20 columnas x 4
filas.
```

```
LiquidCrystal_I2C lcd(0x3F, 20, 4);
```

```
//Instanciacion del teclado matricial.
```

```
const byte ROWS = 4; //4 filas.
const byte COLS = 3; //3 columnas.
char keys[ROWS][COLS] =
{
  {'1', '2', '3'},
  {'4', '5', '6'},
  {'7', '8', '9'},
  {'*', '0', '#'}
};
```

```
byte rowPins[ROWS] = {8, 7, 6, 5}; //Pines de las filas.
byte colPins[COLS] = {4, 3, 2}; //Pines de las columnas.
```

```
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins,
ROWS, COLS);
```



```
//Declaración e inicialización de variables.
```

```
int x=0;  
int y=0;  
int pos_ini_x = 0;  
int pos_ini_y = 0;  
int duracion = 0;  
char teclax;  
int aux;  
int aux2;
```

```
Servo pan; //Movimiento horizontal.
```

```
Servo tilt; //Movimiento vertical.
```

### 3.2.1.3 Función void setup()

```
void setup()
{
    //Servo horizontal.
    //(Pin digital 10, Pulso en us para girar 0°, Pulso en us
    // para girar 180°).
    pan.attach(10, 650, 2550);

    //Servo vertical.
    //(Pin digital 11, Pulso en us para girar 0°, Pulso en us
    // para girar 180°).
    tilt.attach(11, 650, 2550);

    pan.write(70); //Posición inicial servo horizontal.
    tilt.write(35); //Posición inicial servo vertical.

    pinMode(12, OUTPUT); //Pin digital relé 5V.

    //Parámetros iniciales para LCD.

    lcd.init();
    lcd.backlight();
    lcd.begin(20, 4);
}
```

### 3.2.1.4 Función void loop()

```
void loop()
{
    //Inicialización de variables.

    aux = 0;
    aux2 = 0;
    teclax = NO_KEY;

    menu(); //Llamada a la función menu().

    //Se escribe en el servo horizontal
    //la posición introducida por el usuario.

    pan.write(x);

    //Se escribe en el servo vertical
    //la posición introducida por el usuario.

    tilt.write(y);

    do
    {
        digitalWrite(12, HIGH); //Se activa el relé.
        lcd.setCursor(0, 0);
        lcd.print("Capturando...");
        digitalWrite(12, LOW); //Se desactiva el relé.
        lcd.setCursor(0, 2);
        lcd.print("Pulse cualquier");
        lcd.setCursor(0, 3);
        lcd.print("tecla para salir");
    }
```

```
//Bucle para la cuenta atrás del intervalo.

for (int i = 0; i < duracion; i = i + 1)

{

    lcd.setCursor(14, 0);
    lcd.print("      ");
    lcd.setCursor(14, 0);
    lcd.print(duracion - i);

    teclax = keypad.getKey(); //Lee el teclado.

    //Si se pulsa cualquier tecla, sale del bucle
    //y vuelve a la función menu().

    if (teclax != NO_KEY)

    {

        aux = 1;
        break;

    }

    for (int j = 0; j < 10; j = j + 1)

    {

        teclax = keypad.getKey(); //Lee el teclado.

        //Si se pulsa cualquier tecla, sale del bucle
        //y vuelve a la función menu().

        if (teclax != NO_KEY)

        {

            aux2 = 1;
            break;

        }

    }

}
```

```
//Se divide la función delay, en 10 veces 100 ms,  
//para que haya tiempo suficiente para leer la  
// tecla pulsada  
  
    delay(100);  
  
}  
  
//Condición para que salga del bucle for principal  
//y vuelva al inicio.  
  
if (aux2 == 1)  
{  
    break;  
}  
  
}  
  
//Condición para que salga del bucle for principal  
//y vuelva al inicio.  
  
if (aux == 0 && aux2 == 0)  
{  
    teclax = keypad.getKey();  
}  
  
} while (teclax == NO_KEY);  
  
}
```

### 3.2.1.5 Función Num1()

```
//Función que guarda el valor de un número
//introducido por teclado y lo muestra en la pantalla.
//Esta función es para guardar y mostrar las posiciones
//de los servos, introducidas por el usuario.

int Num1()

{
    int num = 0;
    char tecla = keypad.getKey(); //Lee el teclado.

    //Con la tecla #, continúa a la siguiente pantalla.

    while (tecla != '#')

    {
        switch (tecla)

        {
            case NO_KEY:
                break;
            case '0':
            case '1':
            case '2':
            case '3':
            case '4':
            case '5':
            case '6':
            case '7':
            case '8':
            case '9':

                lcd.print(tecla);

                //Transforma el valor decimal(dec) en
                //carácter(char) (Tabla ASCII)
                //El carácter '0' en ASCII es 48 en decimal, por lo
                //que, por ejemplo para la tecla 1 (en decimal 49)
                // se resta 48
                //y se obtiene 1 que es el número
                //que se quiere mostrar en pantalla.

                num = num * 10 + (tecla - '0');
```

```
//Restricción de la posición del servo, entre 0° y
//180°

if (num > 180)

{
  num = 0;
  lcd.setCursor(0, 1);
  lcd.print("Num.entre 0 y 180!");
  lcd.setCursor(16, 0);
  lcd.print("  ");
  lcd.setCursor(16, 0);
}

break;

//Con la tecla *, se borra el numero introducido.

case '*':

  num = 0;
  lcd.setCursor(16, 0);
  lcd.print("  ");
  lcd.setCursor(16, 0);
  break;
}

tecla = keypad.getKey();

}

return num;

}
```

### 3.2.1.6 Función Num2()

```
//Función que guarda el valor de un número
//introducido por teclado y lo muestra en la pantalla.
//Esta función es para guardar y mostrar el valor
//del intervalo entre fotos, introducido por el usuario.
//Esta función es igual a Num1() pero con otra restricción
//a la hora de introducir el dato.
```

```
int Num2()

{
    int num2 = 0;
    char tecla2 = keypad.getKey();

    while (tecla2 != '#')

    {
        switch (tecla2)

        {

            case NO_KEY:
                break;
            case '0':
            case '1':
            case '2':
            case '3':
            case '4':
            case '5':
            case '6':
            case '7':
            case '8':
            case '9':

                lcd.print(tecla2);
                num2 = num2 * 10 + (tecla2 - '0');
```



```
//Restricción del tiempo entre capturas, entre 0 y
//3600s.

if (num2 == 0 || num2 >= 3600)

{
    num2 = 0;
    lcd.setCursor(0, 1);
    lcd.print("Num entre 0 y 3600!");
    lcd.setCursor(11, 0);
    lcd.print("      ");
    lcd.setCursor(11, 0);

}

break;

case '*':

    num2 = 0;
    lcd.setCursor(11, 0);
    lcd.print("      ");
    lcd.setCursor(11, 0);
    break;

}

tecla2 = keypad.getKey();

}

return num2;

}
```

### 3.2.1.7 Función menu()

//Función con la configuración para el menú principal.

```
void menu()

{
  //Pantalla de inicio.

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Equipo Timelapse");
  lcd.setCursor(0, 2);
  lcd.print("Pulse cualquier");
  lcd.setCursor(0, 3);
  lcd.print("tecla para continuar");

  //Devuelve la tecla pulsada si la hay.

  char tecla = keypad.getKey();

  while (tecla == NO_KEY)

  {
    //Carga el valor de la tecla.

    tecla = keypad.getKey();

  }

  //Segunda pantalla del menú.

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Escoja un modo:");
  lcd.setCursor(0, 2);
  lcd.print("1:Modo Manual");
  lcd.setCursor(0, 3);
  lcd.print("2:Modo Auto");

  //Devuelve la tecla pulsada si la hay.

  char tecla_escogida = keypad.getKey();
```

```
while (tecla_escogida == NO_KEY)
{
    //Carga el valor de la tecla.
    tecla_escogida = keypad.getKey();
}

if (tecla_escogida)
{
    switch (tecla_escogida)
    {
        case '1': //Modo manual.

            //Pantalla para introducir
            //la posición del servo horizontal.

            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("Posicion Eje X:");
            lcd.setCursor(0, 2);
            lcd.print("*: Borrar");
            lcd.setCursor(0, 3);
            lcd.print("#: Continuar");
            lcd.setCursor(16, 0);

            //Llamada a la función Num1().

            pos_ini_x = Num1();

            //Se guarda el numero devuelto
            //por la función en la variable x.

            x = pos_ini_x;

            lcd.clear();
```

```
//Pantalla para introducir
//la posición del servo vertical.

lcd.setCursor(0, 0);
lcd.print("Posicion Eje Y:");
lcd.setCursor(0, 2);
lcd.print("*: Borrar");
lcd.setCursor(0, 3);
lcd.print("#: Continuar");
lcd.setCursor(16, 0);

//Llamada a la función Num1().

pos_ini_y = Num1();

//Se guarda el numero devuelto
//por la función en la variable y.

y = pos_ini_y;

lcd.clear();

//Pantalla para introducir
//el tiempo entre fotos.

lcd.setCursor(0, 0);
lcd.print("Intervalo:");
lcd.setCursor(0, 2);
lcd.print("*: Borrar");
lcd.setCursor(0, 3);
lcd.print("#: Continuar");
lcd.setCursor(11, 0);

//Llamada a la función Num2().
//Se guarda el número devuelto
//por la función en la variable duracion.

duracion = Num2();

lcd.clear();

break;
```

```
//Modo automático con valores por defecto.  
case '2':  
  
    lcd.clear();  
    x = 70;  
    y = 35;  
    duracion = 3;  
    lcd.clear();  
    break;  
  
    }  
}  
}
```

### 3.2.2 Código fuente Python.

Este programa realiza las siguientes funciones:

- A través del pin GPIO 15 recibe tensión del relé y da la orden a la cámara de capturar una foto.
- Guarda las fotografías en un directorio en la Raspberry Pi.
- Lista dicho directorio para saber cuántas fotografías contiene.

A continuación, se muestra el código fuente para Raspberry Pi, en lenguaje de programación Python.

```
"""Módulos de la biblioteca estándar
de Python que proporcionan un conjunto
de funciones para trabajar con fechas y horas"""
import time
import datetime

"""Módulo de sistema que permite acceder
a funcionalidades dependientes del S.O."""
import os

"""Módulo de sistema que permite trabajar de forma
directa con órdenes del sistema"""
import subprocess

"""Interfaz para trabajar en Python
con los pines GPIO"""
from gpiozero import Button

"""Interfaz para trabajar en Python
con el módulo de cámara para RPi"""
from picamera import PiCamera

"""Asignación de pin GPIO"""
rele = Button(15)
```

```
with PiCamera() as camera:

    """Resolución de la cámara"""
    camera.resolution = (1024, 768)

    """Se lista el directorio donde se guardan
    las fotografías para saber cuántas contiene"""
    ruta, dirs, archivos =
next(os.walk('/home/pi/timelapse'))

    """Se imprime por pantalla ese número de fotos"""
    cont = len(archivos)
    print(cont)
    frame = 1 + cont

while True:

    """Espera la activación del relé"""
    rele.wait_for_press()

    print("Capturando foto...")

    """Captura la fotografía y la guarda en la ruta
    dispuesta con formato de fecha y hora actual"""
    dt = datetime.datetime.now()
    camera.capture('/home/pi/timelapse/IMG %s.jpg' %
        dt.strftime("%Y-%m-%d %H:%M:%S"))
    print(frame)
    frame+=1
```

### 3.3 Configuración del hardware.

A continuación, se incluye una imagen del conexionado del hardware. Este esquema se ha realizado con el programa Fritzing.

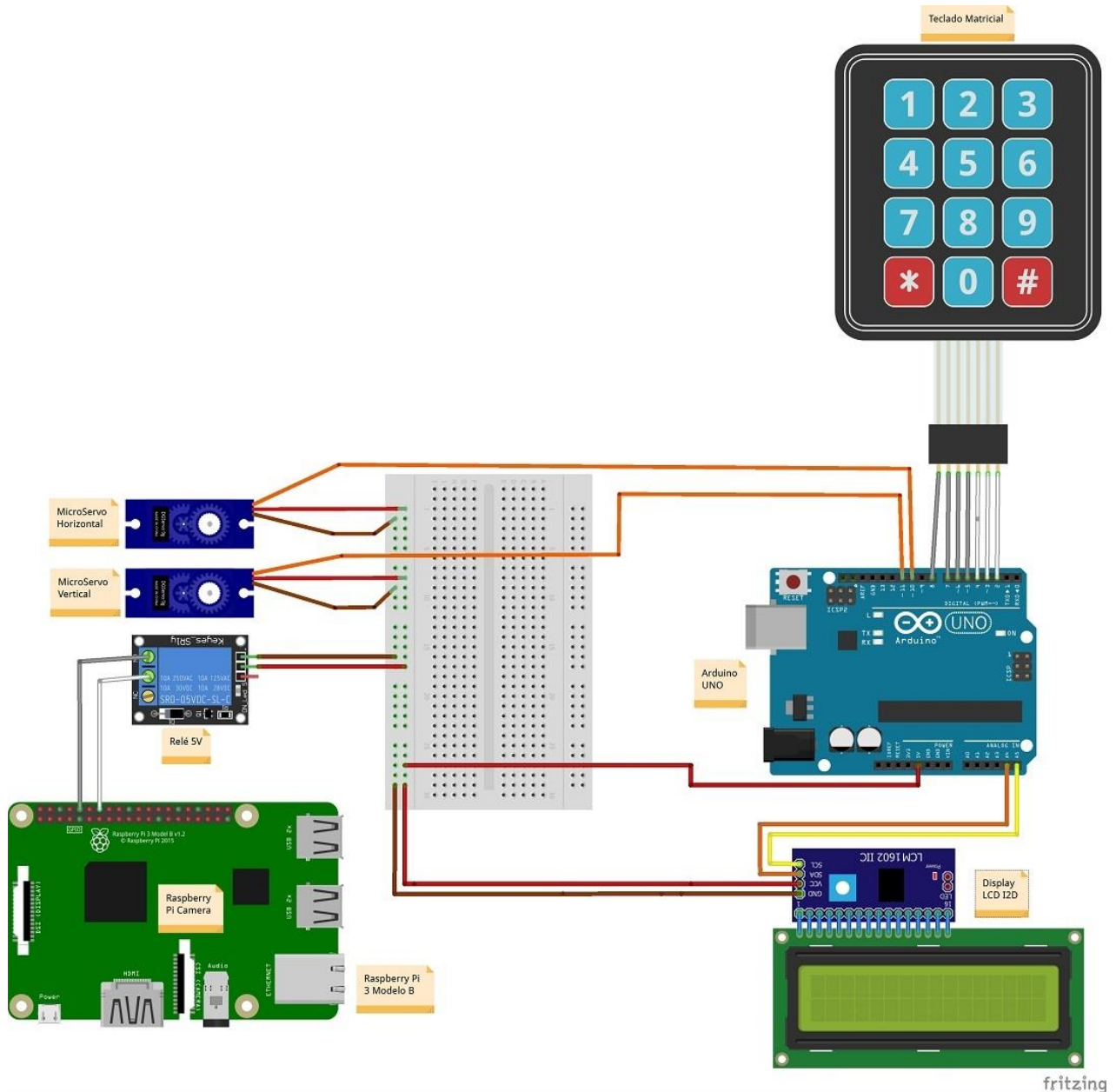


Figura 3.3.1 - Conexionado del hardware



### 3.4 Configuración del firmware.

En este apartado se detallan los pasos a seguir para realizar la instalación y configuración del software utilizado.

#### 3.4.1 Instalación del software de Arduino.

Para poder comenzar la programación, es necesario instalar el software de Arduino. La descarga del software se realiza a través de la web de Arduino:

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

En esta página se descarga el archivo adecuado en su versión comprimida para diferentes S.O. La última versión de este programa es la 1.8.5, la cual se utilizó para este proyecto.

#### Download the Arduino IDE



Figura 3.4.1.1 - Descarga Arduino IDE 1.8.5

A continuación, después de terminar la descarga, se ejecuta la instalación del software desde el archivo ejecutable *arduino-1.8.5-windows.exe*.

Una vez instalado el software, se deben de añadir las librerías necesarias para trabajar con los componentes utilizados en el proyecto. Estas librerías se deben de añadir a la carpeta *libraries* que se encuentra en los archivos de programa de Arduino. Cabe destacar que la mayoría de estas librerías ya vienen incluidas en el programa.

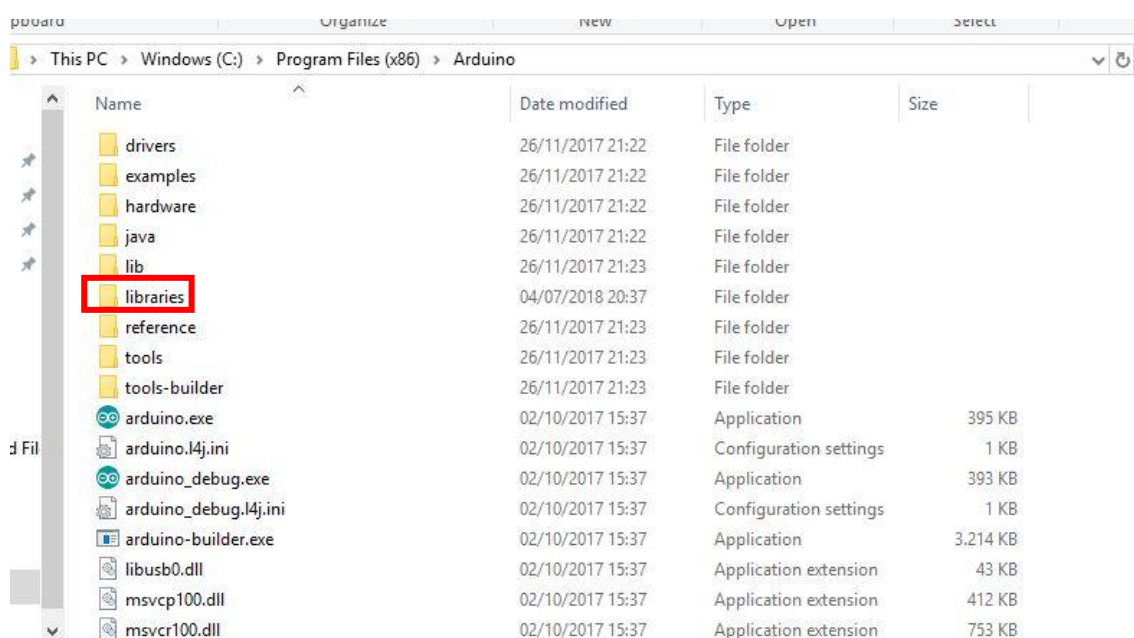


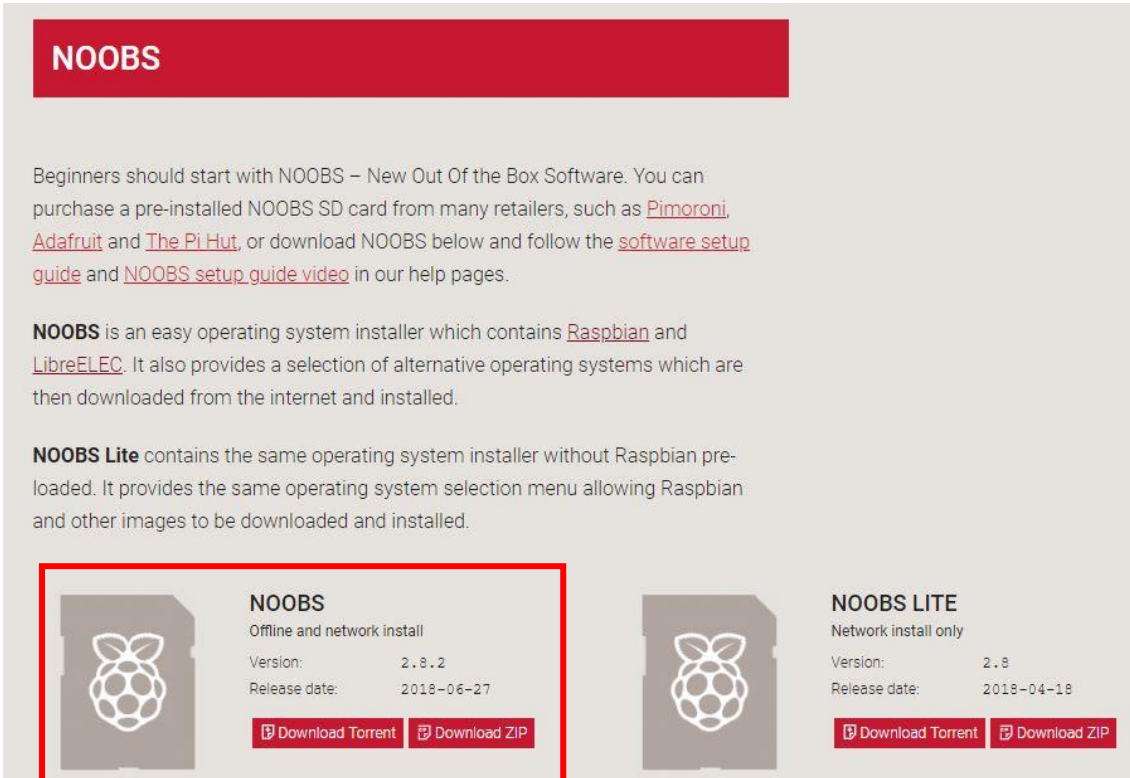
Figura 3.4.1.2 - Carpeta Libraries Arduino

### 3.4.2 Instalación del S.O Raspbian.

Para comenzar a trabajar con la Raspberry Pi se ha de instalar un S.O. Previamente se han analizado distintos S.O. que existen para Raspberry y finalmente el elegido ha sido Raspbian. Para realizar la instalación se ha utilizado NOOBS, ya que es la manera más sencilla y rápida de instalar este S.O. u otros que contiene este instalador.

Lo primero que se ha hecho es descargar la versión comprimida de NOOBS desde la página web oficial de Raspberry.

<https://www.raspberrypi.org/downloads/noobs/>



**NOOBS**

Beginners should start with NOOBS – New Out Of the Box Software. You can purchase a pre-installed NOOBS SD card from many retailers, such as [Pimoroni](#), [Adafruit](#) and [The Pi Hut](#), or download NOOBS below and follow the [software setup guide](#) and [NOOBS setup guide video](#) in our help pages.

**NOOBS** is an easy operating system installer which contains [Raspbian](#) and [LibreELEC](#). It also provides a selection of alternative operating systems which are then downloaded from the internet and installed.

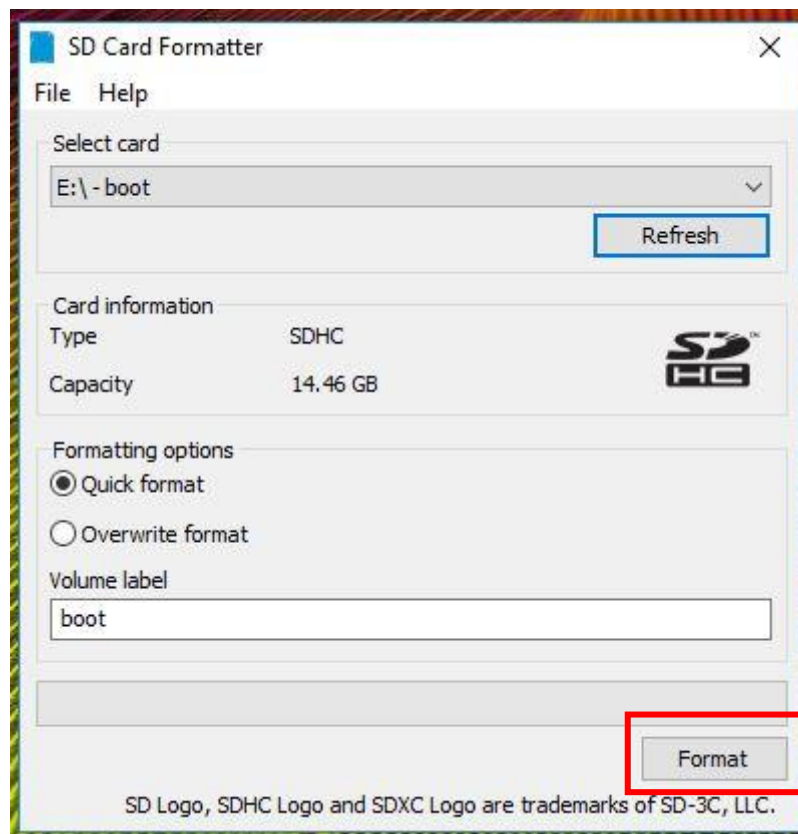
**NOOBS Lite** contains the same operating system installer without Raspbian pre-loaded. It provides the same operating system selection menu allowing Raspbian and other images to be downloaded and installed.

NOOBS	NOOBS LITE
Offline and network install	Network install only
Version: 2.8.2	Version: 2.8
Release date: 2018-06-27	Release date: 2018-04-18
<a href="#">Download Torrent</a> <a href="#">Download ZIP</a>	<a href="#">Download Torrent</a> <a href="#">Download ZIP</a>

Figura 3.4.2.1 - Descarga de NOOBS

Se deben de extraer los archivos y guardarlos en una tarjeta microSD, de mínimo 8GB, la cual se ha de formatear previamente. Para ello se ha descargado el programa SD Card Formatter desde la siguiente página web.

<https://www.sdcard.org>



**Figura 3.4.2.2 - Programa SD Card Formatter**

Una vez copiados los archivos de NOOBS en la tarjeta SD, se introduce dicha tarjeta en la Raspberry Pi. Para poder continuar con la instalación desde la Raspberry Pi se deben de conectar a la misma un monitor, un teclado y un ratón. Una vez hecho esto, se conecta el cable de alimentación eléctrica.

A continuación, se marca el S.O que se quiere instalar, en este caso Raspbian, y se procede a su instalación.

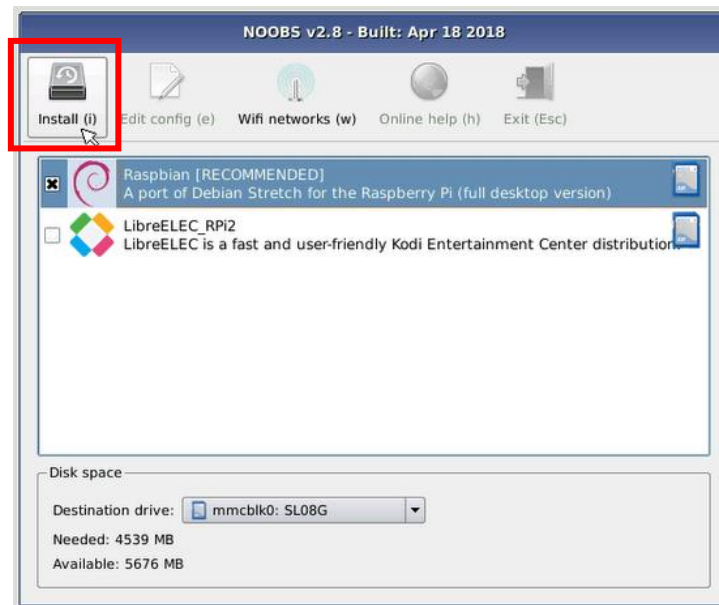


Figura 3.4.2.3 - Interfaz de NOOBS

Cuando haya finalizado la instalación, aparecerá el siguiente mensaje. Se pulsa sobre el botón OK.



Figura 3.4.2.4 - Instalación de Raspbian

Posteriormente, la Raspberry Pi se reiniciará automáticamente con el nuevo S.O. ya instalado y preparado para su utilización.

### 3.4.3 Instalación de VNC Viewer y configuración de dirección IP Estática.

Para poder acceder de manera remota a la Raspberry Pi desde un ordenador, sin necesidad de monitor y teclado, se utiliza el programa de escritorio remoto VNC Viewer.

En primer lugar, es necesario habilitar VNC en la Raspberry Pi antes de empezar a utilizarlo. Para realizar esta configuración, se accede a la ruta Menú → Preferencias → Raspberry Pi Configuración y se selecciona Enabled en el apartado VNC.

En esta misma configuración, se habilitará también el acceso al terminal vía SSH, para poder acceder a este en caso de que no fuera posible vía VNC, y se activará la cámara.

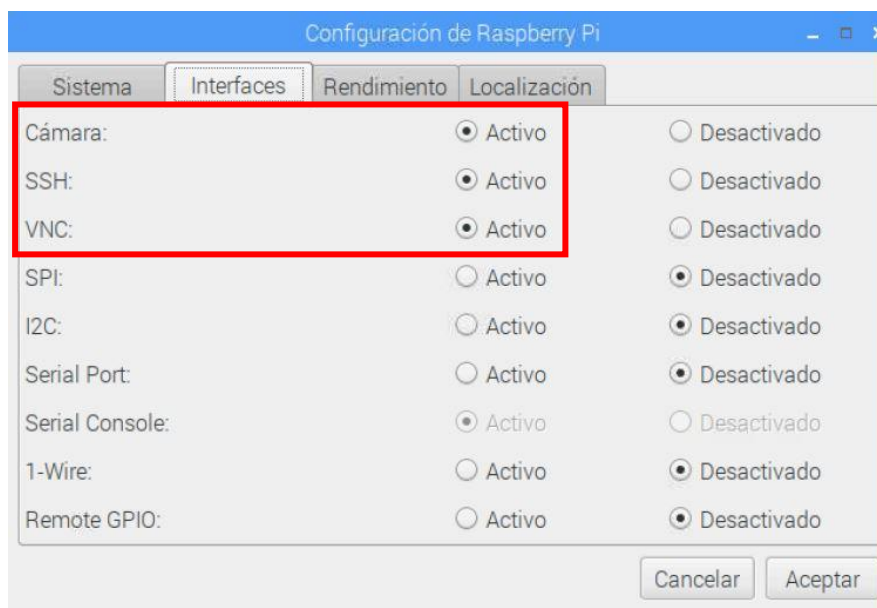
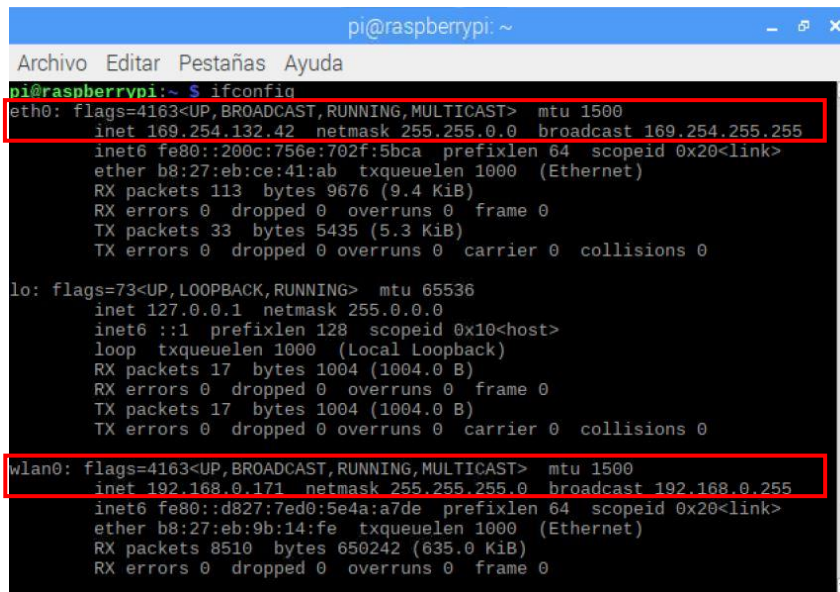


Figura 3.4.3.1 - Configuración de Raspberry Pi

A continuación, para poder acceder remotamente es necesario saber la dirección IP de la Raspberry Pi. Para ello, se abre el Terminal y se introduce el siguiente comando (ifconfig).



```
pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 169.254.132.42 netmask 255.255.0.0 broadcast 169.254.255.255
    inet6 fe80::200c:756e:702f:5bca prefixlen 64 scopeid 0x20<link>
    ether b8:27:eb:ce:41:ab txqueuelen 1000 (Ethernet)
    RX packets 113 bytes 9676 (9.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 33 bytes 5435 (5.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 17 bytes 1004 (1004.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 17 bytes 1004 (1004.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.171 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::d827:7ed0:5e4a:a7de prefixlen 64 scopeid 0x20<link>
    ether b8:27:eb:9b:14:fe txqueuelen 1000 (Ethernet)
    RX packets 8510 bytes 650242 (635.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
```

Figura 3.4.3.2 - Comando ifconfig

La conexión vía VNC puede hacerse desde una conexión Ethernet o WiFi, con lo cual se crearán dos perfiles en el programa VNC con las direcciones IP correspondientes a cada conexión. En este caso, las direcciones IP son las siguientes:

- Ethernet (eth0): 169.254.132.42
- WiFi (wlan0): 192.168.0.171

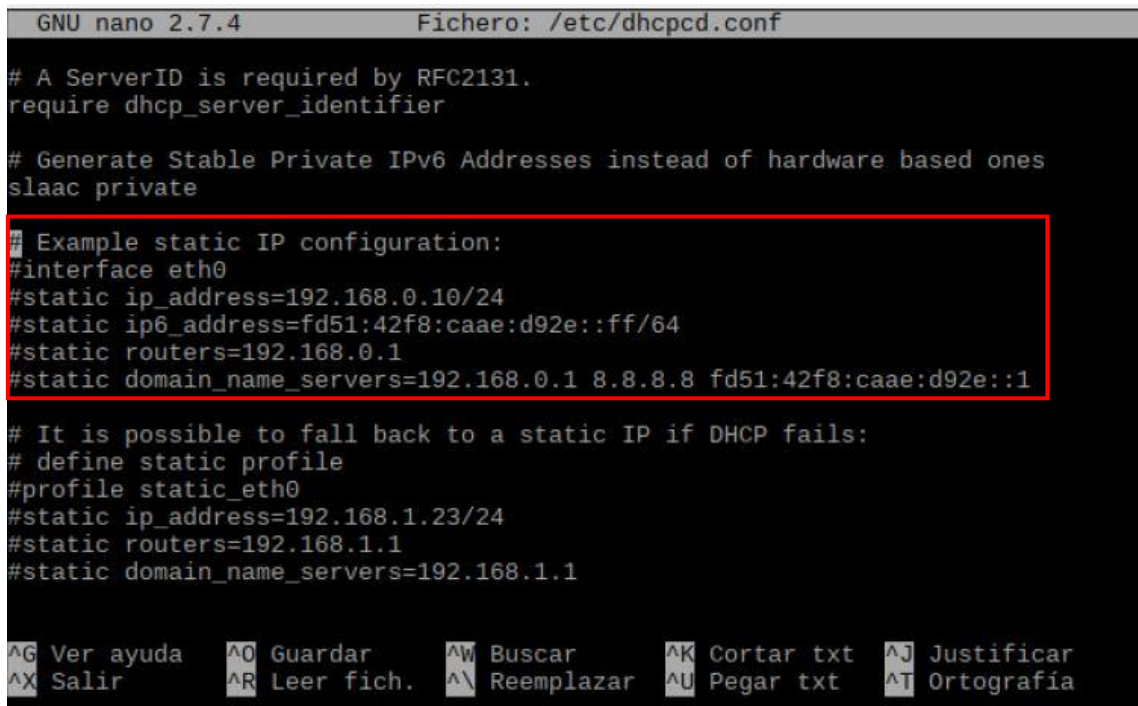
La dirección IP asignada a wlan0, se trata de una dirección IP dinámica, la cual puede cambiar cada vez que se reinicie el dispositivo. Para este proyecto, es de suma importancia que la dirección IP sea fija, para poder acceder de forma remota, mediante VNC o SSH, en cualquier momento, a las fotografías que almacena la Raspberry Pi.

Para configurar una dirección IP estática en Raspbian se ha realizado la siguiente configuración.

En primer lugar, se debe editar el fichero `/etc/dhcpd.conf` con el siguiente comando:

```
sudo nano /etc/dhcpd.conf
```

En el fichero aparecen unas líneas comentadas, con #, que contienen un ejemplo de configuración IP estática.



```
GNU nano 2.7.4 Fichero: /etc/dhcpd.conf
# A ServerID is required by RFC2131.
require dhcp_server_identifier

# Generate Stable Private IPv6 Addresses instead of hardware based ones
slaac private

# Example static IP configuration:
#interface eth0
#static ip_address=192.168.0.10/24
#static ip6_address=fd51:42f8:caae:d92e::ff/64
#static routers=192.168.0.1
#static domain_name_servers=192.168.0.1 8.8.8.8 fd51:42f8:caae:d92e::1

# It is possible to fall back to a static IP if DHCP fails:
# define static profile
#profile static_eth0
#static ip_address=192.168.1.23/24
#static routers=192.168.1.1
#static domain_name_servers=192.168.1.1

^G Ver ayuda  ^O Guardar    ^W Buscar     ^K Cortar txt ^J Justificar
^X Salir      ^R Leer fich. ^\ Reemplazar ^U Pegar txt  ^T Ortografía
```

Figura 3.4.3.3 - Configuración IP Estática I



Para crear la dirección IP estática deseada, se ha copiado el fragmento comentado y se ha modificado introduciendo los datos correspondientes a la red donde se encuentra el dispositivo, en donde:

- interface: Es el nombre de la interfaz que se quiere configurar.
- static ip\_address: Dirección IP fija que se desea.
- static routers: Dirección del gateway del router.
- static domain\_name\_servers: Dirección del servidor DND, normalmente la misma que la del router.

```
GNU nano 2.7.4          Fichero: /etc/dhcpd.conf          Modificado
#static domain_name_servers=192.168.0.1 8.8.8.8 fd51:42f8:caae:d92e::1
# It is possible to fall back to a static IP if DHCP fails:
# define static profile
#profile static_eth0
#static ip_address=192.168.1.23/24
#static routers=192.168.1.1
#static domain_name_servers=192.168.1.1
# fallback to static profile on eth0
#interface eth0
#fallback static_eth0
interface wlan0
static ip_address=192.168.0.161/24
static routers=192.168.0.1
static domain_name_servers=192.168.0.1

```

Ver ayuda   Guardar   Buscar   Cortar txt   Justificar  
Salir   Leer fich.   Reemplazar   Pegar txt   Ortografía

Figura 3.4.3.4 - Configuración IP Estática II

Para finalizar, se guardan los cambios con Ctrl+O, se cierra el fichero con Ctrl+X y después se reinicia la Raspberry con el comando.

```
sudo reboot
```

Continuando con la instalación de VNC Viewer, el siguiente paso es descargar el programa VNC Viewer y realizar su instalación en el ordenador. La descarga se realiza desde la siguiente página web.

<https://www.realvnc.com/es/connect/download/viewer/>

Una vez finalizada la descarga, se abre la aplicación y se crean dos nuevas conexiones, una con la dirección IP vía Ethernet y otra con la dirección IP vía WiFi.

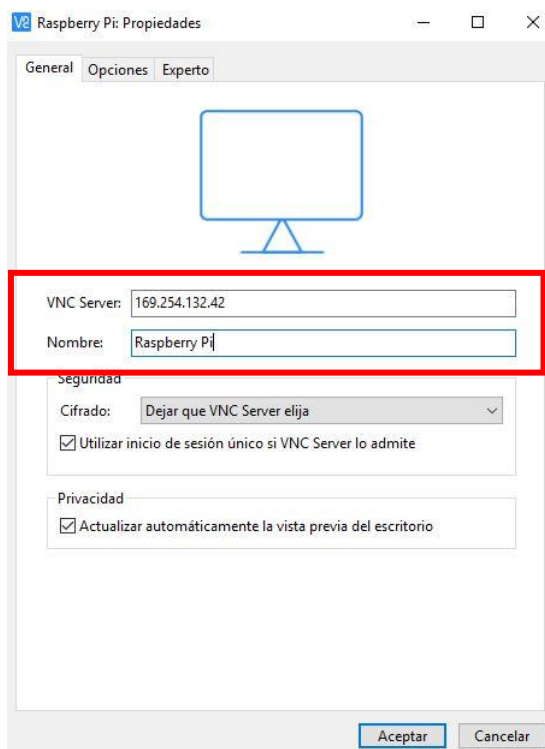


Figura 3.4.3.5 - Nueva conexión (Ethernet)

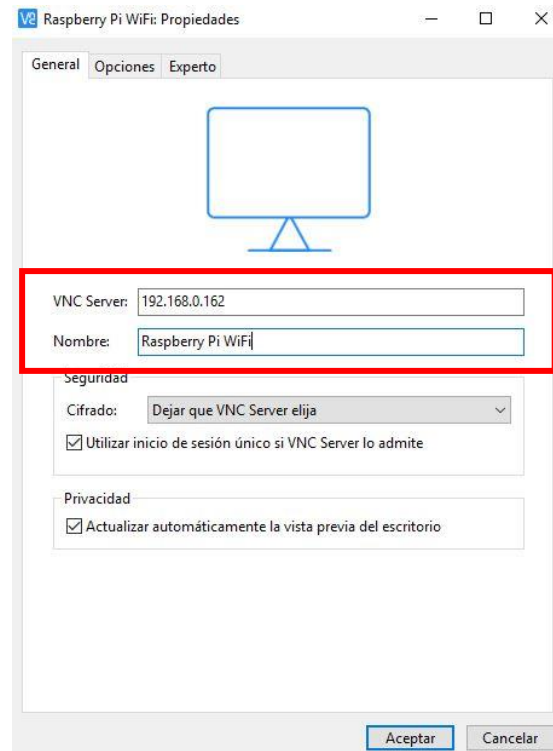


Figura 3.4.3.6 - Nueva conexión (WiFi)

Finalmente, se debe introducir una contraseña para el usuario pi, que por defecto es raspberry. Esta contraseña puede cambiarse en la configuración de Raspbian.

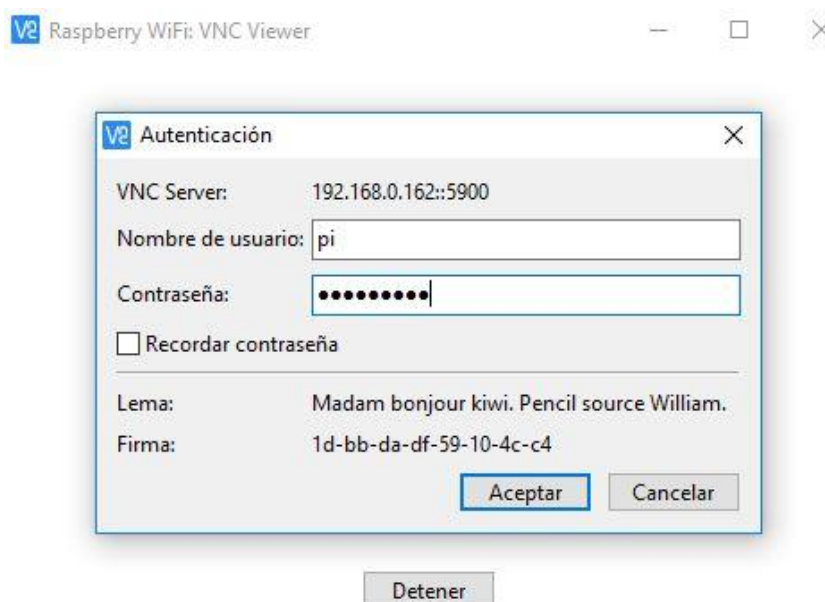


Figura 3.4.3.7 - Autenticación VNC

Una vez introducida la contraseña, aparecerá el escritorio de Raspbian en la pantalla del ordenador y se tendrá acceso a los programas instalados en la Raspberry, al menú de configuración, así como introducir comandos en el Terminal.



Figura 3.4.3.1 - Escritorio Raspbian VNC

### 3.4.4 Ejecución automática del script de Python.

Debido a que el script de Python no se ejecuta de forma automática por defecto al arrancar el equipo time-lapse, se ha realizado una configuración desde el Terminal de la Raspberry Pi para que se ejecute de dicha manera.

En primer lugar, se ejecuta el siguiente comando en el Terminal, el cual crea un documento en la ruta /etc/init.d/ llamado timelapse-init.

```
sudo nano /etc/init.d/timelapse-init
```

A continuación, se copia el siguiente contenido en el archivo.

```
#!/bin/sh
# /etc/init.d/timelapse-init

### BEGIN INIT INFO
# Provides:          timelapse-init
# Required-Start:    $all
# Required-Stop:     $remote_fs $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Script de ejemplo de arranque
automático
# Description:       Script para arrancar el timelapse
### END INIT INFO

case "$1" in
  start)
    echo "Arrancando timelapse-init"
    # Aquí hay que poner el programa que se quiera arrancar
    automáticamente
    /usr/bin/python /home/pi/timelapse.py
    ;;
  stop)
```

```
    echo "Deteniendo timelapse-init"

    ;;

*)
    echo "Modo de uso: /etc/init.d/timelapse-init
{start|stop}"
    exit 1
    ;;
esac

exit 0
```

Una vez hecho esto, se guarda con Ctrl+O y se sale con Ctrl+X.

Para hacer el script ejecutable, se introduce el siguiente comando en el Terminal, el cual da permisos de lectura y ejecución a todos los usuarios, excepto al usuario propietario que tiene permisos de lectura, escritura y ejecución.

```
sudo chmod 755 /etc/init.d/timelapse-init
```

Por último, se activa el arranque automático con el siguiente comando.

```
sudo update-rc.d timelapse-init defaults
```

### 3.4.5 Instalación de FFmpeg en Raspbian.

Para realizar la instalación de FFmpeg en la Raspberry Pi, se han seguido los siguientes pasos.

En primer lugar, se instala el soporte para H.264 o MPEG-4 con los siguientes comandos.

```
# Se realiza la descarga, se compila y se instala Libx264.
```

```
cd /usr/src
```

```
sudo git clone git://git.videolan.org/x264
```

```
cd x264
```

```
sudo ./configure --host=arm-unknown-linux-gnueabi --enable-static --disable-openc1
```

```
sudo make
```

```
sudo make install
```

Una vez ejecutados los comandos anteriores, se procede a instalar FFmpeg a través de los siguientes comandos.

```
# Se realiza la descarga, se compila y se instala FFmpeg.
```

```
cd /usr/src
```

```
sudo git clone https://github.com/FFmpeg/FFmpeg.git
```

```
cd FFmpeg
```

```
sudo ./configure --arch=armel --target-os=linux --enable-gpl --enable-libx264 --enable-nonfree
```

```
sudo make
```

```
sudo make install
```

Al finalizar la instalación se ha de reiniciar la Raspberry Pi con el comando

```
sudo reboot
```

### 3.5 Configuración del menú de usuario.

En este apartado se realiza un resumen de la configuración del menú de usuario el cual se muestra en las imágenes a continuación.

#### 3.5.1 Pantalla inicial.

La siguiente imagen muestra la pantalla inicial que aparece al conectar el equipo.



Figura 3.5.1.1 - Menú: Pantalla inicial

#### 3.5.2 Pantalla para modos de funcionamiento.

En la imagen a continuación, se muestra la pantalla que aparece después de la pantalla inicial.

Pulsando la tecla 1, se escoge el modo de funcionamiento manual, que ha sido explicado anteriormente.

Pulsando la tecla 2, se escoge el modo automático.



**Figura 3.5.2.1 - Menú: Modos de funcionamiento**

### 3.5.3 Primera pantalla para modo manual: Posición eje X.

En la siguiente imagen se muestra la primera pantalla del modo de funcionamiento manual, en la que se introduce el valor, en grados, de la posición de la cámara horizontal.



**Figura 3.5.3.1 - Menú – Posición servo horizontal**



Si el usuario introdujera un valor que no estuviera entre  $0^{\circ}$  y  $180^{\circ}$ , este sería borrado automáticamente y aparecería este mensaje de error:



Figura 3.5.3.2 - Menú: Error posición servo horizontal

Una vez introducido, se pulsa en la tecla '#' para continuar. Si se introdujera un valor equivocado, el usuario podría corregirlo pulsado la tecla '\*'.



Figura 3.5.3.3 - Menú: Posición servo horizontal, dato introducido

### 3.5.4 Segunda pantalla para modo manual: Posición eje Y.

La configuración de la siguiente pantalla es igual a la anterior, con la diferencia que el valor introducido es el de la posición de la cámara vertical.



Figura 3.5.4.1 - Menú: Posición servo vertical

Se mantiene la restricción del valor entre  $0^{\circ}$  y  $180^{\circ}$ .



Figura 3.5.4.2 - Menú: Error posición servo vertical

Igual que en la pantalla anterior, una vez introducido, se pulsa en la tecla '#' para continuar. Si se introdujera un valor equivocado, el usuario podría corregirlo pulsando la tecla '\*'.

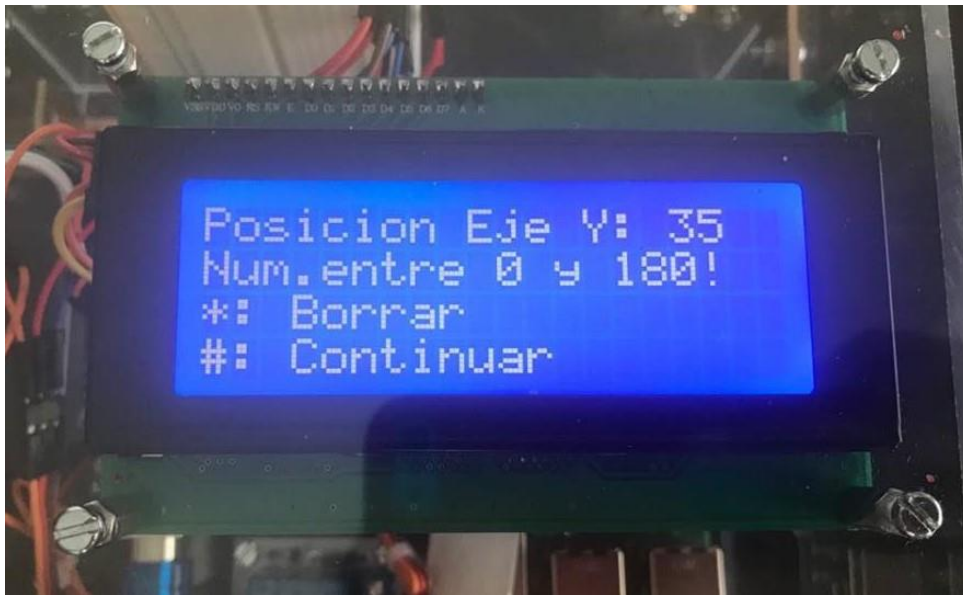


Figura 3.5.4.3 - Menú: Posición servo vertical, dato introducido

### 3.5.5 Tercera pantalla para modo manual: Intervalo entre capturas.

En esta pantalla se introduce el valor, en segundos, del tiempo entre capturas de las fotografías que conforman el vídeo time-lapse.

Igual que en las pantallas anteriores, una vez introducido, se pulsa en la tecla '#' para continuar. Si se introdujera un valor equivocado, el usuario podría corregirlo pulsando la tecla '\*'.

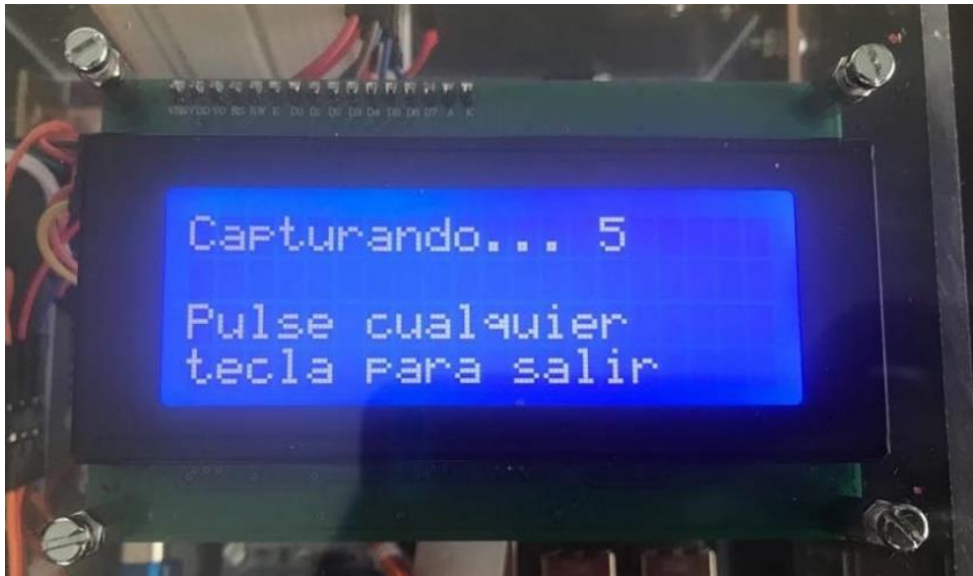


Figura 3.5.5.1 - Menú: Intervalo entre fotografías

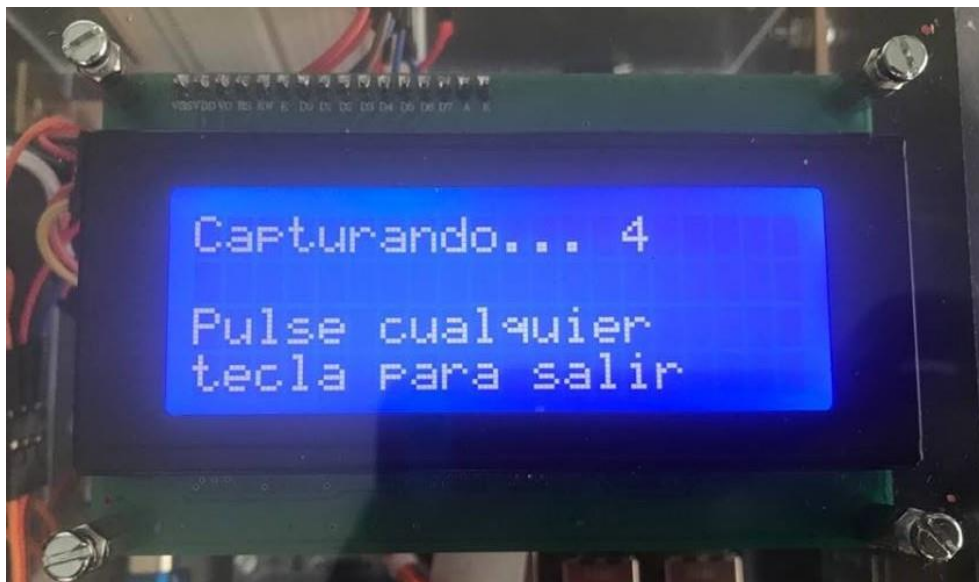
### 3.5.6 Pantalla durante captura de fotografías.

Por último, se muestra la pantalla que aparece mientras el sistema realiza las fotografías. Esta pantalla es común para los modos de funcionamiento manual y automático.

En ella se puede ver una cuenta regresiva del intervalo entre fotografías. A continuación, se muestran una serie de imágenes en la que se aprecia esta cuenta regresiva, con un intervalo entre capturas de 5 segundos.



**Figura 3.5.6.1 - Menú: Cuenta regresiva I**



**Figura 3.5.6.2 - Menú: Cuenta regresiva II**



Figura 3.5.6.3 - Menú: Cuenta regresiva III

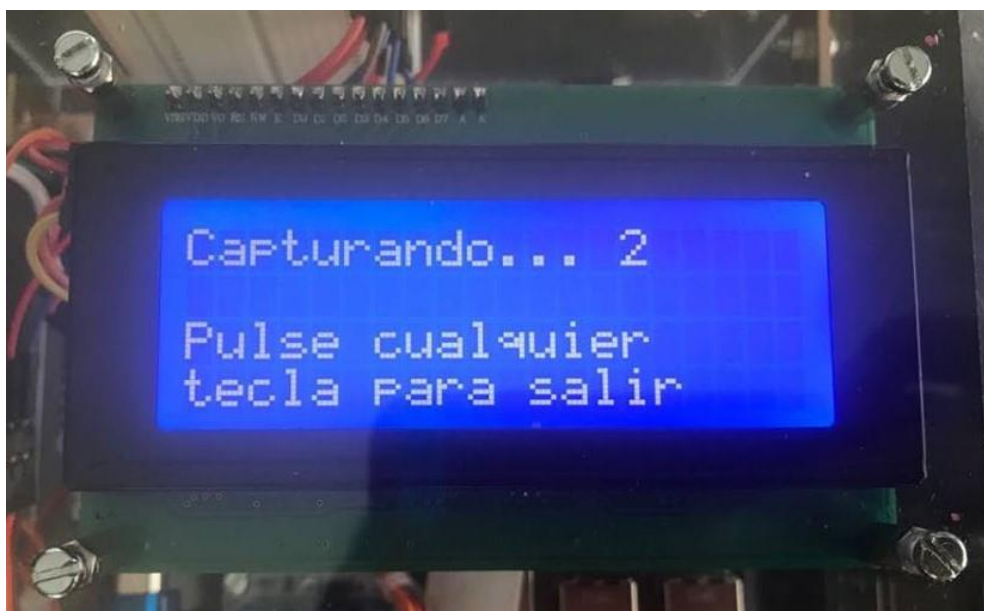


Figura 3.5.6.4 - Menú: Cuenta regresiva IV



**Figura 3.5.6.5 - Menú: Cuenta regresiva V**

### 3.6 Montaje de la caja.

Para la realización de la caja contenedora se ha partido de dos planchas de metacrilato de 50x25x4 mm, cortadas posteriormente en seis partes.

Una vez cortadas las planchas, se han seguido los siguientes pasos para realizar el montaje de la caja.

#### 3.6.1 Paso 1: Montaje de los componentes electrónicos.

El primer paso ha sido el de realizar el ensamblado de los componentes electrónicos en la base de la caja.

Para ello, se ha procedido a taladrar la plancha, realizando los agujeros necesarios para unir todos los componentes electrónicos a la base. La unión se ha hecho mediante separadores metálicos, en orden de disipar mejor el calor que desprenden los componentes, y diferente tornillería.

En este paso también se han realizado los agujeros necesarios para atornillar las escuadras metálicas, que sirven como unión entre las diferentes partes de la caja.



Figura 3.6.1.1 - Taladrado de la base



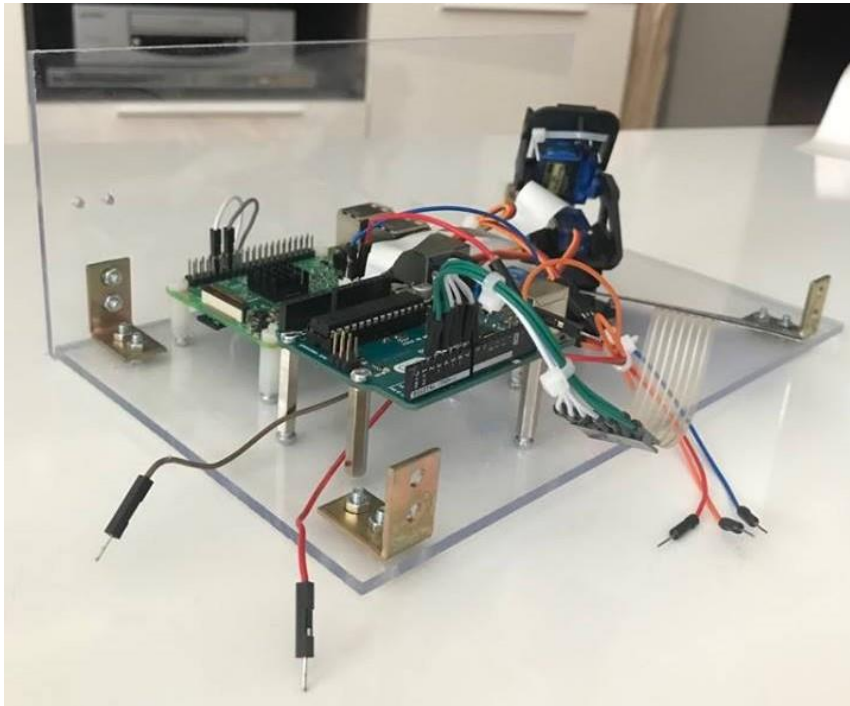


Figura 3.6.1.2 - Ensamblado de los componentes a la base

### 3.6.2 Paso 2: Unión de las partes de la caja.

El siguiente paso, es el de unir las diferentes partes de la caja entre ellas y a la base de esta. Esta unión se ha realizado mediante diferente tornillería y escuadras angulares metálicas.

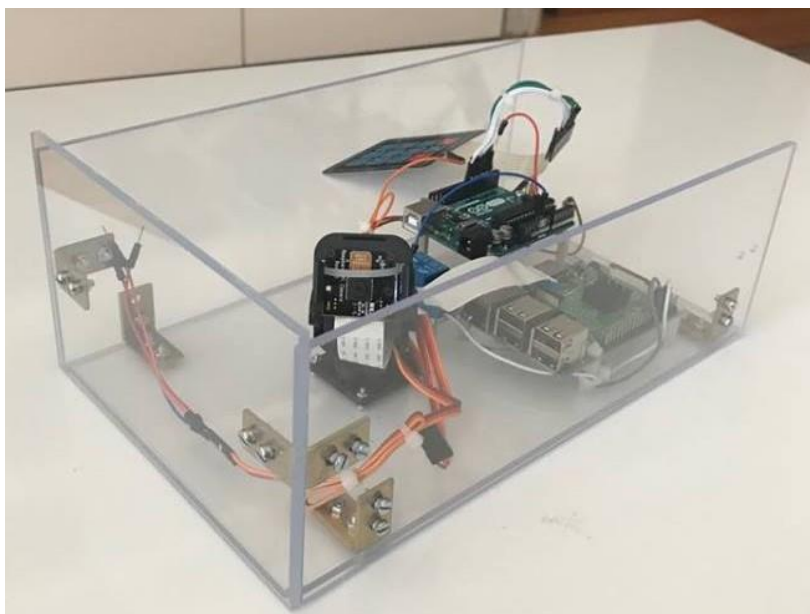
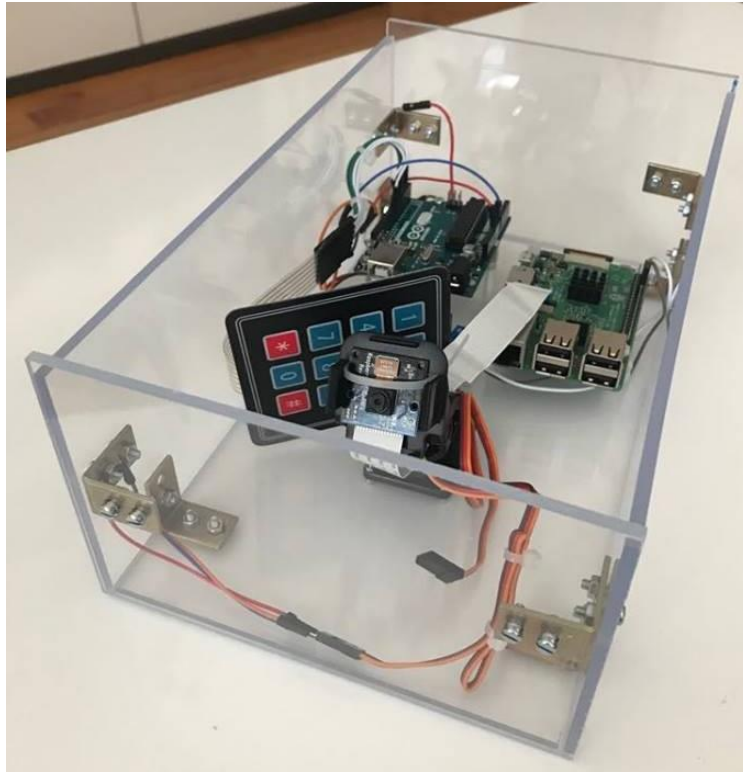


Figura 3.6.2.1 - Unión de las partes de la caja I



**Figura 3.6.2.2 - Unión de las partes de la caja II**

### **3.6.3 Paso 3: Montaje de la tapa.**

En este último paso, se han realizado los agujeros necesarios con el taladro para unir la pantalla LCD con la tapa de la caja y para pasar el cable del teclado matricial desde el interior al exterior de esta.

También se han colocado dos bisagras metálicas, para unir la tapa con el resto de la caja y para facilitar su apertura.



**Figura 3.6.3.1 - Montaje de la tapa**

### 3.7 Montaje del vídeo time-lapse.

Una vez obtenidas las fotografías, la realización del vídeo resulta muy sencilla. Para obtener el vídeo, es suficiente con ejecutar el siguiente comando desde el Terminal de la Raspberry Pi, configurando los parámetros a conveniencia del usuario.

```
ffmpeg -r 25 -pattern_type glob  
-i "/home/pi/timelapse/*.jpg" -s hd1080 -vcodec libx264  
timelapse.mp4
```

Donde:

- r: Fotogramas por segundo (fps). En este caso se configura con 25 fps.
- pattern type glob -i "/home/pi/timelapse/\*.jpg". Lista todas las imágenes en formato .JPG en el directorio actual.
- s hd1080. Resolución del vídeo.
- vcodec libx264. Códec que se desea utilizar. En este caso se trata del H.264 o MPEG-4.

En la parte final de la línea se ha de poner el nombre del vídeo creado con el formato que se desee.

### **3.8 Posibles mejoras para realizar.**

En este apartado se recopilan posibles mejoras a llevar a cabo en caso de realizar otro proyecto que complemente al actual.

#### **3.8.1 Subida de las fotografías a la nube.**

En el proyecto actual, las fotografías se almacenan en la memoria de la Raspberry Pi. Una posible mejora sería la de liberar memoria de la tarjeta, subiendo las fotografías a un servicio en la nube, como por ejemplo Dropbox. También existen otras alternativas a la solución de Dropbox, de software libre, como por ejemplo ownCloud y Nextcloud.

#### **3.8.2 Mejora de la caja contenedora.**

Otra posible mejora que realizar sería la de cambiar la caja actual, la cual se ha diseñado principalmente con el objetivo de hacer el prototipo más atractivo visualmente para el cliente, por otra con protección IP65. Este grado de protección, contra polvo y agua, sería el más adecuado para un equipo que debe estar situado en el exterior, como es el caso de una obra.

**TÍTULO: DISEÑO DE UN EQUIPO TIME-LAPSE PARA  
SEGUIMIENTO DE OBRAS**

---

## **PLANOS**

---

**PETICIONARIO: ESCUELA UNIVERSITARIA POLITÉCNICA  
AVDA. 19 DE FEBRERO, S/N  
15405 - FERROL**

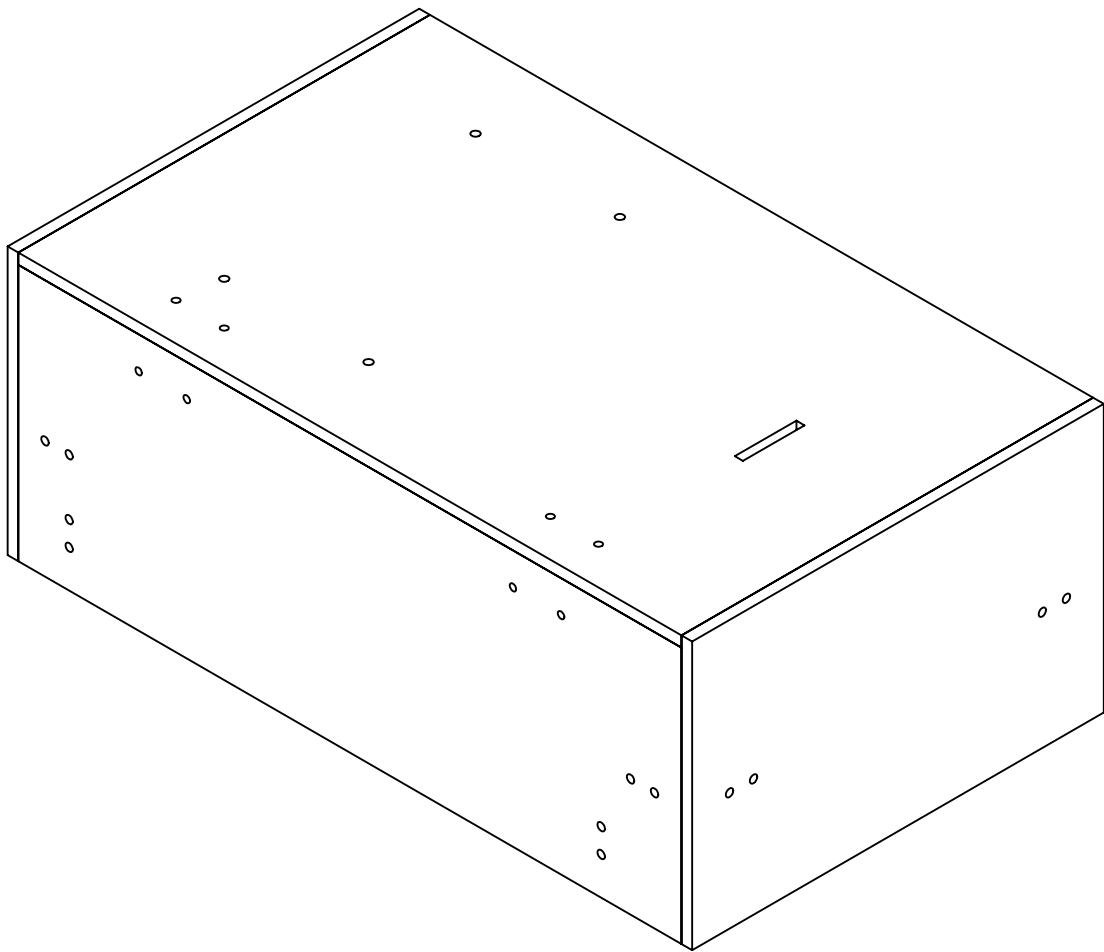
**FECHA: SEPTIEMBRE DE 2018**

**AUTOR: EL ALUMNO**

**Fdo.: YOLANDA PICALLO CAAMAÑO**

## INDICE PLANOS

	Páginas
4 PLANOS .....	104
4.1 Ensamblaje de la caja. ....	104
4.2 Partes frontal y trasera de la caja.....	105
4.3 Tapa de la caja.....	106
4.4 Lateral izquierdo de la caja. ....	107
4.5 Lateral derecho de la caja. ....	108
4.6 Base de la caja (Parte 1 de 3).....	109
4.7 Base de la caja (Parte 2 de 3).....	110
4.8 Base de la caja (Parte 3 de 3).....	111
4.9 Esquemas del hardware de Raspberry Pi.....	112



UNIVERSIDADE DA CORUÑA ESCUELA UNIVERSITARIA POLITÉCNICA

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TFG Nº: 770G01A138

TÍTULO DEL TFG:

DISEÑO DE UN EQUIPO TIME-LAPSE PARA SEGUIMIENTO DE OBRAS

TÍTULO DEL PLANO:

ENSAMBLAJE DE LA CAJA

FECHA: SEPTIEMBRE-2018

ESCALA: 1:2

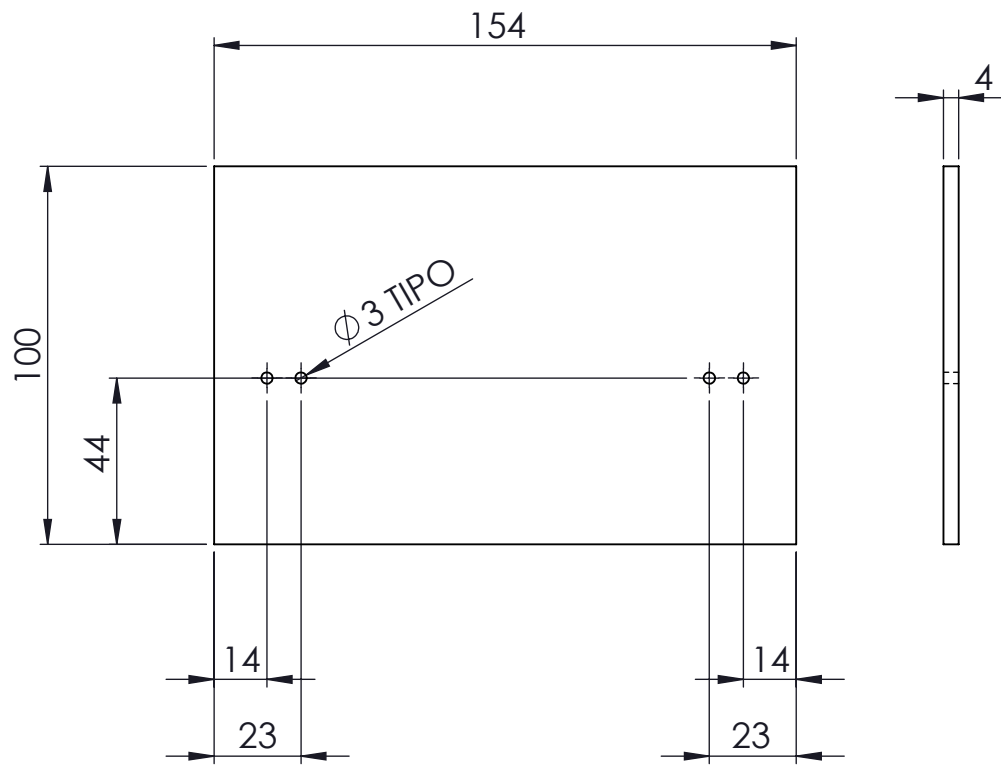
AUTOR:


YOLANDA PICALLO CAAMAÑO

FIRMA:

PLANO Nº:01






**UNIVERSIDADE DA CORUÑA** ESCUELA UNIVERSITARIA POLITÉCNICA  
 GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TFG Nº: 770G01A138

TÍTULO DEL TFG:  
**DISEÑO DE UN EQUIPO TIME-LAPSE PARA SEGUIMIENTO DE OBRAS**

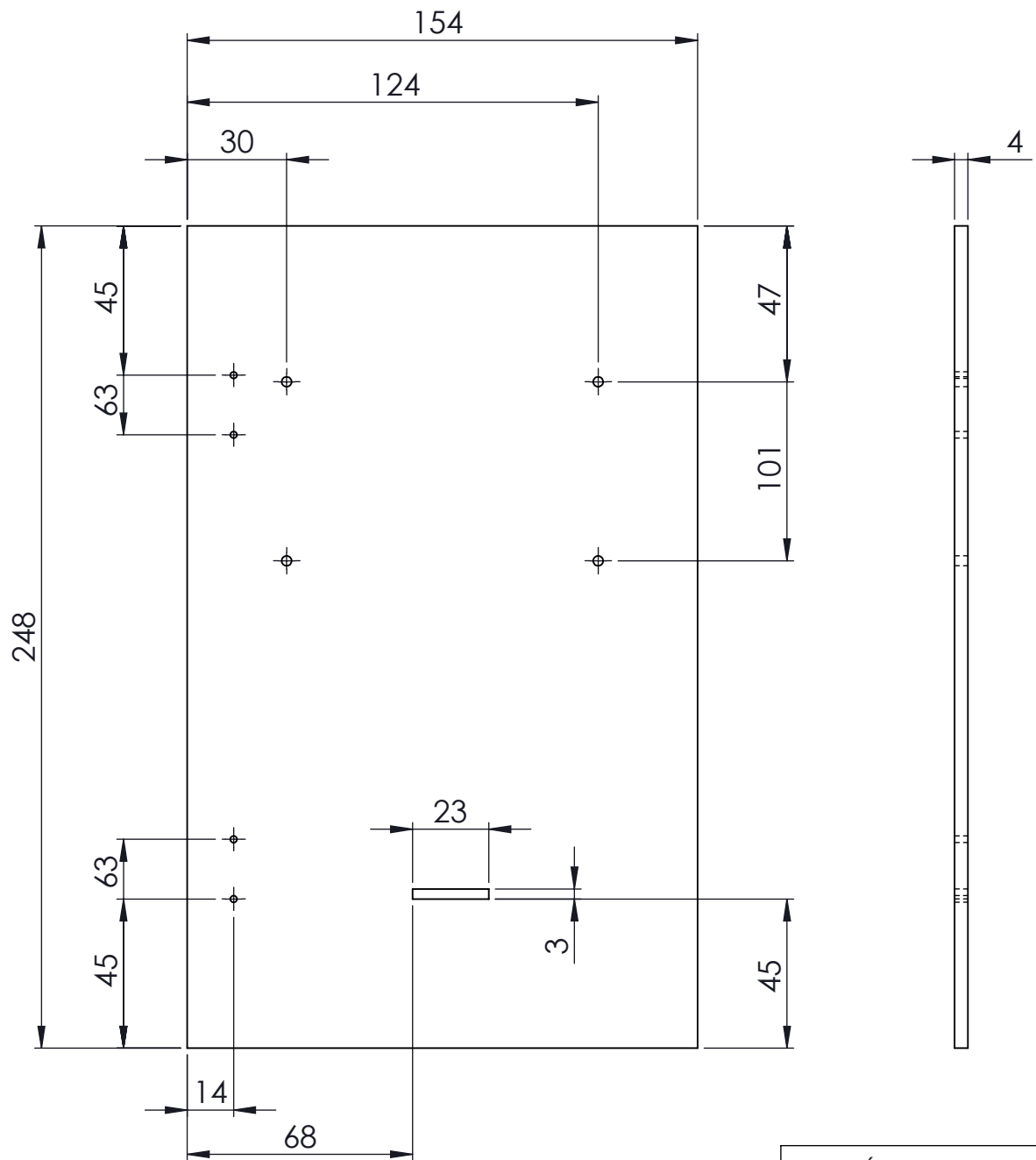
TÍTULO DEL PLANO:  
**PARTES FRONTAL Y TRASERA DE LA CAJA**

FECHA: SEPTIEMBRE-2018  
 ESCALA: 1:2

AUTOR:  
 YOLANDA PICALLO CAAMAÑO

FIRMA:

**PLANO Nº:02**



DIÁMETROS DE LOS AGUJEROS	
BISAGRAS	Ø2 mm
RESTO DE AGUJEROS	Ø3 mm


**UNIVERSIDADE DA CORUÑA** ESCUELA UNIVERSITARIA POLITÉCNICA  
 GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TFG Nº: 770G01A138

TÍTULO DEL TFG:  
**DISEÑO DE UN EQUIPO TIME-LAPSE PARA SEGUIMIENTO DE OBRAS**

TÍTULO DEL PLANO:  
**TAPA DE LA CAJA**

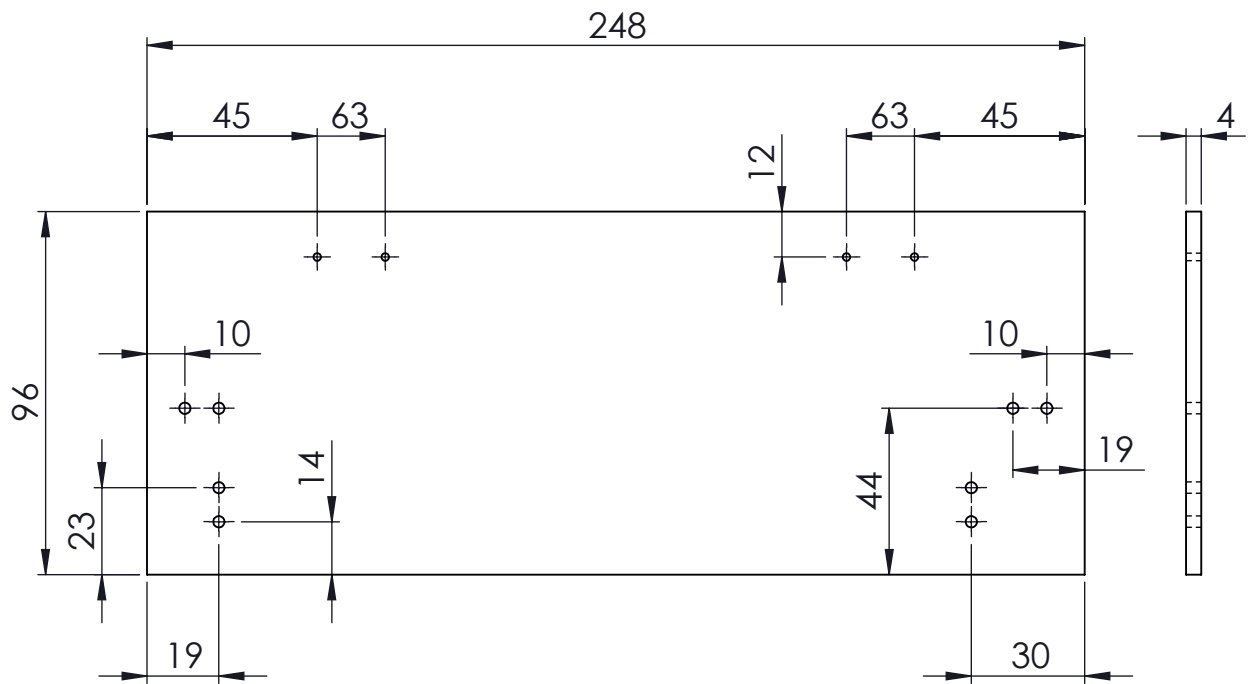
FECHA: SEPTIEMBRE-2018

ESCALA: 1:2

AUTOR:  
 YOLANDA PICALLO CAAMAÑO

FIRMA:

PLANO Nº:03



DIÁMETROS DE LOS AGUJEROS	
BISAGRAS	Ø2 mm
RESTO DE AGUJEROS	Ø3 mm


**UNIVERSIDADE DA CORUÑA** ESCUELA UNIVERSITARIA POLITÉCNICA  
 GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TFG Nº: 770G01A138

TÍTULO DEL TFG:

DISEÑO DE UN EQUIPO TIME-LAPSE PARA SEGUIMIENTO DE OBRAS

TÍTULO DEL PLANO:

LATERAL IZQUIERDO DE LA CAJA

FECHA: SEPTIEMBRE-2018

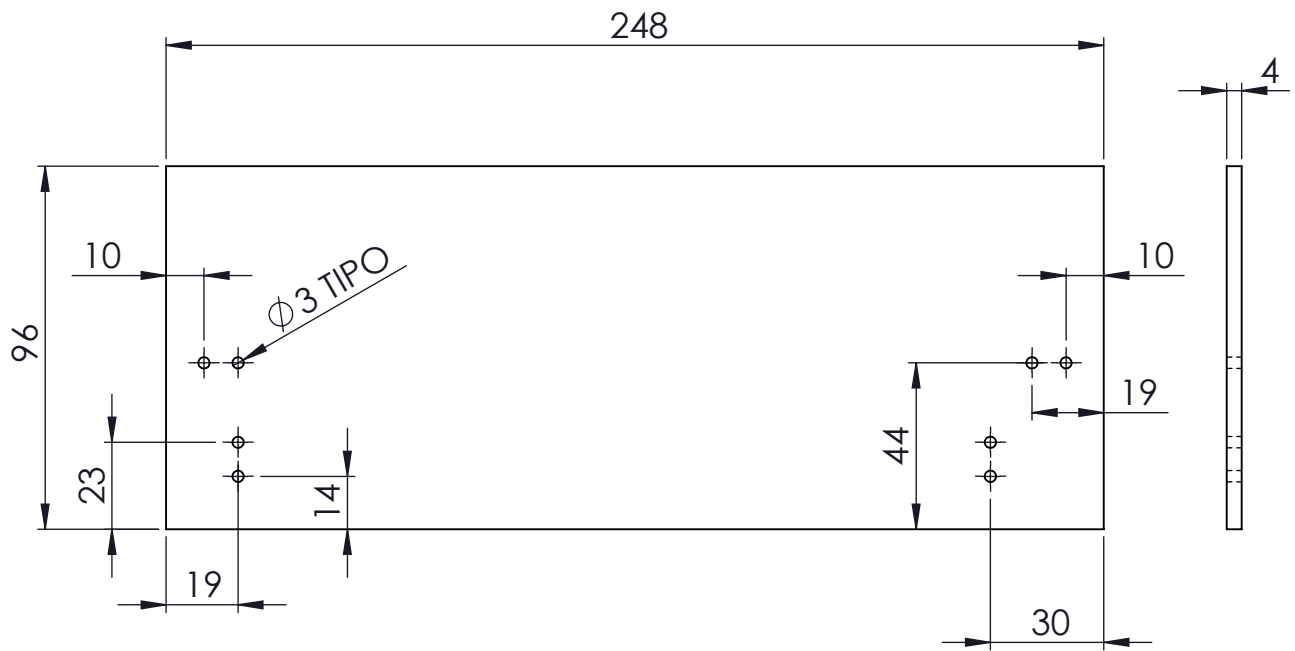
ESCALA: 1:2

AUTOR:

YOLANDA PICALLO CAAMAÑO

FIRMA:

PLANO Nº:04



UNIVERSIDADE DA CORUÑA ESCUELA UNIVERSITARIA POLITÉCNICA

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TFG Nº: 770G01A138

TÍTULO DEL TFG:

DISEÑO DE UN EQUIPO TIME-LAPSE PARA SEGUIMIENTO DE OBRAS

TÍTULO DEL PLANO:

LATERAL DERECHO DE LA CAJA

FECHA: SEPTIEMBRE-2018

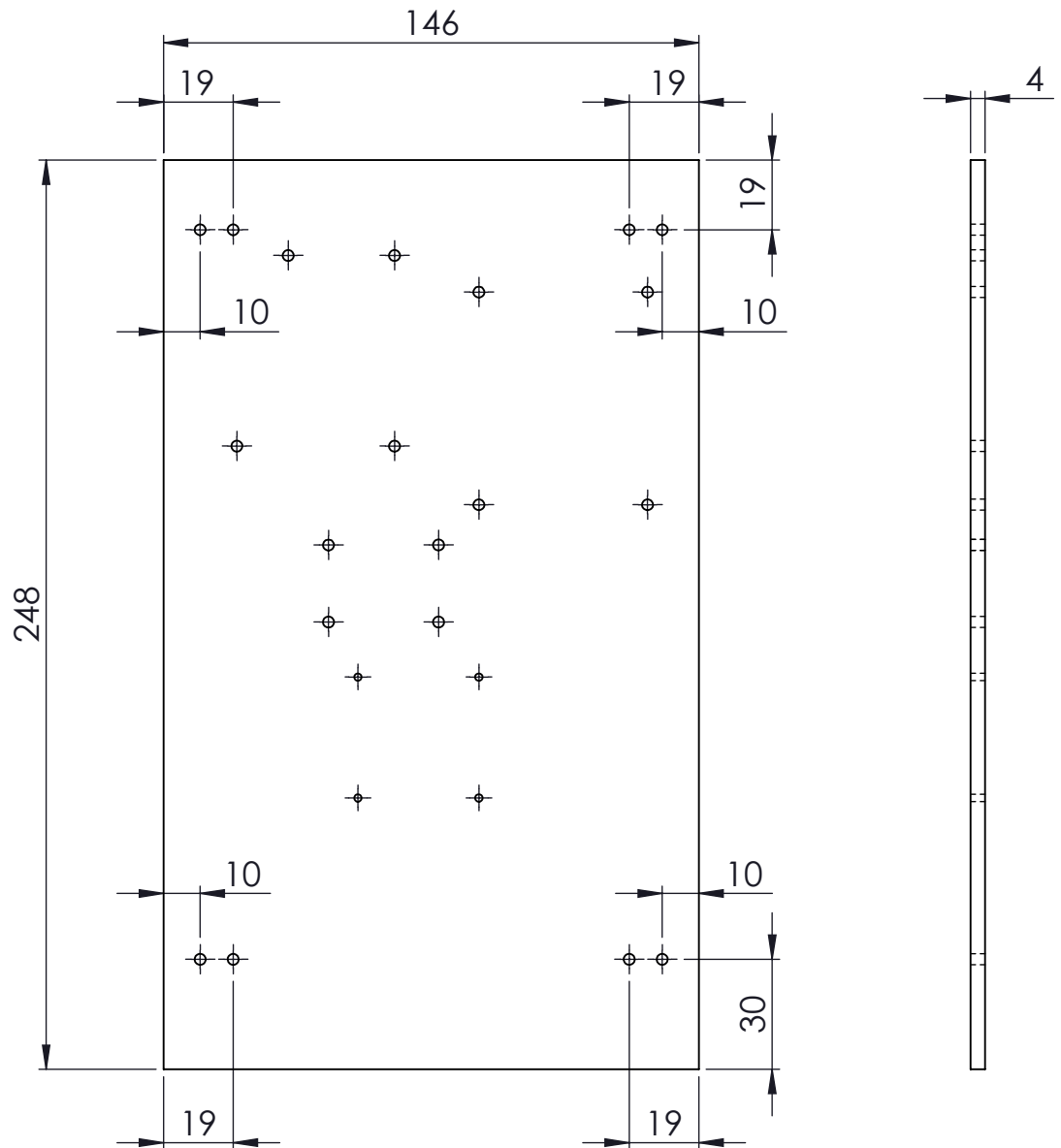
ESCALA: 1:2

AUTOR:

YOLANDA PICALLO CAAMAÑO

FIRMA:

PLANO Nº:05



UNIVERSIDADE DA CORUÑA ESCUELA UNIVERSITARIA POLITÉCNICA

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TFG Nº: 770G01A138

TÍTULO DEL TFG:

DISEÑO DE UN EQUIPO TIME-LAPSE PARA SEGUIMIENTO DE OBRAS

TÍTULO DEL PLANO:

BASE DE LA CAJA (PARTE 1 DE 3)

FECHA: SEPTIEMBRE-2018

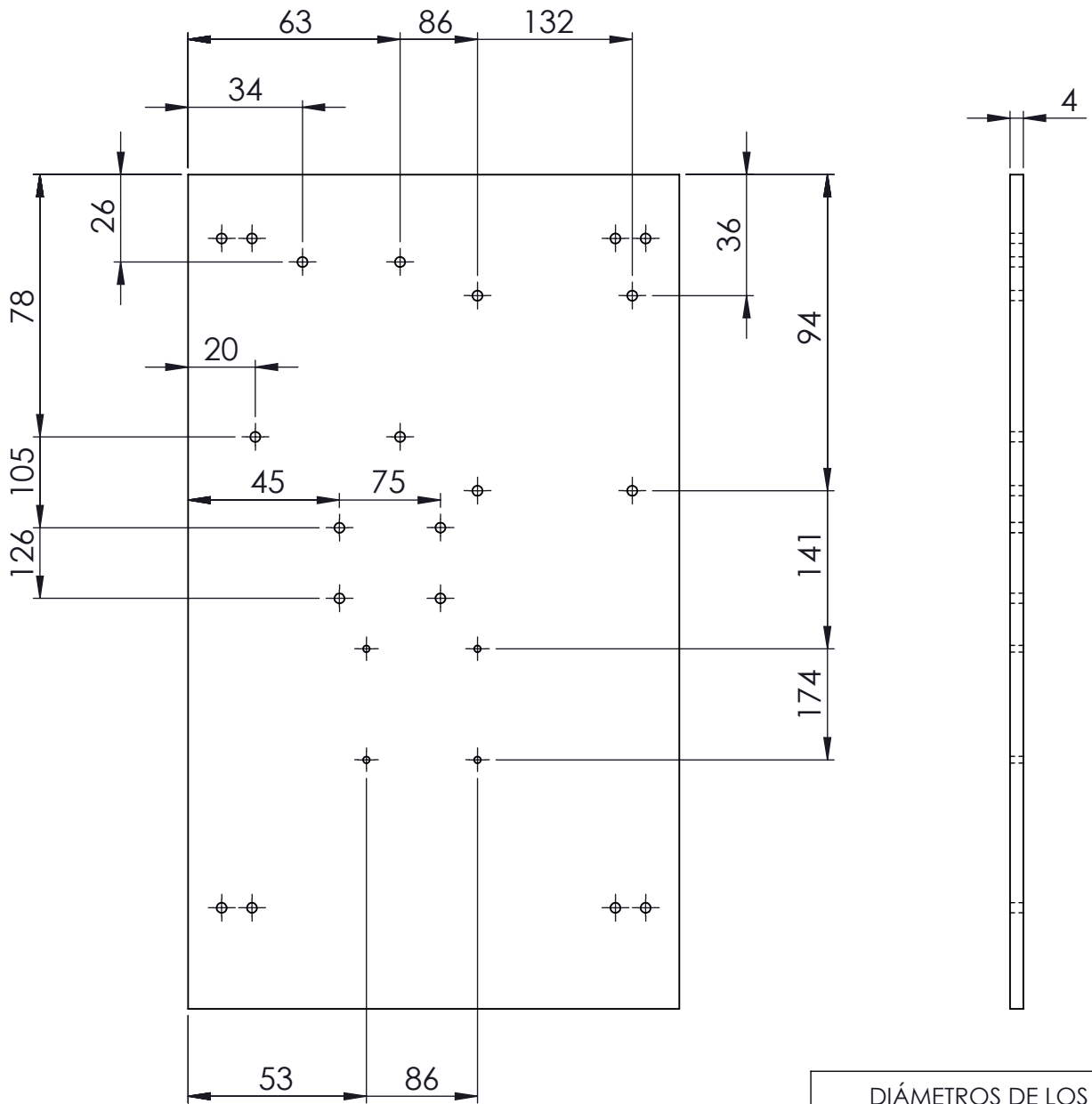
ESCALA: 1:2

AUTOR:

YOLANDA PICALLO CAAMAÑO

FIRMA:

PLANO Nº:06



DIÁMETROS DE LOS AGUJEROS	
SOPORTE DE LA CÁMARA	Ø2 mm
RESTO DE AGUJEROS	Ø3 mm


**UNIVERSIDADE DA CORUÑA** ESCUELA UNIVERSITARIA POLITÉCNICA  
 GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TFG Nº: 770G01A138

TÍTULO DEL TFG:

DISEÑO DE UN EQUIPO TIME-LAPSE PARA SEGUIMIENTO DE OBRAS

TÍTULO DEL PLANO:

BASE DE LA CAJA (PARTE 2 DE 3)

FECHA: SEPTIEMBRE-2018

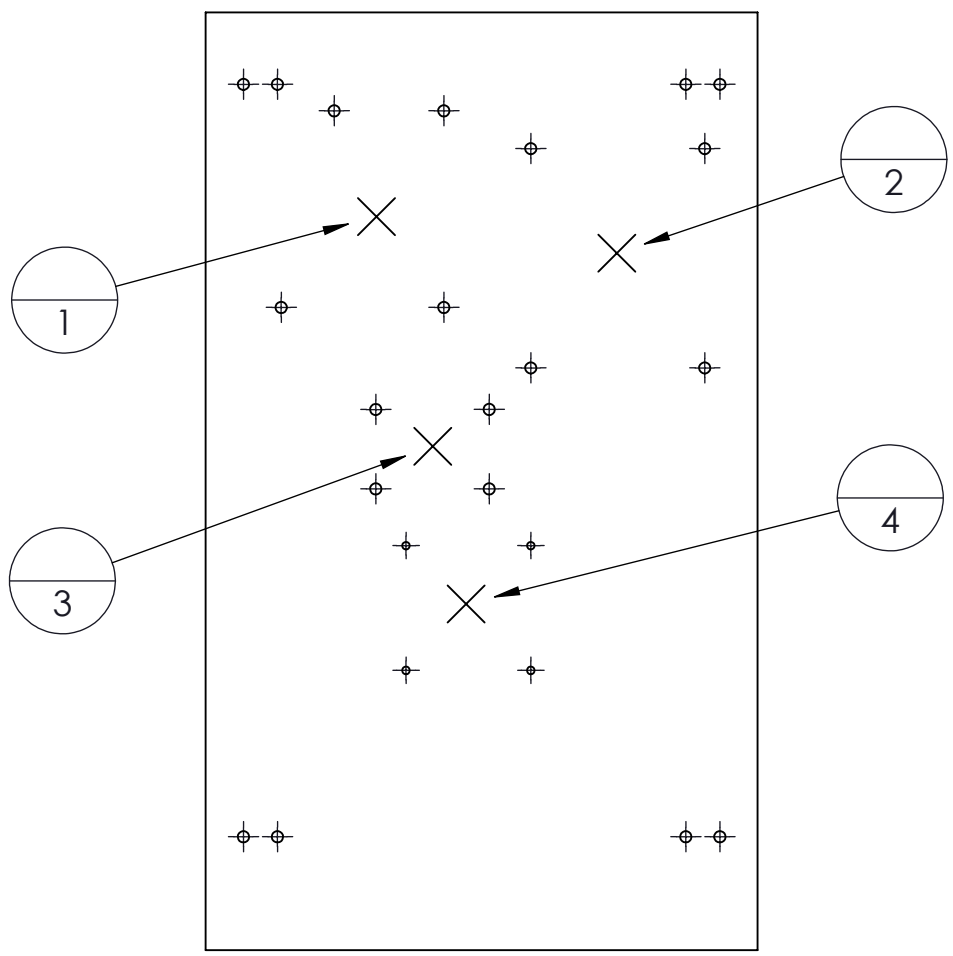
ESCALA: 1:2

AUTOR:


YOLANDA PICALLO CAAMAÑO

FIRMA:

PLANO Nº:07



DESIGNACIÓN DE COMPONENTES	
1	ARDUINO UNO
2	RASPBERRY PI 3
3	RELÉ 5V
4	SOPORTE CÁMARA


**UNIVERSIDADE DA CORUÑA** ESCUELA UNIVERSITARIA POLITÉCNICA  
 GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TFG Nº: 770G01A138

TÍTULO DEL TFG:  
**DISEÑO DE UN EQUIPO TIME-LAPSE PARA SEGUIMIENTO DE OBRAS**

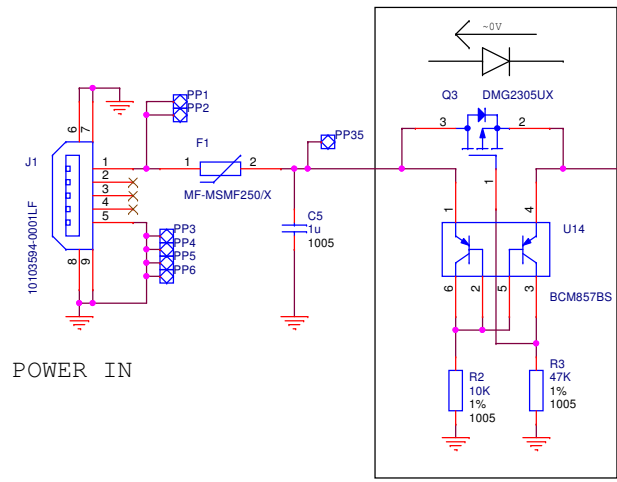
TÍTULO DEL PLANO:  
**BASE DE LA CAJA (PARTE 3 DE 3)**

FECHA: SEPTIEMBRE-2018  
 ESCALA: 1:2

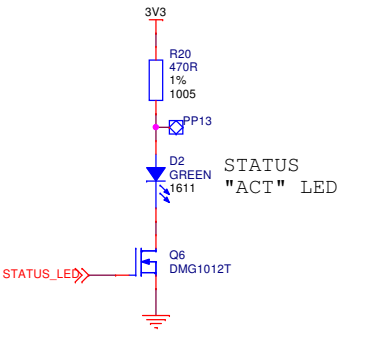
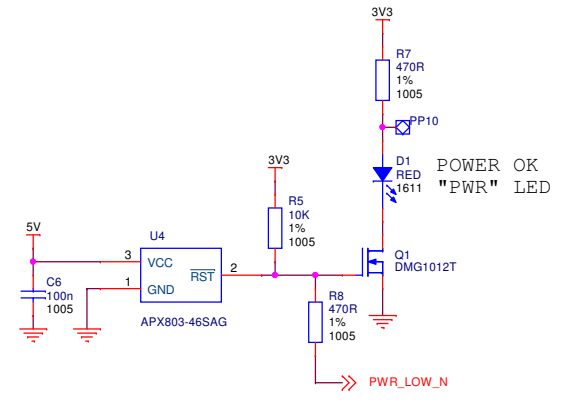
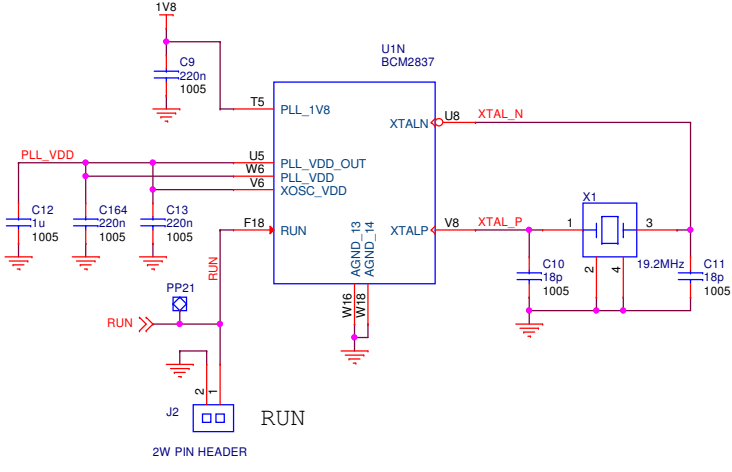
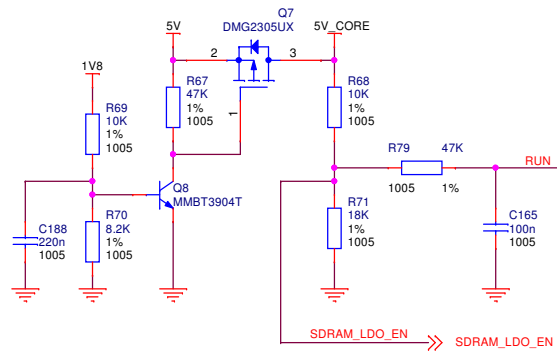
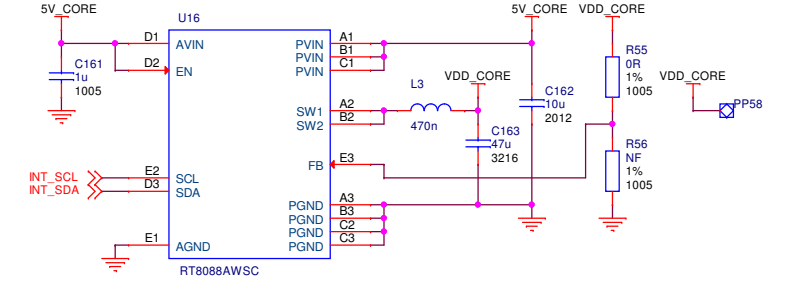
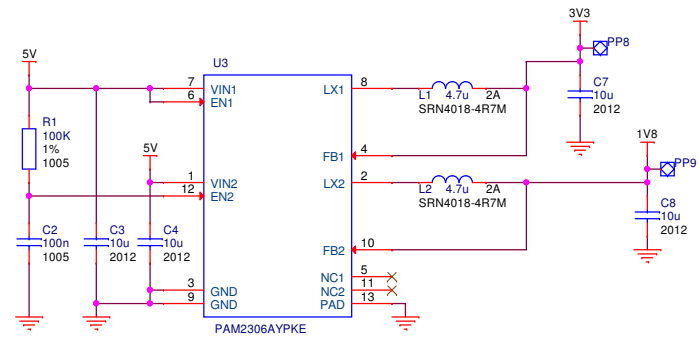
AUTOR:  
 YOLANDA PICALLO CAAMAÑO

FIRMA:

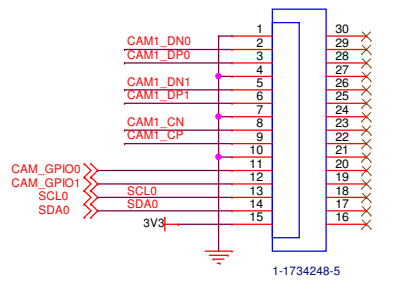
**PLANO Nº:08**



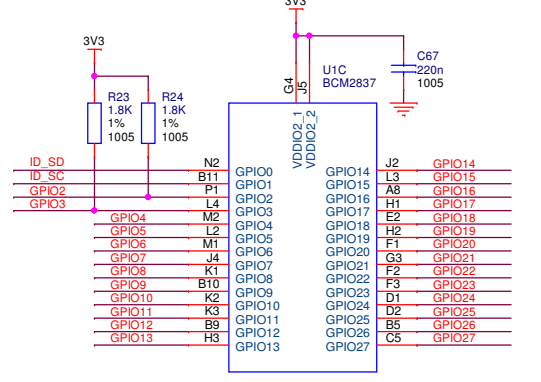
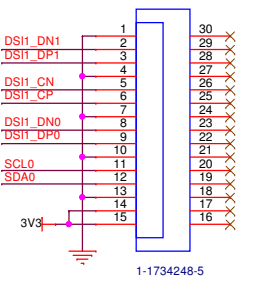
POWER IN



CAMERA

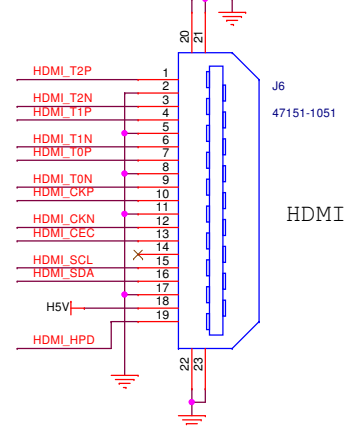
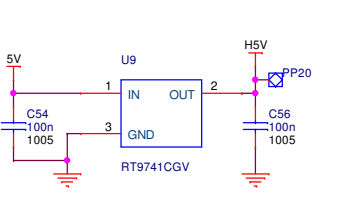


DISPLAY

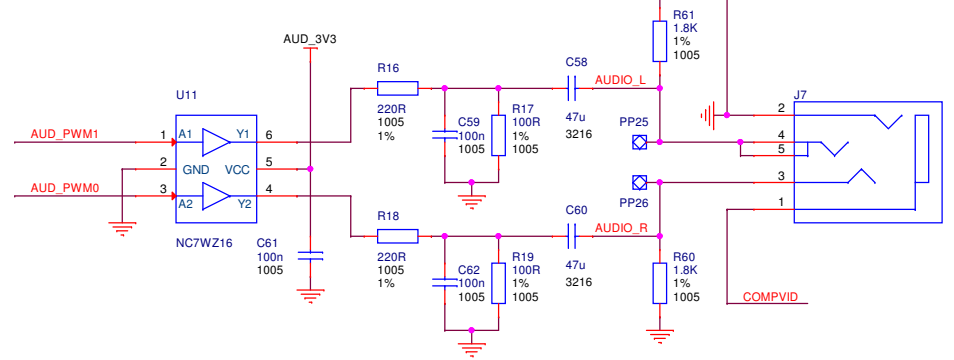


GPIO EXPANSION

**ID\_SD and ID\_SC PINS:**  
 These pins are reserved for HAT ID EEPROM. At boot time this I2C interface will be interrogated to look for an EEPROM that identifies the attached board and allows automatic setup of the GPIOs (and optionally, Linux drivers). **DO NOT USE** these pins for anything other than attaching an I2C ID EEPROM. Leave unconnected if ID EEPROM not required.



HDMI



A/V



**TÍTULO: DISEÑO DE UN EQUIPO TIME-LAPSE PARA  
SEGUIMIENTO DE OBRAS**

---

## **PLIEGO DE CONDICIONES**

---

**PETICIONARIO: ESCUELA UNIVERSITARIA POLITÉCNICA**

**AVDA. 19 DE FEBRERO, S/N**

**15405 - FERROL**

**FECHA: SEPTIEMBRE DE 2018**

**AUTOR: EL ALUMNO**

**Fdo.: YOLANDA PICALLO CAAMAÑO**

## INDICE PLIEGO DE CONDICIONES

	Páginas
5 PLIEGO DE CONDICIONES .....	115
5.1 Condiciones operativas del sistema.....	115
5.2 Especificación de los materiales. ....	115
5.3 Pruebas de verificación.....	116
5.4 Condiciones de almacenamiento. ....	116
5.5 Manual de uso.....	117
5.5.1 Alimentación del equipo.....	117
5.5.2 Configuración del menú.....	118
5.5.3 Apagado del equipo.....	119
5.5.4 Precauciones.....	120

## **5 PLIEGO DE CONDICIONES**

El pliego de condiciones tiene como finalidad establecer los criterios técnicos, económicos, administrativos y legales en orden de que el objeto del proyecto pueda materializarse bajo las citadas condiciones especificadas, evitando así posibles interpretaciones diferentes a las deseadas.

### **5.1 Condiciones operativas del sistema.**

Para garantizar el correcto funcionamiento del sistema se tendrá en cuenta que su instalación debe realizarse bajo las indicaciones especificadas en el manual de uso.

### **5.2 Especificación de los materiales.**

Los componentes han sido seleccionados de tal manera que cumplan con los requisitos de diseño, asegurando un correcto funcionamiento a lo largo del tiempo de operación del equipo.

Para cada componente, el suministrador debe garantizar el correcto funcionamiento según lo especificado en sus hojas de características y especificaciones.

En caso de que alguno de los componentes que conforman el equipo tuviera que ser sustituido por avería o cualquier otro motivo, y haya sido descatalogado o no estuviera disponible, se deberán de seleccionar otros componentes con características similares a los instalados.

Todos los componentes del equipo deberán tener obligatoriamente el certificado CE que garantice el cumplimiento de las directivas europeas de calidad. Además, serán preferentemente de fabricación nacional.

### **5.3 Pruebas de verificación.**

Se verificará antes de la puesta en marcha del equipo, que el conexionado se haya realizado de forma correcta y se deberá comprobar que todos los componentes se encuentran en buen estado, libres de deterioros.

También, se ha de verificar que tanto los componentes como la caja que los contiene se encuentren en un estado de limpieza óptimo, para garantizar el correcto funcionamiento del sistema.

### **5.4 Condiciones de almacenamiento.**

Todos los componentes que forman parte del equipo diseñado se almacenarán según las condiciones especificadas por sus fabricantes. Es aconsejable su situación en ambientes libres de polvos, preferiblemente protegidos de la radiación solar, con poca humedad relativa y cuyas temperaturas no sean extremas, estando las mismas en un rango de entre 15° y 35°.

## 5.5 Manual de uso.

### 5.5.1 Alimentación del equipo.

Para dar alimentación al equipo se incorpora una batería externa, la cual se debe de comprobar previamente que tiene carga suficiente. El equipo también incluye dos cables USB, que alimentan a la Raspberry Pi (cable blanco) y al Arduino (cable azul). Se deben de conectar los citados cables a la batería externa, como se muestra en la siguiente imagen. Una vez conectados se introducirá la batería externa dentro de la caja para evitar posibles deterioros.



Figura 5.5.1.1 - Alimentación del equipo

### 5.5.2 Configuración del menú.

Una vez conectado el equipo, se mostrará en el display LCD la pantalla inicial de la configuración del menú y se encenderá el LED rojo de conexión de la cámara. En el caso de que este LED no se encendiera, se ha de verificar que la cámara haya sido habilitada en la configuración de la Raspberry Pi, tal como se indica en el apartado 3.3.3 de Anexos.

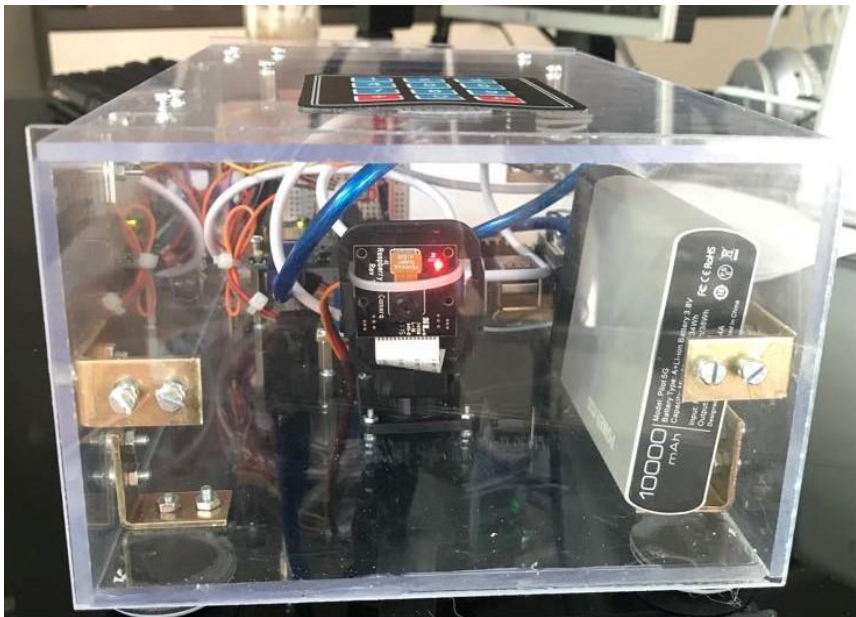


Figura 5.5.2.1 - LED Rojo cámara Raspberry Pi

A continuación, el usuario debe de introducir los datos deseados en el menú de configuración, tal y como se detalla en el apartado 3.4 de Anexos, donde se explica el funcionamiento y configuración de dicho menú.

Se podrá acceder a las fotografías de manera remota, vía VNC, con las cuales el usuario creará el video time-lapse, tal y como se detalla en los anexos 3.3.3 y 3.4.6, respectivamente.

### 5.5.3 Apagado del equipo.

A continuación, se muestra el procedimiento más adecuado para apagar el equipo.

Desde la propia Raspberry Pi, a la cual se tiene acceso remoto, se hace click en el icono de la frambuesa en la esquina superior izquierda.

En el menú desplegable que aparece, se hace click sobre el icono Shutdown.

En la ventana emergente de Shutdown Options, se escoge la opción Shutdown.

Antes de desconectar la batería, se ha de verificar que el LED de la cámara se haya apagado, lo cual es un indicativo de que la Raspberry Pi se ha apagado de manera correcta.

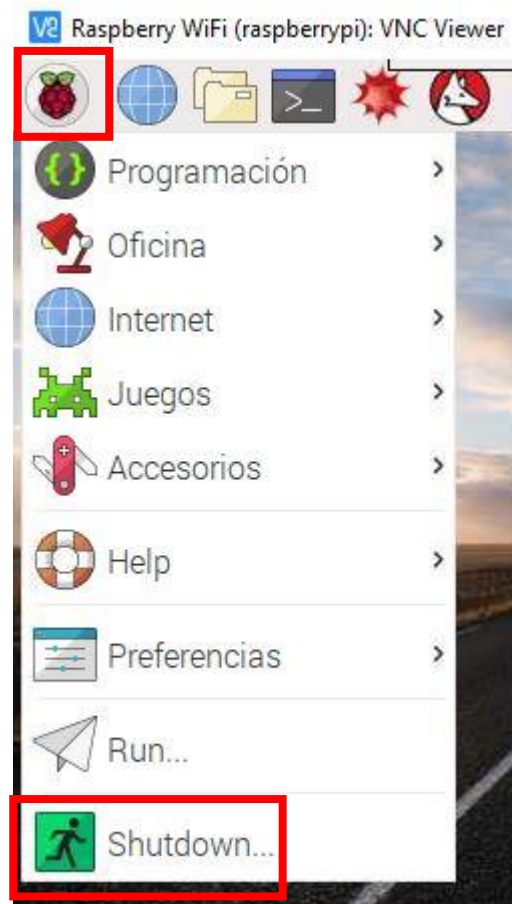


Figura 5.5.3.1 - Apagado de Raspberry Pi (Parte I)

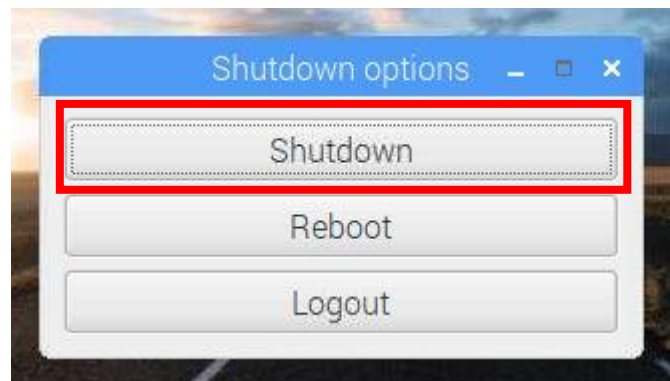


Figura 5.5.3.2 - Apagado de la Raspberry Pi (Parte II)

#### 5.5.4 Precauciones.

No se debe desconectar la alimentación sin haber apagado previamente la Raspberry Pi como se indica en el apartado 5.5.3 de este manual de uso. La desconexión de la alimentación de manera inadecuada puede provocar daños en el S.O. instalado en la Raspberry Pi, y como consecuencia, habría que instalarlo de nuevo.



**TÍTULO: DISEÑO DE UN EQUIPO TIME-LAPSE PARA  
SEGUIMIENTO DE OBRAS**

---

## **ESTADO DE MEDICIONES**

---

**PETICIONARIO: ESCUELA UNIVERSITARIA POLITÉCNICA  
AVDA. 19 DE FEBRERO, S/N  
15405 - FERROL**

**FECHA: SEPTIEMBRE DE 2018**

**AUTOR: EL ALUMNO**

**Fdo.: YOLANDA PICALLO CAAMAÑO**

## INDICE ESTADO DE MEDICIONES

	Páginas
6 ESTADO DE MEDICIONES .....	123
6.1 Materiales.....	123
6.1.1 Dispositivos electrónicos .....	123
6.1.2 Fungibles .....	126
6.1.3 Elementos de estructura.....	127
6.2 Mano de obra.....	129

## 6 ESTADO DE MEDICIONES

En este apartado se realiza una determinación de las unidades de producto que componen el proyecto, así como una breve descripción de estas.

### 6.1 Materiales

#### 6.1.1 Dispositivos electrónicos









Dispositivos electrónicos				
ID	Nombre	Imagen	Descripción	Unidades
E-001	Arduino UNO Rev.3		Tarjeta de microcontrolador	1
E-002	Raspberry Pi 3 Model B		Tarjeta de microprocesador	1
E-003	Raspberry Pi Camera Rev. 1.3		Módulo de cámara para Raspberry Pi	1

Tabla 6.1.1.1 - Tabla de componentes electrónicos (Parte I)

Dispositivos electrónicos				
ID	Nombre	Imagen	Descripción	Unidades
E-004	Kit Pan-Tilt con microservos		Incluye: Soporte y dos microservos SG90 9g	1
E-005	Módulo LCD I2D 20x4		Incluye: Display LCD, controlador I2C y 3x Cables M-H	1
E-006	Teclado matricial 3x4		Teclado matricial de membrana	1
E-007	Módulo Relé 5V		Módulo de relé 5V para Arduino	1
E-008	Batería portátil		Batería portátil de 10000 mAh	1

**Tabla 6.1.1.2 - Tabla de componentes electrónicos (Parte II)**

Dispositivos electrónicos				
ID	Nombre	Imagen	Descripción	Unidades
E-009	Tarjeta MicroSD	 A black Kingston microSD adapter with a white Kingston logo and 'microSD ADAPTER' text. Next to it is a small black Kingston microSD card with a white Kingston logo and '16GB' text.	Tarjeta MicroSD de 16GB	1

**Tabla.6.1.1.3 - Tabla de componentes electrónicos (Parte III)**

## 6.1.2 Fungibles




Fungibles				
ID	Nombre	Imagen	Descripción	Unidades
F-001	Cables Arduino		Incluye: 40x cables M-M, 40x cables M-H, 40x cables H-H	1
F-002	Placa de prototipos		Módulo board de 30 filas y 10 columnas	1
F-003	Bridas plásticas		Incluye: 50x bridas de poliamida	1

Tabla 6.1.2.1 - Tabla de fungibles

### 6.1.3 Elementos de estructura






Elementos de estructura				
ID	Nombre	Imagen	Descripción	Unidades
ES-001	Plancha de policarbonato		Plancha de 25x50 cm y 4 mm de espesor	2
ES-002	Separadores metálicos		Separadores hexagonales de M3x10 mm para circuito impreso	8
ES-003	Separadores metálicos		Separadores hexagonales de M3x25 mm para circuito impreso	3
ES-004	Pack de tornillos y arandelas M3		25x tornillos de M3x8.25mm y 25x arandelas de M3	2
ES-005	Pack de tornillos M2		10x tornillos de M2x10mm	2

Tabla 6.1.3.1 - Tabla de elementos de estructura (Parte I)

Elementos de estructura				
ID	Nombre	Imagen	Descripción	Unidades
ES-006	Pack de arandelas M2		20x arandelas de M2	1
ES-007	Pack de brocas Ø 2mm		Lote de 2x brocas de Ø 2 mm y 49 mm de longitud	1
ES-008	Pack de brocas Ø 3mm		Lote de 2x brocas de Ø 3 mm y 61 mm de longitud	1
ES-009	Escuadras angulares		Escuadras angulares bicromatada de 25x25x15x2 mm	8
ES-010	Bisagras cuadradas		Bisagras cuadradas de acero zincado de 30x30 mm	2

Tabla 6.1.3.2 - Tabla de elementos de estructura (Parte II)



## 6.2 Mano de obra.

<b>Mano de obra</b>		
<b>Nombre</b>	<b>Descripción</b>	<b>Unidades</b>
Ingeniería	Horas dedicadas al diseño del proyecto y a la realización de la programación	160
Montaje	Horas dedicadas al montaje de los dispositivos y el montaje de la caja que los contiene	30

**Tabla 6.2.1 – Tabla de mano de obra**

**TÍTULO: DISEÑO DE UN EQUIPO TIME-LAPSE PARA  
SEGUIMIENTO DE OBRAS**

---

## **PRESUPUESTO**

---

**PETICIONARIO: ESCUELA UNIVERSITARIA POLITÉCNICA**

**AVDA. 19 DE FEBRERO, S/N**

**15405 - FERROL**

**FECHA: SEPTIEMBRE DE 2018**

**AUTOR: EL ALUMNO**

**Fdo.: YOLANDA PICALLO CAAMAÑO**

## INDICE PRESUPUESTO

	Páginas
7 PRESUPUESTO.....	132
7.1 Presupuesto de materiales.....	132
7.1.1 Dispositivos electrónicos. ....	132
7.1.2 Fungibles. ....	134
7.1.3 Elementos de estructura. ....	135
7.1.4 Presupuesto total de materiales. ....	137
7.2 Presupuesto total de mano de obra. ....	138
7.3 Presupuesto final. ....	139

## 7 PRESUPUESTO

En este apartado se realiza una determinación del coste de las unidades de producto que componen el proyecto, así como de la mano de obra.

### 7.1 Presupuesto de materiales.

#### 7.1.1 Dispositivos electrónicos.





Dispositivos electrónicos						
ID	Nombre	Imagen	Descripción	PVP (sin IVA)	Ud	Total
E-001	Arduino Uno Rev.3		Tarjeta de microcontrolador	19,95 €	1	19,95 €
E-002	Raspberry Pi 3 Model B		Tarjeta de microprocesador	31,41 €	1	31,41 €
E-003	Raspberry Pi Camera Rev. 1.3		Módulo de cámara para Raspberry Pi	22,65 €	1	22,65 €
E-004	Kit Pan-Tilt		Incluye: Soporte y dos microservos SG90 9g	4,66 €	1	4,66 €

Tabla 7.1.1.1 - Tabla de presupuesto de dispositivos electrónicos (Parte I)






Dispositivos electrónicos						
ID	Nombre	Imagen	Descripción	PVP (sin IVA)	Ud	Total
E-005	Módulo LCD I2D 20x4		Incluye: Display LCD, controlador I2C y 3x Cables M-H	8,45 €	1	8,45 €
E-006	Teclado matricial 3x4		Teclado matricial de membrana	3,12 €	1	3,12 €
E-007	Módulo Relé 5V		Módulo de relé 5V para Arduino	4,66 €	1	4,66 €
E-008	Batería portátil		Batería portátil de 10000 mAh	11,05 €	1	11,05 €
E-009	Tarjeta MicroSD		Tarjeta MicroSD de 16GB	4,73 €	1	4,73 €
<b>TOTAL DISPOSITIVOS ELECTRÓNICOS (SIN IVA)</b>						<b>110,68 €</b>

Tabla 7.1.1.2 - Tabla de presupuesto de dispositivos electrónicos (Parte II)

### 7.1.2 Fungibles.




Fungibles						
ID	Nombre	Imagen	Descripción	PVP (sin IVA)	Ud	Total
F-001	Cables Arduino		Incluye: 40x cables M-M, 40x cables M-H, 40x cables H-H	6,00 €	1	6,00 €
F-002	Placa de prototipos		Módulo board de 30 filas y 10 columnas	3,32 €	1	3,32 €
F-003	Bridas plásticas		Incluye: 50x bridas de poliamida	0,66 €	1	0,66 €
<b>TOTAL FUNGIBLES (SIN IVA)</b>						<b>9,98 €</b>

Tabla 7.1.2.1 - Tabla de presupuesto de fungibles

### 7.1.3 Elementos de estructura.








Elementos de estructura						
ID	Nombre	Imagen	Descripción	PVP (sin IVA)	Ud	Total
ES-001	Plancha de policarbonato		Plancha de 25x50 cm y 4 mm de espesor	8,49 €	2	16,98 €
ES-002	Separadores metálicos		Separadores hexagonales de M3x10 mm para circuito impreso	0,12 €	8	0,96 €
ES-003	Separadores metálicos		Separadores hexagonales de M3x25 mm para circuito impreso	0,20 €	3	0,60 €
ES-004	Pack de tornillos y arandelas M3		25x tornillos de M3x8.25mm y 25x arandelas de M3	1,24 €	2	2,48 €
ES-005	Pack de tornillos M2		10x tornillos de M2x10mm	1,57 €	2	3,14 €
ES-006	Pack de arandelas M2		20x arandelas de M2	1,57 €	1	1,57 €
ES-007	Pack de brocas Ø 2mm		Lote de 2x brocas de Ø 2 mm y 49 mm de longitud	1,78 €	1	1,78 €

Tabla 7.1.3.1 - Tabla de presupuesto de elementos de estructura (Parte I)


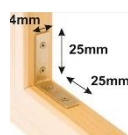

Elementos de estructura						
ID	Nombre	Imagen	Descripción	PVP (sin IVA)	Ud	Total
ES-008	Pack de brocas Ø 3mm		Lote de 2x brocas de Ø 3 mm y 61 mm de longitud	1,78 €	1	1,78 €
ES-009	Escuadras angulares		Escuadras angulares bicromatada de 25x25x15x2 mm	0,23 €	8	1,84 €
ES-010	Bisagras cuadradas		Bisagras cuadradas de acero zincado de 30x30 mm	0,51 €	2	1,02 €
<b>TOTAL ELEMENTOS DE ESTRUCTURA (SIN IVA)</b>						<b>32,15 €</b>

Tabla 7.1.3.2 - Tabla de presupuesto de elementos de estructura (Parte II)



#### 7.1.4 Presupuesto total de materiales.

<b>Presupuesto total de materiales</b>	
<b>Concepto</b>	<b>Subtotal (sin IVA)</b>
Dispositivos electrónicos	110,68 €
Fungibles	9,98 €
Elementos de estructura	32,15 €
<b>TOTAL</b>	<b>152,81 €</b>

Tabla 7.1.4.1 - Tabla de presupuesto total de materiales

## 7.2 Presupuesto total de mano de obra.

<b>Mano de obra</b>				
<b>Nombre</b>	<b>Descripción</b>	<b>Precio/hora (sin IVA)</b>	<b>Ud</b>	<b>Total</b>
Ingeniería	Horas dedicadas al diseño del proyecto y a la realización de la programación	50	160	8000 €
Montaje	Horas dedicadas al montaje de los dispositivos y el montaje de la caja que los contiene	40	30	1200 €
<b>TOTAL (SIN IVA)</b>				<b>9200 €</b>

**Tabla 7.2.1 - Tabla de presupuesto total de mano de obra**

### 7.3 Presupuesto final.

<b>Concepto</b>	<b>Subtotal</b>
Presupuesto total de materiales	152,81 €
Presupuesto total de mano de obra	9200 €
IVA (21%)	1964,1 €
<b>TOTAL</b>	<b>11316,91 €</b>

**Tabla 7.3.1 - Tabla de presupuesto final**

El presupuesto total de este proyecto es de:

*Once mil trescientos dieciséis euros con noventa y un céntimos.*