

Universitat Politècnica de València

Departamento de Ingeniería Mecánica y Materiales.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

CAMPUS D'ALCOI

Grado en Ingeniería Mecánica

Trabajo Fin de Grado

DISEÑO E IMPLEMENTACIÓN DE
LA AUTOMÁTICA EN UN BANCO DE
ENSAYOS A TRACCIÓN

Autor:

David Rodríguez Cruz

Dirigido por:

Miguel Ángel Sellés Cantó

Samuel Sánchez Caballero

Julio 2017

RESUMEN

“DISEÑO E IMPLEMENTACIÓN DE LA AUTOMÁTICA EN UN BANCO DE ENSAYOS A TRACCIÓN”

En la presente memoria se detallará la automatización y control remoto de un antiguo banco de ensayos a tracción. El cual se encuentra situado en el laboratorio FIL5 de la EPSA.

Para obtener una correcta programación se divide el trabajo en las siguientes tareas:

- Diseño 3D del banco de ensayos.
- Análisis y sustitución de la instalación eléctrica.
- Explicación del sistema hidráulico.
- Programación vía Wifi mediante Arduino.
- Puesta en marcha mediante control remoto.

Durante la realización del proyecto surgen algunos problemas, como la modificación de la estructura del banco, la necesidad de introducir de nuevos componente eléctricos en la instalación o la adquisición de una nueva red Wifi denominada FATIGA, entre otros.

Obteniendo un final de proyecto completamente satisfactorio, al conseguir resolver todos los problemas y obtener un banco completamente automatizado, capaz de ser controlado remotamente desde cualquier lugar de la EPSA donde se reciba conexión a la red UPVNET.

Para el diseño y elaboración de los planos correspondientes se han empleado diferentes herramientas CAD, como *Autodesk AutoCAD 2016* o *SolidWorks 2016*. Por otra parte se adjuntan todos los anexos necesarios, como los planos de construcción, instalación eléctrica y el programa de Arduino.

RESUM

“DISSENY I IMPLEMENTACIÓ DE L’AUTOMÀTICA EN UN BANC D’ASSATJOS A TRACCIÓ”

En la present memòria es detallarà l’automatització i control remot d’un banc d’assatjos a tracció. El qual es troba situat en el laboratori F1L5 de l’EPSA.

Per a obtindre una correcta programació es divideix el treball en les tasques següents:

- Disseny 3D del banc d’assatjos.
- Anàlisi y substitució de l’instal·lació elèctrica.
- Explicació del sistema hidràulic.
- Programació via Wifi per mitjà d’Arduino.
- Posada en marxa per mitjà de control remot.

Durant la realització del projecte sorgixen alguns problemes, com la modificació de l’estructura del banc, la necessitat d’introduir de nous component elèctrics en la instal·lació o l’adquisició d’una nova xarxa Wifi denominada FATIGA, entre altres.

Obtenint un final de projecte completament satisfactori, a l’aconseguir resoldre tots els problemes i obtindre un banc completament automatitzat, capaç de ser controlat remotament des de qualsevol lloc de l’EPSA on es reba connexió a la xarxa UPVNET.

Per al disseny i elaboració dels planols corresponents s’han empleat diferents ferramentes CAD, com Autodesk AutoCAD 2016 o SolidWorks 2016. D’altra banda s’adjunten tots els annexos necessaris, com els plans de construcció, instal·lació elèctrica i el programa d’Arduino.

SUMMARY

“DESIGN AND IMPLEMENTATION OF AUTOMATION IN A TRACTION TEST BENCH”

This project details the automation and remote control of an old traction test bench. Which is located in the laboratory FIL5 of the EPSA.

To obtain a correct programming, the work is divided into the following tasks:

- 3D design of the test bench.
- Analysis and replacement of the electrical installation.
- Explanation of the hydraulic system.
- Arduino Wifi programming
- Start up by remote control.

During the implementation of the project some problems arise, such as the modification of the bank structure, the need to introduce new electrical components in the installation or the acquisition of a new Wifi network called FATIGA, among others.

Obtaining a completely satisfactory project end, by solving all the problems and obtaining a fully automated bank, able to be controlled remotely from any place of the EPSA where it receives a connection to the network UPVNET.

Different CAD tools, such as Autodesk AutoCAD 2016 or SolidWorks 2016, have been used for the design and development of the corresponding drawings. All necessary attachments, such as the construction drawings, electrical installation and the Arduino program, are attached.

Tabla de Contenidos

RESUMEN.....	3
RESUM	4
SUMMARY	5
ABREVIATURAS.....	8
LISTADO DE FIGURAS	9
LISTADO DE TABLAS	13
1. INTRODUCCIÓN	14
1.1. Objetivos del Proyecto.....	15
1.1.1. Objetivos Generales	15
1.2.1. Objetivos Específicos	15
1.2. Automática.....	16
1.3. Hidráulica.....	18
1.4. Sistemas Eléctricos.....	22
1.5. Planificación temporal y banco de tracción.....	26
2. Antecedentes	27
2.1. Máquina de ensayos.....	28
2.1.1. Tipos de banco según su estructura	28
2.1.2. Tipos de banco según su accionamiento	30
2.2. Sistemas de control.....	32
3. Descripción del banco BTL-01	34
3.1. Ensayo a tracción.....	35
3.2. Estructura del Banco.....	37

3.3. Sistema Eléctrico.....	38
3.4. Sistema Hidráulico.....	40
4. Modificaciones del banco BTL-01	42
4.1. Estructura del Banco.....	43
4.2. Sistema Eléctrico.....	48
4.2.1. Cálculos y componentes.....	48
4.2.2. Instalación y planos.....	55
4.3. Sistema Hidráulico.....	59
5. Diseño de la automática	60
5.1. Programación.....	62
5.1.1. Programación vía serie.....	63
5.1.2. Programación vía Wifi.....	69
6. Implementación.....	73
6.1. Comunicación vía Wifi.....	74
6.1.1. Conexión en red doméstica.....	74
6.1.2. Conexión en la red universitaria.....	77
7. Pruebas y Resultados.....	80
8. Estudio económico.....	85
9. Bibliografía.....	87
Anexos.....	89
Anexo 1: Programa Wifi.....	90
Anexo 2: Planos de fabricación.....	95
Anexo 3: Planos eléctricos.....	103

ABREVIATURAS

V	Voltios
A	Amperios
I	Intensidad
L	Longitud
Lo	Longitud inicial
T	Tensión
t	Tiempo
S	Sección del cable/ Sección probetas
C	Constante de conductividad del cobre.
%V	Caída de tensión
γ	Factor de potencia
F	Fuerza
ϵ	Deformación unitaria o alargamiento
σ	Tensión

LISTADO DE FIGURAS

Figura 1.2.1 <i>Brazos robóticos completamente automatizados en una cadena de montaje de automóviles.</i>	16
Figura 1.3.1 <i>Compresor ABAC 100Hp 2.</i>	18
Figura 1.3.2 <i>Bomba HIDROLEDUC 400bar.</i>	18
Figura 1.3.3 <i>Filtro regulador y lubricante.</i>	19
Figura 1.3.4 <i>Válvula hidráulica de secuencia PARKER R4S.</i>	19
Figura 1.3.5 <i>Motor ABB 3'3 kW.</i>	20
Figura 1.3.6 <i>Esquema explicativo de un Sistema Hidráulico básico.</i>	20
Figura 1.3.7 <i>Esquema explicativo de un Sistema Neumático básico.</i>	21
Figura 1.4.1 <i>Torres de alta tensión.</i>	22
Figura 1.4.2 <i>Postes eléctricos de tensión media.</i>	23
Figura 1.4.3 <i>Instalación eléctrica a baja tensión.</i>	23
Figura 1.4.4 <i>Gráfico de la variación del voltaje en el tiempo en corriente continua.[1]</i>	24
Figura 1.4.5 <i>Gráfico de la variación del voltaje en el tiempo en corriente alterna.[1]</i>	24
Figura 1.4.6 <i>Componentes de un Sistema Eléctrico.</i>	25
Figura 2.1.1.1 <i>Banco de ensayos a tracción EAST mono espacio.</i>	28
Figura 2.1.1.2 <i>Banco de ensayos a tracción EAST doble espacio.</i>	29
Figura 2.1.1.3 <i>Banco de ensayos a tracción automatizado.</i>	29
Figura 2.1.2.1 <i>Banco de ensayos hidráulico IBTM4-1000.</i>	30
Figura 2.1.2.2 <i>Banco de ensayos electromecánico de sobremesa Serie TESTCOM.</i>	30
Figura 2.1.2.3 <i>Banco de ensayos mecánico con accionamiento manual.</i>	31
Figura 2.2.1 <i>Clepsidra. Reloj de agua árabe.</i>	32
Figura 2.2.2 <i>Microcontrolador Arduino genuino UNO.</i>	33

Figura 3.1.1 Esquema del funcionamiento del ensayo a tracción.....	35
Figura 3.1.2 Grafica tensión σ desplazamiento ϵ de un ensayo de tracción.	36
Figura 3.2.1 Plano 3D del antiguo banco de tracción.....	37
Figura 3.3.1 Antiguo cuadro eléctrico del banco de ensayos.	38
Figura 3.3.2 Antiguo esquema eléctrico del banco de ensayos.....	39
Figura 3.4.1 Sistema hidráulico del banco de ensayos.	41
Figura 4.1.1 Planos de alzado y planta del banco de ensayos.....	43
Figura 4.1.2 Conjunto A.....	44
Figura 4.1.3 Eje guías.....	44
Figura 4.1.4 Conjunto B.....	45
Figura 4.1.5 Conjunto C.....	45
Figura 4.1.6 Camisa del pistón hidráulico.....	46
Figura 4.1.7 Mordazas para realizar ensayos a tracción.....	46
Figura 4.1.8 Esquema del desplazamiento en el eje Y del banco.....	47
Figura 4.2.1.1 Interruptores de avance, retroceso y marcha del antiguo cuadro eléctrico.....	48
Figura 4.2.1.2 Esquema para el cálculo de potencias.....	49
Figura 4.2.1.3 Interruptor general Lovato C16 P1 MB.....	50
Figura 4.2.1.4 Motor hidráulico ABB IP55 3'3 kW.....	50
Figura 4.2.1.5 Interruptores magnetotérmicos ABB y Schneider de 3 y 5 Amperios.....	51
Figura 4.2.1.6 Contactor Lovato BG1201A230.....	51
Figura 4.2.1.7 Fuente de alimentación AMR1-12.....	52
Figura 4.2.1.8 Arduino Uno Wifi Shield.....	52
Figura 4.2.1.9 Transformador siemens T50/B 0550 T1.....	53
Figura 4.2.1.10 Puente rectificador Semikron SKB 30/04 A1.....	53

Figura 4.2.1.11 <i>Relé Finder 30.22.7.005.0010</i>	54
Figura 4.2.2.1 <i>Plano potencia.</i>	55
Figura 4.2.2.2 <i>Plano alimentación Arduino.</i>	56
Figura 4.2.2.3 <i>Plano control Arduino.</i>	57
Figura 4.2.2.4 <i>Plano control de válvulas.</i>	57
Figura 4.2.2.5 <i>Cuadro eléctrico actual.</i>	58
Figura 4.3.1 <i>Esquema hidráulico del banco de ensayos.</i>	59
Figura 5.1 <i>Estructura base de un sketch de Arduino.</i>	61
Figura 5.1.1 <i>Arduino 4 Relays Shield.</i>	62
Figura 5.1.2 <i>Conexión del Arduino mediante el puerto COM.</i>	62
Figura 5.1.1.1 <i>Sketch, Introducción de las variables en Arduino.</i>	63
Figura 5.1.1.2 <i>Sketch, Introducción de los comandos y variables para la comunicación serie.</i> 64	
Figura 5.1.1.3 <i>Sketch, Programación del Void serialEvent para la obtención un nuevo lenguaje.</i>	66
Figura 5.1.1.4 <i>Sketch, programación del Void paro.</i>	67
Figura 5.1.1.5 <i>Sketch, programación de los comandos para el control del banco de ensayos.</i> 68	
Figura 5.1.1.6 <i>Comprobación del Sketch mediante la consola Wifi.</i>	69
Figura 5.1.2.1 <i>Sketch, programación de librerías y comandos para la comunicación Wifi.</i>	71
Figura 5.1.2.2 <i>Sketch, recolocación del Void serialEvent y Void paro.</i>	72
Figura 6.1.1.1 <i>Conexión a la red Wifi de Arduino.</i>	74
Figura 6.1.1.2 <i>Obtención de la puerta de enlace predeterminada.</i>	75
Figura 6.1.1.3 <i>Obtención de la nueva IP doméstica.</i>	75
Figura 6.1.1.4 <i>Comprobación del sketch por la consola Wifi de Arduino.</i>	76
Figura 6.1.1.5 <i>Conexión del programa Arduino 1.8.2 por puerto red.</i>	77
Figura 6.1.2.1 <i>Propiedades de la red FATIGA.</i>	78

Figura 6.1.1.5 <i>Conexión del Arduino con la red FATIGA.</i>	78
Figura 7.1 <i>Consola Wifi desde dispositivo móvil.</i>	81
Figura 7.2 <i>Devolución del comando marcha en la consola Wifi.</i>	82
Figura 7.3 <i>Ensayo a tracción. Rotura un eslabón.</i>	83
Figura 7.4 <i>Ensayo a tracción. Rotura conjunto en cadena dos eslabones.</i>	83

LISTADO DE TABLAS

Tabla 1.5 <i>Diagrama de Gantt. Planificación temporal del proyecto.</i>	26
Tabla 5.1.1 <i>Comandos de control del Arduino</i>	68
Tabla 7.1 <i>Resultados obtenidos de la comunicación entre el banco y los controladores</i>	84
Tabla 8. <i>Presupuesto del Proyecto.</i>	86

1. INTRODUCCIÓN

INTRODUCCIÓN

1. Introducción

1.1. Objetivos del Proyecto.

Los principales objetivos para este proyecto se pueden dividir en dos grupos.

1.1.1. Objetivos Generales

El principal objetivo del presente Trabajo Fin de Grado (TFG) es obtener la correcta automatización de un banco de ensayos a tracción, mediante la programación de un Arduino Wifi.

1.2.1. Objetivos Específicos

Del proyecto también se esperan obtener una serie de objetivos más específicos:

- Revisión y sustitución de la instalación eléctrica del banco.
- Realización de nuevos planos eléctricos.
- Obtención de nuevos planos estructurales.
- Realización de un diseño 3D en SolidWorks.
- Obtener un control más exacto y directo sobre los ensayos de tracción mediante la automatización del banco.
- Realizar un control remoto del banco, pudiéndolo controlar desde cualquier rincón de la universidad.
- Evitar la retirada del banco del laboratorio de la universidad debido a su antiguo sistema de control mecánico.
- Innovar en la forma de control de las maquinarias de la universidad.
- Facilitar la revisión de los ensayos sin necesidad de permanecer físicamente en el aula de ensayo.

1. Introducción

1.2. Automática.

El control de procesos y maquinaria debido a la automática es una de las disciplinas que más se ha desarrollado en las últimas décadas, dando así las bases a lo que algunos autores actuales denominan la segunda revolución industrial.

El estudio y aplicación de la automática es reconocido universalmente gracias a la gran cantidad de beneficios y ventajas que nos otorga en el ámbito de la industria. En este ámbito es donde más aplicaciones le encontramos debido al cuantioso número de variables y la complejidad de sus sistemas.

Para entender un poco mejor de que trata la automática es necesario definir el significado de sistema.

Un sistema automático de control es un conjunto de componentes físicos conectados o relacionados entre sí, de manera que regulen o dirijan su actuación por sí mismos, es decir, sin intervención de agentes exteriores, corrigiendo además los posibles errores que se presenten en su funcionamiento.



Figura 1.2.1 Brazos robóticos completamente automatizados en una cadena de montaje de automóviles.

Por otra parte la automática nos permite gozar de una gran cantidad de ventajas en varios campos y no tan sólo en el ámbito industrial.

1. Introducción

- En los procesos industriales:

-Aumentando las cantidades y mejorando la calidad del producto, gracias a la producción en serie y a las cadenas de montaje.

-Reduciendo los costes de producción.

-Fabricando artículos que no se pueden obtener por otros medios.

-En los hogares: Mejorando la calidad de vida. Desde un lavavajillas hasta la completa automatización de un edificio (domótica).

-Para los avances científicos: Como las misiones espaciales.

-Para los avances tecnológicos: Por ejemplo, la aparición de alumbrado inteligente en los automóviles.

1. Introducción

1.3. Hidráulica.

Actualmente los sistemas hidráulicos se pueden dividir en dos grandes grupos la neumática y la oleo hidráulica. Ambos grupos utilizan los mismos principios científicos sobre los fluidos, pero presentan varias diferencias en cuanto a fluidos y elementos utilizados. Dotando de diferentes características a cada sistema.

Tanto en la neumática como en la oleo hidráulica se distinguen 4 elementos.

-Elementos generadores de energía: Compresores en la neumática y bombas hidráulicas para la oleo hidráulica.



Figura 1.3.1 *Compresor ABAC 100Hp 2.*



Figura 1.3.2 *Bomba HIDROLEDUC 400bar.*

1. Introducción

-Elementos de tratamiento: Filtros y reguladores de presión en ambos casos.



Figura 1.3.3 *Filtro regulador y lubricante.*

-Elementos de control y mando: Tuberías y válvulas para ambos sistemas.



Figura 1.3.4 *Válvula hidráulica de secuencia PARKER R4S.*

1. Introducción

-Elementos actuadores: Para ambos sistemas los elementos actuadores utilizados son motores y cilindros.



Figura 1.3.5 Motor ABB 3'3 kW.

Respecto a los fluidos es donde encontramos la mayor diferencia. La neumática utiliza como fluido gases con alta capacidad de compresión, mientras que la hidráulica utiliza aceites con una capacidad de compresión muy baja. Es por esto que los sistemas neumáticos pueden ser abiertos con salidas al ambiente. Mientras que los hidráulicos son completamente cerrados, lo que nos obliga a la instalación de tanques de aceite.

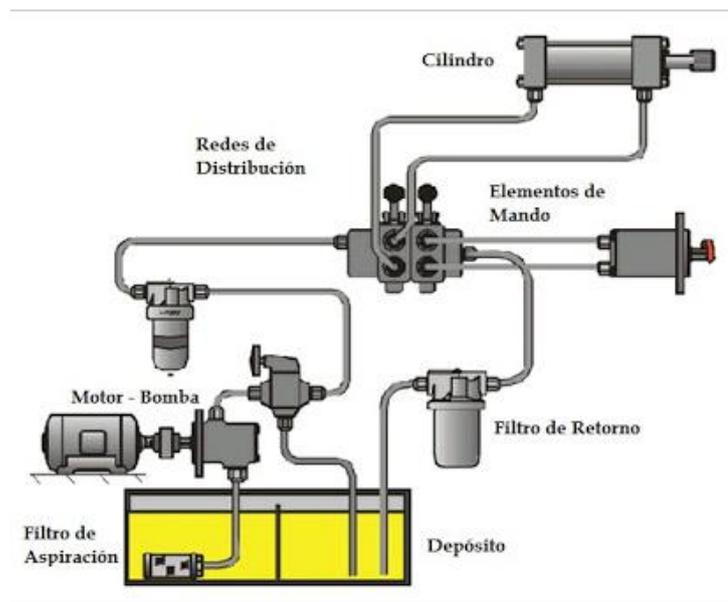


Figura 1.3.6 Esquema explicativo de un Sistema Hidráulico básico.

1. Introducción

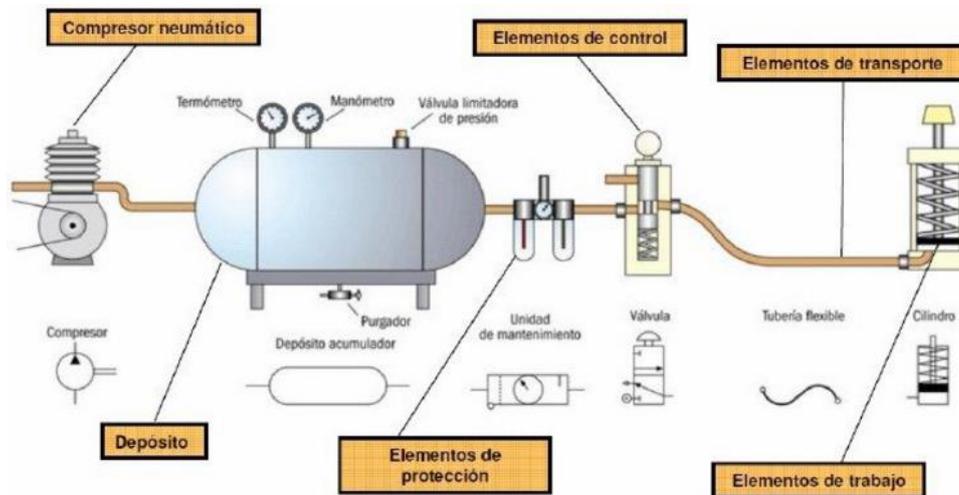


Figura 1.3.7 Esquema explicativo de un Sistema Neumático básico.

Finalmente en cuanto a las prestaciones de los sistemas encontramos las siguientes diferencias:

-Sistema Neumático:

- Cargas por debajo de los 3000KG.
- Motores de alta velocidad capaces de superar las 500.000rpm.
- Aplicaciones en un gran ámbito de tareas. Limpieza a presión, elevación en montacargas, apertura y cierre de puertas, utilización en herramientas de odontología, etc.

-Sistema Oleo Hidráulico:

- Cargas mucho más elevadas
- Control exacto de la velocidad y del paro.
- Aplicaciones para la industria metalúrgica, Sistemas de transporte, maquinaria de obras públicas, industria naval, aeronáutica, etc.

1.4. Sistemas Eléctricos.

En la actualidad es tan común la aplicación de sistemas eléctricos en todo tipo de aplicaciones que no se le da la importancia adecuada a este tipo de sistemas. Esto se debe a que la electricidad es la forma de energía más utilizada en la industria y los hogares.

La definición de sistema eléctrico viene dada como el conjunto de sistemas, conductores y equipos necesarios para el transporte, distribución y generación de energía eléctrica.

Todo sistema eléctrico necesita de la transmisión de corriente, para ello se debe administrar corriente mediante una fuente de alimentación, la cual es la encargada de suministrar la energía necesaria al sistema.

Para obtener esta energía eléctrica es necesario la creación de infraestructuras capaces de obtener energía eléctrica a partir de otros medios o sistemas, como las plantas hidroeléctricas, plantas nucleares, aerogeneradores, placas solares, a través del calor de la tierra o de las mareas, etc.

Esta energía eléctrica obtenida se cataloga según su voltaje.

-Alta tensión: Se emplea para la transmisión de energía desde las centrales generadoras a grandes distancias. Esto es debido a que la pérdida de energía a altos voltajes es menor. El voltaje mínimo para considerar alta tensión son 25kV. Su trasportación se efectúa mediante altas torres metálicas con un gran aislamiento eléctrico.



Figura 1.4.1 Torres de alta tensión.

-Tensión media: Debido a la peligrosidad de las altas tensiones en sitios cercanos a la población se instalan subestaciones capaces de transformar la energía a valores entre 1kV y 25kV. La instalación se puede realizar en postes de madera o metálicos o situarse por el subsuelo.

1. Introducción



Figura 1.4.2 Postes eléctricos de tensión media.

-Baja tensión: Son todas aquellas tensiones inferiores a 1 kW. En España la tensión regulada para la industria oscila entre los 220V y 440V y en los hogares a 230V. El transporte de la baja tensión se realiza a través de cableado, normalmente de cobre. Las dimensiones del cableado pueden variar dependiendo del consumo de la instalación. Estas dimensiones se regulan mediante el Reglamento Electrotécnico de Baja Tensión.



Figura 1.4.3 Instalación eléctrica a baja tensión.

Cabe indicar que la corriente puede transmitirse de forma plana (corriente continua) o en forma de onda (corriente alterna).

-Corriente continua: En corriente continua se genera una tensión constante que no varía a lo largo del tiempo. La gran mayoría de aparatos tecnológicos, aunque se conecten a corriente alterna en su interior, la transforman en continua; ordenadores, televisores, teléfonos móviles, etc. La mayoría de generadores de corriente continua son baterías y pilas.

1. Introducción

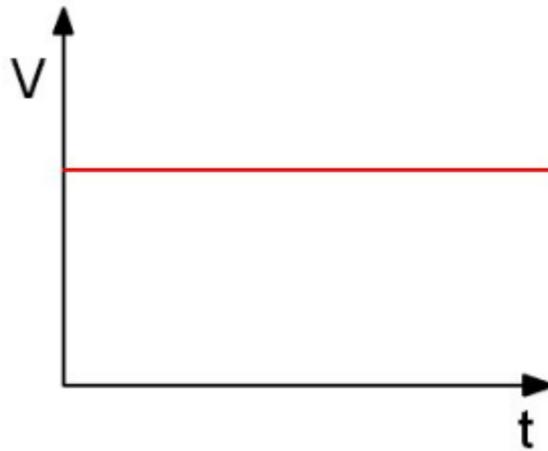


Figura 1.4.4 Gráfico de la variación del voltaje en el tiempo en corriente continua.[1]

-Corriente alterna: Este tipo de corriente es producida por los alternadores y es la que se genera en las centrales eléctricas. La corriente que usamos en los enchufes o tomas de corriente de las viviendas es de este tipo. Este tipo de corriente es la más habitual porque es la más fácil de generar y transportar. En corriente alterna, la intensidad varía con el tiempo (número de electrones variable) y además, cambia de sentido de circulación a razón de 50 veces por segundo (frecuencia de 50Hz). También la tensión generada entre los dos bornes (polos) varía con el tiempo en forma de onda senoidal, por lo que no es constante.

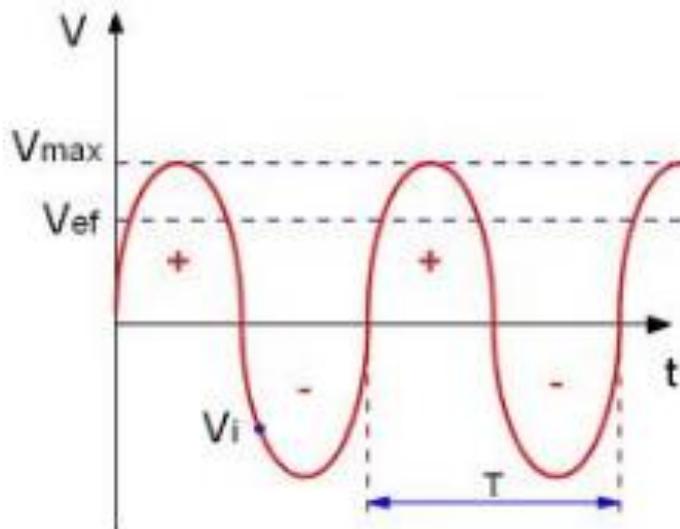


Figura 1.4.5 Gráfico de la variación del voltaje en el tiempo en corriente alterna.[1]

1. Introducción

Actualmente existen una gran cantidad de componentes eléctricos que nos permiten la realización de infinitas combinaciones en los sistemas de baja tensión, contactores, controladores, generadores de corriente, conductores, microcontroladores, etc.



Figura 1.4.6 *Componentes de un Sistema Eléctrico.*

1. Introducción

1.5. Planificación temporal y banco de tracción.

Para el desarrollo del proyecto se hace uso de las instalaciones de un antiguo banco de ensayos a tracción, situado en el aula F1L5 de la EPSA.

A partir de esta parte del proyecto y por motivos de simplificación el banco de ensayos a tracción se identificara como BTL-01 (Banco de ensayos a Tracción situado en el Laboratorio de la planta 1)

En el presente proyecto se procederá a realizar los objetivos marcados en la memoria. Para ello se realiza un plano temporal indicando la cantidad de objetivos cumplidos dentro de los 3 meses de plazo para realizar el Trabajo Final de Grado (TFG).

Para ello haremos uso de un diagrama de Gantt.

Tabla 1.5 Diagrama de Gantt. Planificación temporal del proyecto.

Objetivos	Fecha Inicial	Fecha Final	03-abr	10-abr	17-abr	24-abr	01-may	08-may	15-may	22-may	29-may	05-jun	12-jun	19-jun	26-jun	30-jun
1	03/04/2017	10/04/2017	■	■												
2	11/04/2017	01/05/2017			■	■	■									
3	02/05/2017	08/05/2017					■									
4	09/05/2017	02/06/2017						■	■	■						
5	05/06/2017	16/06/2017										■	■			
6	19/06/2017	23/06/2017												■		
7	25/06/2017	30/06/2017													■	■

Los objetivos definidos en la tabla anterior son:

- Objetivo 1: Diseño 3D del banco y creación de planos.
- Objetivo 2: Revisión y sustitución del antiguo sistema eléctrico.
- Objetivo 3: Realización de los planos eléctricos.
- Objetivo 4: Programación vía Serie.
- Objetivo 5: Programación vía Wifi.
- Objetivo 6: Comunicación entre los dispositivos de control y el Arduino Wifi.
- Objetivo7: Pruebas y resultados

2. Antecedentes

Antecedentes

2. Antecedentes

2.1. Máquina de ensayos.

En ingeniería se denomina máquina universal a una máquina semejante a una prensa, con la que es posible someter materiales a ensayos de tracción y compresión para medir sus propiedades. La presión se otorga mediante placas o mandíbulas accionadas por tornillos o sistemas hidráulicos.

Las máquinas se pueden clasificar según su estructura o accionamiento.

2.1.1. Tipos de banco según su estructura

-Mono espacio o de una sola columna: Este tipo de banco se caracteriza por tener un único espacio de trabajo. La máquina solo puede realizar un tipo de ensayo.



Figura 2.1.1.1 Banco de ensayos a tracción EAST mono espacio.

2. Antecedentes

-Doble espacio o doble columna: Los bancos de doble columna nos permiten realizar más de un ensayo a la vez, como se observa en la figura, y con las pinzas adecuadas podemos realizar un ensayo a tracción y compresión al mismo tiempo.



Figura 2.1.1.2 Banco de ensayos a tracción EAST doble espacio.

-De automatización: Los bancos automatizados vienen integrados con sistemas auxiliares, como ordenadores, capaces de controlar y regular las variantes del banco de forma externa a este. Actualmente, tanto los bancos mono espacio como los doble pertenecerían a este tipo de bancos.



Figura 2.1.1.3 Banco de ensayos a tracción automatizado.

2. Antecedentes

2.1.2. Tipos de banco según su accionamiento

-Hidráulicas: Los bancos de ensayo hidráulicos presentan la gran ventaja de poder generar altas presiones con facilidad. Podemos encontrar bancos de cualquier tipo, tanto mono como de doble espacio. El accionamiento de la máquina es impulsado por un sistema bomba pistón.



Figura 2.1.2.1 Banco de ensayos hidráulico IBTM4-1000.

-Electromecánicas: Sus principales cualidades son una máxima precisión en los resultados, gran versatilidad, configuraciones de uno o más espacios de trabajo, fácil instalación y mínimo mantenimiento. Son los bancos más adecuados para trabajar a altas y bajas temperaturas y para ensayos de larga duración.



Figura 2.1.2.2 Banco de ensayos electromecánico de sobremesa Serie TESTCOM.

2. Antecedentes

-Mecánicas: Son las menos comunes actualmente. El accionamiento puede ser de motor, tornillo, cadena, palanca, etc. El inconveniente principal de estos bancos es que son los que mayor intervención humana necesitan y en los que menor control y precisión se puede tener.



Figura 2.1.2.3 Banco de ensayos mecánico con accionamiento manual.

Actualmente es muy sencillo poder encontrar cualquier tipo de banco de ensayos ya que existen empresas, especializadas en el diseño y construcción de bancos, que facilitan la adquisición del banco necesario para cada aplicación. Entre las empresas más destacadas en el mercado podemos encontrar Ibertest, MTS, Tenson, Zwick, etc.

2. Antecedentes

2.2. Sistemas de control.

Desde la prehistoria el hombre ha intentado sustituir todo esfuerzo humano en otra forma de esfuerzo que pudiera ser controlado y manejado por el hombre de forma más simple y sencilla, como por ejemplo el levantamiento de pesos por poleas. Incluso se datan en el 300 AC algunos mecanismos que simulaban autómatas. Estos mecanismos se basaban en flujos canalizados de agua o vapor para producir acciones simples y repetitivas, tal como figuras de agua, creación de canciones o juegos. Un ejemplo de esto son los proto-autómatas descubiertos en el antiguo Egipto o artefactos similares encontrados en la India o China.

Durante el paso de los siglos han aparecido una gran variedad de intentos de autómatas, como por ejemplo los primeros relojes datados en el siglo VII. Los ingenieros árabes se convierten en los mejores constructores de clepsidras y autómatas, recuperando el concepto de realimentación con la construcción de reguladores de flotador.



Figura 2.2.1 Clepsidra. Reloj de agua árabe.

Acercándonos un poco más a la actualidad, en 1801 aparece la patente de un taller automático que utilizaba tarjetas perforadoras. La patente fue dada por Joseph Marie Jacquard, el cual revolucionó la industria del textil.

Es a mediados del siglo XX cuando la automatización sufre un boom industrial. Hasta entonces la automatización se podía observar a pequeña escala, sobretodo en la industria. Con la evolución de las computadoras digitales ya era posible manejar casi cualquier tipo de tarea. Las primeras aplicaciones de estas computadoras fueron en 1960. Hasta este entonces las computadoras eran únicamente analógicas y no aportaban la flexibilidad necesaria.

Pero no es hasta 2005, y debido a una necesidad, cuando aparece lo que ahora denominamos como Arduino. En la universidad de Ivrea (Italia), el estudiante Hernando Barragán, con la ayuda de su tutor Massimo Banzi, desarrollo una tarjeta electrónica denominada Wiring.

2. Antecedentes

La universidad, envuelta en la necesidad de obtención de microcontroladores baratos para evitar el cierre, se ve obligada a desarrollar sus propias placas. Massimo Banzi se basa en el proyecto de su alumno para desarrollar una plataforma de hardware y software libre más pequeña y económica, a la que se le denominó Arduino.

Poco después la universidad cierra sus puertas y los profesores se ven obligados a intentar sobrevivir con el nuevo sistema de Arduino. Es aquí cuando el proyecto empieza a ganar fuerza; nuevos integrantes como Tom Igoe se unen al proyecto. Tom se encarga de distribuir y enseñar a estudiantes de Estados Unidos el nuevo sistema. Las placas gustaron tanto que poco después de su lanzamiento desplazaron del mercado a BasicStamp y Pics. Finalmente, y con la ayuda de Google para el desarrollo del Kit ADK (Accessory Development Kit), se busca realizar un primera versión capaz de comunicarse con múltiples sistemas operativos, y no superar los 30 €.

Actualmente, Arduino se trata de una compañía de hardware libre y una compañía tecnológica que diseña y manufactura placas de computadora compuestas por circuitos impresos que integran un microcontrolador y un entorno de desarrollo IDE en donde se programa cada placa. Debido a la gran cantidad de sistemas a controlar, Arduino se ve obligado a diseñar una gran cantidad de placas con diferentes características.

Los Arduinos más conocidos son:

- Arduino Uno
- Arduino Mega 2560
- Arduino Robot
- Arduino Mini
- LilyPad Arduino Simple
- Arduino Pro
- Arduino Fio
- Arduino Pro Mini
- Arduino Due
- Arduino Ethernet
- Arduino Mega ADK
- Arduino UNO Wifi
- Arduino Nano
- Arduino Leonardo
- LilyPad Arduino USB
- Arduino Esplora
- Arduino Micro
- Arduino Yún



Figura 2.2.2 Microcontrolador Arduino genuino UNO.

3. Descripción del banco BTL-01

Descripción del banco BTL-01

3. Descripción del banco BTL-01

Para poder plasmar el trabajo realizado en el proyecto es necesario describir en qué condiciones se encontraban las instalaciones del banco de ensayos BTL-01.

El banco se encontraba en una situación un tanto delicada, ya que al tratarse de un banco con accionamiento mecánico el personal de la universidad necesitaba una presencia completa a la hora de realizar el ensayo.

Esto provocó el planteamiento de la adquisición de un nuevo banco automatizado, pero el gran desembolso que planteaba esta opción permitió dar pie al actual proyecto.

El primer paso para la automatización del banco es realizar un análisis del banco para comprobar en qué condiciones se encuentra y comprobar si es posible la instalación de Arduino para la automatización.

Para ello se divide el análisis en cuatro partes:

3.1. Ensayo a tracción.

Lo más importante antes de empezar con la automática, es entender qué tipo de ensayos se realizan. Ya que dependiendo de los ensayos, el movimiento y los componentes de la máquina, como por ejemplo las mordazas, serían completamente distintos.

Teniendo en cuenta que actualmente los ensayos a tracción son uno de los ensayos más realizados para todo tipo de investigación de materiales, se decide centrar el proyecto y la reforma del banco para este tipo de ensayos.

Por tanto, es necesario explicar brevemente en qué consisten.

El ensayo a tracción de un material consiste en someter una o varias probetas normalizadas a uno o varios esfuerzos axiales crecientes hasta producir la rotura de la misma.

Las probetas se colocan en las mordazas del banco de ensayo las cuales se encargan de sostener la probeta. Una vez sujeta al travesero o cabezal móvil, se desplaza. Normalmente las velocidades de ensayo son lentas, aplicando sobre la probeta una fuerza axial aplicada por la célula de carga.

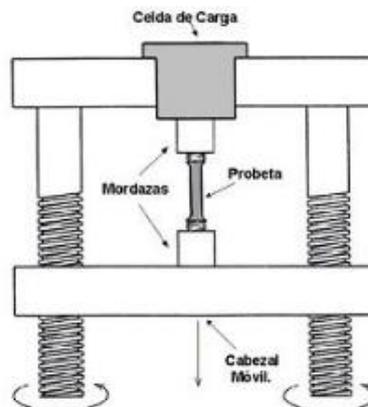


Figura 3.1.1 Esquema del funcionamiento del ensayo a tracción.

3. Descripción del banco BTL-01

Durante el ensayo se analiza el comportamiento de la probeta a estos esfuerzos y se hace una toma de datos.

Los datos a obtener son:

- Módulo de elasticidad o módulo de Young
- Coeficiente de Poisson
- Límite de fluencia.
- Límite elástico.
- Resistencia a tracción o carga de rotura.
- Alargamiento de rotura, Longitud calibrada y Reducción de área y estricción.

Para poder obtener estos datos se necesita la creación de un gráfico tensión/alargamiento de la probeta. Para ello es necesario la aplicación de una serie de fórmulas.

$$\sigma = \frac{F}{S}$$

Sabiendo la fuerza F que aplica el banco y la superficie de la probeta S obtenemos la tensión aplicada.

Para obtener el alargamiento necesitaremos la longitud de la probeta en un instante determinado L y la longitud inicial de la probeta L_0 .

$$\varepsilon = \frac{L - L_0}{L_0}$$

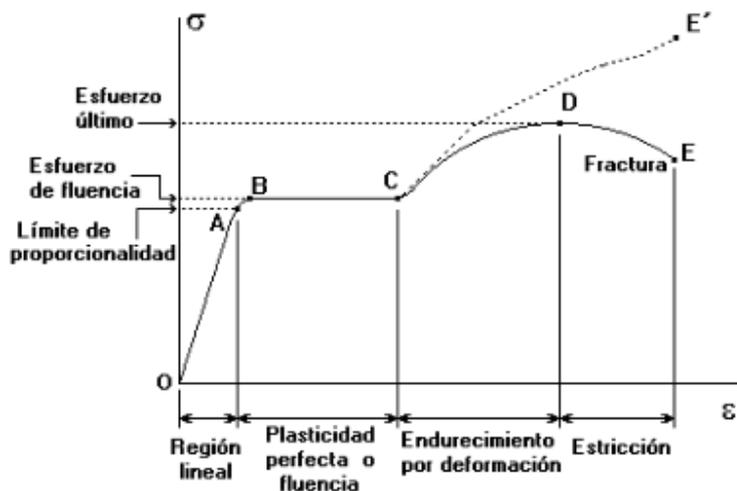


Figura 3.1.2 Gráfica tensión σ desplazamiento ε de un ensayo de tracción.

Como se observa en la imagen anterior es muy sencillo obtener de la gráfica una gran cantidad de información. Estos parámetros no se analizan en el proyecto, ya que no es información relevante.

3.2. Estructura del Banco.

Al igual que con los ensayos es imprescindible entender y analizar las partes de la estructura para tener una idea del movimiento y funcionamiento de la máquina.

Pese a tratarse de un antiguo bancos de ensayos, al revisar la estructura se observa que presenta grandes similitudes con los bancos actuales. De este modo se decide que no sería necesario realizar ningún cambio en la estructura.

Pero justo en el momento de abrir los planos de fabricación para observar las dimensiones del banco, se observa que la estructura de la máquina ha sufrido alguna pequeña modificación.

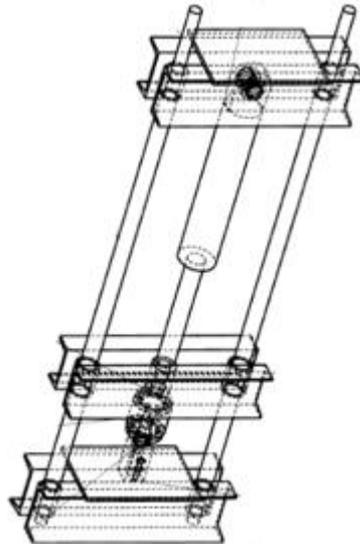


Figura 3.2.1 Plano 3D del antiguo banco de tracción.

3. Descripción del banco BTL-01

3.3. Sistema Eléctrico.

Una vez comprendida la estructura se pasa a analizar el funcionamiento de la instalación del laboratorio.

Como se ha comentado con anterioridad, la máquina se accionaba mecánicamente. Este accionamiento se realizaba a partir de 3 interruptores instalados en la parte delantera del cuadro eléctrico; avance-retroceso, marcha y paro.



Figura 3.3.1 Antiguo cuadro eléctrico del banco de ensayos.

Teniendo en cuenta el antiguo funcionamiento de la máquina, se observa al instante que será necesario realizar una remodelación en el cuadro eléctrico.

Antes de iniciar las modificaciones se realiza un análisis del antiguo esquema eléctrico para observar qué componentes se podrían reutilizar en la instalación, evitando así la compra de nuevos componentes.

3. Descripción del banco BTL-01

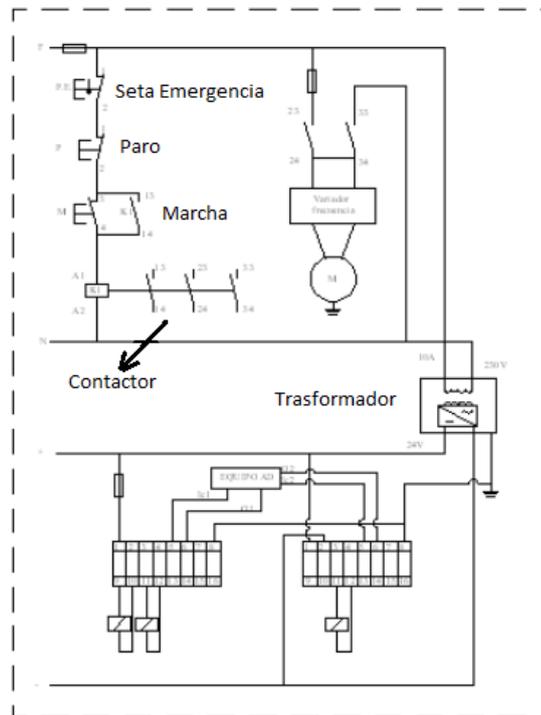


Figura 3.3.2 Antiguo esquema eléctrico del banco de ensayos.

Del esquema anterior se interpreta la siguiente información:

La corriente proveniente del enchufe, ya que el banco no dispone de ninguna toma auxiliar. Se alimenta directamente de la toma de corriente más cercana. Alimenta el transformador, el cual transforma los 230V en 24V.

Estos 24V son dirigidos a las electroválvulas, las cuales se encargan de controlar la presión y el movimiento en la instalación hidráulica.

Por otra parte tenemos la instalación del motor y los interruptores; como se observa en el esquema los 3 interruptores y el contactor se encuentran en serie. El interruptor de paro y emergencia se sitúan en la parte delantera de la instalación; lo que se consigue con esto es que presionando cualquiera de los dos, el circuito queda abierto inhabilitando el contactor y el interruptor de marcha.

A continuación se cablea el interruptor de marcha, el cual comparte una salida con el contactor; es decir, cuando se active un sentido, avance o retroceso, el interruptor cerrará el circuito conjuntamente con el contactor, encargado de encender el motor. Se observa también un variador de frecuencia el cual se encarga de variar los 10A a la frecuencia necesaria para poder activar el motor.

Al comprender el sistema se identifican rápidamente los elementos que pueden ser de utilidad en la modificación del mismo; transformador, seta de emergencia, contactor, y los interruptores de seguridad como el general o magnetotérmicos.

3. Descripción del banco BTL-01

3.4. Sistema Hidráulico.

En cuanto al sistema hidráulico no se facilita ningún plano ni esquema del circuito. De lo único de lo que se dispone es de las especificaciones técnicas.

- Especificaciones técnicas del banco:

Fuerza máxima: 68000N.

Recorrido máximo: 300 mm.

Célula de carga: U3 – 100kN.

Error de la célula de carga: < 10N

Sensor de presión: P8PA – 200

Motor ABB IP55 IEC34 con potencia 3,3 kW

Velocidad de desplazamiento de mordaza móvil: Regulable por programa, de 0 a 109 mm/min.

Velocidad de aplicación de la carga: Regulable por programa.

Equipo de adquisición: MGCPlus de HBM.

Software de adquisición: Aplicación propia desarrollada con Catman 4.0

Electroválvulas de Bucher hydraulics 5 vias.

- Probetas:

-Ancho máximo: 50 mm

-Longitud máxima: 240 mm

-Espesor máximo: 6 mm

- Tipos de ensayos:

-Tracción.

-Módulo elástico y coeficiente de Poisson.

-Compresión

- Tipos de material:

-Compuestos.

-Polímeros.

-Fundiciones.

-Aceros (con probetas de sección reducida).

3. Descripción del banco BTL-01

Para confirmar que no ha habido ninguna modificación en el sistema hidráulico se revisan los componentes hidráulicos para comprobar que concuerdan con los nombrados en la ficha técnica.

Una vez comprobados los componentes se puede afirmar que el banco no ha sufrido ninguna modificación en el sistema hidráulico y que se trata de un banco mono cilíndrico con accionamiento hidráulico.



Figura 3.4.1 Sistema hidráulico del banco de ensayos.

4. Modificaciones del banco BTL-01

Modificaciones del banco BTL-01

4.1. Estructura del Banco.

En cuanto a la estructura, la única tarea a realizar es la modificación de los antiguos planos de fabricación., debido a que la estructura actual del banco difiere en pequeña medida de la estructura mostrada en los antiguos planos.

Para realizar los nuevos planos se realiza un diseño 3D del banco con la ayuda del programa *SolidWorks*, a partir del cual obtendremos los nuevos planos.

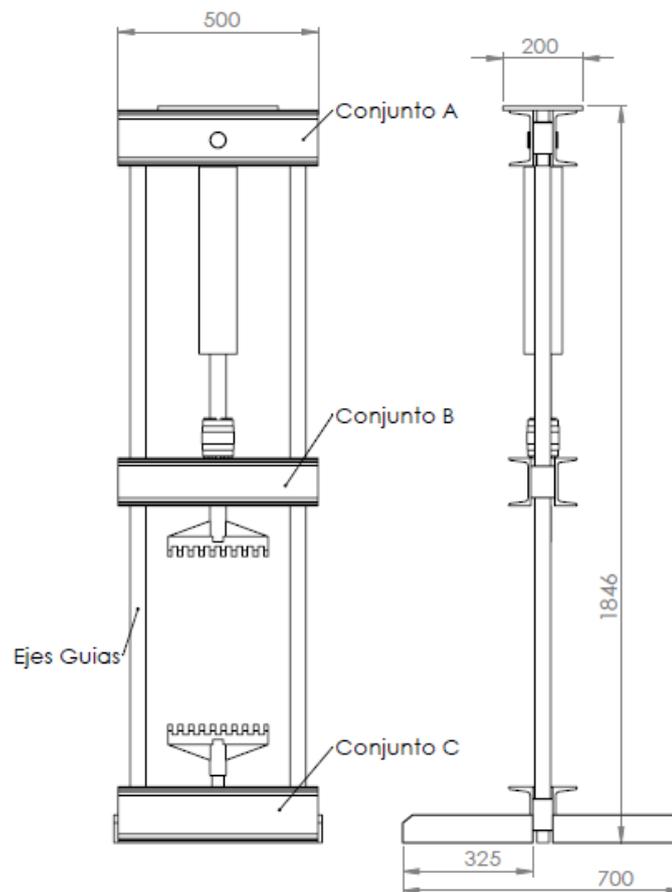


Figura 4.1.1 Planos de alzado y planta del banco de ensayos.

Como se observa en la anterior figura, en el banco existen unas pequeñas modificaciones; como el cambio de mordazas, la aparición de soportes en la parte inferior, la reducción de tamaño en los ejes, etc.

El banco presenta una estructura mono espacial, es decir, sólo puede realizar ensayos en una parcela de la máquina. Esto se debe a que el pistón hidráulico invade la otra parcela. Presenta un accionamiento hidráulico.

4. Modificación del banco BTL-01

Para entender un poco la estructura del banco se realiza un listado con las partes más importantes del banco.

-Travesero superior fijo: Aporta mayor rigidez al conjunto.

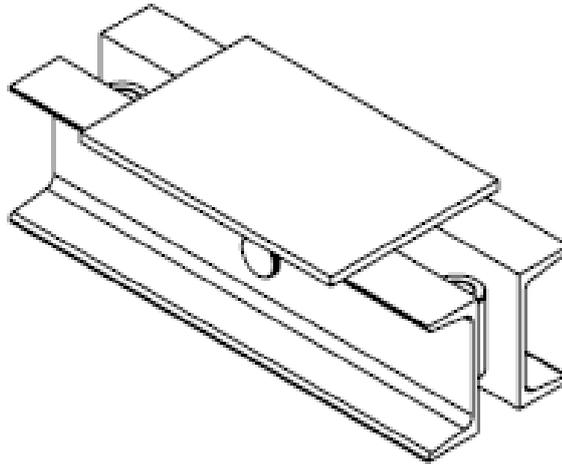


Figura 4.1.2 Conjunto A.

-Columnas: Aportan soporte y sirven de guía al travesero móvil.

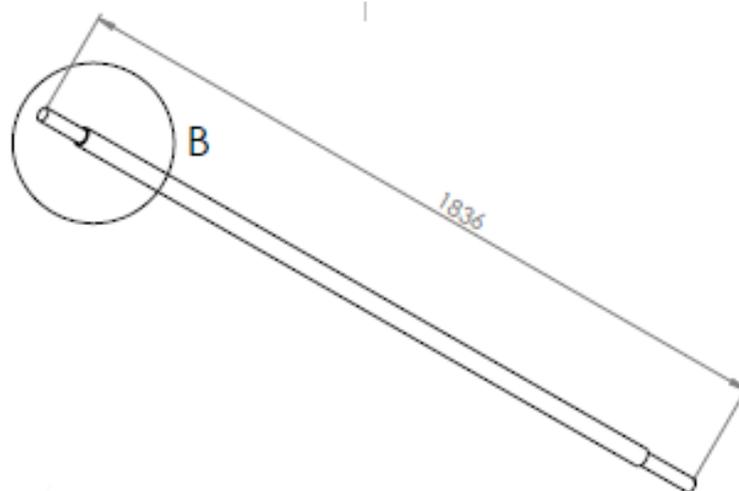


Figura 4.1.3 Eje guías.

4. Modificación del banco BTL-01

-Travesero móvil: Es el encargado de realizar el desplazamiento.

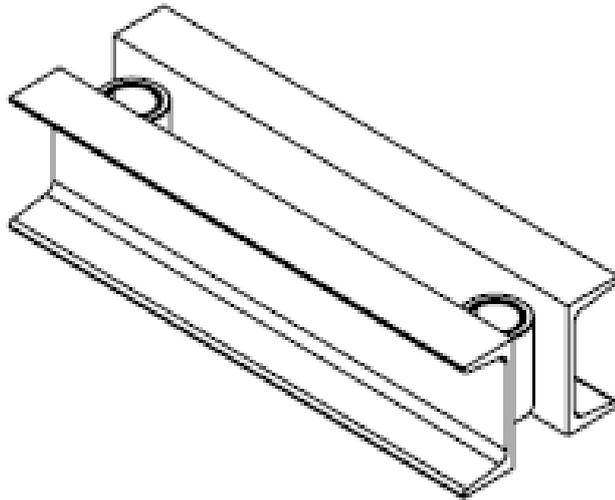


Figura 4.1.4 *Conjunto B.*

-Travesero inferior o base del banco: Es el encargado de soportar todas las cargas de la estructura y darle estabilidad al conjunto.

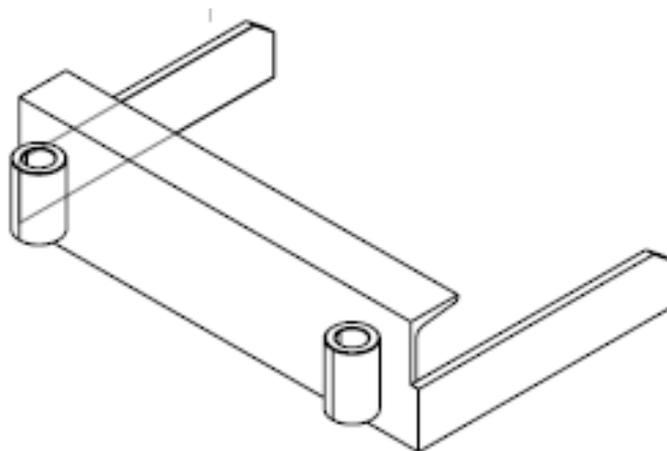


Figura 4.1.5 *Conjunto C.*

4. Modificación del banco BTL-01

-Pistón hidráulico: Es el encargado de transformar la energía del motor en desplazamiento.

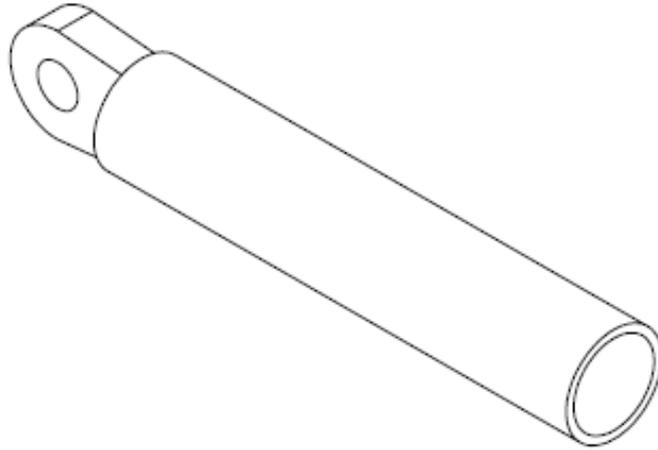


Figura 4.1.6 *Camisa del pistón hidráulico.*

-Mordazas: Son las encargadas de sujetar las probetas. Las mordazas pueden ser sustituidas según el ensayo.

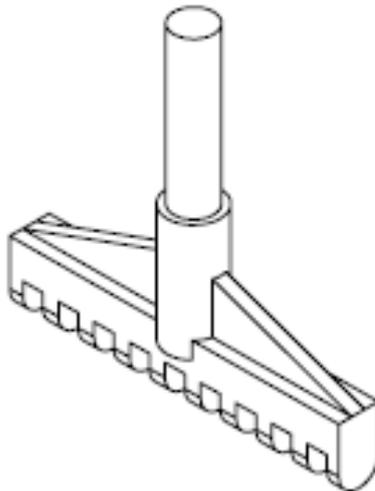


Figura 4.1.7 *Mordazas para realizar ensayos a tracción.*

4. Modificación del banco BTL-01

Una vez aclarada la estructura de la máquina es sencillo hacerse una idea del tipo de movimiento que se obtiene y cuáles son los elementos que propagan dicho desplazamiento.

Como se observa en la siguiente imagen el banco realiza un desplazamiento tanto positivo como negativo a lo largo del eje Y. Este desplazamiento viene dado por el sistema hidráulico que mueve el pistón encargado de desplazar el conjunto C a lo largo de las columnas.

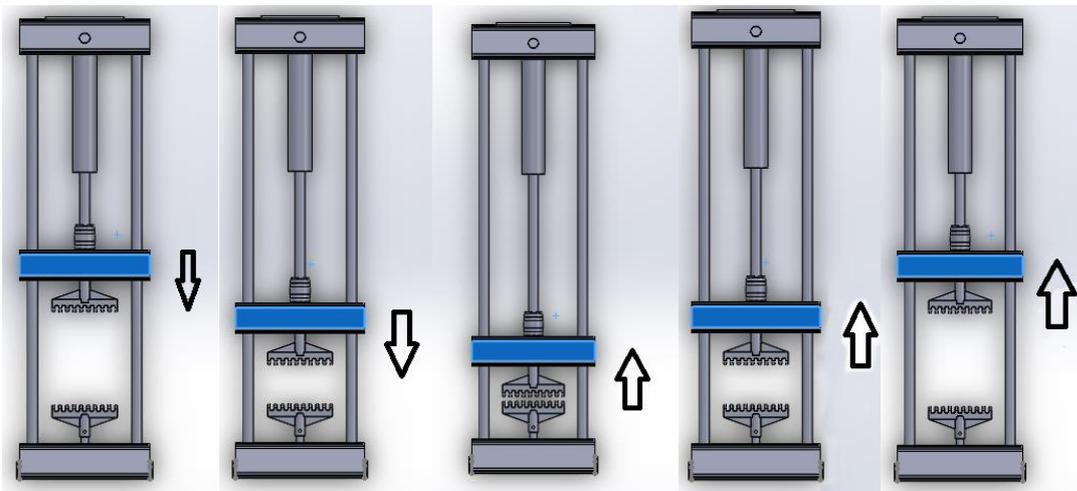


Figura 4.1.8 Esquema del desplazamiento en el eje Y del banco.

4.2. Sistema Eléctrico.

Como se ha comentado con anterioridad en el apartado 3.3 Sistema Eléctrico, es de interés intentar reutilizar todos aquellos componentes que puedan soportar las condiciones de nuestro nuevo sistema.

Para ello se realizan una serie de cálculos eléctricos acorde con el Reglamento Electrotécnico para Baja Tensión.

4.2.1. Cálculos y componentes

Antes de iniciar los cálculos se realiza un listado previo de los posibles componentes que serán imprescindibles para la nueva instalación.

- Arduino, alimentado con una fuente de alimentación de 12V.
- Motor 3,3kW y contactor.
- Transformador de 24V y puente rectificador.
- Componentes de seguridad. Interruptor general y magnetotérmicos.

Una vez realizado el listado, se extraen de la antigua instalación aquellos que puedan servir para la nueva, y el resto se dejan inhabilitados como por ejemplo los interruptores de avance-retroceso y marcha.



Figura 4.2.1.1 *Interruptores de avance, retroceso y marcha del antiguo cuadro eléctrico.*

A continuación se realiza un esquema, con todos aquellos componentes que consuman energía eléctrica para poder obtener los resultados acorde con la normativa y comprobar qué componentes son útiles para la nueva instalación o el tipo de cableado necesario.

4. Modificación del banco BTL-01

En primer lugar se identifican los componentes con consumo.

-Motor 3300W -Electroválvulas 3·30W -Arduino 12V · 0'5A ≈ 6W

En el caso del Arduino se obtiene un valor aproximado, ya que no se nos aporta el consumo. El valor del consumo es considerablemente bajo como para que afecte al cálculo de los resultados.

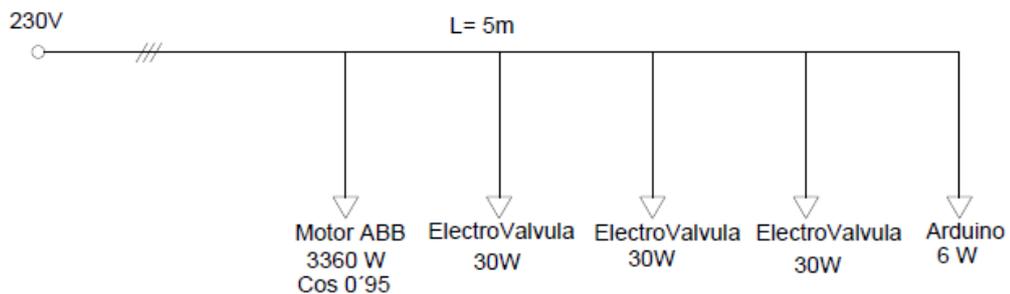


Figura 4.2.1.2 Esquema para el cálculo de potencias.

Los cálculos realizados son:

$$I = \frac{P \cdot 1'25}{\sqrt{3} \cdot V \cdot \cos\gamma} = \frac{3300 + 90 + 6}{\sqrt{3} \cdot 230 \cdot 0'95} = 11'22A$$

Acorde con el ITC-BT-19 la sección para cable de cobre necesaria sería de 2'5mm.

Para comprobar si la instalación es correcta se calcula la caída de tensión, la cual no debe superar los 3'7%.

$$\%V = \frac{100 \cdot P \cdot L}{C \cdot S \cdot V^2} = \frac{100 \cdot 3396 \cdot 5}{56 \cdot 2'5 \cdot 230^2} = 0'23\%$$

Una vez obtenidos los cálculos ya se puede pasar a realizar la instalación. Pero antes de ello se comprueba que todos los componentes de la antigua instalación soporten las nuevas condiciones. Y se realiza un listado más concreto de todos los componentes necesarios para la instalación.

4. Modificación del banco BTL-01

-Interruptor general Lovato D1 10A del antiguo cuadro:

Es el encargado de proteger al sistema de sobrecargas o cortocircuitos. Al realizar los cálculos nos damos cuenta que la protección de 10A es inadecuada para nuestro sistema, ya que al recorrer 11A por el mismo, el interruptor saltaría durante la realización de ensayos. Por tanto se decide cambiar el general y adquirir un nuevo interruptor, Lovato C16 16A.



Figura 4.2.1.3 Interruptor general Lovato C16 P1 MB

-Motor ABB IP55 IEC 34.

Encargado de transformar la energía eléctrica en energía mecánica para poner en funcionamiento la bomba del sistema hidráulico.



Figura 4.2.1.4 Motor hidráulico ABB IP55 3'3 kW.

4. Modificación del banco BTL-01

-Magnetotérmicos 1 ABB S201 C3 3A / 2 Schneider GB2-CB10 5A

Encargados de la protección del sistema a sobrecargas o cortocircuitos.



Figura 4.2.1.5 Interruptores magnetotérmicos ABB y Schneider de 3 y 5 Amperios.

-Contactor Lovato BF09 10D 9A 2,2kW del antiguo sistema:

Es el encargado de cortar o establecer corriente en los circuitos de potencia. En nuestro sistema es el encargado de dar marcha al motor. Al igual que con el interruptor general es necesario sustituir el contactor debido a las nuevas condiciones del sistema. Se sustituye por un nuevo contactor Lovato BG1201A230.



Figura 4.2.1.6 Contactor Lovato BG1201A230.

4. Modificación del banco BTL-01

-Fuente de alimentación AMR1-12.

Encargada de transformar los 230V en 12V y alimentar el Arduino.



Figura 4.2.1.7 Fuente de alimentación AMR1-12.

-Arduino Uno Wifi Shield / Arduino UNO 4 Relays Shield

Son microcontroladores encargados de realizar todo el control automático del sistema.



Figura 4.2.1.8 Arduino Uno Wifi Shield.

4. Modificación del banco BTL-01

-Transformador Siemens 4AM4095-4LA00-0N T50/B 0550 T1

Encargado de transformar los 230V del sistema en 24V. Se reutiliza del antiguo sistema.



Figura 4.2.1.9 Transformador siemens T50/B 0550 T1.

-Puente rectificador Semikron SKB 30/04 A1

Nos asegura unos 24V netos en todas las líneas salientes del transformador.

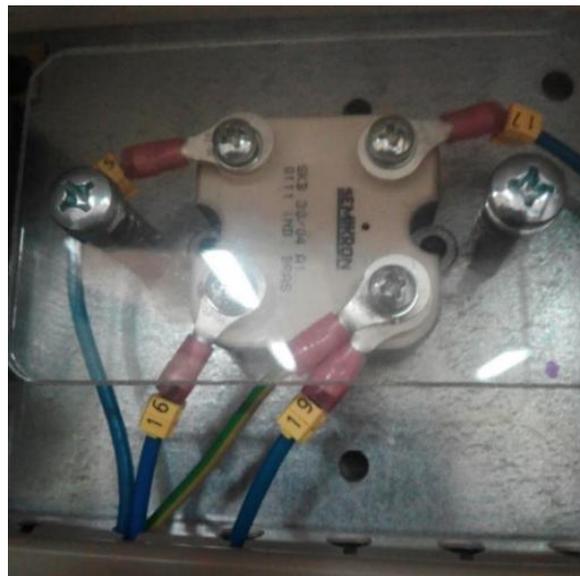


Figura 4.2.1.10 Puente rectificador Semikron SKB 30/04 A1.

4. Modificación del banco BTL-01

-Relé Finder 5V 30.22.7.005.0010

Son los actuadores del Arduino. Se encargan de recibir las órdenes del Arduino y actuar sobre el circuito eléctrico.



Figura 4.2.1.11 Relé Finder 30.22.7.005.0010

Una vez identificados todos los componentes necesarios para la instalación se pasa a realizar una pequeña explicación del sistema eléctrico, con la ayuda de los planos realizados.

4. Modificación del banco BTL-01

4.2.2. Instalación y planos

Para una correcta instalación se retiran todos los componentes del antiguo cuadro y se deja completamente vacío. Una vez vaciado se colocan de forma ordenada los componentes del nuevo sistema eléctrico y se realiza el cableado acorde a los siguientes planos.

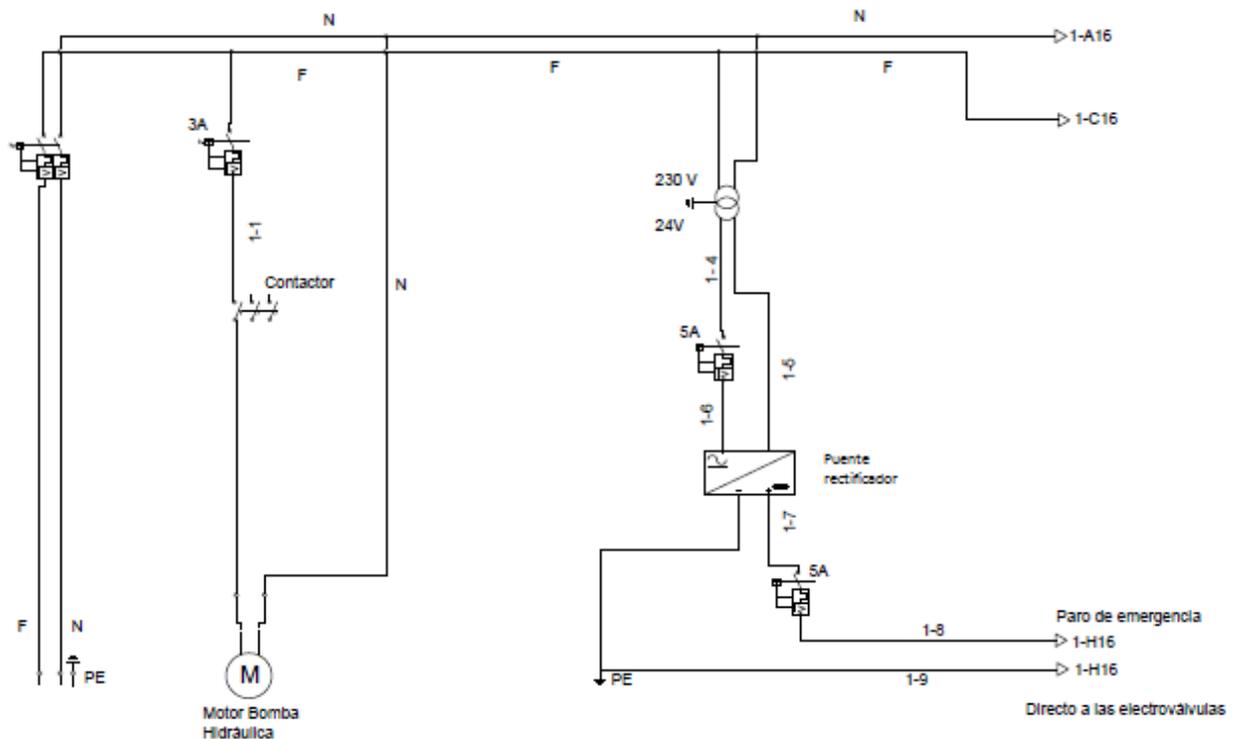


Figura 4.2.2.1 Plano potencia.

Debido a que el banco puede ser transportado a otros laboratorios se decide alimentar el cuadro mediante un cable directo a cualquier enchufe, en vez de realizar una toma auxiliar desde la instalación principal.

Una vez se alimenta el cuadro, la corriente pasa a través del general encargado de cortar la corriente en caso de sobretensiones. Lo ideal sería instalar un diferencial justo antes del general. Pero por el hecho de reutilizar los tres magnetotérmicos se obvia el primer diferencial y se colocan los 3 magnetotérmicos, uno a la entrada del contactor y los otros dos a la entrada y la salida del puente rectificador.

En cuanto la corriente supera el general la línea se divide en tres partes.

La primera línea cablea el contactor encargado de dar marcha al motor.

4. Modificación del banco BTL-01

La segunda es la encargada de transformar los 230V de la instalación en 24V, para ello se utiliza el transformador y el puente rectificador que nos asegura unos 24V netos. De esta línea se cablean los 1-H16 que se dirigirán uno a la seta de emergencia y el otro directo a la alimentación de las válvulas.

La tercera línea es la encargada de dar corriente al alimentador del Arduino 1-A16/1-C16. En la siguiente figura se muestra el transformador de 230V/12V que alimenta el Arduino UNO Wifi.

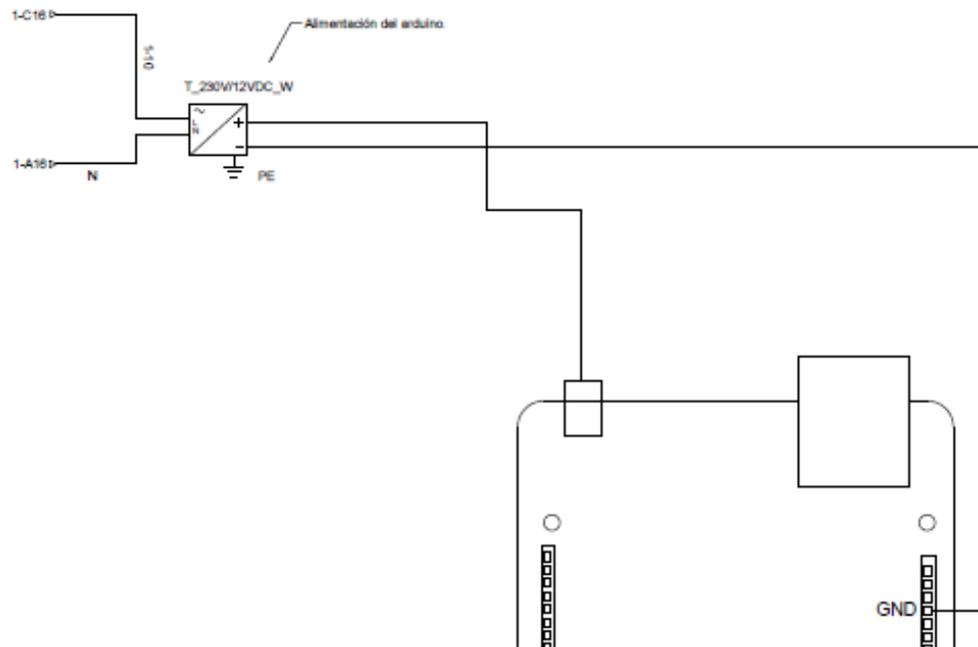


Figura 4.2.2.2 Plano alimentación Arduino.

Una vez alimentado el Arduino UNO Wifi se acopla una placa 4 Relays Shield. Esta placa será la encargada de controlar los relés que activan las válvulas. La incorporación de este tipo de placa es completamente necesaria, ya que el Arduino de lo único que se encarga es de cortar o suministrar corriente a 5V. Por ello se instala una placa con 4 relés capaces de actuar como interruptores con un flujo de corriente. El funcionamiento de los relés es simple cuando el Arduino recibe la señal de la consola suministra corriente por uno de los pines, dicha corriente es suministrada al relé el cual cambia de posición y cierra el circuito, en cuanto la corriente deja de suministrarse el relé vuelve a su estado inicial.

4. Modificación del banco BTL-01

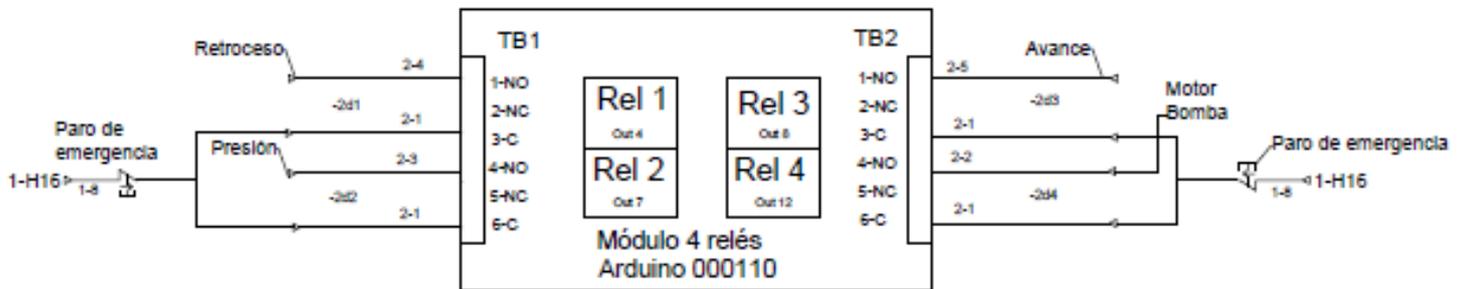


Figura 4.2.2.3 Plano control Arduino.

Como se observa en los planos, el cable 1-8 proveniente del puente rectificador en los planos de potencia, pasa por la seta de emergencia, y se convierte en el cable 2-1, el cual se conecta a la salida 3-C del Arduino. Con esto se consigue que al presionar la seta de emergencia se corte el circuito, quedando el control del Arduino inhabilitado y parando la máquina instantáneamente.

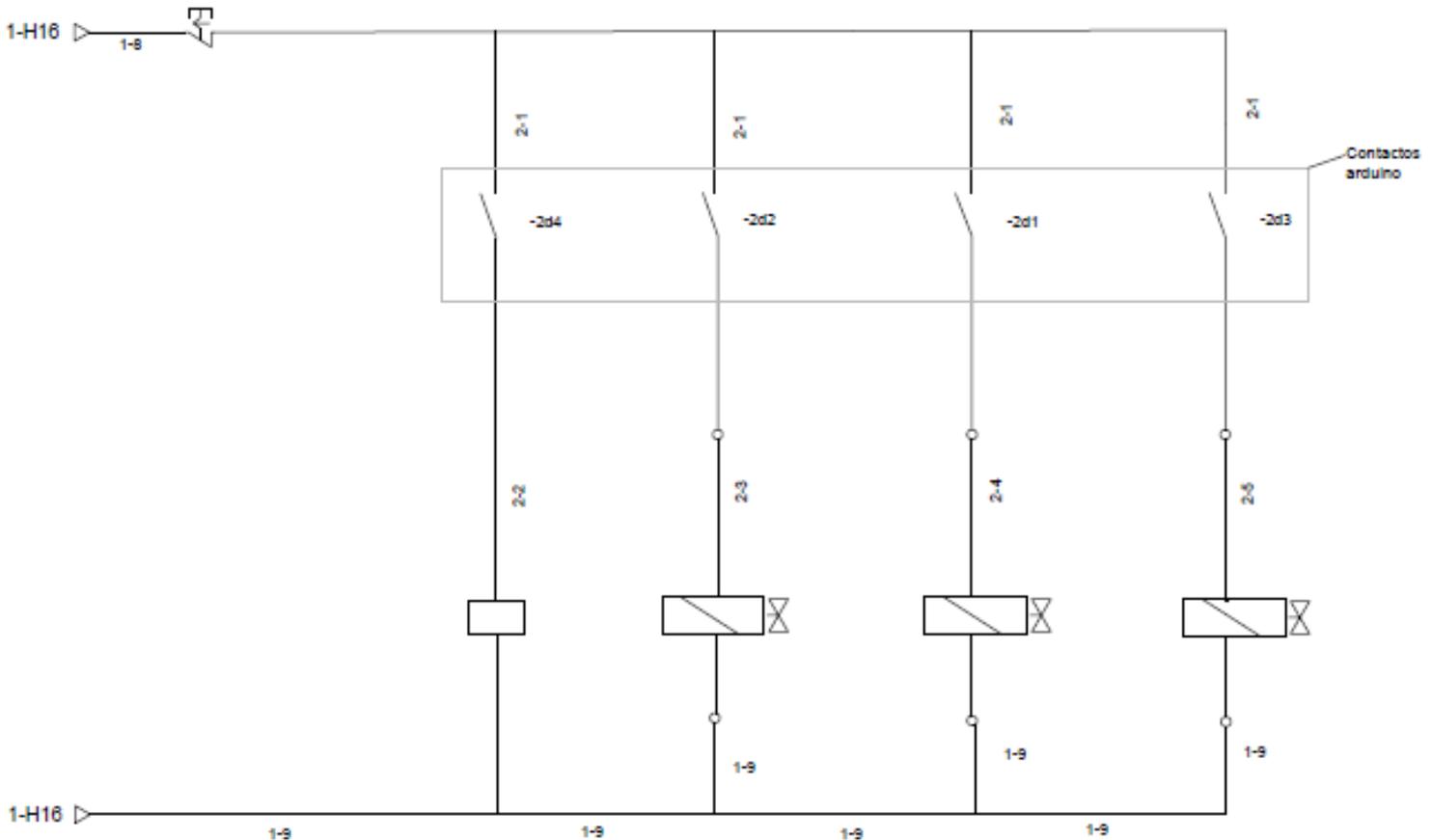


Figura 4.2.2.4 Plano control de válvulas.

4. Modificación del banco BTL-01

Para acabar de realizar la instalación se conectan las válvulas con el Arduino para ello se realiza el siguiente cableado:

- Cable 1-9 con 2-2, control motor.
- Cable 1-9 con 2-3, electro presión.
- Cable 1-9 con 2-4, electro retroceso.
- Cable 1-9 con 2-5, electro avance.

Con esto la instalación quedaría completamente definida dando forma al siguiente cuadro eléctrico.



Figura 4.2.2.5 Cuadro eléctrico actual.

Una vez realizada la instalación eléctrica ya se dispone de un banco completamente apto para poder ser controlado y automatizado.

4.3. Sistema Hidráulico.

Con la ayuda de las especificaciones técnicas y sin entrar en detalle de cálculos hidráulicos. Se realiza un pequeño esquema explicativo sobre el funcionamiento de la instalación.

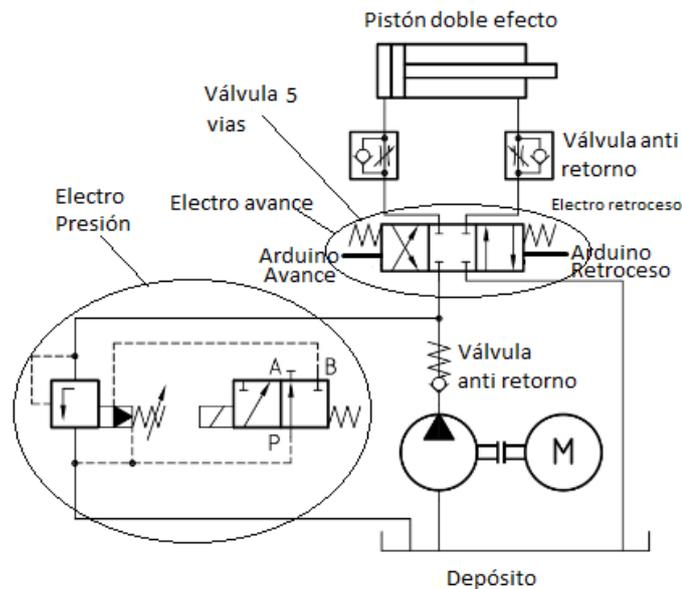


Figura 4.3.1 Esquema hidráulico del banco de ensayos.

Una vez el sistema eléctrico le da marcha al motor, el sistema hidráulico se pone en funcionamiento. El primer paso es la transformación de energía eléctrica del motor en energía hidráulica en la bomba. La bomba recoge la mezcla de agua y aceite del depósito y lo lanza hacia la válvula 5 vías.

El depósito se divide en dos partes, una en la cual la bomba recoge el aceite limpio pasando por un filtro, y la otra donde se recoge el aceite caliente y sucio proveniente del pistón.

Por otra parte, la válvula de electro presión es la encargada de regular la presión en el sistema, por ello siempre estará conectada ante cualquier desplazamiento del pistón.

Una vez introducido el aceite a presión en el circuito, es la válvula 5 vías la que se encarga de controlar el sentido de circulación. Como se observa en el esquema la posición central mantendrá el sistema en paro. Pero una vez el Arduino mande la señal, ya sea por la parte de avance o retroceso, la válvula cambiara la posición abriendo el circuito de avance o retroceso, dependiendo lo que nos interese.

Una vez elegido el sentido, el aceite entrará por una parte del pistón y saldrá por la otra en dirección al depósito. Las válvulas instaladas entre el pistón y la válvula 5 vías se tratan de válvulas anti retorno en ambos sentidos.

5. Diseño de la automática

Diseño de la automática

5. Diseño de la automática

Para el control del banco se hace uso del microcontrolador Arduino. La elección de este microcontrolador viene impulsada a que se trata de una plataforma con código abierto lo cual nos permite la creación de proyectos de forma rápida y sencilla, además presenta unas prestaciones muy similares a las de otros microcontroladores como PIC o ATMEL.

Arduino se basa en el lenguaje C++, este lenguaje de programación está asociado al sistema operativo UNIX.

El lenguaje requiere un nivel medio y trata tanto con caracteres como con números, bits, direcciones de memoria, etc.

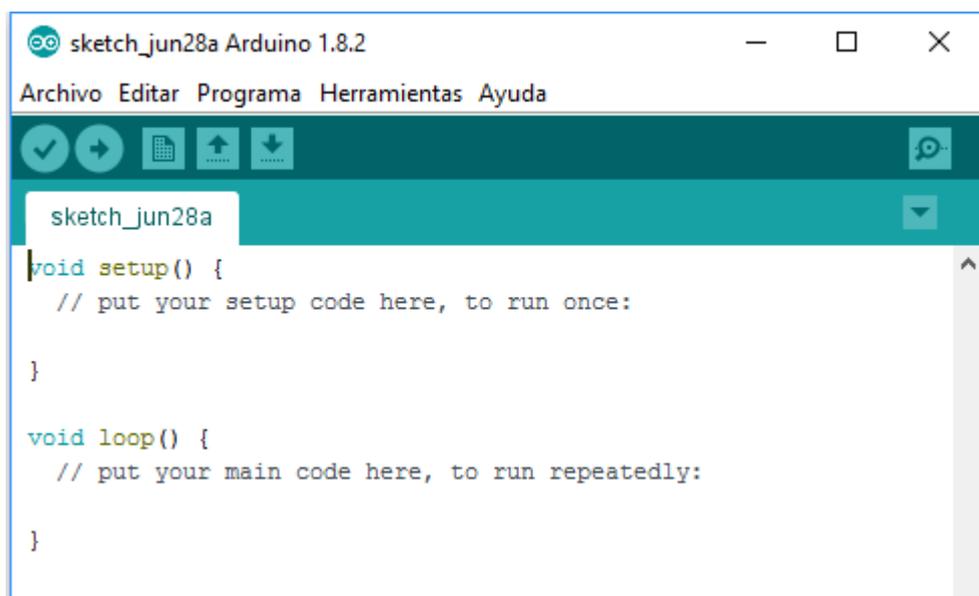
Arduino lo que ha conseguido es socializar el lenguaje mediante librerías, tutoriales y ayudas en su software, pero manteniendo también la posibilidad de programar mediante los comandos estándares de C++.

Para la programación en Arduino se dispone de un software libre denominado Aduino. Los programas en este software son denominados sketch o proyecto y se guardan con la extensión .ino.

Todos los proyectos de Arduino siguen una estructura base:

-*Void setup ()*: Setup es la función que llamará el Arduino cada vez que inicie o se realice un reset. Por ello es la estructura perfecta para introducir los pines de entrada o salida. Esta función solo es leída una vez por el Arduino una vez se inicie.

-*Void loop ()*: Una vez obtenidas las variables iniciales el *Void loop* lo que permite es poder comunicarse con el Arduino creando lazos o bucles que interactúen según nuestros comandos. Es la función perfecta para desarrollar el programa de control.



```
void setup() {  
    // put your setup code here, to run once:  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
}
```

Figura 5.1 Estructura base de un sketch de Arduino.

5. Diseño de la automática

Una vez comprendida la estructura base del programa se procede a la programación del banco de ensayos.

5.1. Programación.

En un primer momento la idea del proyecto era conseguir la automatización de la máquina mediante el ordenador más cercano a la máquina, encargado de recoger los datos de ensayo mediante el Catman 4.0.

Para ello se utilizó un Arduino UNO con una placa 4 Relays Shield incorporada conectada por vía serie al ordenador.



Figura 5.1.1 Arduino 4 Relays Shield

El primer paso antes de empezar con la programación es conseguir la forma adecuada de cargar los programas sin necesidad de estar conectando y desconectando el Arduino de la máquina. Para ello se ha de conectar un cable por puerto serie del ordenador al Arduino. Una vez detectado el Arduino en el apartado de *Herramientas, puerto*, se selecciona el COM adecuado.

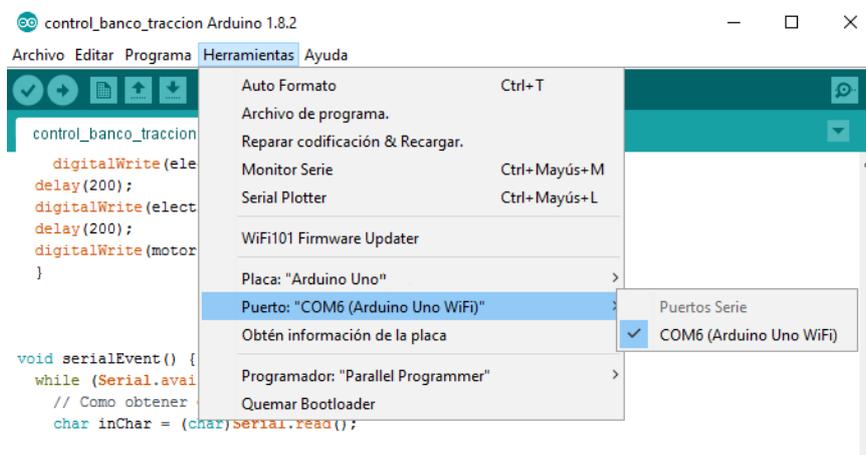


Figura 5.1.2 Conexión del Arduino mediante el puerto COM.

5.1.1. Programación vía serie

Para conseguir una correcta programación el primer paso que se realiza es una introducción de las variables y los pines de salida en el Arduino. Para ello se hace uso de la función *Setup* (). Antes de introducir las variables lo primero es identificar correctamente las salidas de los relés y qué controla cada salida.

-Relé 1 / Salida 4: Retroceso

-Relé 2 / Salida 7: Presión

-Relé 3 / Salida 8: Avance

-Relé 4 / Salida 12: Motor

Una vez identificada cada salida se pasa a programarlas en Arduino 1.8.2.

Para evitar errores con la numeración de las salidas se introducen unas constantes *const int*. Con esto lo que se consigue es ponerle un nombre a cada salida, de este modo el programa será más didáctico y fácil de leer por cualquier usuario.

Una vez introducidas las constantes se abre el *Void setup* () para introducir los pines *pinMode*. En este caso lo que se desea es controlar las salidas de las electroválvulas, por tanto en primera instancia se coloca el nombre de la electroválvula y luego se indica si es entrada *INPUT* o salida *OUTPUT*.

```
control_banco_traccion$  
  
// Definición de las salidas de las electroválvulas  
const int electro_presion = 7;  
const int electro_avance = 8;  
const int electro_retroceso = 4 ;  
const int motor = 12;  
|  
void setup() {  
// Definición del tipo de salidas  
pinMode(motor, OUTPUT);  
pinMode(electro_presion, OUTPUT);  
pinMode(electro_avance, OUTPUT);  
pinMode(electro_retroceso, OUTPUT);
```

Figura 5.1.1.1 Sketch, Introducción de las variables en Arduino.

Para conseguir una correcta comunicación entre el Arduino y la consola serie del programa es necesario integrar una serie de constantes y comandos específicos.

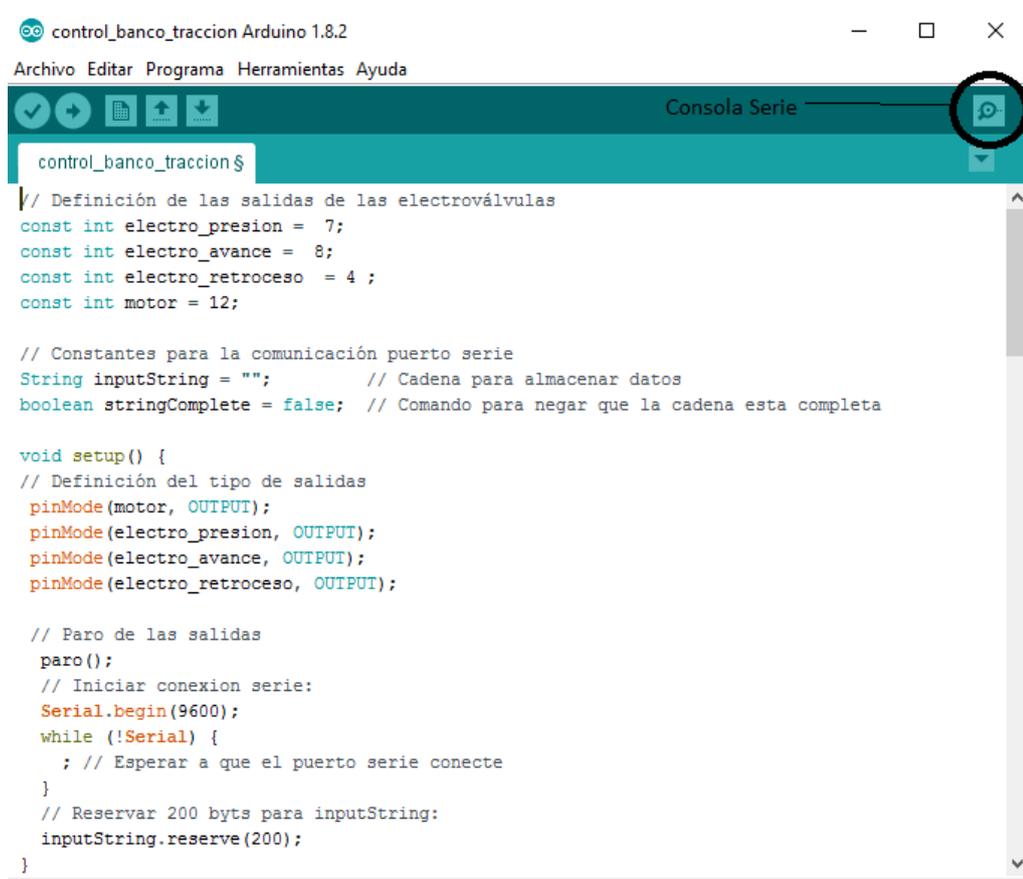
5. Diseño de la automática

Lo primero que se introduce son dos constantes. La primera un *String* llamado `inputString` el cual crea una constante capaz de almacenar datos. La segunda, *Boolean*, lo que realiza es verificar o negar si la cadena de comandos creada en el *String* está completa o no.

Una vez introducidas todas las constantes y pines, el Arduino debe iniciar la conexión entre él y la consola del ordenador, para ello, dentro del *Void setup* utilizamos el comando *Serial.begin*. Con este comando se consigue iniciar la conexión serie; en este punto el único problema es tener en cuenta la velocidad de transmisión de datos de la consola y el Arduino, para ello se fija la transmisión a 9600 baudios.

Finalmente se coloca un bucle *While* para que el Arduino no pare de buscar la consola mientras no establezca conexión y se reservan 200 bytes para la cadena.

La aparición *paro ()*: en la siguiente imagen servirá más adelante para la creación de un *Void* y facilitar la programación.



```
control_banco_traccion Arduino 1.8.2
Archivo  Editar  Programa  Herramientas  Ayuda
Consola Serie
control_banco_traccion $
// Definición de las salidas de las electroválvulas
const int electro_presion = 7;
const int electro_avance = 8;
const int electro_retroceso = 4 ;
const int motor = 12;

// Constantes para la comunicación puerto serie
String inputString = ""; // Cadena para almacenar datos
boolean stringComplete = false; // Comando para negar que la cadena esta completa

void setup() {
// Definición del tipo de salidas
pinMode(motor, OUTPUT);
pinMode(electro_presion, OUTPUT);
pinMode(electro_avance, OUTPUT);
pinMode(electro_retroceso, OUTPUT);

// Paro de las salidas
paro();
// Iniciar conexión serie:
Serial.begin(9600);
while (!Serial) {
; // Esperar a que el puerto serie conecte
}
// Reservar 200 bytes para inputString:
inputString.reserve(200);
}
```

Figura 5.1.1.2 Sketch, Introducción de los comandos y variables para la comunicación serie.

Con esto se obtiene la parte más sencilla de la programación.

Una vez finalizada la programación de las variables se continúa con la parte más complicada, la programación del *Void loop*.

Lo primero que se realiza es la búsqueda de los comandos necesarios para una correcta comunicación vía serie.

5. Diseño de la automática

Para ello dentro del *Void loop* () escribimos los siguientes comandos:

```
if (stringComplete) {  
    inputString.trim();  
    Serial.println(inputString);  
}
```

Con estos comandos lo que se consigue es comunicarle al Arduino que para una cadena completa de caracteres *stringComplete* él escriba el carácter que se desea *inputString*. Es decir, si se introduce en la consola la cadena de letras *marcha* el Arduino debe de reconocer la cadena y devolver la cadena de letras correspondiente *marcha motor*.

Una vez conseguido esto es más sencillo poder programar, ya que nombrando de distintas formas la cadena y condicionándola con las variables adecuadas se pueden obtener infinidad de combinaciones.

Pero en el momento de probar el programa aparece un grave problema. La cuestión es que el Arduino no está capacitado para entender las palabras como tal, es decir el Arduino se comunica con el Código ASCII.

Con este código se permitiría programar únicamente con letras sueltas o números dentro de los 255 caracteres del código. Por lo tanto cuando se introducía una cadena de letras el Arduino no reaccionaba o reaccionaba de forma incorrecta.

Para poder solucionar el problema es necesario realizar un nuevo *Void*.

El nuevo *Void* se nombra *serialEvent* (). Dentro de él se crea un bucle *while* al igual que en el *setup* para iniciar la conexión serie. Una vez realizado el bucle se crea una *array* tipo *char*.

Con esto lo que se consigue es almacenar los datos en forma de cadena de caracteres, es decir se consigue obtener un nuevo lenguaje apto para el *String*.

Para finalizar el *Void* y no tener problemas a la hora de enviar más de un comando a la vez, colocamos un salto de línea el cual reactivará el bucle siempre que se envíe un nuevo comando.

```
void serialEvent() {
  while (Serial.available()) {
    // Como obtener el nuevo lenguaje:
    char inChar = (char)Serial.read();

    // Si el comando nuevo es una nueva linea,
    //haz que el bucle principal haga algo al respecto
    if (inChar == '\n') {
      stringComplete = true;
    }
    else{
      // Si no agregalo a la cadena de entrada:
      inputString += inChar;
    }
  }
}
```

Figura 5.1.1.3 Sketch, Programación del Void serialEvent para la obtención un nuevo lenguaje.

Una vez conseguido el nuevo lenguaje es posible retomar el apartado de *Void loop*.

En este apartado se programan los movimientos del banco dependiendo del estado de las electroválvulas.

Para evitar la rotura de algún componente de la maquina debido a una mala programación, se analiza previamente el resultado de la activación de cualquiera de los pines.

-Salida 4: Al tratarse de la válvula de retroceso, lo que se crea en el sistema, al darle corriente, es un movimiento positivo en el eje Y de la máquina. Siempre y cuando el motor y la presión del sistema estén activados.

-Salida 7: Es el encargado de introducir la presión en el sistema. Por lo tanto al igual que el pin 12 debe estar conectado en todo momento para permitir los desplazamientos en el sistema.

-Salida 8: Se obtiene un comportamiento idéntico al del pin 4 pero con un movimiento negativo en el eje Y de la máquina.

-Salida 12: Es el pin más importante del sistema ya que sin la activación de este la máquina no realizaría ningún tipo de movimiento. Es el pin encargado de activar el motor hidráulico.

Al instante en que se identifican los pines ya se puede realizar el resto de la programación.

Para continuar la programación se usan los comandos *if* y *else if* junto a los comandos *digitalWrite* y *Serial.println*, con esto lo que se consigue es condicionar el programa y seleccionar que pines activar o desactivar.

5. Diseño de la automática

En el siguiente ejemplo se observa perfectamente el comportamiento del programa. En primer lugar se le indica al Arduino el comando que recibirá *marcha* y el pin que debe activar *motor*, *HIGH*, pintando en la consola serie *marcha maniobra*. En cambio si el comando recibido es *paro* realiza *paro ()* y pinta en la consola *paro banco*.

```
if (inputString == "marcha"){  
    digitalWrite (motor, HIGH);  
    Serial.println ("marcha maniobra");  
} else if (inputString == "paro"){  
    Paro ();  
    Serial.println ("paro banco");
```

Para este ejemplo aparece la referencia *paro ()*. Esto se debe a que por motivos de comodidad en la programación se ha decidido componer un *Void* aparte para el paro, y referenciarlo de esta forma. Es decir, no habría ningún inconveniente en eliminar el *Void* y colocar, donde aparece el comando *paro ()* todos los *digitalWrite*.

```
// Definición de la función de paro de la máquina  
void paro(){  
    digitalWrite(electro_presion, LOW);  
    delay(200);  
    digitalWrite(electro_avance, LOW);  
    delay(200);  
    digitalWrite(electro_retroceso, LOW);  
    delay(200);  
    digitalWrite(motor, LOW);
```

Figura 5.1.1.4 *Sketch, programación del Void paro.*

Finalmente se realiza el mismo procedimiento para el resto de movimientos a realizar. Lo que implica identificar qué tipos de movimientos serán necesarios para realizar un test de ensayo a tracción.

Lo primero y más importante será activar tanto el motor como la presión para cualquier tipo de movimiento, por lo tanto son dos pines que deben estar activos en todo momento.

Por otra parte, para realizar un test de ensayo a tracción, el programa debe permitir realizar avances y retrocesos a la hora de colocar las probetas.

Finalmente se debe colocar un programa mediante el cual la máquina empiece a realizar el ensayo al recibir el comando.

5. Diseño de la automática

Los comandos elegidos para identificar cada movimiento son:

Tabla 5.1.1 Comandos de control del Arduino

	Pines activos	Pines apagados
“Marcha”	12	8,4,7
“Presión”	7	8,4,12
“Avance”	7,8,12	4
“Retroceso”	7,4,12	8
“Test”	7,4,12	8
“Paro”	Ninguno	Todos

Para evitar problemas a la hora de enviar los comandos, se realiza una última línea de programación en el *Void loop*. La función de esta línea es reconocer cualquier comando que sea diferente a cualquiera de los anteriores y mostrarnos en la consola serie “comando no reconocido”.

```
    } else if (inputString == "retroceso"){
        digitalWrite(electro_avance, LOW);
        delay(100);
        digitalWrite(motor, HIGH);
        delay(100);
        digitalWrite(electro_retroceso, HIGH);
        delay(100);
        digitalWrite(electro_presion, HIGH);
        Serial.println("retroceso");
    } else if (inputString == "test"){
        digitalWrite(electro_avance, LOW);
        delay(100);
        digitalWrite(motor, HIGH);
        delay(100);
        digitalWrite(electro_retroceso, HIGH);
        delay(100);
        digitalWrite(electro_presion, HIGH);
        Serial.println("realizando ensayo");
    } else {
        Serial.println(inputString+" comando no reconocido");
    }
    // Limpia el string
    inputString = "";
    stringComplete = false;
}}
```

Figura 5.1.1.5 Sketch, programación de los comandos para el control del banco de ensayos.

Una vez terminada la programación se introduce el sketch en el Arduino y se comprueba que todos los comandos funcionen adecuadamente.

Para ello se conecta el Arduino vía serie al ordenador que controla los parámetros del banco, situado al lado de la máquina. En el momento en el que Arduino 1.8.2 detecta el

5. Diseño de la automática

Aunque se traten de Arduino diferentes, la programación de estos es muy similar. Por tanto se decide mantener el programa serie y modificarlo de forma que cambiando los comandos adecuados y reubicando algunos *Void*s el programa quede perfectamente válido para nuestro nuevo Arduino.

El primer paso es abrir un nuevo sketch y copiar las constantes y pines del programa serie. A diferencia de la comunicación serie, para comunicar por Wifi es necesario incluir librerías.

Estas librerías lo que permiten es poder programar de forma mucho más sencilla, ya que sin ellas la programación en Arduino se vería altamente modificada.

Antes de la aparición de los Arduino UNO con módulo Wifi incorporado, la programación básica para el Wifi se realizaba con comandos AT.

Estos comandos eran la única vía de obtener comunicación entre los Arduinos con módulos Wifi ESP8266 y pequeños servidores web a los cuales se accede mediante un IP en la cual aparecen los componentes que se programen con anterioridad en el sketch.

Actualmente con la incorporación de los módulos Wifi incorporados y la creación de librerías en el propio programa de Arduino, se puede programar de forma muy similar a cualquier otro Arduino.

Sabiendo esto, la única dificultad es elegir correctamente las librerías. En este caso las librerías elegidas son las siguientes.

```
#include <Wire.h>    #include <UnoWifiDevEd.h>
```

Una vez seleccionadas las librerías se procede a introducir los comandos necesarios para una comunicación vía Wifi.

Para ello se introduce el comando “WifiData” que permite nombrar la red “cliente” y se indica la velocidad de transmisión de datos 9800 Baudios.

Una vez hecho esto, dentro del *Void setup* se introducen los comandos equivalentes al *Serial.begin* que permitan iniciar la comunicación Wifi.

Por lo tanto, para el Wifi se coloca el nombre de la red seguido de *begin* y la velocidad a la que queremos que transmita.

```
BancoTraccion_FINAL_
//librerías y velocidad de transmisión
#include <Wire.h>
#include <UnoWiFiDevEd.h>
#define BAUD 9600

// indicar datos de la red
WifiData cliente;

// Definición de las salidas de las electroválvulas
const int electro_presion = 7;
const int electro_avance = 8;
const int electro_retroceso = 4 ;
const int motor = 12;

// Constantes para la comunicación vía wifi.
String inputString = ""; // Cadena para almacenar datos
boolean stringComplete = false; // Comando para negar que la cadena esta completa

void setup(){
// Definición del tipo de salidas
pinMode(motor, OUTPUT);
pinMode(electro_presion, OUTPUT);
pinMode(electro_avance, OUTPUT);
pinMode(electro_retroceso, OUTPUT);
  cliente.begin(BAUD); // Inicializar la conexion wifi
  inputString.reserve(200); // Reserva 200 bytes
}
```

Figura 5.1.2.1 Sketch, programación de librerías y comandos para la comunicación Wifi.

Una vez obtenidos los comandos que permitan la comunicación vía Wifi se modifica el resto del programa.

Para finalizar el programa bastaría con la sustitución de todos los comandos Serial por comandos cliente, manteniendo así el resto de la estructura.

Serial.println(inputString); → *cliente.println(inputString);*

Al modificar estos comandos se obtiene la estructura final del programa, el cual debería ser capaz de hacer funcionar el banco por la consola Wifi.

Pero al realizar unas pequeñas pruebas se observa que tanto el *Void paro* como el *Void serialEvent* separados por comodidad en el programa por serie, no son capaces de funcionar correctamente en el programa Wifi.

5. Diseño de la automática

Para solucionar el problema de forma rápida y sencilla se introducen los dos *Void* dentro del *Void loop* obligando así a que tengan que ser leídos por el Arduino.

```
void loop(){
//si existen datos disponibles los leemos
while (cliente.available()) {
  char inChar = cliente.read();    // obten los nuevos bytes

  // Si el comando nuevo es una nueva linea,
  //haz que el bucle principal haga algo al respecto
  if (inChar == '\n') {
    stringComplete = true;
  }
  else{
    // Si no agregalo a la cadena de entrada:
    inputString += inChar;
  }
}

//leemos la opción enviada por wifi.
if (stringComplete){
  inputString.trim();
  cliente.println(inputString);

  // Condicionantes para realizar la maniobra
  if (inputString == "marcha") {
    digitalWrite(motor, HIGH);
    cliente.println("Marcha maniobra");
  }

}

else if(inputString=="paro") {
  inputString="";
  digitalWrite(motor, LOW);
  delay(200);
  digitalWrite(electro_presion, LOW);
  delay(200);
  digitalWrite(electro_avance, LOW);
  delay(200);
  digitalWrite(electro_retroceso, LOW);
  cliente.println("paro máquina");
}
}
else if (inputString == "presion"){
  digitalWrite(electro_presion, HIGH);
}
```

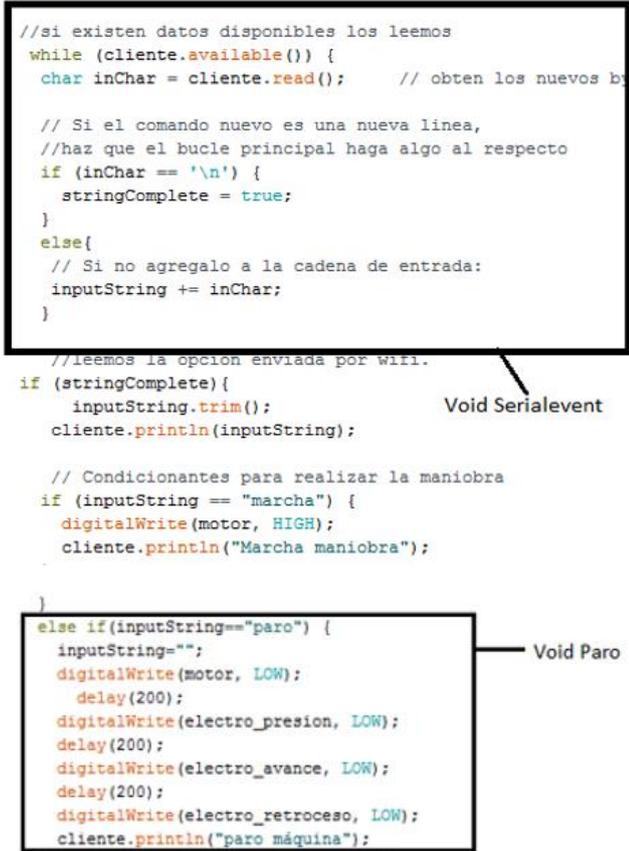


Figura 5.1.2.2 Sketch, recolocación del Void serialEvent y Void paro.

Finalmente con la recolocación de estos dos *Void* se obtiene el programa final adjunto en los anexos.

Una vez obtenido el programa, el siguiente paso es conseguir la comunicación entre el Arduino Wifi y los dispositivos que se deseen; como ordenadores, móviles, tabletas, etc.

6. Implementación

Implementación

6. Implementación

La implementación se entiende como la puesta en marcha de un proyecto, misión o idea programada, a partir de medidas o métodos.

A diferencia de la comunicación vía serie, la comunicación Wifi precisa de una serie de pasos para poder conectar el Arduino al dispositivo controlador.

Es cierto que este Arduino es capaz de funcionar por serie de la misma forma que el Arduino UNO, es decir el nuevo programa puede ser cargado por el puerto COM sin ningún problema. Pero este no funcionara de ninguna de las maneras además que lo que nos interesa es evitar tener conectado el dispositivo de control al Arduino instalado en el cuadro eléctrico.

6.1. Comunicación vía Wifi.

Para obtener una correcta comunicación vía Wifi el primer paso es entender el funcionamiento de estos dispositivos.

Las placas Wifi tienen la capacidad de crear una red Wifi a la cual se puede acceder desde cualquier dispositivo Wifi. Esta red lo único que nos permite es entrar a un servidor web base creado por Arduino y enlazar nuestro Arduino Wifi a la red doméstica o en el caso del proyecto a la red de la universidad a partir de unos sencillos pasos.

Para realizar una clara explicación del proceso de conexión via Wifi, se utiliza un ordenador portátil conectado a una red Wifi domestica TP-link.

6.1.1. Conexión en red doméstica.

Como se ha indicado con anterioridad el primer paso es conectar el ordenador a la red Wifi del Arduino.

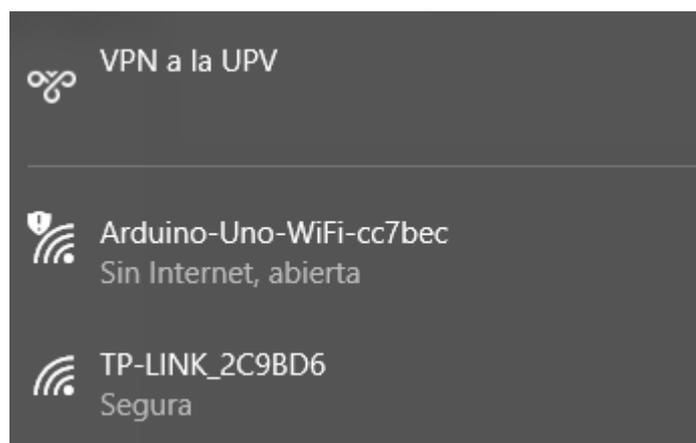


Figura 6.1.1.1 *Conexión a la red Wifi de Arduino.*

6. Implementación

Una vez conectados a la red Wifi de Arduino se busca la puerta de enlace predeterminada. Normalmente todos los Arduinos tienen como puerta de enlace la IP 192.168.240.1. Esto se debe a que Arduino dispone de una plataforma web en esta dirección IP que permite la modificación de ciertos parámetros de nuestro Arduino al acceder a ella.

Puerta de enlace predeterminada:	192.168.240.1
Servidores DNS:	192.168.240.1
Nombre de dominio DNS:	
Sufijo de conexión de DNS:	
Lista de sufijos de búsqueda DNS:	
Nombre de red:	Arduino-Uno-WiFi-cc7bec

Figura 6.1.1.2 Obtención de la puerta de enlace predeterminada.

Al acceder a la plataforma se observa que podemos cambiar tanto el nombre del Arduino como a la red a la que se conecte. Con esto lo que se consigue es modificar la red del Arduino y redirigirla a la red a la cual conectaremos todos los dispositivos de control.

Una vez introducida la red y la clave la web nos proporciona una nueva IP a partir de la cual podremos controlar nuestro Arduino desde cualquier dispositivo conectado a esta misma red.

The screenshot displays the Arduino web interface with three main sections:

- HOSTNAME:** A text input field containing "DavidRodTFG" and a "CHANGE" button below it.
- WIFI CONNECTION:** A section with instructions: "If you are in the same network, go to [192.168.1.17](#), else connect to network TP-LINK_2C9BD6 first." Below this is a note: "To connect to a WiFi network, please select one of the detected networks, enter the password, and hit the connect button...". A red box highlights the IP address "192.168.1.17", with an arrow pointing to it from the text "Nueva IP".
- WIFI STATUS:** A table showing the current WiFi configuration:

WiFi channel	1
Configured network	TP-LINK_2C9BD6
WiFi status	got IP address
WiFi address	192.168.1.17
WiFi rssi	-41dB
WiFi phy	11n
WiFi MAC	18:fe:34:cc:7b:ec
WiFi mode	AP+STA

A "SWITCH TO STA MODE" button is located at the bottom of this section.

Figura 6.1.1.3 Obtención de la nueva IP doméstica.

6. Implementación

Una vez obtenida la nueva IP se desconecta el ordenador de la red Arduino y se conecta a la red doméstica TP-link.

Lo que se consigue al realizar esto es poder tener el Arduino alimentado en cualquier parte de la red doméstica donde reciba conexión a la red TP-Link y acceder a él desde otra habitación o incluso desde fuera de la vivienda siempre y cuando el controlador también pueda acceder a la misma red.

Una vez realizada la prueba en una red doméstica se realiza el mismo procedimiento en la universidad. Pero antes de ello para evitar la rotura de cualquier componente de la máquina se comprueba por la consola Wifi, que aparece en la nueva IP, si nuestro programa reacciona adecuadamente a los comandos programados.

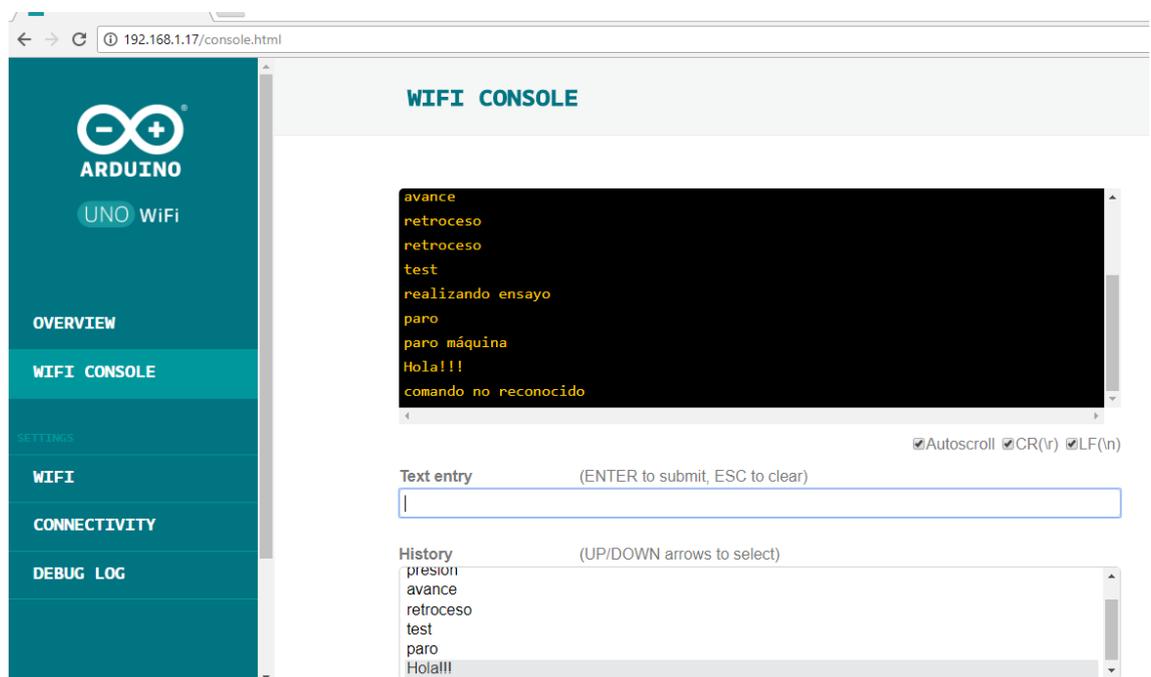


Figura 6.1.1.4 Comprobación del sketch por la consola Wifi de Arduino.

Como se observa en la imagen el Arduino responde perfectamente a nuestros comandos lo que implica un gran avance en el proyecto. Ya que con esto se puede lograr un control del banco a distancia.

Además el programa Arduino una vez se conectan el Arduino y el controlador, con el programa instalado, a la misma red detecta esta conexión y ofrece la opción de conectarse por puerto red pudiendo transmitir nuevos sketches al Arduino sin la necesidad del cable por el puerto COM.

6. Implementación

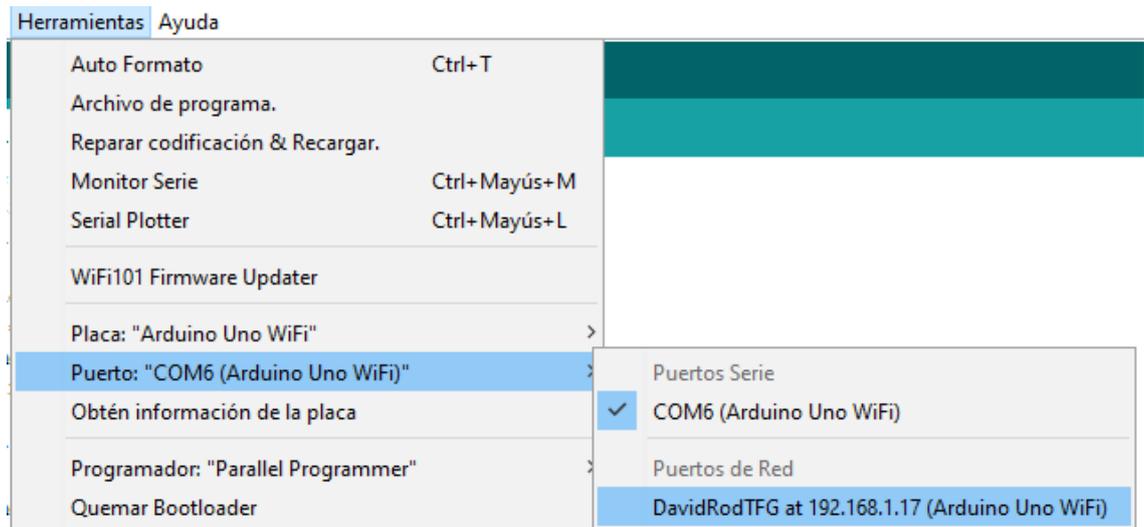


Figura 6.1.1.5 Conexión del programa Arduino 1.8.2 por puerto red.

6.1.2. Conexión en la red universitaria.

Para conectar el Arduino a la red universitaria lo primero es seleccionar la red de la universidad a la que se conectara el Arduino. La red seleccionada es la red UPVNET, la selección de esta red se debe a que es la red interna de la universidad con más rango.

Al realizar los pasos anteriores para la conexión de la red aparece un grave problema; el Arduino no es capaz de conectarse a la red UPVNET. Esto se debe a que el cifrado de seguridad de las redes de la universidad tienen un nivel demasiado alto y no permiten la conexión de Arduinos u otro tipo de aparatos capaces de conectarse a redes con cifrados de seguridad más bajos como las redes domésticas.

Llegados a este punto no cabe la posibilidad de abandonar el proyecto y dejar al banco sin conexión Wifi. Por ello se investiga un poco sobre la conexión de Arduinos en la Universidad politécnica de valencia y se descubre que se puede habilitar una red Wifi con validación MAC dentro de la red UPVNET.

Es decir, de la red Wifi UPVNET se obtiene una red secundaria, sin acceso a internet, a la cual se puede acceder únicamente con una clave específica adjunta a la MAC de cada aparato. La posibilidad de conectar aparatos a esta red se debe a que el cifrado de seguridad de la nueva red presenta unos requisitos mucho más bajos.

Una vez solucionado el problema se envía la MAC a los informáticos de la Universidad Politécnica de valencia, los cuales crean una red llamada FATIGA a la cual solo se puede acceder desde la primera planta del edificio Ferrándiz donde se encuentra el laboratorio, siempre y cuando se acceda con la contraseña validada a través de la MAC del aparato.

6. Implementación

SSID:	FATIGA
Protocolo:	802.11n
Tipo de seguridad:	WPA2-Personal
Banda de red:	2.4 GHz
Canal de red:	11
Dirección IPv4:	172.18.228.12
Fabricante:	Qualcomm Atheros Communications Inc.
Descripción:	Qualcomm Atheros AR928X Wireless Network Adapter
Versión del controlador:	3.0.2.202
Dirección física (MAC):	00-26-5E-FD-83-A0

Figura 6.1.2.1 *Propiedades de la red FATIGA.*

Al obtener la red ya se puede realizar el mismo protocolo de conexión que con la red doméstica. Esto será posible debido a que la protección de la red FATIGA es de tipo WPA2-Personal capaz de ser descifrada con la contraseña adjunta a la MAC del arduino.

En primer lugar se conecta el controlador, ordenador portátil, móvil o tableta, a la red Wifi del Arduino para obtener su puerta de enlace.

Una vez obtenido el puerto de enlace se accede al servidor web de Arduino donde se procede al cambio de nombre del microcontrolador y su conexión a la nueva red FATIGA.

Automáticamente se obtiene una nueva dirección IP a la cual podemos acceder desde cualquier dispositivo conectado a la red UPVNET.

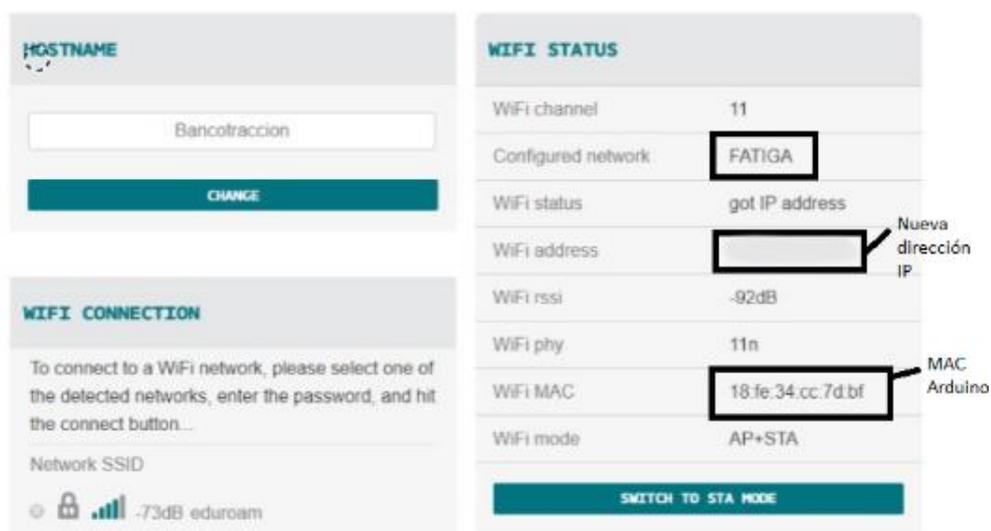


Figura 6.1.1.5 *Conexión del Arduino con la red FATIGA.*

6. Implementación

Finalmente con la obtención de la nueva dirección IP se tiene acceso directo, desde cualquier rincón de la universidad con acceso a la UPVNET, al completo control del banco de ensayos a tracción, desde la consola Wifi de la página web que aparece al introducir la nueva IP.

Ahora si se puede afirmar que se ha obtenido una completa automatización y control remoto del banco

Por motivos de seguridad no se muestra la nueva IP ya que cualquiera que supiera de la conexión del Arduino en la red FATIGA con introducir esta IP en su dispositivo móvil, Tablet u ordenador sería capaz de controlar el banco desde cualquier rincón de la universidad donde llegue la red UPVNET.

7. Pruebas y Resultados

Pruebas y Resultados

7. Pruebas y resultados

Para la obtención de los resultados es necesario realizar una serie de pruebas, a través de las cuales se comprobaba si realmente el resultado obtenido es el esperado.

La primera prueba consistirá en la comprobación del movimiento de la maquina a través de los comandos, tanto por dispositivo móvil como por ordenador. Para ello se instalan los dispositivos en la esquina opuesta al banco del laboratorio y conectan a la red UPV. Una vez conectados se accede a la IP del Arduino donde aparece la siguiente consola.

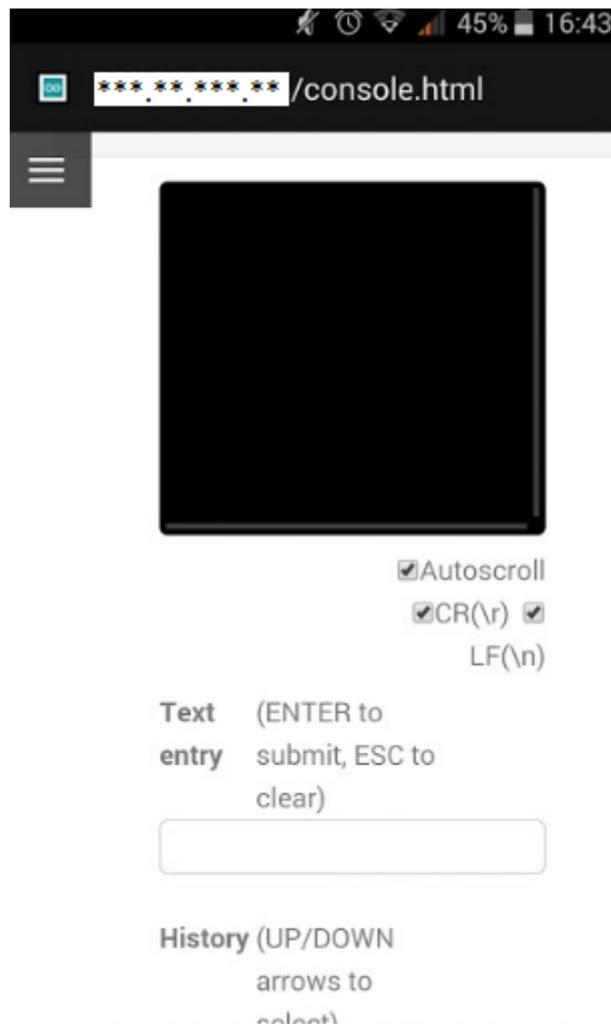


Figura 7.1 Consola Wifi desde dispositivo móvil.

Tanto en el móvil como en el ordenador aparece la misma consola. A continuación se introduce un primer comando, el comando elegido es *marcha* ya que este comando lo único que activa es el motor el cual realiza un ruido considerable, apreciable desde cualquier parte del laboratorio. Al introducir el comando se obtiene una respuesta instantánea del motor y se recibe en la consola la devolución del comando *marcha maniobra*.

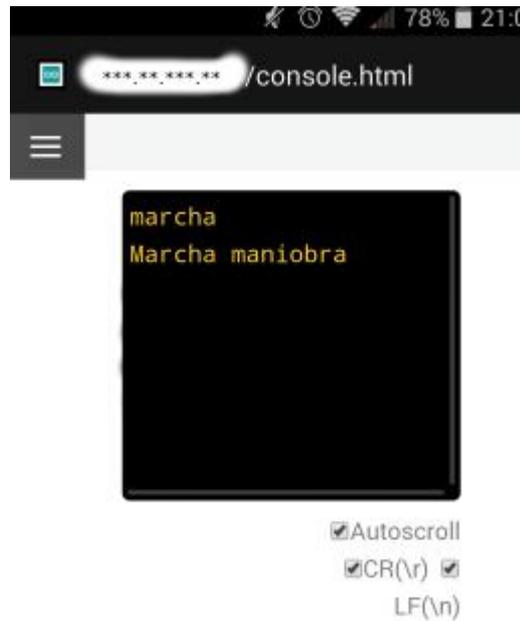


Figura 7.2 Devolución del comando *marcha* en la consola Wifi.

Tras obtener un resultado satisfactorio con el comando *marcha* se procede a introducir el resto de comandos de forma alternativa para comprobar que la maquina reacciona correctamente ante cualquier comando.

Los resultados obtenidos del control remoto hasta el momento en el laboratorio son realmente satisfactorios, ya que la maquina reacciona y realiza los movimientos al instante.

A continuación, una vez comprobados los movimientos se realizar una serie de pruebas con ensayos reales. Para comprobar si el banco además de realizar los movimientos es capaz de realizar ensayos correctamente con el comando *test*.

Para esta prueba se utilizaran dos probetas facilitas de antiguos ensayos y se comprobara que los datos obtenidos por Catman, permiten crear la curva tensión-deformación unitaria característica de los ensayos a tracción.

7. Pruebas y resultados

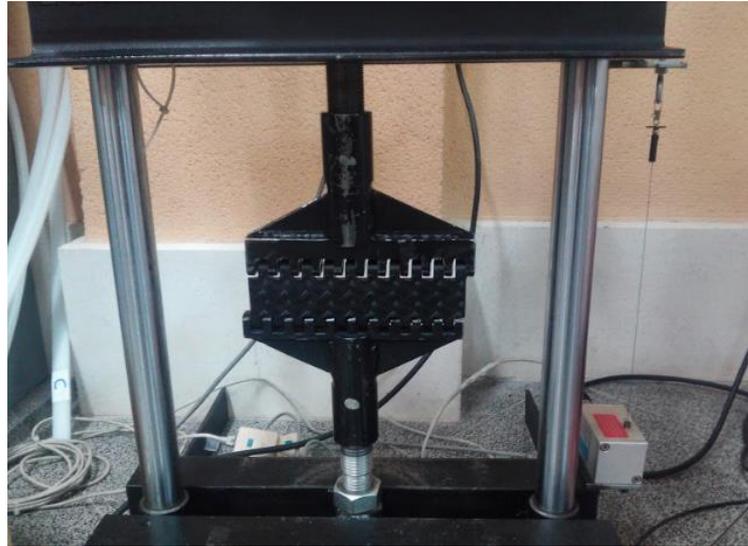


Figura 7.3 *Ensayo a tracción. Rotura un eslabón.*



Figura 7.4 *Ensayo a tracción. Rotura conjunto en cadena dos eslabones.*

Al realizar los ensayos obtenemos dos gráficas en el Catman una de la fuerza realizada en cada momento del ensayo y otra de la deformación. Con los datos obtenidos y la aplicación de las fórmulas explicadas en el punto 3.1 se obtiene una gráfica tensión-deformación unitaria para cada uno de los ensayos comprobando así que la máquina mantiene las mismas capacidades para realizar ensayos.

7. Pruebas y resultados

Una vez comprobado el perfecto funcionamiento del banco ante ensayos reales, se realiza una última prueba.

En esta prueba lo que se pretende es comprobar el alcance real del control remoto.

Para ello se realizara la misma prueba que en el apartado uno pero situando el controlador, dispositivo móvil u ordenador portátil, en diferentes lugares de la universidad.

Debido a las grandes distancias entre el controlador y el banco es imposible para una sola persona poder comprobar al instante si se están recibiendo los comandos correctamente. Para ello se necesita la ayuda de un compañero, el cual permanece en el laboratorio durante las pruebas para comprobar si el banco realiza los movimientos al instante de enviar los comandos.

Al realizar la prueba se obtienen los siguientes resultados:

Tabla 7.1 Resultados obtenidos de la comunicación entre el banco y los controladores

	Teléfono móvil	Ordenador portátil
Laboratorio F1L5	✓	✓
Aula F1A2	✓	✓
Aula F2A1	✓	✓
Aula F3A3	✓	✓
Laboratorio C1L7	✓	✓
Reprografía	X	X
Biblioteca EPSA	✓	✓
Parking EPSA	X	X
Patio EPSA	✓	✓
Puente VPN desde domicilio	X	X

Como se observa en la tabla los únicos lugares en los que no se permitía el control del banco era en los lugares en los cuales no se detectaba correctamente la red UPVNET.

Por lo que se puede afirmar que el banco puede ser controlado desde cualquier rincón de la universidad donde se tenga acceso a la red UPVNET.

Para finalizar con el proyecto se intentó el control del banco realizando un puente VPN desde el domicilio. Pero una vez conectados en la red y colocar la IP nos aparece un error de conexión y no nos permite acceder a la IP.

Por otra parte para culminar el trabajo se pensó en la instalación de una cámara Wifi mediante la cual poder observar el banco las 24h. Ya que esto nos permitiría poder ver en todo momento el punto en el que se encuentra el ensayo y poder realizar cambios a distancia en el momento deseado. Pero no fue posible la instalación por motivos en el protocolo de seguridad en la red interna de la universidad.

8. Estudio económico

Estudio económico

8. Estudio económico

En el estudio económico se ha realizado un análisis de todos los componentes adquiridos para el proyecto, además de introducir las horas empleadas en él.

Para ello se realiza un pequeño presupuesto.

Tabla 8. *Presupuesto del Proyecto.*

Presupuesto			
Descripción Componentes	Precio	Cantidad	Total
Arduino UNO	25,41	1	25,41 €
Arduino Wifi	48,28	1	48,28 €
Fuente de alimentación AMR1-12	28,04	1	28,04 €
Contactador Lovato BG1201A230	27,67	1	27,67 €
Interruptor General Lovato C16 2P	4,34	1	4,34 €
Arduino 4 Relays Shield	21,95	1	21,95 €
Elaboracion Planos y Programación	20,00	150	3.000,00 €
Total Fungible			155,69 €
Total mano de obra			3.000,00 €
Total			3.155,69 €

9. Bibliografía

Bibliografía

9. Bibliografía

Diego Moñux Chércoles. “Historia de la Automática Una introducción al estudio de los autómatas y el control desde la historia de la tecnología”. 2001

<http://isa.uniovi.es/~gojea/funding/documentos/historia%20automatica.pdf>

PA Gallego Botero, C Claros, R Ferney - 2007 - repositorio.utp.edu.co

<http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/1113/621815G166dm.pdf?sequence=1>

FESTO, “Fundamentos de la técnica de automática”, Libro técnico.< https://lehrerfortbildung-bw.de/u_matnatech/nwt/gym/weiteres/fb1/atechnik/grundlagen/es/kapitel/563062_Fundamentos_de_la_tecnica_de_automatizacion.pdf>

<http://www.arduino.org/>

<https://www.arduino.cc/>

<http://www.zwick.es/>

<http://www.ibertest.es/>

<http://www.bucherhydraulics.com/>

<http://diymakers.es/>

<https://www.luisllamas.es/>

<https://openhomeautomation.net/>

<http://panamahitek.com/>

<http://www.areatecnologia.com/> [1]

<http://factorelectrico.blogspot.com.es/2014/02/diferencia-entre-baja-media-y-alta.html>

<http://www.monografias.com/trabajos73/sistema-electrico/sistema-electrico.shtml#sistemaela>

<http://assets.mheducation.es/bcv/guide/capitulo/844814807X.pdf>

Anexos

Anexos

Anexo 1: Programa comunicación Wifi

```

//librerias y velocidad de transmisión
#include <Wire.h>
#include <UnoWiFiDevEd.h>
#define BAUD 9600

// indicar datos de la red
WifiData cliente;

// Definición de las salidas de las electroválvulas
const int electro_presion = 7;
const int electro_avance = 8;
const int electro_retroceso = 4 ;
const int motor = 12;

// Constantes para la comunicación vía wifi.
String inputString = "";    // Cadena para almacenar datos
boolean stringComplete = false; // Comando para negar que la cadena esta completa

void setup(){
// Definición del tipo de salidas
pinMode(motor, OUTPUT);
pinMode(electro_presion, OUTPUT);
pinMode(electro_avance, OUTPUT);
pinMode(electro_retroceso, OUTPUT);
cliente.begin(BAUD);        // Inicializar la conexion wifi
inputString.reserve(200);    // Reserva 200 byts
}

void loop(){

//si existen datos disponibles los leemos

```

```

while (cliente.available()) {
    char inChar = cliente.read(); // obten los nuevos byts

    // Si el comando nuevo es una nueva linea,
    //haz que el bucle principal haga algo al respecto
    if (inChar == '\n') {
        stringComplete = true;
    }
    else{
        // Si no agregalo a la cadena de entrada:
        inputString += inChar;
    }

    //leemos la opcion enviada por wifi.
    if (stringComplete){
        inputString.trim();
        cliente.println(inputString);

        // Condicionantes para realizar la maniobra
        if (inputString == "marcha") {
            digitalWrite(motor, HIGH);
            cliente.println("Marcha maniobra");
        }
        else if(inputString=="paro") {
            inputString="";
            digitalWrite(motor, LOW);
            delay(200);
            digitalWrite(electro_presion, LOW);
            delay(200);
            digitalWrite(electro_avance, LOW);
            delay(200);
        }
    }
}

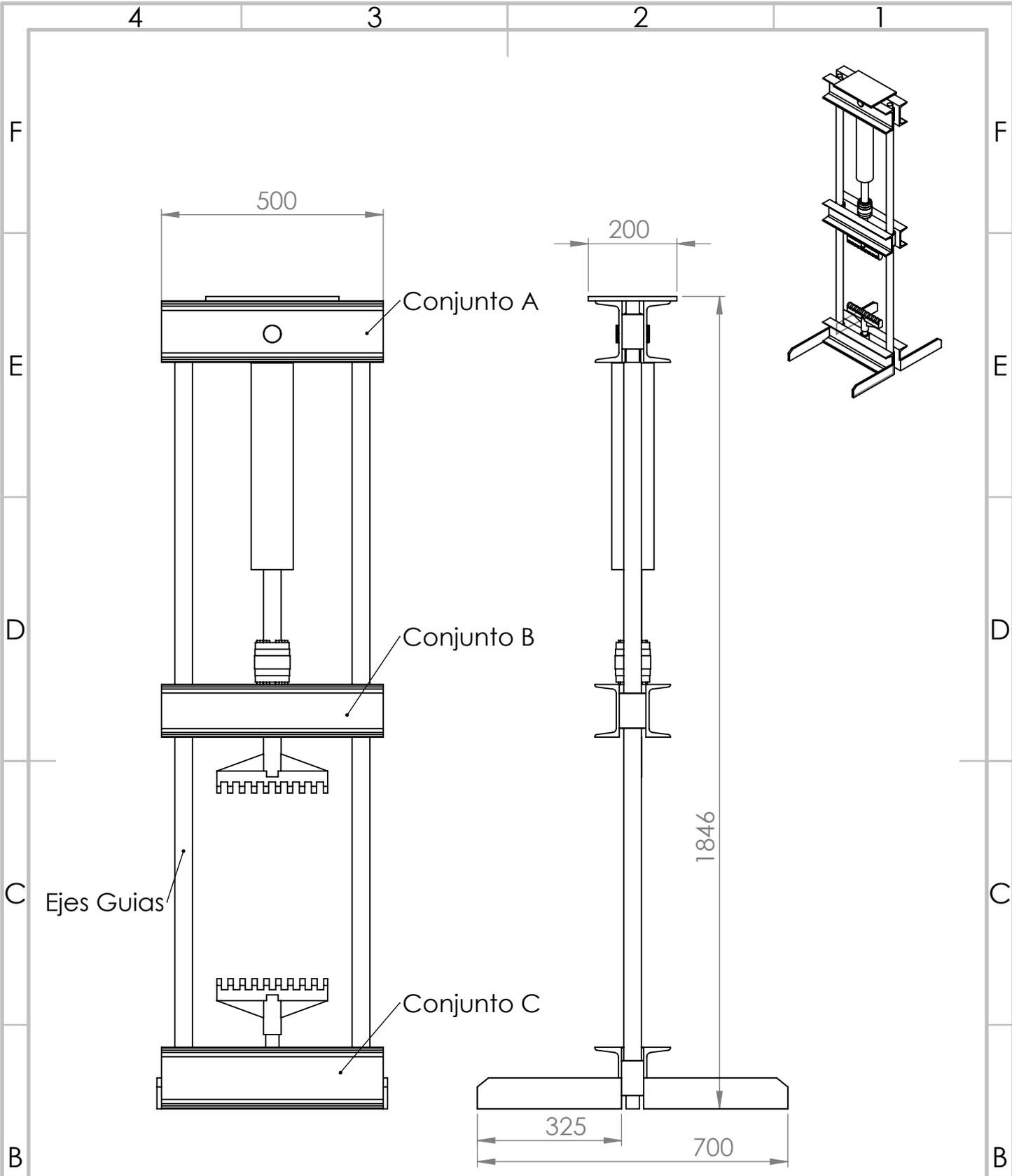
```

```
digitalWrite(electro_retroceso, LOW);
cliente.println("paro máquina");
}else if (inputString == "presion"){
digitalWrite(electro_presion, HIGH);
cliente.println("marcha presion");

}else if (inputString == "avance"){
digitalWrite(electro_retroceso, LOW);
delay(100);
digitalWrite(motor, HIGH);
delay(100);
digitalWrite(electro_avance, HIGH);
delay(100);
digitalWrite(electro_presion, HIGH);
cliente.println("avance");
} else if (inputString == "retroceso"){
digitalWrite(electro_avance, LOW);
delay(100);
digitalWrite(motor, HIGH);
delay(100);
digitalWrite(electro_retroceso, HIGH);
delay(100);
digitalWrite(electro_presion, HIGH);
cliente.println("retroceso");
} else if (inputString == "test"){
digitalWrite(electro_avance, LOW);
delay(100);
digitalWrite(motor, HIGH);
delay(100);
digitalWrite(electro_retroceso, HIGH);
delay(100);
```

```
digitalWrite(electro_presion, HIGH);
cliente.println("realizando ensayo");
} else{
    cliente.println("comando no reconocido");
}
    inputString = "";
    stringComplete = false;
}
}
}
```

Anexo 2: Planos de fabricación



SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:		ACABADO:	REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA	REVISIÓN																		
<table border="1"> <tr> <td>NOMBRE</td> <td>FIRMA</td> <td>FECHA</td> </tr> <tr> <td>DIBUJ.</td> <td></td> <td></td> </tr> <tr> <td>VERIF.</td> <td></td> <td></td> </tr> <tr> <td>APROB.</td> <td></td> <td></td> </tr> <tr> <td>FABR.</td> <td></td> <td></td> </tr> <tr> <td>CALID.</td> <td></td> <td></td> </tr> </table>			NOMBRE	FIRMA	FECHA	DIBUJ.			VERIF.			APROB.			FABR.			CALID.			TÍTULO:		
NOMBRE	FIRMA	FECHA																					
DIBUJ.																							
VERIF.																							
APROB.																							
FABR.																							
CALID.																							
Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.			N.º DE DIBUJO MaquinaTraccion ^{A4}																				
PESO:			ESCALA:1:50		HOJA 1 DE 1																		

4 3 2 1

F
E
D
C
B
A

F
E
D
C
B
A

4 3 2 1

4 3 2 1

F

F

E

E

D

D

C

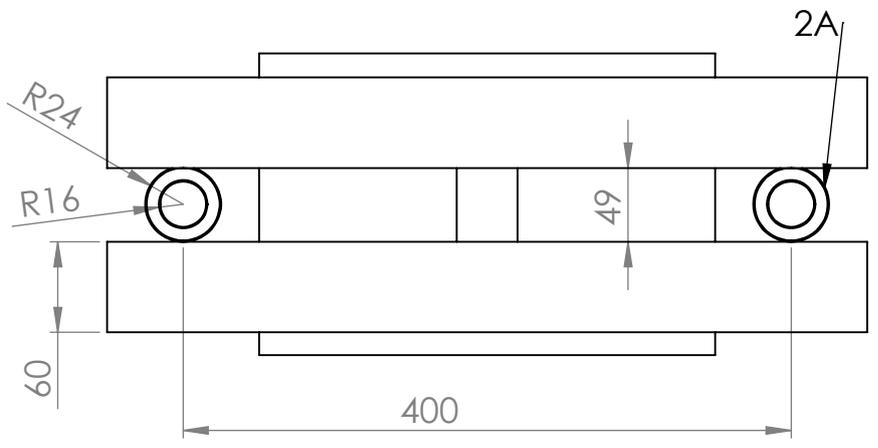
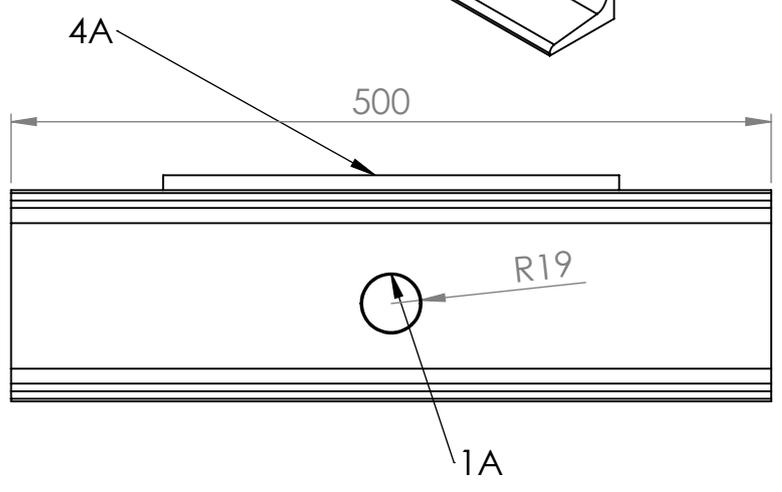
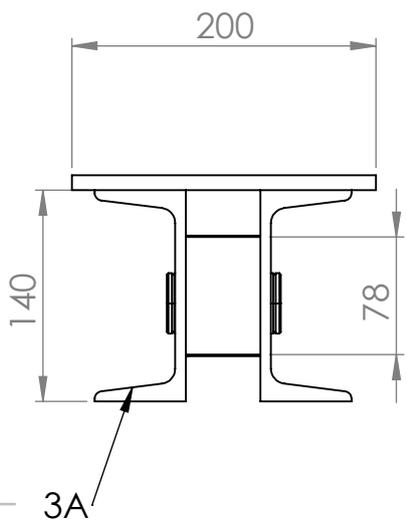
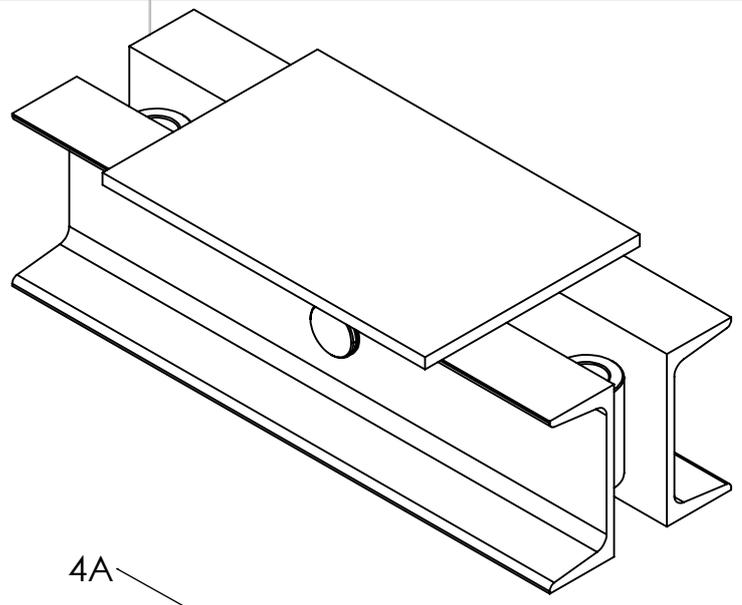
C

B

B

A

A



SI NO SE INDICA LO CONTRARIO:
 LAS COTAS SE EXPRESAN EN MM
 ACABADO SUPERFICIAL:
 TOLERANCIAS:
 LINEAL:
 ANGULAR:

ACABADO:

REBARBAR Y
 ROMPER ARISTAS
 VIVAS

NO CAMBIE LA ESCALA

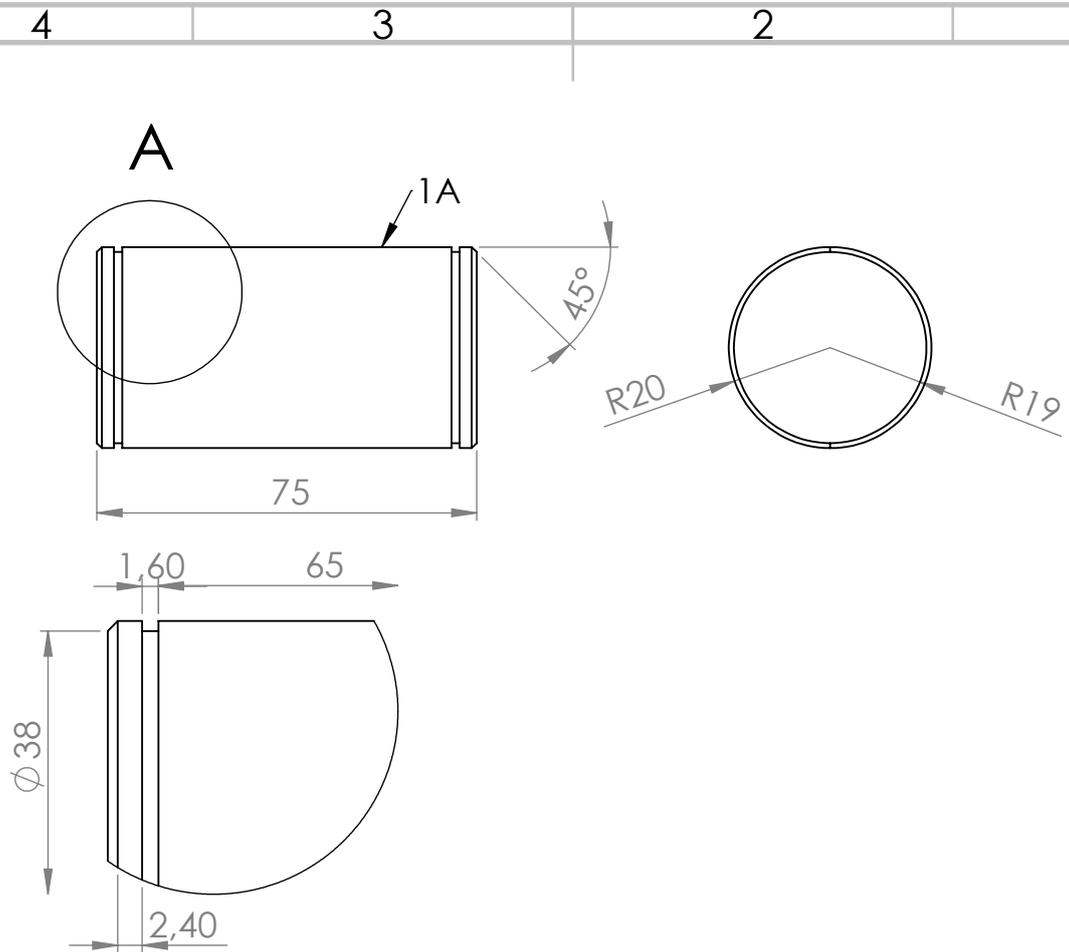
REVISIÓN

	NOMBRE	FIRMA	FECHA
DIBUJ.			
VERIF.			
APROB.			
FABR.			
CALID.			

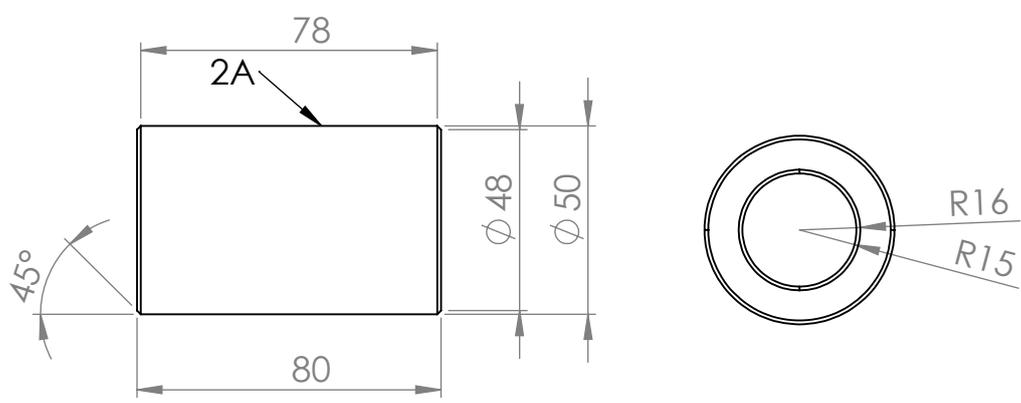
TÍTULO:	
<h1>Conjunto A</h1>	
N.º DE DIBUJO	A4
<h1>Conjunto A</h1>	
PESO:	ESCALA: 1:10
	HOJA 1 DE 1

Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

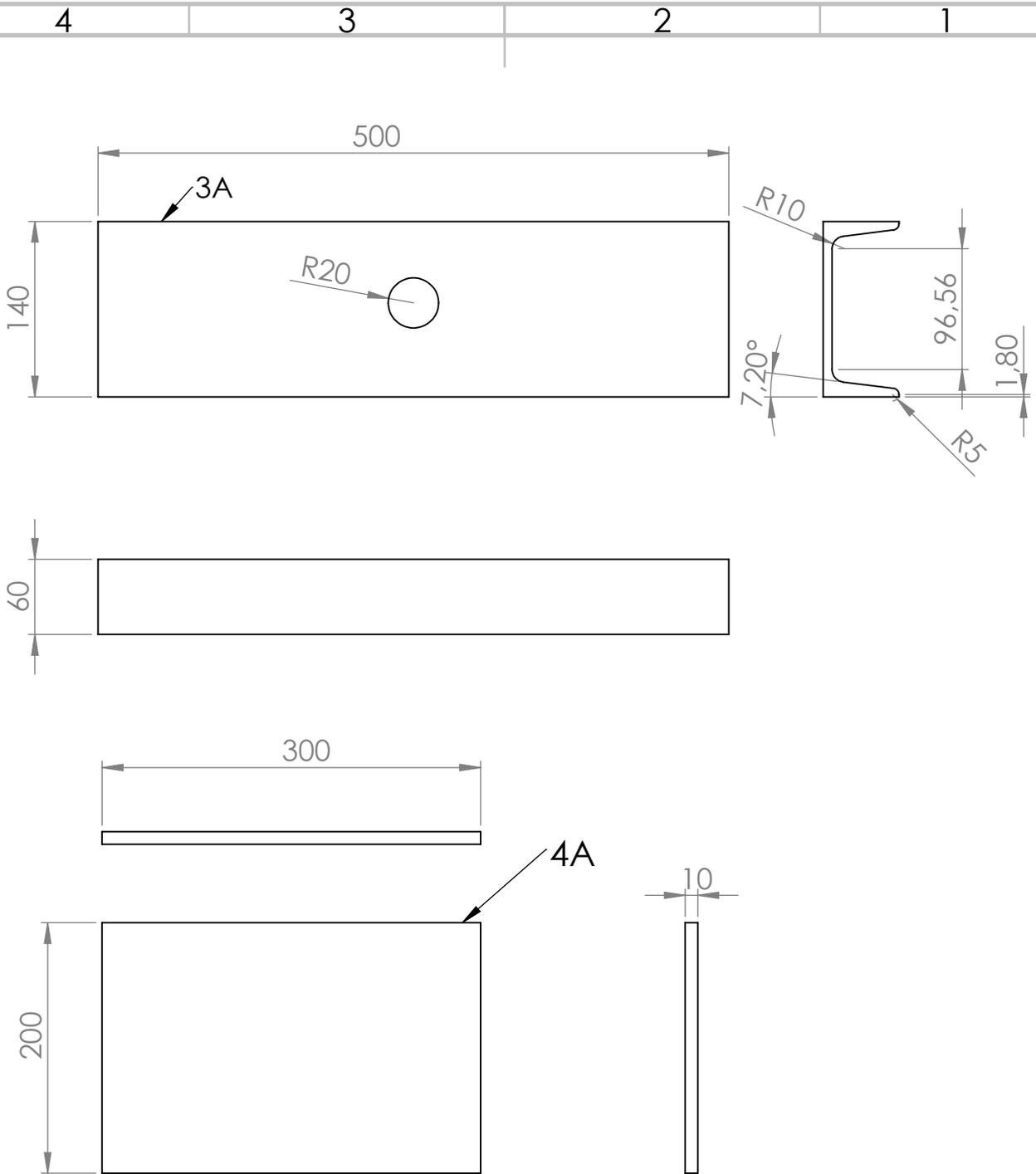
4 3 2 1



DETALLE A
ESCALA 2 : 1.5



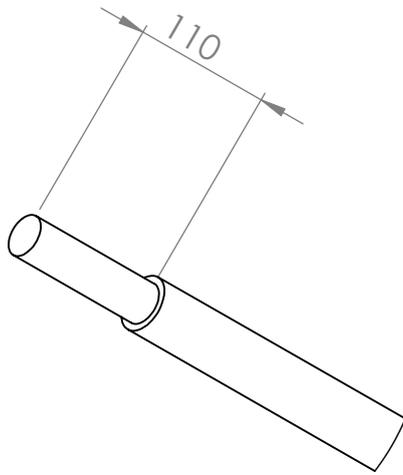
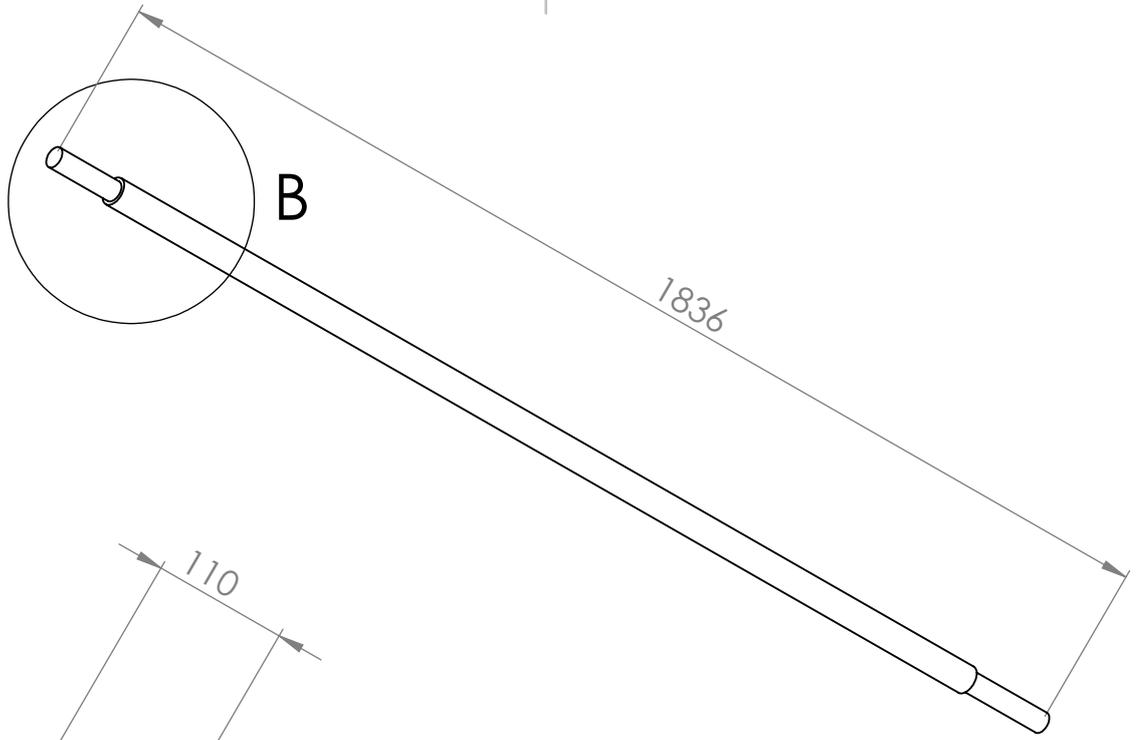
SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:		ACABADO:	REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA	REVISIÓN																		
<table border="1"> <tr> <th>NOMBRE</th> <th>FIRMA</th> <th>FECHA</th> </tr> <tr> <td> </td> <td> </td> <td> </td> </tr> </table>			NOMBRE	FIRMA	FECHA																TÍTULO: Partes conjunto A		
NOMBRE	FIRMA	FECHA																					
Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.			N.º DE DIBUJO PartesConjuntoA 1-2																				
PESO:			ESCALA:1:1		HOJA 1 DE 1																		



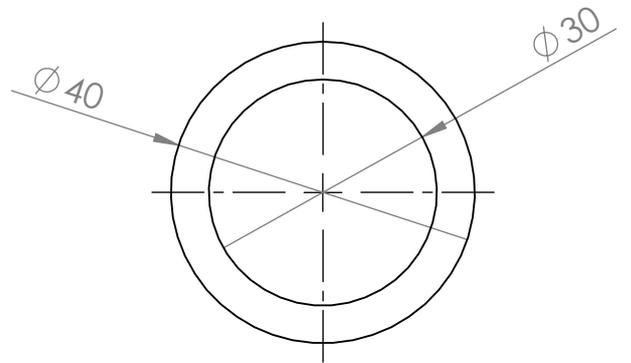
SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:		ACABADO:	REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA	REVISIÓN																		
<table border="1"> <tr> <th>NOMBRE</th> <th>FIRMA</th> <th>FECHA</th> </tr> <tr> <td> </td> <td> </td> <td> </td> </tr> </table>			NOMBRE	FIRMA	FECHA																TÍTULO: Partes conjunto A		
NOMBRE	FIRMA	FECHA																					
MATERIAL:			N.º DE DIBUJO																				
PESO:			ESCALA:1:10																				
HOJA 1 DE 1			HOJA 1 DE 1																				

Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

PartesConjuntoA 3-4 A4



DETALLE B
ESCALA 1 : 5



SI NO SE INDICA LO CONTRARIO:
LAS COTAS SE EXPRESAN EN MM
ACABADO SUPERFICIAL:
TOLERANCIAS:
LINEAL:
ANGULAR:

ACABADO:

REBARBAR Y
ROMPER ARISTAS
VIVAS

NO CAMBIE LA ESCALA

REVISIÓN

	NOMBRE	FIRMA	FECHA	
DIBUJ.				
VERIF.				
APROB.				
FABR.				
CALID.				
Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.			MATERIAL:	
			PESO:	

TÍTULO:

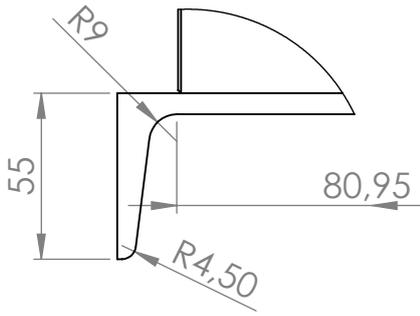
Ejes Guías
pieza 1

N.º DE DIBUJO

A4

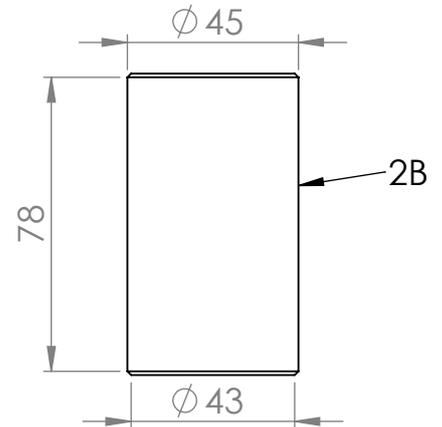
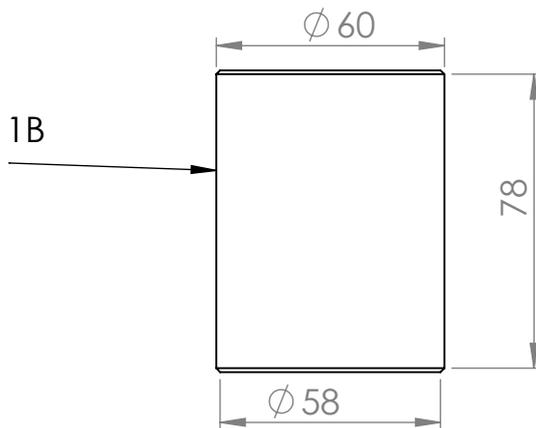
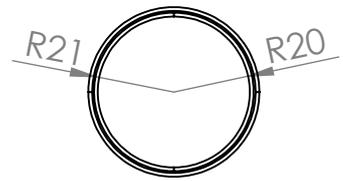
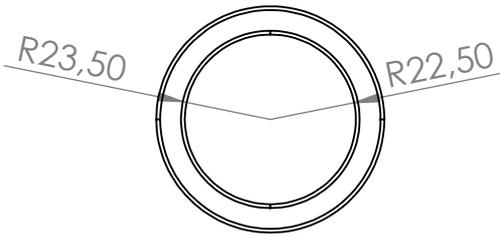
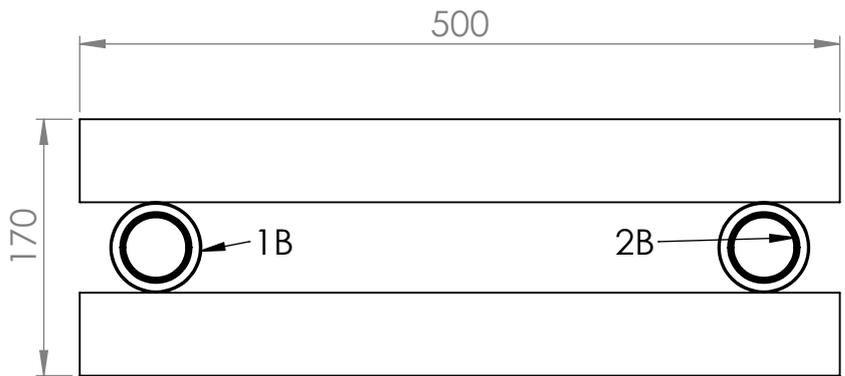
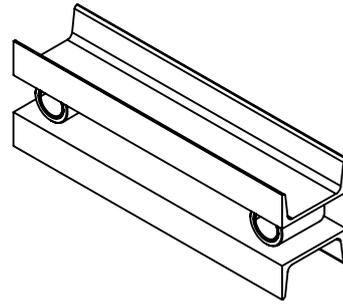
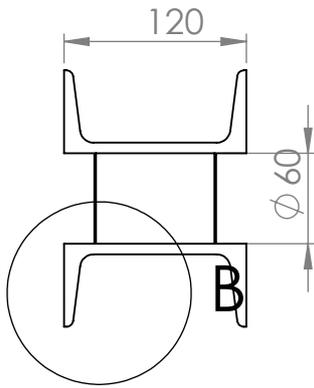
ESCALA:1:20

HOJA 1 DE 1



DETALLE B

ESCALA 2 : 5



SI NO SE INDICA LO CONTRARIO:
LAS COTAS SE EXPRESAN EN MM
ACABADO SUPERFICIAL:
TOLERANCIAS:
LINEAL:
ANGULAR:

ACABADO:

REBARBAR Y
ROMPER ARISTAS
VIVAS

NO CAMBIE LA ESCALA

REVISIÓN

	NOMBRE	FIRMA	FECHA
DIBUJ.			
VERIF.			
APROB.			
FABR.			
CALID.			

MATERIAL:

Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

PESO:

TÍTULO:

Conjunto B

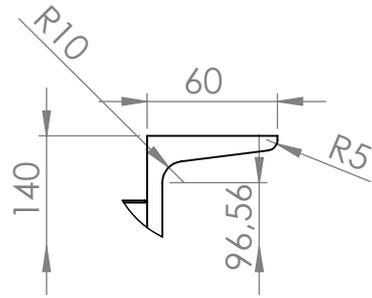
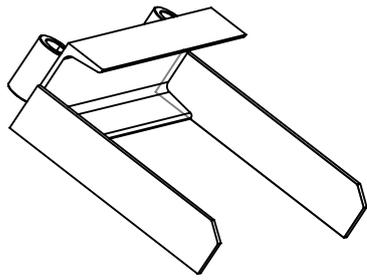
N.º DE DIBUJO

Conjunto B

A4

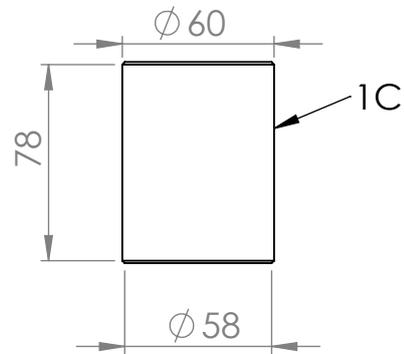
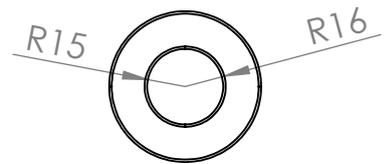
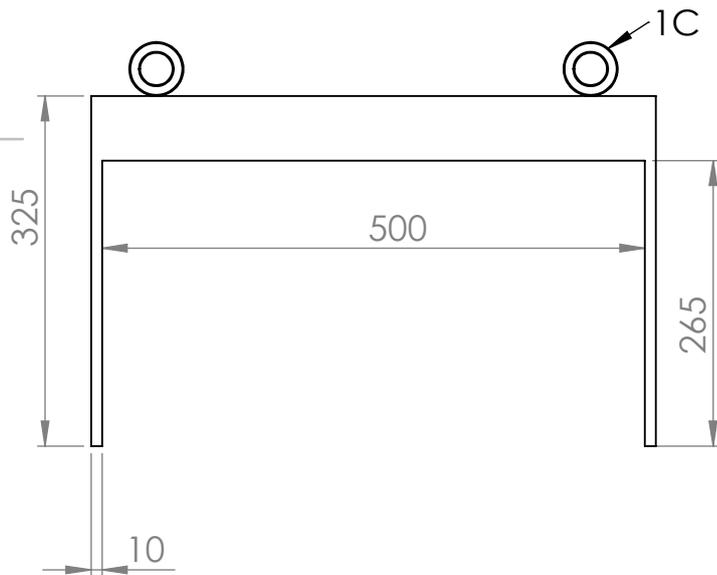
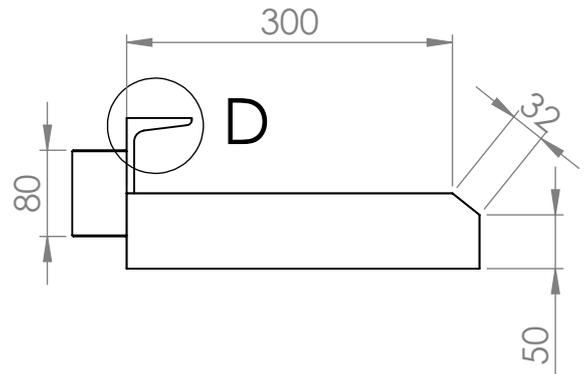
ESCALA: 1:10

HOJA 1 DE 1



DETALLE D

ESCALA 2:7



SI NO SE INDICA LO CONTRARIO:
LAS COTAS SE EXPRESAN EN MM
ACABADO SUPERFICIAL:
TOLERANCIAS:
LINEAL:
ANGULAR:

ACABADO:

REBARBAR Y
ROMPER ARISTAS
VIVAS

NO CAMBIE LA ESCALA

REVISIÓN

	NOMBRE	FIRMA	FECHA
DIBUJ.			
VERIF.			
APROB.			
FABR.			
CALID.			

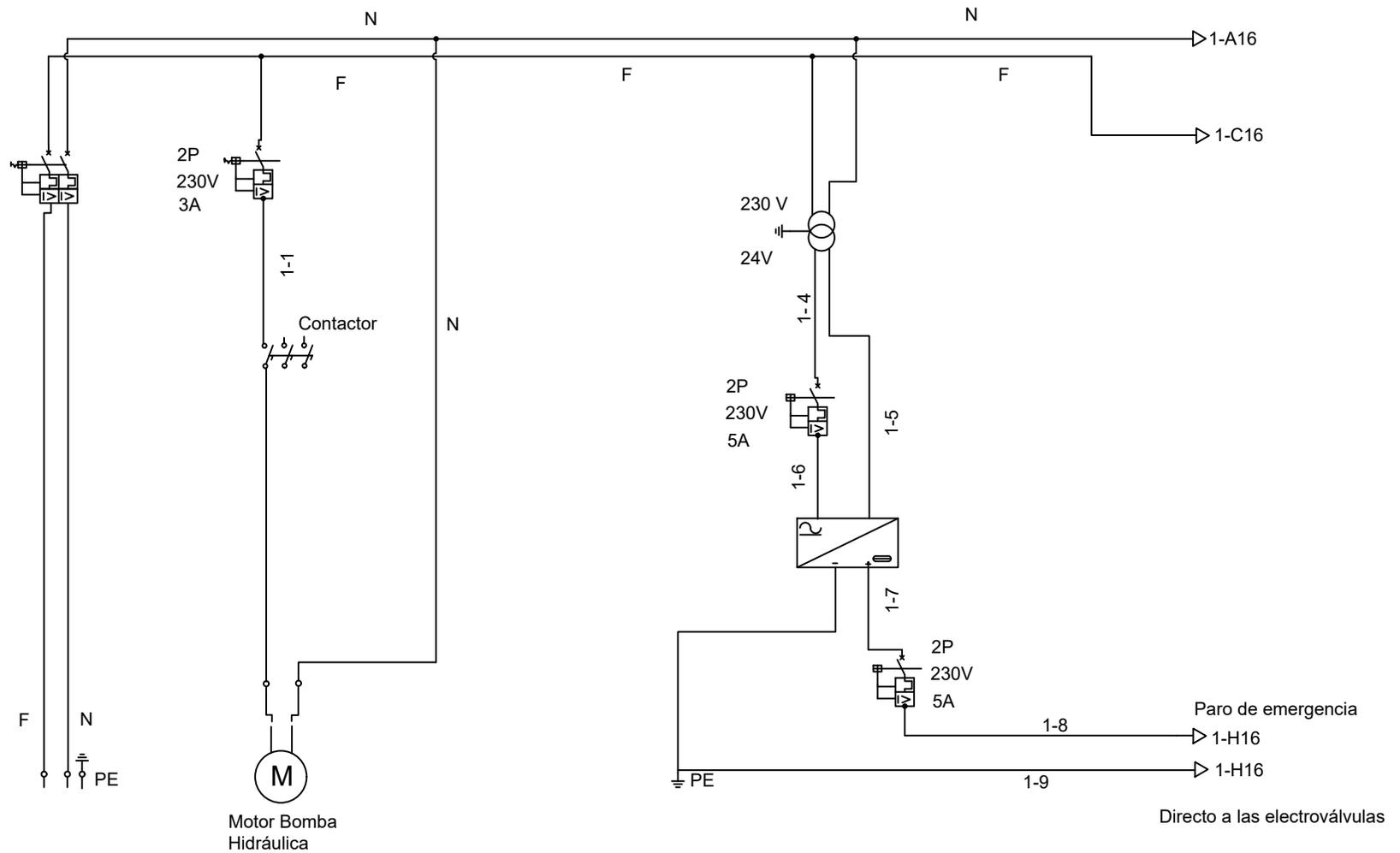
MATERIAL:

Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

PESO:

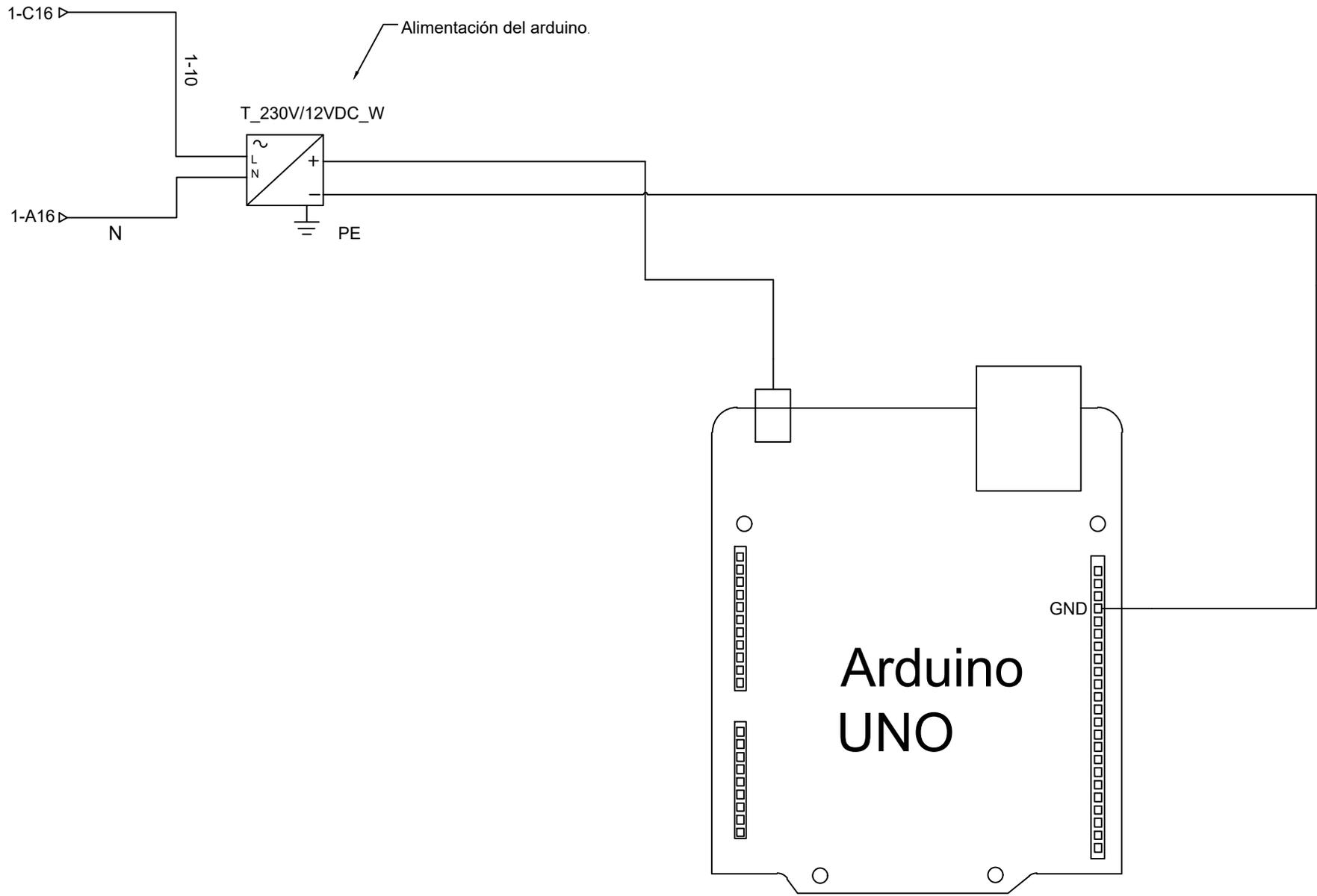
TÍTULO:	Conjunto C	
N.º DE DIBUJO	ConjuntoC	A4
ESCALA:1:10	HOJA 1 DE 1	

Anexo 3: Planos eléctricos



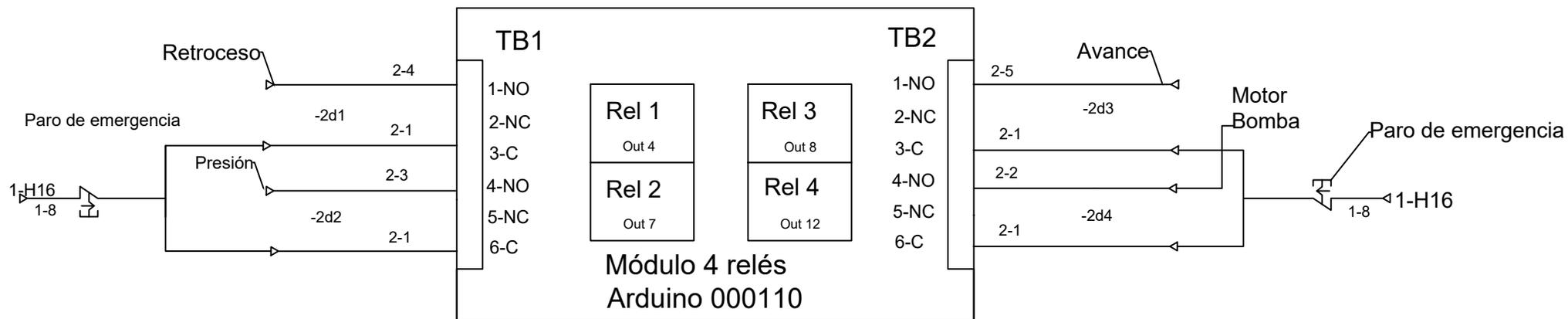
Planos Potencia

Escala	5/06/2017	Firma
	David	
	Rodriguez Cruz	
	TFG	



Alimentación Arduino

Escala	5/06/2017	Firma
	David	
	Rodriguez Cruz	
	TFG	



Maniobra Arduino

Escala

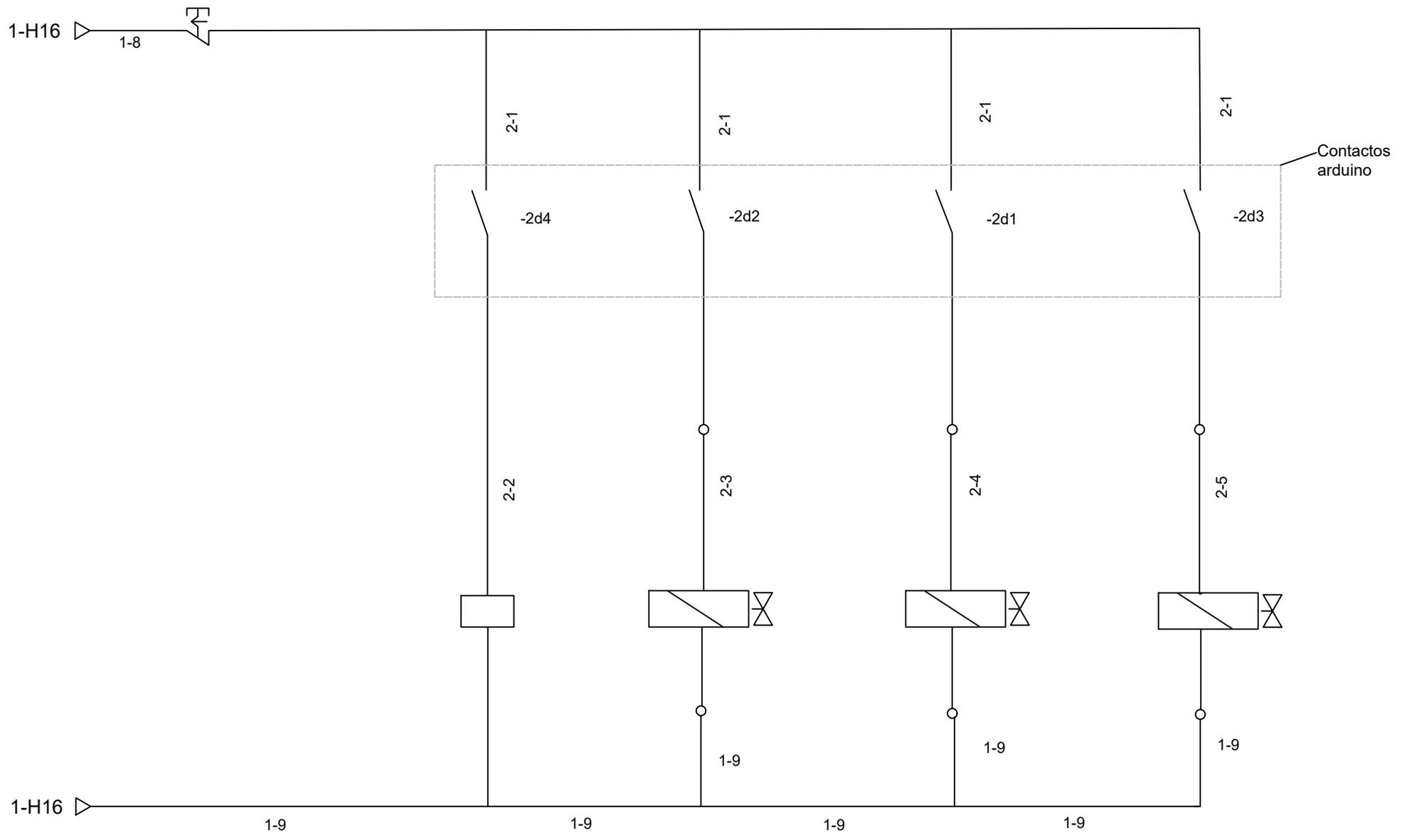
5/06/2017

Firma

David

Rodriguez Cruz

TFG

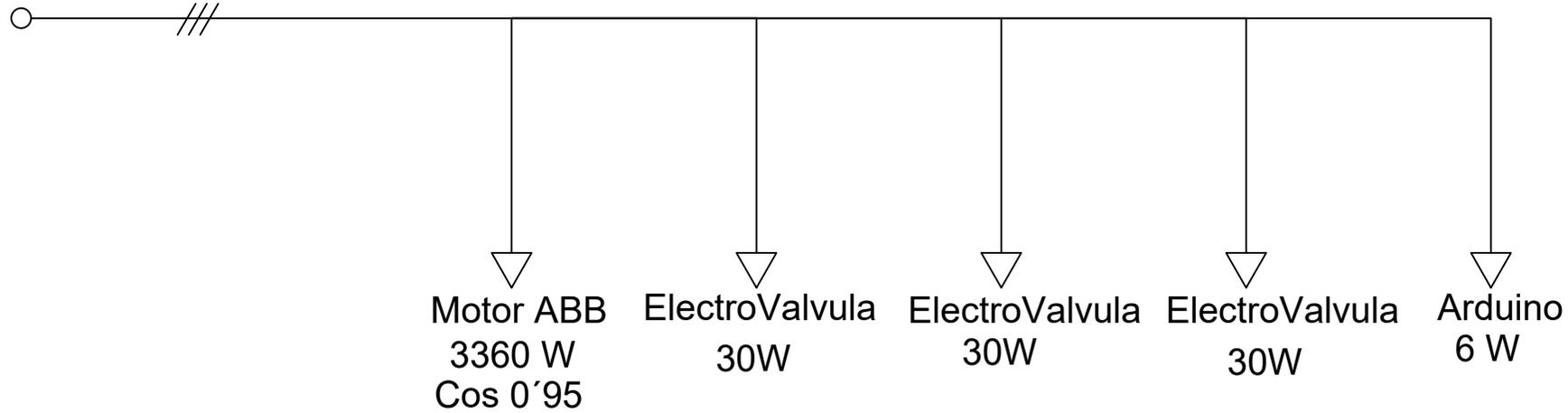


Maniobra Válvulas

Escala	5/06/2017	Firma
	David	
	Rodriguez Cruz	
	TFG	

230V

L= 5m



Esquema Consumo

Escala	5/06/2017	Firma
	David	
	Rodriguez Cruz	
	TFG	