



**UNIVERSIDAD DEL BÍO-BÍO**

FACULTAD DE INGENIERÍA  
DEPTO. INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

“ESTUDIO Y DISEÑO DE UN SISTEMA PROTOTIPO DE  
DETECCIÓN AUTOMÁTICA PARA GESTIÓN DE  
INVENTARIO DE INSERTOS DE TORNO”

DIEGO MAYOLAFQUÉN BUSTOS

SEMINARIO PARA OPTAR AL TÍTULO DE  
INGENIERO DE EJECUCIÓN EN ELECTRÓNICA

CONCEPCIÓN – CHILE

2018



**UNIVERSIDAD DEL BÍO-BÍO**

FACULTAD DE INGENIERÍA  
DEPTO. INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

“ESTUDIO Y DISEÑO DE UN SISTEMA PROTOTIPO DE  
DETECCIÓN AUTOMÁTICA PARA GESTIÓN DE  
INVENTARIO DE INSERTOS DE TORNO”

DIEGO MAYOLAFQUÉN BUSTOS

PROFESORES GUÍA:

LUIS VERA QUIROGA

JOHN CORREA TOLOZA

GUSTAVO SANHUEZA GARRIDO



*“Haz tu voluntad”*

– Aleister Crowley

## Agradecimientos

Este seminario va en agradecimiento a todas las personas que me brindaron su apoyo a lo largo de mi vida universitaria y escolar.

A mi familia, que desde siempre procuró otorgarme las herramientas para desarrollarme como profesional, al igual que la esperanza de un futuro mejor. A mi padre, cuya calidez ha sido fuente de inspiración y ejemplo a lo largo de mi vida. A mi madre, cuyo apoyo y valores han sido un pilar fundamental en mi desarrollo como persona y profesional. A mi abuela, cuyos cuidados y cariños me hicieron tan feliz, a pesar de no saberlo entonces. A mis queridas tías, por su cariño y apoyo incondicional en tantas, tantas dificultades. A mi abuelo, por su incansable apoyo hogareño y aquellos consejos de vida. A mis tíos y primos hermanos, cuya compañía y apoyo ha estado en las buenas y en las malas, en fiestas y desgracias; no sé qué habría hecho sin ellos. A mi hermano, que no dimensiona el grato respiro y descanso que han significado para mí esas tardes mirando series juntos. Y a mi hermanita, que me asiste cada vez que me ve lleno de cosas por hacer; que me defiende, me escucha y regalonea; que me hace bowlin y también me lo aguanta; que es una amiga, entre los pocos amigos que a esta altura quedan. Gracias a todos ellos.

A Carolina, que durante esos años en la universidad se convirtió en la mejor compañera que he tenido jamás. Por brindarme su compañía, afecto y apoyo, tanto emocional, como moral y académico. Pero por sobre todo, gracias por ser la persona que más me enseñó e hizo crecer en la vida. De no haberla conocido, en este punto no sería ni la sombra de la persona que soy ahora.

A mis amigos, que también son mi familia: Luciano, Felipe, Diego, Maximiliano y Alexander; mis hermanos del alma. Su apoyo moral, todas las risas y llantos que hemos compartido me han ayudado a mantenerme en pie, por tantos años. Allá donde estamos juntos, me siento en casa, y esa compañía es irremplazable. También a Ricardo Varela, que siempre estuvo conmigo esos primeros y lejanos años de universidad –los mejores años–, tratando de sobrevivir juntos a todos esos ramos horribles; nunca olvidaré aquellos viajes en bus. A Diego Miranda, que daba siempre los mejores consejos, un amigo fundamental en mis últimos años de carrera, y una máquina. Y a Cristóbal Novoa, uno de los hombres más apasionados que he conocido; no sé qué habría hecho sin sus palabras de aliento, o sin su amistad.

A mis compañeros de universidad, Claudio, Parkour, Purén, Daniel, Camilo, Tronco, Leo Garay, Ricky, Vejar, Parrita, Paulo. Por todas esas tardes y noches de estudio, por todas esas conversaciones de carrera, por la paciencia, y por el apoyo mutuo.

Al profesor Luis Vera, que estuvo muy presente a lo largo de todo el desarrollo del proyecto, brindándome apoyo académico, logístico y moral. Este seminario no hubiese sido posible sin él.

A todas aquellas fuentes de información gratuita, especialmente a los hilos de Arduino Forum, que tanto me enseñaron de programación. Mención especial para el gran Surbyte, bendito Surbyte.

Y por último, a la música, mi música. Eterna fuente de inspiración, contención, motivación, desahogo y esperanza. Porque el mundo sin música sería un grave error.

Diego

## ÍNDICE DE CONTENIDOS

ÍNDICE DE CONTENIDOS.....	5
ÍNDICE DE FIGURAS .....	8
ÍNDICE DE TABLAS .....	11
ÍNDICE DE ECUACIONES .....	11
TÍTULO DEL SEMINARIO .....	12
OBJETIVOS .....	12
RESUMEN .....	13
INTRODUCCIÓN .....	14
CAPÍTULO 1: PROBLEMÁTICA.....	15
1.01 Descripción de la empresa y rubro .....	15
1.02 Identificación del Problema .....	16
1.03 Resumen Capítulo I .....	17
CAPÍTULO 2: MARCO TEÓRICO .....	18
2.01 Insertos de torno.....	18
2.02 Introducción a los sistemas logísticos de control de inventario .....	19
2.03 Tecnologías usadas en control de inventario perpetuo.....	21
2.03.1 Código de barras .....	21
2.03.2 RFID .....	25
2.04 Estado del arte .....	27
2.04.1 SUPPLYPRO.....	27
2.04.2 ELA INNOVATION .....	28
2.04.3 AUTOCRIB.....	28
2.05 Resumen Capítulo 2 .....	29
CAPÍTULO 3: ESQUEMA GENERAL PARA LA ADQUISICIÓN DE DATOS.....	31
3.01 Entorno.....	32
3.02 Funcionamiento general .....	35
3.02.1 Microcontrolador .....	35
3.02.2 Acondicionamiento .....	36
3.02.3 Interfaz .....	37
3.02.4 Comunicación.....	37
3.03 Diagrama general .....	39

CAPÍTULO 4: PRINCIPIO DE FUNCIONAMIENTO.....	40
4.01 Sensor.....	40
4.02 Sensores potenciales.....	41
4.03 Selección del sensor .....	46
4.04 Principio de funcionamiento .....	47
4.05 Requerimientos .....	51
4.05.1 Iluminación controlada .....	51
4.05.2 Limit Switch .....	54
CAPÍTULO 5: ACONDICIONAMIENTO DE SEÑAL.....	55
5.01 Divisor de tensión.....	55
5.02 Multiplexación.....	60
5.03 Resumen Capítulo 5 .....	63
CAPÍTULO 6: MICROCONTROLADOR Y PERIFÉRICOS .....	64
6.01 Configuración del ATmega328P.....	65
6.02 Algoritmo de sensado .....	67
6.03 Conexión de periféricos .....	71
6.03.1 Etapa de Multiplexación.....	71
6.03.2 Iluminación.....	72
6.03.3 Limit Switch .....	73
6.03.4 Pantalla LCD.....	74
6.04 Esquema Resultante.....	76
6.05 Resumen capítulo 6.....	77
CAPÍTULO 7: COMUNICACIÓN Y NODEMCU.....	78
7.01 Configuración del NodeMCU.....	78
7.02 Acoplamiento al sistema .....	80
7.03 Infraestructura de internet para la recepción.....	82
7.04 Algoritmo de comunicación .....	83
7.05 Resumen capítulo 7 .....	86
CAPÍTULO 8: RESULTADOS EXPERIMENTALES .....	87
8.01 Parámetros de iluminación .....	88
8.02 Parámetros de la LDR.....	89
8.03 Conteo en <i>display LCD</i> .....	91
8.04 Recepción a la base de datos .....	96

8.05	Consumo del sistema .....	97
8.05.1	Mediciones de corriente .....	98
8.05.1	Cálculo y resultados de consumo.....	101
CONCLUSIONES .....		104
ANEXO A.....		105
ANEXO B .....		109
ANEXO C.....		114
ANEXO D.....		117



## ÍNDICE DE FIGURAS

Fig. 1.1: Logo de empresa metal-mecánica.....	15
Fig. 1.2: Muestra de algunos servicios ofrecidos por la empresa. ....	15
Fig. 1.3: Uso habitual de las herramientas (como por ejemplo insertos, engranajes, pie de metro, etc.) en un día común y corriente, en el taller de la empresa. ....	16
Fig. 2.1: Insertos de corte para torno. a) Inserto triangular. b) Inserto cuadrado.....	18
Fig. 2.2: Herramientas de corte. Se pueden apreciar tanto los insertos como sus respectivos porta-insertos o vástagos.....	19
Fig. 2.3: Ejemplo del escaneo de un código de barra.....	21
Fig. 2.4: Ejemplos de códigos de barra europea y norteamericano. ....	22
Fig. 2.5: Ejemplo de un código de barra Code 39.....	22
Fig. 2.6: Ejemplo de un código de barra Code 128. ....	23
Fig. 2.7: Ejemplo de un código de barra PDF417. ....	23
Fig. 2.8: Ejemplo de un código de barra Datamatrix.....	23
Fig. 2.9: Ejemplo de un código de barra Datamatrix marcado por deformación sobre una pieza metálica.....	24
Fig. 2.10: Ejemplo de un código de barra QR.....	24
Fig. 2.11: Ejemplos de algunos formatos de etiquetas RFID, entre una gran variedad existente. ...	25
Fig. 2.12: Ejemplo de un lector RFID. ....	25
Fig. 2.13: Esquema general del sistema RFID.....	26
Fig. 2.14: Producto comercial SBDII, de SupplyPro. a) Viste completa del SBDII. b) Compartimiento iluminado de un contenedor de plástico iluminado, con una caja porta-insertos en su interior. Imágenes comerciales.....	27
Fig. 2.15: Producto comercial Authomatic Tool Inventory, de ELA INNOVATION. Imagen comercial. ....	28
Fig. 2.16: Productos comerciales de AutoCrib. a) RoboCrib TX750. b) Vista de una de las gavetas del Autolocker FX Scale. Imágenes comerciales. ....	29
Fig. 3.1: Dos cajas contenedoras para los insertos, una grande y una pequeña. La caja de la parte superior corresponde a la mencionada, con compartimientos de 28x28 [mm], mientras que la pequeña es de 19x19 [mm].....	32
Fig. 3.2: Caja contenedora de 28x28 [mm], con insertos en su interior. ....	32
Fig. 3.3: Dimensiones de caja porta-insertos con compartimientos de 28x28 [mm]. ....	33
Fig. 3.4: Dimensiones de caja porta-insertos con compartimientos de 19 [mm]. ....	33
Fig. 3.5: Muebles usados en el taller.....	34
Fig. 3.6: Microcontrolador ATMEL ATmega328P-PU. ....	35
Fig. 3.7: Pantalla LCD 16x2. Muestra hasta 32 caracteres repartidos en 2 filas y 16 columnas. ....	37
Fig. 3.8: Módulos basados en el chip ESP8266: a) SoC ESP-01. b) SoC ESP-12. c) Development board NodeMCU.....	38
Fig. 3.9: Esquema general del sistema. Cada bloque representa una etapa. ....	39
Fig. 4.1: Sensores de proximidad inductivos.....	41
Fig. 4.2: Esquema del campo electromagnético generado por el sensor.....	42
Fig. 4.3: Campo electromagnético. a) Sensor no blindado. b) Sensor blindado. ....	42

Fig. 4.4: Foto-resistor: a) Vista de un LDR convencional. b) Símbolos eléctricos usuales.....	43
Fig. 4.5: Esquema del estampado metálico de una galga extensiométrica común. ....	45
Fig. 4.6: Ejemplo de galga extensiométrica de hilo conductor. ....	45
Fig. 4.7: Ejemplo de galga extensiométrica de semiconductor.....	46
Fig. 4.8: Estados del sensor en vista lateral: a) Inserto ausente, plena luz, baja resistencia. b) Inserto presente, nula luz, alta resistencia. ....	47
Fig. 4.9: Disposición de los compartimientos en forma de matriz, instalada en la base de la gaveta. ....	48
Fig. 4.10: Comportamiento de los eventos de intervención al stock en el tiempo. ....	48
Fig. 4.11: Vista lateral de la gaveta, y dentro de ella los sensores; en la cara interna superior se ubica la iluminación.....	49
Fig. 4.12: Vista lateral de la gaveta: a) Gaveta al abrirse. b) Gaveta al cerrarse, momento en el cual la iluminación se enciende, permitiendo el sensado. ....	50
Fig. 4.13: Diagrama de flujo que resume el modo de funcionamiento desde el punto de vista del sensor. ....	51
Fig. 4.14: Imagen referencial de una tira LED 5050. ....	52
Fig. 4.15: Características de flujo, tabla extraída del datasheet. ....	53
Fig. 4.16: Características eléctricas, tabla extraída del datasheet. ....	53
Fig. 4.17: Esquema físico frontal de la LED strip con sus respectivas dimensiones, extraído del datasheet. ....	53
Fig. 4.18: Módulo Collision Switch YL-99. ....	54
Fig. 5.1: Divisor de tensión compuesto por una resistencia y una fotorresistencia. ....	56
Fig. 5.2: Curva característica Iluminancia-Resistencia. ....	58
Fig. 5.3: Divisor de tensión en: a) oscuridad y b) iluminación.....	59
Fig. 5.4: Esquema del IC 74HC151N, extraído del Datasheet.....	60
Fig. 5.5: Circuito completo de sub-etapa de multiplexación. ....	61
Fig. 6.1: Esquema de pines del Atmega328P-UP.....	66
Fig. 6.2: Configuración del Atmega328P-UP. ....	67
Fig. 6.3: Cadena de caracteres de ejemplo, que se envía al NodeMCU por medio de comunicación serie.....	69
Fig. 6.4: Diagrama de flujo del algoritmo de funcionamiento para el ATmega. ....	70
Fig. 6.5: Esquema de conexionado de la etapa MUX al ATmega.....	71
Fig. 6.6: Esquema de conexionado de las tiras LED al ATmega328P. ....	73
Fig. 6.7: Esquema de conexionado del módulo YL-99 al ATmega328P. ....	74
Fig. 6.8: Esquema de conexionado de la pantalla LCD 16x2 al ATmega328P. ....	75
Fig. 6.9: Esquema de conexionado total de los periféricos al ATmega328P. ....	76
Fig. 7.1: Diagrama de pines del NodeMCU v1.0.....	79
Fig. 7.2: Esquema interno y de conexionado del 2N7000.....	80
Fig. 7.3: Esquema de comunicación serie entre NodeMCU y ATmega328P. El circuito intermedio corresponde al adaptador de nivel, basado en el MOSFET 2N7000G. ....	81
Fig. 7.4: Ventana de operación del sistema WAMP utilizado, Uniform Server.....	82
Fig. 7.5: Ejemplo de la cadena de caracteres de datos resultante en el formato del método POST.84	
Fig. 7.6: Diagrama de flujo del algoritmo de funcionamiento para el NodeMCU.....	85
Fig. 8.1: Circuito electrónico de prueba, montado sobre protoboard.....	87

Fig. 8.2: Estado de tiras LED: a) OFF. b) ON.....	88
Fig. 8.3: Tira LED dispuesta por sobre la LDR, a corta distancia.....	89
Fig. 8.4: LDR bajo plena luz de la tira LED. A la derecha se observa un multímetro midiendo tensión.....	90
Fig. 8.5: LDR bajo la sombra, luz de la tira LED, bloqueada manualmente.....	90
Fig. 8.6: Circuito de entrada con switches, para prueba de conteo. ....	92
Fig. 8.7: Display al iniciarse.....	92
Fig. 8.8: Display al comenzar una detección por primera vez.....	93
Fig. 8.9: Mensaje arrojado cuando la cantidad de entradas altas detectadas es 0. ....	93
Fig. 8.10: Mensaje arrojado cuando la gaveta es abierta y la última cantidad detectada fue 0. ....	94
Fig. 8.11: Mensaje arrojado cuando la gaveta vuelve a cerrarse y ha hecho una detección de 0 justo antes.....	94
Fig. 8.12: Mensaje arrojado cuando la cantidad de insertos es 1.....	94
Fig. 8.13: Mensaje arrojado cuando se vuelve a abrir la gaveta, esta vez con la nueva cantidad detectada. ....	95
Fig. 8.14: Mensajes arrojados para varias cantidad de estado alto detectados. ....	95
Fig. 8.15: Mensaje arrojado cuando el proceso de detección es interrumpido. ....	96
Fig. 8.16: Tabla de datos de phpMyAdmin. Contiene una muestra aleatoria de estados de entrada. .....	97
Fig. 8.17: Medición de corriente a GND en etapas de acondicionamiento (divisor de tensión y MUX) y microprocesador. ....	98
Fig. 8.18: Medición de corriente a GND en etapa de comunicación (NodeMCU y Level shifter). ....	99
Fig. 8.19: Medición de corriente a GND en etapa de indicador LCD: a) En reposo. b) En detección. .....	99
Fig. 8.20: Medición de corriente a GND en etapa de indicador LCD, sin back light.....	100
Fig. 8.21: Medición de corriente a GND en limit switch: a) En reposo (abierto). b) En detección (cerrado).....	100
Fig. 8.22: Medición de corriente a GND en etapa de iluminación: a) En reposo. b) En detección.	101

## ÍNDICE DE TABLAS

Tabla 1: Principales ventajas y desventajas del sensor inductivo.....	43
Tabla 2: Principales ventajas y desventajas de los foto-resistores.....	44
Tabla 3: Principales ventajas y desventajas de las galgas extensiométricas.....	46
Tabla 4: Tabla de verdad, comportamiento de la sub-etapa de multiplexación.....	62
Tabla 5: Configuración de GPIOs en el ATmega328P para el presente algoritmo.....	68
Tabla 6: Principales parámetros eléctricos de la etapa de iluminación.....	89
Tabla 7: Parámetros eléctricos de una rama del divisor de tensión.....	91
Tabla 8: Tabla de corrientes demandadas por etapa, y respectivas tensiones de alimentación. ..	101
Tabla 9: Potencias eléctricas parciales del sistema.....	101
Tabla 10: Tiempos de reposo y detección para varios lapsos de uso definido.....	102
Tabla 11: Consumos eléctricos totales del sistema.....	103

## ÍNDICE DE ECUACIONES

Cantidad de GPIOs mediante multiplexor (3.1).....	36
Corriente estimada en las tiras LED_(4.1).....	53
Criterio preliminar de Rmin para divisor de tensión en LDR_(5.1).....	57
Criterio preliminar de Rmax para divisor de tensión en LDR_(5.2).....	57
Condición de R para estado bajo en LDR_(5.3).....	57
Condición de R para estado alto en LDR_(5.4).....	57
Tensión de LDR en estado bajo_(5.5).....	58
Tensión de LDR en estado alto_(5.6).....	58
Tensión en la resistencia de Base_(6.1).....	72
Corriente en la resistencia de Base_(6.2).....	72
Consumo eléctrico (8.1).....	98
Potencia eléctrica (8.2).....	98
Consumo probable (8.3).....	102

## TÍTULO DEL SEMINARIO

“ESTUDIO Y DISEÑO DE UN SISTEMA PROTOTIPO DE DETECCIÓN AUTOMÁTICA PARA GESTIÓN DE INVENTARIO DE INSERTOS DE TORNO”

### OBJETIVOS

#### Objetivos generales:

- Estudiar y diseñar un sistema electrónico de sensado que permita recoger, de forma eficiente y práctica, la información necesaria para la puesta en marcha de un sistema de gestión de inventario, en función de la problemática de los *insertos de torno* y de las condiciones presentes en el taller de la empresa *Orecal*, ejemplo real de la existencia del problema.
- Implementar el circuito en *protoboard* con elementos de bajo costo, que pruebe el correcto funcionamiento del sistema, apuntando así a validar el prototipo como proyecto escalable.

#### Objetivos específicos:

- Diseñar un sistema electrónico de sensado que permita detectar la presencia y ausencia de *insertos de torno*.
- Comunicar los datos recogidos por el sistema electrónico a un computador, datos con los cuales se debiera alimentar a un software de gestión de inventario.
- Especificar las condiciones de entorno adecuadas para la implementación de dicho sistema, condiciones que sienten las bases para implementar una maqueta que lo sostenga, permitiendo su sencillo y eficaz uso.
- Estudiar el consumo del sistema y minimizarlo en lo posible.

## RESUMEN

El proyecto electrónico descrito en este informe plantea una solución tecnológica a un problema en particular de un rubro industrial, el de los talleres metal-mecánicos. La dificultad con la que cuentan es un problema de tipo organizacional, en cuanto a la difícil gestión de los *insertos de torno*, una herramienta especializada para corte de metales. Se requiere un sistema que permita detectar y registrar en un PC, de forma automática, la cantidad de *insertos*, de tal forma que dicha información se encuentre disponible para ser recogida por un software informático de gestión de inventario.

Este estudio sienta las bases para generar un prototipo de bajo costo, capaz de detectar y contar eficazmente las herramientas mencionadas, y se plantea como una alternativa a otros dispositivos sumamente robustos y costosos que, apenas disponibles incluso en el mercado internacional, permiten abordar una tarea como esta.

El diseño electrónico documentado en este informe permite la implementación en *protoboard* de un sistema de detección de hasta 32 insertos. Su correcto funcionamiento valida la idea de desarrollar un prototipo a mayor escala, que cumpla la misma función para una cantidad real de herramientas en un taller metal-mecánico.

A lo largo del diseño e implementación se consiguió satisfactoriamente la funcionalidad general del circuito, así como los objetivos planteados. Además, se plantean formas de optimizar el consumo, así como la posibilidad de utilizar otros principios de funcionamiento en la detección.

## INTRODUCCIÓN

El presente informe plantea un proyecto de implementación tecnológica destinado a solucionar una problemática específica, hallada en los talleres de una empresa metal-mecánica llamada Orecal. Dicha problemática, que consiste en la dificultad de organización que presupone el uso de una gran cantidad de herramientas de uso especializado, los *insertos de torno*, ha ido exigiendo, con creciente ímpetu, de un sistema de inventariado capaz de facilitar el adecuado uso de estas piezas.

El proyecto consiste en el estudio y diseño de un sistema electrónico prototipo de bajo costo que permita registrar de forma práctica y eficaz la presencia o ausencia de las mencionadas herramientas, sentando las bases para, en conjunto a un sistema informático, obtener la información del stock disponible en la pantalla de un computador personal.

El desarrollo del prototipo apunta a una eventual implementación a mayor escala para su uso real y efectivo, que permita al usuario contar con un inventario computacional continuamente actualizado.

## CAPÍTULO 1: PROBLEMÁTICA

### 1.01 Descripción de la empresa y rubro

Orecal es una empresa del rubro metalmecánico, ubicada en la ciudad de Concepción, Chile, que se dedica a suministrar servicios y soluciones al sector industrial, marítimo y forestal. Sus servicios de maestranza consisten principalmente en la preparación y fabricación de partes y piezas de conjuntos mecánicos utilizados en la industria.

En la Fig. 1.1 se presenta el logo de la empresa.



Fig. 1.1: Logo de empresa metal-mecánica.

Las piezas fabricadas por esta empresa son, en su mayoría, engranajes y ejes para su utilización como repuestos en maquinarias de diversas marcas, manufacturados en acero al carbono e inoxidables, bronce, aluminio y plástico industrial. Con este objetivo, la empresa cuenta con talleres especializados en el tratamiento de dichos materiales, en los cuales se utiliza, a su vez, máquinas de CNC para corte por torneado y por agua a alta presión. En la Fig. 1.2 pueden apreciarse una tomas de los servicios en acción ofrecidos por la empresa.



Fig. 1.2: Muestra de algunos servicios ofrecidos por la empresa.



## 1.02 Identificación del Problema

Los servicios prestados por la empresa Orecal implican el uso de una significativa cantidad de instrumentación e insumos especializados en sus talleres, cuyo adecuado manejo resulta de vital importancia para la ejecución de los trabajos allí realizados. Bajo esta premisa, la empresa ha hecho esfuerzos por mantener una correcta organización mediante inventarios manuales y otras soluciones prácticas, de tal forma que exista una conveniente ubicación y un ágil acceso a los instrumentos, así como un registro del stock disponible.

Sin embargo, la enorme cantidad de herramientas especializadas, y más específicamente, la gran cantidad de *insertos para corte por torneado*, hace dificultosa la tarea organizacional: muchos insumos y herramientas son utilizados sin que se deje un registro del uso ni del stock restante, y al hacerlo, permanecen por largo tiempo fuera de lugar. En la Fig. 1.3 se aprecia el habitual e inconveniente almacenamiento y trato que se le da a las herramientas en un día de trabajo cualquiera.



Fig. 1.3: Uso habitual de las herramientas (como por ejemplo insertos, engranajes, pie de metro, etc.) en un día común y corriente, en el taller de la empresa.

Esta situación tiene consecuencias negativas para la empresa, ya que al no existir un adecuado registro del stock, hay ocasiones en las que el taller de la empresa requiere con urgencia un cierto insumo y solo en ese momento el personal nota que no dispone de él. La producción se detiene hasta reponer el stock y, evidentemente, la empresa se ve en apuros, además de tener que cubrir gastos adicionales.

La gestión de inventario es una problemática recurrente en las empresas que trabajan con productos –aunque sea relativamente– numerosos, ya que incide directamente sobre la productividad, y por consiguiente, sobre la rentabilidad de la empresa. Sin embargo, no todas las empresas manejan los mismos productos, y por tanto, algunas requieren de soluciones diferentes. Tal es el caso de las maestranzas metalmeccánicas, como *Orecal*.

### 1.03 Resumen Capítulo I

La empresa metalmeccánica Orecal utiliza diversos tipos de herramientas para ofrecer sus servicios, entre los cuales se encuentran los insertos de torno, pequeñas piezas metálicas para uso de corte por torneado. El manejo de la numerosa cantidad de dicha herramienta vuelve dificultosa su adecuada organización, cuyo consecuente desorden y desconocimiento del *stock*, se traduce en el eventual estancamiento de las actividades en forma repetitiva.

Así, habiéndose detectado una problemática que entorpece las actividades de la empresa, se vuelve evidente la necesidad de buscar una nueva medida que apunte a solucionarla.

De manera preliminar, resulta intuitivo pensar que dicha solución deba ser un sistema que detecte e informe sobre el stock de herramientas en tiempo real mientras mantiene una adecuada organización de las herramientas. Los siguientes capítulos se basan en el estudio, diseño e implementación de una solución.

## CAPÍTULO 2: MARCO TEÓRICO

### 2.01 Insertos de torno

Los *insertos de corte para mecanizado*, también llamados vulgarmente “*pepas*”, “*plaquitas*” o simplemente *insertos*, son pequeñas herramientas de corte de uso especializado, hechas de metal recubierto con químicos que le permiten presentar una mayor resistencia al desgaste a altas temperaturas. Son ampliamente utilizados en la industria manufacturera metalmeccánica, usualmente mediante máquinas CNC. En la Fig. 2.1, a modo de ejemplo, puede apreciarse dos tipos de inserto, entre la gran variedad existente.



Fig. 2.1: Insertos de corte para torno. a) Inserto triangular. b) Inserto cuadrado.

Los insertos son fabricados a base de carburos. Los carburos son compuestos químicos formados por la unión entre carbono y otro elemento menos electronegativo que él. En el caso de los carburos metálicos, se trata de la aleación entre carbono y otro metal menos electronegativo que él, como por ejemplo el litio, magnesio, aluminio, o hierro, entre otros. En el caso de los insertos para corte por mecanizado se suele utilizar el carburo de tungsteno, titanio o tántalo, que aleados sobre una matriz de níquel o cobalto, permiten obtener un material de extrema dureza, cercana a la del diamante, solo que a un costo mucho menor.

Los insertos son herramientas de uso especializado y, como tal, se clasifican en muchos tipos o variedades según sus parámetros mecánicos, en base a las normas ISO-1832 e ISO-513. Entre sus parámetros más importantes se encuentra: la forma del inserto, ángulo de alivio frontal, tolerancia en las dimensiones, sistema de sujeción, longitud de la arista de filo, espesor y radio de la punta, entre otros. De estos parámetros también depende el porta-insertos a utilizarse. En la Fig. 2.2 puede verse cómo se disponen los insertos en los vástagos porta-insertos, que han de mantener bien sujeto a estas herramientas mientras entran en contacto con los objetos metálicos a mecanizar.



Fig. 2.2: Herramientas de corte. Se pueden apreciar tanto los insertos como sus respectivos porta-insertos o vástagos.

Para efectos de este estudio, los parámetros más importantes a considerar, relativas al objeto, son sus dimensiones, peso y material. Las dimensiones de los insertos a almacenarse en estas cajas pueden variar de 15 a 25 milímetros de diámetro, considerando el mayor círculo inscrito dentro de su perímetro más un cierto margen. El peso del inserto varía según su tipo, pero para las dimensiones consideradas varía entre 5 y 20 gramos.

## 2.02 Introducción a los sistemas logísticos de control de inventario

En la economía actual, todas las empresas buscan optimizar sus utilidades por diversos medios, ya sea mediante innovación, uso de tecnologías, publicidad, disminución de costos, etc. Es por esto que, entre ellos, la práctica de mantener un inventario, es decir, un registro detallado de todos los bienes con los que se cuenta, se ha vuelto casi una obligación en cualquier empresa --e incluso en otras organizaciones sin fines de lucro-- con tal de permitir una óptima gestión en la cadena de suministro y, por consiguiente, de los recursos, sobre todo cuando existe un constante flujo de compra y venta.

Un adecuado control de inventario permite al administrador decidir correctamente cuándo reabastecer el stock de un determinado producto y en qué cantidad. Por consiguiente, contar con un adecuado control de inventario permite evitar que el stock de un determinado producto se agote cuando en realidad se necesita, mermando así la cadena de producción; o por el contrario, permite evitar un sobreabastecimiento de algún bien, situación que solo se traducirá en el uso innecesario de los espacios y recursos de la organización. En términos generales, contribuye a sostener la rentabilidad y competitividad de una empresa.

En resumidas cuentas, un adecuado control de inventario permite:

- Cuantificar con mayor exactitud cuánta mercancía se tiene.
- Determinar cómo rotar adecuadamente los productos.
- Cuantificar las pérdidas por mercancía dañada o extraviada.
- Determinar qué decisiones tomar respecto a la producción.
- Contar con una herramienta que ayude a detectar y eliminar el robo de mercancías.

Los sistemas de control de inventario se pueden clasificar en dos tipos: sistema periódico y sistema perpetuo.

- a. Sistema periódico: Tal como su nombre lo indica, corresponde aquellos sistemas donde se realiza un control del inventario cada cierto período de tiempo, que suele ser mensual, trimestral, semestral o anual. El control se realiza mediante un conteo físico de las mercancías, obteniendo la información del inventario disponible solo en una determinada fecha. En base a la información obtenida en dicho control, se obtiene un balance en el costo asociado a las compras y ventas realizadas con respecto al anterior control, para así poder determinar en qué medida reabastecer los bienes, entre otras decisiones importantes.

Con este tipo de sistemas, las empresas no conocen en todo momento el stock de sus mercancías, ni cuánto es el costo asociado a ellas. Esta situación facilita la pérdida de inventario, no pudiendo hacerse seguimientos ni verificaciones sino hasta el final del período donde se haga un control, lo cual facilita posibles fraudes.

- b. Sistemas perpetuos: Estos sistemas, también llamados permanentes o continuos, corresponden aquellos donde se realiza un control de inventario en tiempo real, es decir, se actualiza constantemente el sistema con cada nueva operación de mercancía realizada. Esto reduce en gran medida la necesidad de hacer recuentos físicos del stock disponible y, por el contrario, permite conocer de forma bastante exacta el valor total del inventario, y los costos y ganancias asociadas en todo momento. Además no es necesario cerrar el negocio para mantener el control del inventario, a diferencia del sistema periódico.

Estos sistemas están basados en la utilización de equipos electrónicos de escaneo de productos mediante código de barras, y también de un software que registre los datos recogidos en cada operación.

Como desventaja presentan los costos de instalación de los equipos y software. También se requiere contar con el personal administrativo capacitado para navegar por el software, al igual que el operario capacitado para el uso de escáneres (herramienta de detección, habitual en este tipo de sistemas) y la caja. Además, estos sistemas no garantizan la veracidad de la información recopilada en casos de mal uso, ya sea por error de los operarios o bien por los robos que puedan ocurrir, si bien contribuye a sortear dichas problemáticas.

La elección del tipo de sistema está basada principalmente en la magnitud de las mercancías a inventariar, y en última instancia en los recursos que la empresa u organización puedan destinar a la implementación de dicho control. Sin embargo, mientras mayor es la cantidad de stock a manejar, más necesario se vuelve el uso de los sistemas perpetuos, en cuanto a la información y control que solo las tecnologías sofisticadas pueden ofrecer.

## 2.03 Tecnologías usadas en control de inventario perpetuo

El rápido desarrollo de las nuevas tecnologías ha contribuido a optimizar la producción de la industria en todas sus áreas, y también lo ha hecho en los sistemas de control de inventario. La información de las mercancías, que antiguamente se registraba obligatoriamente a mano, ha pasado a ser digitalizada y procesada por computadoras, agilizando notablemente los procesos, y por tanto la productividad.

Actualmente, para poder mantener un sistema de control perpetuo se vuelve necesario contar con la tecnología de adquisición de datos que permita recoger rápidamente la información de un producto. Así es cómo, desde los años setenta, los sistemas de código de barras tradicionales se han ido masificando hasta dominar casi todo el mercado, a pesar de no ser el único sistema existente.

En términos generales existen dos sistemas de identificación de productos que tienen cabida en la adquisición de datos para los sistemas de gestión de inventario:

- Código de barras.
- RFID.

### 2.03.1 Código de barras

El código de barras es una representación óptica de datos, leíble por medio de un lector láser, compuesta por una serie de líneas paralelas de distinto grosor y espaciado que representan un mensaje. Dicho mensaje suele identificar y/o describir algo acerca del objeto que lo porta. De esta forma, al ser portados por un objeto, y leídos instantáneamente por una pistola láser, como se muestra en la Fig. 2.3, los códigos de barra ofrecen un rápido y efectivo reconocimiento de mercancías, que suele ser aprovechado en control de inventario.



*Fig. 2.3: Ejemplo del escaneo de un código de barra.*

Los códigos de barra representan la información variando el ancho de las líneas y el espaciado entre ellas, y por tanto se pueden considerar como representaciones unidimensionales. El escáner lee el símbolo del código y lo convierte en una señal eléctrica digital, para ser enviada a un decodificador que interpreta la señal y la envía,

por algún medio (cable, bluetooth, wifi), a la computadora para que la información contenida sea procesada.

Por otro lado, el nombre puede no solo referirse a los códigos de barras tradicionales de una dimensión, sino a también englobar a los códigos bidimensionales. Existe una gran variedad de estándares de códigos de barra de ambos tipos para suplir las demandas de diferentes países y necesidades particulares. A continuación se presentan los principales tipos.

#### a. Códigos 1D:

- **EAN/UPC:** *European Article Number, Universal Product Code*. Inicialmente dos estándares de código de barras creados en USA y Europa respectivamente, se fusionaron para tener un registro de códigos universales. En la Fig. 2.4 se ejemplifica la equivalencia entre ambos códigos.

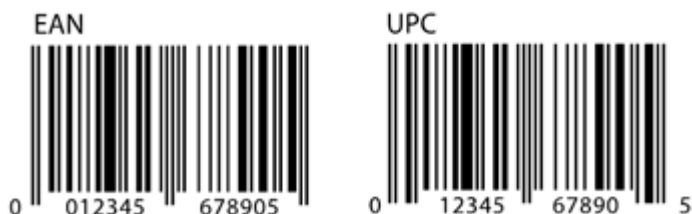


Fig. 2.4: Ejemplos de códigos de barra europea y norteamericano.

- **Code 39:** Creados bajo la necesidad de contar con un código alfa-numérico. A cambio, presentan una longitud de código demasiado grande, como se evidencia en la imagen de la Fig. 2.5, a pesar de lo cual hoy en día siguen siendo utilizados.



Fig. 2.5: Ejemplo de un código de barra Code 39.

- **Code 128:** Una versión mejorada del Code 39, bajo la necesidad de tener mayor información en un espacio más reducido, como se muestra en la Fig. 2.6. No genera un código de longitud tan amplia como su predecesor.



Fig. 2.6: Ejemplo de un código de barra Code 128.

b. Códigos 2D:

- **PDF417:** *Portable Document File*, es un estándar dedicado a almacenar altos contenidos de información, destinado principalmente a lugares donde no se pueda acceder directamente al sistema logístico, siendo usado en la industria, bancos, instituciones gubernamentales, transporte de mercadería, entre otros. La cantidad de símbolos o módulos a utilizar depende de la magnitud de la información, variando el área total del código, pudiendo almacenar hasta 1800 caracteres alfanuméricos en un solo símbolo. Un ejemplo es el que se muestra en la Fig. 2.7.



Fig. 2.7: Ejemplo de un código de barra PDF417.

- **Datamatrix:** Es un sistema de codificación industrial con altos volúmenes de información en tamaños muy reducidos, y detectable en entornos muy rigurosos. En la Fig. 2.8 se muestra un ejemplo del código.



Fig. 2.8: Ejemplo de un código de barra Datamatrix.

El código se marca por deformación sobre una pieza que se adhiere al producto final o directamente sobre el mismo, como se aprecia en la Fig. 2.9, asegurando así que la identificación no se desprenda nunca del objeto, cuya detección se hace por medio de cámaras.



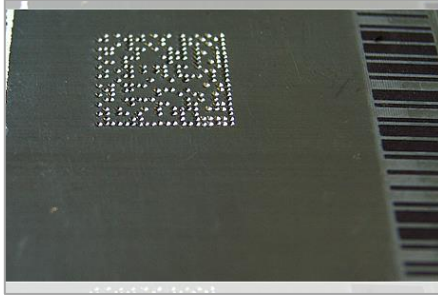


Fig. 2.9: Ejemplo de un código de barra Datamatrix marcado por deformación sobre una pieza metálica.

La codificación se realiza en módulos de celdas que ocupan espacios de 3 mm<sup>2</sup>, y que permiten almacenar hasta 3116 caracteres cada uno. El evidente costo asociado y el tipo de impresión por deformación sobre el objeto, hace que esta codificación no sea apta para entornos comerciales, sino para aplicaciones que requieren de altos estándares de calidad, como por ejemplo la industria electrónica, farmacéutica o aeroespacial.

- **QR Code:** *Quick Response Code*, es el sistema de código de barras bidimensional más difundido durante los últimos años, teniendo toda variedad de utilidades como identificador. Este auge se debe en gran medida a que su sistema de detección se basa en cámaras fotográficas, que hoy en día se encuentran en todos los smartphones, bajo el único pre-requisito de contar con su software de decodificación que, a su vez, es de código abierto. En la Fig. 2.10 se aprecia un ejemplo de código QR.



Fig. 2.10: Ejemplo de un código de barra QR.

Su implementación está basada en una cámara fotográfica, método más sencillo en comparación al de los otros tipos de códigos 2D, lo que le da un lugar de preferencia en los sistemas de control de inventario actuales, en conjunto a los códigos de barra 1D tradicionales. Su versatilidad le permite también ser usada en gestión de inventario, ya que su módulo puede contener hasta 4296 caracteres alfanuméricos, leíbles a gran velocidad, y con hasta un 30% tolerancia a errores de lectura.

## 2.03.2 RFID

Los sistemas RFID, *Radiofrequency Identification* (Identificación por Radiofrecuencia), son sistemas de almacenamiento y recuperación de datos a larga distancia, por medio de ondas de radio. Su funcionamiento está basado en tres elementos fundamentales:

### a. Etiqueta RFID (transponedor):

Se compone por una antena y un transductor de radio, enlazados a una memoria electrónica, que permite almacenar una gran cantidad de información orientada al objeto que porte la etiqueta. La información contiene la identificación de la etiqueta, que está destinada a identificar al objeto, además de poder contener una gran cantidad de información adicional, según se requiera. Se fabrican de forma compacta, de tal modo que pueden ser adheridas con facilidad a casi cualquier mercancía. Existen varios formatos según los requerimientos del objeto portador. A continuación, en la Fig. 2.11 se ven algunos ejemplos.



Fig. 2.11: Ejemplos de algunos formatos de etiquetas RFID, entre una gran variedad existente.

### b. Lector RFID (transceptor):

Se compone por una antena, un transceptor y un decodificador. Se mantiene constantemente transmitiendo señales de radio con el fin de detectar si hay alguna etiqueta que leer en sus inmediaciones. Al detectar una etiqueta, lee su información y se la envía al software de procesamiento. En la Fig. 2.12 se puede apreciar un tipo de lector RFID.



Fig. 2.12: Ejemplo de un lector RFID.

### c. Software de procesamiento:

Sistema informático conectado al lector, como se indica en la Fig. 2.13 , capaz de procesar los datos y gestionar su información en base a los requerimientos.

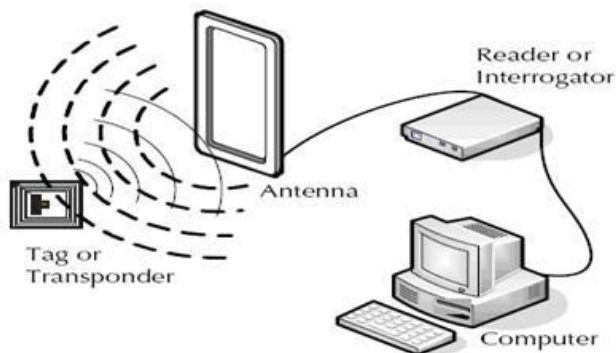


Fig. 2.13: Esquema general del sistema RFID.

La capacidad de identificación en tiempo real de objetos bajo circunstancias rigurosas que presenta este sistema, lo vuelve idóneo para la gestión de inventario. El hecho de que la detección se pueda hacer mediante la mera entrada del objeto al radio de lectura, sin contacto ni necesidad de ser alineado ópticamente, presenta una ventaja enorme con respecto a los sistemas de código de barras. Además, el uso de electrónica integrada para el almacenamiento de datos permite el uso de una enorme cantidad de memoria por ítem, característica que no solo es una ventaja en sí misma, sino que además presupone la posibilidad de identificar de forma única a cada ítem sin ningún problema, en lugar de repetir la codificación para cada serie de mercancía. Mejor aún, en base a esto es posible detectar incluso la ubicación de cada ítem en particular, dentro del alcance de los lectores.

Antes de usar cada tag RFID, se le escribe toda la información del producto al cual irá adherido. El lector RFID, conectado al software de procesamiento, ha de estar ubicado en el lugar de recepción de la mercancía, de tal forma que pueda detectar la llegada de esta, al igual que su eventual salida. De esta forma, se consigue una gestión de inventariado en tiempo real, rápida y eficaz de los productos.

Su única desventaja es el costo de implementación, que es bastante mayor al de los códigos de barra 1D y QR, sus principales competidores, ya que el costo de una antena, si bien no es excesivo por sí misma (medio dólar), no justifica su uso en productos de bajo costo. Sin embargo, sus prestaciones exceden con creces a las de los demás sistemas, y por tanto es altamente probable que los RFID solo requieran de una reducción en sus costos de implementación para dominar el mercado de los sistemas de control de inventario, sino también el de los sistemas de identificación en general, desarrollo que se encuentra actualmente en investigación.

## 2.04 Estado del arte

En el mercado internacional actual ya existen algunas empresas dedicadas a gestión de inventario que ofrecen sistemas de detección automatizados que permiten cuantificar la cantidad de alguna herramienta en particular, y que también pueden utilizarse para el control y detección de insertos. A continuación se hace un breve repaso a algunos productos comercializados por las principales empresas dedicadas al rubro.

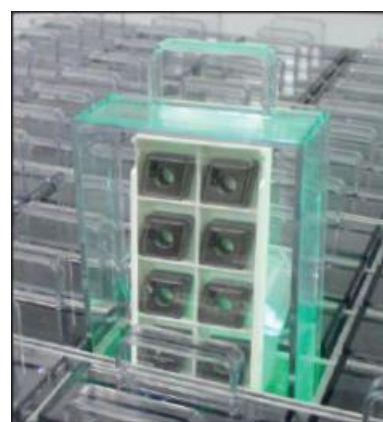
### 2.04.1 SUPPLYPRO

Empresa fundada en 1997, ubicada en San Diego, USA. Es una empresa dedicada a soluciones tecnológicas de inventario. Entre sus principales productos se encuentra:

- SmartDrawer: una gaveta cuyos cajones cuentan con múltiples compartimientos con cierre automatizado en la parte superior, controlados por software.
- SupplyScale: una gaveta con la misma tecnología de SmartDrawer, con la característica de que los compartimientos permiten detectar la presencia de ítems en base al peso.
- Single Box Dispenser II: una gaveta con compartimientos discretos dedicado al almacenamiento de cajas porta-insertos de corte en contenedores de plástico transparentes. Cuenta con indicadores LED que permiten señalar la ubicación del ítem a usar, como se observa en imagen comercial del producto (Fig. 2.14).



a)



b)

Fig. 2.14: Producto comercial SBDII, de SupplyPro. a) Viste completa del SBDII. b) Compartimiento iluminado de un contenedor de plástico iluminado, con una caja porta-insertos en su interior. Imágenes comerciales.

## 2.04.2 ELA INNOVATION

Fundada en 2000, ubicada en Montpellier, Francia. Es una empresa dedicada al desarrollo de sensores y antenas inalámbricas de larga distancia y bajo consumo, principalmente basados en *RFID activo* y *bluetooth*. Entre las principales soluciones orientadas a gestión de inventario que ofrece se encuentra:

- *Authomatic Inventory*: Sistema de etiquetas *RFID activas* que permiten no solo detectar a larga distancia la presencia de los ítems, sino también monitorear en tiempo real su ubicación dentro de recintos de gran tamaño.
- *Authomatic Tool Inventory*: Sistema de etiquetas *RFID activas* previamente instaladas en herramientas y gaveta contenedora, la cual cuenta con el monitor y software correspondiente. Permite monitorear la ubicación de las herramientas en tiempo real y alertar sobre los ítems faltantes. La gaveta puede apreciarse en la Fig. 2.15.



Fig. 2.15: Producto comercial *Authomatic Tool Inventory*, de ELA INNOVATION. Imagen comercial.

## 2.04.3 AUTOCRIB

Empresa fundada en 1994, ubicada en California, USA. Se dedica a fabricar sistemas de distribución de insumos para el área industrial (*industrial vending machines*). Entre sus productos se encuentran robustas máquinas, con capacidad para numerosas cantidades y gran tamaño de insumos a almacenar, cierre controlado por software, y monitor para el usuario. A continuación se presentan algunos de los principales productos comerciales que la empresa fabrica (Fig. 2.16).

- RoboCrib TX750: Una herramienta de gestión de inventario en formato de máquina expendedora para dispensar todo tipo de objetos.
- Autolocker FX Scale: Un sistema dispensador de objetos con tecnología de sensado basada en peso, capaz de identificar la existencia de pequeñas piezas. Cuenta con cierres de seguridad controlados.

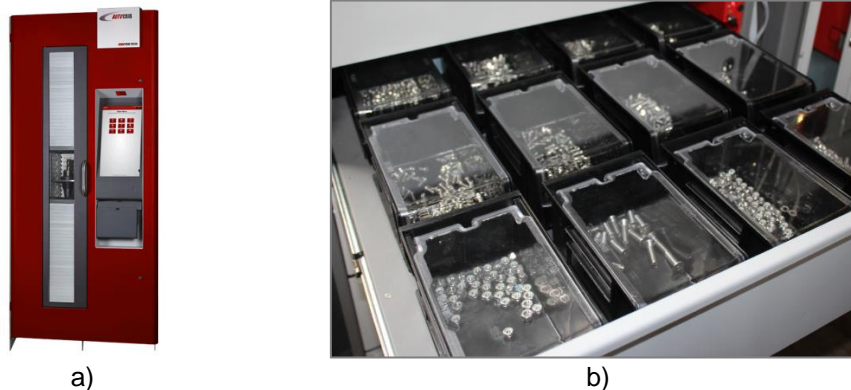


Fig. 2.16: Productos comerciales de AutoCrib. a) RoboCrib TX750. b) Vista de una de las gavetas del Autolocker FX Scale. Imágenes comerciales.

Como puede apreciarse, si bien la cantidad de fabricantes dedicados al área no es mucho más amplia que la mostrada en el anterior repaso, existen algunos productos comerciales capaces de satisfacer parcialmente la necesidad planteada, e incluso satisfacerla completamente, en el caso particular del *SBD II* de *Supply Pro*, que corresponde al único producto dedicado a insertos de torno en el mercado.

Cabe destacar que la tecnología de detección “basada en peso” equivale al uso de galgas extensiométricas como elemento sensor, información útil que podría validar el uso de este componente, al estar ya presente en el mercado del rubro.

Por otro lado, el costo de cualquiera de estos productos –asociado a su tecnología especializada, moneda del país procedente y dificultad de importación— es altísimo, y por tanto es difícilmente adquirido por maestranzas locales como *OrecaI*. De allí la necesidad de desarrollar nuevas tecnologías que permitan satisfacer la necesidad planteada, orientadas a la industria local.

## 2.05 Resumen Capítulo 2

Del estudio arrojado por el presente capítulo se obtiene una vista global del panorama actual que enmarca a los sistemas logísticos y comerciales de gestión de inventario. Esta vista ha de ser aplicada a resolver la problemática planteada en el Capítulo 1,

referente a los insertos de torno, en base a sus principales características, también estudiadas.

Es claro que para solucionar la problemática se requiere de un sistema de gestión de inventario perpetuo, el cual, a su vez, requiere de una tecnología electrónica adecuada.

De las tecnologías usadas en la actualidad para la gestión de inventario perpetuo, se concluye que las etiquetas RFID se presentan como la mejor alternativa para una detección automatizada. Sin embargo, su instalación en los insertos de torno, y su costo asociado, dificulta en gran medida su implementación, que por sí misma requeriría de un completo nuevo estudio, orientado no solo al área electrónica sino también metalúrgica. Por otro lado, la tecnología de sensado basada en peso, es decir, en base al uso de galgas extensiométricas, se presenta como una adecuada alternativa para el inventariado de insertos, respaldada por su uso en los productos comerciales de las pocas empresas en el mundo dedicadas al rubro de la gestión de inventario de herramientas.

Con todo esto, se continúa el estudio del presente sistema, teniendo en cuenta la posible utilización de la galga extensiométrica, pero sin descartar el uso de otras tecnologías de sensado que permitan dar una solución de gestión dedicada a los insertos de torno.

## CAPÍTULO 3: ESQUEMA GENERAL PARA LA ADQUISICIÓN DE DATOS

La particular naturaleza de la problemática, propia de un taller metalmecánico y por tanto poco explotable comercialmente, explica la inexistencia de un producto comercial dedicado a los insertos, que supla la necesidad planteada. Sin embargo, resulta intuitivo pensar que la tecnología actual pueda brindar una solución eficaz a la problemática. Es bajo estos términos que se plantea la idea de implementar un sistema electrónico que permita sensar de forma automática los insertos, de tal forma que la gestión del inventario sea más sencilla y efectiva.

El sistema electrónico a diseñar corresponde en un circuito de adquisición de datos, el cual consiste en la toma de muestras de las variables física reales, a partir de las cuales recoger la información necesaria. Para la problemática en particular, corresponde a un circuito que conste de un sensor que permita detectar la presencia y ausencia de los insertos, transformar dicho dato a una señal eléctrica, y acondicionarla de tal forma que un microcontrolador pueda procesarla y enviarla a la base de datos de un computador.

En el presente capítulo se plantean las consideraciones iniciales y se diseña un esquema general de adquisición de datos. Se subdivide en las siguientes partes:

1. Entorno: Se describe el entorno en el cual se trabaja actualmente en los talleres, se extraen de allí algunas consideraciones y se sientan las bases para el posterior diseño físico del prototipo.
2. Esquema general: Se plantea un diagrama en bloque, en términos generales, sobre el cual diseñar la etapa de adquisición de datos.
3. Diagrama general: Se presenta un diagrama en bloque resultante del funcionamiento general descrito.



### 3.01 Entorno

Los fabricantes de insertos de corte venden y distribuyen estas herramientas en pequeñas cajas de plástico “porta-inserto” que facilitan su transporte. Tal como puede observarse en las Fig. 3.1 y Fig. 3.2, cada caja cuenta con diez compartimientos iguales, que permiten el almacenamiento de un solo inserto por compartimiento, pudiendo almacenarse hasta diez de ellos por caja.



Fig. 3.1: Dos cajas contenedoras para los insertos, una grande y una pequeña. La caja de la parte superior corresponde a la mencionada, con compartimientos de 28x28 [mm], mientras que la pequeña es de 19x19 [mm].



Fig. 3.2: Caja contenedora de 28x28 [mm], con insertos en su interior.

Las dimensiones de cada compartimiento dependen del tamaño de la caja. Existen varios tamaños estandarizados que agrupan varias clases de inserto según su tamaño, pero en los talleres de Orecal se utilizan principalmente dos tamaños de porta-inserto: con compartimientos de 28x28 milímetros (Fig. 3.4), y de 19x19 milímetros (Fig. 3.4). A su vez, la altura de cada compartimiento es de 10 y 8 milímetros, respectivamente. Mayores de talles sobre los porta-insertos pueden apreciarse en las figuras.

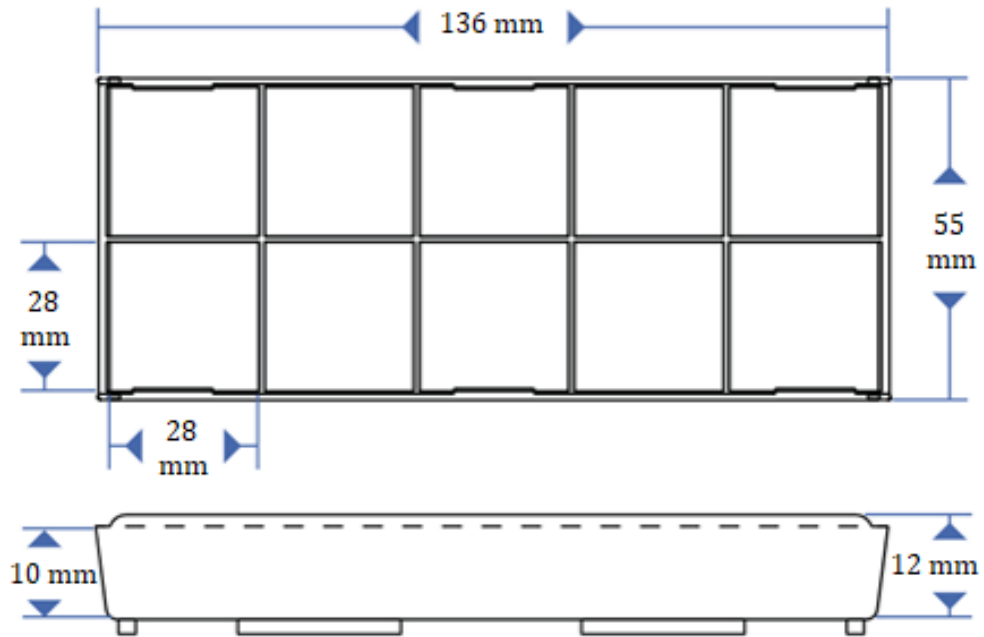


Fig. 3.3: Dimensiones de caja porta-insertos con compartimientos de 28x28 [mm].

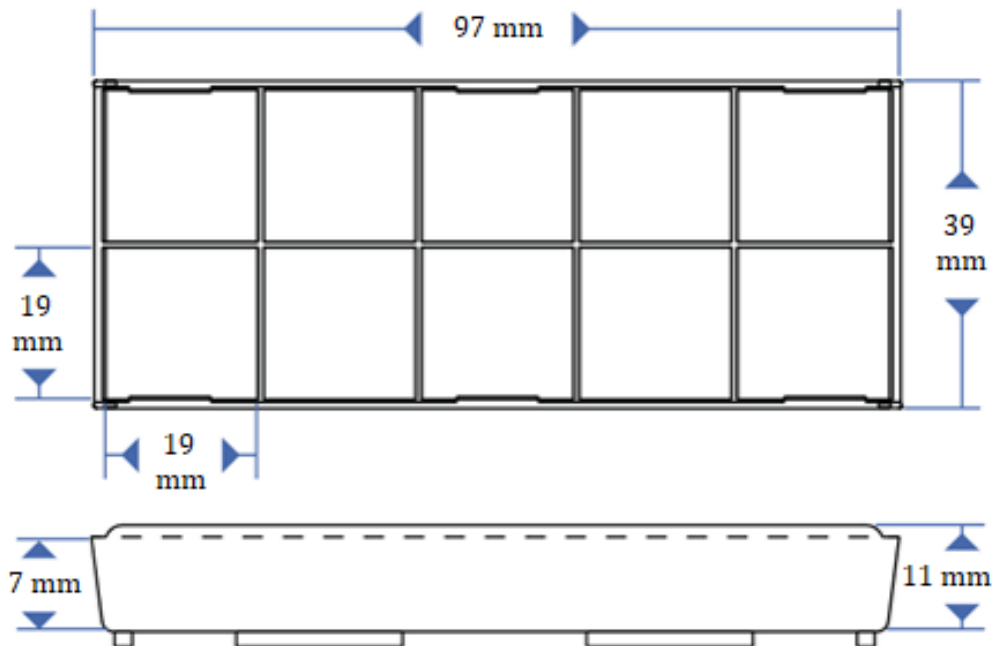


Fig. 3.4: Dimensiones de caja porta-insertos con compartimientos de 19 [mm].

Cabe mencionar que las cajas cuentan con una etiqueta, adherida por el fabricante al reverso de cada una, que señala diversos parámetros, propios de cada inserto, según la nomenclatura estandarizada. De momento estos parámetros contenidos en la etiqueta resultan de gran importancia al momento de identificar el modelo de inserto necesario para una aplicación particular. Por esta razón, los talleres metalmecánicos, en general, suelen conservar los insertos en sus respectivas cajas para poder identificar con facilidad las herramientas; situación que, por supuesto, también se da en los talleres de Orecal.

A su vez, los porta-insertos, ya sea que se encuentren llenos o parcialmente llenos, son almacenados -sin mucho orden- en las gavetas de muebles de uso industrial, que a su vez almacenan otras herramientas de toda clase. Estos muebles no solo están constituidos de madera, sino también de plástico y metal. Ejemplos de ellos pueden apreciarse en la Fig. 3.5.



Fig. 3.5: Muebles usados en el taller.

De las condiciones anteriormente expuestas puede extraerse las siguientes conclusiones, de importancia para el posterior diseño:

- a. Cada inserto es almacenado dentro de unas dimensiones de 28x28x10 [mm] y 20x20x8 [mm], respectivamente, dimensiones correspondientes a los compartimientos de los dos tipos de caja porta-insertos utilizadas por Orecal.
- b. Los compartimientos se distribuyen en forma de matriz de 2x5, pudiendo almacenarse hasta 10 insertos por cada caja porta-insertos.
- c. Las cajas porta-insertos se guardan en gavetas constituidas de madera, plástico y metal, en conjunto a otras herramientas, desordenadamente.

## 3.02 Funcionamiento general

Este capítulo tiene como objetivo trazar un esquema general de las etapas que conformarán al sistema de adquisición de datos y definir su funcionamiento, desde el sensado de los insertos hasta su comunicación con la base de datos. Se estudia cada una de las etapas previstas, caso a caso, para finalizar con un diagrama de flujo que representa en términos generales el funcionamiento del sistema.

### 3.02.1 Microcontrolador

El sistema de adquisición, para ser automático, requiere del uso de un microcontrolador que funcione en base a un algoritmo que permita automatizar el proceso de sensado. A él se conectan todos los sensores y actuadores del sistema, de tal forma que el algoritmo pueda realizar todas las tareas programadas de forma secuencial. Con este fin, se apunta a utilizar un microcontrolador de bajo costo que permita un rápido prototipaje del proyecto.

Existe una buena variedad de microcontroladores en el mercado, y más aún, abundan placas de desarrollo debido a su gran potencial para el diseño de prototipos. Módulos como Arduino, Raspberry Pi o Tessel (Fig. XX), debido a sus amigables softwares de código abierto y amplia documentación de respaldo, han hecho crecer el mundo del desarrollo *Do It Yourself* y del *Internet of Things*, planteándose como alternativas de aprendizaje mucho más accesibles frente a los microcontroladores integrados, como por ejemplo PIC de Microchip, si bien fundamentalmente se utilizan para los mismos propósitos.

Para efectos del desarrollo de este proyecto, se decide utilizar la placa de desarrollo Arduino UNO Rev. 3, debido a su sencillez de uso y precio. No obstante, el circuito final hace uso del microcontrolador ATmega328P-PU de la marca ATMEL, el cual se puede apreciar en la Fig. 3.6.



Fig. 3.6: Microcontrolador ATMEL ATmega328P-PU.

### 3.02.2 Acondicionamiento

Además del microcontrolador, se tiene la etapa de los sensores. Esta etapa consta de cada uno de los dispositivos que detecta la presencia de los insertos. Sin embargo, los sensores requieren también de una etapa de acondicionamiento, la cual ha de adaptar el parámetro eléctrico captado por el sensor, sea cual sea, a una señal de tensión TTL, que pueda ser leída por el microcontrolador. Dependiendo del sensor a utilizar, esta etapa de acondicionamiento podría incluir circuitos como un amplificador, sumador, puente resistivo, entre otras sub-etapas. No obstante, su uso y diseño queda pendiente hasta definir adecuadamente el sensor.

Por otro lado, es importante recordar que la cantidad de insertos a sensar, al ser parte de un stock almacenado, es bastante numerosa. En este punto resulta intuitivo pensar que, para una gran cantidad de objetos, se requiere también una gran cantidad de sensores, y por consiguiente, una gran cantidad de GPIOs (*General Purpose Input/Output*, entradas y salidas de propósito general). Esto se ve traducido en la utilización de un microcontrolador con las suficientes prestaciones. Sin embargo, suponiendo que la cantidad de sensores a utilizar sea mayor a cien, esta situación no solo supondría una dificultad en el hecho de conseguir un microcontrolador con tales prestaciones, sino que además es muy poco eficiente. Ahora bien, asumiendo que no es necesario recoger datos de los sensores de manera ininterrumpida, es posible reducir significativamente la cantidad de GPIOs utilizadas mediante una etapa de multiplexado.

La multiplexación es la técnica mediante la cual se combinan varios canales de información en uno solo, pudiendo seleccionar el canal deseado mediante un arreglo de *bites*. Se lleva a cabo por medio de un multiplexor, y electrónicamente corresponde a una sub-etapa de la etapa de acondicionamiento. De esta forma se reduce notoriamente la cantidad de GPIOs a utilizar, para así prescindir de un microcontrolador demasiado costoso, y al mismo tiempo reducir el consumo eléctrico.

La cantidad de GPIOs a utilizar queda determinada por la cantidad de canales, es decir, la cantidad de sensores a utilizar, ya que el número de pines de control del multiplexor está dado por el logaritmo en base dos del número de canales, a lo cual se le suma una unidad, que corresponde al pin de salida, que arroja la señal del canal seleccionado. Matemáticamente:

$$GPIOs = \log_2(\text{Canales}) + 1 = \log_2(32) + 1 = 6 \text{ GPIO} \quad (3.1)$$

Así, por ejemplo, si se requiere gestionar una cantidad máxima de 128 insertos, es decir, 128 sensores, entonces mediante el uso de una etapa de multiplexado se ha de requerir de tan solo 8 GPIOs del microcontrolador, lo cual permite prescindir de la significativa cantidad de 128 GPIOs. Siguiendo esta lógica, es posible escalar en gran medida la cantidad de insertos a sensar, lo cual resulta muy útil para una aplicación de gestión de inventario, que suele trabajar con gran cantidad de ítems.

Para efectos del diseño de prototipo, la cantidad de insertos a sensar no ha de ser necesariamente la cantidad real que se podría requerir en terreno, que debiera ser de

los cientos de ítems. Sin embargo, igualmente se busca validar la idea del manejo de una cantidad numerosa de ítems, razón por la cual no se diseñará para solo uno. Bajo estos criterios, se considera que 32 insertos a sensar es una cantidad razonable, y por tanto la etapa de acondicionamiento, y en particular, la sub-etapa de multiplexación, así como el resto del sistema, se diseña para esta cantidad de ítems. Se sobreentiende que esta cantidad sería, sin mayores complicaciones, escalable hasta los cientos de ítems para un producto comercial que disponga de mayores recursos.

### 3.02.3 Interfaz

Un sistema de gestión de inventario entrega información útil al usuario para que este pueda tomar decisiones, y lo puede hacer por medio de varios tipos de interfaz. Resulta esencial que uno de estos medios sea un computador, ya que el usuario debe poder consultar el registro de datos recogidos de forma clara y ordenada, con tal de poder gestionar bien el inventario, prestación que solo un ordenador puede cumplir a cabalidad.

Así, se plantean los dos medios siguientes:

- **Software:** Una interfaz de consulta y control para el usuario final. La información que trabaja es la que una base de datos le entrega, la cual, a su vez, es alimentada por el sistema de adquisición. Esta etapa, puramente informática, no es desarrollada en el presente estudio.
- **Pantalla LCD:** Un indicador, instalado en terreno, que entregue en tiempo real la información recogida por el sistema, de tal forma que los operarios puedan ver el registro al hacer cualquier cambio sobre el stock. Una pantalla LCD de 16x2, como la que se aprecia en la Fig. 3.7, es una buena opción para mostrar rápida y eficazmente dichos cambios.

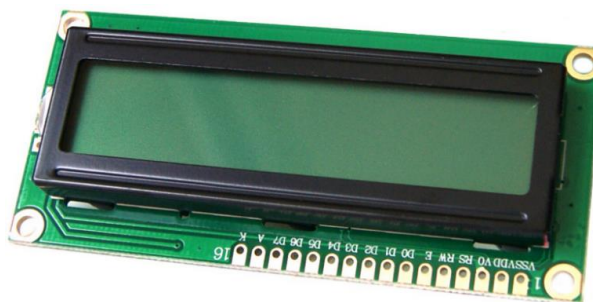


Fig. 3.7: Pantalla LCD 16x2. Muestra hasta 32 caracteres repartidos en 2 filas y 16 columnas.

### 3.02.4 Comunicación

Los datos recogidos por el microcontrolador son procesados para poder interpretar su información, la cual ha de alimentar la base de datos de un software para inventarios, y



para conseguirlo es necesario comunicar de algún modo dicha información a un computador. La comunicación a un computador puede hacerse por varios métodos, típicamente: alámbrico, *Bluetooth* o *Wi-Fi*. La decisión de usar un método u otro se basa en las condiciones a las que está sometido el sistema.

Considerando que el sistema electrónico se ha de encontrar en la gaveta de un mueble en el interior de un taller, la idea de usar comunicación alámbrica o bluetooth resulta poco convincente ya que supondría la necesidad de contar de forma permanente con un computador encendido a muy pocos metros, que pueda sostener la base de datos, situación que acota la flexibilidad del prototipo en cuanto a la distancia.

En cambio, el uso de Wi-Fi en comunicación directa con la base de datos presupone tan solo la condición de contar con una conexión permanente a redes Wi-Fi dentro del recinto en donde se encuentre instalado, lo cual contribuye a la versatilidad del sistema, en tanto que el mueble podría ubicarse sin problemas en cualquier lugar del recinto donde haya presente una señal de Wi-Fi estable. Aun mejor, un correcto uso del microcontrolador podría permitir que la comunicación se mantenga incluso si la señal presente es inestable.

Actualmente, gracias al desarrollo de tecnologías del *IoT*, como por ejemplo los chips Wi-Fi ESP8266 de bajo costo desarrollados por *Espressif*, es posible implementar la comunicación sin incurrir en elevados costes. Existe una gran variedad de módulos basados en el ESP8266, entre los cuales destacan los SoC (*System of a Chip*, Sistema en Chip) ESP-01, su versión mejorada ESP-12 y también la placa de desarrollo NodeMCU, cuyos módulos pueden apreciarse en la Fig. 3.8.

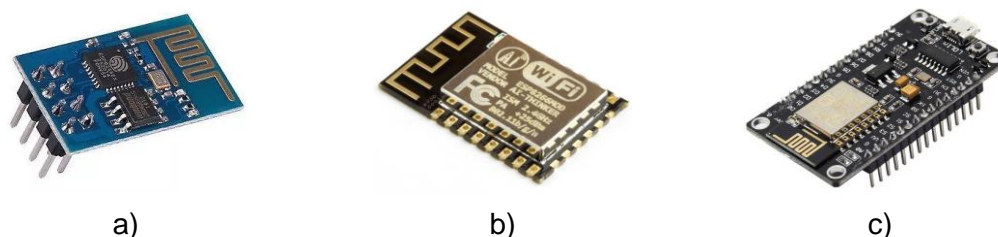


Fig. 3.8: Módulos basados en el chip ESP8266: a) SoC ESP-01. b) SoC ESP-12. c) Development board NodeMCU.

Para esta etapa de comunicación se decide utilizar la tarjeta de desarrollo *NodeMCU V1.0*, ya que es más versátil y agiliza el desarrollo del sistema. Cabe señalar que un desarrollo posterior al presente proyecto pudiese considerar el uso del SoC ESP-12, ya que optimiza el consumo del sistema, y en cuanto a sus características de operación, ofrece las mismas que las del NodeMCU, ya que, de hecho, este último está basado en el propio ESP-12.

### 3.03 Diagrama general

En base a las etapas previstas, es posible trazar tempranamente, tal como se indica en la Fig. 3.9, un esquema general del funcionamiento del sistema. Este será posteriormente desarrollado en detalle, en función a los requerimientos eléctricos y electrónicos de cada etapa, y también en base al sensor a utilizar.

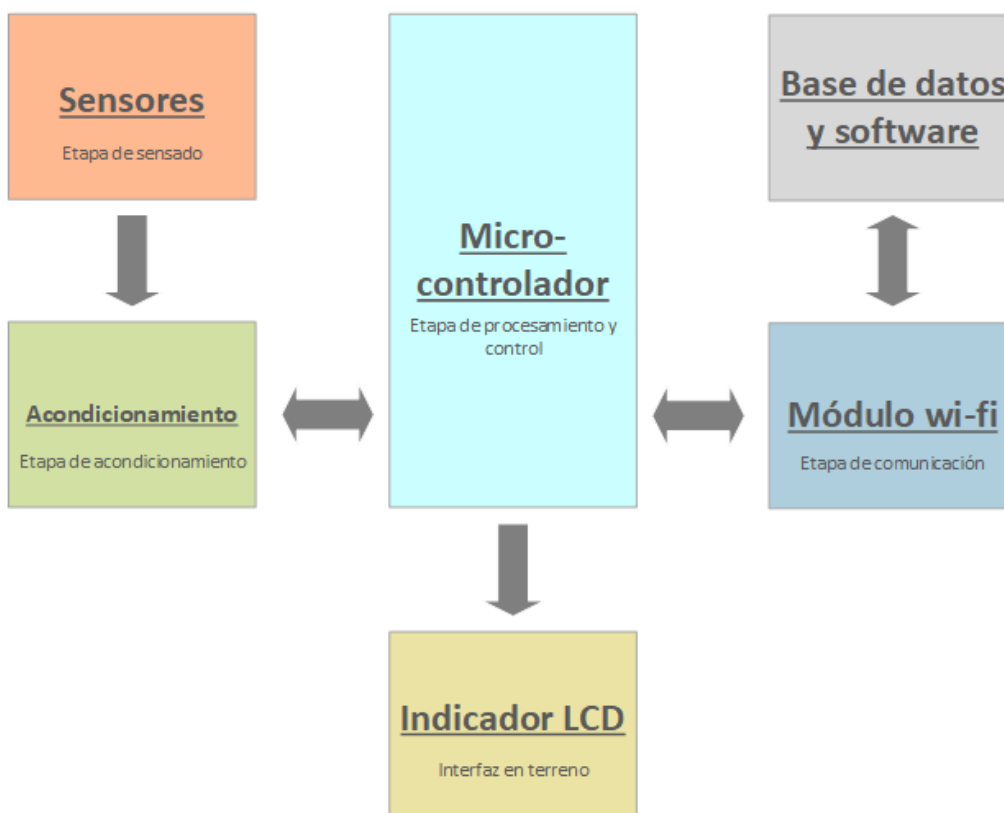


Fig. 3.9: Esquema general del sistema. Cada bloque representa una etapa.

Como puede apreciarse, el microcontrolador juega un papel central en el funcionamiento. Por otro lado, el sensor también representa un papel clave en el circuito, ya que de este depende, en gran medida, toda la etapa de acondicionamiento. Ambas etapas son estudiadas y diseñadas posteriormente.



## CAPÍTULO 4: PRINCIPIO DE FUNCIONAMIENTO

Para conseguir que el sistema de adquisición de datos funcione tal como se ha diseñado, es crucial contar con un sensor adecuado, más aun considerando las particulares características de la problemática. No solo se requiere de un sensor que detecte la herramienta, sino que además se adecúe a las limitaciones físicas y permita sensar un gran número de objetos.

En el presente capítulo se estudia cuál sensor utilizar y bajo qué condiciones utilizarlo. Se subdivide en las siguientes partes:

1. Sensor: Se repasa brevemente lo que se espera del sensor a utilizar.
2. Potenciales sensores: Se hace un breve repaso a los principales sensores de potencial uso, y se selecciona uno que se adecue a la aplicación, sobre el cual diseñar el sistema.
3. Selección del sensor: Se explica cuál sensor se decide utilizar.
4. Principio de funcionamiento: Se plantean las condiciones bajo las cuales utilizar el sensor seleccionado.
5. Requerimientos: Se exponen los requisitos que permiten utilizar el principio de funcionamiento planteado.

### 4.01 Sensor

Un sensor es un objeto capaz de modificar alguno de sus parámetros en función de alguna magnitud del exterior. En un contexto eléctrico-electrónico, se trabaja con sensores que modifiquen alguno de sus parámetros eléctricos con tal de obtener señales eléctricas que permitan comunicar al sistema con su entorno. Así es como existe una gran variedad de sensores para captar todo tipo de magnitudes físicas, y entregar diferentes parámetros eléctricos que de un modo u otro han de ser interpretados por el sistema.

Para efectos de la aplicación, el parámetro que el sensor debe ser capaz de medir ha de ser un parámetro físico que dependa de la presencia del inserto en el compartimiento de la caja. Existen diferentes sensores que pueden detectar la presencia/ausencia de los insertos. Entre ellos, son los sensores de proximidad los que mejor cumplen esta tarea (debido a su capacidad de detección sin entrar en contacto con el objeto) aunque no son los únicos. Los sensores ópticos y los de fuerza (peso) también presentan buenas alternativas, cada uno con sus propias ventajas y desventajas.

Por otro lado, es necesario tomar en cuenta las condiciones del entorno en el cual trabajará el sensor. Aspectos como el modo de funcionamiento en espacios reducidos, dimensiones, y consumo, además del costo asociado a cada uno, juegan un papel crucial en el

desempeño global del sistema. De ahí la importancia de seleccionar adecuadamente un sensor que ofrezca un fácil modo de uso, sin menoscabar su efectividad ni explotar su costo de implementación.

## 4.02 Sensores potenciales

A continuación, se hace un breve repaso de los principales sensores que podrían cumplir correctamente la función de detectar la presencia de un inserto de torno, en base a las condiciones del objeto a sensar y del entorno, con el objetivo de presentar las mejores alternativas en base a sus ventajas y desventajas, además de poder fundamentar el uso de una de ellas para el desarrollo del proyecto.

**Sensor de Proximidad Inductivo:** Este sensor es principalmente utilizado en la industria para la detección de metales, aunque también es capaz de detectar objetos no metálicos a una muy corta distancia, dependiendo del material. Suelen ser cilíndricos y alargados, de unos cuantos centímetros de longitud, como se muestra en la Fig. 4.1, aunque también los hay rectangular, cúbicos, etc.

Un sensor inductivo contiene en su interior un oscilador que consta de un devanado o bobina. A este devanado se le hace circular una corriente, generando así un campo electromagnético en torno a la cara frontal del sensor, como puede apreciarse en la Fig. 4.2. El alcance de este campo magnético es entregado por el fabricante y suele ir desde fracciones de milímetro hasta los 40 milímetros. Es importante señalar que la distancia de detección varía sustancialmente con el blindaje de protección, como puede verse en la Fig. 4.3.



a)



b)

Fig. 4.1: Sensores de proximidad inductivos.

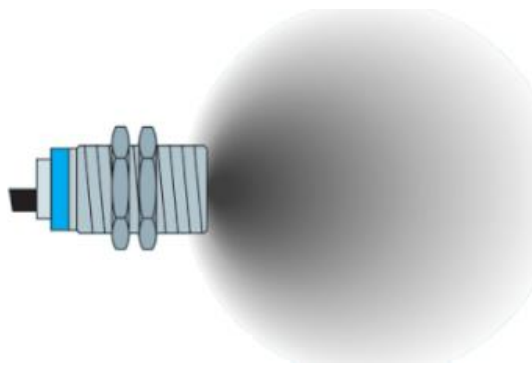


Fig. 4.2: Esquema del campo electromagnético generado por el sensor.

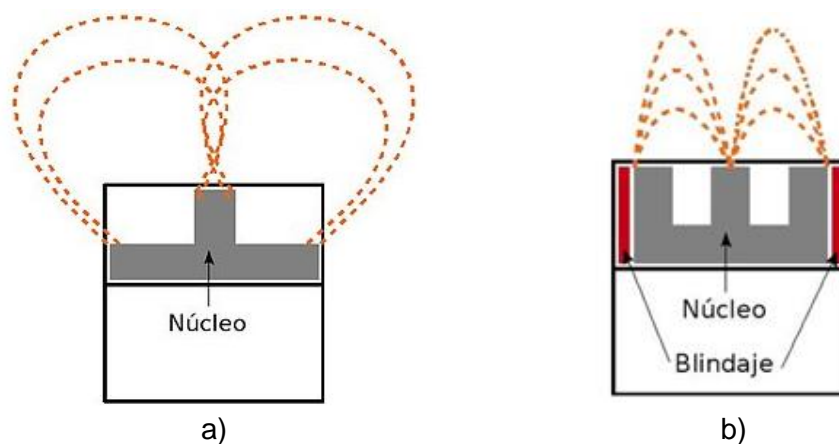


Fig. 4.3: Campo electromagnético. a) Sensor no blindado. b) Sensor blindado.

Cuando un objeto se aproxima a la cara frontal, este entra en contacto con el campo magnético del sensor y entonces se ve expuesto a corrientes de Foucault o parásitas, que a su vez también generan un campo electromagnético alrededor del objeto. La intensidad de estas corrientes parásitas y campo magnético inducido depende de la conductividad, permeabilidad, distancia y dimensiones del objeto en cuestión. El campo magnético del objeto es opuesto al del sensor, y como resultado, se ve reducida la impedancia y, por consiguiente, la inductancia del devanado del sensor, y así mismo la amplitud de la oscilación. Este fenómeno es su principio de funcionamiento, y es aprovechado para detectar la presencia de objetos a cierta distancia, dentro del campo magnético generado por la bobina.

Los materiales metálicos son más susceptibles a las corrientes parásitas y por eso este tipo de sensores resultan mucho más efectivos en la detección de objetos ferrosos. Algunos materiales no ferrosos también son detectados, pero a una distancia mucho menor.

A continuación, se resumen las principales consideraciones a tomar en cuenta en la utilización de sensores inductivos. Además, en la Tabla 1 se presentan las principales ventajas y desventajas de utilizar este sensor.

- El material del objeto a detectar debe ser metálico.
- La superficie del objeto a detectar debe ser mayor al diámetro de la cara frontal del sensor.
- Las dimensiones del sensor varían según su modelo pero suelen ser de varios centímetros de largo y ancho.

Tabla 1: Principales ventajas y desventajas del sensor inductivo.

Ventajas	Desventajas
<ul style="list-style-type: none"> <li>• Detección sin contacto con el objeto.</li> <li>• No se desgastan. Tiempo de vida largo, independiente del número de detecciones.</li> <li>• Capacidad para ignorar o “ver a través” de ciertos materiales no metálicos como el plástico.</li> <li>• Rápida velocidad de respuesta, del orden de los microsegundos.</li> <li>• Detección insensible al polvo y humedad.</li> </ul>	<ul style="list-style-type: none"> <li>• Objeto de detección muy limitado, casi restringido a materiales metálicos.</li> <li>• Sensibilidad a campos electromagnéticos intensos.</li> <li>• Tamaño considerable. Varios centímetros de ancho y largo.</li> </ul>

- a. **Fotorresistencia:** También llamadas foto-resistores o LDR (*Light-Dependent Resistor*) son componentes electrónicos que consisten en una celda fotoeléctrica cuya resistencia eléctrica varía en función de la intensidad de luz incidente. Son fabricados con semiconductores de estructura cristalina y alta resistencia como el sulfuro de cadmio o el seleniuro de cadmio. El uso de los LDR no es tan extendido debido a su lento tiempo de respuesta de alrededor de 0.1 [s], pero suelen usarse en aplicaciones de activado/apagado nocturno, donde se necesita detectar la presencia de la luz del día para activar un relé o sistema de iluminación. En la Fig. 4.4 puede apreciarse un LDR común y su esquema eléctrico.

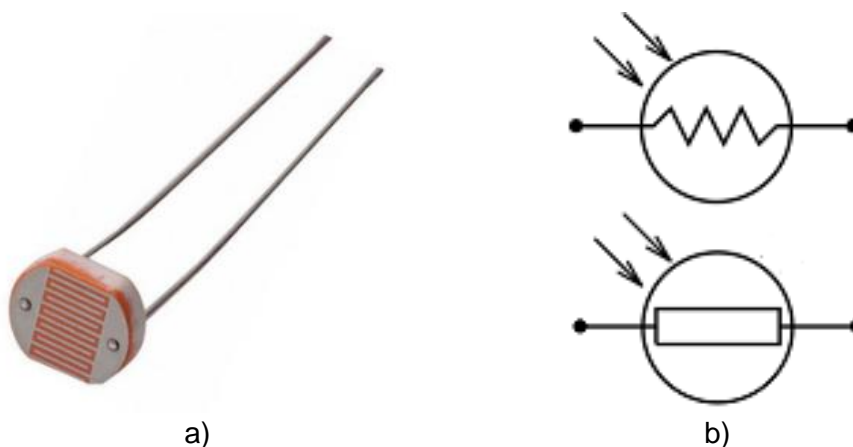


Fig. 4.4: Foto-resistor: a) Vista de un LDR convencional. b) Símbolos eléctricos usuales.

Su principio de funcionamiento está basado en el efecto fotoeléctrico, que consiste en la emisión de electrones por parte de un material cuando incide luz sobre él. Dentro de este efecto se engloban las subcategorías del efecto fotovoltaico y la fotoconductividad, siendo esta última el principio específico sobre el cual trabajan los LDR. La resistencia o conductividad varía en función de la intensidad lumínica: a mayor luz, mayor conductividad, es decir, menor resistencia. Los valores típicos entre los cuales oscila un LDR suele ir desde cientos de ohmios en condiciones de iluminación total hasta varios mega ohmios en oscuridad total.

Es importante señalar algunas desventajas en el uso de LDR, como por ejemplo que la rapidez de respuesta no es instantánea, es decir, existe un tiempo de respuesta de alrededor de 0.1 [s] para que el cambio desde buena iluminación a completa oscuridad se vea reflejado en su resistencia. Además no presentan mucha precisión en sus valores de resistencia, esto es, al repetirse un mismo estado de intensidad lumínica.

A continuación, en la tabla de la Tabla 2 se muestran las principales ventajas y desventajas de utilizar sensor LDR.

Tabla 2: Principales ventajas y desventajas de los foto-resistores.

Ventajas	Desventajas
<ul style="list-style-type: none"> <li>• Tamaño muy reducido.</li> <li>• Alta sensibilidad.</li> <li>• Instalación y empleo sencillo.</li> <li>• Muy bajo costo.</li> </ul>	<ul style="list-style-type: none"> <li>• Dependiente de la estabilidad lumínica del entorno.</li> <li>• Respuesta espectral estrecha.</li> <li>• Efecto de histéresis.</li> <li>• Tiempo de respuesta lento.</li> <li>• Respuesta poco precisa para un mismo estado lumínico.</li> </ul>

- b. **Galga extensiométrica:** Es un transductor sensible a las fuerzas de compresión, tracción y torsión, es decir, es utilizado para medir la fuerza, deformación, presión o carga aplicada sobre el él. Su principio de funcionamiento está basado en el efecto piezorresistivo, que corresponde a la capacidad de algunos materiales para cambiar su valor de resistencia eléctrica cuando este es sometido a una deformación física. Estos sensores son ampliamente utilizados en la industria e ingeniería de investigación para la medición de magnitudes mecánicas debido a sus pequeñas dimensiones, inmunidad a las ondas electromagnéticas, versatilidad y bajo costo.

Las galgas suelen constituirse de un estampado de lámina metálica fijada a una base aislante y flexible que pueda ser adherido a la superficie sobre la cual se desee estudiar una fuerza mecánica. El estampado metálico suele consistir de un alambre muy fino dispuesto en forma de rejilla, como se muestra en la Fig. 4.5.

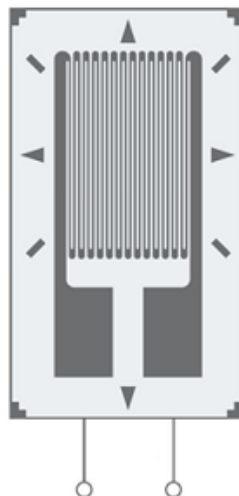


Fig. 4.5: Esquema del estampado metálico de una galga extensiométrica común.

La galga, o más específicamente, la lámina metálica, al recibir una fuerza se deforma y produce una variación en su resistencia eléctrica, parámetro que se ve reflejado en tensión o corriente mediante un circuito de acondicionamiento, pudiendo así sensarse la fuerza aplicada.

De forma general existen dos tipos de galga extensiométrica:

- **De hilo conductor:** Las más comúnmente utilizadas. Se trata de una base delgada aislante y flexible sobre la cual va adherido una lámina o hilo metálico muy fino en un cierto patrón. Entre sus características se tiene que suelen tener un factor de galga de aproximadamente 2, baja resistencia y buena linealidad. En la Fig. 4.6 se muestra una típica galga de hilo conductor.

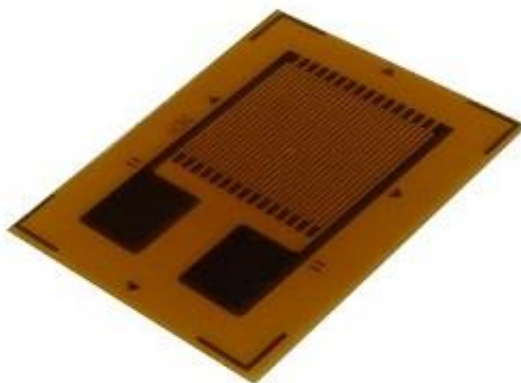


Fig. 4.6: Ejemplo de galga extensiométrica de hilo conductor.

- **Semiconductor:** Como su nombre lo indica, están fabricadas con hilo de silicón semiconductor. Tienen factores de galga mayor a 50, mucho mayor al de las galgas de hilo conductor, pero en contraste son el tipo que se ve más afectado por la

temperatura. Además sus valores de resistencia también tienden a ser mayores que las de hilo conductor. En la Fig. 4.7 se muestra una típica galga de semiconductor.



Fig. 4.7: Ejemplo de galga extensiométrica de semiconductor.

El uso de una u otra está ligado a las condiciones de la aplicación que se requiera. Es importante añadir que entre las desventajas que supone el uso de galgas está su baja resistencia a las altas temperaturas, pudiendo en su mayoría soportar no más de 150 °C. Sus principales características se señalan en la Tabla 3.

Tabla 3: Principales ventajas y desventajas de las galgas extensiométricas.

Ventajas	Desventajas
<ul style="list-style-type: none"> <li>• Tamaño muy reducido.</li> <li>• Capacidad para detectar todo tipo de objetos dentro del rango de peso.</li> <li>• Inmunidad electromagnética.</li> <li>• Alimentación con corriente continua o alterna.</li> <li>• Excelente respuesta en frecuencia.</li> </ul>	<ul style="list-style-type: none"> <li>• Señal de salida débil.</li> <li>• Muy sensible a las vibraciones.</li> <li>• Muy sensible a temperaturas altas.</li> </ul>

### 4.03 Selección del sensor

Teniendo en cuenta que todos los sensores anteriormente expuestos satisfacen, en términos generales, las condiciones impuestas por el objeto a sensar y su entorno, se decide seleccionar el sensor en base a sus costos asociados. Luego de cotizar en el mercado los diversos precios que se ofrecen, así como el de los dispositivos necesarios para la implementación de cada uno, se llega a la conclusión de que el sensor más económico es la **foto-resistencia** o **LDR**, el cual es por el lejos el más económico, y no por ello menos efectivo, sin mencionar que sus dimensiones se adecúan a los entornos reducidos. Se procede a plantear entonces un principio o modo de funcionamiento basado en el uso de la LDR.

#### 4.04 Principio de funcionamiento

Partiendo de la premisa de que la LDR corresponde a un sensor analógico sensible a la luz incidente, y considerando que el área de un inserto es mayor a la de la cara de la LDR, resulta intuitivo pensar que el principio de funcionamiento ha de estar basado en posicionar al inserto por encima de la LDR, esta última en la superficie del compartimiento. De esta forma, la sombra que el inserto proyecta sobre la LDR al encontrarse presente, y a su vez, la luz que incide al encontrarse ausente, tal como se muestra en la Fig. 4.8, permite obtener dos estados diferentes, captados por el sensor.

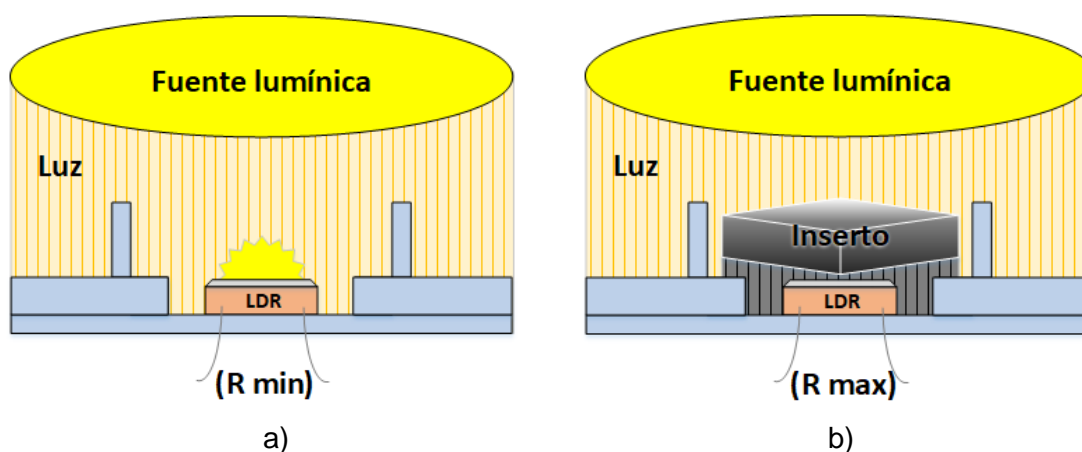


Fig. 4.8: Estados del sensor en vista lateral: a) Inserto ausente, plena luz, baja resistencia. b) Inserto presente, nula luz, alta resistencia.

Estos dos estados, de plena y nula luz incidentes, se traducen en un abrupto cambio de resistencia por parte de la LDR. Para el estado de plena luz incidente, el LDR presenta un mínimo valor resistivo, del orden de los cientos de ohm, mientras que, para el estado de oscuridad o nula luz, presenta su máximo valor resistivo, del orden de los mega-ohm. Esta situación de abrupto cambio resistivo resulta muy favorable, ya que no solo es efectivo para detectar la presencia del objeto, sino que además puede aprovecharse para discretizar la señal.

El método descrito ha de usarse para el sensado individual de cada inserto. Por lo tanto, no existiendo la posibilidad de apilarse insertos, uno sobre otro, dentro de un mismo compartimiento, la manera más adecuada de replicar el método es distribuyendo a los insertos –y por tanto también a los sensores– en forma de matriz, como una versión extendida de los compartimientos que traen las cajas porta-insertos. Esta distribución, como se observa en la Fig. 4.9, habría de estar instalada, de forma fija, sobre la superficie del interior de gaveta, ordenada como una matriz de compartimientos donde colocar a cada uno de los insertos, y donde hacer también el sensado.



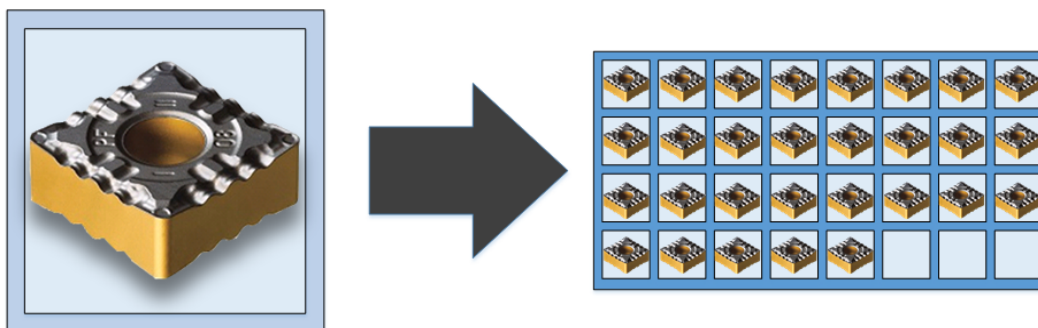


Fig. 4.9: Disposición de los compartimentos en forma de matriz, instalada en la base de la gaveta.

Sin embargo, resulta problemático el hecho de que el objeto a sensor –el inserto de tornó– se encuentre dentro de una gaveta, cuyo interior, por naturaleza, se ve expuesto a un persistente cambio en su iluminación con cada apertura y cierre. Es decir, se tiene que el interior de la gaveta, al estar cerrada, percibe nula o mínima iluminación, mientras que, al estar abierta, hay plena o mucha iluminación. La situación se acompleja aún más al considerar que la iluminación en estado de apertura depende de la iluminación del ambiente, la cual difícilmente podría controlarse ni tomarse como una referencia ya que puede ser variable y afectar la medición, y si bien esto podría solucionarse con una LDR dedicada solo a sensor la iluminación de ambiente y usar su lectura como referencia, hay que considerar también las variaciones de luz producidas por las sombras que proyectarían los usuarios al acercarse a la gaveta. En última instancia, se necesitaría un poderoso algoritmo de inteligencia artificial para mantener este método, lo cual de inmediato vuelve mucho más compleja la tarea. Por lo tanto, se plantea la necesidad de agregar una condición adicional que permita sortear dicha problemática.

Ahora bien, es importante considerar lo siguiente: Los cambios en el stock, es decir, la adición o sustracción de insertos, se realiza con cierta frecuencia, y por tanto es claro que el sistema debe poder sensor en tiempo real. Más específicamente, los cambios en el stock son eventos puntuales que se llevan a cabo en un momento en particular, y que tienen como acción asociada la intervención de la gaveta. Como consecuencia, quedan ciertos períodos de tiempo en los cuales el sistema se encuentra en “estado de reposo” o *standby*, como se señala en la Fig. 4.10, ya que no se realiza ningún cambio al stock.

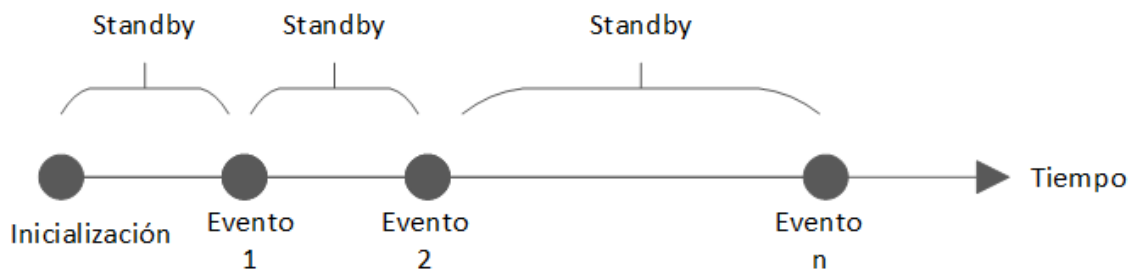


Fig. 4.10: Comportamiento de los eventos de intervención al stock en el tiempo.

Esta situación, tal como se había adelantado anteriormente al introducir la sub-etapa de multiplexación, deja implícita una importante ventaja: no es necesario sensar ininterrumpidamente. Así, es posible redefinir el principio de funcionamiento del sistema.

Como no es necesario sensar ininterrumpidamente, basta con definir un momento adecuado para hacerlo, que claramente debe ser en evento de intervención al stock. Entonces la acción de sensado queda limitada a hacerse durante un evento de apertura o cierre de la gaveta. Pero en este punto de decisión, de nuevo es importante recalcar la dificultad que presupone llevar a cabo el sensado con la iluminación externa, ya que esta resulta demasiado inestable.

Todo lo anteriormente señalado representa el fundamento para la siguiente solución: sensar solo con la gaveta cerrada, utilizando iluminación controlada interna. Programar al microcontrolador para que obtenga la lectura de los sensores solo cuando la gaveta se encuentra cerrada, permite aislar lumínicamente a los sensores, evitando así todas las interferencias que pudieren producirse debido al uso de la iluminación local. A su vez, estando el interior de la gaveta en completa oscuridad, es necesario la utilización de iluminación interna que permita a los LDR hacer su trabajo, en una disposición como la que se aprecia en la Fig. 4.11.

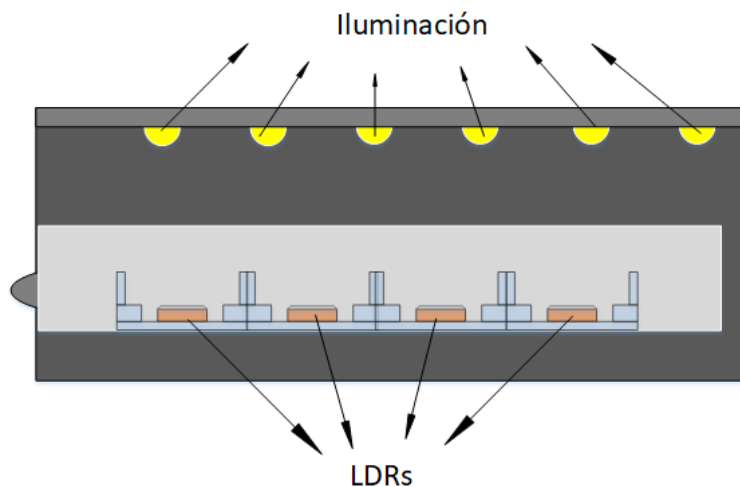


Fig. 4.11: Vista lateral de la gaveta, y dentro de ella los sensores; en la cara interna superior se ubica la iluminación.

De esta forma, todo el sistema de adquisición se mantiene a la espera de que la gaveta se abra, donde podría hacerse cualquier cambio en el stock. Luego, justo en el instante en que la gaveta se cierra, la iluminación debe encenderse, por tan solo un instante, para que los LDR puedan sensar, único momento en el cual el sistema de adquisición lee los sensores. El esquema de la Fig. 4.12 permite apreciar mejor este modo de funcionamiento. Luego de esto, el sistema vuelve a permanecer en *standby* y el ciclo se repite.

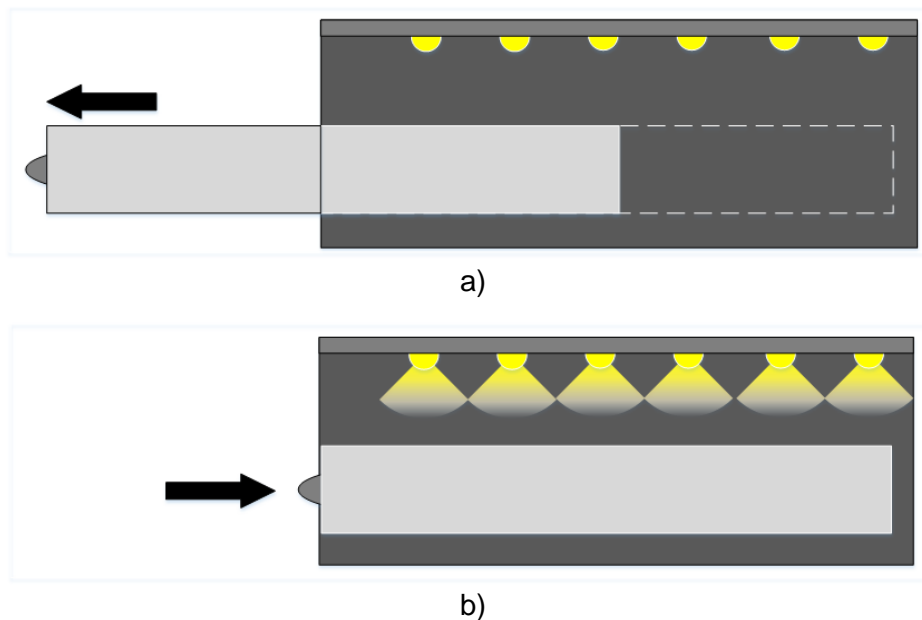


Fig. 4.12: Vista lateral de la gaveta: a) Gaveta al abrirse. b) Gaveta al cerrarse, momento en el cual la iluminación se enciende, permitiendo el sensado.

Para obtener una buena iluminación interna basta con utilizar ampolletas pequeñas o LEDs que permitan mantener un bajo consumo, el cual de por sí ya es bajo, considerando que no requiere mantener la iluminación encendida todo el tiempo, sino por tan solo un instante por cada evento de apertura, a modo de flash.

Además de la iluminación como elemento necesario para conseguir este funcionamiento, también es necesario algún sensor que permita detectar cuando la gaveta sea abierta y cerrada. Un final de carrera en el fondo de la gaveta es suficiente para la aplicación.

El siguiente diagrama de flujo, en la Fig. 4.13, describe el algoritmo de funcionamiento desde el punto de vista del sensor.

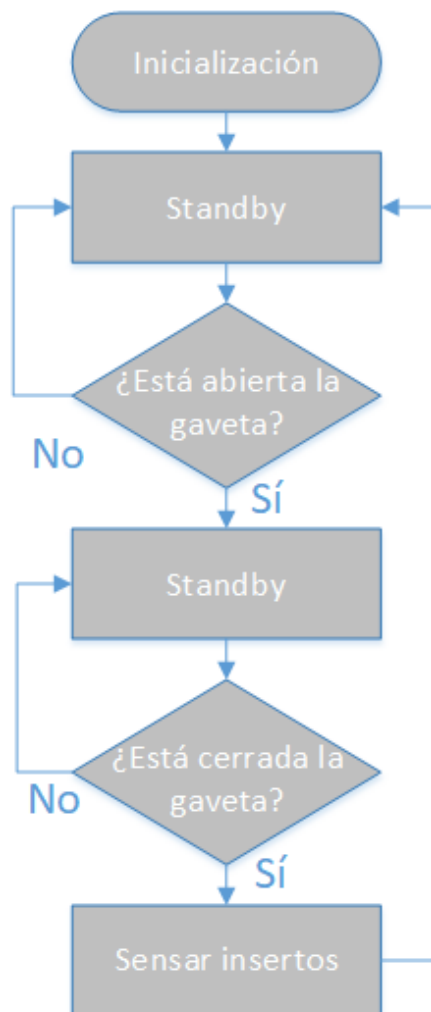


Fig. 4.13: Diagrama de flujo que resume el modo de funcionamiento desde el punto de vista del sensor.

## 4.05 Requerimientos

Como se expuso en el ítem anterior, el principio de funcionamiento expuesto requiere del uso de elementos adicionales para operar correctamente: iluminación controlada y final de carrera.

### 4.05.1 Iluminación controlada

La iluminación en el interior de la gaveta debe ser, al igual que el sensor, de tamaño reducido y bajo consumo, por lo tanto de inmediato surge la idea de utilizar LEDs. Sin embargo, el uso de LEDs, por sí solos, supondría también la implementación de PCBs adicionales. Es por eso que se decide utilizar tiras LED (*LED strip*), las cuales presentan

la ventaja de ser de reducido tamaño y muy prácticas, ya que pueden cortarse para ajustarse al largo requerido.

Existen diversas variedades de tiras LED, para diversas tensiones, potencias y colores. Para esta aplicación se decide utilizar las LED strip 5050, como la que se observa en la Fig. 4.14.

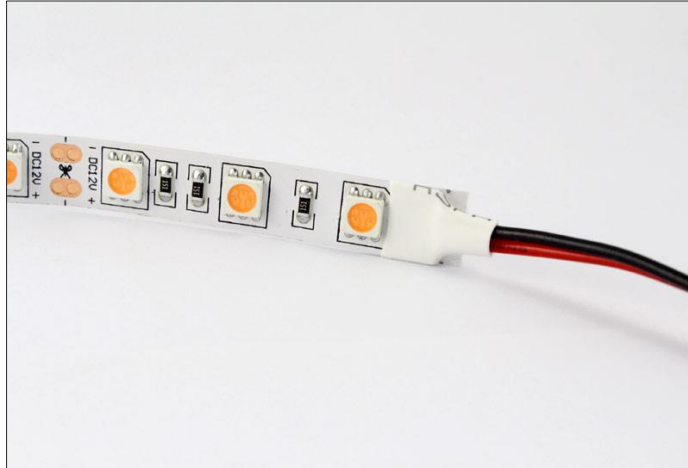


Fig. 4.14: Imagen referencial de una tira LED 5050.

Considerando que la cantidad de LEDs en una tira puede ser muy numerosa, y que se busca obtener la máxima intensidad que puedan ofrecer, es de esperar que la corriente que exijan sea relativamente alta. Se vuelve entonces necesario estimar la cantidad de corriente que las tiras demandarán, para poder continuar con el diseño del circuito posteriormente.

Cada tira LED debiera estar a lo largo de cada fila o columna, de tal forma que la luz que estas emitan incida perpendicularmente sobre los sensores. Si la matriz sobre la cual quedarán distribuidos los insertos es de 4 filas por 8 columnas, entonces se decide que las tiras irán ubicadas sobre las filas, por practicidad, ya que son menos que las columnas. Por lo tanto se utilizarán 4 tiras LED.

Del datasheet se extrae las tablas de las Fig. 4.15 y Fig. 4.16.

Table 1: Flux Characteristics

Code	Number of LEDs (M)	Case	Color	Wavelength	Lm (M)	Angle
5100	60	SMD 5050	Red	610-625 nm	180lm/m	120°
5101	60	SMD 5050	Yellow	590-595 nm	220lm/m	120°
5102	60	SMD 5050	Blue	460-470 nm	170lm/m	120°
5103	60	SMD 5050	Green	515-525 nm	550lm/m	120°
5104	60	SMD 5050	Cold White	6000-6500K	1080lm/m	120°
5105	60	SMD 5050	Warm White	3000-3500K	898lm/m	120°

Fig. 4.15: Características de flujo, tabla extraída del datasheet.

Table 2: Electrical Characteristics

Code	Color	Forward Current (mA/M)	Input Voltage (V)	Power (W/m)	Lm (M)
5100	Red	1200mA	12 VDC	14,4W	180lm/m
5101	Yellow	1200mA	12 VDC	14,4W	220lm/m
5102	Blue	1200mA	12 VDC	14,4W	170lm/m
5103	Green	1200mA	12 VDC	14,4W	550lm/m
5104	Cold White	1200mA	12 VDC	14,4W	1080lm/m
5105	Warm White	1200mA	12 VDC	14,4W	898lm/m

Fig. 4.16: Características eléctricas, tabla extraída del datasheet.

Cada metro de tira demanda 1200 [mA] y contiene 60 LEDs. Pero las tiras se pueden cortar solo cada 3 LED, es decir, hay 20 secciones cortables. Por lo tanto cada sección demanda 60 [mA].

Considerando que cada sección mide 50 [mm], como se muestra en la Fig. 4.17, y que se requiere a lo sumo (peor caso) 30 [cm] por fila, entonces la cantidad de secciones por fila es 6. Por lo tanto, la corriente total demandada por 4 tiras LEDs es:

$$I_{TIRAS\ LED} = 4 * 6 * 60 [mA] = 1440 [mA] \tag{4.1}$$

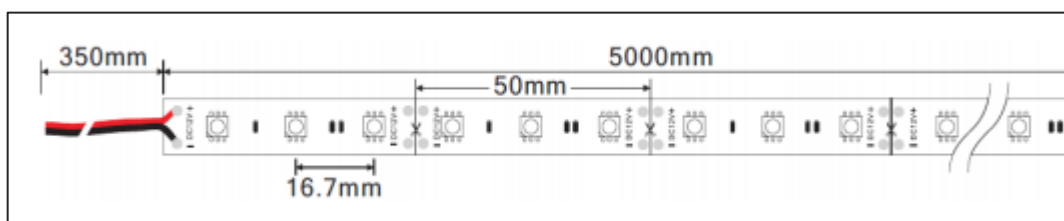


Fig. 4.17: Esquema físico frontal de la LED strip con sus respectivas dimensiones, extraído del datasheet.

Es importante considerar este valor de corriente, ya que de él dependerá cuál fuente de alimentación usar luego. Por otro lado, si bien este valor pudiera parecer importante también en el consumo del sistema, en realidad no es sustancial, ya que la iluminación se requiere durante tan solo fracciones de segundo por cada evento de sensado.

#### 4.05.2 Limit Switch

Un interruptor final de carrera es necesario para esta aplicación ya que los eventos de sensado responden a un movimiento mecánico: la detección comienza cuando la gaveta se cierra.

Existe una enorme variedad de limit switches, para diversos parámetros mecánicos y eléctricos. No obstante, dado que las condiciones mecánicas de la gaveta no son particularmente importantes, basta con utilizar un final de carrera por colisión que funcione con 5V TTL, de tal forma que pueda conectarse directamente al microcontrolador.

Se decide utilizar el módulo switch de colisión YL-99, que se puede apreciar en la Fig. 4.18, el cual es un limit switch normalmente cerrado.



Fig. 4.18: Módulo Collision Switch YL-99.

Trabaja de 3-12 [Vdc] y puede conectarse directamente al AVR.

## CAPÍTULO 5: ACONDICIONAMIENTO DE SEÑAL

El parámetro eléctrico recogido por el sensor, típicamente, no se adecúa de forma inmediata a las condiciones de los sistemas digitales, y esta no es la excepción. Es por eso que el parámetro debe ser acondicionado electrónicamente para poder ser leído por las etapas subsiguientes.

La etapa de acondicionamiento de señal consiste en el procesamiento de la señal obtenida por el sensor hasta convertirla en otra señal con parámetros adecuados, en este caso adecuados a los sistemas digitales, de tal forma que pueda ser leído por el microcontrolador.

En el presente capítulo se diseña el acondicionamiento electrónico para la señal recogida por la LDR. Se subdivide en las siguientes partes:

1. Divisor de tensión: Se definen las condiciones necesarias para obtener una salida acondicionada a un nivel TTL mediante un divisor de tensión.
2. Multiplexación: Se diseña una etapa de multiplexado de 32 a 1 mediante multiplexores de 8 a 1.

### 5.01 Divisor de tensión

El parámetro eléctrico obtenido por cada una de las LDR es una variación de resistencia que va desde los cientos de ohm hasta los mega-ohm. Esto representa una variación de resistencia considerable, por no decir grande, situación que permite prescindir de acondicionamientos demasiado sofisticados, como por ejemplo un puente *Wheatstone*. En su lugar, basta con aplicar un divisor de tensión, como el que se indica en la Fig. 5.1.



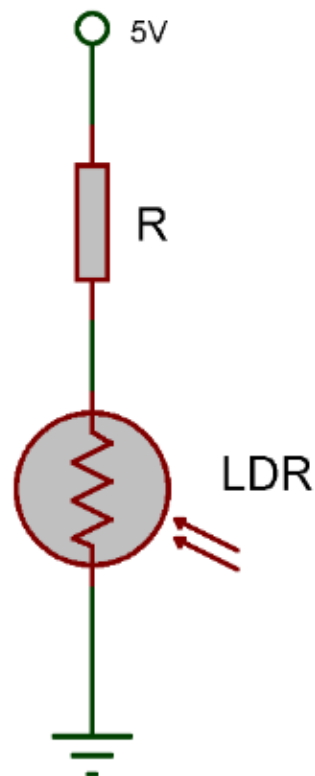


Fig. 5.1: Divisor de tensión compuesto por una resistencia y una fotorresistencia.

Conectar una resistencia en serie a la LDR permite no solo limitar la corriente, sino también conseguir el divisor de tensión, es decir, permitir que la tensión total aplicada en la malla se reparta entre ambos componentes de forma proporcional a sus valores resistivos. De esta forma, se consigue una cierta variación de tensión en la LDR. Ahora bien, considerando que la LDR solo trabaja en sus estados de mínima y máxima iluminación, esto es, en su mínimo y máximo valores de resistencia, resulta importante considerar la situación particular de ambos estados. Después de todo, la señal obtenida depende de la proporción resultante entre los valores de ambos componentes.

Bajo esta premisa, se busca un valor para la resistencia  $R$  que en conjunto al funcionamiento de la LDR permita obtener a su salida una señal de tensión discreta, acorde a los límites de una señal TTL, es decir, con dos niveles, uno alto y uno bajo, para los estados de iluminación y oscuridad, respectivamente. Para el estado bajo se requiere que la tensión sea de entre 0 y 0.8 [V], mientras que para el estado alto se requiere que la tensión se encuentre entre 2 y 5 [V].

Con tal de obtener las condiciones mencionadas, el valor de  $R$  debe ser tal que, para un estado resistivo de la foto-resistencia,  $R$  sea considerablemente mayor, mientras que para el otro estado, sea considerablemente menor. Dicho de otro modo, sea  $R_{LDR\ MIN}$  la resistencia de la LDR en plena luz, y  $R_{LDR\ MAX}$  su resistencia en plena oscuridad, se han de cumplir los siguientes criterios preliminares:

$$R \gg R_{LDR\ MIN} \quad (5.1)$$

$$R \ll R_{LDR\ MAX} \quad (5.2)$$

Estos criterios aclaran la idea de que el valor de la resistencia R ha de estar en un nivel intermedio. Si  $R_{LDR\ MIN}$  fuese al menos 9 veces inferior a R, de tal forma que la tensión sobre la LDR fuese un décimo de los 5 [V] aplicados, entonces la señal obtenida a la salida de la LDR sería a lo sumo de 0.5 [V]; así mismo, de manera análoga, si la  $R_{LDR\ MAX}$  fuese al menos 9 veces superior, de tal forma que la tensión sobre la LDR fuese nueve décimas partes de los 5 [V] aplicados, entonces la señal obtenida a la salida de la LDR sería de al menos 4.5 [V]. Estos dos niveles de tensión se encuentran dentro de los límites válidos para los niveles lógicos de una TTL. Dicho de otro modo, los criterios (X) y (X) se convierten, respectivamente, en las siguientes condiciones de diseño:

$$R \geq 9 R_{LDR\ MIN} \quad (5.3)$$

$$R \leq 9 R_{LDR\ MAX} \quad (5.4)$$

Entonces se procede a buscar un valor de R que cumpla las condiciones (X) y (X), en base los valores resistivos reales, es decir,  $R_{LDR\ MIN}$  y  $R_{LDR\ MAX}$ , que pueda proporcionar una LDR comercial, bajo condiciones controladas de iluminación. Para ello, primero se establecen los valores de  $R_{LDR\ MIN}$  y  $R_{LDR\ MAX}$  de una LDR real, tal que permitan aplicar dichas condiciones.

Considérese entonces una LDR modelo GL5539, de cuyo *datasheet* (Series GL55XX) se extrae la siguiente gráfica, mostrada en la Fig. 5.2.

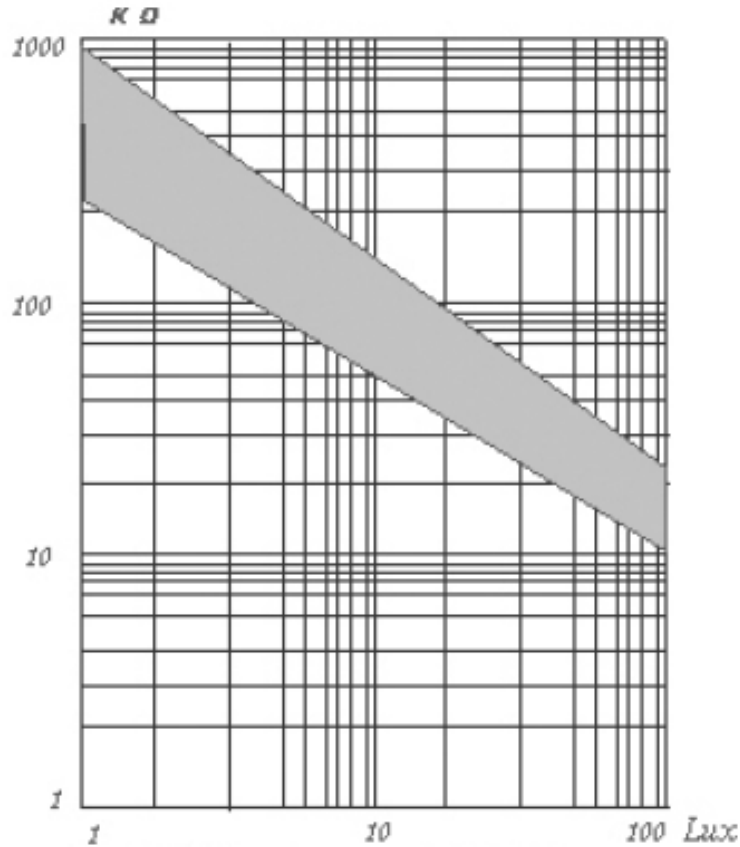


Fig. 5.2: Curva característica Iluminancia-Resistencia.

De la gráfica se observa que la resistencia en la LDR corresponde a 1 [MΩ] para una iluminancia de 1 [Lux] (casi absoluta oscuridad), y de 10 [kΩ] para 100 [Lux] (iluminación ambiente en una sala de estar).

Considerando estos dos niveles de resistencia, se busca un valor de resistencia R, a conveniencia, que cumpla las condiciones (X) y (X), y que al mismo tiempo se ajuste a los límites de la curva característica de la LDR. Así, un valor de resistencia comercial que se ajusta adecuadamente, por ejemplo, resulta ser el de 100 [kΩ], ya que es 10 veces mayor que la  $R_{LDR\ MIN}$  y al mismo tiempo 10 veces menor que la  $R_{LDR\ MAX}$ .

Por lo tanto, aplicando el divisor de tensión para ambos estados, se obtienen los siguientes niveles de tensión:

$$V_{LDR\ LOW} = \frac{V_{CC} * R_{LDR\ MIN}}{R + R_{LDR\ MIN}} = \frac{5V * 10k\Omega}{100k\Omega + 10k\Omega} \approx 0.46\ V \quad (5.5)$$

$$V_{LDR\ HIGH} = \frac{V_{CC} * R_{LDR\ MAX}}{R + R_{LDR\ MAX}} = \frac{5V * 1M\Omega}{100k\Omega + 1M\Omega} \approx 4.55\ V \quad (5.6)$$

En resumen, al usar una resistencia de 100 [kΩ], se obtienen los dos siguientes estados en la LDR, que a su vez sirven como lectura de la presencia y ausencia de los insertos:

- Plena oscuridad: Iluminancia de 1 [Lux] y resistencia de 1 [MΩ]. Caída de tensión obtenida de 4.55 [V], equivalente al estado alto de una TTL. Fig. 5.3 a.
- Plena luz: Iluminancia de 100 [Lux] y resistencia de 10 [kΩ]. Caída de tensión obtenida de 0.46 [V], equivalente al estado bajo de una TTL. Fig. 5.3 b.

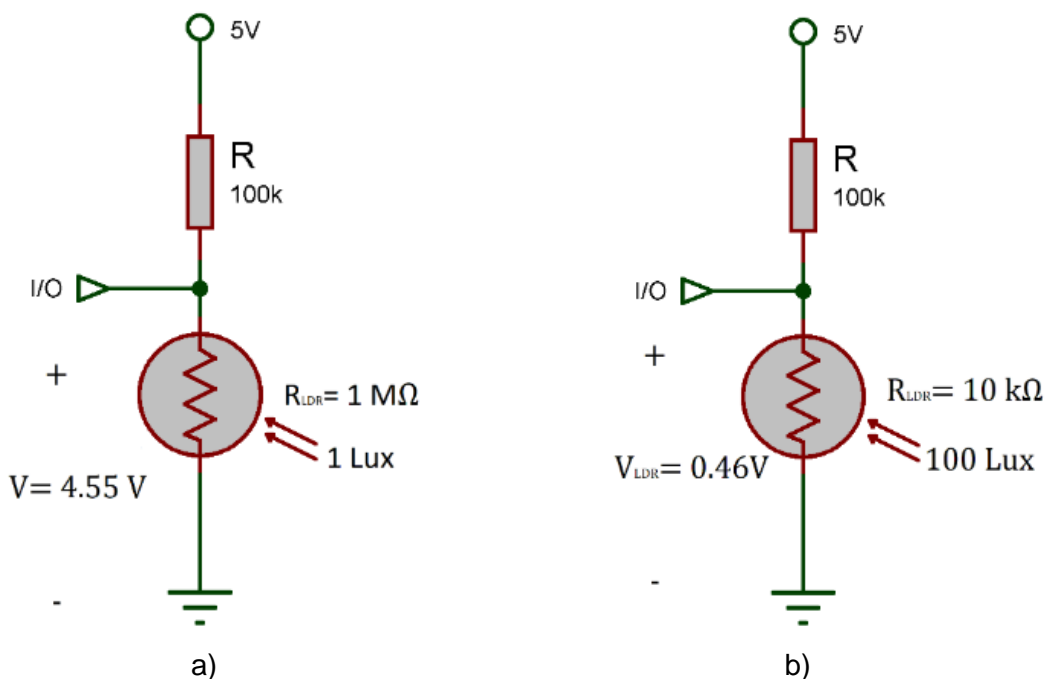


Fig. 5.3: Divisor de tensión en: a) oscuridad y b) iluminación.

Un posible inconveniente de esta técnica es que, al depender en gran medida de la iluminación para obtener el estado lógico, y al encontrarse el estado bajo un tanto cerca del umbral TTL (de 0.46V a 0.8V hay una diferencia de solo 0.34V), cabe la posibilidad de que leves errores de iluminación y/o ruido eleven la señal hasta entrar al rango de incertidumbre de una TTL, comprendido entre los 0.8 y 2 [V], errando así la lectura.

No obstante, en la práctica, el nivel de iluminancia incidente sobre la LDR puede ser, de hecho, mayor a 100 [Lux], aplicando una fuente con la suficiente intensidad lumínica para obtener en la LDR un menor valor de resistencia, situación que, si bien no está especificada en la curva del datasheet, puede comprobarse experimentalmente. Así mismo, la suficiente proximidad de la fuente lumínica, presumiblemente menor a los 5 [cm] debido a las limitaciones en las dimensiones de una gaveta, contribuye a obtener una iluminancia superior a los 100 [Lux], provocando así que el valor de la resistencia  $R_{LDR\ MIN}$  sea, finalmente, menor a los 10 [kΩ], reduciendo así la tensión de la señal en estado bajo a un nivel aún menor que los 0.46 [V] previstos por el diseño.

## 5.02 Multiplexación

Tal como se había adelantado anteriormente en el Capítulo 2 acerca del esquema general, la gran cantidad de objetos a sensor exige un circuito de multiplexación en la etapa de acondicionamiento, que permita al sistema prescindir de una cantidad demasiado grande de GPIOs en el microcontrolador. Para efectos de este diseño se considera que se requiere sensor 32 ítems, y esta sub-etapa debe ser capaz de manejar dicha cantidad.

Ahora bien, el objetivo planteado de multiplexar 32 ítems presupone el siguiente problema. Una búsqueda en el mercado actual permite comprobar que los integrados multiplexores de 16 o más entradas son poco comunes y se comercializan solo en formato SOIC (montaje superficial), lo cual dificulta las pruebas al no poder montarse sobre la *protoboard*, sin mencionar que no se comercializan con tanta facilidad. Por lo tanto se decide utilizar multiplexores de 8 a 1, los cuales sí se comercializan en formato DIP y son de fácil adquisición. La cantidad de entradas necesaria se puede aumentar a 32 combinando varios de ellos correctamente.

Para conseguir 32 entradas, usando multiplexores de 8 a 1, se requiere usar cinco de estos integrados: 4 para conseguir la cantidad total de entradas y 1 para unir a los cuatro anteriores en una única salida. Para este diseño se decide utilizar entonces el integrado 74HC151N, el cual es un multiplexor con tecnología TTL de 8 bits y encapsulado DIP16, que cuenta con 3 entradas de control y una salida. Su esquemático puede apreciarse en la Fig. 5.4.

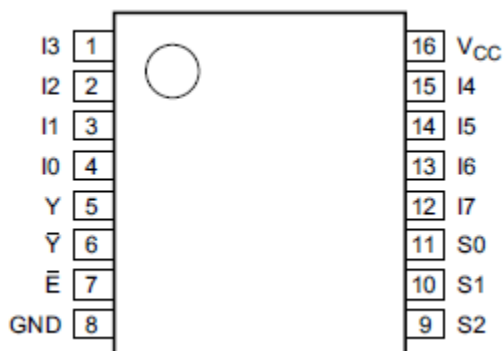


Fig. 5.4: Esquema del IC 74HC151N, extraído del Datasheet.

El conexionado de los multiplexores queda entonces de la siguiente forma: la salida de cada LDR, cuya señal quedó acondicionada por el divisor de tensión en la sub-etapa anterior, se conecta a cada una de las entradas de los cuatro multiplexores, desde la I0 hasta las I7 correlativamente, obteniendo 32 canales. Las 4 salidas obtenidas a partir de estos se dirigen luego al quinto multiplexor, desde I0 hasta I3, respectivamente. Las cuatro entradas restantes quedan puestas a GND ya que no se utilizan. El esquema resultante puede apreciarse en Fig. 5.5.

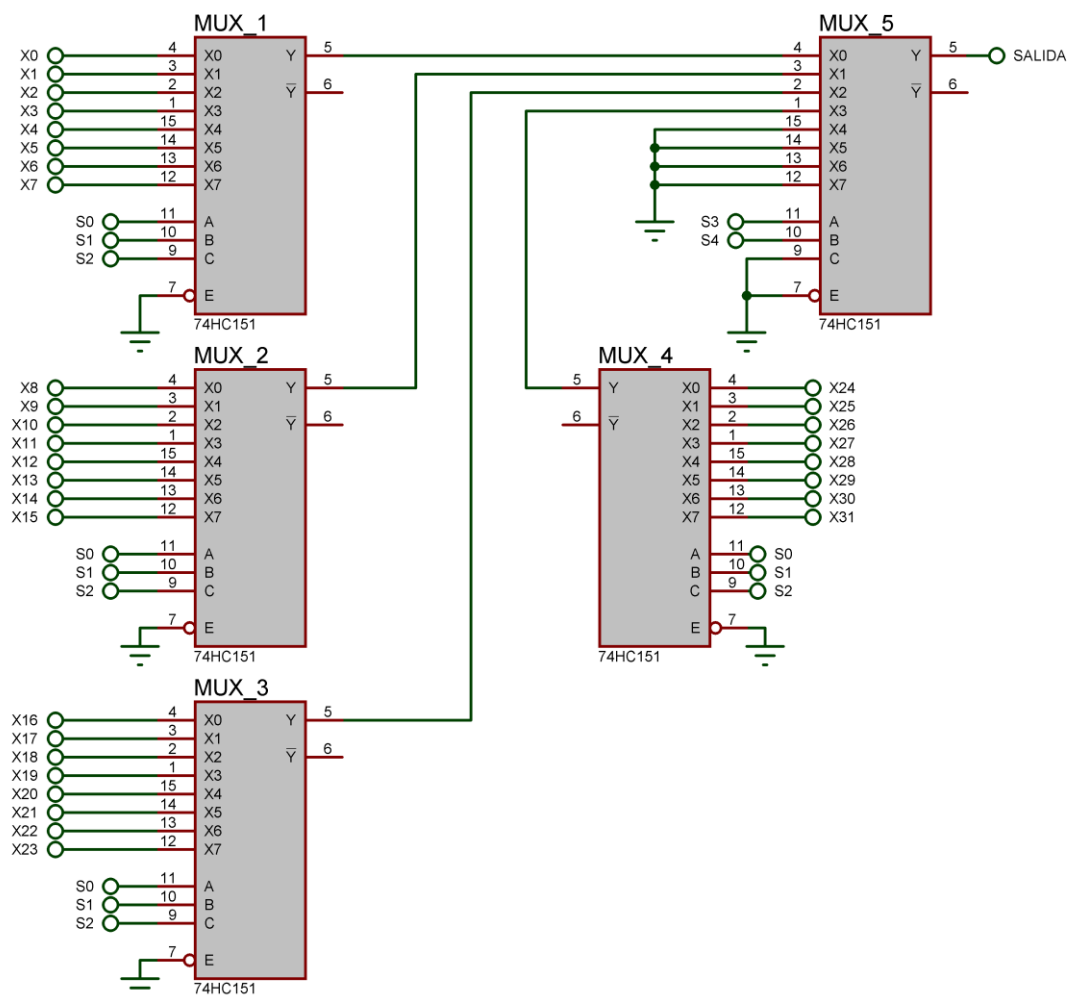


Fig. 5.5: Circuito completo de sub-etapa de multiplexación.

El pin 5 del MUX 5 es la salida, y va conectada a la siguiente etapa que corresponde al microcontrolador. Los pines 7 de *Enable* quedan conectados a GND para mantener a los chips activos en todo momento, mientras que la salida negada del pin 6 queda al aire ya que está fuera de uso. Además, los pines 12, 13, 14 y 15 del MUX 5 tampoco se utilizan, si bien podrían usarse para escalar a 64, sin mayores complicaciones, la cantidad de entradas.

Cabe destacar el uso de los pines de control 9, 10 y 11 de cada integrado, también llamados A, B y C. Estos pines de control permiten seleccionar el canal que entregará la salida de cada MUX. Para esta aplicación, como la cantidad de canales a leer es de 32, es necesario 5 bits, es decir, 5 pines de control para manejar toda esta sub-etapa. Estos pines de control quedan conectados a S0, S1, S2, S3 y S4, que le corresponden al microcontrolador, siendo S4 el bit más significativo. Los pines S0, S1, y S2 les corresponden a los primeros cuatro MUX, mientras que los S3 y S4 le corresponden al MUX 5, quedando el pin 9 (C) de este, que debiera pertenecer a un posible S5, conectado a GND, ya que está fuera de uso también.

A continuación, en la Tabla XX, se presenta la tabla de verdad que resume el comportamiento de toda la etapa, donde L significa estado bajo (Low) y H significa estado alto (High). Los pines de control o selección quedan definidos por S4, S3, S2, S1 y S0, donde S4 es el bit más significativo. Estos pines controlan el canal que se vea reflejado a la salida, donde los canales quedan enumerados desde X0 hasta X31, cada uno proveniente de su respectivo sensor, cuya señal puede ser alta o baja. Cabe señalar que este comportamiento es válido siempre y cuando se respete el circuito de la Tabla 4, donde la señal a *Enable* es siempre baja.

Tabla 4: Tabla de verdad, comportamiento de la sub-etapa de multiplexación.

S4	S3	S2	S1	S0	Y
L	L	L	L	L	X0
L	L	L	L	H	X1
L	L	L	H	L	X2
L	L	L	H	H	X3
L	L	H	L	L	X4
L	L	H	L	H	X5
L	L	H	H	L	X6
L	L	H	H	H	X7
L	H	L	L	L	X8
L	H	L	L	H	X9
L	H	L	H	L	X10
L	H	L	H	H	X11
L	H	H	L	L	X12
L	H	H	L	H	X13
L	H	H	H	L	X14
L	H	H	H	H	X15
H	L	L	L	L	X16
H	L	L	L	H	X17
H	L	L	H	L	X18
H	L	L	H	H	X19
H	L	H	L	L	X20
H	L	H	L	H	X21
H	L	H	H	L	X22
H	L	H	H	H	X23
H	H	L	L	L	X24
H	H	L	L	H	X25
H	H	L	H	L	X26
H	H	L	H	H	X27
H	H	H	L	L	X28
H	H	H	L	H	X29
H	H	H	H	L	X30
H	H	H	H	H	X31

Obteniéndose el comportamiento mostrado en la anterior tabla, se disminuye drásticamente la cantidad de GPIOs requeridos en el microcontrolador.

### 5.03 Resumen Capítulo 5

El divisor de tensión obtenido permite acondicionar la señal del sensor a los niveles de una TTL sin la necesidad de un sofisticado circuito de acondicionamiento. Sin embargo, se debe tener en cuenta que las condiciones de uso, en principalmente las de iluminación, sean las indicadas.

Luego, la etapa de multiplexación permite recoger la señal obtenida por cada uno de los sensores y transmitirla a un solo canal, reduciendo así sustancialmente la cantidad de GPIOs requeridas por el microcontrolador.

A partir de aquí, teniendo ya las señales de los sensores adecuadamente acondicionadas, se procede a diseñar conectar el circuito del microcontrolador y a procesar la señal mediante un algoritmo.



## CAPÍTULO 6: MICROCONTROLADOR Y PERIFÉRICOS

La señal acondicionada debe ser procesada, y es el microcontrolador el que se encarga de hacerlo, si bien no es su única labor. El microcontrolador, es decir, el ATmega328P, se encarga también de controlar los periféricos que permiten el correcto desempeño del principio de funcionamiento diseñado, así como de enviar los datos recogidos hacia los indicadores, que permitirán disponer de la información recogida.

El presente capítulo presenta los circuitos que se conectan al microcontrolador, así como también se expone el algoritmo de funcionamiento bajo el cual trabajará. Se subdivide en las siguientes partes:

1. Configuración del ATmega328P: Se presenta la configuración electrónica para la puesta en marcha del ATmega328P.
2. Algoritmo de sensado: Se diseña el algoritmo con el cual trabajará el microcontrolador para sensar y gestionar periféricos.
3. Conexión de periféricos: Se especifica el conexionado de los elementos periféricos al microcontrolador, etapa por etapa.
4. Esquema resultante: Se presenta el circuito total del microcontrolador con sus periféricos, hasta este punto.

Cabe señalar que si bien el módulo NodeMCU también podría considerarse un periférico, sus características propias de un microcontrolador vuelven un poco más complejo su análisis, y por tanto se estudian en el capítulo siguiente.

## 6.01 Configuración del ATmega328P

Como se había adelantado anteriormente, el microcontrolador a utilizarse es el ATmega328P-PU, un microcontrolador CMOS de 8 bits y 28 pines basado en la arquitectura RISC AVR de la marca ATmel). Entre sus principales características de interés para la aplicación, se encuentran:

- Voltaje de operación de 1.8 – 5.5V.
- Hasta 20 MHz de velocidad, para alimentación de entre 4.5 – 5.5V.
- 32 registros de propósito general.
- 32 KB de memoria flash interna programable.
- 1KB de memoria EEPROM.
- 2KB de memoria SRAM.
- 2 timers/contadores de 8 bits.
- 1 timer/contador de 16 bits.
- 23 entradas/salidas de propósito general.
- 6 modos de sleep (inactividad): Idle, ADC Noise Reduction, Power-save, Power-down, Standby, y Extended-standby.
- Consumo a 25 °C y en modo Power-down de 0.1  $\mu$ A.

Este microcontrolador, cuyas características se asemejan en gran medida a las del PIC18F2550, si bien no ofrece las mejores prestaciones posibles en el mercado, cumple con lo suficiente para cubrir los requisitos del proyecto, a un precio razonable. De entre sus características es importante resaltar que cuenta con 23 GPIOs que, gracias al uso de multiplexación en otra etapa del sistema, permiten controlar satisfactoriamente todos los periféricos.

El ATmega328P-PU se comercializa en encapsulados DIP, MLF y TQFP. Para el presente diseño, y principalmente por motivos prácticos a la hora de implementar posteriormente el circuito en protoboard, se considera en su formato DIP-28.

El esquemático de terminales, extraído del *datasheet*, es el que se muestra a continuación, en la Fig. 6.1.

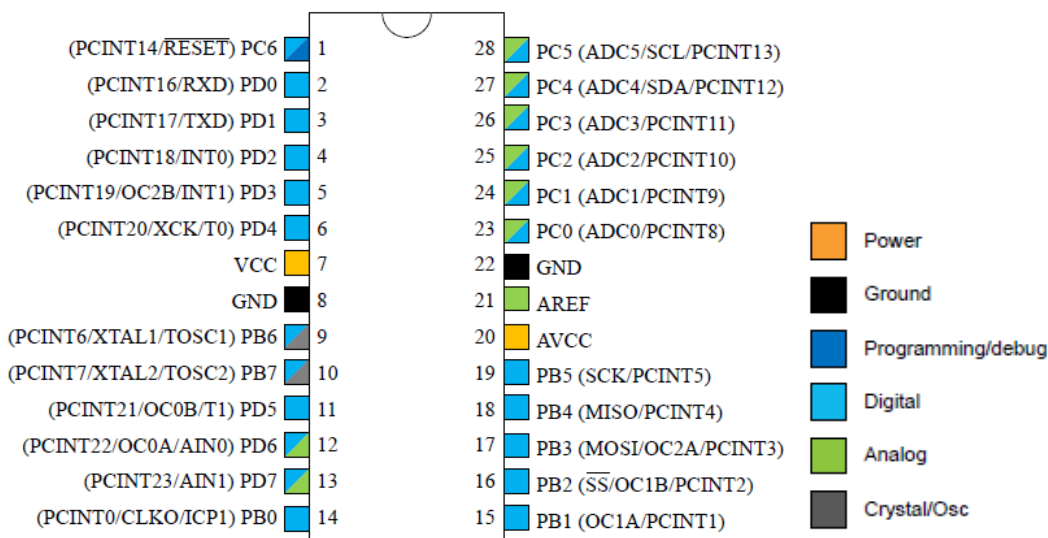


Fig. 6.1: Esquema de pines del Atmega328P-UP.

Para conseguir una velocidad de reloj mayor a 10 MHz, según *datasheet*, es necesario alimentar al AVR con una tensión de entre 4.5 y 5.5VDC, además de seleccionar un oscilador que provea una determinada frecuencia de trabajo. Para este diseño, la alimentación se fija en 5V, la cual se conecta a VCC (pin 7), y también a AVCC (Alimentación para el conversor A/D, pin 20) y a AREF (Referencia análoga para el conversor A/D, pin 21), los cuales se recomienda mantener conectados a Vcc incluso si el conversor A/D no se encuentra en uso.

Para la velocidad de reloj, se decide trabajar a una frecuencia de 16 MHz, valor estandarizado y de amplio uso en las actuales librerías de programación de Arduino. Cabe mencionar que, si bien el microcontrolador puede funcionar a esta y a otras velocidades gracias a su oscilador interno, lo cual habría de ser configurado dentro del código, la precisión de este reloj presenta errores que pueden llegar hasta el 10%, razón por la cual se decide utilizar un cristal de cuarzo de 16 MHz que funcione como oscilador externo. El cristal se conecta entre los pines 9 y 10 del AVR (TOSC1 y TOSC2, respectivamente), y se conecta a GND por medio de dos condensadores de igual capacitancia, que permitan estabilizar el pulso de reloj. Se decide utilizar capacitores de cerámico de 22 pF cada uno (valor que según *datasheet* debiera estar comprendido entre 12 y 22 pF).

Por último se conecta RESET (pin 1) a VCC por medio de una resistencia *pull-up* de 10K. Esto mantendrá el pin en un estado alto, evitando así que el AVR se reinicie sin previo aviso.

Toda la configuración anteriormente descrita da como resultado el circuito expuesto en la Fig. 6.2, el cual, en su conjunto, representa la puesta a punto del microcontrolador. Es decir, a partir de aquí es posible ejercer control sobre los demás periféricos a conectar.

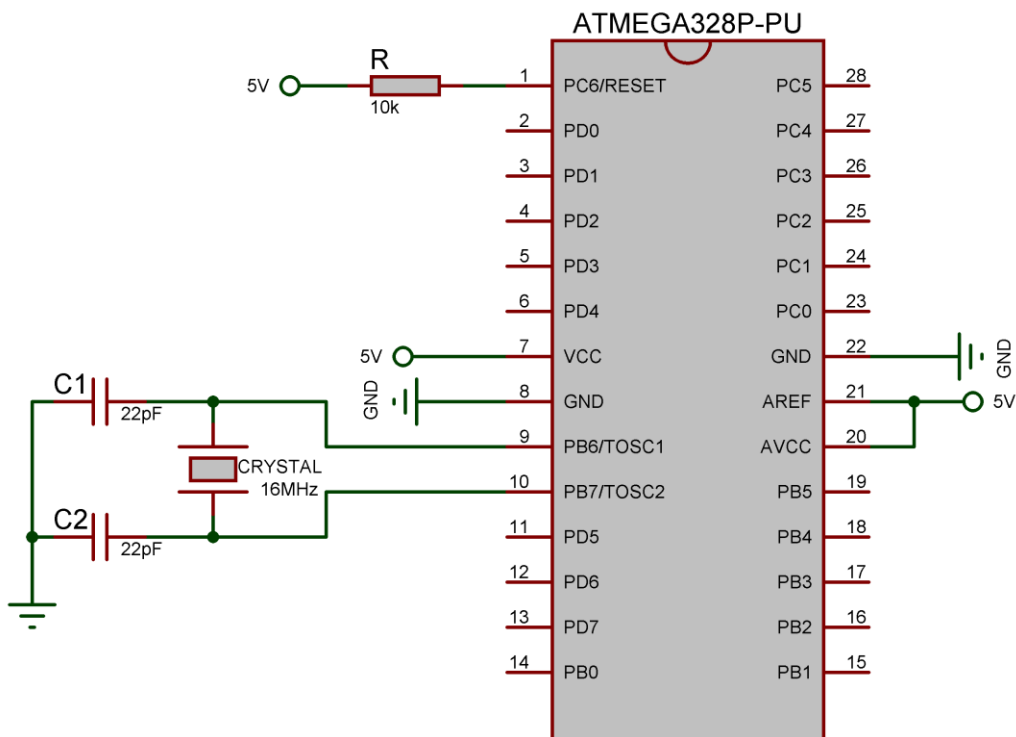


Fig. 6.2: Configuración del Atmega328P-UP.

Cabe recalcar que los GPIOs disponibles para su uso en el algoritmo son 20, y corresponden a los pines: 2, 3, 4, 5, 6, 11, 12, 13, 14, 15, 16, 17, 18, 19, 23, 24, 25, 26, 27 y 28.

## 6.02 Algoritmo de sensado

El algoritmo anteriormente presentado en el diagrama de flujo de la Fig. 4.13 correspondía al funcionamiento desde el punto de vista del sensor, pero también está asociado al algoritmo de funcionamiento del microcontrolador, si bien este va más allá. Este algoritmo debe gestionar a todos los periféricos de manera secuencial, y debe verse adecuadamente reflejado en el código que se ha de cargar al integrado. A continuación se describen las principales consideraciones y operaciones del algoritmo, para finalizar con un diagrama de flujo que lo represente, así como con el código real que maneja al microcontrolador.

Primeramente, el micro inicializa todas las variables que utilizará para procesar la información. En este sentido, lo principal a considerarse es el conexionado de sus GPIO. Para este diseño, los pines se configuran de la siguiente manera, como se indica en la Tabla 5.

Tabla 5: Configuración de GPIOs en el ATmega328P para el presente algoritmo.

Pin físico	Pin IDE	Función periférico
2	D0	Rx Puerto Serie
3	D1	Tx Puerto Serie
4	D2	Rx Comunicación NodeMCU
5	D3	Tx Comunicación NodeMCU
6	D4	MUX Lectura
11	D5	<i>Limit switch</i> Lectura
12	D6	MUX <i>Enable</i> Control (Fuera de uso)
13	D7	Tira LED Control
14	D8	LCD R/W
15	D9	LCD Enable
16	D10	LCD DB4
17	D11	LCD DB5
18	D12	LCD DB6
19	D13	LCD DB7
23	A0	MUX S0
24	A1	MUX S1
25	A2	MUX S2
26	A3	MUX S3
27	A4	MUX S4
28	A5	MUX S5 (Fuera de uso)

Luego de la inicialización de pines, librerías y variables, se procede con el funcionamiento cíclico del algoritmo, el cual se ejecutará indefinidamente.

Considerando que son los eventos de apertura y cierre los que desencadenan la detección de insertos, el código requiere chequear con frecuencia el estado del *limit switch*, es decir, del pin 11. Por lo tanto se parte leyendo el *limit switch*.

Si la gaveta está abierta (pin 11 en estado bajo) entonces el sistema se mantiene en *standby* a la espera de que la gaveta se cierre. Si la gaveta se cierra (pin 11 en estado alto) entonces comienza la secuencia de detección.

Se enciende la tira LED (pin 13 estado alto) y se imprime “Calculando...” en la pantalla LCD. Entonces comienza un retardo de 300 milisegundos antes de continuar, para darle tiempo a las LDR de que alteren sus valores resistivos, ya que su tiempo de respuesta no es muy rápido, y las instrucciones anteriormente realizadas tardan menos en ejecutarse que la LDR en responder al cambio de iluminación.

Se corrobora que la puerta permanezca cerrada, o de otro modo se cancela la detección. Esta comprobación se realiza en varios puntos de la secuencia, para evitar errores de sincronía.

Entonces se comienza una secuencia de lectura a cada uno de los canales del multiplexor, es decir, una lectura a cada una de las LDR, a través del pin 6 (estado bajo es ausencia, estado alto es presencia). Para ello se envía a los pines de selección del MUX la secuencia

de bits 00000, a través de los pines 23 al 27 (correspondiente a S0, S1, S2, S3 y S4), que permiten seleccionar la primera LDR. Entonces se lee el pin 6, y si el estado es alto, se suma un 1 a un contador. Además, se guarda el estado, ya sea bajo o alto, en un *array* de 32 posiciones que guardará, eventualmente, todos los estados.

En este punto se hace una nueva comprobación de la gaveta, y en caso de abrirse, se detiene la detección, se resetea el contador y se imprime “INTERRUMPIDO” en la LCD. En cambio, si la gaveta permanece cerrada, se continúa con la detección.

Entonces se envía una nueva secuencia de bits a los pines de selección del MUX, esta vez 00001, es decir, se ha sumado un 1 binario a la secuencia de bits. Esto selecciona el segundo canal para leer el estado de una nueva LDR. El proceso se repite para la secuencia 00010, luego para 00011, y así sucesivamente, hasta alcanzar la secuencia de control 11111, correspondiente al canal número 32.

Si la secuencia de lectura no se detuvo en ningún momento, la detección termina. La iluminación se apaga, los bits de control se resetean, y se imprime “Numero de Insertos”, seguido del número que el contador acumuló durante la secuencia.

En este punto, teniendo todos los datos recogidos, se entabla comunicación con el NodeMCU, a través de los pines 4 y 5 (Rx y Tx). Para ello se crea una cadena de texto que contiene todos los 0 y 1 del *array* de posiciones, a la cual también se agrega el número total de insertos que se obtuvo en el contador. La secuencia de 1 y 0 se encuentra separada por un guion (“-”) con el objetivo ordenar los datos de la cadena de texto que se transmitirá. Una vez creada la cadena, esta se transmite al NodeMCU mediante el puerto serie, y luego se resetea la cadena y el contador. Esta cadena queda finalmente de la forma que se muestra en la Fig. 6.3.

00000000100001001101000011000000-7

Fig. 6.3: Cadena de caracteres de ejemplo, que se envía al NodeMCU por medio de comunicación serie.

Por último, terminada la detección y transmisión, el sistema entra en *standby*, a la espera de que la gaveta se abra. Si la gaveta se abre, se imprime “Puerta Abierta” en la LCD. Entonces se vuelve a entrar en *standby*, hasta que la puerta vuelva a estar cerrada, para repetir todo el ciclo.

A continuación se, en la Fig. 6.4, se presenta un diagrama de flujo que refleja todo el funcionamiento recién explicado.

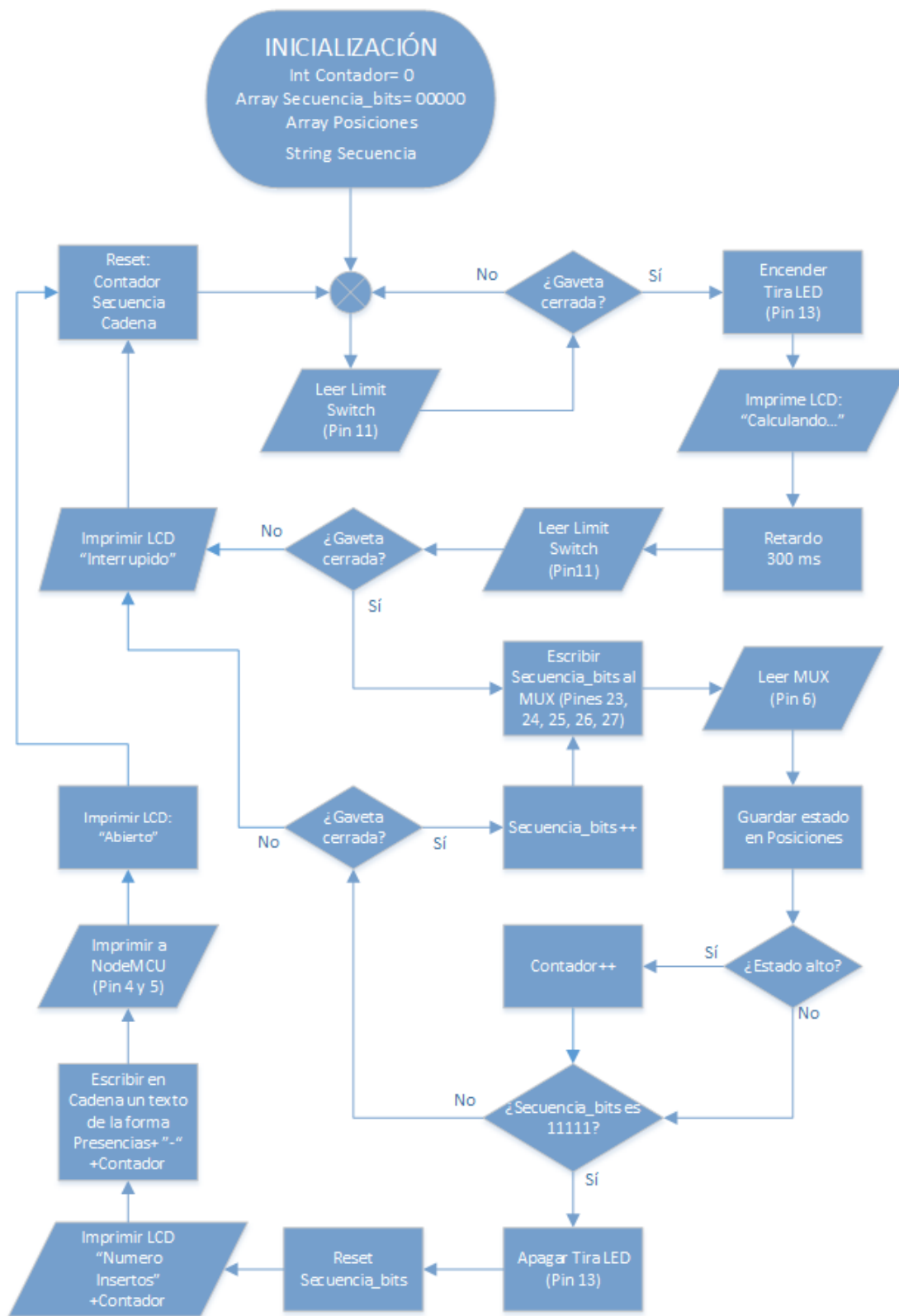


Fig. 6.4: Diagrama de flujo del algoritmo de funcionamiento para el ATmega.

El código que refleja el algoritmo anteriormente descrito queda señalado en el Anexo A. Este se programa en C mediante el IDE de Arduino, y se carga al microcontrolador por medio de la placa de desarrollo, para posteriormente ser desmontado y utilizado en modo *standalone*, con la configuración anteriormente señalada en la Fig. 6.2.

Una vez descrita toda la secuencia del algoritmo, se procede a señalar el conexionado de todos los periféricos comandados por el microcontrolador, en base a la configuración de pines señalada en la Tabla 5.

### 6.03 Conexionado de periféricos

A continuación se especifican los diagramas de conexionado para cada uno de los periféricos anteriormente estudiados.

#### 6.03.1 Etapa de Multiplexación

Los multiplexores utilizan 6 GPIOs del ATmega328P: cinco de estos pines corresponden a los de selección de canal, y el restante corresponde a la salida del multiplexor, donde se lee el estado del canal seleccionado. El esquema de conexión queda expuesto en la Fig. 6.5.

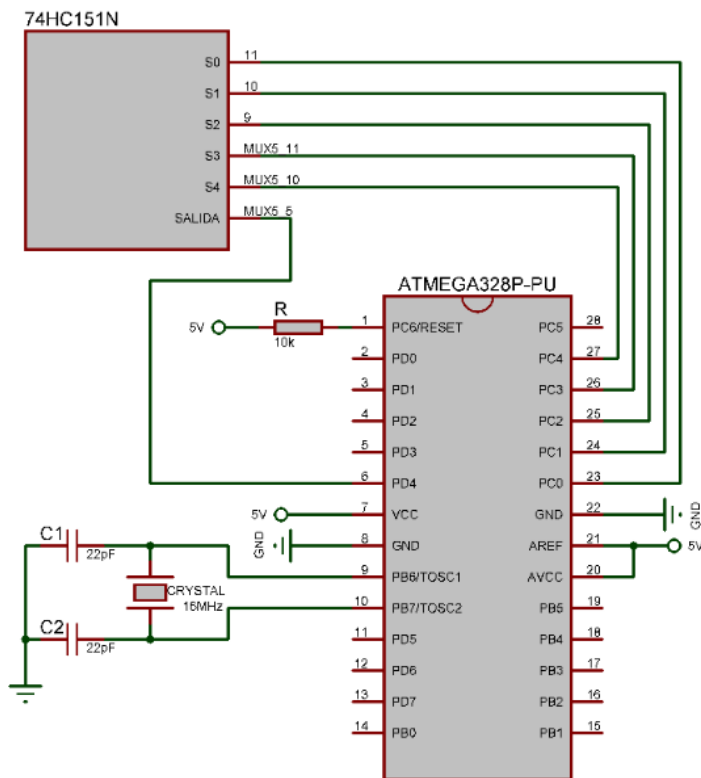


Fig. 6.5: Esquema de conexionado de la etapa MUX al ATmega.



### 6.03.2 Iluminación

En el capítulo anterior se calculó la cantidad esperada de corriente que demandarán las tiras LED, y dio como resultado casi 1.5 [A], un valor de intensidad demasiado grande como para ser manejado por el microcontrolador, que como cabe esperar de un circuito digital, soporta tan solo 20 [mA] por GPIO. Por esta razón, se requiere de un circuito de control de corriente que permita acoplar la iluminación.

Un transistor de alta ganancia Darlington puede utilizarse como switch para controlar un flujo de corriente relativamente alta mediante una señal de baja intensidad en su base. Su alta ganancia hace que el transistor conmute entre corte y saturación, y de esta forma, puede actuar como un switch para encender y apagar las tiras LED.

El integrado TIP120 es un transistor Darlington que permite un flujo de corriente colector de hasta 5 [A], con lo cual puede manejar sin problemas los 1.5 [A] esperados. Se decide utilizar este chip para acoplar las tiras LED al microcontrolador.

Para su correcto funcionamiento, basta con polarizar la juntura base-emisor, regulando la corriente que entra a la base mediante una resistencia. Si el ATmega polariza la base del transistor con 5 [V], y la corriente de base máxima soportada es de 120 [mA], entonces basta con utilizar una resistencia que limite la corriente por debajo de este nivel, y debido a la alta ganancia de este transistor, la corriente de base puede ser, sin problemas, muy pequeña, asegurando un correcto funcionamiento al mismo tiempo que un menor consumo. Si se utiliza una resistencia en la base de 1 [kΩ], entonces, aplicando Ley de Kirchhoff, la tensión que cae sobre ella está dada por:

$$V_R = V_{in} - V_{BE} = 5V - 1.4V = 3.6 [V] \quad (6.1)$$

Por lo tanto, aplicando Ley de Ohm, la corriente en la base está dada por:

$$I_B = \frac{V_R}{R} = \frac{3.6V}{1k\Omega} = 3.6 [mA] \quad (6.2)$$

Valor de corriente bastante pequeño, y que está dentro de los límites del integrado. Así, y considerando que el pin de control elegido para la iluminación es el 13, el circuito de conexión al ATmega es el que se presenta en la Fig. 6.6.

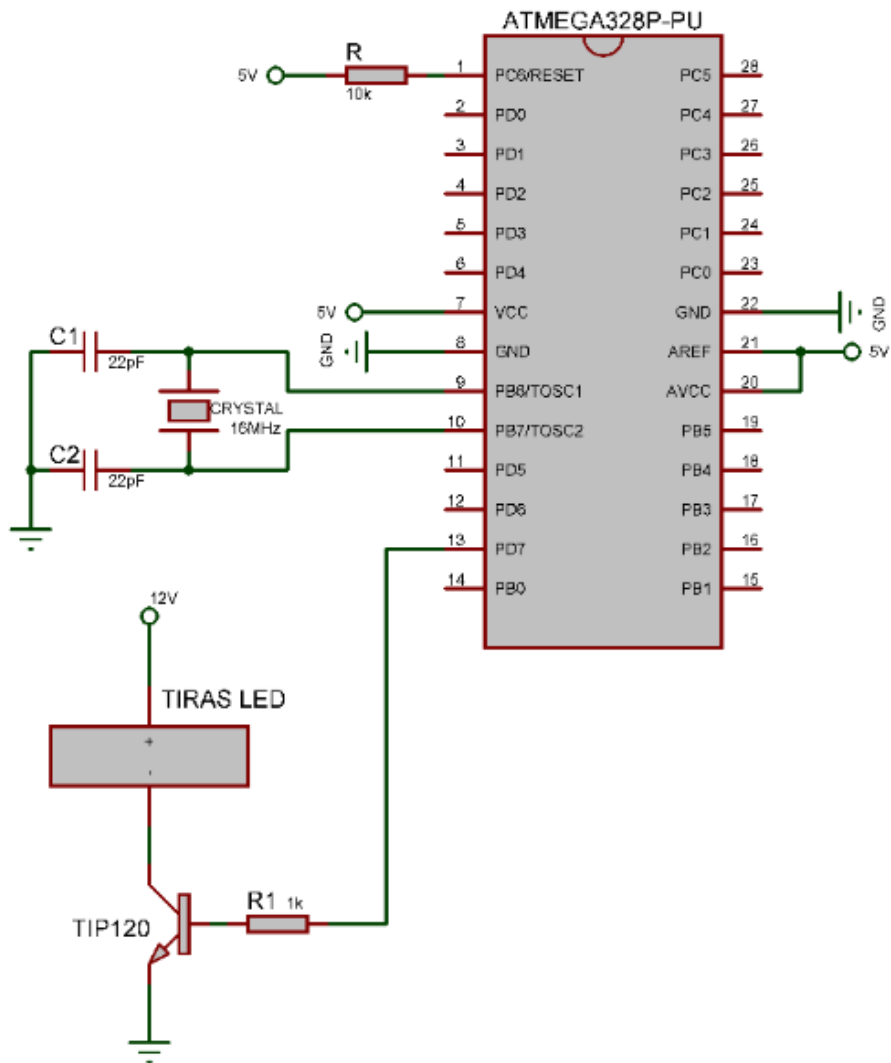


Fig. 6.6: Esquema de conexionado de las tiras LED al ATmega328P.

De esta forma se consigue preparar las condiciones de iluminación para el correcto sensado con las LDR.

### 6.03.3 Limit Switch

El módulo YL-99, al ser un switch normalmente cerrado, mientras no sufra ningún cambio físico, es decir, mientras no sea presionado el interruptor, permite aplicar a su salida una tensión en función de la tensión Vcc con el cual es alimentado. Por el contrario, al presionarse el interruptor, el circuito se abre y entrega a su salida un tensión 0 [V]. Mediante este funcionamiento se puede producir los 0 y 1 lógicos necesarios para controlar el circuito mediante la lectura del pin 11 del ATmega, tal como se había señalado anteriormente. El circuito resultante es el que se presenta en la Fig. 6.7.

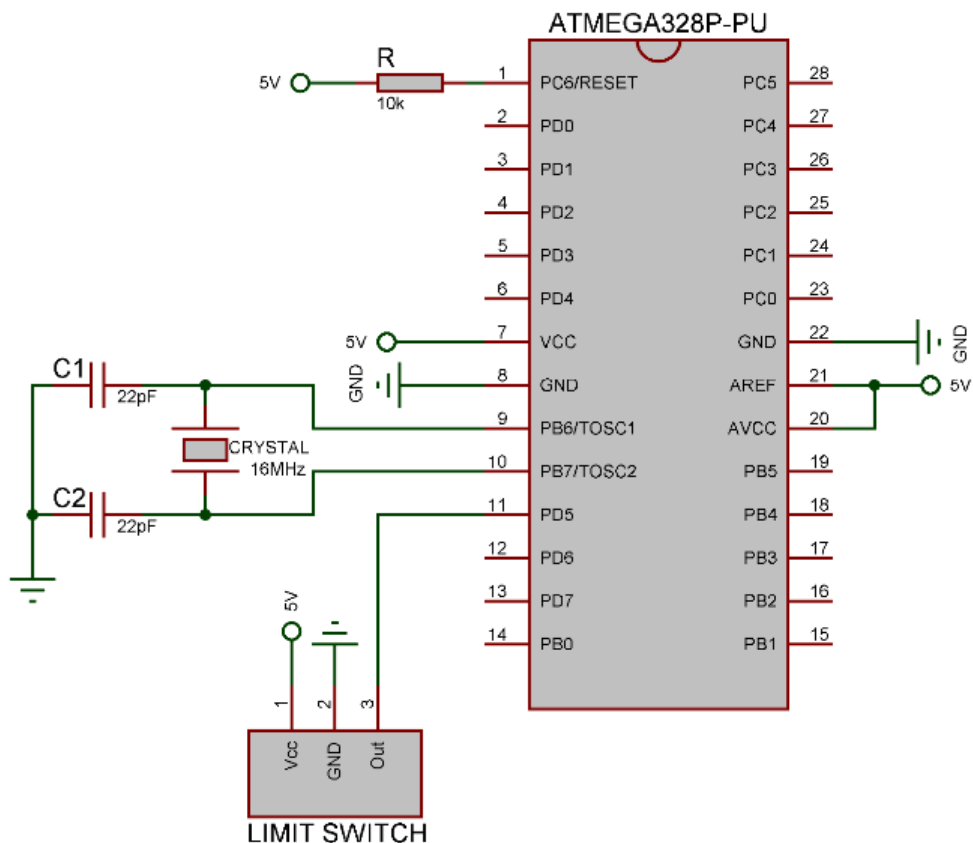


Fig. 6.7: Esquema de conexionado del módulo YL-99 al ATmega328P.

Cabe destacar que el funcionamiento normalmente cerrado del módulo implica una consideración adicional sobre el código, respectivo a cuándo se considera que la gaveta está abierta o cerrada. Considerando que el *limit switch* se encontrará al fondo de la gaveta, se produce lo siguiente:

- Si la gaveta está abierta, el switch se encuentra en reposo y entrega un 1 lógico a su salida.
- Si la gaveta está cerrada, el switch se encuentra presionado y entrega un 0 lógico a su salida.

El código debe considerar esta situación, en cuanto a cuándo activar la secuencia de detección. La detección debe comenzar en el instante en que el pin 11 lea un 0, entregado por el *limit switch*.

#### 6.03.4 Pantalla LCD

La pantalla LCD funciona como indicador inmediato, en terreno, de los resultados obtenidos por el sistema. Su comunicación al ATmega328P es en serie, y utiliza 6 GPIOs del microcontrolador para ello. A continuación se describe su conexionado.

Se alimenta con 5 [V] al pin 1. El pin 3 se usa para regular el contraste de los caracteres, y para ello se le conecta un potenciómetro de 10 [KΩ], necesario ya que leves variaciones en la resistencia provocan un cambio que podría dificultar la lectura visual de la información. El pin 15 se utiliza para regular el brillo de la pantalla, y si es útil conectarle un potenciómetro de hasta 1 [KΩ], se ha decidido dejar una resistencia fija de de 22 [Ω], de tal forma que siempre provea brillo. El pin 4 y 6 gestiona la escritura sobre la pantalla, controlada por el microcontrolador, por medio de los pines 14 y 16. Por último, los pines del 7 al 14 del LCD corresponden al bus de datos, el cual se encarga de transmitir los la información a imprimirse. Debido a que no se utilizarán caracteres complejos, no es necesario usar los 8 pines del bus, sino que con 4 de ellos basta, los pines 11, 12, 13 y 14, los cuales son controlados por los pines 16, 17, 18 y 19 del microcontrolador. El esquema resultante puede observarse en la Fig. 6.8.

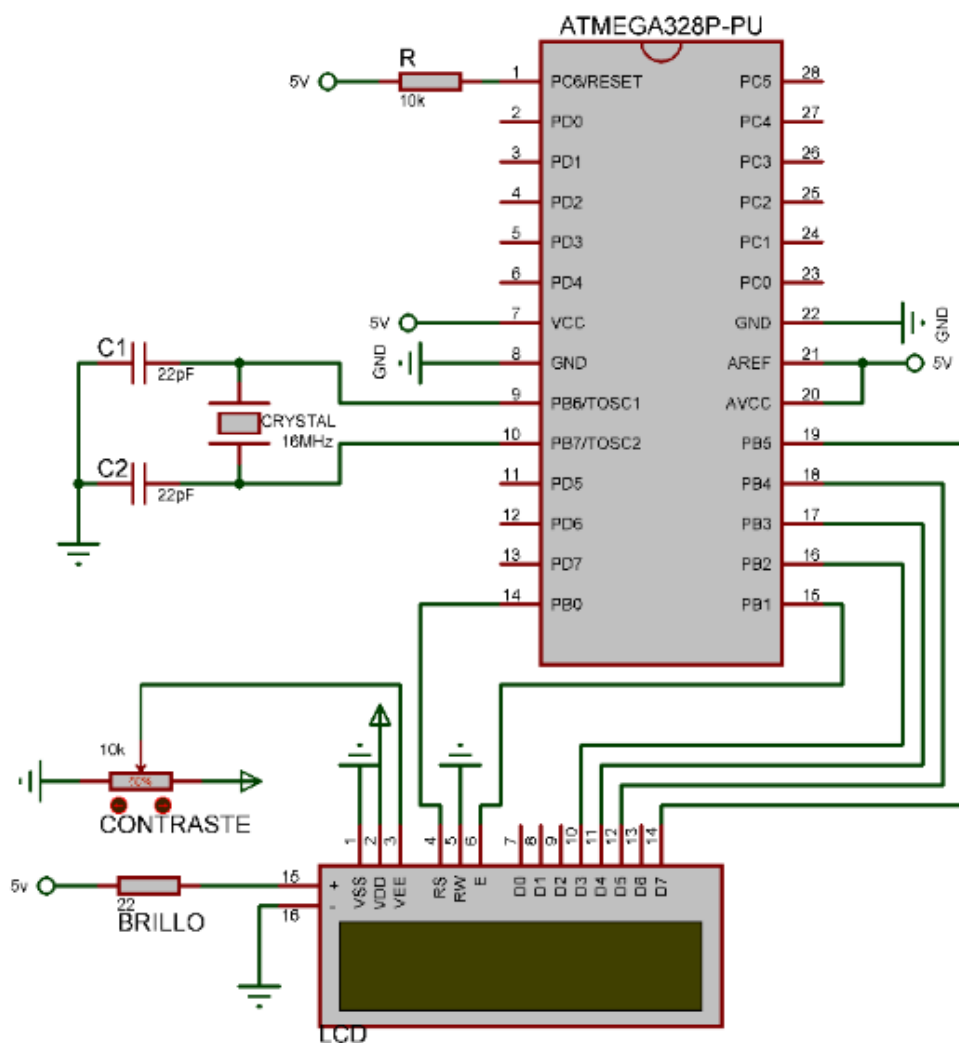


Fig. 6.8: Esquema de conexionado de la pantalla LCD 16x2 al ATMEGA328P.

## 6.04 Esquema Resultante

En la Fig. 6.9 se presenta el esquema total resultante del ATmega y sus periféricos.

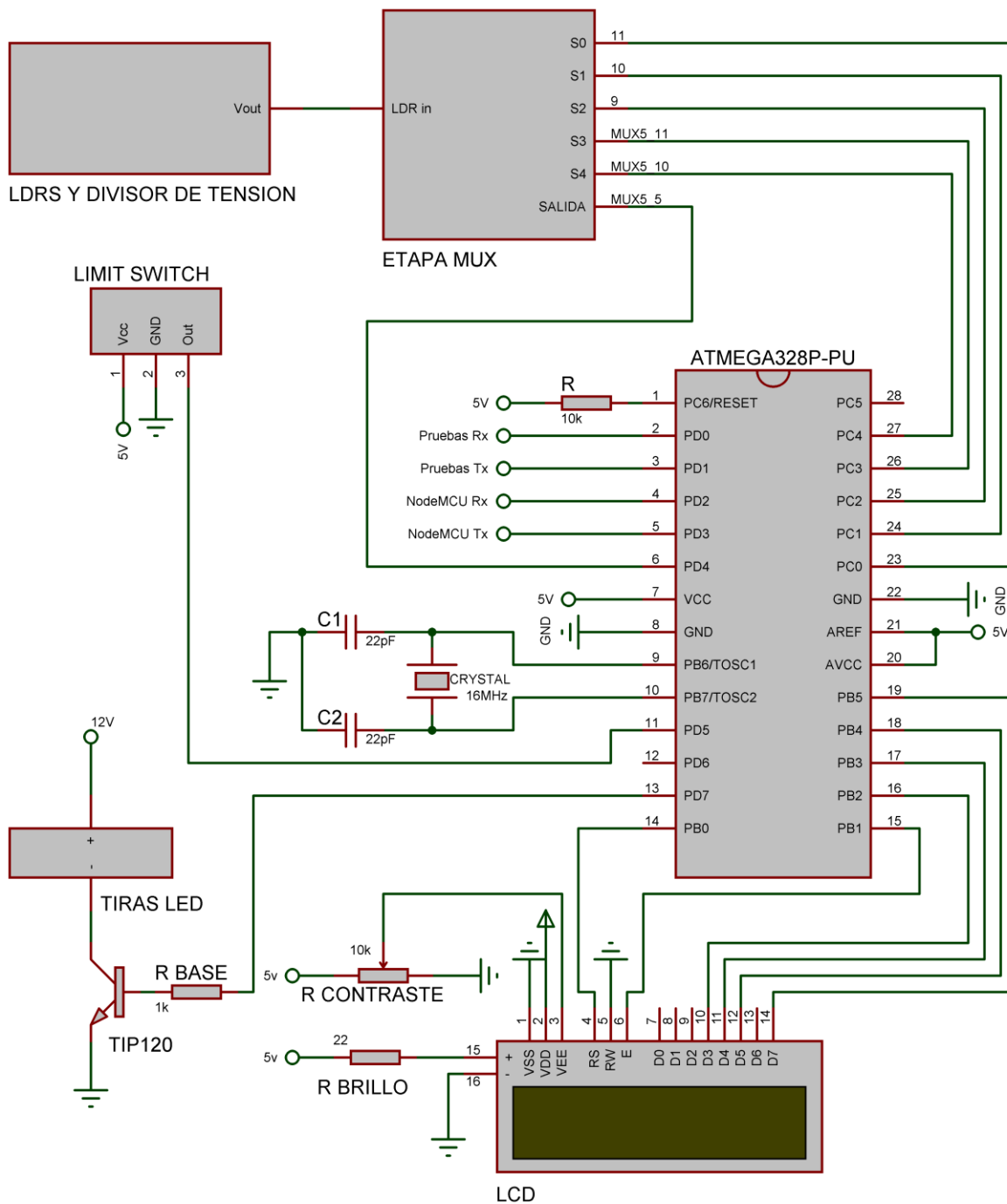


Fig. 6.9: Esquema de conexionado total de los periféricos al ATmega328P.

Cabe señalar las siguientes consideraciones respecto al anterior esquema:

- Las etapas de los LDRs, divisor de tensión y multiplexación se han dejado como bloques para una mejor visualización del esquema.
- Los pines 2 y 3 se han dejado disponibles para la comunicación serial con el computador, con el cual hacer pruebas durante el desarrollo.
- Los pines 4 y 5 se han dejado disponibles para la comunicación serial con el NodeMCU, cuyas consideraciones de funcionamiento, conexionado y comunicación quedan pendientes para el siguiente capítulo, donde se aborda con mayor detención.

## 6.05 Resumen capítulo 6

A lo largo de este capítulo se abordó el algoritmo de funcionamiento del sistema de detección, así como todos los respectivos conexionados a los periféricos, a excepción del módulo NodeMCU.

Se puede concluir que para un correcto funcionamiento del sistema, además de requerir la alimentación prevista de 5 [V] para el microcontrolador, se requiere también una de 12 [V] y 1 [A] que alimente a la iluminación.

Con todo esto, si se monta el diseño sobre una protoboard, ya es posible obtener un satisfactorio resultado virtual del funcionamiento, en cuanto a la detección. Bajo las correctas condiciones de iluminación, el circuito arroja, por medio de la pantalla LCD la cantidad detectada de LDRs a la sombra, situación que representa la presencia de insertos. Los resultados obtenidos hasta aquí se presentan en el Capítulo 8.

En este punto, al sistema electrónico solo le falta la comunicación Wi-Fi para transmitir los datos a un computador. De esto se encarga el NodeMCU, módulo que se estudia a continuación, en el Capítulo 7.

## CAPÍTULO 7: COMUNICACIÓN Y NODEMCU

Como se había adelantado en el Capítulo 3.02.4, el método de comunicación seleccionado para enviar los datos desde el sistema electrónico a un PC es la tecnología Wi-Fi, mientras que el módulo seleccionado para encargarse de esta tarea es la placa de desarrollo NodeMCU v1.0.

En el presente capítulo se repasan las principales características del NodeMCU, con el propósito de acoplarlo al sistema, de tal forma que se pueda comunicar tanto con el ATmega328P, como con un servidor LAN por medio de Wi-Fi. Además, siendo el NodeMCU –esencialmente– un microcontrolador, también se diseña un algoritmo de funcionamiento que le permita gestionar la comunicación, acorde a los requerimientos del sistema. El capítulo se subdivide en las siguientes partes:

1. Configuración del NodeMCU: Se presenta la configuración para la puesta en marcha del NodeMCU V1.0.
2. Acoplamiento al sistema: Se diseña un circuito intermediario que permita adaptar la tensión de trabajo del NodeMCU a la del ATmega328P.
3. Infraestructura de internet para la recepción: Se especifican los protocolos usados para comunicar el NodeMCU al servidor LAN.
4. Algoritmo de comunicación: Se diseña el algoritmo con el cual trabajará la placa de desarrollo para comunicarse con el ATmega y con el servidor LAN.

### 7.01 Configuración del NodeMCU

La placa de desarrollo NodeMCU v1.0, es un módulo electrónico basado en el SoC ESP8266EX. Este SoC es un chip que integra dos elementos fundamentales: un microcontrolador de 32 bit, el *Tensilica L106*, y un módulo Wi-Fi, además de una memoria RAM. A su vez, el chip ESP8266EX forma parte del módulo ESP-12, el cual es una placa que integra una antena para permitir la comunicación inalámbrica. Por último, el NodeMCU v1.0 integra al ESP-12, además de contener la electrónica adecuada para permitir una directa conexión USB y un rango de alimentación más amplio, además de una adecuada disposición de pines para su conexión en protoboard, características diseñadas para agilizar el desarrollo de proyectos.

Entre sus principales características, se encuentran las siguientes:

- Tensión de operación: 3.3V ó 5V
- Corriente de operación: 12-200 mA
- Corriente por GPIO: 15 mA
- Corriente standby: 200 uA

- Estándar inalámbrico: IEEE 802.11 b/g/n
- Rango de frecuencia: 2.412 - 2.484 GHz
- Poder de transmisión:
  - 802.11b :  $+16 \pm 2$  dBm (at 11 Mbps)
  - 802.11g :  $+14 \pm 2$  dBm (at 54 Mbps)
  - 802.11n :  $+13 \pm 2$  dBm (at HT20, MCS7)
- Sensibilidad de transmisión:
  - 802.11b : -93 dBm (at 11 Mbps, CCK)
  - 802.11g : -85 dBm (at 54 Mbps, OFDM)
  - 802.11n : -82 dBm (at HT20, MCS7)
- Método inalámbrico: Antena PCB, integrada en módulo
- Capacidades de I/O: UART, I2C, PWM, GPIO, 1 ADC
- Transmisión serie: 110 - 921600 bps, TCP Client 5
- Tipo conexión inalámbrica: STA / AP / STA + AP
- Tipo de seguridad: WEP / WPA-PSK / WPA2-PSK
- Protocolos de conexión: IPv4, TCP / UDP / FTP / HTTP

A continuación, en la Fig. 7.1 se presenta el diagrama de pines del NodeMCU v1.0.

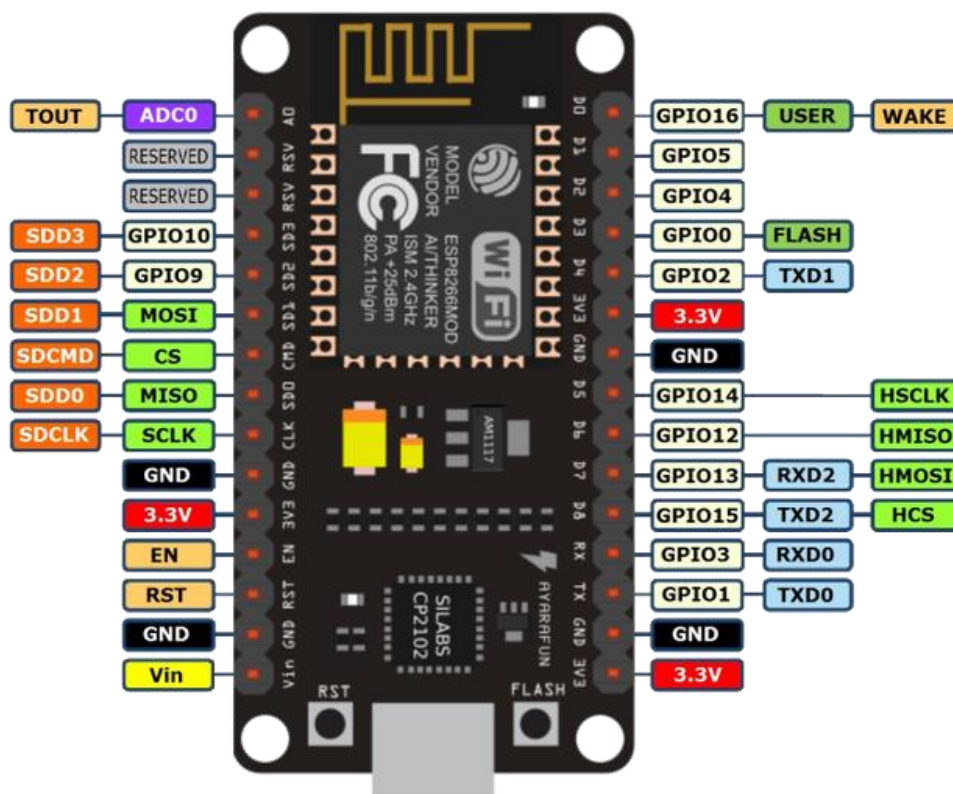


Fig. 7.1: Diagrama de pines del NodeMCU v1.0.

Contando esta tarjeta de desarrollo con la electrónica de acondicionamiento necesaria para la puesta en marcha del ESP8266EX, basta con conectarla a 5 [V] por medio del pin Vin.



Es importante resaltar que la tensión de operación de todos los GPIO de la tarjeta es de 3.3 [V]. Esta situación es un inconveniente a la hora de acoplarla al resto del sistema, más exactamente al ATmega, el cual funciona a 5 [V].

## 7.02 Acoplamiento al sistema

Le método de comunicación entre el NodeMCU y el ATmega es por cableado en serie, lo cual consiste en conectar el pin transmisor del NodeMCU al pin receptor del ATmega y viceversa, siendo luego el código de cada uno lo que gestione el intercambio de datos. Sin embargo, como ambos microcontroladores manejan niveles lógicos diferentes, es decir, 3.3 [V] y 5 [V], no es posible conectarlos directamente.

La incompatibilidad entre las tensiones de operación no debe pasarse por alto, ya que se produciría lo siguiente. Si el NodeMCU envía un 1 lógico, equivalente a 3.3 [V] a recibirse por el ATmega, como este nivel se encuentra dentro del rango de operación del ATmega y además por sobre el umbral de su estado alto, es bastante probable que el ATmega reconozca sin problemas la señal recibida. Sin embargo, si el ATmega envía un 1 lógico, equivalente a 5 [V] a recibirse por el NodeMCU, se tiene que el nivel de la señal se encuentra muy por sobre el del NodeMCU, y por tanto es altamente probable que la GPIO en cuestión, sino toda la tarjeta, se estropee.

Para conseguir un adecuado acoplo de ambos microcontroladores, es preciso contar con una etapa intermedia que convierta la tensión de operación de uno a la del otro. Una solución es el uso de una resistencia y un diodo que ajusten la tensión de 5 [V] del ATmega a los 3.3 [V] del NodeMCU. Sin embargo, este método no es bidireccional, y en etapas tempranas de la implementación y pruebas, así como en etapas de desarrollo posterior, el uso de este método podría resultar una limitante. En su lugar, se propone el uso de un adaptador de nivel bidireccional.

Un adaptador de nivel consiste en un circuito basado en un transistor que, trabajando con dos tensiones de referencia distintos, transforma de un nivel lógico a otro, en base a dichos niveles de referencia. Para esta aplicación se decide utilizar el 2N7000, que corresponde a un MOSFET canal N de pequeña señal, cuya disposición de pines se observa en la Fig. 7.2.

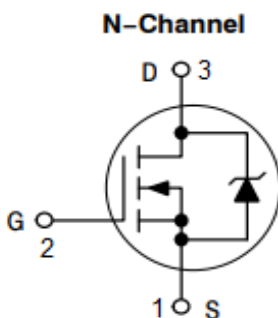


Fig. 7.2: Esquema interno y de conexionado del 2N7000.

El uso de este MOSFET como adaptador de nivel consiste en conmutarlo de corte a saturación. Para ello se presenta el esquema electrónico de la Fig. 7.3. El esquema muestra la conexión serie para la transmisión del ATmega al NodeMCU, desde el pin 5 del ATmega (Tx) al GPIO 3 del NodeMCU (Rx), a través del *Level Shifter*. El circuito para la transmisión inversa, desde el GPIO 1 al pin 4, es exactamente el mismo, pero se ha decidido no incluirlo para visualizar mejor el funcionamiento.

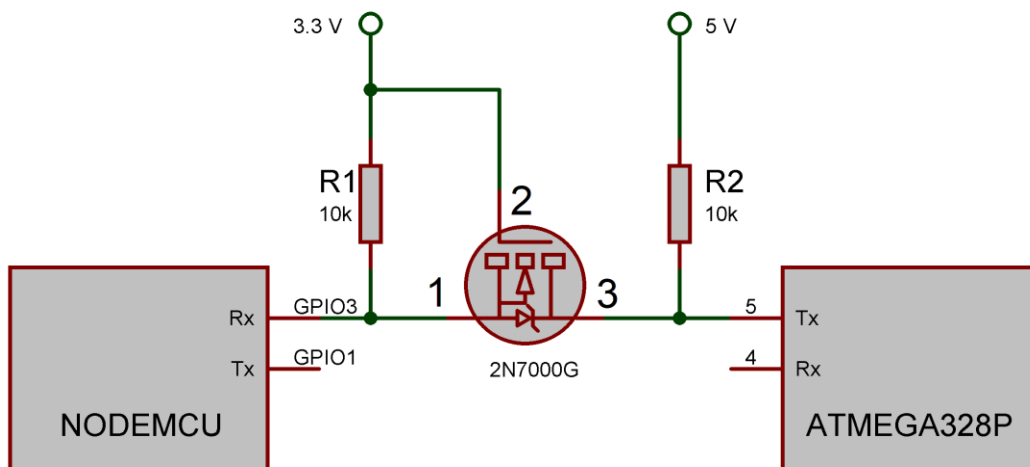


Fig. 7.3: Esquema de comunicación serie entre NodeMCU y ATmega328P. El circuito intermedio corresponde al adaptador de nivel, basado en el MOSFET 2N7000G.

Según su datasheet, la tensión de umbral  $V_{gs}$  del 2N7000 varía desde los 0.8 a 3 [V], pero como la referencia de los 3.3 [V] es la que controla al gate, y esta se encuentra por sobre los 3 [V], entonces se espera que el MOSFET trabaje siempre en conmutación.

Para la transmisión NodeMCU-ATmega, cuando el NodeMCU envíe un 1 lógico, la tensión  $V_{gs}$  será nula, y por tanto el Source-Drain estará abierto, cayendo entonces toda la referencia de los 5 [V] sobre el ATmega, y produciéndose así la transmisión del 1 lógico. Por el contrario, cuando el NodeMCU envíe un 0 lógico, se superará el umbral  $V_{gs}$  y el Source-Drain quedará en corto circuito, conectando al ATmega a los 0 [V], y produciéndose así la transmisión del 0 lógico.

Para la transmisión ATmega-NodeMCU, cuando el ATmega envíe un 1 lógico, el Source-Drain estará abierto y toda la referencia de los 3.3 [V] caerá sobre el NodeMCU, produciéndose así la transmisión del 1 lógico. Por el contrario, cuando el ATmega envíe un 0 lógico, el diodo interno Drain-Substrate se polarizará y provocará que el  $V_{gs}$  supere su umbral, quedando el Source-Drain en corto circuito, conectando al NodeMCU a los 0 [V], y produciéndose así la transmisión del 0 lógico.

### 7.03 Infraestructura de internet para la recepción

Los datos recogidos por el sistema, y más en concreto por el NodeMCU, han de ser enviados a una base de datos con tal de poder ser almacenados en un PC. Además, esta base de datos debe estar asociada a un servidor web que reciba los datos transmitidos a internet. Es bajo estas necesidades que se decide utilizar el software libre *Uniform Server*, el cual es un sistema *WAMP*, es decir, una infraestructura de internet basada en: *Windows* como sistema operativo, *Apache* como servidor web, *MySQL* como gestor de base de datos, y *PHP* como lenguaje de programación. Es necesario contar con este servidor para poder comprobar la correcta recepción de los datos.

Teniendo instalado *Uniform Server* en el computador en el cual hacer las pruebas de funcionamiento, cuya ventana de operación se visualiza en la Fig. 7.4, y teniendo arrancado tanto *Apache* como *MySQL*, se procede a crear dos elementos fundamentales: la base de datos y el servidor web.

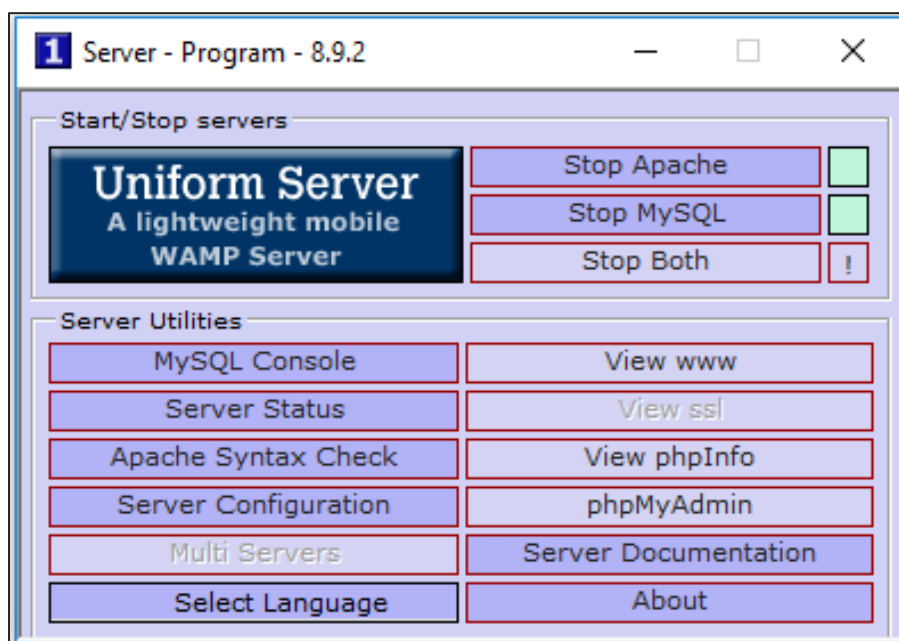


Fig. 7.4: Ventana de operación del sistema WAMP utilizado, *Uniform Server*.

La base de datos, gestionada por *MySQL*, se vale de la herramienta *phpMyAdmin*, la cual permite, por medio de páginas web, manejar y administrar a *MySQL*. Con este propósito, se ingresa a *phpMyAdmin* para crear una base de datos, y en ella crear una tabla, que es la estructura que permitirá almacenar ordenadamente los datos recibidos, así como también visualizarlos.

El servidor web es el encargado de la recepción de datos vía *Wi-Fi* para enviarlos a la base de datos. Este servidor se crea por medio de un *script* en *PHP*, y estará programado para esperar la conexión *Wi-Fi* por parte del NodeMCU, y de inmediato recibir los datos por

medio del método *HTTP POST*. Este método es un protocolo para crear formularios, en base a una cadena de caracteres con un orden preestablecido. El *script* en *PHP* mencionado, que gestiona el método *POST*, se adjunta al **Anexo C**.

El resultado final de la ejecución del *script* es la actualización de los datos en la tabla de *phpMyAdmin*, la cual contiene 33 filas, de las cuales 32 se usan para guardar los bit que representan la presencia/ausencia en cada una de las posiciones, mientras que la última fila queda reservada para guardar el conteo total de insertos durante la detección, el cual es el principal dato de interés en cuanto a la gestión de inventario, tarea última para la cual está diseñado este sistema. Así, los datos de esta tabla se actualizan solo cuando ocurre un evento de sensado por parte del sistema electrónico.

#### 7.04 Algoritmo de comunicación

El algoritmo de funcionamiento del NodeMCU está asociado al algoritmo de sensado correspondiente al ATmega, ya que es este el elemento que recoge los datos de interés y los envía al NodeMCU, como pudo observarse anteriormente en la Fig. 6.4. A continuación se describen las principales consideraciones y operaciones del algoritmo, para finalizar con un diagrama de flujo que lo represente, así como con el código real que maneja al microcontrolador.

Primeramente, se inicializan todas las variables a utilizar, entre las cuales hay un sustancial uso de cadenas de texto, característica muy propia de los protocolos de internet. Cabe destacar la especial importancia de incluir las datos de autenticación de la red de Wi-Fi a la cual se conectará el sistema, es decir, el ID y contraseña, así como también es importante incluir la URL y dirección IP del computador en la cual estará alojado el servidor *Apache*.

Terminada la inicialización y conexión, el algoritmo del NodeMCU entra en una secuencia que se repite indefinidamente, y que ante todo se mantiene chequeando el estado del *buffer* del puerto serie. El envío de datos al servidor comienza cuando el algoritmo detecta bits que leer en el *buffer*, datos que, en tal caso, han sido enviados por el ATmega a partir de un nuevo evento de detección por parte de los sensores.

Una vez que se detectan bits en el *buffer*, se leen uno por uno a medida que se van guardando en un *array* o vector, el cual, al final de la lectura del puerto serie, contendrá el estado lógico de los 32 insertos, siendo estos 1 ó 0, según haya presencia o ausencia.

Luego de leer las 32 posiciones, se hace la lectura del guión ("-"), la cual se ignora, para luego hacer la lectura de los dos siguientes enteros correspondientes a la cantidad total de insertos detectada.

Entonces, contando ya con los datos correspondientes a la cantidad total y a la presencia/ausencia en cada posición, se procede a crear una cadena de caracteres acorde al método *POST*. Este método permite enviar variables que consisten en una etiqueta y su valor, separadas por un signo "=". A su vez, para enviar varias variables de una sola vez, estas deben ser concatenadas en una cadena de caracteres, con cada una de las variables separadas entre sí por un carácter "&". El resultado de toda esta cadena de datos queda,

finalmente, de la forma en que se aprecia en la Fig. 7.5, y contiene toda la información relevante a transmitirse.

```
Cantidad=7&A1=0&A2=0&A3=0&A4=0&A5=0&A6=0&A7=0&A8=0&A9=1&A10=0&A11=0&A12=0&A13=0&A14=1&A15=0&A16=0&A17=1&A18=1&A19=0&A20=1&A21=0&A22=0&A23=0&A24=0&A25=1&A26=1&A27=0&A28=0&A29=0&A30=0&A31=0&A32=0
```

Fig. 7.5: Ejemplo de la cadena de caracteres de datos resultante en el formato del método *POST*.

Esta cadena se concatena, a su vez, a una nueva cadena de caracteres de varias líneas según el método *POST*, que incluye otras informaciones relevantes sobre la transmisión, como la dirección IP del computador, la URL del servidor y el largo de la cadena de información.

Entonces el *NodeMCU* hace un llamado al servidor y en seguida envía toda la cadena de caracteres del método *POST*. El servidor ha de recibir la información y responder si se ha recibido correctamente o no, antes de procesarla. El *NodeMCU* recibe la respuesta del servidor y finalmente vuelve al inicio de la secuencia, para comprobar si hay nuevos datos que leer del *buffer*.

Todo el anterior algoritmo descrito se resume en el diagrama de flujo de la Fig. 7.6. Así mismo, el código que refleja el algoritmo anteriormente descrito queda señalado en el **Anexo B**. Este se programa en C mediante el *IDE* de *Arduino*, y se carga al microcontrolador por conexión USB, directamente al *NodeMCU*.

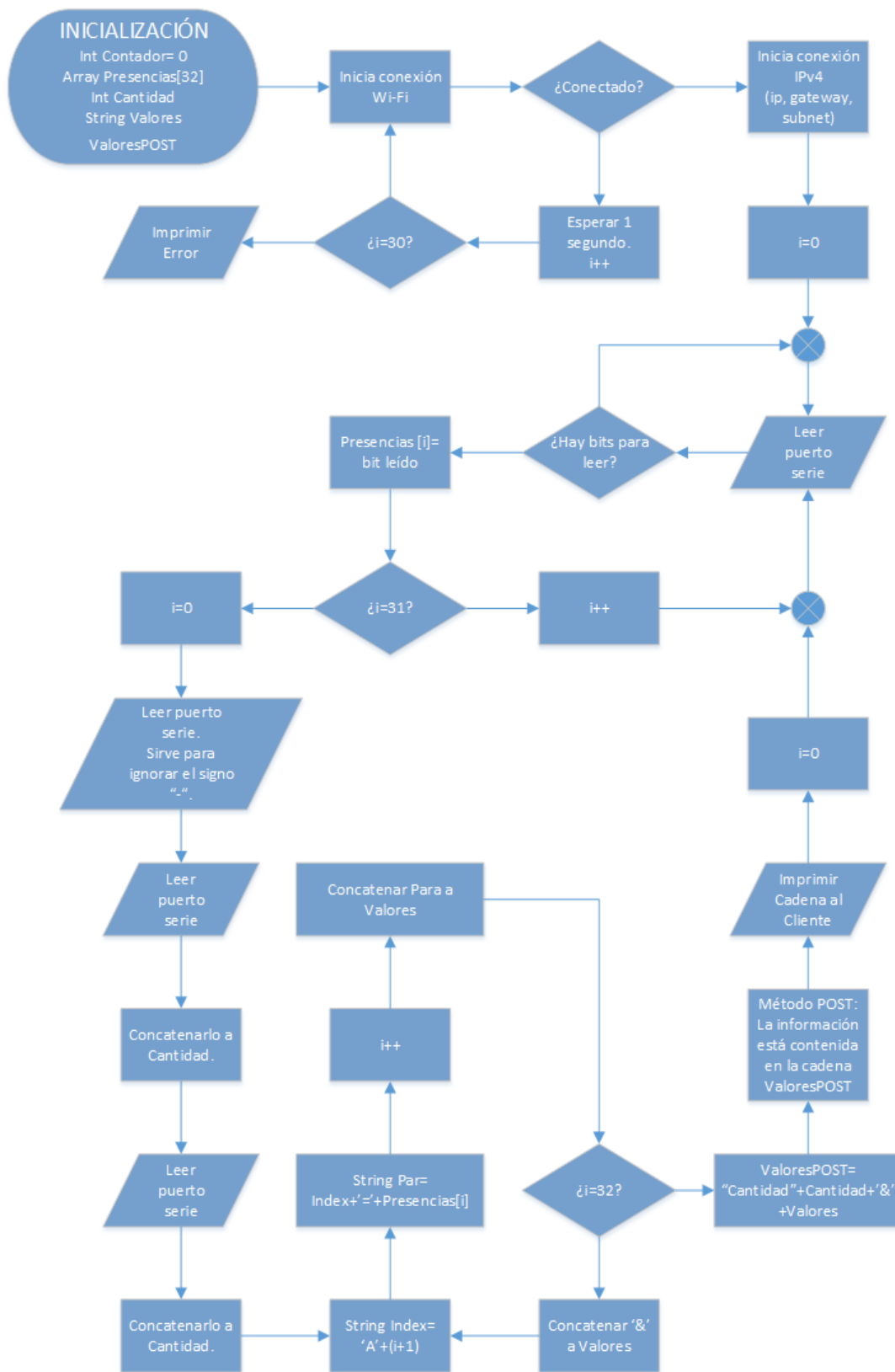


Fig. 7.6: Diagrama de flujo del algoritmo de funcionamiento para el NodeMCU.

## 7.05 Resumen capítulo 7

A lo largo del presente capítulo se abordó el algoritmo de funcionamiento y las condiciones necesarias para la puesta en marcha y acoplo del NodeMCU.

Respecto a la electrónica, se concluye que el sistema requiere necesariamente de una referencia de 3,3 [V], así como del adaptador de nivel, que permita el acoplo de los microcontroladores.

También se concluye que el sistema, además de requerir toda la electrónica anteriormente diseñada, requiere también de un entorno informático que permita recibir adecuadamente los datos recogidos y guardarlos en una base datos. Aquí es donde entra en juego el servidor *WAMP*.

En este punto, ya se tiene diseñadas todas las etapas del sistema. El sensado, en base al funcionamiento descrito de todas las etapas en su conjunto, debiera estar operativo y funcionar adecuadamente. El resultado final, así como la comprobación del funcionamiento general y el estudio de sus parámetros asociados, se aborda en el siguiente capítulo.

## CAPÍTULO 8: RESULTADOS EXPERIMENTALES

A partir del circuito electrónico diseñado en los anteriores capítulos, se monta un circuito en protoboard que permita corroborar el comportamiento descrito y así validar el funcionamiento del sistema. El circuito experimental obtenido es el que se observa en la Fig. 8.1.

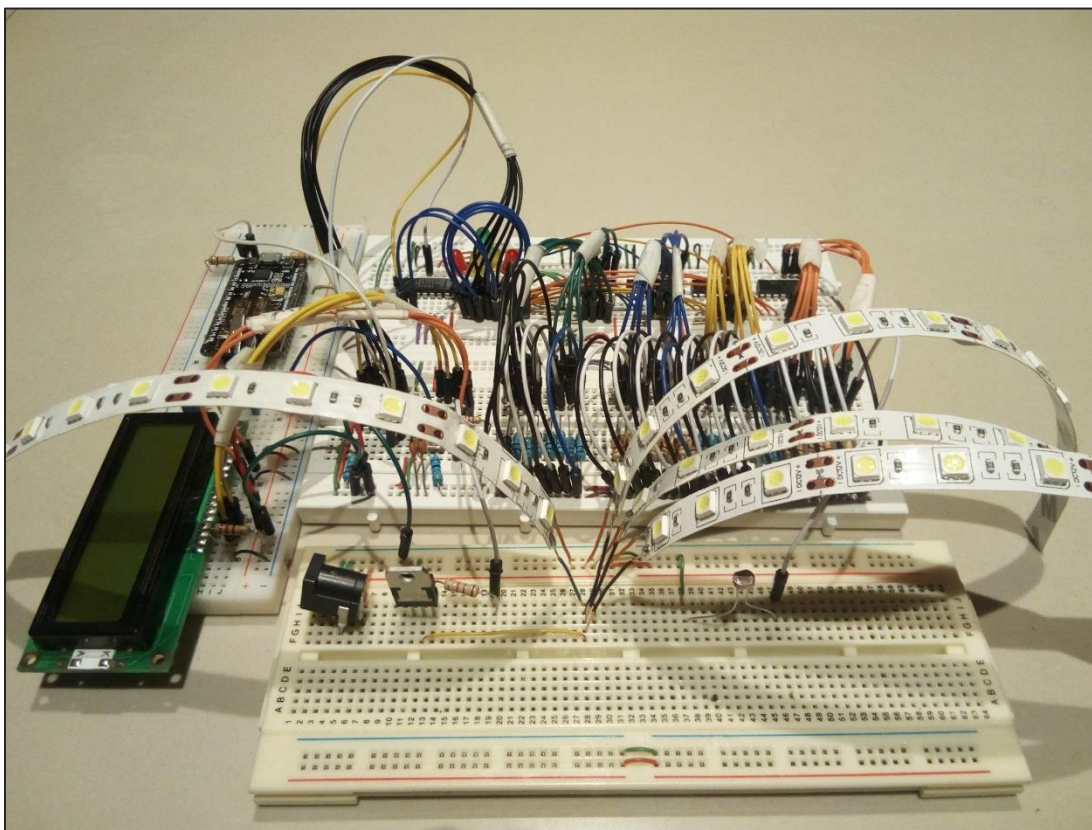


Fig. 8.1: Circuito electrónico de prueba, montado sobre protoboard.

Este circuito experimental permite llevar a cabo diversas pruebas y mediciones que indican el adecuado funcionamiento del sistema electrónico diseñado. Las pruebas y mediciones a realizarse sobre el circuito son las siguientes:

1. Parámetros de iluminación: Se estudian los parámetros eléctricos presentes en la etapa de iluminación y se comparan con los valores de diseño.
2. Parámetros de la LDR: Se estudian los parámetros eléctricos presentes en una LDR de prueba y en su respectiva rama, y se comparan con los valores de diseño.



3. Conteo en display LCD: Se corrobora que el conteo se efectúa adecuadamente en función de entras de prueba.
4. Recepción a la base de datos: Se corrobora la recepción del conteo obtenido a la base de datos en *phpMyAdmin*.
5. Consumo del sistema: Se estudia el consumo total y parcial del sistema.

## 8.01 Parámetros de iluminación

Del correcto comportamiento de la iluminación, proveída por la tira LED, depende el correcto funcionamiento del principio de funcionamiento, dado por la LDR. A continuación se estudian los parámetros eléctricos obtenidos en la etapa de iluminación.

Si bien el encendido y apagado de la tira LED lo controla el pin 13 del ATmega328P, para efectos de mediciones se conecta la resistencia base del Darlington a 5 [V] o GND según corresponda, para así analizar cada estado.

En primer lugar se observa que efectivamente la tira LED se enciende al conectar la base a 5 [V], y se apaga al conectarla a masa, como se observa en la Fig. 8.2. Luego, al conectar la base al pin 13 del microcontrolador y activar el *limit switch*, se produce el *flash* esperado.

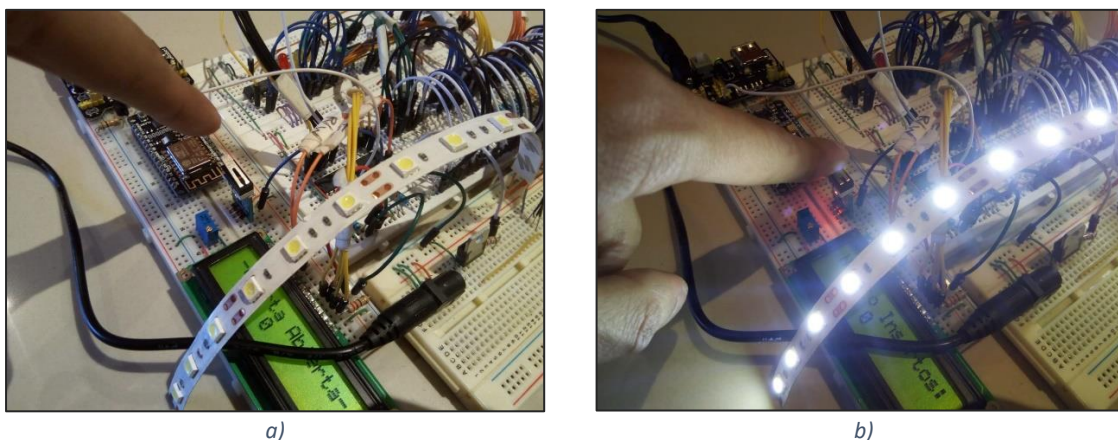


Fig. 8.2: Estado de tiras LED: a) OFF. b) ON.

Corroborado el funcionamiento, se procede a medir los parámetros: tensión colector-emisor, tensión en la tira LED, corriente de base y corriente colector. Los resultados se observan en la Tabla 6. La resistencia  $R_b$  usada es de 10.3 [k $\Omega$ ].

Tabla 6: Principales parámetros eléctricos de la etapa de iluminación.

Vcc Tira LED [V]	V <sub>PIN-13</sub> ATmega [V]	Estado tira LED	V <sub>C-E</sub> [V]	V <sub>T. LED</sub> [V]	I <sub>B</sub> [mA]	I <sub>C</sub> [mA]
12.3	0	OFF	7.9	3.8	0	0
	4.5	ON	0.9	10.6	0.32	150

## 8.02 Parámetros de la LDR

A continuación se comprueba que la LDR, en conjunto al acondicionamiento, arroje una señal de dos estados lógicos, según detecte o no una luminosidad significativa.

Se recalca que el presente estudio no cuenta con una gaveta con las características señaladas en el Capítulo 4.04, ni tampoco contempla el diseño de una maqueta que lo reemplace. Entonces, considerando que no se cuenta con las condiciones lumínicas idóneas, se espera que el resultado sea, si bien no exacto al diseño, cercano y que, cuando menos, los niveles de la señal de salida se asemejen a los de una señal digital que pueda ser interpretada por el microcontrolador.

Teniéndose la tira LED en correcto funcionamiento, se dispone por sobre la LDR y de cara a ella, a unos 5 [cm] de distancia, como se presenta en la Fig. 8.3. Se utiliza solo una tira en lugar de las cuatro, para facilitar la prueba.

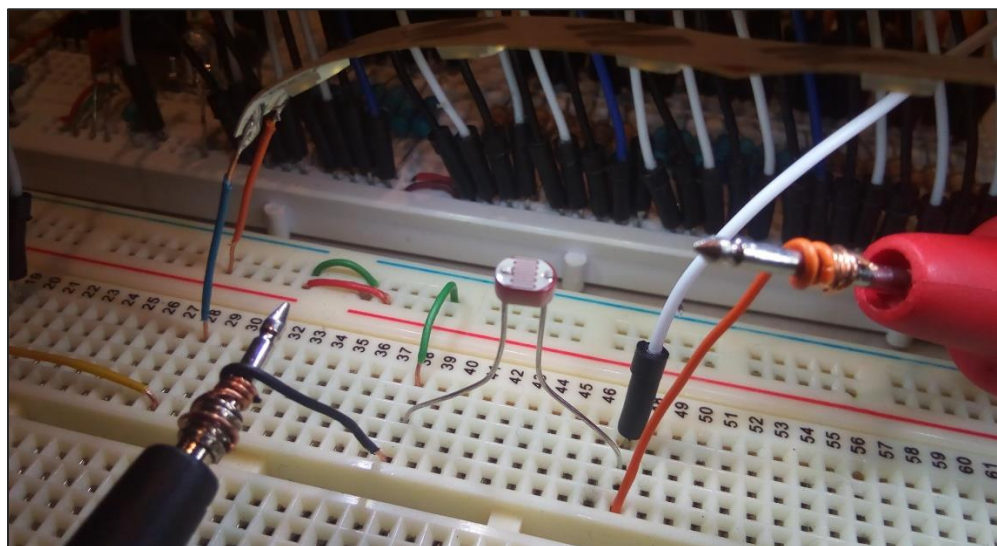


Fig. 8.3: Tira LED dispuesta por sobre la LDR, a corta distancia.

Se procede a medir la corriente y tensiones presentes en la rama del divisor de tensión, para los dos estados de la tira LED, encendida y apagada.

Para el caso de la tira LED encendida, se puentea la base del transistor a 5 [V] para que se mantenga encendida ininterrumpidamente durante la prueba. Esta prueba se aprecia en la Fig. 8.4.

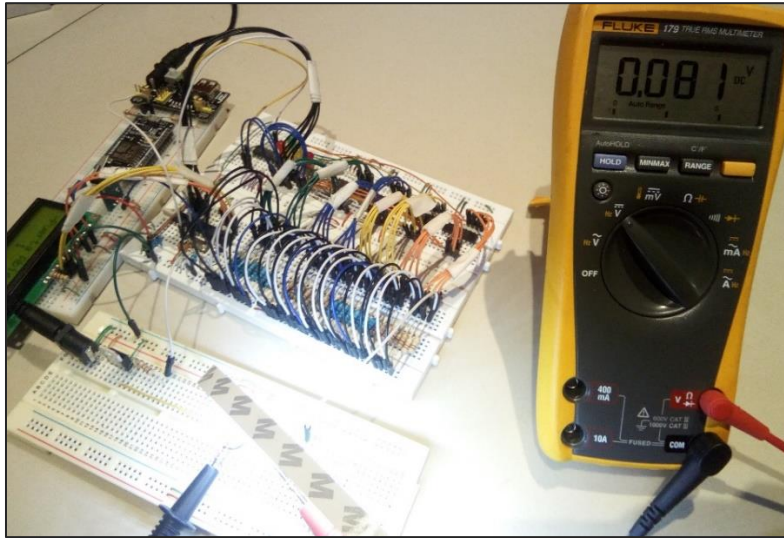


Fig. 8.4: LDR bajo plena luz de la tira LED. A la derecha se observa un multímetro midiendo tensión.

Así mismo, se analiza la rama para la LDR con la luz bloqueada, manualmente, como se observa en la Fig. 8.5.

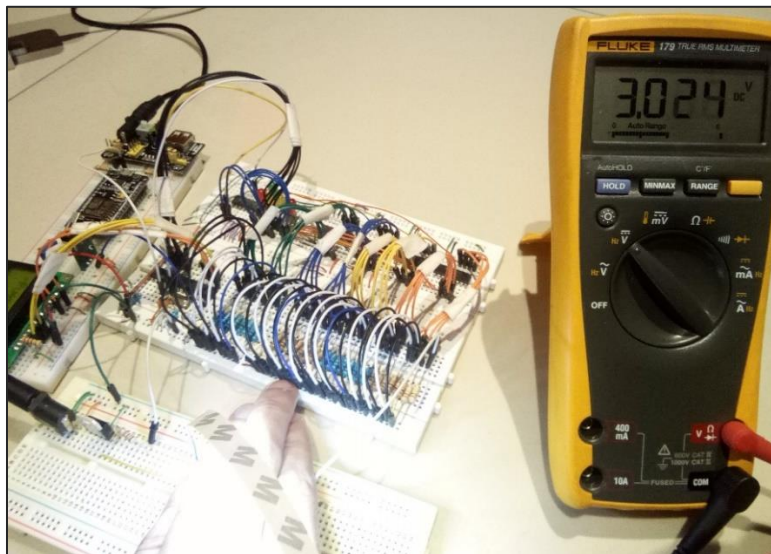


Fig. 8.5: LDR bajo la sombra, luz de la tira LED, bloqueada manualmente.

Como se esperaba de las condiciones lumínicas no-ideales, los niveles de tensión obtenidos, sobre todo la LDR bajo sombra, se aleja un tanto de lo esperado. No obstante, los niveles de tensión obtenidos se encuentran ambos dentro de los rangos adecuados de una TTL. Los resultados se detallan en la

Tabla 7: Parámetros eléctricos de una rama del divisor de tensión.

	$V_{LDR}$ [V]	$V_R$ [V]	$I_{LDR}$ [ $\mu$ A]
<b>Plena luz</b>	0.08	4.83	60
<b>Luz bloqueada</b>	3.02	1.9	30

Los niveles de la señal de salida que llega a la etapa MUX son 0.08 y 3.02 [V] respectivamente. El resto de la tensión recae sobre la resistencia R. Hay alrededor de 0.1 [V] perdido producto del uso de protoboard, cables largos y gran cantidad de componentes. Además cabe resaltar que el significativo valor de resistencia en cada rama da como resultado una corriente pequeñísima, evitando que el consumo aumente.

### 8.03 Conteo en *display LCD*

El display LCD es el elemento indicador primario del conteo, y sirve como medio inmediato para comprobar que el conteo de las entradas se esté haciendo correctamente.

La prueba de conteo requiere tener una entrada lógica adecuada, obtenida mediante la iluminación controlada anteriormente descrita, la cual contempla el uso de una gaveta con compartimientos o cajón similar, y con las tiras LED ubicadas en la cara superior interna de la gaveta, todo en una correcta y ordenada disposición, en base a las consideraciones indicadas en el Capítulo 4.04. Conseguir esto dificulta la prueba de funcionamiento ya que exige contar con dicho mueble o, cuando menos, con una maqueta que lo reemplace y permita controlar la iluminación. Sin embargo, el diseño real de tal maqueta no se contempla en el presente estudio. Además, si bien podría pensarse que la luz incidente sobre las LDR podría controlarse simplemente obstaculizando la iluminación ambiente con algún objeto – o incluso con las manos–, hacerlo para las 32 LDR de forma simultánea no es para nada efectivo, ya que es importante revisar de forma estable el comportamiento para cada uno de los 32 estados. No obstante, para efectos de la prueba de conteo no es necesario que la entrada sea necesariamente por medio de las 32 LDR. Basta con tener a la entrada de la etapa MUX un adecuado estado lógico, señal que puede conseguirse de forma muy sencilla usando un *switch*, mediante la conexión/desconexión de un cable que simule serlo, uno para cada una de las 32 LDR, como se indica en la Fig. 8.6.



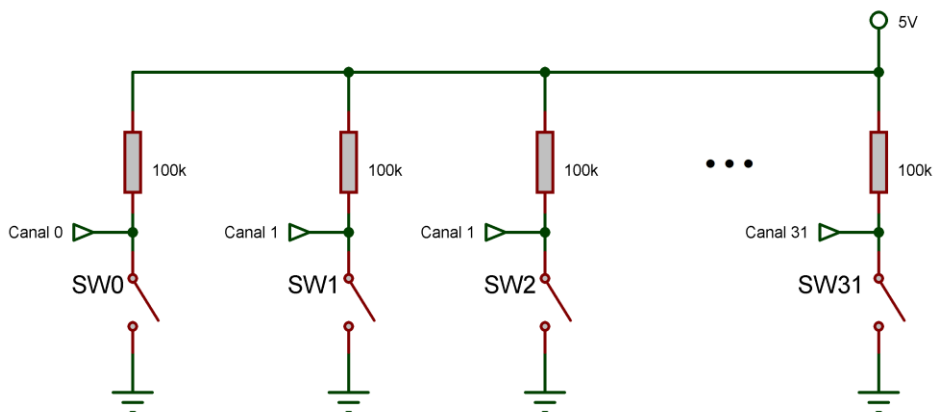


Fig. 8.6: Circuito de entrada con switches, para prueba de conteo.

Al usar este método, como la impedancia de entrada de los multiplexores es altísima, se produce un conveniente divisor de tensión entre la resistencia  $R$  de 100 [K $\Omega$ ] de cada rama y el pin del MUX. El resultado es que casi toda la tensión aplicada (5V) recae sobre la entrada del MUX, obteniéndose un 1 lógico. Por el contrario, al cerrar el switch de alguna rama, toda la tensión recae sobre la resistencia de 100 [K $\Omega$ ] y se produce un 0 lógico a la respectiva entrada del MUX. De esta forma se simulan los dos estados lógicos casi sin modificar el circuito original: la única diferencia es el reemplazo de las LDR por *switches*. Con esto ya es posible comprobar el correcto funcionamiento del conteo sin ir más allá del circuito en protoboard.

Se procede entonces a alimentar el circuito. A continuación se describen las respuestas arrojadas en el display para las diversas situaciones, en base al algoritmo descrito en el diagrama de flujo de la Fig. 4.13. Se parte con todos los *switches* cerrados, es decir, como si no hubiera ningún inserto, y luego se abre uno a uno cada switch hasta abrirlos todos, mientras se observa la respuesta en el *display*. Cabe recordar que el *limit switch* es el responsable de desencadenar cualquier cambio, y se vale de este elemento para el siguiente análisis.

Al alimentar el circuito, estando el *limit switch* en reposo, lo primero que arroja el display es “Puerta abierta”, como se presenta en la Fig. 8.7.



Fig. 8.7: Display al iniciarse.

El display se mantiene así hasta que el ATmega detecte algún cambio proveniente del *limit switch*. Cuando este es presionado, comienza la detección de entradas y de inmediato el display arroja el mensaje “Calculando...”, como se indica en la Fig. 8.8, el cual aparece por tan solo unas fracciones de segundo, mientras se realiza la detección.

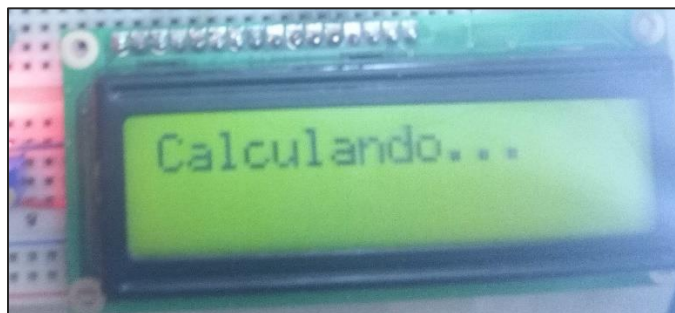


Fig. 8.8: Display al comenzar una detección por primera vez.

Al cabo de un instante, terminada la detección, el display arroja “Numero Insertos”, seguido de un número que indica la cantidad de entradas altas leídas. Como para esta prueba se parte con todos los switches cerrados, es decir, con 0 insertos, la cantidad arrojada es 0, como se indica en la Fig. 8.9.



Fig. 8.9: Mensaje arrojado cuando la cantidad de entradas altas detectadas es 0.

El mensaje aparece así mientras el *limit switch* se mantenga presionado. Al volver al reposo, se asume que la gaveta ha sido abierta, y el mensaje cambia. La primera línea vuelve al mensaje “Puerta abierta”, mientras que la segunda línea mantiene última cantidad detectada, como se indica en la Fig. 8.10.



Fig. 8.10: Mensaje arrojado cuando la gaveta es abierta y la última cantidad detectada fue 0.

Cuando el limit switch es presionado nuevamente, comienza una nueva detección. El display vuelve a arrojar "Calculando...", pero esta vez muestra también en su segunda línea la última cantidad detectada, como se indica en la Fig. 8.11.



Fig. 8.11: Mensaje arrojado cuando la gaveta vuelve a cerrarse y ha hecho una detección de 0 justo antes.

Si no ha habido ningún cambio en la cantidad de entradas altas, entonces se repite todo lo anterior. De lo contrario, si la cantidad de entradas altas ha sido modificada, por el ejemplo si ha aumentado en 1, entonces el display arroja la nueva cantidad, como se indica en la



Fig. 8.12: Mensaje arrojado cuando la cantidad de insertos es 1.

Cuando el *limit switch* vuelve al reposo, se vuelve a arrojar el mensaje “Numero Insertos”, mientras que la segunda línea mantiene el nuevo valor obtenido en la última detección, como se indica en Fig. 8.13.



Fig. 8.13: Mensaje arrojado cuando se vuelve a abrir la gaveta, esta vez con la nueva cantidad detectada.

Este comportamiento se repite para cualquier cantidad de estados altos detectados en el MUX, como se indica en la Fig. 8.14.

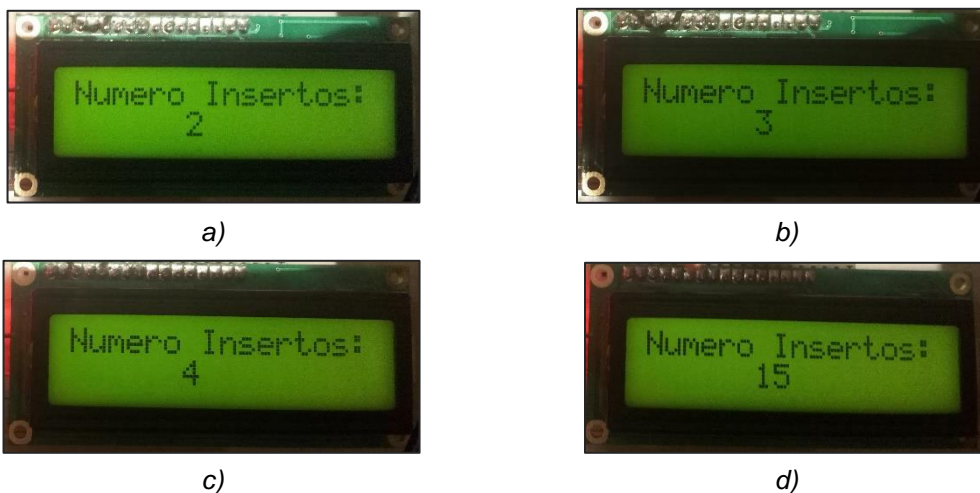


Fig. 8.14: Mensajes arrojados para varias cantidad de estado alto detectados.

Ya sea que la cantidad disminuya o aumente en uno o más elementos, el sistema es capaz de detectar desde 0 hasta 32 insertos y arrojar el respectivo mensaje a través del *display LCD*.

Por último es importante señalar que el algoritmo del microcontrolador está preparado para la situación hipotética en que la gaveta se abra durante el proceso de detección, que si bien es muy rápido, tiene una duración del orden de los milisegundos. Si esta situación ocurriera sin cuidado, es posible que el sistema arroje un valor errado por no alcanzar a terminar la lectura, provocando problemas. El sistema está preparado para arrojar el mensaje



“INTERRUMPIDO” bajo dicha situación, como se indica en la Fig. 8.15, cancelando la detección y obligando al usuario a volver a cerrar la gaveta para comenzar una nueva secuencia de detección.



Fig. 8.15: Mensaje arrojado cuando el proceso de detección es interrumpido.

Con todo esto, y corroborando que la cantidad arrojada por el *display* siempre se corresponde con la cantidad de *switches* abiertos, se comprueba el correcto funcionamiento del conteo. El sistema es capaz de detectar e informar efectivamente la cantidad de 1 lógicos a la entrada de la etapa MUX.

#### 8.04 Recepción a la base de datos

La comprobación del correcto funcionamiento de la comunicación entre el NodeMCU y el servidor se logra observando al administrador de la base de datos. Los cambios hechos en el circuito han de verse reflejados en la tabla de *phpMyAdmin*, de la forma en que se observa en la Fig. 8.16.

Con este propósito, se alimenta el sistema y se hace una prueba exhaustiva sobre las entradas, es decir, se cambia el estado lógico de cada entrada, una a una, observando si el cambio de cada entrada se ve reflejado en la respectiva fila de la tabla en *phpMyAdmin*. También se corrobora si el número contado en el *display LCD* es el mismo que arroja la tabla en su última fila, la número 33, que corresponde al número total de insertos. Cabe señalar que para estas pruebas se han utilizado los *switches* de prueba anteriormente señalados en la Fig. 8.6.

Para documentar la prueba anterior, y considerando que no vale la pena documentar el cambio de todas y cada una de las entradas, se decide presentar 4 muestras para distintas combinaciones de entradas: todas en 0, todas en 1, intercaladas una tras otra, y de forma aleatoria. El resultado de estas muestras se presenta en el Anexo D, y corresponde a la tabla total arrojada por el *phpMyAdmin* para la prueba de cada una de las combinaciones señaladas.

+ Options				id	ChipID	fecha	Valor
<input type="checkbox"/>				1	8454785	2018-10-11 21:43:13	0
<input type="checkbox"/>				2	8454785	2018-10-11 21:43:13	0
<input type="checkbox"/>				3	8454785	2018-10-11 21:43:13	0
<input type="checkbox"/>				4	8454785	2018-10-11 21:43:13	0
<input type="checkbox"/>				5	8454785	2018-10-11 21:43:13	0
<input type="checkbox"/>				6	8454785	2018-10-11 21:43:13	0
<input type="checkbox"/>				7	8454785	2018-10-11 21:43:13	0
<input type="checkbox"/>				8	8454785	2018-10-11 21:43:13	0
<input type="checkbox"/>				9	8454785	2018-10-11 21:43:26	1
<input type="checkbox"/>				10	8454785	2018-10-11 21:43:13	0
<input type="checkbox"/>				11	8454785	2018-10-11 21:43:13	0
<input type="checkbox"/>				12	8454785	2018-10-11 21:43:13	0
<input type="checkbox"/>				13	8454785	2018-10-11 21:43:13	0
<input type="checkbox"/>				14	8454785	2018-10-11 21:43:26	1
<input type="checkbox"/>				15	8454785	2018-10-11 21:43:13	0
<input type="checkbox"/>				16	8454785	2018-10-11 21:43:13	0
<input type="checkbox"/>				17	8454785	2018-10-11 21:43:26	1
<input type="checkbox"/>				18	8454785	2018-10-11 21:44:03	1
<input type="checkbox"/>				19	8454785	2018-10-11 21:43:13	0
<input type="checkbox"/>				20	8454785	2018-10-11 21:43:26	1
<input type="checkbox"/>				21	8454785	2018-10-11 21:43:13	0
<input type="checkbox"/>				22	8454785	2018-10-11 21:43:13	0
<input type="checkbox"/>				23	8454785	2018-10-11 21:43:13	0
<input type="checkbox"/>				24	8454785	2018-10-11 21:43:13	0
<input type="checkbox"/>				25	8454785	2018-10-11 21:43:26	1
<input type="checkbox"/>				26	8454785	2018-10-11 21:43:26	1
<input type="checkbox"/>				27	8454785	2018-10-11 21:43:13	0
<input type="checkbox"/>				28	8454785	2018-10-11 21:43:13	0
<input type="checkbox"/>				29	8454785	2018-10-11 21:43:13	0
<input type="checkbox"/>				30	8454785	2018-10-11 21:43:13	0
<input type="checkbox"/>				31	8454785	2018-10-11 21:43:13	0
<input type="checkbox"/>				32	8454785	2018-10-11 21:43:13	0
<input type="checkbox"/>				33	8454785	2018-10-11 21:44:03	7

Check All    With selected: Change    Delete    Export

Fig. 8.16: Tabla de datos de phpMyAdmin. Contiene una muestra aleatoria de estados de entrada.

Con esto se da por concluida la revisión del funcionamiento general del sistema, el cual ha conseguido exitosamente la detección de entradas y adquisición de datos al PC.

## 8.05 Consumo del sistema

El consumo del sistema es una medida de la energía que este requiere para operar, y resulta de particular importancia cuando se considera que el prototipo que pueda obtenerse de este diseño está pensando para funcionar a diario y continuamente durante horas.

El consumo energético depende tanto de la tensión aplicada, así como de la corriente demandada por cada uno de los componentes, o de forma más global, por cada una de las etapas. Queda definido por la siguiente expresión:

$$E = P * t \quad (8.1)$$

A su vez, la potencia eléctrica P usada en la anterior relación queda definida como:

$$P = V * I \quad (8.2)$$

Estas relaciones se aplican tanto para el consumo global como para el consumo parcial de cualquier etapa, en diferentes estados, que suelen ser dos: reposo, el lapso de tiempo en el cual el sistema se encuentra en *standby*, y en detección, mientras hace la lectura de señales y procesamiento de datos. Para efectos de este estudio se mide el consumo de cada una de las etapas en ambos estados, considerándose especialmente el estado de detección, que es un indicador de la máxima potencia que pudiese demandar el sistema, es decir, el peor caso.

### 8.05.1 Mediciones de corriente

Teniéndose las tensiones aplicadas según el diseño, basta con medir la corriente total en cada etapa, y más específicamente, la corriente a GND. Se parte midiendo la corriente de la etapa del acondicionamiento y microprocesador juntas, como se muestra en la Fig. 8.17, ya que ambas consisten casi solo de integrados.

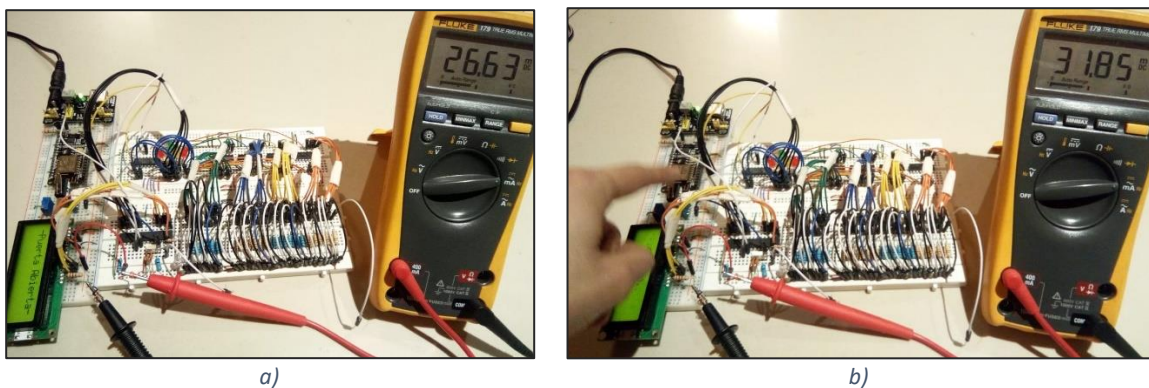


Fig. 8.17: Medición de corriente a GND en etapas de acondicionamiento (divisor de tensión y MUX) y microprocesador.

Luego se mide la corriente de la tarjeta *NodeMCU*, como se aprecia en la Fig. 8.18. El valor obtenido oscila entre 80 y 90 [mA], y es indiferente al cambio de estado del *limit switch*. Se considera indistintamente el nivel máximo de 90 [mA].

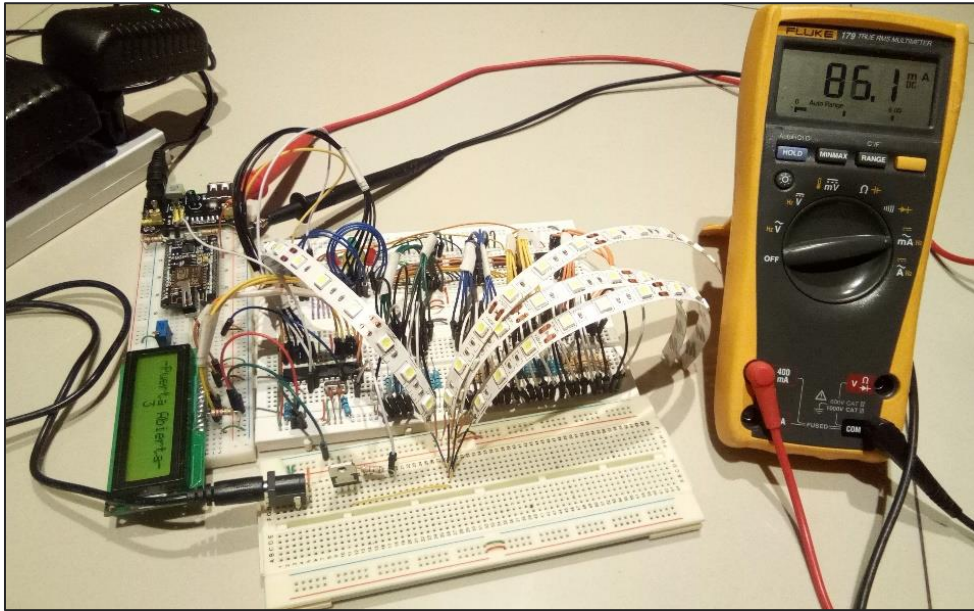
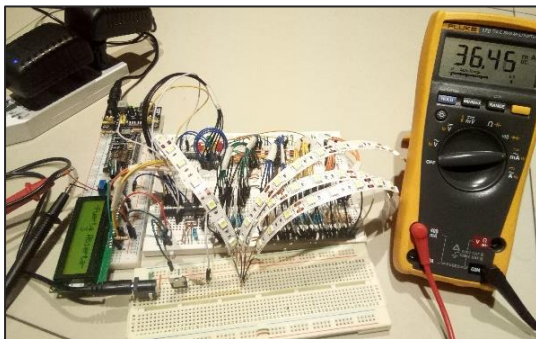
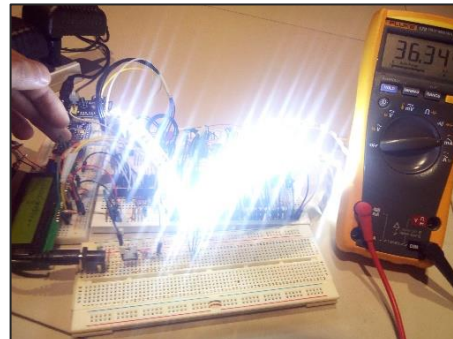


Fig. 8.18: Medición de corriente a GND en etapa de comunicación (*NodeMCU* y *Level shifter*).

Luego se mide la corriente del *display LCD*, como se observa en la Fig. 8.19. Presenta una muy leve variación de consumo durante el período de detección.



a)



b)

Fig. 8.19: Medición de corriente a GND en etapa de indicador LCD: a) En reposo. b) En detección.

Además se mide su consumo sin iluminación *back light*, esto es, con el pin 15 a GND, como se observa en la Fig. 8.20. Se aprecia cómo el consumo se reduce drásticamente sin el uso de iluminación en la pantalla.



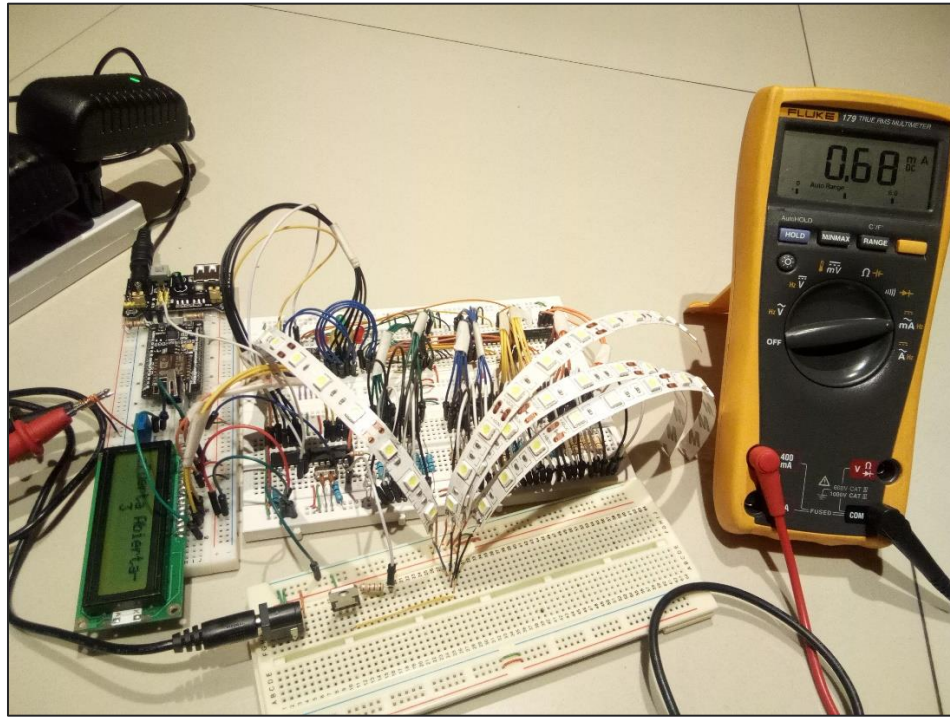
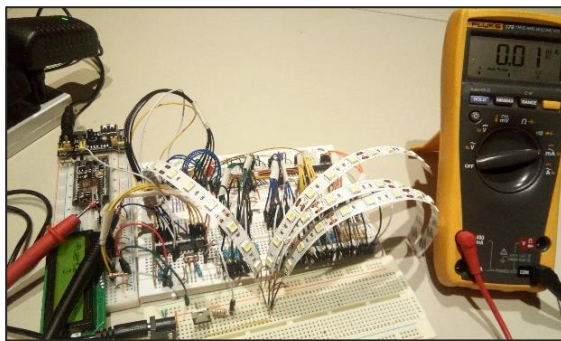
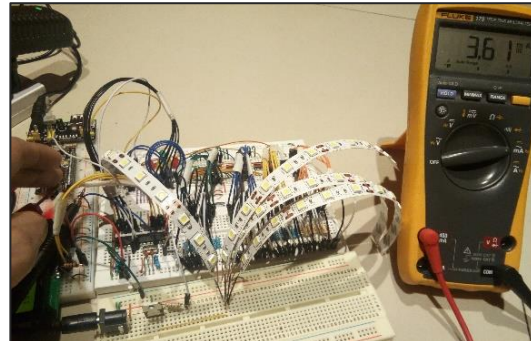


Fig. 8.20: Medición de corriente a GND en etapa de indicador LCD, sin back light.

Se mide el consumo del *limit switch*, como se aprecia en la Fig. 8.21. Su consumo es nulo al estar abierto, mientras que al cerrarse presenta un leve valor.



a)



b)

Fig. 8.21: Medición de corriente a GND en limit switch: a) En reposo (abierto). b) En detección (cerrado).

Por último, se mide la corriente en la etapa de iluminación, como se aprecia en la Fig. 8.22.

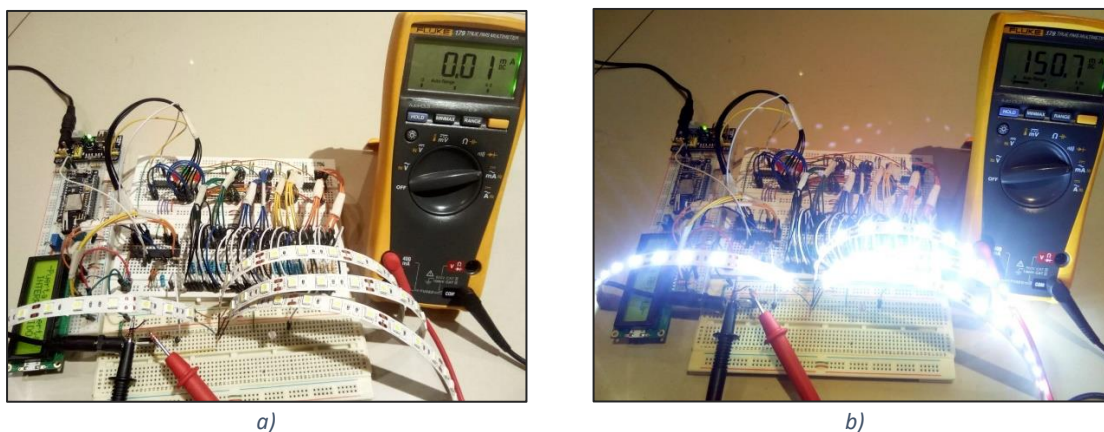


Fig. 8.22: Medición de corriente a GND en etapa de iluminación: a) En reposo. b) En detección.

### 8.05.1 Cálculo y resultados de consumo

Los valores obtenidos de todas las corrientes se resumen en la Tabla 8. Estos valores tienen directa repercusión sobre el consumo energético, dependiendo también de las respectivas tensiones de alimentación.

Tabla 8: Tabla de corrientes demandadas por etapa, y respectivas tensiones de alimentación.

	5 [V]				12.3V [V]
	I <sub>NodeMCU</sub> [mA]	I <sub>μ-Acond.</sub> [mA]	I <sub>L. Switch</sub> [mA]	I <sub>LCD</sub> [mA]	I <sub>Iluminación</sub> [mA]
<b>Reposo</b>	90	25	0	36.45	0
<b>Detección</b>	90	33	3.61	36.34	150.7
				0.68 (sin back light)	

Aplicando la ecuación (8.2) a partir de los datos de la Tabla 8, se obtiene el resultado de la potencia usada por cada etapa, para los estados de reposo y detección. Estos se presentan en la Tabla 9.

Tabla 9: Potencias eléctricas parciales del sistema.

	P <sub>NodeMCU</sub> [W]	P <sub>μ-Acond.</sub> [W]	P <sub>L. Switch</sub> [W]	P <sub>LCD</sub> [W]	P <sub>5V</sub> [W]	P <sub>Iluminación</sub> [W]	P <sub>12V</sub> [W]	P <sub>TOTAL</sub> [W]
<b>Reposo</b>	0.45	0.125	0	0.182	0.757	0	0	0.757
<b>Detección</b>	0.45	0.165	0.018	0.182	0.815	1.854	1.854	2.669

Las columnas correspondientes a  $P_{5V}$  y  $P_{12V}$  corresponden a la suma de las potencias de todos los dispositivos alimentados por 5 [V] y 12 [V], respectivamente. La columna  $P_{TOTAL}$  corresponde a la suma de la potencia de dispositivos de 5 [V] y 12 [V], para los dos casos de reposo y detección. Cabe señalar que, si bien podría pensarse que al sumar estas dos potencias se obtendría el consumo total del sistema, debe considerarse que, en la práctica, el lapso de tiempo de detección es pequeño, del orden de los mili-segundos. Y considerando además que los eventos de apertura y cierre –por parte del posible usuario final– no debieran de hacerse con demasiada frecuencia, es bastante probable que el consumo real del sistema no se vea afectado de forma sustancial por la etapa de iluminación. Por lo tanto, la potencia de principal interés para el consumo es la demandada por los dispositivos de 5 [V], mientras que la potencia correspondiente a 12 [V] es relevante en cuanto a los picos de potencia demandados durante los eventos de detección, parámetro de interés a la hora de seleccionar una fuente adecuada.

Ahora bien, con el propósito de obtener el consumo eléctrico del sistema, se hacen algunas consideraciones acerca de los tiempos, que permiten estudiar el comportamiento bajo casos particulares:

- Considerando que el consumo eléctrico queda determinado por el tiempo de uso, se estudia el consumo del sistema para dos muestras de tiempo definidas: 1 día y 30 días.
- Considerando que un evento de detección dura aproximadamente 300 [ms], y que cada evento se realiza 10 veces por hora, es decir, 240 veces al día (lo cual representa un caso de extrema frecuencia de uso, poco probable pero útil en cuanto se quiera estudiar un “peor caso”), entonces el tiempo total del sistema en estado de detección es de 3 [s] por hora, es decir, 72 [s] por día, y 2.160 [s] por mes.
- Considerando el punto anterior, se tiene que el tiempo del sistema en estado de reposo es de 57 [s] por hora, es decir, 86.328 [s] por día, y 2.589.840 [s] por mes.

Estas consideraciones se resumen en la Tabla 10.

Tabla 10: Tiempos de reposo y detección para varios lapsos de uso definido.

	<b>t<sub>reposo</sub></b>		<b>t<sub>detección</sub> [s]</b>	
	[Segundos]	[Horas]	[Segundos]	[Horas]
<b>Lapso 1 hora</b>	3.597	0.999	3	0.001
<b>Lapso 1 día</b>	86.328	23.98	72	0.02
<b>Lapso 30 días</b>	2.589.840	719.4	2.160	0.6

Entonces, teniendo en cuenta la anterior tabla, es posible obtener el consumo del sistema para un caso de alto uso por parte del posible usuario final. Llámese este caso como “probable”, su consumo queda definido por la siguiente expresión:

$$E_{PROBABLE} = t_{reposo} * P_{reposo} + t_{detección} * P_{detección} \quad (8.3)$$

Aplicando entonces la ecuación (8.3) a partir de los datos de la Tabla 9 de potencias y de la Tabla 10 de tiempos, se obtiene el resultado del consumo demandado por el sistema para los períodos de tiempo de 1 hora, 1 día y 30 días, tal como se observa en la Tabla 11.

Además del caso probable de uso, se presenta también el consumo del sistema en reposo total y en detección total, ya que si bien estos valores no representarían un uso normal del sistema, sí pueden considerarse como valores absolutos que teóricamente podrían alcanzarse bajo un uso inadecuado, como por ejemplo al tener encendido el sistema sin abrir ni una sola vez la gaveta, o por el contrario, al abrirla y cerrarla una y otra vez sin parar.

Tabla 11: Consumos eléctricos totales del sistema.

	<b>Consumo en reposo total</b> <b><math>E_{\text{reposo total}}</math> [Wh]</b>	<b>Consumo en detección total</b> <b><math>E_{\text{detección total}}</math> [Wh]</b>	<b>Consumo caso probable</b> <b><math>E_{\text{probable}}</math> [Wh]</b>
<b>Lapso 1 hora</b>	0.757	2.669	0.759
<b>Lapso 1 día</b>	18.168	64.056	18.206
<b>Lapso 30 días</b>	545.04	1921.68	546.187

Como puede observarse en la tabla, la energía utilizada para los lapsos de tiempo presentados es relativamente baja, si se le compara con la mayoría de otros electrodomésticos de uso común. Para el caso probable, el consumo se encuentra entorno al medio kilo-watt mensual, mientras que en estado de total detección no supera los 2 kilo-watt.

Cabe señalar que el consumo podría disminuirse si se considerara el uso del display LCD sin *backlight*, ya que de esa forma el consumo de este dispositivo se reduce drásticamente, como queda señalado anteriormente en la Tabla 8. Además, el consumo podría reducirse aún más si se utilizaran los modos *sleep* o de bajo consumo que ofrecen los dos microcontroladores utilizados en este proyecto, el ATmega y el ESP8266.



## CONCLUSIONES

Se concluye que los esquemas diseñados para la puesta en marcha del sistema fueron satisfactorios, ya que se cumple la principal funcionalidad planteada en los objetivos: registrar la cantidad de insertos disponibles. El presente estudio validó el principio de funcionamiento basado en LDR que se planteó en las etapas tempranas de diseño, y además se logró con materiales de costo relativamente bajo.

Los parámetros eléctricos arrojados por el sistema no fueron ideales, debido a las limitaciones propias de cualquier implementación en protoboard. No obstante, estos valores se acercaron bastante a lo diseñado, lo cual permitió que el circuito funcionara sin mayores complicaciones.

El consumo eléctrico no fue del todo satisfactorio ya que, si bien es bajo en comparación a otros dispositivos de uso común, podría reducirse mucho más mediante el reemplazo de algunos dispositivos y modos de uso. Por ejemplo, usar el módulo ESP-12 reduciría en gran medida la energía demandada, y lo mismo ocurriría si los microcontroladores trabajasen en modo *standby*, o por ejemplo, sin luz backlight en el caso del display LCD.

El sistema es altamente escalable mediante la sencilla inclusión de más multiplexores y LDR, todo mediante el uso de un simple microcontrolador integrado como el ATmega328P-PU. La cantidad de insertos a detectar puede ser escalada a 256 sin grandes complicaciones, mientras que un número mayor a este es perfectamente factible, mediante algunos ajustes al algoritmo.

Por último, no se descarta el uso de algún otro principio de funcionamiento, como lo es la galga extensiométrica o el sensor inductivo, como se planteó en el Capítulo 4, tecnologías de potencial utilidad en algún desarrollo posterior que pudiese surgir del presente estudio.

## ANEXO A

Código programado en C, en base al algoritmo diseñado, el cual se carga al ATmega328P-PU.

```

/* CONFIGURACIÓN PUERTO SERIE EXTRA */ (Para poder usar pantalla del puerto serie)
#include <SoftwareSerial.h>
SoftwareSerial ESPserial(2,3); //Rx, Tx

/* CONFIGURACIÓN de los LCD */
#include <LiquidCrystal.h> //Librería LCD.
const int COL=16, FIL=2; //Define cantidad de columnas y filas
del display LCD.
const int RW=8, E=9, D4=10, D5=11, D6=12, D7=13; //Define pines del Arduino destinados
al Read/Write y Enable del LCD a. //Define también los pines de datos
D4, D5, D6 y D7.*/
LiquidCrystal LCD(RW, E, D4, D5, D6, D7); //Asigna dichos pines a los correspondientes
del LCD a.

const int Numero_Sensores= 64; //Número de sensores.*/
byte Multiplexor[Numero_Sensores]; //Array de tamaño igual al número de
sensores. //Contendrá los códigos binarios que
controlan a los multiplexores.*/
int Presencias[Numero_Sensores]; //Array de tamaño igual al número de
sensores. //Contendrá el estado binario (presencia
o ausencia) de cada posición.*/
String Cadena_Presencias; //Permite concatenar y guardar los bits de
las presencias/ausencias*/

int Canal; //Variable que representa el canal a
usarse. //Tomará el valor del número de cada uno
de los sensores.*/

const int Lectura=4, Close=5, Enable=6, Lamp=7; //Define al pin 3 como la lectura
del canal seleccionado (presencia o ausencia), //al pin 2 como el switch Abierto/Cerrado
del cajón // (debiera estar conectado a un Final de
carrera), // y al pin 13 como la luz de la lampara
que hará variar la lectura de los LDR.*/

int i;

char Insertos[COL]; //Array del tamaño de las columnas (ancho)
del LCD. //Contendrá la cantidad de insertos
contados, rodeado // por espacios a su derecha e izquierda
(para centrar el número arrojado).*/

String Mensaje; //Guarda el mensaje completo a transmitir*/

unsigned long Tiempo_Previo, Tiempo_Actual;
int Intervalo_LDR= 300; //Milisegundos que los LDRs necesitan
para reaccionar a la luz y variar su resistencia. // Puede que sea necesario aumentar este
intervalo durante las pruebas.*/

```

```

void setup() {

    DDRC= B111111; //Setea todo el canal A (pines 22 al 29)
    como salidas.

    pinMode(Close,INPUT); //Setea a Close (pin 3) como entrada.
    pinMode(Lectura,INPUT); //Setea a Presencia (pin 4) como entrada.
    pinMode(Lamp,OUTPUT); //Setea a Lamp (pin 13) como salida.
    pinMode(Enable, OUTPUT);

    LCD.begin(COL, FIL); //Configura al LCD a con las
    cantidades de columnas y filas definidas anteriormente.
    LCD.clear(); //Limpia el LCD.

    Serial.begin(115200); //Inicializa puerto serie standard
    Serial.setTimeout(50); //Define el tiempo de espera del puerto
    serie. Puede afectar la velocidad de respuesta.
    ESPserial.begin(115200); //Inicializa puerto serie "extra", físico
    para comunicación con ESP.

    digitalWrite(Enable, HIGH); //Desactiva los multiplexores (Enable=1).

    for(i=0 ; i<=Numero_Sensores ; i++){ //Ciclo for que asigna al "array
    Multiplexor" todos los códigos binarios //que controlan que permiten controlar las
        Multiplexor[i]=i; //posiciones del multiplexor.
    } //El conteo es decimal, pero dichos
    números son posteriormente usados en //su formato binario (de la forma
    "Bxxxxxxxx"). //La secuencia va aumentando desde 0 hasta
    el Número de Sensores, asignando //así un número (en decimal y binario) a
    cada una de las posiciones que luego //serán sensadas.
}

void loop(){

    LCD.setCursor(0,0); //Cursor del LCD en 0,0.
    LCD.print("-Puerta Abierta-"); //Imprime "-Puerta Abierta-" en LCD
    0,0.
    while(digitalRead(Close)==1); //STANDBY mientras Close=1 (puerta
    abierta).

    if(digitalRead(Close)==0){ //Si la puerta se cierra (Close=0),
    comienza el sensado.
        /*COMIENZA ETAPA DE CICLOS DE LECTURA-----
        -----*/

        digitalWrite(Lamp, HIGH); //Enciende luz para comenzar lectura
        de cada sensor.
        digitalWrite(Enable, LOW); //Activa los multiplexores
        (Enable=0).

        Tiempo_Previo= millis(); //Muestra 1 de tiempo para provocar
        un retardo.

        LCD.setCursor(0,0); //Cursor del LCD en 0,0.

```

```

LCD.print("Calculando... "); //Imprime "Calculando... " en LCD
0,0.

int Contador=0; //Resetea el contador.

/*COMIENZA EL RETARDO*/
Tiempo_Actual= millis(); //Muestra 2 de tiempo
para provocar retardo.
while( (Tiempo_Actual-Tiempo_Previo) < Intervalo_LDR ){ //Ciclo de retardo.
Compara la diferencia entre las muestras hasta
Tiempo_Actual= millis(); //superar el intervalo
de tiempo que requieren los LDR para excitarse.
if(digitalRead(Close)==1)break; //Condición que cancela
el retardo si la puerta es abierta.
}
/*TERMINA EL RETARDO*/

for(Canal=0; Canal<Numero_Sensores; Canal++){ //Ciclo de lectura para todos los
sensores desde 0
definido.*/ //hasta el número de sensores
PORTC= Multiplexor[Canal]; //Define estados altos y bajos según
el binario de 8 bits que //le corresponde al decimal del
ciclo.*/

if(digitalRead(Lectura)==1){ //Si la puerta continúa cerrada,
Contador++; //entonces suma uno al contador.
Presencias[Canal]=1; //Asigna al array Presencias el estado 1
(presencia de inserto) en posición correspondiente.
}else{
Presencias[Canal]=0; //Asigna al array Presencias el estado 0
(ausencia de inserto) en posición correspondiente.
}

if(digitalRead(Close)==1){ //Si en cambio la puerta se ha
abierto antes que termine el ciclo
entonces se cancela el sensado*/
digitalWrite(Lamp, LOW); //Apaga luz al detectar apertura de
cajón.
digitalWrite(Enable, HIGH); //Desactiva los multiplexores
(Enable=1).
LCD.clear(); //Limpia LCD.
LCD.setCursor(0,0); //Posiciona el cursor del LCD en
0,0.
LCD.print("-Puerta Abierta-"); //Imprime "-Puerta Abierta-".
LCD.setCursor(0,1); //Posiciona el cursor del LCD en
0,1.
LCD.print(" INTERRUMPIDO "); //Imprime " INTERRUMPIDO ".
Canal=Numero_Sensores; //El Canal se lleva al número
máximo, provocando así el termino del ciclo for.
}
}

digitalWrite(Lamp, LOW); //Apaga luz al terminar todas las
lecturas.
digitalWrite(Enable, HIGH); //Desactiva los multiplexores
(Enable=1).
PORTC= B000000; //Terminado el ciclo for, todo el
canal A (pines 22 al 29) se llevan a LOW.

if(digitalRead(Close)==0){ //Llegado a este punto, si la puerta
aun permanece cerrada (Close=0).

/**IMPRIMIR EN LCD INDICADOR**/
LCD.clear(); //Limpia LCD.

```

```

        LCD.setCursor(0,0); //Posiciona el cursor del LCD en
0,0. LCD.print("Numero Insertos:"); //Imprime "Numero Insertos:".

        sprintf(Insertos, " %d ", Contador); //Une la variable Contador al
string llamado Insertos para agregarle espacios.
        LCD.setCursor(0,1); //Posiciona el cursor del LCD en
0,0. LCD.print(Insertos); //Imprime el string Insertos que
contiene al Contador (variable sensada).

/**ENVIAR A ESP POR PUERTO SERIE**/
    for(int i=0 ; i<32 ; i++){ //Ciclo para concatenar los 32 bits
        Cadena_Presencias += Presencias[i]; //Cada bit se concatena a Cadena_Presencias
    }
    Mensaje= Cadena_Presencias+"-"+Contador; //Mensaje lleva el formato, cadena_Presencias-
Contador.
    Serial.println(Mensaje); //Se imprime el mensaje por el puerto serie.
    ESPserial.println(Mensaje); //Se envía el mensaje por el puerto serie.
    Cadena_Presencias= ""; //Se vacía la cadena.

    }

        while(digitalRead(Close)==0){ //STANDBY mientras Close=0 (puerta
cerrada).
        }
        /*TERMINA ETAPA DE CICLOS DE LECTURA-----*/
    }

```

## ANEXO B

Código programado en C, en base al algoritmo diseñado en el Capítulo 7.04, el cual se carga al NodeMCU v1.0.

```

/*-----MAP PIN NODEMCU V1.0-----*/
Pin  =  GPIO
D0   =  16
D1   =  5
D2   =  4
D3   =  0
D4   =  2    (LED de la placa)
D5   =  14
D6   =  12
D7   =  13
D8   =  15
D9   =  3
D10  =  1
-----*/

#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <Arduino.h>           //NECESARIO PA QUE FUNCIONEN LOS
OPERADORES LÓGICOS

//-----VARIABLES GLOBALES-----
int ContConexion =
0;                               //Conteo de conexión al
WIFI para cancelarla en caso de demora
unsigned long previousMillis =
0;                               //FUERA DE USO

int
ChipID;                          //Variabl
e para guardar ChipID, número identificador de la placa
String
Valores;                          //String
donde concatenar todos los valores de las presencias/ausencias
char Num;                          //Variable que recibe cada bit como un
caracter por separado
String Aux;                          //Variable auxiliar de tipo string que
permite hacer conversiones y cadenas
int Presencias[32];                //Array donde se guardan todos los bits de
presencia/ausencia
int Cantidad;                       //Guarda la cantidad de insertos total
recibida

const char *ssid = "wifi-ubb-U";    //Nombre
de la red de internet

```

```

const char *password = "soporte-
dci"; //Password de la red de internet

char host[48]; //
String strhost = "192.168.0.100"; //IP
de la red LAN
String strurl = "/nodemcu/enviardatos.php"; //URL
del PHP

//-----Función para Enviar Datos a la Base de Datos SQL-----
String Enviardatos(String datos) { //Función que
gestiona la conexión con PHP
    String linea = "error";
    WiFiClient client;
    strhost.toCharArray(host, 49);
    if (!client.connect(host, 80)) {
        Serial.println("Fallo de conexion");
        return linea;
    }

    client.print(String("POST ") + strurl + " HTTP/1.1" + "\r\n"
+ //Imprime (envía) al PHP los datos mediante método POST
    "Host: " + strhost + "\r\n"
+ //Se basa en varias cadenas de
caracteres con la información
    "Accept: */*" + "\r\n" +
    "Content-Length: " + datos.length() + "\r\n" +
    "Content-Type: application/x-www-form-urlencoded" + "\r\n"
+
    "\r\n" +
datos); //En esta última
cadena se envían los datos relevantes

    Serial.println(String("POST ") + strurl + " HTTP/1.1" + "\r\n"
+ //Imprime las cadenas del POST en el puerto serie
    "Host: " + strhost + "\r\n" +
    "Accept: */*" + "\r\n" +
    "Content-Length: " + datos.length() + "\r\n" +
    "Content-Type: application/x-www-form-urlencoded" + "\r\n"
+
    "\r\n" + datos);
    Serial.println("");

    delay(10);

    Serial.print("Enviando datos a SQL...");

    unsigned long timeout = millis(); //Referencia de
tiempo para cancelar la conexión al server PHP en caso de demora
    while (client.available() == 0) { //Da vueltas
mientras no haya respuesta por parte del PHP
        if (millis() - timeout > 5000) { //Si pasan más de
5 segundos desde la referencia

```

```

        Serial.println("Cliente fuera de tiempo!");
        client.stop(); //Detiene la
conexión
        return linea; //Salta a linea,
que recibe la respuesta del servidor PHP
    }
}

while(client.available()){ //Mientras haya
respuesta por parte del servidor PHP
    linea = client.readStringUntil('\r'); //Recibe la
respuesta del servidor PHP
}
Serial.println(linea); //Imprime la
respuesta
return linea;
}

//-----SETUP-----
void setup() {

    // Inicia Serial
    Serial.begin(115200);
    Serial.setTimeout(50);
    Serial.print ("-----\n");
    Serial.print ("-----\n");

    ChipID=
ESP.getChipId(); //Obtiene el
Chip ID de la placa
    Serial.printf("Chip ID: %d \n" , ChipID );

    // Conexión WIFI
    WiFi.begin(ssid,
password); //Inicia conexión con la
red wifi
    while (WiFi.status() != WL_CONNECTED and ContConexion <100)
    { //Mientras no se conecte ni ConcConexion llegue a 100, cuenta
++ContConexion; //Cuenta
    delay(500); //Imp
Serial.print("."); //Imp
rime puntos mientras cuenta
    }
    if (ContConexion <100) { //Si
el conteo no superó los 100, procede a conectarse a la LAN
//para usar con ip fija
IPAddress
ip(192,168,0,100); //Configura IP de la
LAN, ES NECESARIO CONFIGURARLA según la red

```



```

    IPAddress
gateway(192,168,1,1); //Congigura Puerta de
enlace, no se toca (?)
    IPAddress
subnet(255,255,255,0); //Configura Máscara
de subred, no se toca (?)
    WiFi.config(ip, gateway, subnet); //Se
conecta a la red LAB con las IP configuradas

    Serial.println("");
    Serial.println("");
    Serial.println("WiFi conectado");
    Serial.println(WiFi.localIP()); //Imp
rime la IP por el pueto serie
}
else {
    Serial.println("");
    Serial.println("Error de conexión a red WIFI"); //Si
el conteo supera los 100, imprime Error
}
    Serial.print("-----\n");
}

//-----LOOP-----
-----
void loop() {

    if( Serial.available()>0 ){
        Serial.println("-----");
        Serial.println("");

        for(int i=0 ; i<32 ; i++){ //Ciclo para el dato de
las 32 posiciones
            Num= Serial.read(); //Se lee el puerto serie
como un char
            Aux= Num; //Se transforma el char a
string
            Presencias[i]= Aux.toInt(); //Se transforma el string
a int, en la posición adecuada
        }
        Serial.println("Los bit de la presencia en cada posición son:");
        for(int i=0; i<32 ; i++){
            Serial.print(Presencias[i]); //Se imprime el dato de
cada posición uno tras otro
        }
        Serial.println("");

        Num= Serial.read(); //Se lee una vez extra
para ignorar el guión "-"
        Num= Serial.read(); //Se lee el primer dígito
de la cantidad
        Aux= Num; //Se transforma el dígito
a string

```

```

    Num= Serial.read(); //Se lee el segundo
    dígito de la cantidad
    Aux+= Num; //Se adiciona el segundo
    dígito al primero
    Cantidad= Aux.toInt(); //Se transforma el string
    cantidad a int
    Serial.printf("Cantidad de total de insertos: %d\n",
Cantidad); //Imprime cantidad
    Serial.println("");
    Num= Serial.read(); //Se lee dos veces para
    vaciar el buffer
    Num= Serial.read(); //que aun contiene los
    caracteres '\n' y '\r'
    Aux= ""; //Se reinicia el string
    auxiliar

    for ( int i=0 ; i<32 ; i++ ){ //Ciclo para
concatenar los valores de las posiciones en formato POST
    String Index= 'A' + String(i+1) ; //Concatena de
la forma A1, A2, A3... A32
    String Par= Index + '=' + Presencias[i]; //Concatena de
la forma A1=X, A2=X, A3=X... A32=X

    Valores+= Par; //Adiciona la cadena Par a la
cadena Valores, de la forma A1=X&A2=X&A3=X... según método POST

    if(i==31){ //Si es el
último ciclo, no agrega el "&"
        break;
    }
    Valores+= '&'; //Si no va en el último ciclo,
adiciona "&" a la cadena Valores, para mantener el formato del POST
}

String
ValoresPOST= "ChipID="+String(ChipID) //Concatena
todos los valores relevantes a enviar mediante el POST
        +"&Cantidad="+String(Cantidad) //Inc
luye el Chip ID, la cantidad de insertos
        +"&"+Valores ; //y
también la presencia/ausencia en todas las 32 posiciones
    Enviardatos(ValoresPOST); //Salto a la función Enviardatos,
lleva como parámetro la cadena de caracteres con todos los valores
    Valores.remove(0); //Vacía la cadena de caracteres,
para permitir espacio a los valores del siguiente sensado

}
}

```

## ANEXO C

Código en PHP para comunicarse con el NodeMCU y recibir los datos mediante el método POST. Gestiona el ingreso de los datos a la tabla de phpMyAdmin. Actualiza las filas.

```

<?php
$servername = "localhost";
$username = "root";
$password = "root"; //En esta parte puse una contraseña que tuve que definir al instalar
$dbname = "nodemcu";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$ChipID= $_POST ['ChipID'];
$Cantidad= $_POST ['Cantidad'];
$A1= $_POST ['A1'];
$A2= $_POST ['A2'];
$A3= $_POST ['A3'];
$A4= $_POST ['A4'];
$A5= $_POST ['A5'];
$A6= $_POST ['A6'];
$A7= $_POST ['A7'];
$A8= $_POST ['A8'];
$A9= $_POST ['A9'];
$A10= $_POST ['A10'];
$A11= $_POST ['A11'];
$A12= $_POST ['A12'];
$A13= $_POST ['A13'];
$A14= $_POST ['A14'];
$A15= $_POST ['A15'];
$A16= $_POST ['A16'];
$A17= $_POST ['A17'];
$A18= $_POST ['A18'];
$A19= $_POST ['A19'];
$A20= $_POST ['A20'];
$A21= $_POST ['A21'];
$A22= $_POST ['A22'];
$A23= $_POST ['A23'];

```

```

$A24= $_POST ['A24'];
$A25= $_POST ['A25'];
$A26= $_POST ['A26'];
$A27= $_POST ['A27'];
$A28= $_POST ['A28'];
$A29= $_POST ['A29'];
$A30= $_POST ['A30'];
$A31= $_POST ['A31'];
$A32= $_POST ['A32'];

```

```

$sql= "UPDATE tabla
      SET Valor= CASE
      WHEN id=1 THEN $A1
      WHEN id=2 THEN $A2
      WHEN id=3 THEN $A3
      WHEN id=4 THEN $A4
      WHEN id=5 THEN $A5
      WHEN id=6 THEN $A6
      WHEN id=7 THEN $A7
      WHEN id=8 THEN $A8
      WHEN id=9 THEN $A9
      WHEN id=10 THEN $A10
      WHEN id=11 THEN $A11
      WHEN id=12 THEN $A12
      WHEN id=13 THEN $A13
      WHEN id=14 THEN $A14
      WHEN id=15 THEN $A15
      WHEN id=16 THEN $A16
      WHEN id=17 THEN $A17
      WHEN id=18 THEN $A18
      WHEN id=19 THEN $A19
      WHEN id=20 THEN $A20
      WHEN id=21 THEN $A21
      WHEN id=22 THEN $A22
      WHEN id=23 THEN $A23
      WHEN id=24 THEN $A24
      WHEN id=25 THEN $A25
      WHEN id=26 THEN $A26
      WHEN id=27 THEN $A27
      WHEN id=28 THEN $A28
      WHEN id=29 THEN $A29
      WHEN id=30 THEN $A30
      WHEN id=31 THEN $A31
      WHEN id=32 THEN $A32
      WHEN id=33 THEN $Cantidad
      ELSE 9999

```

```
END";
```

```
if ($conn->query($sql) === TRUE) {  
    echo "New record created successfully";  
} else {  
    echo "Error: " . $sql . "<br>" . $conn->error;  
}
```

```
$conn->close();  
?>
```

## ANEXO D

id	ChipID	fecha	Valor
1	8454785	2018-10-11 21:27:51	0
2	8454785	2018-10-02 17:30:05	0
3	8454785	2018-10-11 21:27:51	0
4	8454785	2018-10-02 17:29:56	0
5	8454785	2018-09-30 05:13:20	0
6	8454785	2018-09-30 05:13:14	0
7	8454785	2018-09-30 05:13:08	0
8	8454785	2018-09-30 05:13:02	0
9	8454785	2018-10-11 21:13:08	0
10	8454785	2018-09-30 05:05:00	0
11	8454785	2018-10-02 17:29:46	0
12	8454785	2018-09-30 05:13:26	0
13	8454785	2018-09-30 05:13:20	0
14	8454785	2018-09-30 05:13:14	0
15	8454785	2018-09-30 05:13:08	0
16	8454785	2018-09-30 05:13:02	0
17	8454785	2018-10-11 21:13:08	0
18	8454785	2018-09-30 05:05:00	0
19	8454785	2018-10-02 17:29:46	0
20	8454785	2018-09-30 05:13:26	0
21	8454785	2018-09-30 05:13:20	0
22	8454785	2018-09-30 05:13:14	0
23	8454785	2018-09-30 05:13:08	0
24	8454785	2018-09-30 05:13:02	0
25	8454785	2018-10-11 21:13:08	0
26	8454785	2018-09-30 05:05:06	0
27	8454785	2018-09-30 05:05:00	0
28	8454785	2018-10-02 17:29:46	0
29	8454785	2018-09-30 05:13:26	0
30	8454785	2018-09-30 05:13:20	0

id	ChipID	fecha	Valor
31	8454785	2018-10-11 21:18:53	0
32	8454785	2018-10-11 21:17:15	0
33	8454785	2018-10-11 21:27:51	0

Fig. D.1: Tabla prueba base de datos phpMyAdmin.  
Todas las entradas en 0.

id	ChipID	fecha	Valor
1	8454785	2018-10-11 21:29:03	1
2	8454785	2018-10-11 21:36:53	1
3	8454785	2018-10-11 21:37:50	1
4	8454785	2018-10-11 21:37:01	1
5	8454785	2018-10-11 21:29:03	1
6	8454785	2018-10-11 21:36:53	1
7	8454785	2018-10-11 21:29:03	1
8	8454785	2018-10-11 21:37:50	1
9	8454785	2018-10-11 21:37:50	1
10	8454785	2018-10-11 21:36:14	1
11	8454785	2018-10-11 21:37:50	1
12	8454785	2018-10-11 21:37:50	1
13	8454785	2018-10-11 21:37:50	1
14	8454785	2018-10-11 21:36:53	1
15	8454785	2018-10-11 21:33:41	1
16	8454785	2018-10-11 21:37:50	1
17	8454785	2018-10-11 21:32:31	1
18	8454785	2018-10-11 21:36:14	1
19	8454785	2018-10-11 21:37:50	1
20	8454785	2018-10-11 21:36:53	1
21	8454785	2018-10-11 21:37:50	1
22	8454785	2018-10-11 21:37:50	1
23	8454785	2018-10-11 21:30:56	1
24	8454785	2018-10-11 21:36:14	1
25	8454785	2018-10-11 21:37:50	1
26	8454785	2018-10-11 21:37:01	1
27	8454785	2018-10-11 21:30:56	1
28	8454785	2018-10-11 21:36:53	1
29	8454785	2018-10-11 21:30:56	1
30	8454785	2018-10-11 21:37:50	1

id	ChipID	fecha	Valor
31	8454785	2018-10-11 21:37:13	1
32	8454785	2018-10-11 21:37:07	1
33	8454785	2018-10-11 21:37:50	32

Fig. D.2: Tabla prueba base de datos phpMyAdmin.  
Todas las entradas en 1.

id	ChipID	fecha	Valor
1	8454785	2018-10-11 21:29:03	1
2	8454785	2018-10-11 21:33:41	0
3	8454785	2018-10-11 21:29:03	1
4	8454785	2018-10-11 21:33:41	0
5	8454785	2018-10-11 21:29:03	1
6	8454785	2018-10-11 21:33:41	0
7	8454785	2018-10-11 21:29:03	1
8	8454785	2018-10-11 21:33:41	0
9	8454785	2018-10-11 21:33:41	1
10	8454785	2018-10-11 21:33:41	0
11	8454785	2018-10-11 21:33:41	1
12	8454785	2018-10-11 21:33:41	0
13	8454785	2018-10-11 21:33:41	1
14	8454785	2018-10-11 21:33:41	0
15	8454785	2018-10-11 21:33:41	1
16	8454785	2018-10-11 21:32:31	0
17	8454785	2018-10-11 21:32:31	1
18	8454785	2018-09-30 05:05:00	0
19	8454785	2018-10-11 21:30:56	1
20	8454785	2018-09-30 05:13:26	0
21	8454785	2018-10-11 21:30:56	1
22	8454785	2018-09-30 05:13:14	0
23	8454785	2018-10-11 21:30:56	1
24	8454785	2018-09-30 05:13:02	0
25	8454785	2018-10-11 21:30:56	1
26	8454785	2018-09-30 05:05:06	0
27	8454785	2018-10-11 21:30:56	1
28	8454785	2018-10-02 17:29:46	0
29	8454785	2018-10-11 21:30:56	1
30	8454785	2018-09-30 05:13:20	0
id	ChipID	fecha	Valor
31	8454785	2018-10-11 21:30:56	1
32	8454785	2018-10-11 21:17:15	0
33	8454785	2018-10-11 21:33:41	16

Fig. D.3: Tabla prueba base de datos phpMyAdmin. Presencias y ausencias intercaladas, 16 presencias en total.

id	ChipID	fecha	Valor
1	8454785	2018-10-11 21:29:03	1
2	8454785	2018-10-11 21:36:53	1
3	8454785	2018-10-11 21:36:14	0
4	8454785	2018-10-11 21:37:01	1
5	8454785	2018-10-11 21:29:03	1
6	8454785	2018-10-11 21:36:53	1
7	8454785	2018-10-11 21:29:03	1
8	8454785	2018-10-11 21:33:41	0
9	8454785	2018-10-11 21:37:01	0
10	8454785	2018-10-11 21:36:14	1
11	8454785	2018-10-11 21:36:14	0
12	8454785	2018-10-11 21:33:41	0
13	8454785	2018-10-11 21:36:14	0
14	8454785	2018-10-11 21:36:53	1
15	8454785	2018-10-11 21:33:41	1
16	8454785	2018-10-11 21:32:31	0
17	8454785	2018-10-11 21:32:31	1
18	8454785	2018-10-11 21:36:14	1
19	8454785	2018-10-11 21:36:14	0
20	8454785	2018-10-11 21:36:53	1
21	8454785	2018-10-11 21:36:14	0
22	8454785	2018-09-30 05:13:14	0
23	8454785	2018-10-11 21:30:56	1
24	8454785	2018-10-11 21:36:14	1
25	8454785	2018-10-11 21:36:14	0
26	8454785	2018-10-11 21:37:01	1
27	8454785	2018-10-11 21:30:56	1
28	8454785	2018-10-11 21:36:53	1
29	8454785	2018-10-11 21:30:56	1
30	8454785	2018-09-30 05:13:20	0
id	ChipID	fecha	Valor
31	8454785	2018-10-11 21:37:13	1
32	8454785	2018-10-11 21:37:07	1
33	8454785	2018-10-11 21:37:13	20

Fig. D.4: Tabla prueba base de datos phpMyAdmin. Presencias y ausencias distribuidas aleatoriamente.