



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Sistema de monitorización y control de inventario de fusiles HK
en el pañol del condestable de la ENM.*

Grado en Ingeniería Mecánica

ALUMNO: Kathong Panjaphon.

DIRECTORES: Norberto Fernández García.

CURSO ACADÉMICO: 2017-2018

Universida_{de}Vigo



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Sistema de monitorización y control de inventario de fusiles HK
en el pañol del condestable de la ENM.*

Grado en Ingeniería Mecánica
Intensificación en Tecnología Naval
Cuerpo General

UniversidadeVigo

RESUMEN

El objetivo de este Trabajo de Fin de Grado consiste en montar físicamente un sistema de monitorización y control de inventario remotos de un pañol. Este sistema ofrece una serie de funcionalidades, tales como: monitorización de los parámetros ambientales (humedad, temperatura, iluminación) en el pañol, detección de movimiento de personas en su interior, y control del inventario de fusiles HK. La información recogida en el pañol se transmite a un puesto de control remoto, presentándola en una interfaz de usuario tipo tablero de mandos (dashboard) para hacer sencilla su interpretación.

Para la implementación del sistema propuesto se ha utilizado el ordenador de placa reducida Arduino, para la captura y procesamiento de datos de sensores, y dispositivos LoRa para la comunicación remota. Toda la información adquirida por el sistema que ha estado dentro del pañol de munición se ha transmitido a un PC, en el que una aplicación implementada en C# actúa de tablero de mandos para permitir la monitorización y el control remoto del estado del pañol.

Para comprobar la viabilidad del sistema desarrollado se han llevado a cabo pruebas dentro del pañol del condestable de la Escuela Naval Militar, con resultados satisfactorios.

PALABRAS CLAVE

Arduino, LoRa, LoRaWAN, Pañol, Control de inventario, Monitorización remota

AGRADECIMIENTOS

Llegando ya al punto final de mi formación en la ENM, quiero agradecerle a la ENM y al CUD, la formación recibida durante estos cinco cursos. En ocasiones, incluso, más allá de lo académico.

Agradezco también al tutor del trabajo, Don Norberto Fernández García, por su gran implicación y ayuda, así como las pautas marcadas y las dudas resueltas.

Por supuesto, a mi familia y amigos que, durante cinco años, han sabido apoyarme cuando lo necesitaba y ayudarme en todo lo que me ha hecho falta.

CONTENIDO

Contenido	1
Índice de Figuras	3
Índice de Tablas.....	5
1 Introducción y objetivos	7
1.1 Definiciones básicas.....	7
1.1.1 Polvorín.....	7
1.1.2 Pañol de Munición.....	7
1.2 Tipos de pañol de munición	7
1.2.1 Pañoles Principales.	7
1.2.2 Pañoles de Urgencia.....	8
1.2.3 Cajas de Urgencia.	8
1.3 Normativa y precauciones de seguridad aplicable a la gestión de pañoles.....	8
1.3.1 Precauciones básicas de seguridad	8
1.3.2 Temperatura de pañoles de munición	9
1.3.3 Humedad en pañoles de munición	11
1.3.4 Alumbrado de pañoles de munición	12
1.3.5 Trabajos en pañoles de munición.....	12
1.4 Objetivos	12
1.5 Estructura de la memoria	12
2 Estado del arte	15
2.1 Programa para control de inventario	15
2.2 Internet de las cosas	15
2.3 LoRa.....	16
2.3.1 Canales y rangos de frecuencias	17
2.4 LoRaWAN	17
2.4.1 Red de LoRaWAN.....	18
2.4.2 Canales y velocidades de transmisión	19
2.4.3 Clasificaciones de nodos LoRaWAN.	19
2.4.4 Seguridad de LoRaWAN	20
2.5 Herramientas de proyecto	21
2.5.1 LoRa Shield para Arduino	21
2.5.2 Sensor de presencia PIR	22
2.5.3 Sensor de temperatura y humedad DHT 22	22

2.5.4 Sensor HC-SR04.....	23
2.5.5 Sensor de huella dactilar	25
2.5.6 Sensor LDR.....	26
2.5.7 Arduino	27
2.5.8 IDE Arduino	30
2.5.9 Microsoft Visual Studio.....	32
3 Desarrollo del TFG.....	35
3.1 Arquitectura del sistema.....	35
3.1.1 Método de paso de mensajes por puerto serie	35
3.1.2 General.....	36
3.2 Funcionamiento del sistema de monitorización	37
3.2.1 Observador.....	37
3.2.2 Servidor.....	43
3.3 Interfaz Gráfica de Usuario.....	44
3.4 Ventana principal del programa	45
3.4.1 Inicial	45
3.4.2 Tabla de situación	48
3.4.3 Tabla de Temperatura/Humedad (T/H)	49
3.4.4 Tabla de accesos al pañol.....	49
4 Prueba y Validación	51
4.1 Distancia máxima del sensor HC-SR04.....	51
4.2 Prueba del sensor de huella dactilares.....	52
4.3 Prueba de alcance de LoRa	53
4.4 Prueba del funcionamiento del sistema.....	54
5 Conclusiones y líneas futuras	55
5.1 Conclusiones	55
5.2 Líneas Futuras	55
6 Bibliografía.....	59

ÍNDICE DE FIGURAS

Figura 1-1 Misil de RIM-66 Standard MR (Fuente: www.upi.com)	10
Figura 1-2 Misil Harpoon block II (Fuente: www.defenseindustrydaily.com).....	10
Figura 1-3 Misil SideWinder (Fuente: www.nspa.nato.int)	10
Figura 1-4 Torpedo MK-46 (Fuente: www.navysite.de).....	11
Figura 2-1 El número de los dispositivos conectados a Internet (Fuente: www.cisco.com).....	16
Figura 2-2 LoRa (Fuente: www.7layers.com).....	16
Figura 2-3 LoRaWAN (Fuente: www.intelilight.eu)	18
Figura 2-4 Esquema modo híbrido de LoRaWAN (Fuente: www.digikey.es)	18
Figura 2-5 Esquema de clases de nodos LoRaWAN (Fuente: www.digikey.es)	19
Figura 2-6 Cifrado de clave simétrica (Fuente: Apuntes de la asignatura Ampliación de Informática).....	20
Figura 2-7 Shield de Dragino (Fuente: www.dragino.com).....	21
Figura 2-8 Sensor PIR (Fuente: elaboración propia).....	22
Figura 2-9 Sensor DHT 22 (Fuente: elaboración propia).....	22
Figura 2-10 Esquema conexión DHT 22 (Fuente: elaboración propia)	23
Figura 2-11 Sensor HC-SR04 (Fuente: elaboración propia)	23
Figura 2-12 Funcionamiento de sensor HC – SR04 (Fuente: www.zonamaker.com)	24
Figura 2-13 Sensor de huellas dactilares(Fuente: elaboración propia)	25
Figura 2-14 Sensor LDR (Fuente: elaboración propia).....	26
Figura 2-15 Arduino (Fuente: www.arduino.cc).....	27
Figura 2-16 El equipo de Arduino (Fuente: blog.bricogeek.com)	28
Figura 2-17 Arduino Uno (Fuente: elaboración propia).....	28
Figura 2-18 IDE Arduino (Fuente: elaboración propia).....	30
Figura 2-19 El menú archivo (Fuente: elaboración propia)	31
Figura 2-20 El menú sketch (Fuente: elaboración propia)	31
Figura 2-21 El menú herramientas (Fuente: elaboración propia).....	32
Figura 2-22 Visual Studio (Fuente: elaboración propia).....	32
Figura 2-23 El menú archivo de Visual Studio (Fuente: elaboración propia).....	33
Figura 2-24 Las herramienta de Visual Studio (Fuente: [16])	33
Figura 2-25 El editor de código (Fuente: elaboración propia)	34
Figura 3-1 Arquitectura del sistema (Fuente: elaboración propio)	35
Figura 3-2 Funcionamiento del método de paso de mensaje por puerto serie (Fuente: elaboración propia).....	36
Figura 3-3 Método de relleno de byte (Fuente: Apuntes de la signatura Ampliación de Informática)	36

Figura 3-4 Diagrama de flujo del servidor (Fuente: elaboración propia).....	36
Figura 3-5 Estructura general de observador (Fuente: elaboración propia)	37
Figura 3-6 Estructura de Arduino con sensores de contacto (Fuente: elaboración propia).....	38
Figura 3-7 Monitor serie mostrando los datos leídos de los sensores de contacto (Fuente: elaboración propia)	39
Figura 3-8 Estructura de Arduino con sensores ambientales (Fuente: elaboración propia).....	39
Figura 3-9 Monitor serie mostrando los datos leídos de los sensores ambientales (Fuente: elaboración propia)	41
Figura 3-10 Monitor serie mostrando los datos leídos en el Arduino central (Fuente: elaboración propia)	42
Figura 3-11 Estructura de servidor (Fuente: elaboración propia)	43
Figura 3-12 El monitor serie para visualizar el envío de datos al PC (Fuente: elaboración propia).....	44
Figura 3-13 Arquitectura de la aplicación (Fuente: elaboración propia)	44
Figura 3-14 Ventana del programa general (Fuente: elaboración propia)	45
Figura 3-15 Ventana del programa inicial (Fuente: elaboración propia)	45
Figura 3-16 Ventana de estado armero (Fuente: elaboración propia)	46
Figura 3-17 Ventana de temperatura (Fuente: elaboración propia).....	46
Figura 3-18 Ventana de humedad (Fuente: elaboración propia)	47
Figura 3-19 Ventana de situación de luz, puerta y movimiento (Fuente: elaboración propia)	47
Figura 3-20 Ventana de mensaje alerta (Fuente: elaboración propia).....	48
Figura 3-21 Ventana de tabla de situación (Fuente: elaboración propia).....	48
Figura 3-22 Ventana de tabla de Temperatura/Humedad (Fuente: elaboración propia)	49
Figura 3-23 Ventana de tabla de persona entrada (Fuente: elaboración propia).....	49
Figura 4-1 Grafica de la prueba de distancia del sensor HC-SR04 (Fuente: elaboración propia) ...	52
Figura 4-2 Prueba de sensor de huella dactilar (Fuente: elaboración propia)	52
Figura 4-3 Monitor serie en la prueba de sensor de huella dactilar (Fuente: elaboración propia) ...	53
Figura 4-4 Grafica de la prueba de sensor de la huella dactilar (Fuente: elaboración propia).....	53
Figura 4-5 Prueba de alcance de LoRa (Fuente: elaboración propia)	54

ÍNDICE DE TABLAS

Tabla 2-1 Las especificaciones LoRa para Europa (Fuente: www.lora-alliance.org).....	17
Tabla 2-2 Características de LoRa Shield (Fuente: www.dragino.com).....	21
Tabla 2-3 Características del sensor de huella dactilar (Fuente: www.instructables.com)	25
Tabla 2-4 Características de Arduino (Fuente: www.arduino.cc)	29

1 INTRODUCCIÓN Y OBJETIVOS

Dado que el objetivo principal de este trabajo es el desarrollo de un sistema de monitorización y control de inventario de fusiles HK en el pañol del condestable de la ENM, comenzaremos en este capítulo definiendo qué es un pañol y su tipología, y presentando normativa básica aplicable a la seguridad de pañoles, normativa que resulta relevante en el desarrollo de los aspectos de monitorización del presente trabajo.

1.1 Definiciones básicas

El Manual Interejércitos de Almacenamiento de Municiones y Explosivos para las Fuerzas Armadas Españolas [1], lo define como almacén de municiones o explosivos que reúne las condiciones de seguridad especificadas en dicho Manual. No obstante, la Armada tradicionalmente utiliza los términos polvorín y pañol de munición.

1.1.1 Polvorín

Equivalente a Depósito de Municiones, situado en tierra y separado de las instalaciones de la UCOs (Unidades Centros u Organismos), que cumple las condiciones de seguridad aprobadas.

1.1.2 Pañol de Munición

Compartimento o recinto ubicado dentro de las instalaciones de las UCOs, que cumple las normas de seguridad establecidas y que emplean las UCOs para almacenar explosivos o municiones de cualquier clase.

1.2 Tipos de pañol de munición

1.2.1 Pañoles Principales.

Son los compartimentos dedicados permanentemente al almacenamiento de la munición del cargo. A bordo de los buques se sitúan generalmente bajo cubierta y si es posible, bajo la línea de flotación.

1.2.2 Pañoles de Urgencia.

Se denominan así a los pañoles que, situados en las proximidades del arma que ha de emplear la munición, contienen una cantidad limitada para poder utilizarla en situaciones de emergencia con el fin de facilitar su rápido suministro al comienzo de un combate. Estos pañoles se emplearán solamente cuando se prevea utilizar las armas próximas a los mismos. La munición almacenada en estos pañoles y la que se encuentre en los tambores de algunas armas automáticas, está expuesta a un deterioro más rápido que en los pañoles principales; por ello se reconocerá antes de devolverla a estos últimos, donde quedará apartada y se empleará en la primera oportunidad.

1.2.3 Cajas de Urgencia.

Son los contenedores de munición que por su escasa entidad o poca capacidad no se consideran pañoles de urgencia. Permanecerán vacías, salvo cuando se prevea un tiro inmediato; pero mientras almacenen munición estarán dotadas con un termómetro de máxima y mínima, cuya lectura se efectuará diariamente y se anotará en un libro registro.

1.3 Normativa y precauciones de seguridad aplicable a la gestión de pañoles

La seguridad se consigue cuando la información, material, personal, actividades y UCOs que se especifiquen están protegidas contra el espionaje, subversión, actos vandálicos (que afectan a la imagen pública de la Armada) y terrorismo, así como contra pérdidas o difusiones no autorizadas. En general, se distinguen tres ámbitos de actuación: seguridad física, seguridad interna y seguridad de la información. Para los propósitos del presente trabajo, la más relevante es la seguridad física, que se podría definir como [2]:

"Es la parte de la seguridad relativa a la organización, medios y procedimientos diseñados para proteger al personal, unidades, instalaciones, material e información, para impedir el acceso no autorizado a equipos y documentos, y para protegerlos contra el espionaje, sabotaje, daños y robos"

En el caso de los pañoles, además de proteger contra las amenazas obvias (sabotaje, robos, etc.) un aspecto que es necesario tener en cuenta es el de mantener unas condiciones de conservación adecuadas, que permitan que la munición se encuentre en un estado óptimo para su uso. Las próximas secciones tratarán estos aspectos con mayor detalle. La información que se incluye en las mismas se ha obtenido de [1].

1.3.1 Precauciones básicas de seguridad

Además de las precauciones de seguridad generales, el pañol de munición en los buques, según el reglamento de pólvoras, explosivos y municiones para buque, fuerzas y dependencias deberán cumplir las siguientes condiciones:

- Está prohibido fumar en los pañoles y sus proximidades. El personal que trabaja con la munición no debe llevar mecheros, cerillas ni otros elementos susceptibles de producir chispas o fuego.
- En los buques, los pañoles de municiones deben estar situados de forma que tengan la mayor protección posible y un acceso fácil a las armas que han de alimentar. Deberán mantenerse perfectamente limpios y exentos de materiales que puedan provocar inflamaciones espontáneas, como trapos impregnados de grasa o aceite, materiales combustibles, pinturas, aguarrás y efectos de conservación y limpieza.
- Los compartimentos que rodean a los pañoles deben ser poco frecuentados y no tener focos caloríficos de importancia, ni contener materiales combustibles.

- Se evitará el empleo de materiales combustibles en la habilitación de estos pañoles, en lo referente a forros, pinturas, chilleras y jarras.
- Se evitará el paso de tuberías por los pañoles de municiones, salvo que sean necesarias para el propio pañol o que no sea posible proporcionar otro recorrido a las tuberías.
- Los pañoles de pólvora sin humo, los de artificios, los de pirotecnia y los de bombas, cercanos a focos caloríficos, tendrán aislamiento ignífugo y térmico en mamparos y cubiertas.
- Todos los pañoles serán estancos, permanecerán secos y estarán dotados con ventilación o acondicionadores de aire, y sistemas de inundación y refresco. Los mandos correspondientes a estos servicios, así como los de apertura y cierre de la ventilación, se controlarán a distancia, de tal manera que no sea necesario entrar en el pañol para accionarlos.
- Para evitar condensaciones en los pañoles dotados con sistemas de aire acondicionado, antes de proceder a un embarque o desembarque de munición, se deberá adecuar la temperatura gradualmente, hasta alcanzar la del exterior del pañol.
- Dentro del pañol, y en todas las zonas de municionamiento, la cubierta tendrá un recubrimiento antideslizante y antichispas, que cumpla la normativa vigente.

1.3.2 Temperatura de pañoles de munición

Como norma general para preservar la vida probable de las municiones y sus componentes, la temperatura deberá mantenerse comprendida entre 10 °C y 35 °C, a no ser que las prescripciones técnicas de la propia munición especifiquen condiciones. En todo caso, se evitarán los cambios bruscos de temperatura. La prescripción técnica de las diferentes municiones se dividiría en los siguientes tipos de pólvoras:

1.3.2.1 Pañoles que contengan pólvoras sin humo

Es muy importante que estos pañoles estén especialmente preparados para mantener la temperatura interior lo más próxima posible a 15 °C. Para conseguirlo además de ventilación normal tendrán las instalaciones necesarias para una ventilación forzada y, en caso de que fuese necesario, con acondicionador de aire en circuito cerrado. La temperatura máxima admisible en pañoles de pólvora sin humo, en circunstancias normales, es de 35 °C. Rebasar la temperatura de 35 °C no implica peligro, sino solamente tomar precauciones. A título informativo se hace notar que una pólvora sin humo nueva puede resistir más de dos años a 65.5 °C sin que aparezcan vapores de nitrógeno ni muestre señales peligrosas de descomposición

1.3.2.2 Pañoles de misiles y cohetes

Los pañoles de misiles, cohetes y motores-cohete, deberán estar acondicionados para mantener la temperatura indicada en sus respectivos manuales procurando no sobrepasar los 35° C, ni que las variaciones que se produzcan sean mayores de 10° C en una hora, ni de 20° C en un día. En el pañol deberán estar equipados con los sistemas de detección de peligro (humo, llama, etc.), adecuados para mantener una vigilancia continua. El sistema de ventilación se empleará en caso de que se inicie accidentalmente un motor, vigilando que la presión en el pañol no alcance niveles peligrosos. Los sistemas de lucha contra incendios de estos pañoles serán de dióxido de carbono, espuma, agua a presión o una combinación de ellos.

A modo de referencia, algunos de los límites de temperatura son:

- Misil Standard (Figura 1-1). MIN: -17 °C (-42 °C en contenedor). MAX: 60 °C.



Figura 1-1 Misil de RIM-66 Standard MR (Fuente: www.upi.com)

- Misil Harpoon (Figura 1-2). MIN: -18 °C. MAX: 49 °C. También: -40 °C para un máximo de 72 horas y hasta 71 °C para un máximo de 2 horas.



Figura 1-2 Misil Harpoon block II (Fuente: www.defenseindustrydaily.com)

- Missile Sidewinder (Figura 1-3). MIN: -54 °C. MAX: 71 °C.



Figura 1-3 Misil SideWinder (Fuente: www.nspa.nato.int)

- Torpedo MK-46 (Figura 1-4). MIN: -40 °C (0 °C en contenedor). MAX: 49 °C (54 °C en contenedor).



Figura 1-4 Torpedo MK-46 (Fuente: www.navysite.de)

1.3.2.3 Pañoles de artificios electro-pirotécnicos

Deberán estar acondicionados para mantener la temperatura por debajo de los 40 °C, aunque existen artificios que pueden soportar hasta los 70 °C, independientemente de que las temperaturas de los componentes electrónicos requieran una temperatura de almacenamiento entre -20 °C y + 65 °C.

1.3.2.4 Pañoles con munición de fósforo blanco.

Si el pañol contiene únicamente proyectiles o cabezas de combate cargadas con fósforo blanco, no existen inconvenientes siempre que la temperatura esté por debajo de 35° C. No deberá superar los 40° C, porque el punto de fusión del fósforo blanco es de 43° C. En caso de que el fósforo blanco forme parte como componente de un disparo completo, se empleará la temperatura más restrictiva.

1.3.3 Humedad en pañoles de munición

La humedad relativa influye grandemente en la conservación de las municiones y sus componentes por eso los pañoles deben mantenerse relativamente secos y deben de tener una humedad relativa que no supere el 70%.

1.3.3.1 Pañoles de misiles y cohetes.

Los contenedores de los misiles y cohetes deben ser estancos; la humedad relativa en su interior no debe superar el 50%. Generalmente tienen un indicador de humedad, de manera que cuando el área del indicador del 50% cambie de tonalidad, debe renovarse el desecante.

1.3.3.2 Pañoles de artificios electro-pirotécnicos.

La humedad deteriora los elementos eléctricos y electrónicos de las espoletas de proximidad, así como los artificios pirotécnicos, las luces de fosforo de calcio para guindolas y luces Holmes, que incluso pueden activarse con la humedad. Por ello, estos pañoles deben permanecer secos. Los envases que contienen artificios y pirotecnia deben ser estancos y estarán libres de corrosión y óxido. En el interior de un pañol nunca se abrirá una caja o envase que contenga material pirotécnico, y nunca se

devolverán al pañol aquellos artificios susceptibles de activarse con la humedad después de que hayan sido abiertos sus envases.

1.3.3.3 Pañoles de proyectiles y munición engarzada.

Es imprescindible que las jarras de munición engarzada sean estancas para preservar su contenido de la humedad. Adquiere todavía mayor importancia este aspecto cuando se trata de munición cuyos proyectiles llevan montadas espoletas de proximidad.

1.3.4 Alumbrado de pañoles de munición

Estarán dotados de alumbrado eléctrico de la red general del barco con protecciones adecuadas que cumplan la normativa de seguridad. Los interruptores correspondientes estarán fuera del pañol. Como alumbrado de emergencia todos los pañoles tendrán lámparas de pilas o acumuladores estancos.

1.3.5 Trabajos en pañoles de munición

Aparte de los correspondientes a las limpiezas, no se permitirán más trabajos que los necesarios para el trasiego, vigilancia, cambio de desecantes e identificación de las municiones. Se empleará el menor número posible de personas. Otras actividades como colocación de estopines, espoletas, ensamblado, desensamblado o cualquier otra tarea de mantenimiento, se harán fuera del pañol en los lugares especialmente diseñados que estén dotados con sistema de rociado. Para ello las municiones o sus componentes deben conectarse a tierra siguiendo los procedimientos indicados y con conectores que no superen los 25 Ω de resistencia.

1.4 Objetivos

Para hacer frente a las limitaciones que el sistema de monitorización y control de inventario de fusiles HK presentan actualmente (que es básicamente manual), se plantea como objetivo de este trabajo el desarrollo de un sistema basado en tecnologías de la información y las comunicaciones, que complemente a los mecanismos actualmente empleados. Entre las funcionalidades que el sistema a desarrollar debe ofrecer, se incluyen:

- 1) Monitorización de las variables ambientales del pañol, centrándose sobre todo en las más relevantes según la normativa para la conservación de la munición (temperatura y humedad).
- 2) Control de inventario de los fusiles HK, controlando los armeros para detectar la presencia o ausencia de cada fusil en su caja.
- 3) Monitorización y detección de accesos al pañol y, en caso de que sean accesos autorizados, registro de personas que han accedido.
- 4) Uso de la tecnología de comunicación inalámbrica LoRa para transmitir la información a distancia, a una estancia (como podría ser por ejemplo la sala en la que se ubica el cuerpo de guardia) desde la que se controlaría remotamente el estado del pañol.
- 5) Presentación de la información en una interfaz gráfica, de tipo panel de control (*dashboard*), en la que se mostrarán los principales indicadores del estado del pañol (variables ambientales, estado de los armeros, etc.) facilitando la adquisición de la información y la toma de decisiones (por ejemplo, activar alarmas en caso de detectar intrusiones).

1.5 Estructura de la memoria

Capítulo 1: Introducción:

En este capítulo se hablará de la normativa básica y el objetivo principal de este trabajo.

Capítulo 2: Estado del arte:

En este capítulo se hablará de las herramientas y los programas a utilizar en nuestro proyecto, así como de otros proyectos o herramientas relacionados con su ámbito.

Capítulo 3: Desarrollo del TFG:

En este capítulo se comenta el programa de monitorización y control remoto del estado del armero.

Capítulo 4: Prueba y Validación:

En este capítulo realizaremos las pruebas para comprobar que los programas instalados funcionan correctamente.

Capítulo 5: Conclusiones y Línea futuras:

Aquí se hablará de los resultados obtenidos en el proyecto, así como de las líneas futuras a seguir.

2 ESTADO DEL ARTE

En este capítulo se exponen las tecnologías y los programas que se utilizan en el trabajo. Además, las características de los sensores que proporciona en el trabajo.

2.1 Programa para control de inventario

En este trabajo no encontré ningún programa de monitorización y control de inventario a pañoles orientado de munición. Pero es un tipo de programa que las empresas utilizan para el control de sus almacenes [3]. El programa control de inventarios permite que la empresa pueda controlar y organizar las existencias de sus productos. El programa puede dar el seguimiento del inventario, y alguno también puede ayudarle a administrar las relaciones con los clientes y proveedores. Es decir que, el programa puede enviar alertas en el momento en que la cantidad de las existencias de un producto alcanzan un valor mínimo. De esta forma, tendrá un control de las existencias, y sabrá cuando debe solicitar nuevas existencias de un producto, para evitar pérdidas al no poder realizar alguna venta.

2.2 Internet de las cosas

Internet [4] es una de las creaciones más importantes y poderosas de toda la historia de la humanidad. IoT (*Internet of things*, Internet de las cosas) representa la evolución de Internet, que será un salto en su capacidad para reunir, analizar y distribuir datos que podemos convertir en información y conocimiento. Hoy en día un gran número de dispositivos inteligentes se comunican entre ellos y con otros sistemas de control. Este concepto se conoce como M2M (comunicaciones de máquina a máquina).

En 2010, el número de los dispositivos conectados a Internet fue de alrededor de 12.500 millones por lo que el número de dispositivos conectados por persona es superior a 1 por primera vez en la historia. Según las previsiones de Cisco [4] se esperaba que el número de dispositivos llegase a 25.000 millones en 2015 y a 50.000 millones en 2020, tal como se muestra en la Figura 2-1.

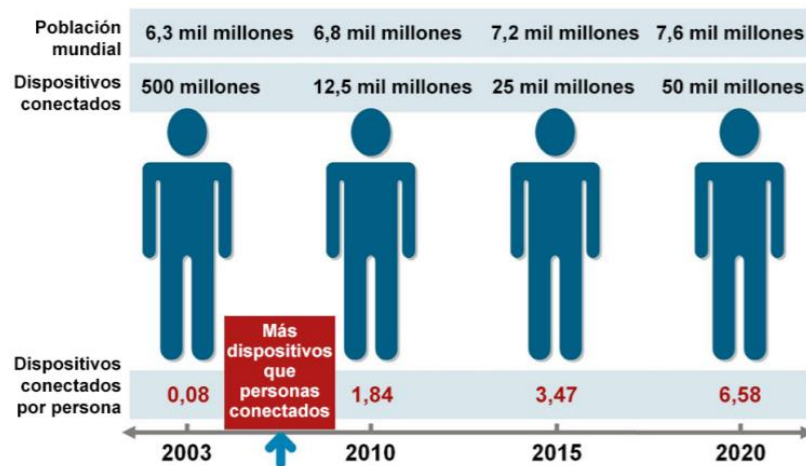


Figura 2-1 El número de los dispositivos conectados a Internet (Fuente: www.cisco.com)

Con respecto a IoT, en el ámbito de este trabajo nos interesan fundamentalmente las tecnologías LPWAN (*Low Power Wide Area Networks*, Redes de baja potencia y área amplia) que proporcionan una gama más amplia de aplicaciones M2M e IoT. Sus características son bajo consumo, largo alcance y coste de dispositivos más bajo que los de las redes móviles tradicionales. Algunos ejemplos de tecnologías LPWAN son LoRa/LoRaWAN, SigFox, IEEE 802.15.4k LECIM, IEEE 802.15.4g SUN, RPMA o DASH-7. En el trabajo, usamos LoRa porque era un requisito del TFG (por eso no se hace un estudio comparando tecnologías)

2.3 LoRa

LoRa Alliance es una organización abierta cuyo objetivo es estandarizar una tecnología de comunicación de red que cumpla con una serie de características necesarias para IoT. LoRa (*Long Range*, Largo alcance) [5] es una tecnología LPWAN que mantiene las mismas características de baja potencia como la modulación FSK (*Frequency Shift Keying*, modulación por desplazamiento de frecuencia) pero aumentando significativamente el alcance de la comunicación. La red LoRa trabaja en modo P2P (*Peer to Peer*) para permitir que los nodos puedan conectarse directamente entre ellos sin costes, ya que no utilizan infraestructura de red. Este modo funciona sin necesidad de contar ni con clientes ni con servidores fijos. Es decir, el modo de comunicación P2P de LoRa permiten el intercambio directo de información, en cualquier formato, entre los equipos interconectados.



Figura 2-2 LoRa (Fuente: www.7layers.com)

2.3.1 Canales y rangos de frecuencias

LoRa es una tecnología propietaria patentada de por Semtech Corporation [6], que funciona en la banda ISM (*Industrial, Scientific and Medical*, Industrial, Científico y Médico). La asignación de frecuencias y los requisitos reglamentarios para ISM son distintos de frecuencia en distintas regiones del mundo. La banda de frecuencias utilizada en Europa es la banda ISM de 867-869 MHz según la normativa vigente [7] que recoge el cuadro nacional de atribución de frecuencias (orden ETU/1033/2017 de 25 de octubre). Otros detalles técnicos relativos al uso de LoRa en Europa se recogen en la Tabla 2-1.

	Europa
Banda de frecuencia	867-869 MHz
Canales	10
Canal banda ancha ascendente	125/250 kHz
Canal banda ancha descendente	125 kHz
Velocidad de datos	250 bps - 50 kbps
Balance del enlace ascendente	155 dB
Balance del enlace descendente	155 dB

Tabla 2-1 Las especificaciones LoRa para Europa (Fuente: www.lora-alliance.org)

2.4 LoRaWAN

Es un protocolo de control de acceso al medio (*MAC, Media Access Control*) para conectar sus dispositivos, mientras que la LoRa ofrece funcionalidad de capa física. Por eso LoRaWAN (Figura 2-3) se implementa en la parte superior de la capa física LoRa con funciones que tienen influencia sobre varios parámetros de funcionamiento, que incluyen [8]:

- La vida útil de la batería de un nodo
- La capacidad de la red
- La seguridad de la red
- Las aplicaciones a las que sirven



Figura 2-3 LoRaWAN (Fuente: www.intelilight.eu)

2.4.1 Red de LoRaWAN

LoRaWAN puede operar de dos maneras: en modo P2P, como en las redes LoRa, y en modo híbrido. En este último caso la arquitectura de red LoRaWAN utiliza una topología en estrella, en la que un nodo central (*Concentrador*) actúa como pasarela de red (*Gateway*) conectando una serie de nodos LoRa con un conjunto de servidores. La conexión de la pasarela con los nodos se hace de manera directa usando el modo P2P de LoRa, mientras que la comunicación con los servidores se lleva a cabo mediante IP. La arquitectura de red del modo híbrido se muestra en la Figura 2-4, donde también se pueden ver los principales elementos involucrados en la misma, estos son:

- Los nodos son dispositivos con la capacidad de recopilar información de un sensor, y comunicarse con otros nodos o pasarelas de red.
- El concentrador actúa como un puente para retransmitir los datos bidireccionales entre los nodos y los servidores de red.
- El servidor de red se conecta a varios concentradores a través de una conexión TCP/IP segura, ya sea por cable o inalámbrica, elimina los mensajes duplicados. Decide qué servidor debe responder a un mensaje de nodo y gestiona la capacidad de la red, las velocidades de transmisión de datos usando una velocidad de datos adaptable (*ADR*, *Adaptive Data Rate*) para extender la vida útil de la batería del nodo.
- El servidor de la aplicación determina las acciones del nodo y analiza los datos de los sensores.

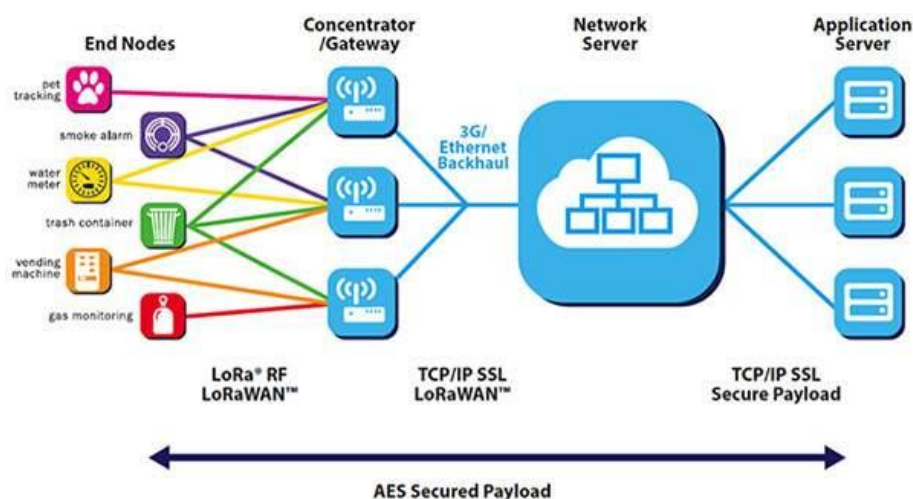


Figura 2-4 Esquema modo híbrido de LoRaWAN (Fuente: www.digikey.es)

2.4.2 Canales y velocidades de transmisión

La comunicación entre nodos y pasarela [9] se realiza utilizando diferentes frecuencias y velocidades de datos. Para maximizar el uso de las baterías de los nodos y la capacidad de la red, el servidor de red puede gestionar la velocidad de datos para cada nodo individualmente por medio de un esquema adaptativo de velocidad de datos. A la hora de transmitir datos los nodos LoRa deben cumplir las siguientes normas:

- El nodo cambia de canal de forma pseudo-aleatoria para cada transmisión. La diversidad de frecuencia resultante permite que el sistema sea más robusto a las interferencias. Los módulos LoRaWAN en Europa utilizan 16 canales de frecuencia en las bandas de 433 MHz y 868 MHz.
- El nodo respeta el máximo ciclo de trabajo de transmisión y la duración máxima en relación con la sub-banda utilizada y a las regulaciones locales.

2.4.3 Clasificaciones de nodos LoRaWAN.

Existen tres clases denominadas Clase A, Clase B y Clase C. Todas permiten la comunicación bidireccional. Es decir, se puede iniciar una subida a los servidores a través de la puerta de enlace y también aceptar mensajes entrantes del servidor (se muestra en la Figura 2-5).

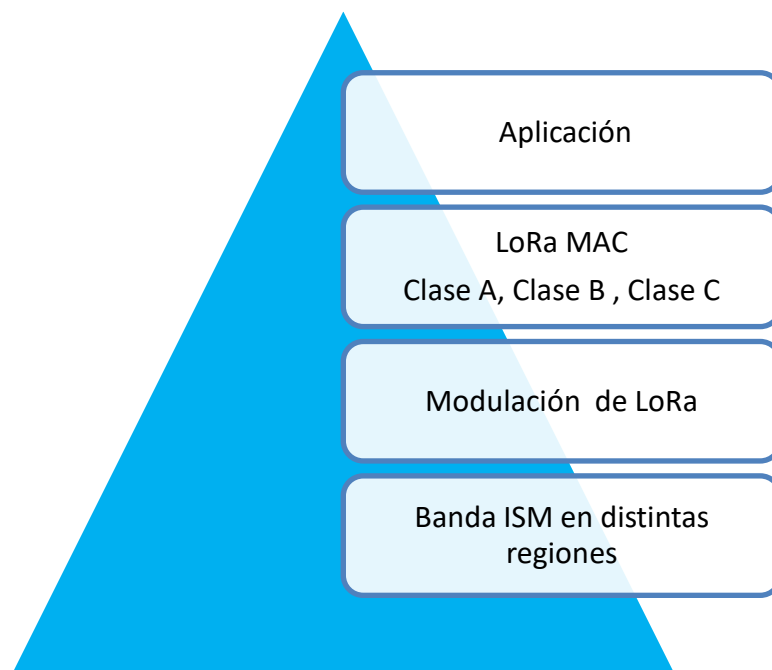


Figura 2-5 Esquema de clases de nodos LoRaWAN (Fuente: www.digikey.es)

- Clase A

Los dispositivos de clase A consumen menos energía y servirán para aplicaciones en las que los dispositivos no deban recibir datos habitualmente. Un nodo clase A sólo permite la comunicación desde el servidor durante dos ventanas de recepción, que se abren durante un corto período de tiempo después de una transmisión de enlace ascendente para recibir un paquete de vuelta que contiene el

asentimiento (*ACK*) del paquete enviado. Así que los mensajes desde el servidor en cualquier otro momento deben esperar hasta la próxima transmisión de enlace ascendente.

- Clase B

Los dispositivos de clase B ofrecen la funcionalidad de los de clase A, pero también añaden la capacidad de recibir datos sin necesidad de enviar un paquete. Es decir que permiten a los dispositivos estar sincronizados con la pasarela de forma que pueden negociar tiempos de recepción de paquetes desde la pasarela al dispositivo. Así, las ventanas de recepción se abren en momentos determinados, acordados previamente con la pasarela.

- Clase C

Los dispositivos de clase C están permanentemente escuchando. Por eso proporcionan los mejores tiempos de respuesta y capacidad de envío desde el servidor a los dispositivos. En estecaso, las ventanas de recepción están continuamente abiertas (excepto cuando transmiten), pero a costa de un consumo energético mucho mayor respecto a las clases A y B.

2.4.4 Seguridad de LoRaWAN

LoRaWAN incluye seguridad en dos niveles independientes usando una clave de sesión de red (*Network Session Key, NwkSKey*) y una clave de sesión de aplicación (*Application Session Key, AppSKey*). Las dos claves hacen uso del algoritmo de cifrado AES (*Advanced Encryption Standard, Estandar Avanzado de Cifrado*) para proteger las comunicaciones (se muestra en la Figura 2-6):

- La clave de sesión de red es una clave de 128 bits que garantiza la seguridad a nivel de red.
- La clave de sesión de aplicación es una clave de 128 bits que garantiza la seguridad extremo a extremo a nivel de aplicación.

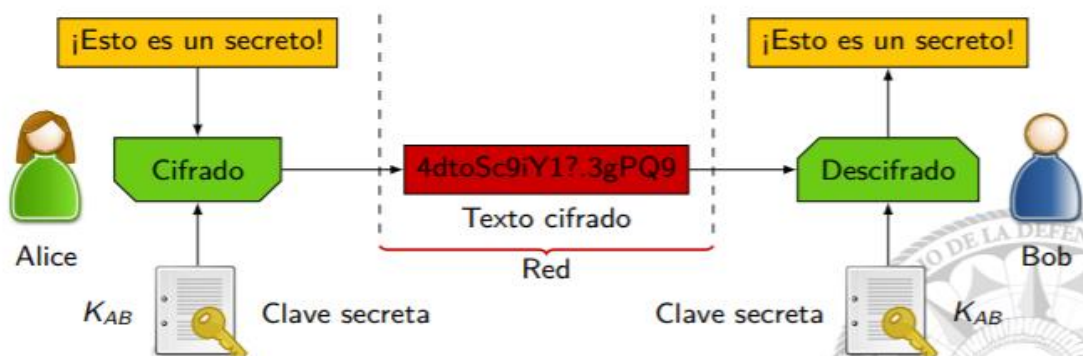


Figura 2-6 Cifrado de clave simétrica (Fuente: Apuntes de la asignatura Ampliación de Informática)

Existen básicamente dos métodos para distribuir las claves que permiten establecer la comunicación segura: preinstalarlas en los nodos y hacer que sean conocidas a priori por las partes en comunicación y negociarlas dinámicamente. Este 2º escenario es más seguro, pero más complejo y requiere de una clave adicional denominada clave de aplicación (*AppKey*) también de 128 bits que se utiliza en el proceso de negociar las claves de sesión.

2.5 Herramientas de proyecto

2.5.1 LoRa Shield para Arduino



Figura 2-7 Shield de Dragino (Fuente: www.dragino.com)

Un *shield* [10] es una placa impresa que se pueden conectar en la parte superior de la placa Arduino de forma que nos permite ampliar el hardware/capacidades de Arduino, pudiendo ser apilada una encima de la otra. Un *shield* se alimenta generalmente a través del Arduino mediante los pines de 5V y GND.

El Dragino LoRa Shield (ver Figura 2-7) [11] es un transceptor de largo alcance basado en una biblioteca de código abierto. Permite el envío de datos con alcance muy largo a baja tasa de datos, y con alta inmunidad a interferencias mientras que minimiza el consumo. La Tabla 2-2 resume sus características:

Peso	50 g
Voltaje operativo	3,3V o 5V
Balance del enlace	Máximo de 168 dB
Rango Dinámico de RSSI	127 dB
Banda de frecuencia	868 MHz/433 MHz (preconfigurado en fábrica)
Compatible con	Arduino Leonardo, Uno, Mega
Paquete máximo	256 bytes con CRC
Alta sensibilidad	Hasta -148 dBm
Modulación	FSK, GFSK, MSK, GMSK, LoRa y OOK

Tabla 2-2 Características de LoRa Shield (Fuente: www.dragino.com)

2.5.2 Sensor de presencia PIR

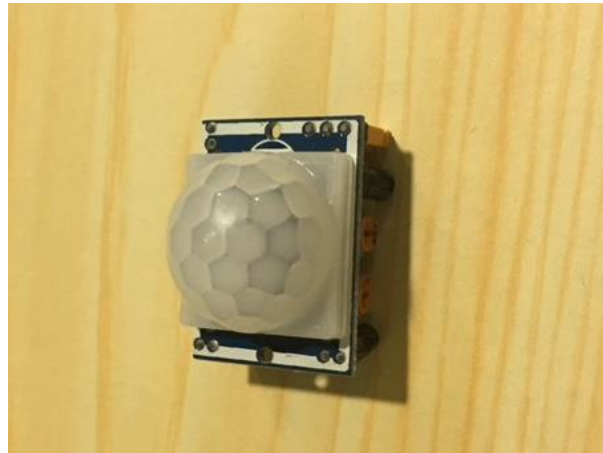


Figura 2-8 Sensor PIR (Fuente: elaboración propia)

Para la detección de movimiento dentro del pañol usaremos un sensor piroeléctrico (Figura 2-8), que básicamente detecta los cambios en la señal infrarroja que recibe. Por lo general, los objetos que tienen la temperatura por encima de cero emiten calor. Esta radiación es invisible para el ojo humano, pero puede ser detectada por dispositivos electrónicos diseñados para tal propósito. En el TFG usaremos el sensor HC-SR501. Este sensor cuenta con reguladores para ajustar el retardo de detección y de restablecimiento.

Características [12]:

- Voltaje de alimentación: 5V.
- Rango de medición: hasta 6m.
- Salida: estado de un pin digital.

2.5.3 Sensor de temperatura y humedad DHT 22

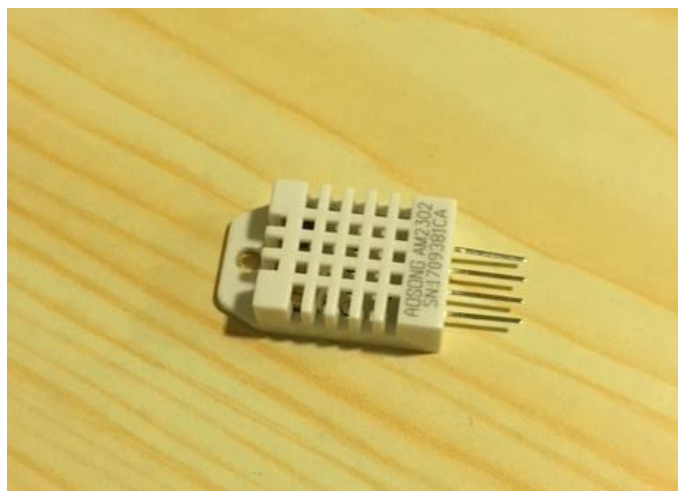


Figura 2-9 Sensor DHT 22 (Fuente: elaboración propia)

Para la medición de las variables ambientales de temperatura y humedad en el pañol utilizaremos un sensor DHT 22 (Figura 2-9), que consiste en la combinación de un sensor capacitivo de humedad y un termistor para medir la temperatura del aire circundante. Este sensor muestra los datos mediante una señal digital en el pin de datos de Arduino (ver Figura 2-10).

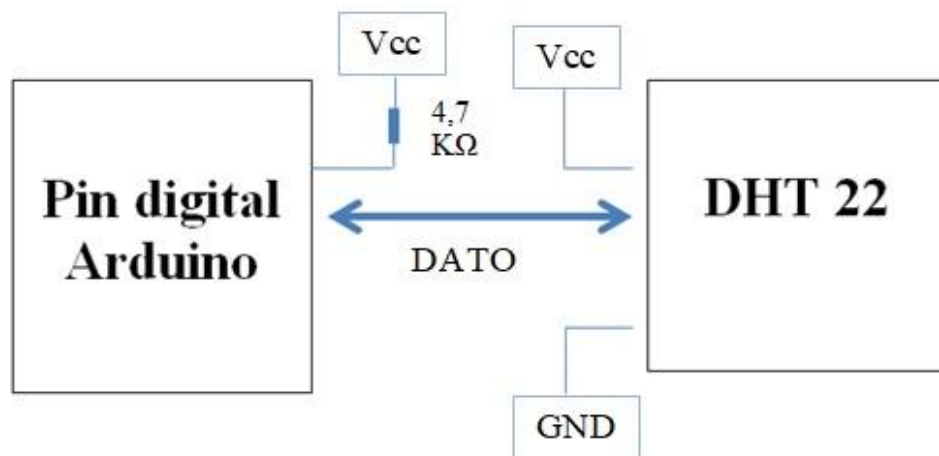


Figura 2-10 Esquema conexión DHT 22 (Fuente: elaboración propia)

Características:

- Alimentación: $3.3\text{Vdc} \leq V_{cc} \leq 6\text{Vdc}$
- Rango de medición de temperatura: -40°C a 80°C
- Precisión de medición de temperatura: $\pm 0.5^{\circ}\text{C}$
- Resolución Temperatura: 0.1°C
- Rango de medición de humedad: De 0 a 100% RH
- Precisión de medición de humedad: 2% RH
- Resolución Humedad: 0.1% RH
- Tiempo de muestreo: 2s

2.5.4 Sensor HC-SR04

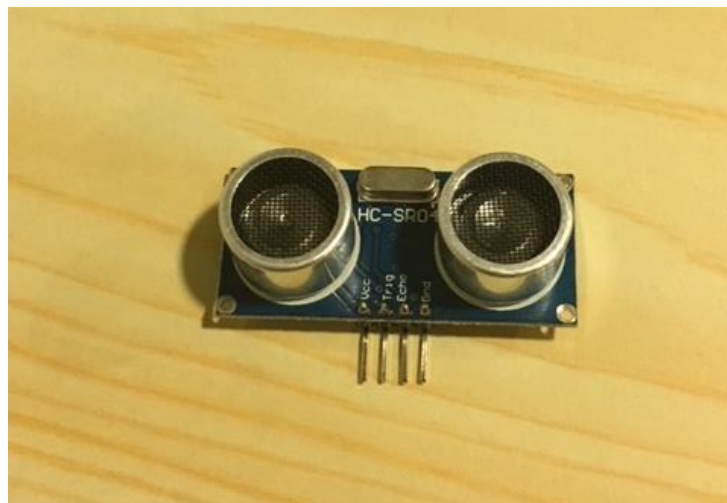


Figura 2-11 Sensor HC-SR04 (Fuente: elaboración propia)

Para detectar el estado de la puerta principal del pañol (abierta o cerrada) usaremos un sensor HC-SR04 (Figura 2-11) se trata de un sensor de distancia que utiliza ultrasonido para determinar el alejamiento o acercamiento de un objeto en un rango de 2 a 450 cm.

El principio de funcionamiento del sensor (que se puede ver en la Figura 2-12) es el siguiente: un emisor piezoeléctrico emite 8 pulsos de ultrasonido con frecuencia de 40KHz tras, después de recibir la orden en el pin TRIG. Las ondas de sonido viajan en el aire y rebotan al encontrar un objeto, el sonido de rebote es detectado por el receptor piezoeléctrico, con lo que el pin ECHO del sensor cambia a valor “alto” (5V) por un tiempo igual al que demoró la onda desde que fue emitida hasta que fue detectada, el tiempo del pulso ECHO es medido por el microcontrolador y así se puede calcular la distancia al objeto.

Características [13]:

- Tensión de alimentación: 5 V
- Frecuencia de trabajo: 40 KHz
- Rango máximo: 4.5 m
- Rango mínimo: 1.7 cm
- Duración mínima del pulso de disparo: 10 μ s.
- Duración del pulso eco de salida: 100-25000 μ s.
- Tiempo mínimo de espera entre una medida y el inicio de otra 20 ms.

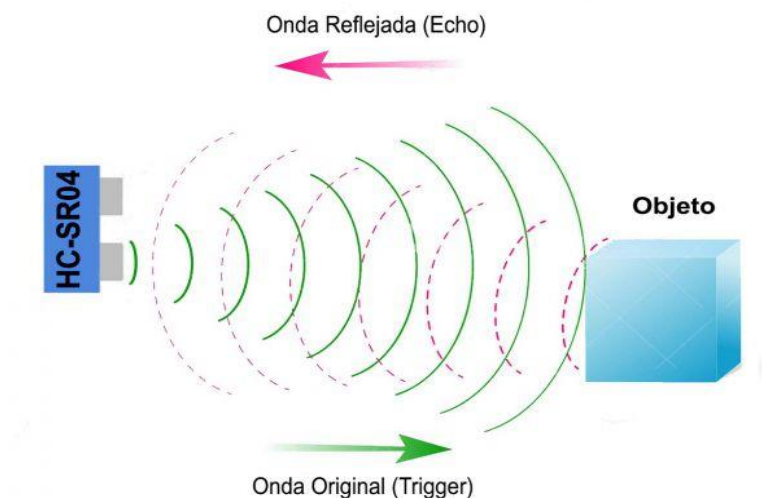


Figura 2-12 Funcionamiento de sensor HC – SR04 (Fuente: www.zonamaker.com)

2.5.5 Sensor de huella dactilar

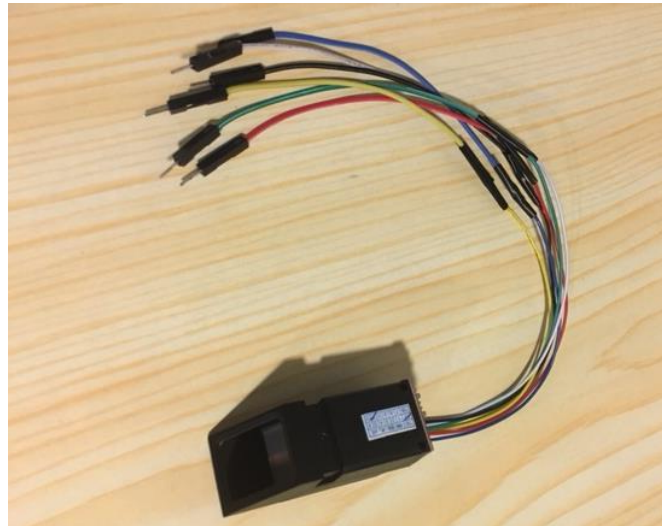


Figura 2-13 Sensor de huellas dactilares(Fuente: elaboración propia)

Con el objetivo de mantener un registro del personal autorizado que acude al pañol es necesario algún mecanismo de identificación de personal. En el contexto de este TFG se ha hecho uso de un sensor de huella dactilar, el sensor óptico de lector de huella dactilar Kookye (Figura 2-13). Es un dispositivo capaz de leer, guardar e identificar posteriormente las huellas de dedos. Para eso pueden inscribirse nuevos dedos directamente hasta un máximo de 162 huellas digitales, que se almacenan en la memoria *flash* del dispositivo. El módulo se comunica con el Arduino a través de un puerto serie. La Tabla 2-3 resume sus características:

Dimensiones	56*20*20.5mm
Voltaje operativo	3,3V o 5V
Corriente de funcionamiento	< 65mA
Tiempo de lectura de huella	< 1s
Área de la ventana	14.5*19.4mm
Tiempo de búsqueda	< 1s
Ambiente de trabajo	-20 ° C a 50 ° C
Tasa de falsa aceptación	< 0.001%
Tasa de falso rechazo	< 1.0%

Tabla 2-3 Características del sensor de huella dactilar (Fuente: www.instructables.com)

Para almacenar las huellas en el sensor se utilizó un código de ejemplo (*sketch*) que permite operar el sensor registrando nuevas huellas y asociado a cada una de ellas un identificador entero (1-99). El operador tiene que añadir este número con su nombre y apellido en la tabla de huellas de personas autorizadas en una base de datos existente en la interfaz de usuario de forma que se pueda asociar un identificador con una persona concreta (se explicará esto posteriormente en la sección 3.4)

2.5.6 Sensor LDR

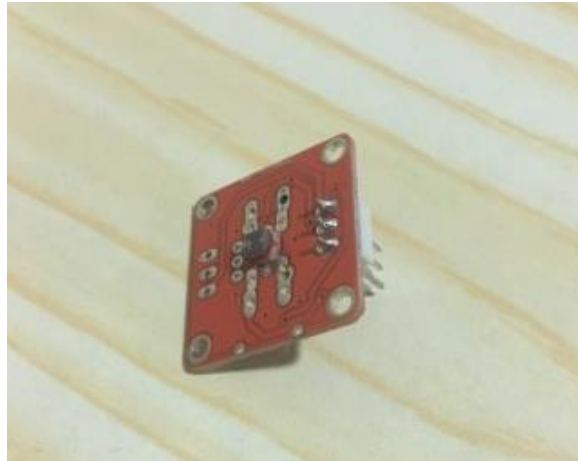


Figura 2-14 Sensor LDR (Fuente: elaboración propia)

Con el objetivo de monitorizar el nivel de iluminación dentro del pañol se ha hecho uso de un sensor LDR (*Light Dependent Resistor*, resistencia dependiente de luz) (Figura 2-14). Es una resistencia que varía su valor en función de la cantidad de luz que incide sobre ella. Cuando no incide luz sobre el sensor el valor de la resistencia es muy alto, y a medida que se incrementa la luz que incide, baja el valor de la resistencia.

Características [14]:

- Consumo máximo de energía: 100 mW
- Resistencia con luz: 10 - 20K Ω
- Resistencia a oscuras: 1M Ω
- Temperatura de funcionamiento: desde -30 °C hasta 70 °C

2.5.7 Arduino

2.5.7.1 Qué es un Arduino



Figura 2-15 Arduino (Fuente: www.arduino.cc)

Arduino [15] es un microcontrolador, una placa, un pequeño sistema de procesamiento de hardware libre. El desarrollo de este tipo de placas de bajo coste, concede una oportunidad a los aficionados de la electrónica y del sector de la educación, para desarrollar sus pequeños proyectos con un bajo coste. El hardware consiste en una placa con un microcontrolador y diferentes puertos de entrada y salida. Una de las características que hacen tan atractivo a Arduino es que se puede utilizar para conectar a otro tipo de software por ejemplo C#, Python o Java. Además, dispone de un IDE (*Integrated Development Environment*, entorno de desarrollo) libre y un lenguaje de programación propio, simplificado. Esto permite acercar el mundo de la electrónica a personas con pocos conocimientos sobre microcontroladores o con pocos recursos económicos, ya que las placas Arduino se programan directamente desde su IDE, con un simple cable USB, sin necesidad de programadores específicos.

2.5.7.2 Historia de Arduino

El proyecto Arduino surgió en el año 2005 de mano de unos estudiantes del Instituto de Diseño Interactivo IVREA como un proyecto para los estudiantes de computación y electrónica del instituto. Además, inicialmente fue un proyecto creado no solo para economizar la creación de proyectos escolares dentro del instituto, sino también tenía la intención de ayudar a su escuela a evitar la quiebra de la misma con las ganancias que produciría vendiendo sus placas dentro del campus a un precio accesible. El primer prototipo de Arduino fue fabricado en el instituto IVREA. Es una simple placa de circuitos eléctricos, donde estaban conectados un microcontrolador junto con resistencias, sensores simples y LEDs pero dado que el coste los prototipos era todavía elevado, hubo que buscar alternativas. Para ello contaron con la colaboración del estudiante de la Universidad de Colombia Hernando Barragán, quien, bajo la dirección de Massimo Banzi, desarrolló la tarjeta electrónica Wiring en colaboración con David Mellis, otro integrante del instituto IVREA, quien más adelante, mejoraría la interfaz de software. Tiempo después, se unieron Tom Igoe como consejero y Gianluca Martino que era quien podía producir las placas. Éste fue y es el equipo de Arduino que se puede ver en la Figura 2-16. El nombre de Arduino nace de un bar donde uno de sus creadores pasaba algunas horas, el cual a su vez viene del nombre de un antiguo rey italiano allá por el año 1002. Un tiempo más tarde, al ver los grandes resultados que tuvo Arduino y la gran aceptación que tuvo por parte del público, comenzó a distribuirse en Italia, después en España, hasta colocarse en el número uno de

herramientas de aprendizaje para el desarrollo de sistemas, siendo además muy económica en comparación con otras placas de microcontroladores.



Figura 2-16 El equipo de Arduino (Fuente: blog.bricogeek.com)

2.5.7.3 Arduino Uno

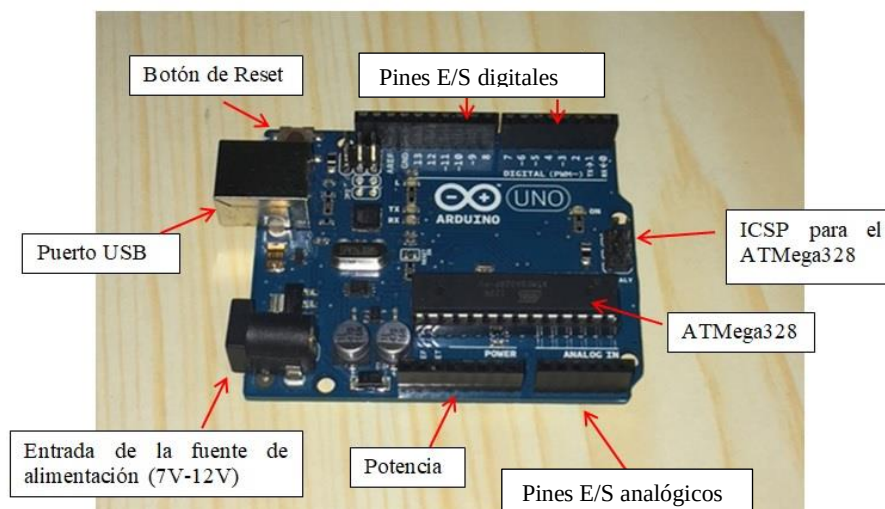


Figura 2-17 Arduino Uno (Fuente: elaboración propia)

En el presente trabajo usaremos el Arduino Uno Rev3. Se trata de una placa electrónica basada en el microprocesador ATmega328, la cual es la evolución de la placa Arduino Duemilanove que incorpora un nuevo chip (ATmega16U2), con convertidor de USB a serie. La placa tiene un tamaño de 75x53 mm. y se muestra en la Figura 2-17.

La siguiente Tabla 2-4 resume sus principales características:

Tamaño	75x53 mm
Microcontrolador	ATMega328
Voltaje operativo	5V
Voltaje de entrada (recomendado)	7-12 V
Voltaje de entrada (límites)	6-20 V
Pines digitales E/S	14(con salida PWM, (<i>Pulse-width modulation</i>))
Pines analógicos	6
Corriente continua para pines E/S	40 mA
Corriente continua para pines de 3,3V	50 mA
Corriente continua para pines 5V	300 mA
Memoria Flash	32 KB, de los cuales 0,5 son para el <i>bootloader</i>
SRAM	2 KB
EEPROM	1 KB
Velocidad del reloj	16 MHz

Tabla 2-4 Características de Arduino (Fuente: www.arduino.cc)

2.5.8 IDE Arduino

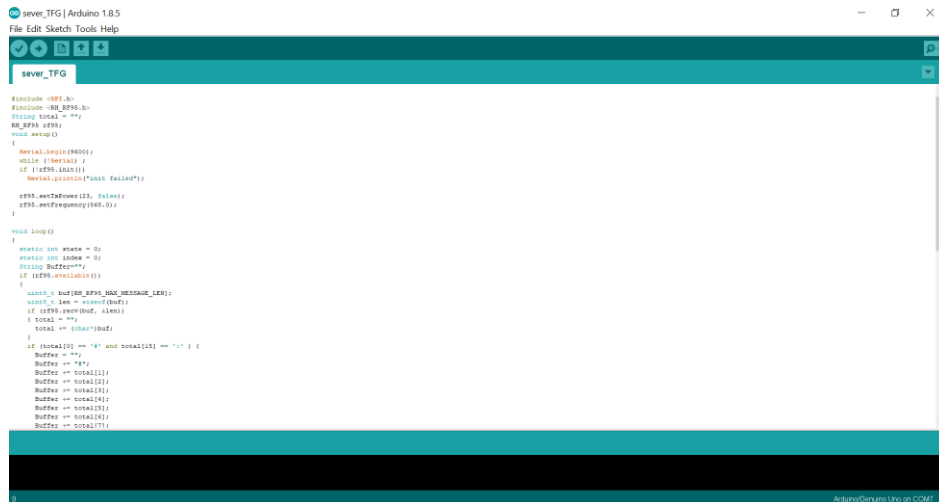


Figura 2-18 IDE Arduino (Fuente: elaboración propia)

Es un entorno de programación que ha sido empaquetado como un programa de aplicación de trabajo con Arduino. Es decir que consiste en un editor de código, un compilador, un depurador y una interfaz gráfica (*GUI, Graphical User Interface*).

2.5.8.1 Estructura básica de un programa Arduino

Todo programa de Arduino suele consistir en dos elementos fundamentales implementados:

- **Setup:** Esta función solo se ejecuta una vez al inicio del programa, y es aquí donde deberemos hacer las tareas de inicialización de variables, abrir canales de comunicación (puerto serie, Ethernet, etc).
- **Loop:** El funcionamiento de Arduino es ejecutar un bucle hasta el infinito. Esta función, es el cuerpo principal del programa, que se ejecutará indefinidamente.

2.5.8.2 El menú archivo

El menú archivo (Figura 2-19) ofrece una serie de opciones de gestión de ficheros de código. Las más utilizadas en el TFG han sido:

- **New y Open:** Respectivamente nos permite crear y abrir el código fuente de un preograma (*sketch*).
- **Examples:** Son sketch que proporcionan ejemplos sobre el uso de la placa de Arduino.

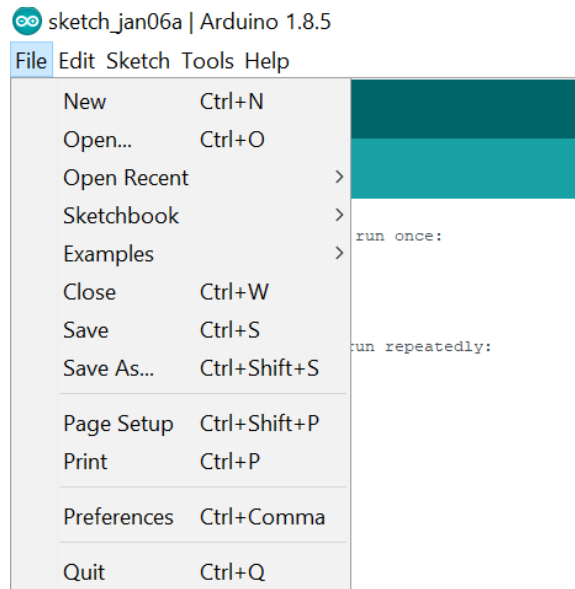


Figura 2-19 El menú archivo (Fuente: elaboración propia)

2.5.8.3 El menú sketch

Dentro del menú sketch (Figura 2-20) aparecen las opciones de verificar y compilar el programa, incluir librerías y mostrar la carpeta del programa.

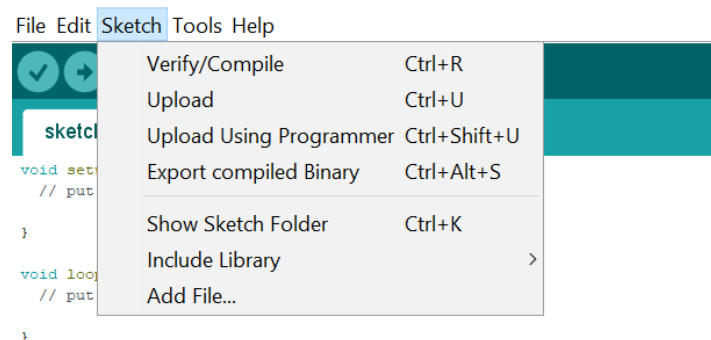


Figura 2-20 El menú sketch (Fuente: elaboración propia)

2.5.8.4 El menú herramientas.

Dentro del menú herramientas (Figura 2-21) se ofrecen algunas utilizadas en el desarrollo, como por ejemplo:

- **Auto format:** Formatea nuestro código de tal forma que sea legible.
- **Archive Sketch:** Comprime toda la carpeta del proyecto para guardarla en un fichero ZIP.
- **Fix Encoding & Reload:** Esta opción es útil para reparar ficheros con diferentes codificaciones.

- **Monitor serie:** Es un simple monitor serie, que nos permite comunicarnos con la placa. Es decir que muestra los datos enviados por el Arduino a través del puerto serie y también nos permite mandar datos al Arduino mediante el puerto serie.
- **Board:** Nos permite seleccionar con qué placa estamos trabajando (modelo).
- **Port:** Nos permite seleccionar en qué puerto está conectado el Arduino.

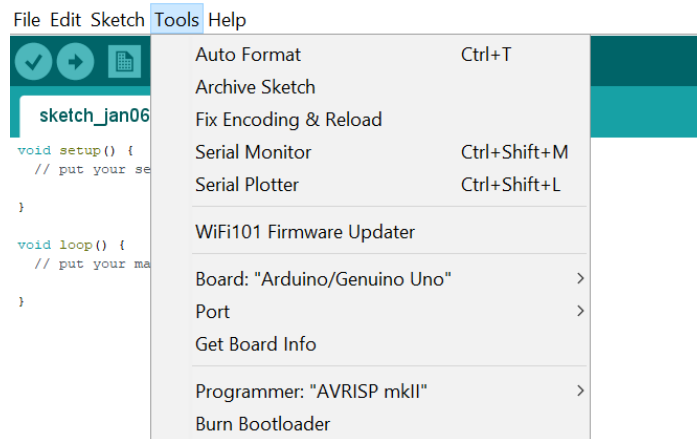


Figura 2-21 El menú herramientas (Fuente: elaboración propia)

2.5.9 Microsoft Visual Studio

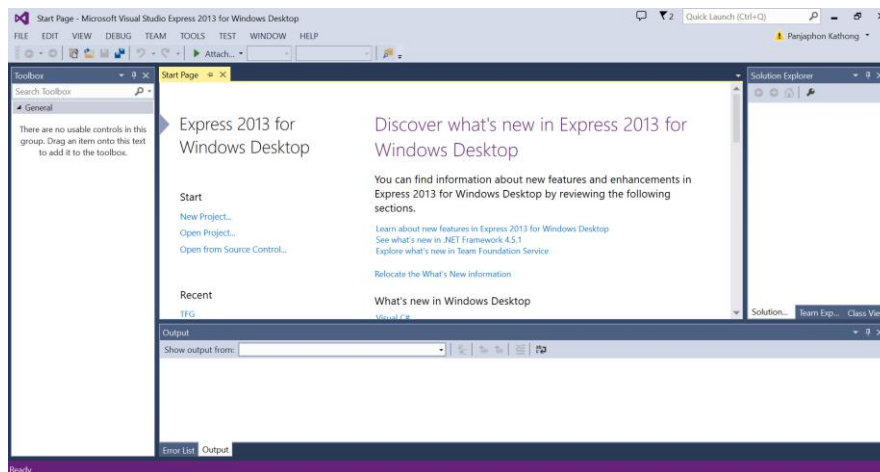


Figura 2-22 Visual Studio (Fuente: elaboración propia)

Se trata de un entorno de desarrollo integrado para sistemas operativos Windows. Soporta múltiples lenguajes de programación. A partir de la versión 2005, Microsoft ofrece gratuitamente las Ediciones Express, que son versiones básicas separadas por lenguajes de programación como plataformas enfocadas a estudiantes. Además, Visual Studio permite a los desarrolladores crear sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002).

En el trabajo de fin de grado, se utiliza Visual Studio para desarrollar de aplicación de interfaz de usuario usando C#, el popular lenguaje de programación de Microsoft, que dispone de forma nativa de librerías para usar de forma muy sencilla los puertos de serie, lo que nos permite comunicarnos con Arduino.

2.5.9.1 Herramientas más utilizadas en el TFG

- **New Project y Open Project:** Respectivamente nos permite crear y abrir un proyecto.

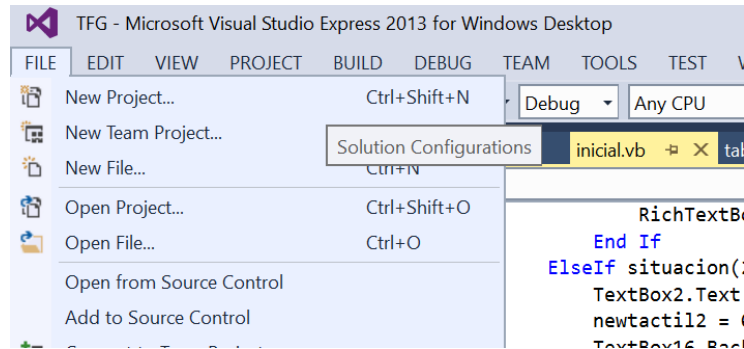


Figura 2-23 El menú archivo de Visual Studio (Fuente: elaboración propia)

- **Toolbox:** se ofrecen algunas herramientas utilizadas en el desarrollo de la interfaz del proyecto

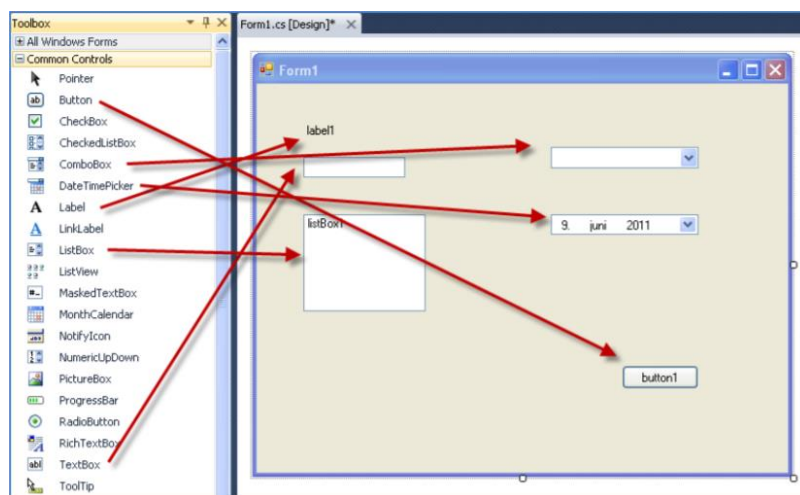
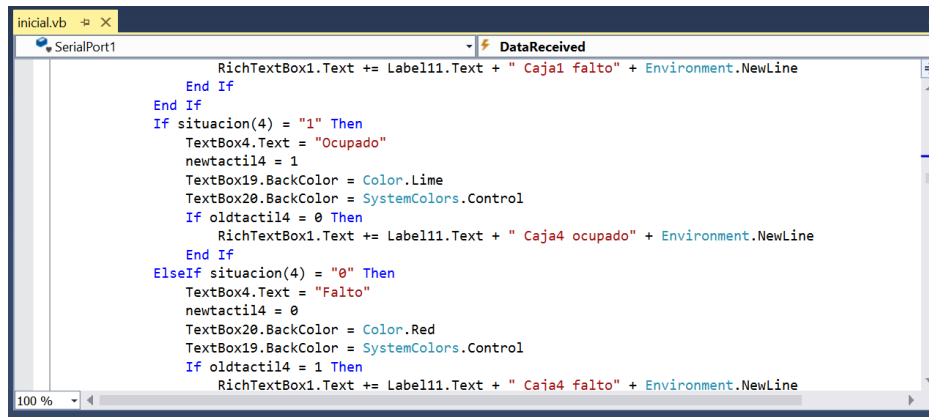


Figura 2-24 Las herramienta de Visual Studio (Fuente: [16])

- **Code Editor:** Nos permite escribir el código. El editor proporciona muchas características que le facilitan la escritura y la administración del código y del texto.



```
inicial.vb  SerialPort1  DataReceived
    RichTextBox1.Text += Label11.Text + " Caja1 falto" + Environment.NewLine
End If
End If
If situacion(4) = "1" Then
    TextBox4.Text = "Ocupado"
    newtactil4 = 1
    TextBox19.BackColor = Color.Lime
    TextBox20.BackColor = SystemColors.Control
    If oldtactil4 = 0 Then
        RichTextBox1.Text += Label11.Text + " Caja4 ocupado" + Environment.NewLine
    End If
ElseIf situacion(4) = "0" Then
    TextBox4.Text = "Falto"
    newtactil4 = 0
    TextBox20.BackColor = Color.Red
    TextBox19.BackColor = SystemColors.Control
    If oldtactil4 = 1 Then
        RichTextBox1.Text += Label11.Text + " Caja4 falto" + Environment.NewLine
    End If
End If
```

Figura 2-25 El editor de código (Fuente: elaboración propia)

3 DESARROLLO DEL TFG

En este capítulo se explica el desarrollo de este Trabajo de Fin de Grado en cada parte, así como el funcionamiento de la interfaz gráfica de usuario.

3.1 Arquitectura del sistema

Como se puede ver en la Figura 3-1 la arquitectura del sistema se divide en dos grandes bloques: en el pañol del condestable instalaremos una serie de Arduinos que contralarán los sensores de monitorización ambientales y control de inventario de HK. Dado que los puertos de Arduino son limitados se ha optado por tener dos Arduinos con sensores que envían su introducción a un tercero mediante comunicación serie. El tercer Arduino integra la información de los sensores y la envía al segundo gran bloque de la arquitectura que estará ubicado en el puesto de control y consistirá en un Arduino para recibir los datos del pañol y pasarlos a través del puerto serie a una aplicación que se estará ejecutando en un PC y actuará como interfaz gráfica de usuario.

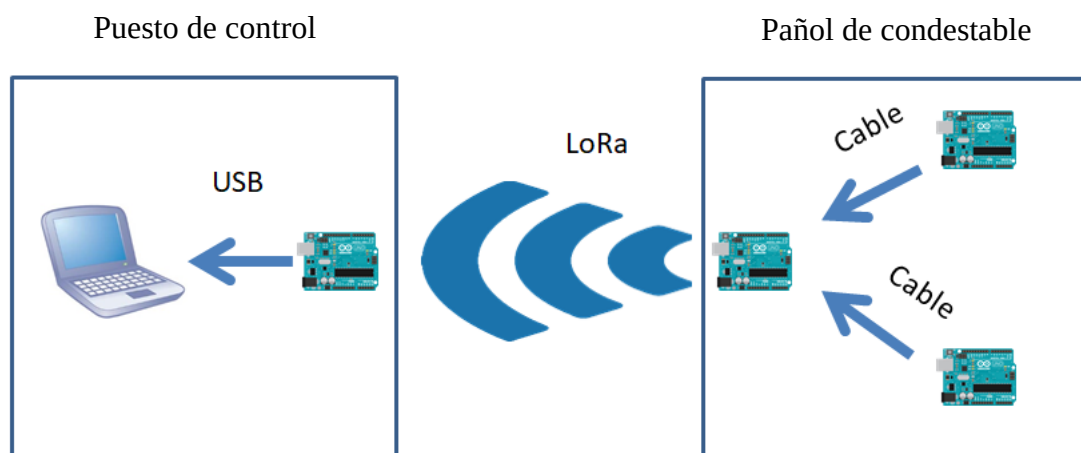


Figura 3-1 Arquitectura del sistema (Fuente: elaboración propio)

3.1.1 Método de paso de mensajes por puerto serie

Tal cómo se puede ver en la arquitectura del sistema representada en la Figura 3-1 los Arduinos necesitan comunicarse entre ellos. Para ello se hará uso del puerto serie. Este método de comunicación trabaja con una librería específica *Software Serial* y se necesita establecer pines para recibir y

transmitir datos. Para transmitir una cadena de texto, el Arduino va a enviar cada letra por separado (ver Figura 3-2), por eso se hace necesaria una marca para indicar cuando comienza el mensaje y cuando finaliza. En este trabajo se utiliza la idea del Método de Relleno de byte consistente en disponer de una marca (*flag*) especial para representar el inicio/fin de una trama (ver Figura 3-3). En nuestro caso, el mensaje comienza cuando se encuentra el indicador inicial “#” y finaliza cuando se encuentra el indicador final “:.”.

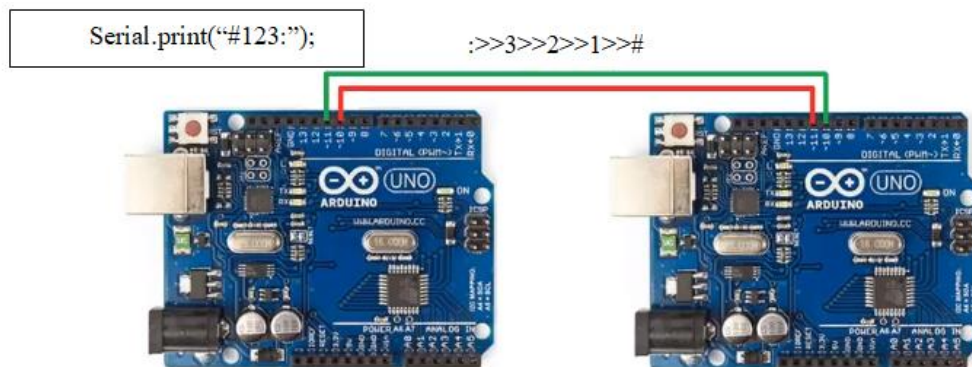


Figura 3-2 Funcionamiento del método de paso de mensaje por puerto serie (Fuente: elaboración propia)

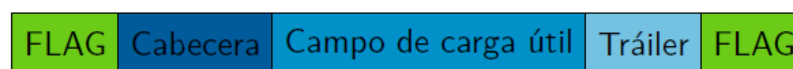


Figura 3-3 Método de relleno de byte (Fuente: Apuntes de la signatura Ampliación de Informática)

3.1.2 General

Siguiendo la arquitectura presentada en la Figura 3-1 vamos a dividir nuestros Arduinos en dos partes (servidor, observador)

- Servidor (con rol similar al concentrador de la arquitectura de LoRaWAN)

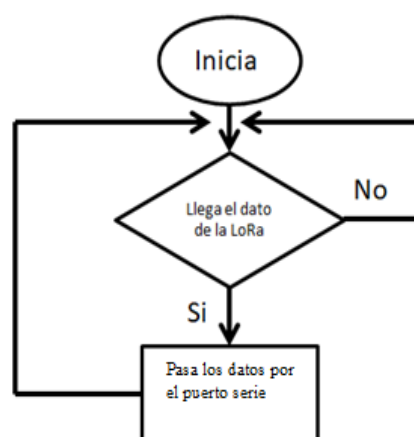


Figura 3-4 Diagrama de flujo del servidor (Fuente: elaboración propia)

En esta parte solo tenemos un Arduino con una *LoRa Shield* para recibir los datos desde el observador y transmitir esos datos a través del puerto serie a un ordenador donde se representará la información. El servidor estará instalado en el puesto de control de pañol. El diagrama de flujo del programa principal del Arduino que actúa como servidor se puede ver en la Figura 3-4.

- Observador (con rol similar a un nodo final en la arquitectura LoRaWAN)

En esta parte tenemos tres placas Arduino. Cada una de las placas lleva a cabo una tarea distinta.

1. Arduino con sensores de contacto (para controlar el inventario de fusiles HK): este Arduino recibe los datos sobre el estado del armero (si cada fusil HK se encuentra o no en su caja) y pasa los datos por el cable hacía un Arduino central utilizando el método de paso de mensajes por puerto serie.
2. Arduino con sensores ambientales (para monitorizar el estado del pañol): este Arduino recibe los datos del resto de sensores (estado de la puerta, detección de los movimientos, temperatura, humedad, detección de la luz e identificación de huella) y pasa los datos por el cable hacía el Arduino central.
3. Arduino central: es un Arduino con una *LoRa Shield* que recibe los datos de los sensores recibidos a través de puerto serie. Cuando tiene los datos de todos los sensores pasa esos datos usando LoRa hacía la parte de servidor.

3.2 Funcionamiento del sistema de monitorización

3.2.1 Observador

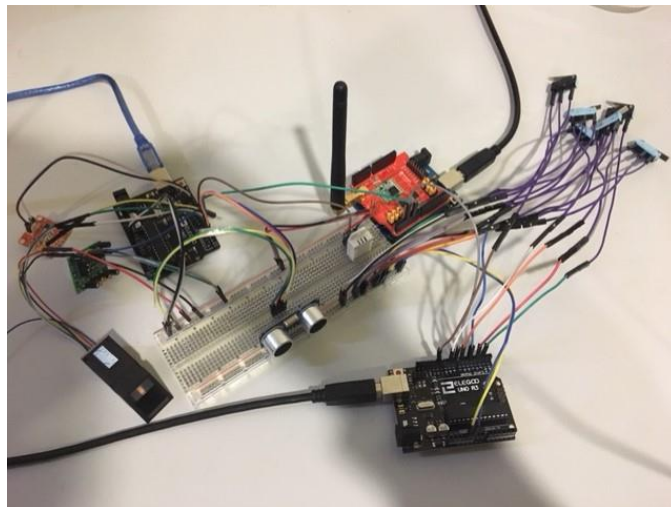


Figura 3-5 Estructura general de observador (Fuente: elaboración propia)

Esta parte se monta en el pañol y es la responsable de observar o monitorizar su estado. Dado que, como se indicó en la sección 3.1.2, tenemos tres Arduinos en este montaje, pasaremos a explicar brevemente a continuación el funcionamiento de cada uno de ellos.

- Arduino con sensores de contacto

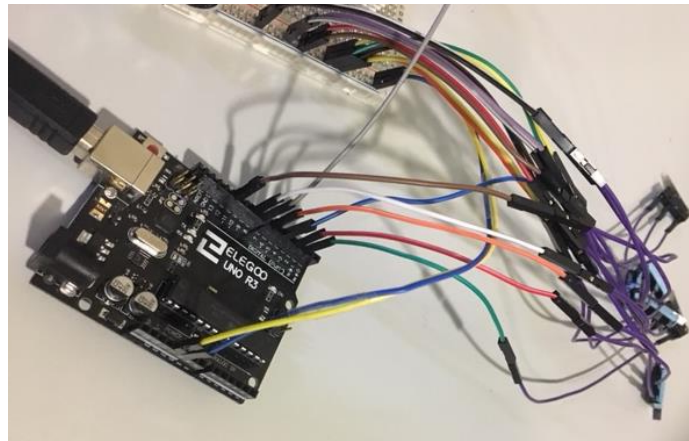


Figura 3-6 Estructura de Arduino con sensores de contacto (Fuente: elaboración propia)

Para que este Arduino pueda conectarse con el Arduino central necesita añadir una librería (ver Listado 3-1) que permite controlar el puerto serie y seleccionar dos pines digitales uno para transmitir datos y otro para recibir datos. En este caso se seleccionó el pin número 12 para recibir y el pin número 8 para transmitir.

```
#include <SoftwareSerial.h>

SoftwareSerial mySerial(12, 8);
```

Listado 3-1 El código de librería del puerto serie Arduino con sensores de contacto (Fuente: elaboración propia)

Dentro de la parte de configuración del código (*setup*) (ver Listado 3-2), se indica la velocidad del puerto serie para Arduino trabajando a 9600 bps (*Serial.begin(9600)*) y la velocidad del puerto serie que se establece para comunicar con el Arduino central igual a 9600 bps (*mySerial.begin(9600)*). Después se configuran los pines digitales (2,3,5,7,9,12) para poder utilizarlos como entradas que permiten recibir los datos de cada uno de los sensores de contacto, uno por cada caja de un armero.

```
Serial.begin(9600);
mySerial.begin(9600);
pinMode(2, INPUT);
pinMode(5, INPUT);
pinMode(7, INPUT);
pinMode(9, INPUT);
pinMode(3, INPUT);
pinMode(12, INPUT);
```

Listado 3-2 El código de la parte de configuración de Arduino con sensores de contacto (Fuente: elaboración propia)

Dentro del cuerpo principal del programa (*loop*) de este Arduino se recibe el dato de cada sensor de contacto con el formato de número tipo entero, es decir, que si una ranura está ocupada (tiene fusil) se indica con el numero “1” y en caso contrario se indica con el numero “0”. Cuando se han recibido todos los datos, se crea un array (*String*) en el que cada posición representa el número de cada contacto, Además se añaden a la cadena de texto indicadores que representan los límites inicial y final.

En este caso, el indicador inicial es “#” y el indicador final es “:”. Finaliza el programa transmitiendo el conjunto de datos hacia el Arduino central (lo que se muestra en Figura 3-7).

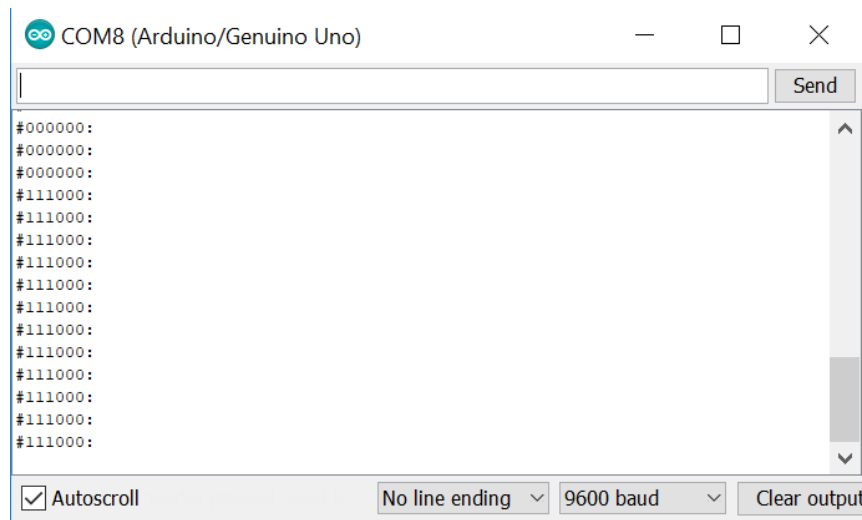


Figura 3-7 Monitor serie mostrando los datos leídos de los sensores de contacto (Fuente: elaboración propia)

- Arduino con sensores ambientales y de huella dactilar

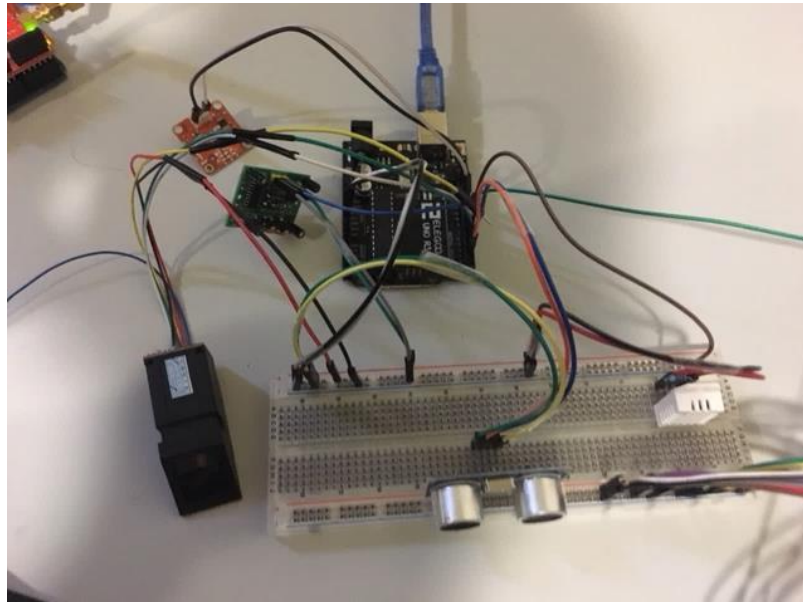


Figura 3-8 Estructura de Arduino con sensores ambientales (Fuente: elaboración propia)

Al igual que en el caso del Arduino que controla los sensores de contacto, es necesario conectarse al Arduino central utilizando comunicación serie y la librería específica (*Software Serial*). Además, se han seleccionado dos pines uno para transmitir los datos y otro para recibir los datos. En este caso se seleccionó el pin número 11 para recibir y el pin número 8 para transmitir. Además, es necesario añadir también las librerías de control de los sensores de temperatura/humedad (*DHT.h*), distancia (*Ultrasonic.h*) y de huella dactilar (*Adafruit_Fingerprint.h*). También es necesario configurar los pines que se utilizarán para comunicarse con los distintos sensores. En este caso se utilizan los pines número 4 y 5 para el sensor de óptico de huella dactilar, los pines número 2 y 3 para el sensor de distancia (los únicos con comunicación serie) tal como se puede ver en la Listado 3-3.

```
#include <SoftwareSerial.h>
#include <Adafruit_Fingerprint.h>
#include <Ultrasonic.h>
#include "DHT.h"
SoftwareSerial mySerial(4, 5);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
SoftwareSerial pinOne(11, 8);
Ultrasonic ultrasonic1(2, 3);
```

Listado 3-3 El código de librería del puerto serie Arduino con sensores ambientales (Fuente: elaboración propia)

Dentro de la parte de inicialización del código (*setup*) se indica la velocidad del puerto serie para Arduino trabajando a 9600 bps (*Serial.begin(9600)*), la velocidad del puerto serie que se establece para comunicar con el Arduino central igual 4800 bps (*pinOne.begin(4800)*) y la velocidad del puerto serie que se establece para comunicar con el sensor óptico de huella dactilar igual 57.600 bps (*finger.begin(57600)*). Después se indica que el pin número 7 se usará para recibir el dato sobre temperatura y humedad (*dht.setup(7)*), el pin número 12 para conectar con un LED, el pin analógico A0 para conectar con el sensor de nivel de iluminación y el pin número 6 para el sensor de presencia PIR(ver Listado 3-4).

```
Serial.begin(9600);
pinOne.begin(4800);
finger.begin(57600);
pinMode(6, INPUT);
pinMode(12, OUTPUT);
dht.setup(7);
```

Listado 3-4 El código de la parte de configuración de Arduino con sensores ambientales (Fuente: elaboración propia)

El programa principal (*loop*) recibe el dato de cada sensor y cuando se reciben todos los datos, crea un array (*String*) en el que cada posición representa un número generado a partir de los datos de los sensores. Al igual que en el caso del Arduino con sensores de contacto, se le añaden a la cadena dos indicadores de inicio “#” y de final “:”. Acontinuación el programa envía los datos hacia el Arduino central (ver Figura 3-9)

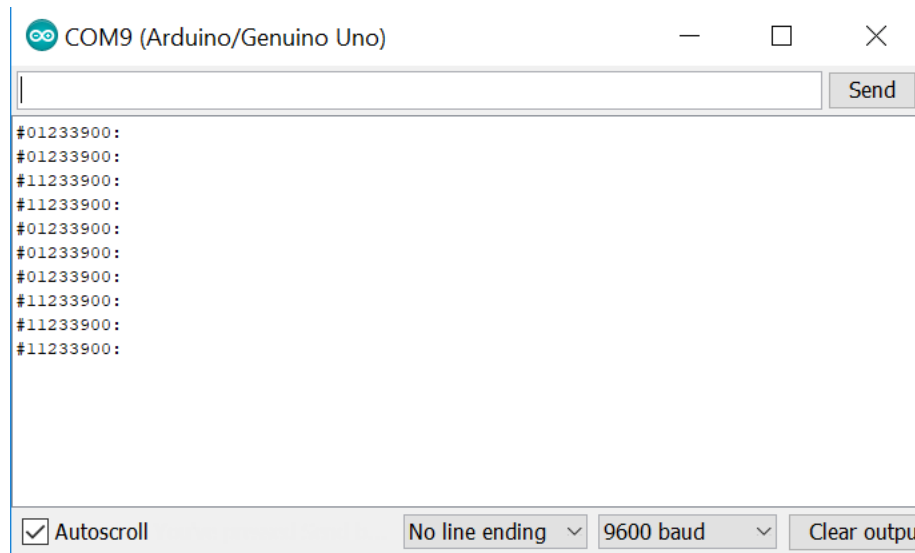


Figura 3-9 Monitor serie mostrando los datos leídos de los sensores ambientales (Fuente: elaboración propia)

Cada uno de los sensores trabaja como se especifica a continuación:

1. Sensor de presencia PIR: Si detecta cambios en la señal infrarroja el Arduino recibe ese cambio del sensor y crea el dato. La detección se representa con el número “1” y la no detección con el “0”.
2. Sensor de distancia: este sensor se utiliza para medir la distancia a la puerta. Es decir que cuando el sistema se enciende, el Arduino va a medir la distancia hasta la puerta y si durante la vigilancia encuentra que la distancia a la puerta cambia más de 4 centímetros, entonces el Arduino envía el dato número “1” y el número “0” si no detecta el cambio. Por ello, se necesita que, al iniciar el sistema de monitorización, la puerta tiene que estar cerrada.
3. Sensor DHT 22: el Arduino va recibir los datos sobre temperatura y humedad en el formato de número entero de dos dígitos.
4. Sensor de LDR: Si el valor de la resistencia medida es mayor que el valor de un umbral de referencia definido en base a la iluminación habitual del compartimiento entonces el Arduino crea el dato número “0” y en caso contrario envía el número “1”.
5. Sensor óptico de huella dactilar: cuando este sensor detecta una huella la compara con los datos almacenados. Si la huella del sensor es coherente con un resultado almacenado en su base de datos, con un valor de confianza mayor que 80, el resultado es un identificador asociado a la huella y si no encuentra la huella o el valor de confianza es más bajo, se envía un “0”.

- Arduino central

El Arduino central necesita la librería de control del puerto serie (*Software Serial*) y selecciona 4 pines digitales dos para transmitir los datos y otros dos para recibir los datos. En este caso se seleccionan los pines 3 y 4 para enviar y recibir datos del Arduino con los sensores de contacto y los pines 5 y 8 para el Arduino con sensores ambientales. Además, se necesita añadir también la librería de LoRa (*SPI.h*, *RH_RF95.h*) para comunicar con el Arduino del servidor (ver Listado 3-5).

```
#include <SPI.h>
#include <RH_RF95.h>
#include <SoftwareSerial.h>
SoftwareSerial portOne(3, 4);
SoftwareSerial portTwo(5, 8);
RH_RF95 rf95;
```

Listado 3-5 El código de librería del puerto serie Arduino central (Fuente: elaboración propia)

Dentro del código de inicialización (*setup*) se indica la velocidad del puerto serie para Arduino trabajando igual 9600 bps (*Serial.begin(9600)*), la velocidad del puerto serie que se establece para comunicar con los dos Arduinos igual 9600 bps y 4800 bps (*portOne.begin(9600)* y *portTwo.begin(4800)*). Además, se establece la potencia a 23 dBm (*rf95.setTxPower(23, false)*) y la frecuencia de LoRa a 868 MHz (*rf95.setFrequency(868.0)*) (ver Listado 3-6).

```
Serial.begin(9600);
portOne.begin(9600);
portTwo.begin(4800);
while (!Serial) ;
if (!rf95.init());
rf95.setTxPower(23, false);
rf95.setFrequency(868.0);
```

Listado 3-6 El código de la parte de configuración del Arduino central (Fuente: elaboración propia)

Dentro del bloque principal (*loop*) se reciben los datos de los dos Arduinos con sensores, se muestran todos los datos en una array (*string*) y se añaden los delimitadores inicial “#” y final “:”. La cadena resultante (ver Figura 3-10) se envía a través de LoRa al servidor.

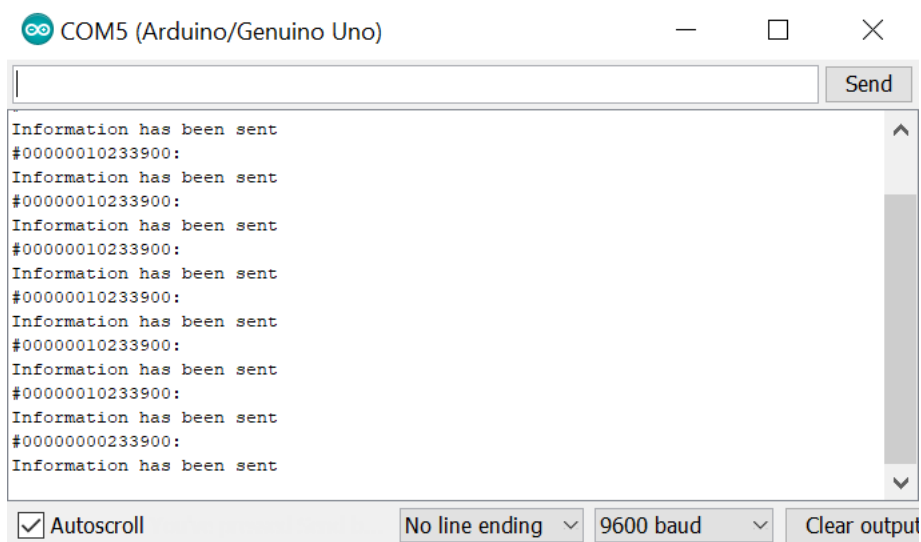


Figura 3-10 Monitor serie mostrando los datos leídos en el Arduino central (Fuente: elaboración propia)

3.2.2 Servidor

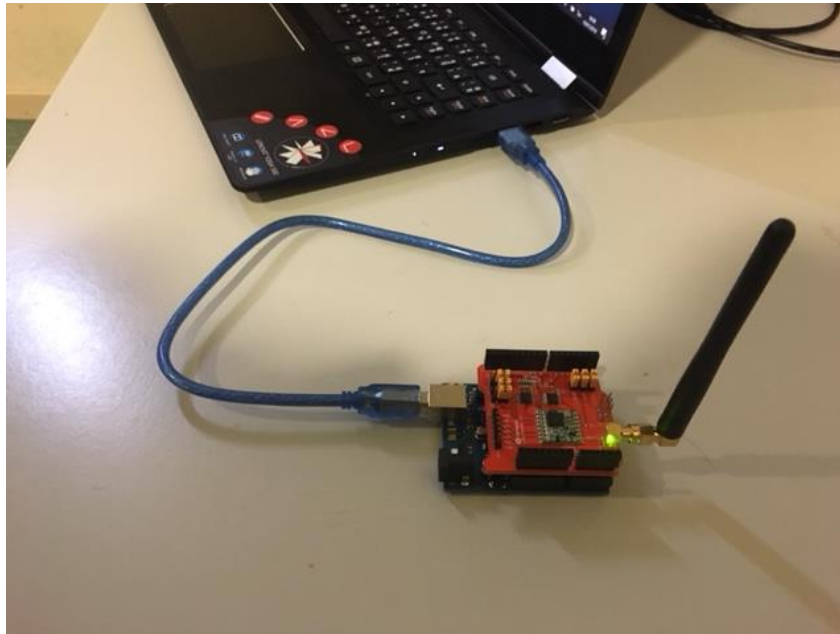


Figura 3-11 Estructura de servidor (Fuente: elaboración propia)

Necesita añadir una librería de LoRa (ver Listado 3-7) para poder comunicarse con el Arduino del observador.

```
#include <SPI.h>
#include <RH_RF95.h>
RH_RF95 rf95;
```

Listado 3-7 El código de librería de LoRa del servidor (Fuente: elaboración propia)

Como se indicó en la sección 2.5.8.1 un programa Arduino consta de dos partes. Una de configuración (*setup*) y otra que representa el bucle principal del programa (*loop*). Dentro de la parte de configuración (ver Listado 3-8) se llevan a cabo una serie de tareas: 1) se indica la velocidad del puerto serie para Arduino trabajando igual 9600 bps (*Serial.begin(9600)*); 2) se establece la potencia de transmisión de LoRa a 23 dBm (*rf95.setTxPower(23, false)*). 3) se establece la frecuencia de transmisión de LoRa a 868 MHz (*rf95.setFrequency(868.0)*).

```
Serial.begin(9600);
rf95.setTxPower(23, false);
rf95.setFrequency(868.0);
```

Listado 3-8 El código de la parte de configuración del servidor (Fuente: elaboración propia)

Dentro del bucle principal del programa se reciben los datos través de LoRa. Después de recibir los datos, se comprueba su validez y se envían al ordenador a través del puerto serie (ver Figura 3-12).

El servidor necesita comprobar todos los datos que le llegan desde el observador. En este trabajo el Arduino del servidor comprueba el tamaño de los datos usando los indicadores inicial y final. Luego se asegura de que todos los datos que han llegado son números. Si se cumplen los dos criterios, pasa

todos los datos a la aplicación de interfaz gráfica de usuario y en caso de no se cumplan los criterios, se descartan los datos y se espera la llegada de nuevos datos.

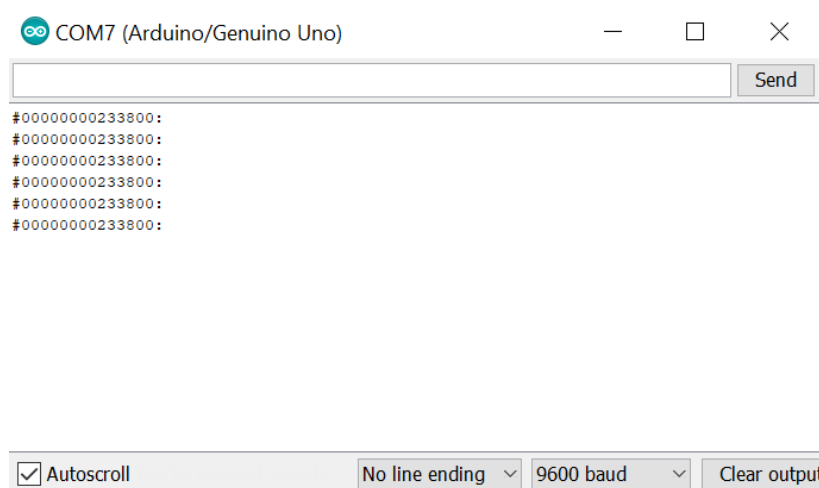


Figura 3-12 El monitor serie para visualizar el envío de datos al PC (Fuente: elaboración propia)

3.3 Interfaz Gráfica de Usuario

Para presentar la información recogida por el sistema de monitorización y control de inventario se ha desarrollado una aplicación de interfaz gráfica de usuario. El programa une dos fuentes de datos los recabados por los Arduinos y una tabla de identificación de huella de personas autorizadas que asocia los nombres de personas al identificador asociado a su huella en el lector. El programa presenta la situación y se guardan datos si se detecta una situación de interés incluidos los datos de temperatura, humedad y hora entrada del personal con autorización (ver Figura 3-13).

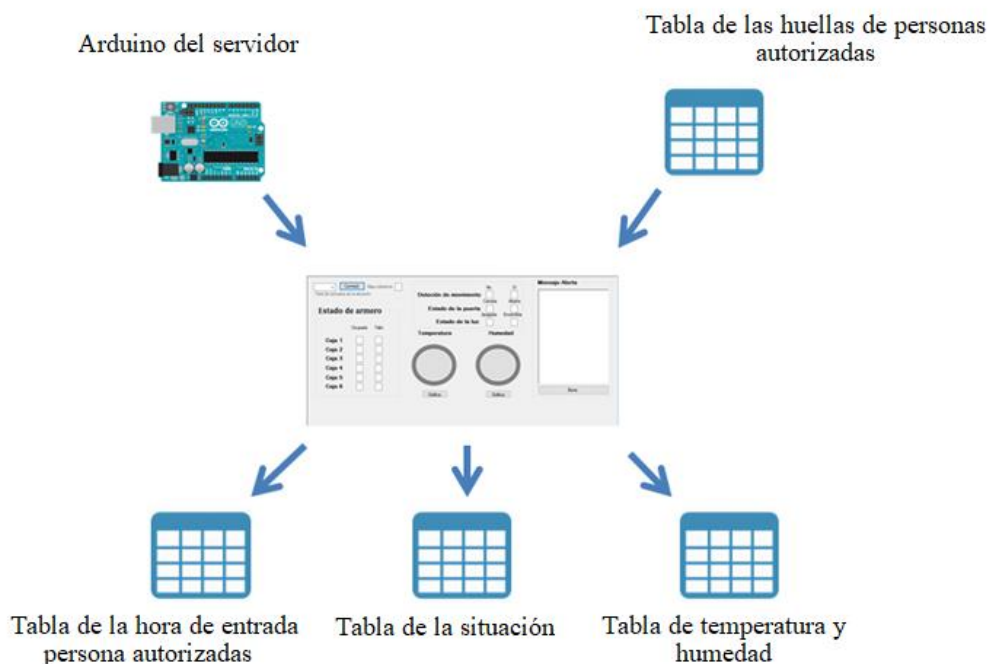


Figura 3-13 Arquitectura de la aplicación (Fuente: elaboración propia)

3.4 Ventana principal del programa

Esta ventana (Figura 3-14) se muestra 4 menús principales (Inicial, Tabla de situación, Tabla de Temperatura/Humedad y Tabla de accesos al pañol). A continuación, se van a explicar con más detalle las ventanas que se muestran al acceder a cada uno de estos menús.



Figura 3-14 Ventana del programa general (Fuente: elaboración propia)

3.4.1 Inicial



Figura 3-15 Ventana del programa inicial (Fuente: elaboración propia)

Esta parte presenta la situación general e indica la hora de actualización de los datos (Figura 3-15) se incluye también un pequeño cuadro (“Baja cobertura”) de forma que si durante 14 segundos no se reciben datos el sistema asume que hay problemas de comunicación y rellena el cuadro en rojo, y en verde si esto no ocurre. Para comenzar la vigilancia es necesario seleccionar el puerto serie que conecta al equipo con el Arduino del servidor. Esta ventana se divide en cuatro zonas:

- Estado del armero



Figura 3-16 Ventana de estado armero (Fuente: elaboración propia)

Esta parte presenta la situación de cada caja del armero (Figura 3-16) si el cuadrado se muestra en color verde significa que el sensor de contacto está activo y se encuentra el HK en la caja, por el contrario, si se presenta en color rojo es que no se encuentra el HK.

- Temperatura

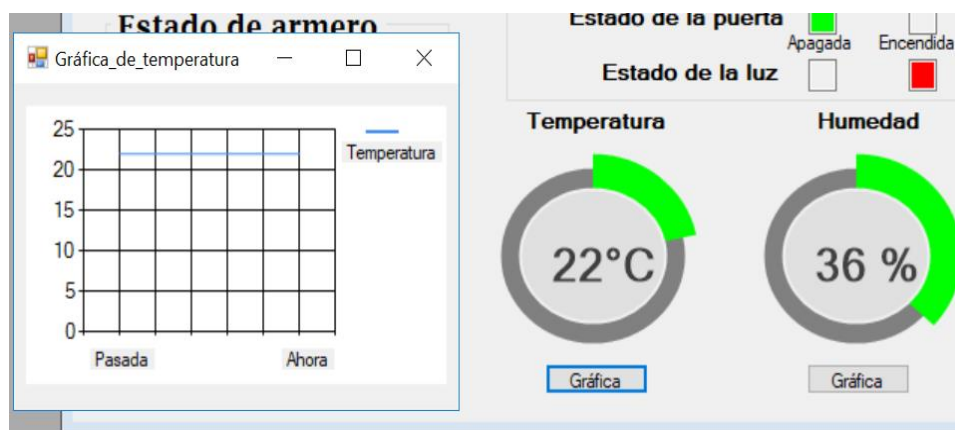


Figura 3-17 Ventana de temperatura (Fuente: elaboración propia)

Esta parte presenta la temperatura en el pañol. Los colores de la gráfica se varían con la temperatura (ver Figura 3-17). En este caso, según el Manual Interejércitos de Almacenamiento de Municiones y Explosivos para las Fuerzas Armadas Españolas y tal como se indicó en la sección 1.3.2, la temperatura deberá mantenerse comprendida entre 10 °C y 35 °C entonces:

- Temperatura $\leq 10^{\circ}\text{C}$ o $\geq 35^{\circ}\text{C}$: la gráfica presenta color rojo lo que significa que se está en una situación peligrosa para el estado de la munición.
- Temperatura entre 10°C y 15°C o entre 30°C y 35°C : la gráfica presenta color amarillo lo que significa que se está en una situación cerca del límite de temperatura debiendo estar atentos a la situación.
- Temperatura entre 15°C y 30°C : la gráfica presenta color verde lo que significa que se está en una situación óptima.

Además, pulsando el botón “Gráfica” de la interfaz se muestra una representación que indica como varía la temperatura con el tiempo en el último minuto.

- Humedad



Figura 3-18 Ventana de humedad (Fuente: elaboración propia)

Esta parte presenta la humedad en el pañol y funciona de manera similar a la de temperatura. Los colores de la gráfica varían con el porcentaje de humedad (ver Figura 3-18). En este caso, según el Manual Interejércitos de Almacenamiento de Municiones y Explosivos para las Fuerzas Armadas Españolas y tal como se indicó en la sección 1.3.3, los pañoles deben mantenerse relativamente secos y deben tener una humedad relativa que no supere el 70% entonces:

- Humedad $\geq 70\%$: la gráfica presenta color rojo para señalar la situación de potencial peligro para la munición.
- Humedad entre 60% y 70%: la gráfica presenta color amarillo, indicando que se está cerca del límite de humedad y se debe estar atento a la situación.
- Humedad $< 60\%$: la gráfica presenta color verde, lo que significa que la situación es adecuada.

Además, debajo de la gráfica hay un botón “Gráfica” en el que, al pulsar, se muestra una representación gráfica que indica cómo varía la humedad con el tiempo, durante el último minuto.

- Situación de movimiento, puerta y luz



Figura 3-19 Ventana de situación de luz, puerta y movimiento (Fuente: elaboración propia)

Esta parte presenta la situación del pañol sobre tres aspectos: 1) si la puerta está abierta (rojo) o cerrada (verde); 2) si la luz está apagada (verde) o encendida (rojo) y 3) si se detecta movimiento, en este caso si detecta movimiento presenta color rojo y si no se detecta presenta color verde.

- Mensajes de alerta

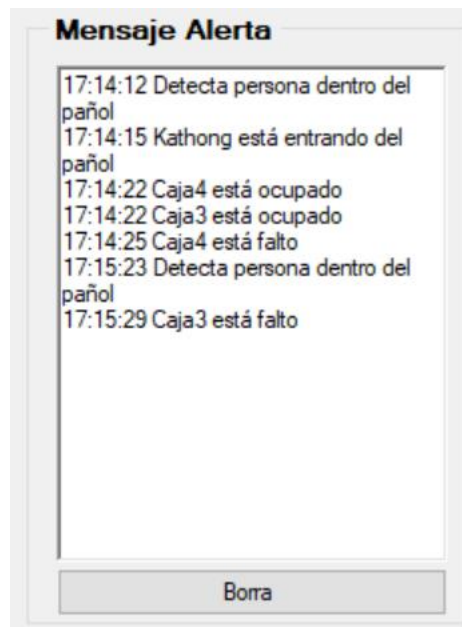


Figura 3-20 Ventana de mensaje alerta (Fuente: elaboración propia)

Esta parte presenta los mensajes que indican todos los eventos interesantes para el operador junto con la hora de cada uno.

3.4.2 Tabla de situación

	ID	Día	Hora	Caja 1	Caja 2	Caja 3
▶	1	23/01/2018	17:13:09	Falto	Ocupado	Falto
	2	23/01/2018	17:13:17	Falto	Falto	Falto
	3	23/01/2018	17:13:40	Falto	Falto	Falto
	4	23/01/2018	17:13:52	Falto	Falto	Falto
	5	23/01/2018	17:13:58	Falto	Falto	Falto
	6	23/01/2018	17:14:43	Falto	Falto	Falto
	7	23/01/2018	17:15:18	Falto	Falto	Falto

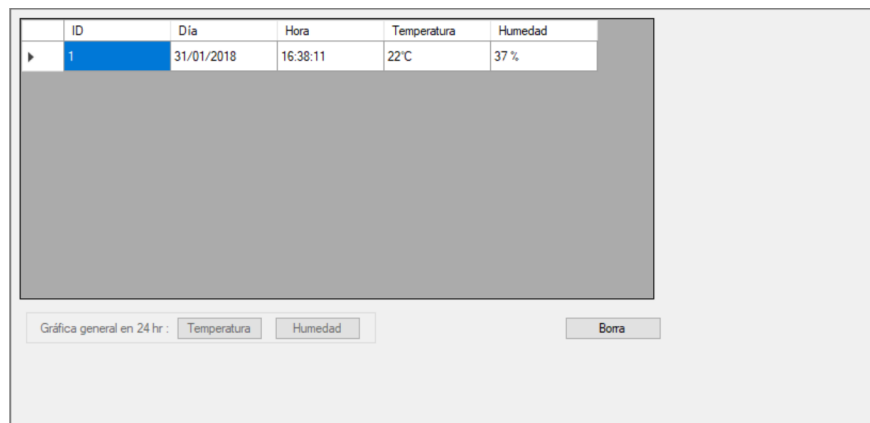
< >

Borra

Figura 3-21 Ventana de tabla de situación (Fuente: elaboración propia)

Esta tabla (Figura 3-21) almacena información sobre la situación del pañol. Cuando el programa detecta una situación de potencial interés por ejemplo cambio de estado de la puerta, detección de movimiento o si se detecta el cambio de estado de una caja del armero, se registra el cambio de situación en esta tabla.

3.4.3 Tabla de Temperatura/Humedad (T/H)



ID	Día	Hora	Temperatura	Humedad
1	31/01/2018	16:38:11	22°C	37 %

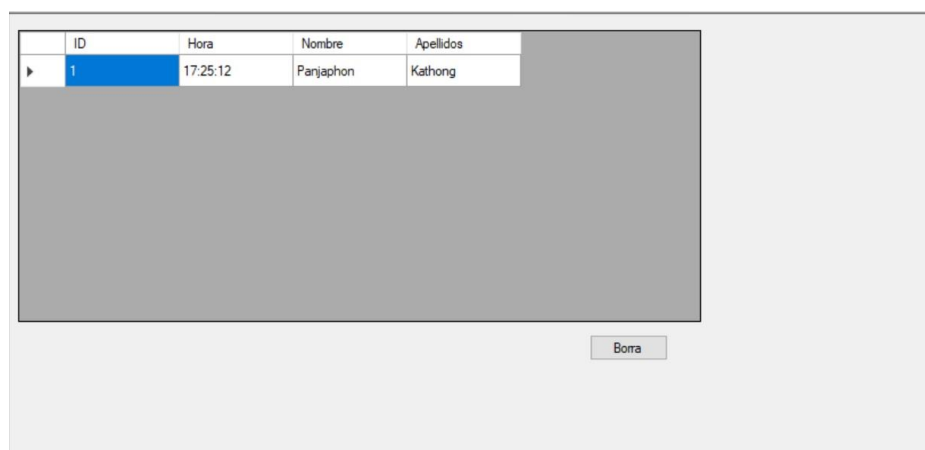
Gráfica general en 24 hr :

Figura 3-22 Ventana de tabla de Temperatura/Humedad (Fuente: elaboración propia)

Esta tabla (Figura 3-22) presenta el resumen de temperatura y humedad por cada hora de trabajo. Es decir que cada 60 minutos los datos de temperatura y humedad se guardan automáticamente en la tabla.

Además, debajo de la tabla hay dos botones que permiten representar la gráfica general de temperatura y humedad de las últimas 24 horas.

3.4.4 Tabla de accesos al pañol



ID	Hora	Nombre	Apellidos
1	17:25:12	Panjaphon	Kathong

Figura 3-23 Ventana de tabla de persona entrada (Fuente: elaboración propia)

Cuando el sensor de huella dactilar envía el número de identificación de una huella el programa compara este número con la tabla de identificación de huella y después guarda la hora, nombre y apellido de esta persona en la tabla de accesos al pañol (Figura 2-23).

4 PRUEBA Y VALIDACIÓN

En este capítulo se detallan algunas pruebas que se llevaron a cabo, para comprobar que el funcionamiento del sistema y sus componentes era correcto, y verificar la viabilidad de utilizar el sistema implementado en un entorno realista, como es el pañol del condestable de la ENM.

4.1 Distancia máxima del sensor HC-SR04

El objetivo de esta prueba es determinar la distancia máxima operativa del sensor HC-SR04 en la situación en la que el sensor HC-SR04 se conecta con Arduino junto con el resto de sensores y se utiliza un cargador para alimentar el Arduino.

Como se ha explicado anteriormente, según la hoja de características, el sensor HC-SR04 puede determinar la distancia de un objeto en un rango de 2 a 450 cm. Pero en la práctica la capacidad se reduce debido a que el Arduino tiene que conectarse con varios sensores y la alimentación se hace a través de un cargador que se conecta con tres Arduinos.

Por tanto, lo que se realizó para determinar la distancia máxima a la que el sensor HC-SR04 es capaz de medir sin introducir mucho error fue un análisis basado en la observación de los valores de distancia del sensor comparados con los valores de distancia obtenidos de una cinta métrica cada 10 cm. La medición del sensor se hace como la media de 10 mediciones. Los resultados se pueden observar en la Figura 4-1

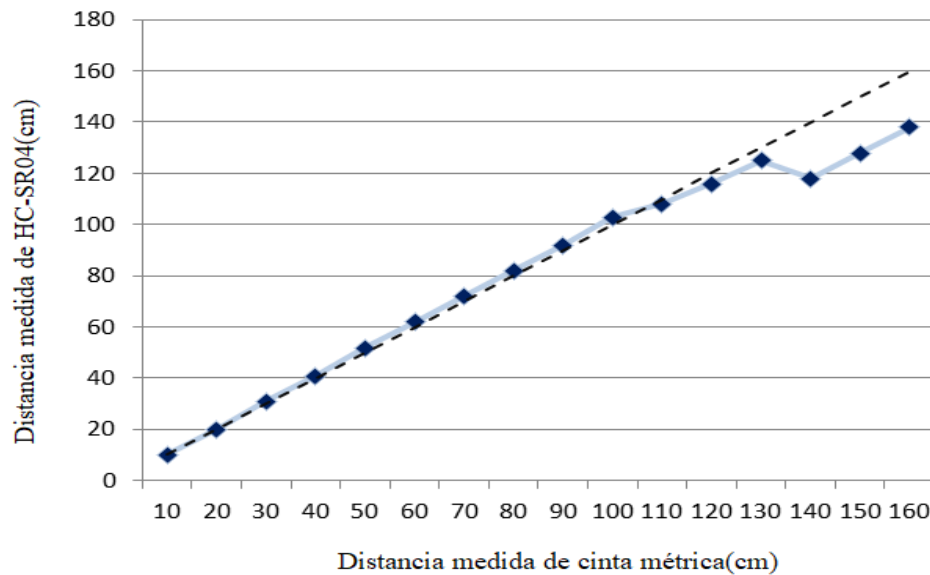


Figura 4-1 Grafica de la prueba de distancia del sensor HC-SR04 (Fuente: elaboración propia)

Vistos los resultados, se estima que el sensor puede determinar con precisión la distancia máxima a unos 130 cm. A partir de esta distancia el error tiene un valor mayor que 10 cm. Es decir que el sensor debe estar situado a una distancia menor que 130 cm desde la puerta para ser capaz de determinar el estado de la misma sin introducir errores.

4.2 Prueba del sensor de huella dactilares



Figura 4-2 Prueba de sensor de huella dactilar (Fuente: elaboración propia)

El objetivo de esta prueba es determinar el valor de confianza mínimo para que el sensor pueda comparar la huella que se ponga encima con las que hay guardadas en su memoria *flash* e informarnos con un número de identificación asociado por el puerto serie (Figura 4-3).

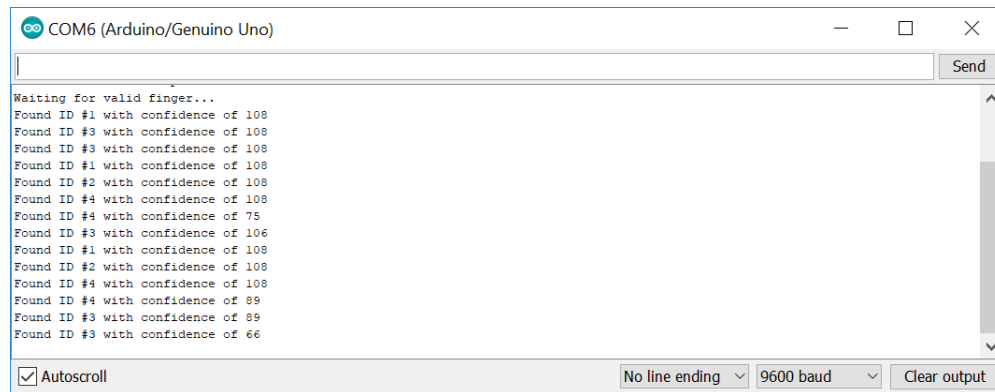


Figura 4-3 Monitor serie en la prueba de sensor de huella dactilar (Fuente: elaboración propia)

Por tanto, lo que se realizó para determinar el valor de confianza mínimo fue un análisis estadístico basado en la observación del porcentaje de error en cada valor de confianza. Para llevar a cabo este análisis, nos basamos en la observación de los errores que se produjeron en cada valor de confianza. En este caso, indicamos un valor de confianza y se comprueba con 4 personas si el sistema acierta o no, probando con cada persona 25 veces (lo que hace un total de 100 muestras). Cuando termina una prueba se aumenta el valor del umbral de confianza en 10 unidades y se repite.

Se obtuvieron los siguientes resultados, representados en la gráfica de la Figura 4-4.

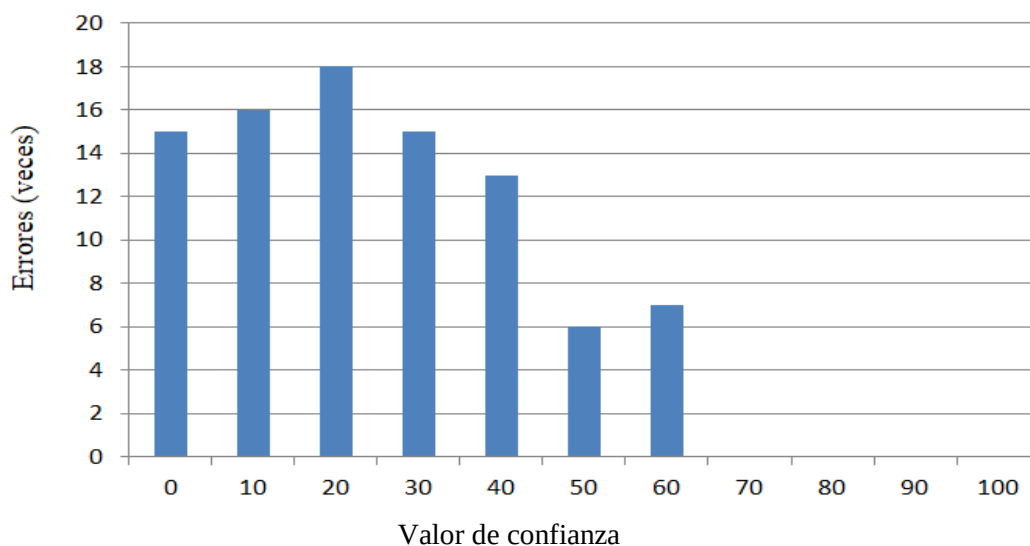


Figura 4-4 Grafica de la prueba de sensor de la huella dactilar (Fuente: elaboración propia)

En la mayoría de los intentos que se realizaron, el sensor fue capaz de determinar un número de identificación correcto en todas veces cuando el valor de confianza estaba entre 70 – 100. Por el contrario, si el valor designado era mayor o igual de 110 el sensor no era capaz de determinar un número de identificación, y cuando era menor que 70, se producían errores, tal como se ve en la Figura 4-4

4.3 Prueba de alcance de LoRa

El objetivo de esta prueba es determinar el alcance máximo de LoRa en el caso en el que una placa de Arduino con LoRa estaba en el pañol del condestable y otro Arduino con LoRa recibía datos a distinta distancia (CUD, Isaac Peral y Muelle de Torpedos).

Por tanto, lo que se realizó para determinar el alcance máximo al que LoRa es capaz de transmitir y recibir datos fue un análisis basado en la observación de la recepción de datos: mediante un temporizador se establece que, si durante 14 s no se reciben datos, hay un problema de falta de cobertura.

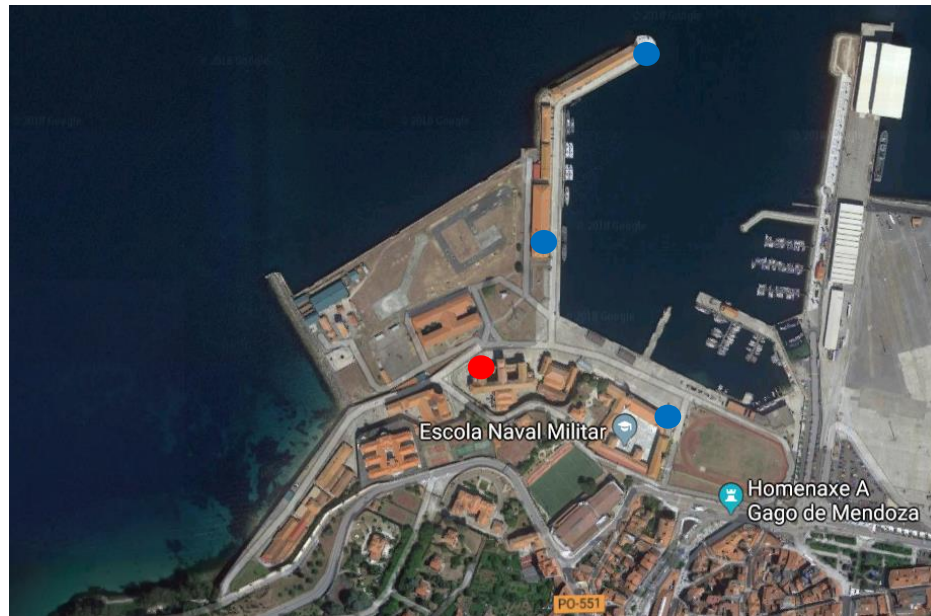


Figura 4-5 Prueba de alcance de LoRa (Fuente: elaboración propia)

El resultado de esta prueba fue que LoRa tiene alcance de cobertura en todas las partes de la Escuela si la antena de LoRa del servidor se sitúa en el exterior de los edificios, pero en algún aula del edificio Isaac Peral y el CUD se encontraron problema de cobertura de la señal.

4.4 Prueba del funcionamiento del sistema

Lo primero que se hace una vez se finaliza el prototipo es ir al pañol del condestable para ponerlo a prueba. El objeto de esta prueba era principalmente comprobar que se establecía una correcta conexión entre los Arduinos y comprobar el funcionamiento del sistema. Para llevar a cabo esta prueba, se llevó a cabo la siguiente instalación:

- Sensor HC-SR04: Se instaló al lado de la parte interior de la puerta del pañol.
- Los sensores de contacto y el sensor LDR: Se instalaron en la parte superior de la caja del armero. Además, el sensor LDR se dirigió hacia la fuente de luz (un tubo fluorescente).
- Sensor de huella dactilar, DHT 22 y PIR: Se instalaron en la mesa al lado de la puerta. El sensor PIR se apuntó en la dirección del pasillo de acceso al pañol.

Una vez realizó la instalación se llevaron a cabo las siguientes pruebas:

- Abrir/cerrar la puerta.
- Moverse por el pasillo de acceso al pañol.
- Encender/apagar la luz.
- Quitar/poner fusiles HK en sus cajas del armero.
- Verificar una huella de un usuario previamente registrado.

La captura de pantalla de la ผิดพลาด! ไม่พบแหล่งการอ้างอิง se corresponde con la primera prueba realizada. Con esta comprobación se demuestra que el sistema es capaz de cumplir con las funciones para las que se diseñó.

5 CONCLUSIONES Y LÍNEAS FUTURAS

En este capítulo se presentan las principales conclusiones obtenidas y se explica el nivel de alcance de los objetivos establecidos al inicio del TFG. Se presentan, asimismo, las posibles líneas futuras en las que se podría seguir trabajando para mejorar y ampliar el TFG presentado.

5.1 Conclusiones

Llegados al final del proyecto, es necesario realizar un estudio de los objetivos logrados. Se puede decir que los objetivos básicos descritos en el apartado 1.4 se han logrado. 1) Monitorización de las variables ambientales del pañol: la aplicación del trabajo tiene la capacidad de obtener la temperatura y humedad del pañol, permite representar la gráfica general de temperatura y humedad de las últimas 24 horas. 2) Control de inventario de los fusiles HK, controlando los armeros para detectar la presencia o ausencia de cada fusil en su caja, como se indicó en la sección 3.4.1 el programa puede representar la situación de cada caja del armero y se presenta con dos colores (verde y rojo). 3) Monitorización y detección de accesos al pañol y, en caso de que sean accesos autorizados, registro de personas que han accedido, el programa puede presentar la hora, nombre y apellido de la persona en una tabla. 4) Uso de la tecnología de comunicación inalámbrica LoRa para transmitir la información a distancia: según la prueba de 4.3, este sistema puede transmitir su información a casi cualquier parte de la Escuela. 5) Presentación de la información en una interfaz gráfica: como se muestra en la sección 3.3, se ha desarrollado una aplicación de interfaz gráfica de usuario para presentar la información recogida por el sistema de monitorización y control del pañol.

En cuanto al objetivo académico del proyecto, se puede afirmar que se trata de un proyecto que abarca varios campos estudiados durante la carrera en asignaturas como Tecnología Electrónica, donde se aprende a analizar el funcionamiento de circuitos electrónicos, Informática y Ampliación de informática, donde se introduce al alumno en el ámbito de la programación, las redes y su funcionamiento, y asignaturas militares como Sistemas de armas y tiro naval y Comunicaciones.

Además de estos objetivos, se ha conseguido profundizar en una tecnología con gran potencial de futuro, como es LoRa/LoRaWAN.

5.2 Líneas Futuras

El presente TFG puede ser utilizado como base para posibles proyectos futuros. Ya que el sistema cuenta con muchas mejoras o modificaciones posibles. Algunas de ellas podrían ser:

- Mejorar el mecanismo de transmisión mediante el uso de cifrado en LoRa.

- Probar arquitecturas más complejas con más nodos LoRa y usando LoRaWAN.
- Añadir otros sensores como por ejemplo una cámara
- Utilizar alguna tecnología inalámbrica (Bluetooth, ZigBee) para comunicar entre sí los Arduinos del sistema de monitorización. Esto permitirá distribuirlos por todo el pañol actuando como una red de sensores.

6 BIBLIOGRAFÍA

- [1] Estado mayor de la Armada, Reglamento de pólvoras, explosivos y municiones para buques, fuerza y dependencias, 2002.
- [2] Centro Nacional de Inteligencia, NS-02. Seguridad en el Personal. Habilitación de Seguridad del Personal, 2012.
- [3] «Web de Gm3s,» 23 Febrero 2018. [En línea]. Available: <http://www.gm3s.com.mx/>.
- [4] «Web de Cisco,» [En línea]. Available: <https://www.cisco.com>. [Último acceso: 12 Enero 2018].
- [5] «Web de LoRa,» [En línea]. Available: <https://www.lora-alliance.org>. [Último acceso: 15 Enero 2018].
- [6] «Web de Semtech,» [En línea]. Available: <https://www.semtech.com>. [Último acceso: 23 Enero 2018].
- [7] «Web de Boletín Oficial del Estado,» 2013. [En línea]. Available: <https://www.boe.es/buscar/act.php?id=BOE-A-2013-4845>. [Último acceso: 22 Enero 2018].
- [8] «Web de LoRaWAN,» [En línea]. Available: <https://www.thethingsnetwork.org/wiki/LoRaWAN/Home>. [Último acceso: 15 Enero 2018].
- [9] «Web de Comunicaciones inalámbricas,» [En línea]. Available: <https://www.comunicacionesinalambricashoy.com>. [Último acceso: 10 Enero 2018].
- [10] «Web de Sensors de LoRa Shield,» [En línea]. Available: http://wiki.dragino.com/index.php?title=Lora_Shield. [Último acceso: 2 Enero 2018].
- [11] «Web de Dragino,» [En línea]. Available: <http://www.dragino.com>. [Último acceso: 12 enero 2018].
- [12] «Web de Sparkfun,» [En línea]. Available: <https://www.sparkfun.com>. [Último acceso: 18 Enero 2018].
- [13] «Web de i-scoop,» [En línea]. Available: <https://www.i-scoop.eu>. [Último acceso: 12 Enero 2018].

- [14] «Web de Digikey,» [En línea]. Available: <https://www.digikey.es>. [Último acceso: 22 Enero 2018].
- [15] «Web de Arduino,» [En línea]. Available: <https://www.arduino.cc/es>. [Último acceso: 2018 Enero 10].
- [16] Southeast Norway universidad, «Libro de Introduction to Visual C#,» 2016.
- [17] «Web de Adafruit,» [En línea]. Available: <https://www.adafruit.com>. [Último acceso: 14 Enero 2018].
- [18] «Web de Create.Arduino,» [En línea]. Available: <https://create.arduino.cc>. [Último acceso: 14 Enero 2018].
- [19] «Web de Github,» [En línea]. Available: <https://www.github.com>. [Último acceso: 26 Enero 2018].
- [20] «Web de Google map,» [En línea]. Available: <https://www.google.es/map>. [Último acceso: 22 Enero 2018].
- [21] «Web de Microsoft Office,» [En línea]. Available: <https://www.microsoft.com>. [Último acceso: 2 Enero 2018].
- [22] «Web de Visual Studio,» [En línea]. Available: <https://www.visualstudio.com>. [Último acceso: 3 Enero 2018].
- [23] «Web de UPI,» [En línea]. Available: <https://www.upi.com>. [Último acceso: 12 Enero 2018].
- [24] «Web de NSPA,» [En línea]. Available: <https://www.nspa.nato.int>. [Último acceso: 16 Enero 2018].
- [25] «Web de Librerías de Arduino,» [En línea]. Available: <https://www.arduino-libraries.info>. [Último acceso: 28 Enero 2018].
- [26] «Web de IDE de Arduino,» [En línea]. Available: <https://www.arduino.cc/es/Main/software>. [Último acceso: 3 Enero 2018].