



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

# Sistema de seguridad perimetral programable inteligente

Proyecto Final de Carrera

**Ingeniero en Informática**

**Autor:** Jesús Brun Conejos

**Director:** Jose Luis Poza Luján

Septiembre de 2014



# Resumen

---

Tras el auge de los sistemas de seguridad se ha decidido desarrollar un sistema de seguridad perimetral empleando tecnología de bajo coste como es el microcontrolador Arduino.

En este documento explica cómo ha sido desarrollado el sistema desde un primer análisis del estado actual, pasando por la especificación y el diseño del sistema hasta llegar a la implementación del mismo. Se busca recorrer todo el proceso llevado a cabo para desarrollar el sistema, el cual puede ser aplicado a muchos otros proyectos.

El producto final permite poder disponer de un sistema de seguridad instalable en multitud de lugares que ofrece una muy buena relación prestaciones/costes.

**Palabras clave:** arduino, shield ethernet, pir, seguridad, alarma.



# Abstract

---

After the rise of security systems has decided to develop a perimeter security system using low cost technology such as Arduino microcontroller.

This document explains how the system has been developed from an initial analysis of the current state, through the specification and design of the system up to the implementation. It seeks to explore the whole process just born to develop the system. This process can be applied to many other projects.

The final product can be provided with a security system installable in many places that offer a good performance / cost.

**Palabras clave:** arduino, shield ethernet, pir, security, alarm.

# Agradecimientos

---

Quería agradecer la ayuda al director del proyecto, José Luis Poza Luján, por haberme dado la oportunidad de realizar este proyecto y por la ayuda recibida durante todo este tiempo.

Por supuesto también quiero agradecer a mi familia los esfuerzos y el apoyo recibido durante todos estos años que me han servido para poder lograr los objetivos.



# Tabla de contenidos

---

1.	Introducción.....	13
1.1.	Motivación .....	13
1.2.	Objetivos .....	13
2.	Estado actual y conceptos teóricos.....	14
2.1.	Introducción .....	14
2.2.	Teoría de la seguridad.....	14
2.2.1.	Conceptos .....	14
2.2.2.	Instalaciones de seguridad .....	17
2.3.	Sistemas similares .....	21
2.3.1.	Funciódomo .....	21
2.3.2.	Powermax Pro .....	22
2.3.3.	MARD.....	24
2.3.4.	CEBEK AL-1.....	25
2.4.	Análisis sistemas similares .....	27
2.4.1.	Cuantitativo .....	27
2.4.2.	Cualitativo .....	27
2.5.	Tecnología.....	29
2.5.1.	Arduino.....	29
2.5.2.	Sensores y actuadores .....	33
2.6.	Conclusiones .....	35
3.	Especificación de requisitos .....	36
3.1.	Introducción .....	36
3.1.1.	Propósito .....	36
3.1.2.	Alcance.....	36
3.1.3.	Personal involucrado.....	37
3.1.4.	Definiciones, acrónimos y abreviaturas .....	38
3.1.5.	Resumen .....	39
3.2.	Descripción general .....	40
3.2.1.	Perspectiva del producto .....	40
3.2.2.	Funcionalidad del producto .....	40



3.2.3.	Características de los usuarios .....	40
3.2.4.	Restricciones .....	40
3.2.5.	Suposiciones y dependencias .....	40
3.2.6.	Evolución previsible del sistema .....	41
3.3.	Requisitos específicos .....	41
3.3.1.	Requisitos comunes de los interfaces.....	41
3.3.2.	Requisitos funcionales.....	42
3.3.3.	Requisitos no funcionales.....	47
4.	Diseño del sistema .....	49
4.1.	Introducción .....	49
4.2.	Descripción conceptual.....	49
4.3.	Diagrama hardware .....	50
4.4.	Especificación software .....	51
4.4.1.	Detección intrusión en el perímetro.....	51
4.4.2.	Consulta del estado del sistema.....	52
4.4.3.	Consulta del historial de eventos del sistema.....	53
4.4.4.	Activar/Desactivar el sistema de seguridad .....	53
4.4.5.	Modificación de la IP del servidor.....	54
4.4.6.	Modificación de la IP del sistema Arduino.....	54
4.4.7.	Gestión de los sensores del sistema.....	55
4.4.8.	Gestión de los actuadores del sistema.....	55
4.4.9.	Gestión de las acciones del sistema.....	56
4.4.10.	Paro de la alarma.....	56
4.5.	Diagrama UML del Arduino .....	57
4.6.	Conclusiones .....	58
5.	Implementación .....	59
5.1.	Introducción .....	59
5.2.	Implementación.....	59
5.2.1.	Base de datos .....	59
5.2.2.	Programación del Arduino .....	61
5.2.3.	Aplicación Web.....	64
5.2.4.	Servidor Web .....	73
5.3.	Prototipo .....	75
5.4.	Conclusiones .....	76
6.	Conclusiones .....	78
6.1.	Trabajo realizado .....	78



## Sistema de seguridad perimetral programable inteligente

6.2. Aportaciones .....	78
6.3. Ampliaciones .....	79
Referencias .....	80





# Ilustraciones

---

Ilustración 1. División de la seguridad realizada en el proyecto. ....	14
Ilustración 2. Esquema instalación de seguridad 1 .....	18
Ilustración 3. Esquema instalación de seguridad 2.....	19
Ilustración 4. Esquema instalación de seguridad 3.....	20
Ilustración 5. Esquema del sistema funciodomo .....	22
Ilustración 6. Central Powermax Pro .....	23
Ilustración 7. Interior del sistema MARD .....	24
Ilustración 8. Parte frontal y trasera del sistema MARD .....	25
Ilustración 9. Central de alarma CEBEK AL-1 .....	25
Ilustración 10. Esquema de instalación del CEBEK AL-1 .....	26
Ilustración 11. Vista frontal del Arduino UNO .....	31
Ilustración 12. Ethernet Shield.....	32
Ilustración 13. Sensor de movimiento PIR.....	33
Ilustración 14. Barrera de infrarrojos.....	34
Ilustración 15. Sensor de ultrasonidos .....	35
Ilustración 16. Módulos del sistema ARDUENTORNO .....	36
Ilustración 17. Ubicación del proyecto en el sistema Arduentorno.....	37
Ilustración 18. Caso de uso detección de intrusión en el perímetro .....	42
Ilustración 19. Caso de uso consulta del estado del sistema .....	43
Ilustración 20. Caso de uso consulta del historial de eventos del sistema.....	43
Ilustración 21. Caso de uso activar/desactivar el sistema de seguridad .....	44
Ilustración 22. Caso de uso modificación de la IP del servidor .....	44
Ilustración 23. Caso de uso modificación de la IP del sistema Arduino .....	45
Ilustración 24. Caso de uso gestión de los sensores del sistema.....	45
Ilustración 25. Caso de uso gestión de los actuadores del sistema .....	46
Ilustración 26. Caso de uso gestión de las acciones del sistema .....	46
Ilustración 27. Caso de uso paro de la alarma.....	47
Ilustración 28. Descripción conceptual del sistema diseñado .....	49
Ilustración 29. Esquema hardware del sistema .....	50
Ilustración 30. Esquema eléctrico del sistema diseñado .....	51
Ilustración 31. Diagrama de secuencia. Detección intrusión en el perímetro .....	52
Ilustración 32. Diagrama de secuencia. Consulta del estado del sistema.....	52
Ilustración 33. Diagrama de secuencia. Consulta del historial de eventos del sistema. 53	
Ilustración 34. Diagrama de secuencia. Activar/Desactivar el sistema de seguridad ...	53
Ilustración 35. Diagrama de secuencia. Modificación de la IP del servidor. ....	54
Ilustración 36. Diagrama de secuencia. Modificación de la IP del sistema Arduino.....	54
Ilustración 37. Diagrama de secuencia. Gestión de los sensores del sistema. ....	55
Ilustración 38. Diagrama de secuencia. Gestión de los actuadores del sistema. ....	55
Ilustración 39. Diagrama de secuencia. Gestión de las acciones del sistema. ....	56
Ilustración 40. Diagrama de secuencia. Paro de la alarma. ....	56
Ilustración 41. Diagrama de flujo del Arduino .....	57
Ilustración 42. Esquema de las tablas que componen la BBDD .....	60
Ilustración 43. Interfaz de la aplicación. Estado de la alarma. ....	65
Ilustración 44. Interfaz de la aplicación. Alarma activada.....	66



Ilustración 45. Interfaz de la aplicación registro de eventos.....	67
Ilustración 46. Interfaz aplicación. Configuración de las IP's .....	68
Ilustración 47. Interfaz aplicación. Configurar entradas .....	69
Ilustración 48. Interfaz aplicación. Configurar nueva salida.....	70
Ilustración 49. Interfaz aplicación. Modificar salida. ....	70
Ilustración 50. Interfaz aplicación. Configurar salidas. ....	71
Ilustración 51. Interfaz aplicación. Configurar acciones. ....	71
Ilustración 52. Interfaz aplicación. Configurar acciones desactualizadas. ....	72
Ilustración 53. Prototipo empleado en el desarrollo y pruebas .....	76

# Tablas

---

Tabla 1. Análisis cuantitativo de los sistemas similares.....	27
Tabla 2. Análisis cualitativo de los sistemas similares desde el punto de vista del usuario.....	28
Tabla 3. Análisis cualitativo de los sistemas similares desde el punto de vista de la empresa. ....	28
Tabla 4. Personal involucrado.....	38
Tabla 5. Definiciones.....	39
Tabla 6. Requisito funcional detección intrusión en el perímetro.....	42
Tabla 7. Requisito funcional consulta del estado del sistema.....	42
Tabla 8. Requisito funcional consulta del historial de eventos del sistema.....	43
Tabla 9. Requisito funcional activar/desactivar el sistema de seguridad.....	43
Tabla 10. Requisito funcional gestión de la IP del servidor.....	44
Tabla 11. Requisito funcional modificación de la IP del sistema Arduino.....	44
Tabla 12. Requisito funcional de la gestión de los sensores del sistema.....	45
Tabla 13. Requisito funcional gestión de los actuadores del sistema.....	45
Tabla 14. Requisito funcional gestión de las acciones del sistema.....	46
Tabla 15. Requisito funcional paro de la alarma.....	47



# Códigos

---

Código 1. Recepción de las acciones en Arduino.....	62
Código 2. Recepción de los sensores PIR. ....	63
Código 3. Método que comprueba el estado de los sensores de ultrasonidos .....	64
Código 4. Función para cargar las IP´s .....	68
Código 5. Función encargada de asociar las acciones. ....	73
Código 6. Código encargado de almacenar las acciones en la BBDD.....	74
Código 7. Método que se encarga de generar el mensaje con las acciones .....	75



# 1. Introducción

## 1.1. Motivación

Cada vez más el ser humano siente la necesidad de sentirse seguro por lo que opta por el uso de algún sistema de seguridad para su protección personal y material. Analizando los sistemas actuales se ha comprobado que los sistemas de seguridad más potentes que existen en la actualidad normalmente están asociados con un gran coste económico y una complejidad que hacen que no pueda ser posible su instalación en gran cantidad de lugares. Por otra parte muchos de los sistemas más económicos ofrecen muy pocas prestaciones que hacen que no sea una buena elección su instalación. Por ello surge la necesidad del desarrollo de un sistema que se sitúe entre estos dos grandes grupos.

El auge de pequeños microcontroladores como Arduino permiten desarrollar sistemas que pueden interconectar diferentes componentes, así como establecer una comunicación entre ellos y entre diferentes dispositivos, por lo que hace que sea interesante su uso para desarrollar un sistema de seguridad que mantenga una buena relación entre prestaciones, precio y sencillez para el usuario.

## 1.2. Objetivos

El objetivo de este proyecto será el diseño y desarrollo de un sistema de seguridad perimetral programable mediante el uso de un microcontrolador Arduino que deberá cumplir una serie de requisitos. Este sistema deberá de interactuar y comunicarse con el cliente a través de internet. Los pasos marcados para lograr los objetivos serán:

- Analizar el estado actual de los sistema de seguridad existentes
- Buscar la tecnología que se empleará en el desarrollo.
- Realizar la especificación de los requisitos que debe cumplir el producto.
- Diseñar el sistema tanto hardware como software.
- Implementar el sistema de alarma cubriendo los requisitos especificados.
- Verificar mediante un prototipo que se han cumplido los objetivos.

Otro de los objetivos será realizar un sistema lo más potente posible mediante el uso de tecnologías libres y por tanto de bajo coste.



## 2. Estado actual y conceptos teóricos

### 2.1. Introducción

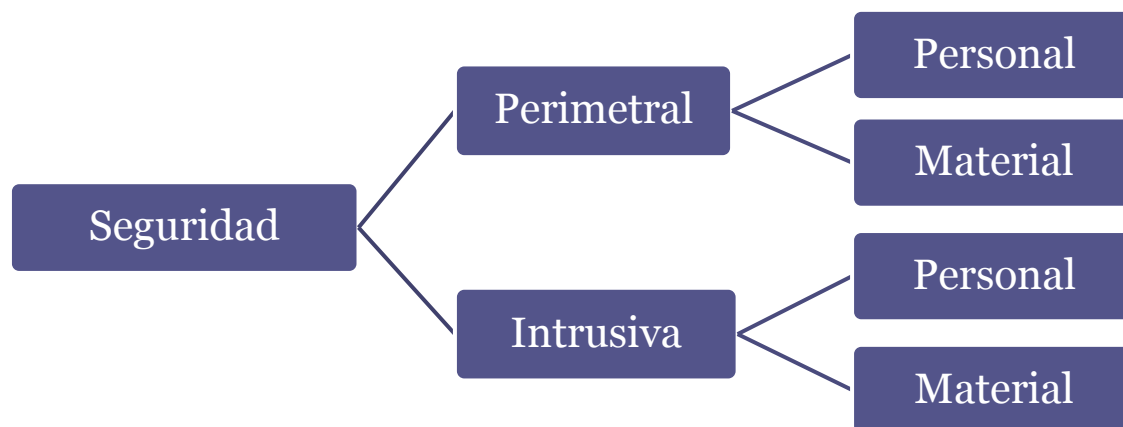
En este apartado repasaremos algunos conceptos teóricos sobre seguridad y sistemas de alarma y mostraremos algunos de los sistemas similares que existen en la actualidad, los cuales analizaremos para ver las características que dispone cada uno. También repasaremos la plataforma Arduino y los sensores que vamos a emplear para desarrollar el sistema de seguridad.

### 2.2. Teoría de la seguridad

#### 2.2.1. Conceptos

El término seguridad posee múltiples usos. A grandes rasgos, puede afirmarse que este concepto que proviene del latín *securitas* hace foco en la característica de seguro, es decir, realza la propiedad de algo donde no se registran peligros, daños ni riesgos. Una cosa segura es algo firme, cierto e indubitable. La seguridad, por lo tanto, puede considerarse como una certeza.

En este proyecto vamos a tratar el concepto seguridad realizando la siguiente división:



**Ilustración 1. División de la seguridad realizada en el proyecto.**

Por un lado tenemos a la seguridad perimetral, es decir la seguridad en el exterior de un recinto donde se va a situar el sistema de seguridad. Esta seguridad intenta detectar algún tipo de actividad sospechosa en una zona delimitada del exterior de un recinto. El otro tipo de seguridad que tenemos en la seguridad intrusiva, es decir la seguridad en el interior de un recinto. Esta seguridad intrusiva intenta que ningún intruso pueda acceder al interior de un recinto.

Cada una de estas dos divisiones espaciales de seguridad contiene otra división según lo que queramos asegurar. Por un lado está la seguridad personal que intenta proteger la

salud de las personas, ya sea de posibles agresiones o simplemente detectar algún problema de salud. En la otra división encontramos la seguridad material que intenta que no se produzcan robos ni deterioro de material, así como también intenta detectar posibles incendios, escapes de gas o inundaciones.

## **Sistema de Alarma**

Un sistema de alarma es un elemento de seguridad pasiva. Esto significa que no evitan una situación anormal, pero sí son capaces de advertir de ella, cumpliendo así también función disuasoria frente a algunos posibles problemas. Son capaces además de reducir el tiempo de ejecución de las acciones a tomar en función del problema presentado, reduciendo así las pérdidas.

## **Funcionamiento**

Una vez que la alarma comienza a funcionar, o se activa dependiendo del sistema instalado, este puede tomar acciones en forma automática. Por ejemplo: Si se detecta la intrusión de una persona a un área determinada, mandar un mensaje telefónico a uno o varios números, El uso de la telefonía para enviar mensajes, de señales o eventos se utilizó desde hace 60 años pero desde el año 2005 con la digitalización de las redes de telefonía, la comunicación deja de ser segura, actualmente la telefonía es solo un vínculo más y se deben enviar mensajes mediante GPRS a direcciones IP de servidores que ofician de receptores de las señales o eventos, también se utiliza la conectividad propia de las redes IP. Si se detecta la presencia de humo, calor o ambos, mandar un mensaje al "servicio de monitoreo" o accionar la apertura de rociadores en el techo, para que apaguen el fuego. Si se detecta la presencia de agentes tóxicos en un área, cerrar las puertas para que no se expanda el problema.

Para esto, la alarma tiene que tener conexiones de entrada, para los distintos tipos de detectores, y conexiones de salida, para activar otros dispositivos que son los que se ocupan de hacer sonar la sirena, abrir los rociadores o cerrar las puertas.

Todos los sistemas de alarmas traen conexiones de entrada para los detectores y por lo menos una de salida para la sirena. Si no hay más conexiones de salida, la operación de comunicar a un servicio de monitoreo, abrir el rociador o cerrar las puertas deberá ser realizada en forma manual por un operador.

Uno de los usos más difundidos de un sistema de alarma es advertir el allanamiento en una vivienda o inmueble. Antiguamente los equipos de alarma podrían estar conectados con una Central Receptora, también llamada Central de Monitoreo, con el propietario mismo (a través de teléfono o TCP/IP) o bien simplemente cumplir la función disuasoria, activando una sirena (la potencia de la sirena estará regulada por las distintas leyes de seguridad del Estado o región correspondiente). En la actualidad existen servicios de "monitoreo por Internet" que no utilizan una "central receptora" ni una "central de monitoreo" sino redes compartidas en Internet donde se derivan directamente las señales o eventos a teléfonos inteligentes (*smartphones*), tabletas y portátiles conectados a Internet utilizando un navegador de código abierto, envían la información directamente a quienes deben recibirla, usuarios o titulares de los servicios, personal técnico para la reparación de falsas alarmas, operadores de monitoreo quienes verifican las señales que requieren de procesamiento humano y la



autoridad de aplicación (Policía, Bomberos, etc) para el caso de hechos reales donde el estado debe intervenir.

Para la comunicación con una vieja Central Receptora de Alarmas o un actual "servicio de monitoreo" en Internet, se necesita de un medio de comunicación, como podrían serlo: la antigua línea telefónica RTB o el canal GPRS de una línea GSM, un transmisor por radiofrecuencia o mediante transmisión TCP/IP que utiliza una conexión de banda ancha ADSL, enlaces TCP/IP inalámbricos y servicios de Internet por cable.

### **Partes de un sistema de alarma**

Un sistema de alarma se compone de varios dispositivos conectados a una central procesadora.

- **Central procesadora:** es la CPU del sistema. En ella se albergan la placa base, la fuente y la memoria central. Esta parte del sistema es la que recibe las diferentes señales que los diferentes sensores pueden emitir, y actúa en consecuencia, disparando la alarma, comunicándose con "el servicio de monitoreo" por medio de un módem, comunicador incorporado o no por TCP/IP, GPRS o Transmisor de radio. Se alimenta a través de corriente alterna y de una batería respaldatoria, que en caso de corte de la energía, le proporcionaría una autonomía al sistema de entre 12 horas y 4 días (dependiendo de la capacidad de la batería).
- **Teclado:** es el elemento más común y fácil de identificar en una alarma. Se trata de un teclado numérico del tipo telefónico. Su función principal es la de permitir a los usuarios autorizados (usualmente mediante códigos preestablecidos) armar (activar) y desarmar (desactivar) el sistema. Además de esta función básica, el teclado puede tener botones de funciones como: Emergencia Médica, Intrusión, Fuego, etc. Por otro lado, el teclado es el medio más común mediante el cual se configura el panel de control.
- **Gabinete de sirena exterior:** es el elemento más visible desde el exterior del inmueble protegido. Se trata de una sirena con autonomía propia (puede funcionar aun si se le corta el suministro de corriente alterna o si se pierde la comunicación con la central procesadora) colocada dentro de un gabinete protector (de metal, policarbonato, etc). Puede tener además diferentes sistemas luminosos que funcionan en conjunto con la disuasión sonora. La sirena exterior es opcional y en algunos sitios desaconsejada, en cambio la sirena interior resulta obligatoria de acuerdo con las normas europeas y americanas.
- **Detectores de movimiento (PIR):** son sensores que detectan cambios de temperatura y movimiento. Si estos sensores detectan movimiento estando el sistema conectado, dispararán la alarma. También se venden detectores con la intención de no detectar mascotas, tales como perros y gatos. No se sugiere el uso de detectores "antimascota" con animales pues tanto la falta de



detección como la confiabilidad para la no producción de falsas alarmas se ve seriamente afectada.

Existen también detectores que utilizan efecto Doppler de microondas. Solos o combinados con sensores PIR son mucho más confiables que con solo PIP; barreras infrarrojas de haz simple o múltiple detectan el paso de un intruso cuando interrumpe los haces.

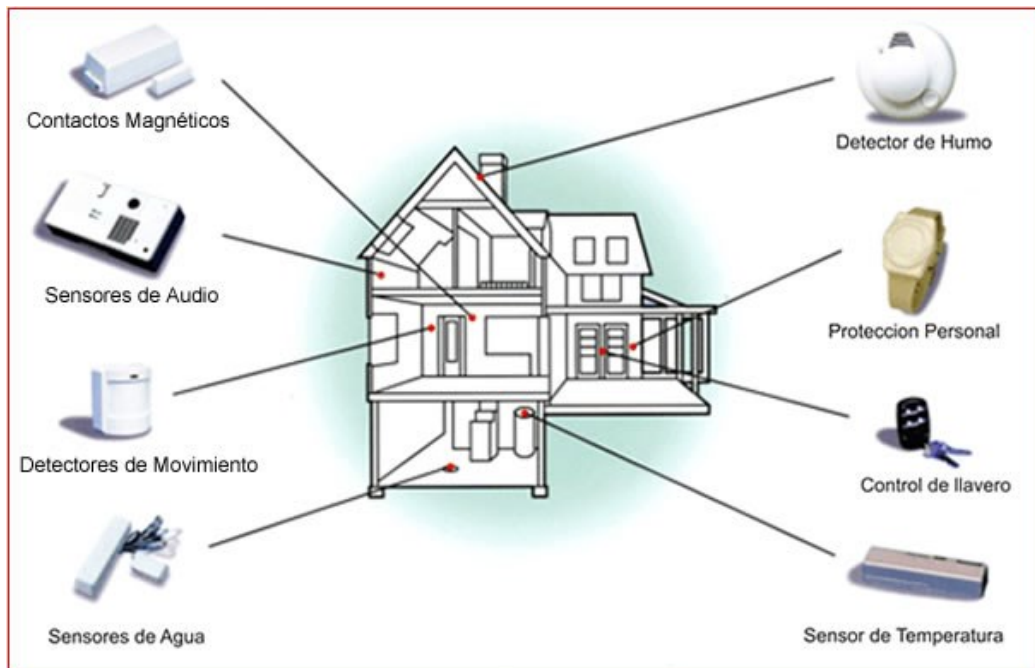
- Detectores magnéticos: se trata de un sensor que forma un circuito cerrado por un imán y un contacto muy sensible que al separarse, cambia el estado (se puede programar como NC o NA) provocando un salto de alarma. Se utiliza en puertas y ventanas, colocando una parte del sensor en el marco y otra en la puerta o ventana misma.
- Sensores inerciales o sísmicos: están preparados para detectar golpes sobre una base. Se colocan especialmente en cajas fuertes, también en puertas, paredes y ventanas. Detectan el intento de forzar su apertura.
- Detectores de rotura de cristales: son detectores microfónicos, activados al detectar la frecuencia aguda del sonido de una rotura de cristal.
- Lapa (detector termovelocimétrico): elemento adherido a una caja fuerte. Advierte de un posible butrón o intento de sabotaje de la misma. Adopta el nombre de termovelocimétrico dado que en su interior alberga tres tipos de detectores seriados, uno de cambio de temperatura, un sísmico, y uno de movimiento.
- Detector personas caídas (hombre muerto): elemento inalámbrico que permite detectar desvanecimientos o caídas de personas solas.

### 2.2.2. Instalaciones de seguridad

En esta sección vamos a mostrar algunos de los esquemas de instalaciones de seguridad más comunes empleados en la actualidad, donde se pueden ver todos los sensores y dispositivos empleados.

#### **Esquema 1**



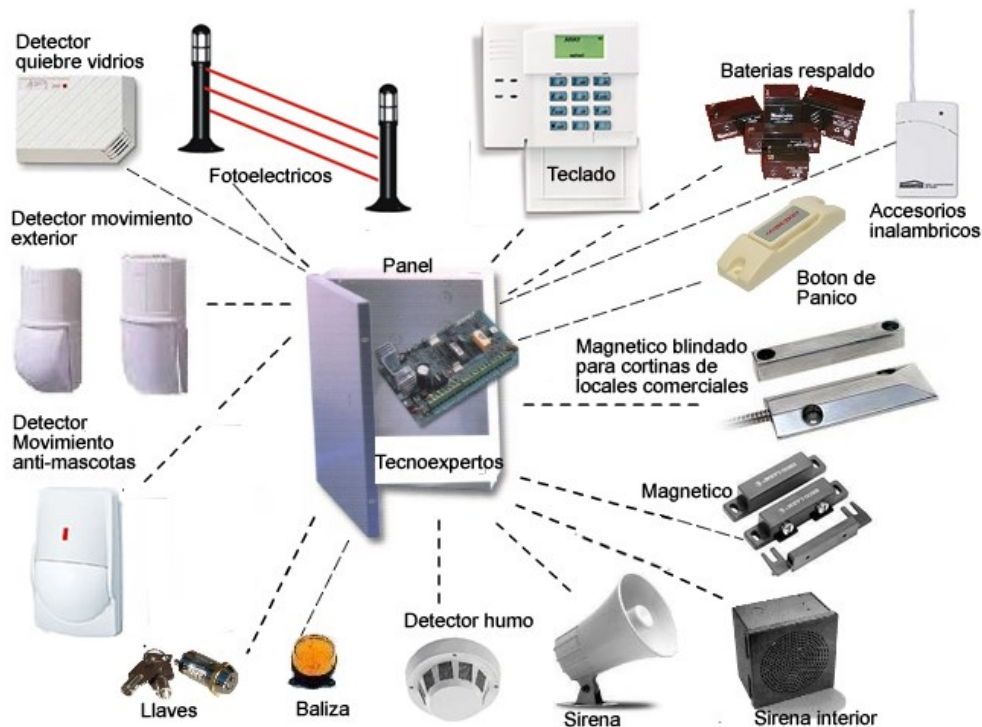


**Ilustración 2. Esquema instalación de seguridad 1**

El primer esquema muestra una instalación común en la mayoría de empresas de seguridad. Cuenta con los siguientes componentes:

- Contactos magnéticos. Estos contactos se colocan en puertas y ventanas. En cuanto se separa la hoja de la puerta o ventana unos centímetros se cierra o abre el circuito , según sea el modelo, y la central produce una alarma.
- Sensores de ruido. Pretenden detectar ruidos como el provocado al romper un cristal y ruidos producidos por un intruso.
- Detectores de movimiento. Buscan detectar cualquier movimiento dentro del radio de alcance.
- Sensores de agua. Sirven para detectar inundaciones.
- Detector de humo. Pretenden avisar si se produce fuego en un recinto.
- Protección personal. Son dispositivos que cumplen funciones de seguridad personal como pueden ser un reloj que si no percibe movimiento de su usuario activa una alarma.
- Control de llavero. Son pequeños dispositivos que sirven para controlar el sistema de alarma. También se puede activar la alarma manualmente desde ellos.
- Sensor de temperatura. Sirven también para detectar fuego o algún problema de exceso de calor.

## Esquema 2



**Ilustración 3. Esquema instalación de seguridad 2**

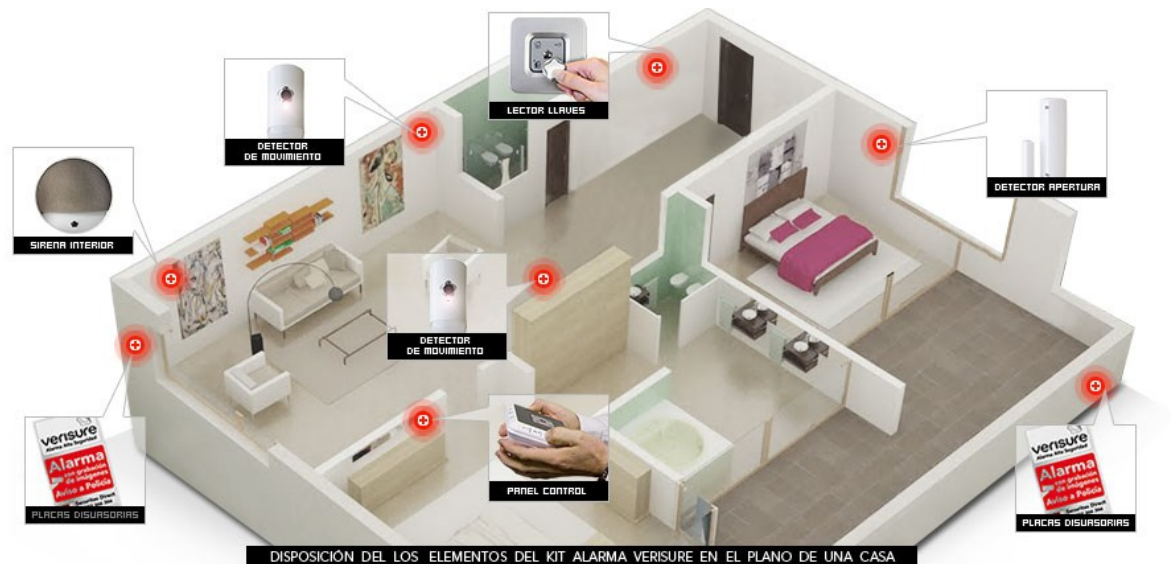
Este esquema añade los siguientes dispositivos que no están presentes en el esquema 1:

- Panel central que controla todo el sistema de alarma. Todos los dispositivos se conectan a este panel.
- Teclado numérico para poder activar y desactivar la alarma mediante contraseña.
- Baterías de respaldo por si se produce un corte de la red eléctrica.
- Sirenas para avisar acústicamente y disuadir a los intrusos.
- Detector de rotura de vidrios como ventanas.
- Barreras de infrarrojos que detectan el paso.
- Sensores de movimiento que no detectan las mascotas para evitar falsas alarmas.
- Baliza que indica el estado de la alarma.

### Esquema 3



## Sistema de seguridad perimetral programable inteligente



**Ilustración 4. Esquema instalación de seguridad 3**

Este esquema cuenta con dispositivos incluidos en el esquema 1 y 2. Añade carteles en el exterior de la casa como método disuasorio. Se trata de un esquema muy comercial.

## 2.3. Sistemas similares

### 2.3.1. Functiodomo

Functiodomo se trata de un sistema de alarma y automatización del hogar que ha desarrollado en la empresa Functio & Ars. El reto principal de este diseño ha sido preparar un sistema de control del hogar. Este sistema se puede instalar en cualquier hogar habitado, su instalación es sencilla y se puede llevar a cabo de forma rápida y sin modificación alguna de los elementos de la vivienda. Otro objetivo del proyecto es su modularidad y evolución con el tiempo, se le puede ir añadiendo más módulos y ampliar sus características y que sea el propio usuario el que lo instale en su hogar. Por este motivo se han utilizado tecnologías abiertas, ampliamente documentadas y con un gran número de expertos en todo el mundo, para que el producto sufra unas evoluciones continuas y el cliente pueda aprovechar esas mejoras y aumente su confort en el hogar con un gasto mínimo. También se tiene en cuenta el ahorro energético. Para ello se podrán hacer adaptaciones de los programas de automatización para poder minimizar el consumo energético en función de la actividad en el hogar de cada cliente.

Todo el sistema Functiodomo se puede controlar desde cualquier dispositivo con conexión a internet y un navegador web. Una vez instalado el sistema functiodomo y conectada su unidad de control a internet se podrá acceder al sistema de control escribiendo en su navegador la dirección IP de su hogar o una dirección web privada. Dentro del sistema Functiodomo se implementa sistemas de alarma y vigilancia:

- Alarma global** integrada en el sistema functiodomo con varias zonas (detección+vigilancia+respuesta).
- **Alarma básica** integrada en el sistema functiodomo (detección+vigilancia)
- **Alarma SMS** con avisos y control por mensajes de texto SMS (sin necesidad de conexión a internet).

El sistema se basa en una unidad de control, la cual se comunica con los sensores y actuadores de forma inalámbrica. A su vez esta unidad está conectada con internet. Esta conexión permite poder acceder a la unidad de control con cualquier dispositivo conectado a internet, ya sea conectado a la propia red donde está la unidad de control o cualquier otra red de internet.





**Ilustración 5. Esquema del sistema funciodomo**

## 2.3.2. Powermax Pro

La alarma Powermax Pro es un completo sistema de seguridad para el hogar. El sistema PowerMax no es solo un sistema de alarma, sino que además te protege del fuego, las inundaciones, escapes de gas, le permite controlar luces y también incorpora un simulador de presencia para cuando no está en casa. Además Powermax avisa telefónicamente cuando hay algún incidente. Todas las funciones se pueden controlar remotamente desde su teléfono móvil, desde el armado del sistema, al encendido de la calefacción a distancia.

El sistema tiene 28 zonas independientes vía radio, 2 zonas cableadas, una salida programable, dos salidas para sirena, un puerto de programación RS 232, ocho códigos diferentes para usuarios, función confirmación de retorno para los hijos, hasta ocho mandos a distancia de seguridad encriptados, hasta dos teclados bidireccionales, control domótico de luces y electrodomésticos, compatible x10, respuesta de voz completa, centro de mensajes hablados, marcador de voz, alarma por zonas, función habla-escucha por teléfono, registro de 100 eventos, teclado con doble función, control remoto por teléfono, compatible con central receptora de alarmas, display multifunción y batería de respaldo.

La consola controla hasta 28 zonas vía radio con Powercode supervisadas y la posibilidad de incorporar dos zonas cableadas. Cada zona se identifica con un nombre y un número el cual se puede especificar. La tecnología Powercode asegura una perfecta comunicación entre la central y los sensores de alarma con un mínimo consumo de energía. La mayoría de los sensores tiene una duración de las pilas superiores a los 3 - 5 años. Las zonas cableadas pueden ser conectadas a sensores y dispositivos de otras marcas que funcionan por apertura o cierre de contacto.



**Ilustración 6. Central Powermax Pro**

La salida programable se puede utilizar para controlar otro aparato como por ejemplo una puerta de garaje, esto permite que el teclado de la consola Powermax o el mando llavero funcione como un control remoto de la puerta de su garaje.

Permite conectar dos sirenas además de la sirena interna que ya lleva instalada. Esto representa una gran ventaja para grandes locales así como la posibilidad de instalar una sirena exterior, fundamental en cualquier sistema de alarma. Además también puede conectar una sirena inalámbrica que no necesita cables entre la sirena y la consola central. También se pueden colocar sirenas internas de tipo X10 que tampoco necesitan cables y que pueden instalarse por toda la casa (requiere el interface bidireccional de X10).

Incorpora un puerto de programación RS-232 que permite programar la central desde un ordenador personal. También sirve para conectar la central a otros dispositivos como un módem GSM externo.

Permite programar hasta 8 códigos de usuarios diferentes, cada uno con su mando a distancia. Los códigos de usuario del 5 al 8 permiten además la función de confirmación de retorno. Esto permite saber y distinguir quien y cuando ha utilizado la alarma o cuando entra y sale cada persona. Las opciones de programación del mando incluyen el armado de protección perimetral o "Modo Casa", el "Modo Total", desarmado del sistema de alarma y funciones auxiliares como el encendido de luces exteriores o la apertura de la puerta del garaje.

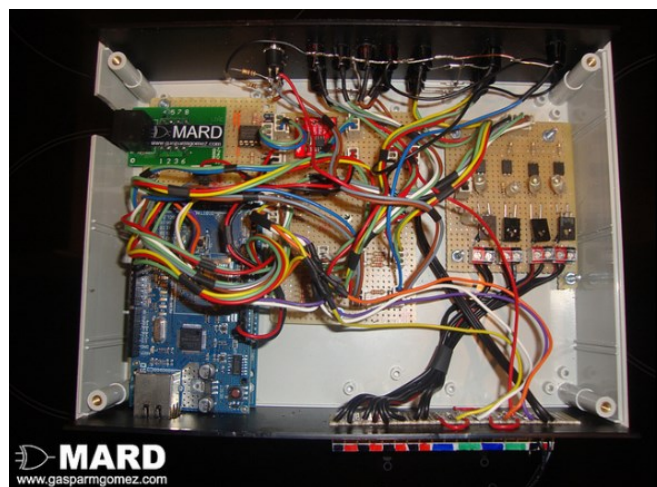


El sistema permite programar 4 números de teléfonos privados o móviles y el propio sistema llamará a estos números en eventos programados, como cuando hay una alarma o el sistema ha sido desarmado. También se puede conectar el enlace GSM de Powermax Pro que le permite instalar la alarma en lugares donde no hay conexión telefónica normal. Para ello solo hay que utilizar una tarjeta sim de teléfono GSM para tener un control y una comunicación completa. Este enlace también puede ser usado como línea de comunicación adicional y alternativa en cualquier instalación ya que proporciona un segundo medio de comunicación a la central de alarma en caso de que corten el cable del teléfono.

El sistema de supervisión de transmisores, que incluye, permite que se puede programar la powermax pro para que le avise por teléfono en caso de que no se produzca movimiento durante un determinado periodo de tiempo. Por ejemplo puede hacer que la alarma se active si los volumétricos no detectan nada en más de 8 horas. Esto resulta muy útil en caso de caídas de personas mayores.

### 2.3.3. MARD

El sistema MARD se trata de un sistema de alarma inteligente basado en la placa Arduino. La placa empleada en este sistema se trata de la versión MEGA.



**Ilustración 7. Interior del sistema MARD**

El sistema cuenta con las siguientes características:

- 1.- Armado automático. El equipo se arma automáticamente en función de una programación horaria diaria, almacenada en una memoria eeprom.
- 2.- Capacidad para dos zonas con sensores PIR.
- 3.- Dos botones anti pánico.
- 4.- Sistema conectado con aplicación web haciendo de central receptora.
- 5.- Envío de señales de alarma vía correo electrónico mediante una Arduino Ethernet Shield.



- 6.- Reprogramación total del equipo vía web.
- 7.- Capacidad para manejar 4 sirenas.
- 8.- Consola.



Ilustración 8. Parte frontal y trasera del sistema MARD

### 2.3.4. CEBEK AL-1

El Cebek AL-1 se trata de un sencillo sistema de alarma para usos domésticos diseñada en base a circuitos C-Mos de elevada inmunidad a falsas alarmas. La salida es con relé asilado del circuito los que permite conectar cualquier tipo de sirena o avisador.



Ilustración 9. Central de alarma CEBEK AL-1

El funcionamiento de la alarma es muy simple. Una vez conectada la alimentación de 12V, para lo que se puede emplear un interruptor, se disponen de 10 segundos para



## Sistema de seguridad perimetral programable inteligente

salir sin que la alarma lo detecte. Pasado este tiempo la alarma queda activada. Al entrar se dispone de un tiempo de entrada ajustable de 10 a 30 segundos para desactivar la alarma. Pasado este tiempo la alarma activa la salida. La salida estará activada durante 3 minutos. Si pasados estos 3 minutos la alarma detecta que sigue abierto el contacto volverá a hacer el ciclo de trabajo, tiempo entrada, indefinidamente hasta que se desactiva la alimentación.

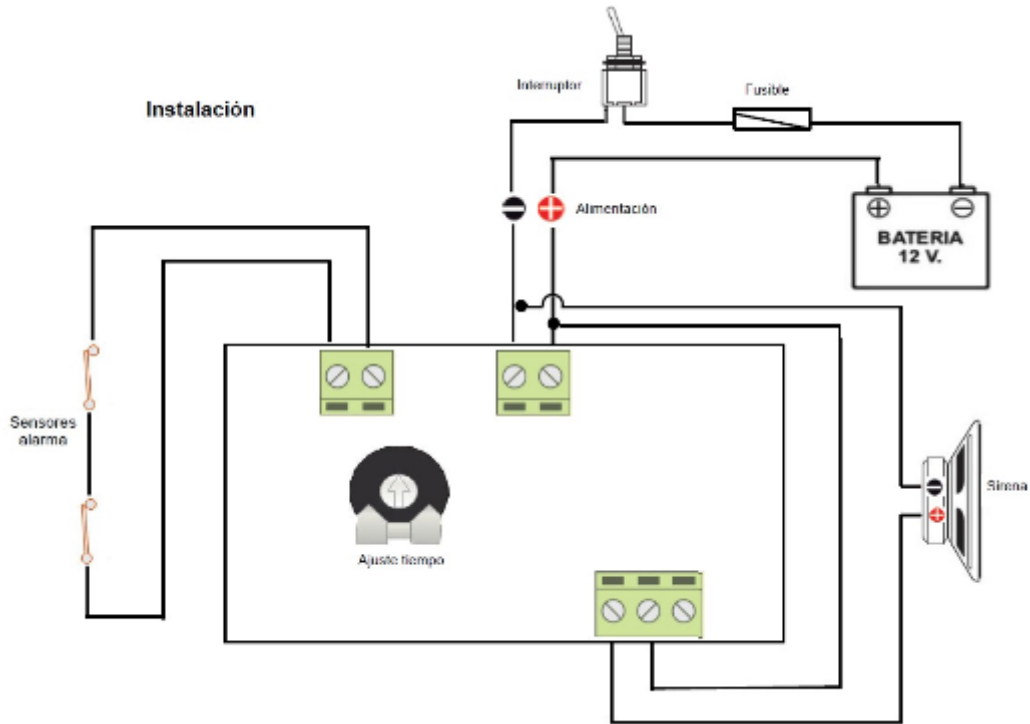


Ilustración 10. Esquema de instalación del CEBEK AL-1

## 2.4. Análisis sistemas similares

### 2.4.1. Cuantitativo

Desde el punto de vista cuantitativo, es decir, parámetros técnicos, se tiene el ejemplo mostrado en la Tabla 1, en el que se comparan los sistemas presentados en el apartado anterior.

	Precio	Microprocesador	Sensores	Aplicación para móvil	Envío de sms	Registro de eventos	Aplicación web
Funciodomo	-----	Arduino	Movimiento Domóticos	Si	Si	No	Si
Powermax Pro	225€ (Solo central)	-----	Movimiento Gas Inundación Humo Magnético	No	Si	Si	No
MARD	-----	Arduino Mod. Mega	Movimiento	No	No	Si	Si
CEBEK AL-1	19,50 (Central)	-----	Movimiento	No	No	No	No

**Tabla 1. Análisis cuantitativo de los sistemas similares.**

Como se puede observar en la tabla, dos sistemas usan Arduino como elemento de control. Además todos ellos tienen sensores de movimiento. Obviamente esto nos sugiere emplear este tipo de sensores y de controlador.

También se puede observar que aspectos como el registro de eventos o aplicación Web de control también aparecen en los distintos sistemas, por tanto, parece conveniente diseñar e implementar el sistema para ofrecer esta funcionalidad.

Si nos fijamos solo un sistema implementa aplicación para el móvil y dos de ellas envió de sms por lo que es un aspecto que se debe tener en cuenta para futuras ampliaciones del sistema.

### 2.4.2. Cualitativo

Desde el punto de vista cuantitativo vamos a realizar una separación entre el punto de vista del usuario y el punto de vista de la empresa.

Desde el punto de vista del usuario:



	Facilidad de uso	Facilidad de instalación	Facilidad de configuración
Funcniodomo	Alta	Media	Media
Powermax Pro	Media	Media	Baja
MARD	Alta	Media	Alta
CEBEK AL-1	Muy Alta	Alta	Alta

**Tabla 2. Análisis cualitativo de los sistemas similares desde el punto de vista del usuario.**

Si analizamos la tabla podemos ver que los sistemas más completos y que nos ofrecen mayores opciones tienen una mayor dificultad en cuanto a su uso y a su configuración. Por lo tanto hay que buscar diseñar el sistema para que sea completo pero a la vez sencillo de configurar y usar. Esto hará que cualquier usuario pueda manejar el sistema sin tener grandes conocimientos y sin tener que leerse un extenso manual de usuario.

En cuanto a la facilidad de instalación dependerá en gran medida de los diferentes sensores utilizados y su dificultad de montaje y conexión. Los sensores empleados en el sistema no necesitarán ningún tipo de configuración al instalarlos y la conexión será muy sencilla.

Desde el punto de vista de la empresa

	Potencia	Escalabilidad	Fiabilidad	Seguridad
Funcniodomo	Media	Media	Media	Media
Powermax Pro	Alta	Alta	Media	Alta
MARD	Baja	Media	Media	Baja
CEBEK AL-1	Baja	Baja	Media	Baja

**Tabla 3. Análisis cualitativo de los sistemas similares desde el punto de vista de la empresa.**

Potencia: Capacidad de integrar diferentes sensores, actuadores (hasta dónde puedo llegar)

Escalabilidad: Capacidad de aumentar el sistema sin impacto.

Fiabilidad: Probabilidad de que el sistema funcione correctamente sin fallos a lo largo del tiempo.

Seguridad: Capacidad del sistema para detectar peligros o manipulaciones.

Si nos fijamos en la tabla los sistemas más baratos y sencillos, el CEBEK AL-1 y el MARD, obtienen los resultados más bajos del análisis ya que tienen una conexión muy limitada de diferentes sensores y actuadores. También no disponen de sistemas anti manipulación que los hacen más vulnerables.

El sistema Functiodomo obtiene unos resultados intermedios que hacen de él un sistema correcto pero mejorable.

La alarma Powermax Pro obtiene los mejores resultados ya que permite la conexión de un gran número de sensores sin apenas modificaciones así como diversos sistemas anti manipulación.

El objetivo del sistema a desarrollar es maximizar cada una de estas características buscando obtener un sistema que mejore o iguale el Powermax Pro.

## 2.5. Tecnología

### 2.5.1. Arduino

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

El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida. Los microcontroladores más usados son el Atmega168, Atmega328, Atmega1280, ATmega8 por su sencillez y bajo coste que permiten el desarrollo de múltiples diseños. Por otro lado el software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring y el cargador de arranque (boot loader) que corre en la placa.

Desde octubre de 2012, Arduino se usa también con microcontroladoras CortexM3 de ARM de 32 bits, que coexistirán con las más limitadas, pero también económicas AVR de 8 bits. ARM y AVR no son plataformas compatibles a nivel binario, pero se pueden programar con el mismo IDE de Arduino y hacerse programas que compilen sin cambios en las dos plataformas. Eso sí, las microcontroladoras CortexM3 usan 3.3V, a diferencia de la mayoría de las placas con AVR que usan mayoritariamente 5V. Sin embargo ya anteriormente se lanzaron placas Arduino con Atmel AVR a 3,3V como la Arduino Fio y existen clónicos de Arduino Nano y Pro como Meduino en que se puede conmutar el voltaje.

Arduino se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado a software del ordenador (por ejemplo: Macromedia Flash, Processing, Max/MSP, Pure Data). Las placas se pueden montar a mano o adquirirse. El entorno de desarrollo integrado libre se puede descargar gratuitamente.

Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles muy fáciles de usar, debido a que el IDE con el que trabaja es fácil de aprender a utilizar, y el lenguaje de programación con el que trabaja es simple, pues se creó para artistas, diseñadores, aficionados y cualquier interesado en crear entornos u objetos interactivos. Arduino puede tomar información del entorno a



través de sus pines de entrada de toda una gama de sensores y puede afectar aquello que le rodea controlando luces, motores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un ordenador, si bien tienen la posibilidad de hacerlo y comunicar con diferentes tipos de software (p.ej. Flash, Processing, MaxMSP). Las placas pueden ser hechas a mano o compradas montadas de fábrica; el software puede ser descargado de forma gratuita. Los ficheros de diseño de referencia (CAD) están disponibles bajo una licencia abierta, así pues eres libre de adaptarlos a tus necesidades.

El proyecto Arduino recibió una mención honorífica en la categoría de Comunidades Digital en el Prix Ars Electrónica de 2006.

### **Historia**

En el año 2003 en Italia específicamente en el instituto Ivrea, el docente Massimo Banzi enseñaba el uso de PICs a estudiantes de Diseño Interactivo, los cuales no tenían el conocimiento técnico para utilizar las herramientas de bajo nivel para la programación que existían en esos momentos, por tal motivo desarrolló una herramienta de programación de PICs bajo la plataforma MAC (ya que la mayoría de los estudiantes utilizaban esta plataforma), esta herramienta y el ambiente de desarrollo Processing sirvieron como ejemplo a el colombiano Hernando Barragan que en ese momento era estudiante del instituto, para desarrollar la tarjeta Wiring, el lenguaje de programación y su ambiente de desarrollo. Poco tiempo después Massimo, David Cuartiles investigador en el instituto y Gianluca Martino desarrollador local contratado para desarrollar hardware para los proyectos de los estudiantes desarrollaron una tarjeta basada en el trabajo de Hernando Barragan la cual era más pequeña y económica que la Wiring a la cual llamaron Arduino.

Más tarde se unieron a este grupo los estudiantes Mellis y Zambetti que mejoraron el modelo de Wiring, logrando construir una tarjeta básica y un ambiente de desarrollo completo. En el año 2005 se une a este equipo de trabajo Tom Igoe quien es conocido por sus trabajos en Computación Física (Construcción de sistemas físicos a través de hardware y software que pueden sentir y responder al mundo análogo) y se encarga de las pruebas del sistema con estudiantes del ITP en Estados Unidos, como también de realizar los contactos para la distribución de la tarjeta en territorio americano.

### **Arduino UNO**

El Arduino Uno es la placa más popular de Arduino. Se trata de una placa basada en el microcontrolador ATmega328. Cuenta con 14 pines digitales de entrada/salidas (de los cuales 6 pueden ser usadas como salidas PWM), 6 entradas analógicas, reloj a 16 Mhz, conexión USB, conector de alimentación, cabecera ICSP y botón de reset.



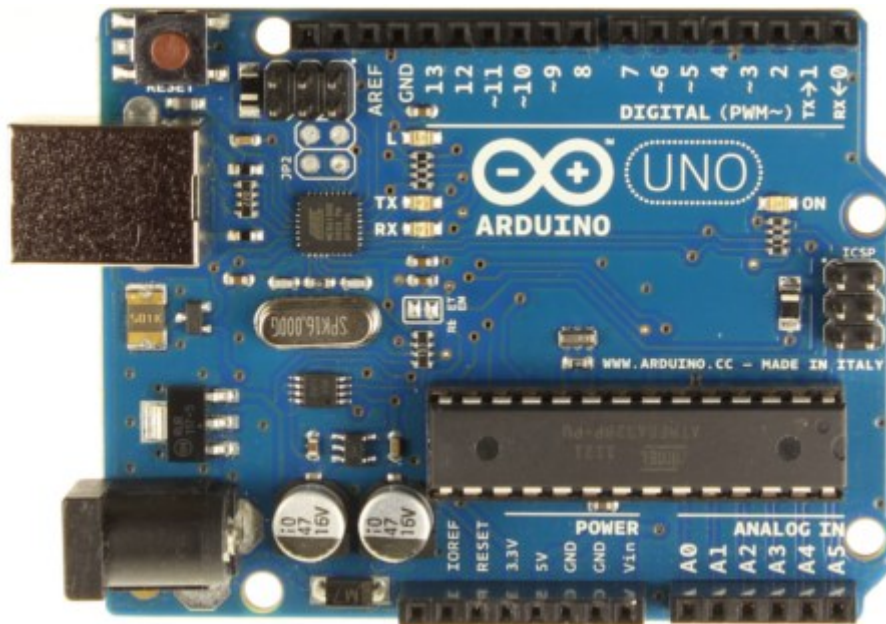


Ilustración 11. Vista frontal del Arduino UNO

Resumen especificaciones Arduino Uno Rev.3

Microcontroladores	ATmega328
Tensión de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (límites)	6-20V
Digital I / O Pins	14 (de los cuales 6 proporcionan PWM)
Pines de entrada analógica	6
Corriente continua para las E / S Pin	40 mA
Corriente de la CC para Pin 3.3V	50 mA
Memoria Flash	32 KB ( ATmega328 ) de los cuales 0,5 KB utilizado por el gestor de arranque
SRAM	2 KB ( ATmega328 )
EEPROM	1 KB ( ATmega328 )
Velocidad del reloj	16 MHz



## Arduino Ethernet Shield

La Arduino Ethernet Shield permite a una placa Arduino conectarse a internet. Está basada en el chip ethernet Wiznet W5100 . El Wiznet W5100 provee de una pila de red IP capaz de TCP y UDP. Soporta hasta cuatro conexiones de sockets simultáneas. La ethernet shield dispone de conectores que permiten conectar a su vez otras placas encima y apilarlas sobre la placa Arduino.



**Ilustración 12. Ethernet Shield**

Arduino usa los pines digitales 10, 11, 12, y 13 (SPI) para comunicarse con el W5100 en la ethernet shield, por lo que estos pines no pueden ser usados para e/s genéricas.

La shield provee un conector ethernet estándar RJ45 para realiar la conexión con otros dispositivos. También incorpora una ranura para tarjetas micro-SD que se puede utilizar para almacenar información.

Sobre la superficie de la placa incorpora un conjunto de LEDs informativos:

- PWR: indica que la placa y la shield están alimentadas
- LINK: indica la presencia de un enlace de red y parpadea cuando la shield envía o recibe datos
- FULLD: indica que la conexión de red es full duplex
- 100M: indica la presencia de una conexión de red de 100 Mb/s (de forma opuesta a una de 10Mb/s)
- RX: parpadea cuando la shield recibe datos
- TX: parpadea cuando la shield envía datos
- COLL: parpadea cuando se detectan colisiones en la red



## 2.5.2. Sensores y actuadores

En el sistema de seguridad desarrollado vamos a hacer uso de varios sensores y actuadores por lo que vamos a hacer un repaso a ellos.

### **Sensor de movimiento infrarrojo (PIR)**

Un sensor infrarrojo pasivo o sensor PIR es un sensor de movimiento electrónico que mide la luz infrarroja (IR) radiada de los objetos situados en su campo de visión.

En cuanto a su principio de operación, el sensor PIR se basa en la idea que todos los objetos emiten energía en forma de radiación a causa de tener un calor corporal por encima del cero absoluto. El sensor del que se componen, piroeléctrico, es capaz de medir la radiación emitida por los objetos.

Los sensores PIR están compuestos por dos ranuras, cada una de ellas sensible a los IR. Cuando un cuerpo caliente pasa por delante del campo de detección del sensor, una de las dos mitades detecta la diferencia de calor, y provoca un diferencial entre las dos mitades de las ranuras. Ocurre lo mismo cuando el cuerpo sale de la zona de detección, la otra mitad detecta un cambio y provoca otra diferencia de potencial igual pero de sentido contrario. De esta manera el sensor es capaz de distinguir si ha habido movimiento en la habitación.

Son sensores de infrarrojo pasivo porque, por un lado, capturan los infrarrojos, y por el otro, como no irradian ninguna energía sobre los objetos, son pasivos.

Las lentes de Fresnel juegan un papel decisivo en los sensores PIR, ya que consiguen ampliar su campo de detección. Una lente de Fresnel es una lente plano-convexa que se utiliza para conseguir focalizar una mayor cantidad de radiación sobre el sensor. Las ventajas de usar sensores PIR es que son baratos, pequeños y fáciles de usar.



**Ilustración 13. Sensor de movimiento PIR**

### **Barrera de infrarrojos**



Las barreras tipo emisor-receptor están compuestas de dos partes, un componente que emite el haz de luz, y otro componente que lo recibe. Se establece un área de detección donde el objeto a detectar es reconocido cuando el mismo interrumpe el haz de luz. Debido a que el modo de operación de esta clase de sensores se basa en la interrupción del haz de luz, la detección no se ve afectada por el color, la textura o el brillo del objeto a detectar. Estos sensores operan de una manera precisa cuando el emisor y el receptor se encuentran alineados. Esto se debe a que la luz emitida siempre tiende a alejarse del centro de la trayectoria.

El receptor de rayos infrarrojos suele ser un fototransistor o un fotodiodo. El circuito de salida utiliza la señal del receptor para amplificarla y adaptarla a una salida que el sistema pueda entender. La señal enviada por el emisor puede ser codificada para distinguirla de otra y así identificar varios sensores a la vez esto es muy utilizado en la robótica en casos en que se necesita tener más de un emisor infrarrojo y solo se quiera tener un receptor.

Las barreras de infrarrojos se emplean en seguridad para detectar el paso de algún intruso u objeto sospechoso. Al no ser visible el haz de luz entre el emisor y el receptor permite poder camuflar esta barrera y así intentar que el intruso no se dé cuenta de que está.



**Ilustración 14. Barrera de infrarrojos**

### **Sensor de ultrasonidos**

Los sensores de ultrasonidos son detectores de proximidad que trabajan libres de roces mecánicos y que detectan objetos a distancias de hasta 8m. El sensor emite un sonido y mide el tiempo que la señal tarda en regresar. Estos reflejan en un objeto, el sensor recibe el eco producido y lo convierte en señales eléctricas, las cuales son elaboradas en el aparato de valoración. Estos sensores trabajan solamente en el aire, y pueden detectar objetos con diferentes formas, colores, superficies y de diferentes materiales. Los materiales pueden ser sólidos, líquidos o polvorientos, sin embargo han de ser deflectores de sonido. Los sensores trabajan según el tiempo de transcurso del eco, es decir, se valora la distancia temporal entre el impulso de emisión y el impulso del eco.

Este sensor al no necesitar el contacto físico con el objeto ofrece la posibilidad de detectar objetos frágiles, como pintura fresca, además detecta cualquier material, independientemente del color, al mismo alcance, sin ajuste ni factor de corrección. Los

sensores ultrasónicos tienen una función de aprendizaje para definir el campo de detección, con un alcance mínimo y máximo de precisión de 6mm.

Estos sensores se emplean en seguridad para detectar la presencia de algún individuo u objeto por una determinada zona, ya que varía la distancia detectada por el sensor al paso del individuo u objeto.



**Ilustración 15. Sensor de ultrasonidos**

## 2.6.Conclusiones

El repaso teórico de la seguridad y de algunos ejemplos de instalaciones nos permite ver el estado en el que se encuentran los sistemas de seguridad actuales. El análisis de algunos sistemas de seguridad nos ha servido para ver las características y la funcionalidad que ofrecen y así poder elegir qué es lo que incorporará el sistema a desarrollar, para así diseñar un sistema completo y diferenciador del resto de sistemas de seguridad que existen.

Tras ver las características que ofrece Arduino nos hemos decantado definitivamente por él como dispositivo central de procesamiento de todo el sistema de alarma, al cual se le podrán conectar fácilmente el conjunto de sensores encargados de la detección de las amenazas. Además el empleo de Arduino permite abaratar los costes al ser una plataforma de hardware libre.



## 3. Especificación de requisitos

### 3.1. Introducción

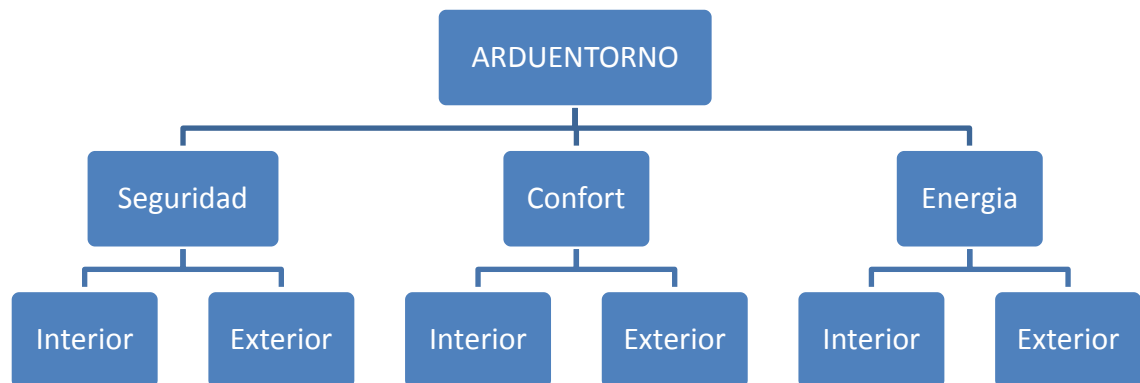
Esta especificación tiene como objetivo analizar y documentar las necesidades funcionales que deberán ser soportadas por el sistema a desarrollar. Para ello, se identificarán los requisitos que ha de satisfacer el nuevo sistema mediante el estudio de mercado y sus necesidades actuales. Además de identificar los requisitos se deberán establecer prioridades, lo cual proporciona un punto de referencia para validar el sistema final que compruebe que se ajusta a las necesidades del usuario.

#### 3.1.1. Propósito

El propósito del documento será la realización de una especificación de requisitos que deberá cumplir el sistema a desarrollar para posteriormente realizar su correcta validación. Dirigido a todas las personas interesadas en el producto.

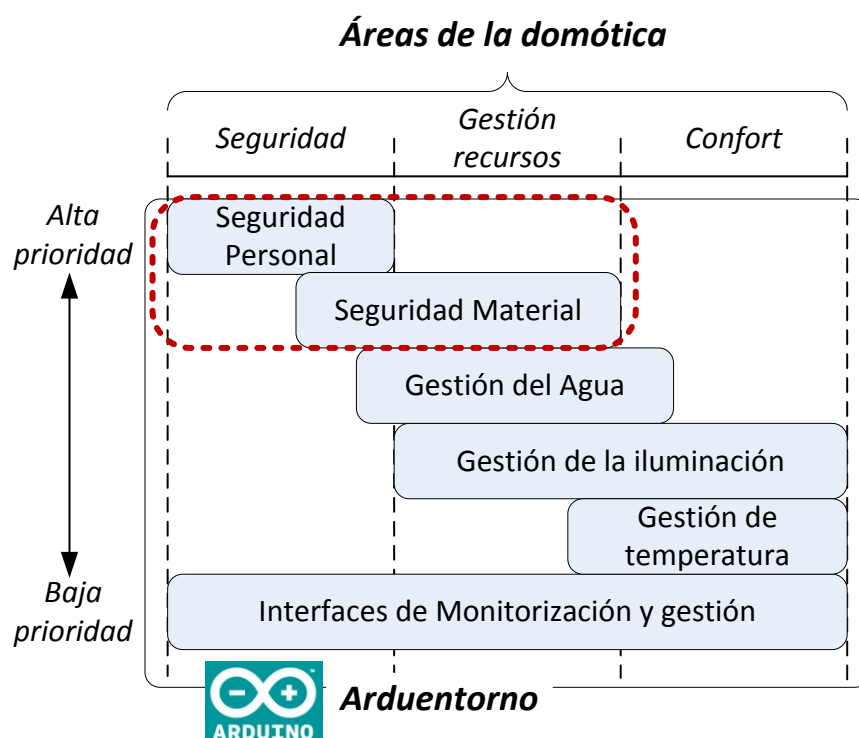
#### 3.1.2. Alcance

El sistema se llama ARDUENTORNO-Seguridad que se encuentra dentro de un proyecto llamado ARDUENTORNO.



**Ilustración 16. Módulos del sistema ARDUENTORNO**

En relación con la domótica y la ubicación del proyecto presentado, en la Ilustración 17, se muestra el actual proyecto.



**Ilustración 17. Ubicación del proyecto en el sistema Arduentorno.**

### 3.1.3. Personal involucrado

Nombre	Jesús Brun Conejos
Rol	Jefe de proyecto
Categoría profesional	Ingeniero en Informática
Responsabilidades	Diseño, desarrollo e implementación de ArduEntorno seguridad
Información de contacto	jesbruco@ei.upv.es
Aprobación	
Nombre	José Luis Poza Luján
Rol	Director del proyecto
Categoría profesional	Profesor Contratado Doctor de la UPV
Responsabilidades	Coordinación de sub-proyectos de Arduentorno. Dirección de los TFG o PFC correspondientes
Información de contacto	jopolu@disca.upv.es
Aprobación	
Nombre	Alberto Pedrera Ros
Rol	Responsable de otro componente del sistema
Categoría profesional	Ingeniero informático
Responsabilidades	Diseño, desarrollo e implementación de ArduEntorno energía
Información de contacto	alpedro@inf.upv.es
Aprobación	
Nombre	Miguel Juan Monter
Rol	Responsable de otro componente del sistema
Categoría profesional	Ingeniero técnico informático
Responsabilidades	Diseño, desarrollo e implementación de ArduEntorno

	iluminación
Información de contacto	mijuamon@inf.upv.es
Aprobación	
Nombre	Alberto Ramírez
Rol	Responsable de otro componente del sistema
Categoría profesional	Ingeniero técnico informático
Responsabilidades	Diseño, desarrollo e implementación de ArduEntorno Android
Información de contacto	
Aprobación	
Nombre	Carlos Gil
Rol	Responsable de otro componente del sistema
Categoría profesional	Ingeniero técnico informático
Responsabilidades	Diseño, desarrollo e implementación de ArduEntorno ambiente
Información de contacto	
Aprobación	
Nombre	Carlos Quer
Rol	Responsable de otro componente del sistema
Categoría profesional	Ingeniero técnico informático
Responsabilidades	Diseño, desarrollo e implementación de ArduEntorno fuente
Información de contacto	
Aprobación	

**Tabla 4. Personal involucrado**

### 3.1.4. Definiciones, acrónimos y abreviaturas

Actuador	Componente electrónico y/o mecánico que ante una señal, es capaz de interactuar con el entorno
Aplicación web	Aplicación software desarrollada para la gestión del sistema de seguridad.
Arduino	Plataforma hardware encargada del control del sistema de seguridad y de gestionar las entradas y salidas del sistema. Se utilizará el modelo Arduino UNO rev3.
Área	Reparto u organización del espacio determinada por los requisitos del cliente. Por ejemplo el área de entrada al jardín o el área de descanso.
Barrera de infrarrojos	Sensor empleado para la detección de paso de un individuo por una determinada zona.
Base de datos	Aplicación en MYSQL para almacenamiento de la topología del sistema así como de eventos detectados. Sus siglas son BBDD.
Detección parcial	Activación de uno o varios sensores, pero no todos, que forman parte de un nodo.
Detección total	Activación de todos los sensores que forman parte de un nodo, como motivo del paso de una persona.
Elemento de control	Cualquier dispositivo final (sensor, actuador o similar) que está conectado a un nodo. Por ejemplo, un sensor de luz o un relé que controla una farola.

Elemento virtual de control	Elemento que emplea un nodo, sin estar conectado a él. Por ejemplo, un sensor que, desde el nodo al que está conectado, envía información a otro nodo.
Entorno	Todo espacio en el que se aplica el sistema de control, aunque no esté controlado (se podría definir también como el conjunto de áreas que el cliente desea automatizar)
Entorno abierto	Espacio a automatizar, por nuestro sistema, el cual se encuentra a la intemperie.
Espacio de control	Conjunto de espacios de interacción con algún tipo de relación controlados por uno o varios nodos de control.
Espacio de interacción	Espacio cubierto por un elemento. Por ejemplo, el espacio que cubre la detección de un sensor de ultrasonidos o el espacio que ilumina una farola.
Ethernet	Estándar empleado para la comunicación entre los Arduinos.
Luminaria	Sistema de iluminación compuesto por bombillas incandescentes, halógenas y/o leds
Microcontrolador	Es un circuito integrado programable capaz de ejecutar órdenes grabadas en su memoria.
Navegador web	Software que permite acceder a internet y abrir la aplicación web que gestiona el sistema de seguridad.
Nodo de comunicaciones	Nodos que, además de poder tener sensores y actuadores, interconectan otros nodos entre sí, u otros nodos con los servidores.
Nodo de control	Conjunto de sensores y actuadores y el Arduino que los controla. Para ser considerado un nodo, el Arduino debe estar conectado al sistema por medio de cualquier tecnología de comunicaciones.
PIR	Passive Infrared Sensor. Sensor pasivo de infrarrojos encargado de la detección de movimiento.
Sección	Conjunto de nodos y un servidor que están conectados.
Sensor	Componente capaz de percibir estímulos del entorno y que es capaz de transmitirlo. Todos los sensores son compatibles con Arduino.
Servidor	Dispositivo al que están conectados un conjunto de nodos y que recibe peticiones o envía órdenes a los nodos.
Sistema Arduino	Conjunto de sensores y actuadores unidos directamente a un micro controlador Arduino. Quedan excluidos aquellos que se unan por una tecnología diferente de la proporcionada por el Arduino.
Sistema de interconexión	Conjunto de componentes encargados de la comunicación entre elementos, nodos y servidor.
Ultrasonido	Sensor que es capaz de detectar la distancia entre este y un obstáculo mediante el uso de ultrasonidos, basándose en el tiempo en que tarda en volver rebotada una señal.

**Tabla 5. Definiciones**

### 3.1.5. Resumen

En las sucesivas secciones realizaremos una descripción general del producto y veremos los requisitos específicos que deberá cumplir el sistema.



## 3.2.Descripción general

### 3.2.1. Perspectiva del producto

El sistema Arduentorno-Seguridad forma parte de un sistema mayor llamado Arduentorno.

### 3.2.2. Funcionalidad del producto

Las funciones que debe realizar el sistema son las siguientes:

- Detectar y avisar posibles amenazas

Mediante el uso de diversos sensores el sistema será capaz de detectar cualquier amenaza y realizar la acción pertinente para avisar e intentar disuadirla. Estas amenazas pueden provenir de intrusiones en el perímetro o en zonas no permitidas.

- Gestionar el sistema de seguridad

El sistema podrá gestionar y visualizar el estado del sistema de seguridad a través de internet. Para ello se desarrollará una aplicación web,

### 3.2.3. Características de los usuarios

Tipo de usuario	Propietario del sistema
Formación	Cualquiera
Habilidades	No se precisan habilidades específicas
Actividades	Manejo del sistema de alarma

Tipo de usuario	Administrador del sistema
Formación	Ingeniero
Habilidades	Programador
Actividades	Mantenimiento y evolución del producto

### 3.2.4. Restricciones

- La conexión entre servidor y Arduino se realizará mediante una conexión Ethernet.
- La base de datos del servidor será MySQL.
- Los sensores serán compatibles con Arduino.
- El Arduino empleado será el modelo UNO rev.3

### 3.2.5. Suposiciones y dependencias

Será necesario un servidor web que tenga instalado el motor de bases de datos MySQL y un intérprete de PHP 5.5.

Para acceder a la aplicación web de control será necesario un navegador web que soporte html5 y css3.



### 3.2.6. Evolución previsible del sistema

Posibles mejoras y evoluciones:

- Comunicación mediante tecnología GSM para el acceso a internet.
- Envío de mensajes SMS.
- Nuevos sensores y actuadores.
- Aplicación de móvil para controlar el sistema de alarma.
- Conexión de varios Arduinos para aumentar la red.

## 3.3.Requisitos específicos

### 3.3.1. Requisitos comunes de los interfaces

#### 3.3.1.1. Interfaces de usuario

La interfaz de usuario se implementara mediante una aplicación web donde se podrá gestionar el sistema de seguridad. La interfaz será sencilla y amigable. Esta aplicación web contará con varias páginas para separar diferentes secciones.

#### 3.3.1.2. Interfaces de hardware

- Sensores PIR
- Sensores Ultrasonido
- Barrera de infrarrojos
- Luces de aviso.

#### 3.3.1.3. Interfaces de software

Navegador Web para el acceso a la aplicación web de gestión del sistema.  
El servidor dispondrá de una base de datos MySQL y un servidor web Apache capaz de interpretar PHP 5.5

#### 3.3.1.4. Interfaces de comunicación

La comunicación entre el Arduino y el servidor será a través de protocolo Ethernet. Esta conexión puede ser directa o través de routers correctamente configurados.



### 3.3.2. Requisitos funcionales

#### 3.3.2.1. Detección intrusión en el perímetro

R.F	1. Detección intrusión en el perímetro
<b>Introducción</b>	El sistema permite detectar la entrada de algún individuo en el perímetro protegido
<b>Entrada</b>	Sensor PIR - Sensor ultrasonidos – Barrera infrarrojos
<b>Proceso</b>	El sistema Arduino analiza la señal de los sensores para ver donde se ha producido la alarma y produce la salida correspondiente.
<b>Salida</b>	Activación actuadores correspondientes y envío del evento de intrusión al servidor.

Tabla 6. Requisito funcional detección intrusión en el perímetro

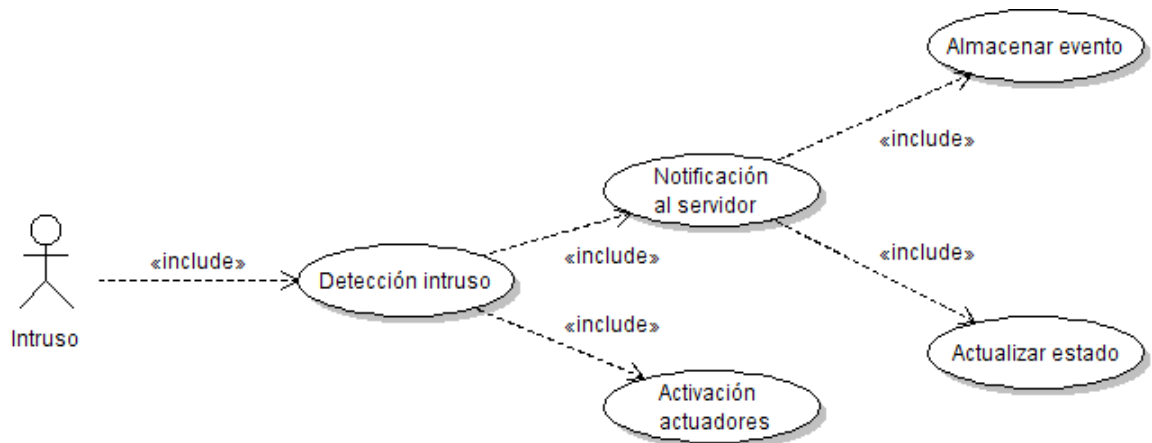


Ilustración 18. Caso de uso detección de intrusión en el perímetro

#### 3.3.2.2. Consulta del estado del sistema

R.F	3. Consulta del estado del sistema
<b>Introducción</b>	El sistema permite ver si el sistema de alarma esta encendido o apagado y si algún sensor de la alarma está activado.
<b>Entrada</b>	Selección de la pestaña de estado en la interfaz web.
<b>Proceso</b>	El servidor consulta el estado actual del sistema
<b>Salida</b>	Se muestra en la interfaz web el estado actual de la alarma y el estado actual de los sensores

Tabla 7. Requisito funcional consulta del estado del sistema

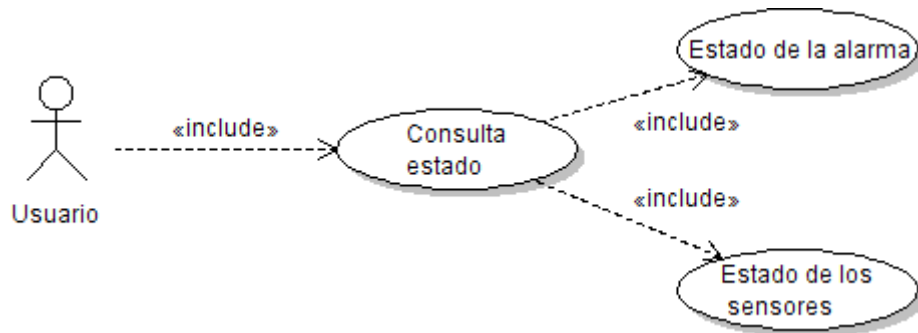


Ilustración 19. Caso de uso consulta del estado del sistema

### 3.3.2.3. Consulta del historial de eventos del sistema

R.F	4. Consulta del historial de eventos del sistema
<b>Introducción</b>	El sistema permite ver el historial de eventos que se han producido.
<b>Entrada</b>	Selección de la pestaña de historial en la interfaz web.
<b>Proceso</b>	Carga el historial de eventos de la base de datos.
<b>Salida</b>	Se muestra en la interfaz web el historial de eventos del sistema

Tabla 8. Requisito funcional consulta del historial de eventos del sistema

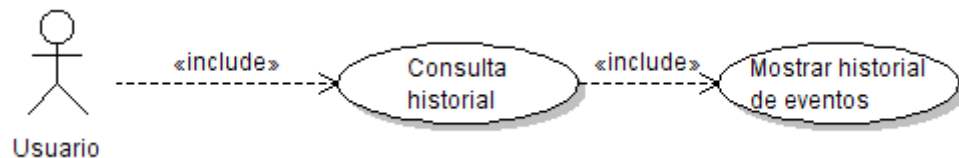


Ilustración 20. Caso de uso consulta del historial de eventos del sistema

### 3.3.2.4. Activar/Desactivar el sistema de seguridad

R.F	5. Activar/Desactivar el sistema de seguridad
<b>Introducción</b>	El sistema permite activar y desactivar el sistema de seguridad de forma remota desde la aplicación web.
<b>Entrada</b>	Acción a realizar indicada en la aplicación web.
<b>Proceso</b>	Envío de acción al sistema Arduino
<b>Salida</b>	Se envía la acción de encender o apagar la alarma al Arduino. Después se muestra en la interfaz web el estado actual del sistema tras el envío de la acción al sistema Arduino.

Tabla 9. Requisito funcional activar/desactivar el sistema de seguridad



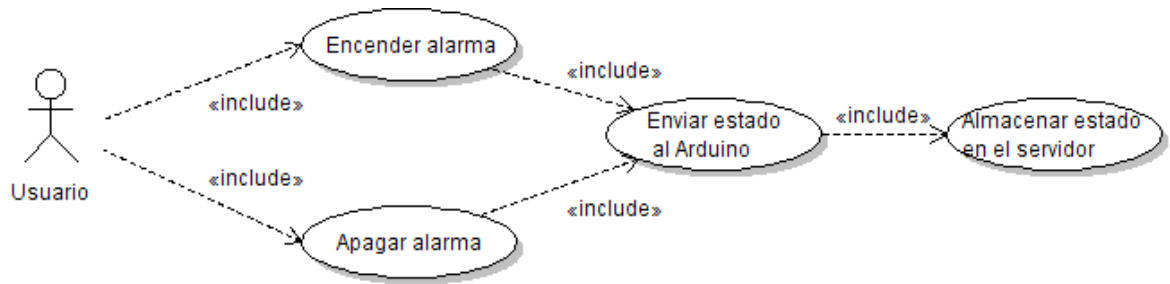


Ilustración 21. Caso de uso activar/desactivar el sistema de seguridad

### 3.3.2.5. Modificación de la IP del servidor

R.F	6. Gestión del sistema de seguridad
<b>Introducción</b>	El sistema permite modificar la IP del servidor y enviársela al sistema Arduino para que pueda comunicarse con él.
<b>Entrada</b>	Nueva IP del servidor.
<b>Proceso</b>	Envío de la IP al sistema Arduino
<b>Salida</b>	La IP del servidor se actualiza en la base de datos mostrándola en la interfaz web.

Tabla 10. Requisito funcional gestión de la IP del servidor

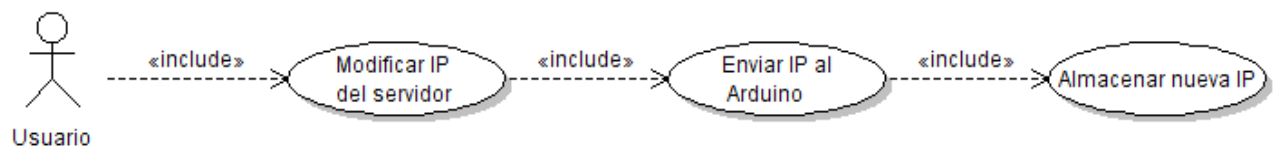


Ilustración 22. Caso de uso modificación de la IP del servidor

### 3.3.2.6. Modificación de la IP del sistema Arduino

R.F	7. Modificación de la IP del sistema Arduino
<b>Introducción</b>	El sistema permite modificar la IP del sistema Arduino y almacenarla en el servidor para poder comunicarse con él.
<b>Entrada</b>	Nueva IP del sistema Arduino.
<b>Proceso</b>	Almacenamiento de la IP en la base de datos.
<b>Salida</b>	La nueva IP se visualiza en la interfaz web.

Tabla 11. Requisito funcional modificación de la IP del sistema Arduino

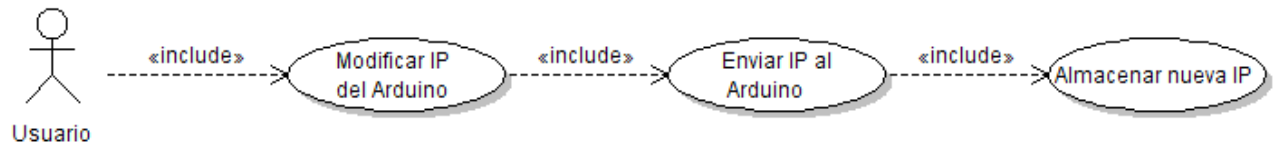


Ilustración 23. Caso de uso modificación de la IP del sistema Arduino

### 3.3.2.7. Gestión de los sensores del sistema

R.F	8. Gestión de los sensores del sistema
<b>Introducción</b>	El sistema permite añadir, eliminar o modificar los sensores que se van a conectar al sistema de alarma.
<b>Entrada</b>	Sensor que queremos añadir, borrar o eliminar.
<b>Proceso</b>	El servidor envía los nuevos cambios al sistema Arduino para que actualice sus entradas y los almacena en la base de datos.
<b>Salida</b>	Los cambios en los sensores se ven reflejados en la interfaz web.

Tabla 12. Requisito funcional de la gestión de los sensores del sistema.

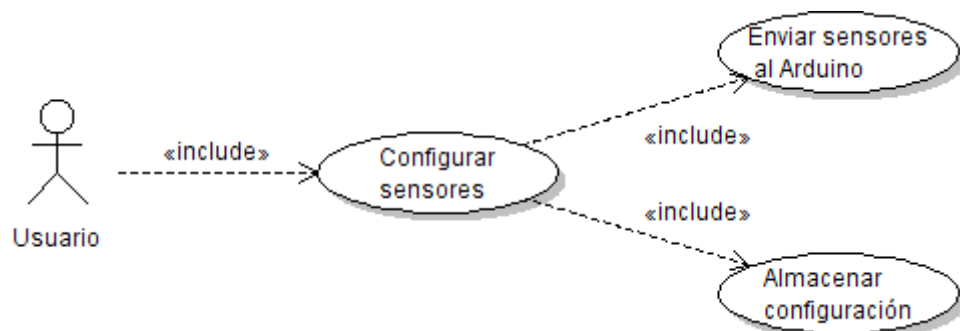


Ilustración 24. Caso de uso gestión de los sensores del sistema

### 3.3.2.8. Gestión de los actuadores del sistema

R.F	9. Gestión de los actuadores del sistema
<b>Introducción</b>	El sistema permite añadir, eliminar o modificar los actuadores que se van a conectar al sistema de alarma.
<b>Entrada</b>	Actuador que queremos añadir, borrar o eliminar.
<b>Proceso</b>	El servidor envía los nuevos cambios al sistema Arduino para que actualice sus salidas y los almacena en la base de datos.
<b>Salida</b>	Los cambios en los sensores se ven reflejados en la interfaz web.

Tabla 13. Requisito funcional gestión de los actuadores del sistema



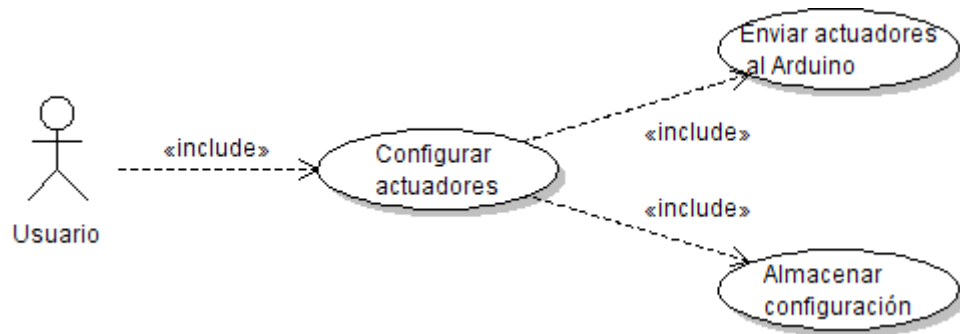


Ilustración 25. Caso de uso gestión de los actuadores del sistema

### 3.3.2.9. Gestión de las acciones del sistema

R.F	10. Gestión de las acciones del sistema
<b>Introducción</b>	El sistema permite asignar que actuadores se activaran cuando se activen cada uno de los sensores.
<b>Entrada</b>	Sensor relacionado con los actuadores.
<b>Proceso</b>	El servidor envía los nuevos cambios al sistema Arduino para que sepa que acciones tiene que realizar en caso de activación de un sensor y los almacena en base de datos.
<b>Salida</b>	Las acciones a realizar se mostraran en la interfaz de la aplicación web.

Tabla 14. Requisito funcional gestión de las acciones del sistema

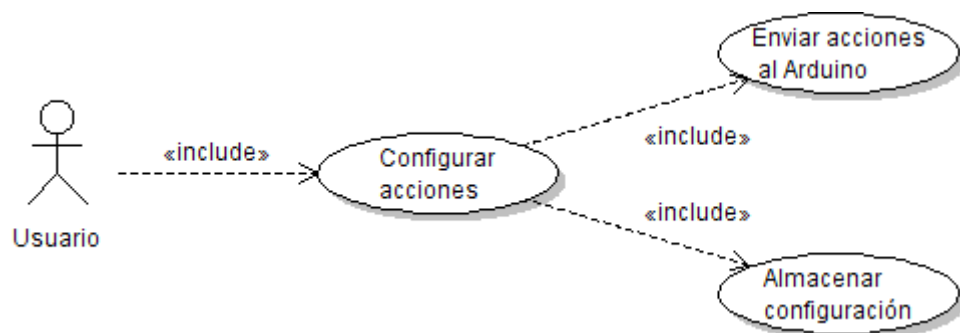


Ilustración 26. Caso de uso gestión de las acciones del sistema

### 3.3.2.10. Paro de la alarma

R.F	11. Paro de la alarma
<b>Introducción</b>	El sistema permite parar la alarma en caso de que este activada.
<b>Entrada</b>	Botón de parar la alarma en la interfaz web.

<b>Proceso</b>	El servidor envía al Arduino la acción de parar la alarma y almacena el estado en el servidor.
<b>Salida</b>	El sistema Arduino desactiva los actuadores que estaban activos.

Tabla 15. Requisito funcional paro de la alarma

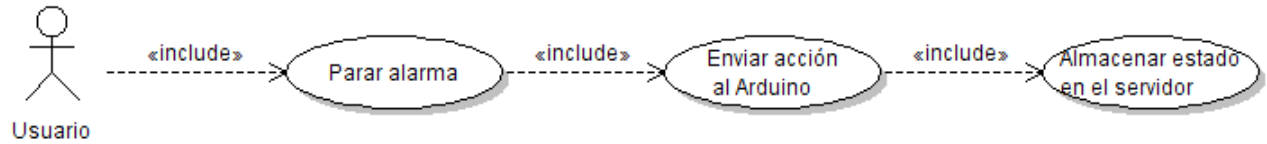


Ilustración 27. Caso de uso paro de la alarma

### 3.3.3. Requisitos no funcionales

#### 3.3.3.1. Requisitos de rendimiento

El tiempo de respuesta entre el Arduino y el servidor será de 3 segundos como máximo en al menos el 98% de las peticiones.

El servidor web permite la conexión de 5 usuarios simultáneamente.

#### 3.3.3.2. Seguridad

No será posible el acceso al sistema de seguridad desde fuera de la red local, en la que se encuentra el servidor y el Arduino, para garantizar mayor seguridad. Aunque el Arduino pierda la conexión con el servidor este seguirá igualmente accionando los actuadores de aviso de alarma.

La seguridad ante la manipulación de los sensores y el microcontrolador deberá ser garantizada por el instalador del sistema.

#### 3.3.3.3. Fiabilidad

Solo se permitirá un fallo del sistema cada 5000 horas de funcionamiento.

La fiabilidad de los componentes instalados como sensores o actuadores vendrá dada por el fabricante de dicho componente.

#### 3.3.3.4. Disponibilidad

El servicio deberá estar disponible el 99,9% del tiempo.

#### 3.3.3.5. Mantenibilidad

No será necesario que el usuario realice ningún mantenimiento del sistema. El mantenimiento del servidor será realizado por el desarrollador mediante actualizaciones.



### 3.3.3.6. Portabilidad

El servicio web y la base de datos podrán moverse sin problemas a otro servidor que tenga instalado el motor de bases de datos MySQL y un servidor web PHP 5.5



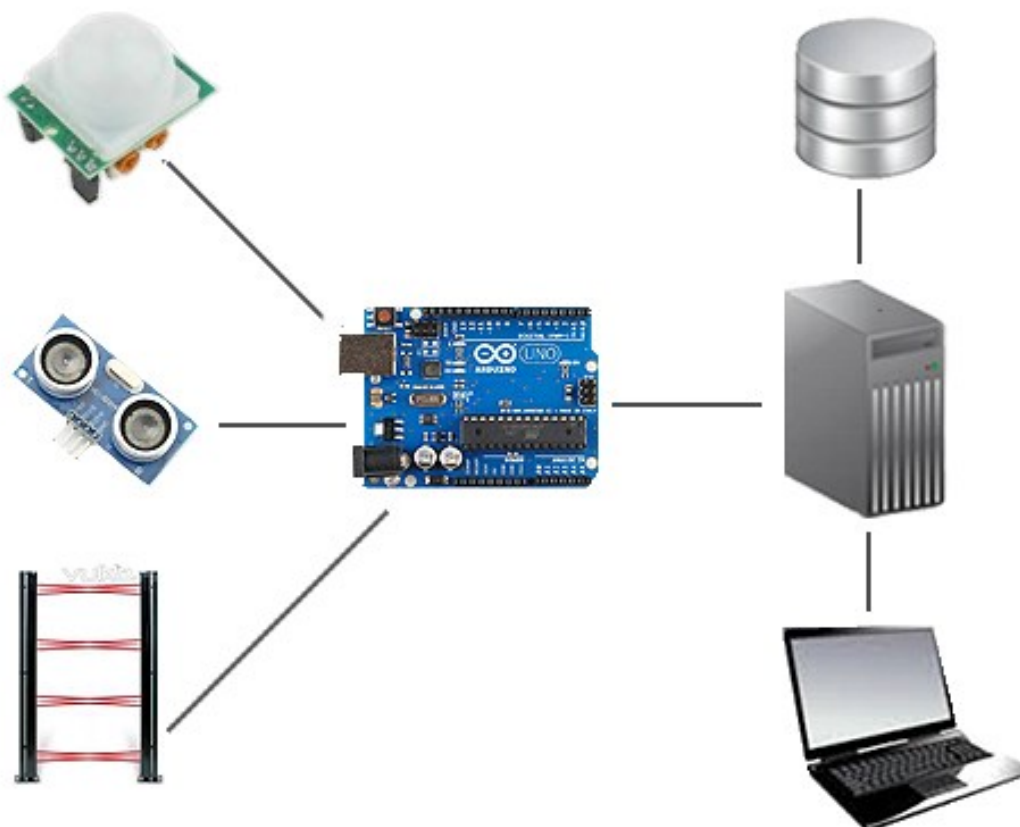
## 4. Diseño del sistema

### 4.1. Introducción

En este apartado se tratará el diseño del sistema desarrollado. Primero se mostrará un modelo conceptual que servirá como primera aproximación lo que será el sistema. Después analizaremos el diseño hardware con un esquema hardware y un esquema eléctrico donde veremos todos los componentes que forman el sistema y como están interconectados. Luego mediante diagramas de secuencia se mostrará la interacción llevada a cabo por las distintas funcionalidades que implementa el sistema. Por último a través de un diagrama de flujo se representará gráficamente el proceso que lleva a cabo el sistema Arduino.

### 4.2. Descripción conceptual

El siguiente esquema muestra una descripción conceptual del sistema implementado.



**Ilustración 28. Descripción conceptual del sistema diseñado**

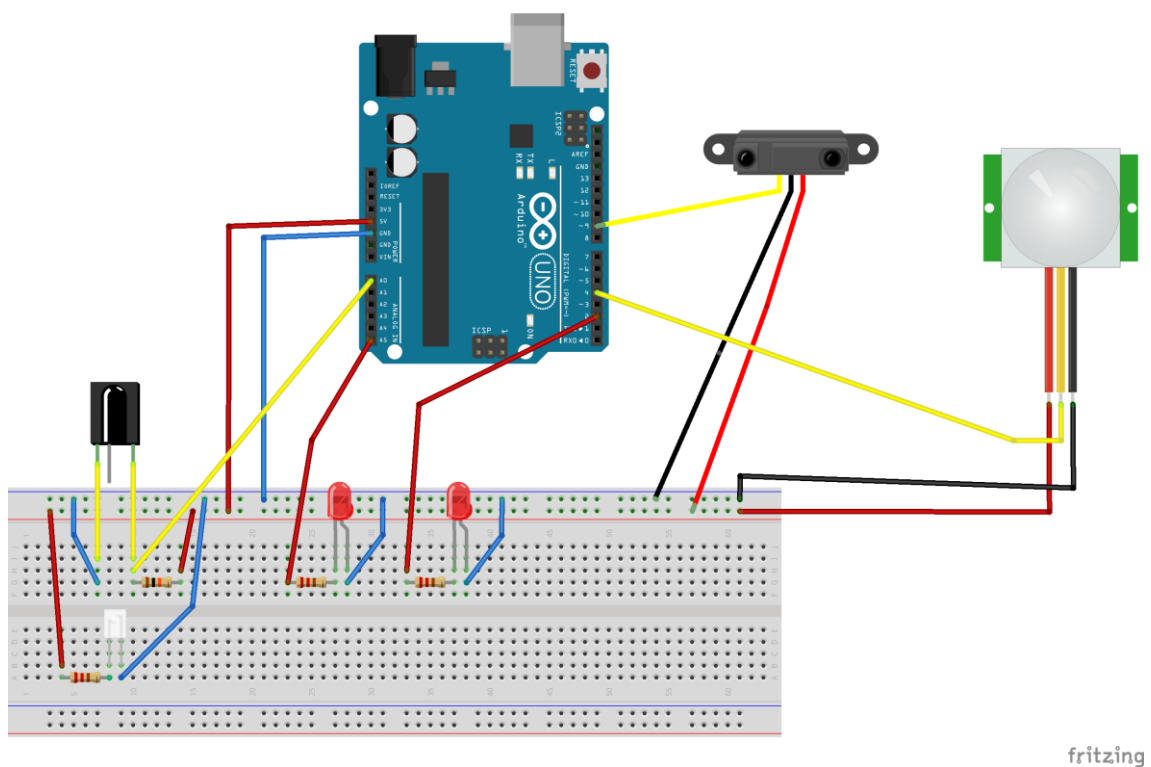
En el esquema se puede ver como el Arduino forma la parte central del sistema de seguridad. A su izquierda se pueden ver los sensores que se conectan al Arduino para la detección de los peligros. El sistema Arduino será el encargado de procesar las señales que llegan desde los sensores para comprobar si alguno se está activado.

El Arduino se conectará a través de Ethernet a un servidor con el que se comunicará tanto para enviarle información como para recibirla. Este servidor ofrecerá una aplicación web a los clientes que se conecten a él. A través de esta aplicación el cliente podrá gestionar el sistema de alarma. El servidor también dispondrá de una base de datos relacional para almacenar la información.

### 4.3. Diagrama hardware

#### Esquema hardware

A continuación se muestra el esquema hardware donde se aprecia la forma de conexión de los diferentes componentes del sistema. Para simplificar el esquema solo aparece un sensor de cada tipo, el resto de sensores se conectarán al sistema de la misma forma.



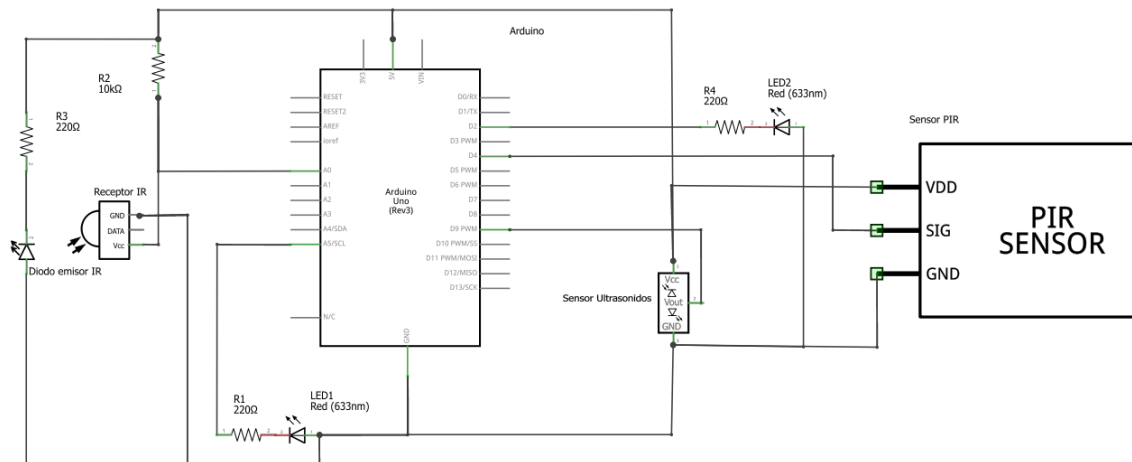
**Ilustración 29. Esquema hardware del sistema**

En el esquema vemos que los sensores de ultrasonidos y los sensores PIR se conectan a las salidas de digitales de Arduino. En cambio la barrera de infrarrojos se ha de conectar a las entradas analógicas. Los actuadores se podrán conectar tanto a las salidas analógicas como a las digitales siempre que funcionen a 5v.

Para simular las luces de aviso se han empleado diodos led. Estos diodos necesitan una resistencia al igual que los sensores emisor y receptor de infrarrojos.

## Esquema eléctrico

La siguiente imagen muestra el esquema eléctrico del sistema de seguridad mostrado en el anterior esquema.



**Ilustración 30. Esquema eléctrico del sistema diseñado**

El esquema muestra los componentes eléctricos y electrónicos que forman el sistema. Tanto para los diodos led como el diodo emisor de infrarrojos será necesario colocar una resistencia de aproximadamente 220 ohmios. El receptor de infrarrojos necesitará una resistencia de 10k ohmios que se situará antes de su entrada de corriente. Entre esta resistencia y la entrada realizará el Arduino las medidas de voltaje.

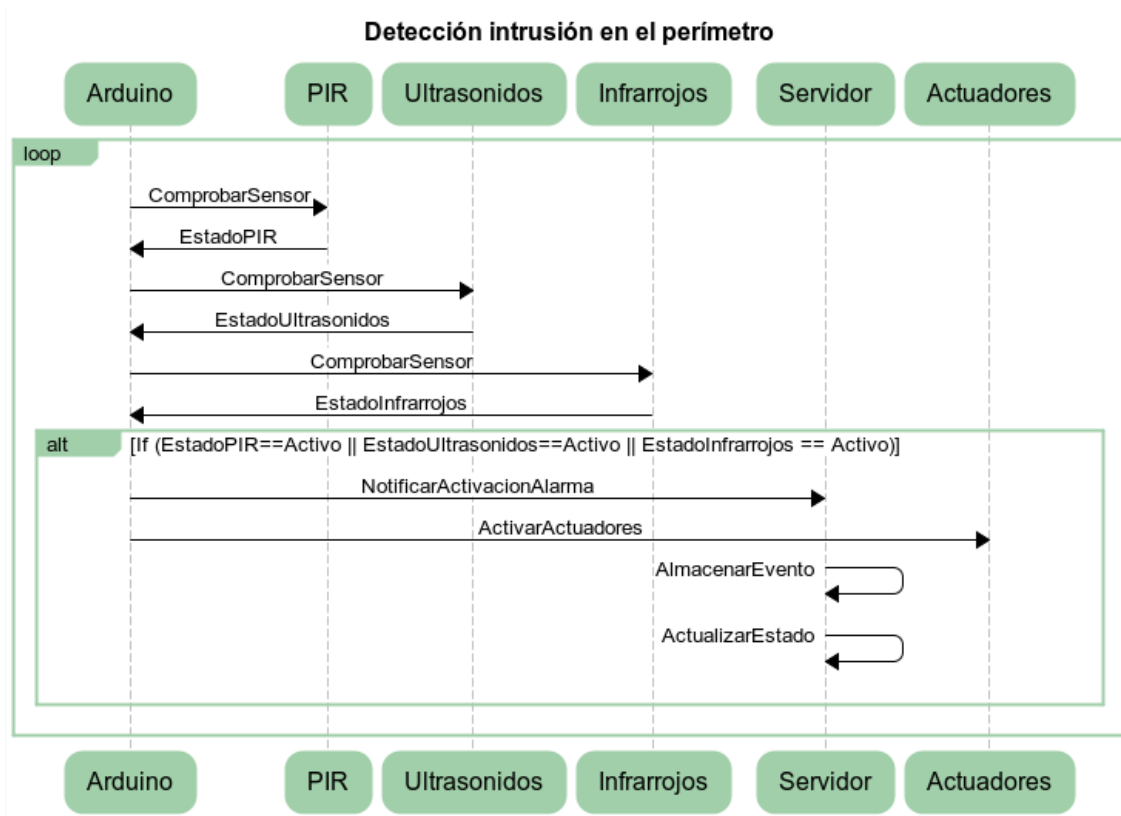
Las señales de control de los sensores PIR y de ultrasonidos se conectarán directamente al Arduino sin la necesidad de ningún otro componente eléctrico. Los distintos componentes se podrán alimentar a través del Arduino siempre que el consumo total no supere el máximo capaz de suministrar por el Arduino.

## 4.4. Especificación software

En este apartado mostraremos mediante diagramas de secuencias la interacción de los objetos en las distintas funcionalidades del sistema software.

### 4.4.1. Detección intrusión en el perímetro

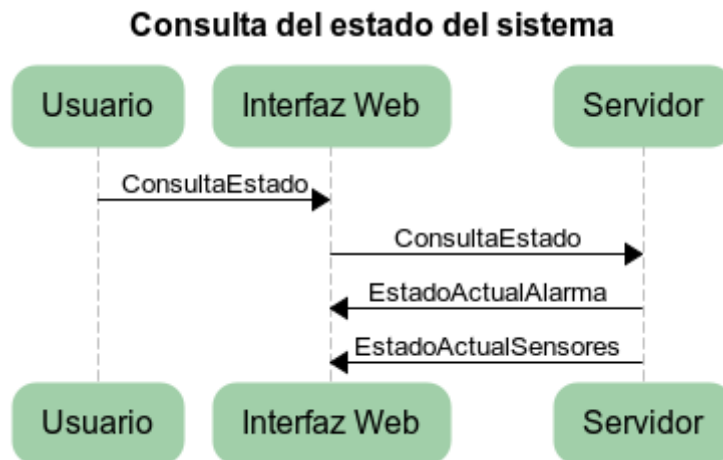




**Ilustración 31. Diagrama de secuencia. Detección intrusión en el perímetro**

En la tarea de detectar intrusos en el perímetro se realiza un bucle continuo donde se comprueban todos los sensores que hay en el sistema. Si algún sensor está activo el Arduino lo comunica al servidor vía Ethernet y activa los actuadores asociados a los sensores activos. El servidor se encargará de almacenar en la base de datos el evento de alarma activada y actualizará el estado de la alarma y de los sensores.

#### 4.4.2. Consulta del estado del sistema



**Ilustración 32. Diagrama de secuencia. Consulta del estado del sistema**

Para consultar el estado de la alarma el usuario accede a la pestaña de estado en la interfaz. Entonces se realiza una consulta al servidor que lee el estado de la BBDD y devuelve tanto el estado general de la alarma como el de cada uno de los sensores conectados. Los datos se visualizan en la interfaz web.

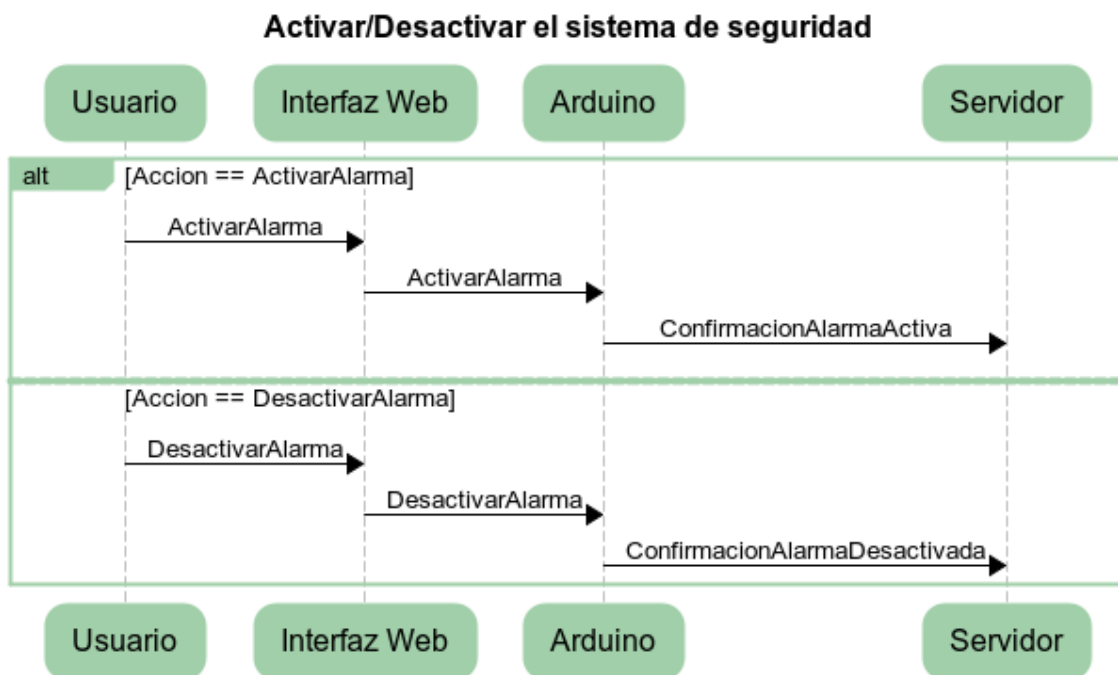
#### 4.4.3. Consulta del historial de eventos del sistema



**Ilustración 33. Diagrama de secuencia. Consulta del historial de eventos del sistema**

Para consultar el historial de eventos el usuario selecciona la pestaña de historial en la interfaz. La aplicación web realiza una consulta al servidor y este lee el historial de la base de datos y lo devuelve para visualizarlo en la interfaz web.

#### 4.4.4. Activar/Desactivar el sistema de seguridad

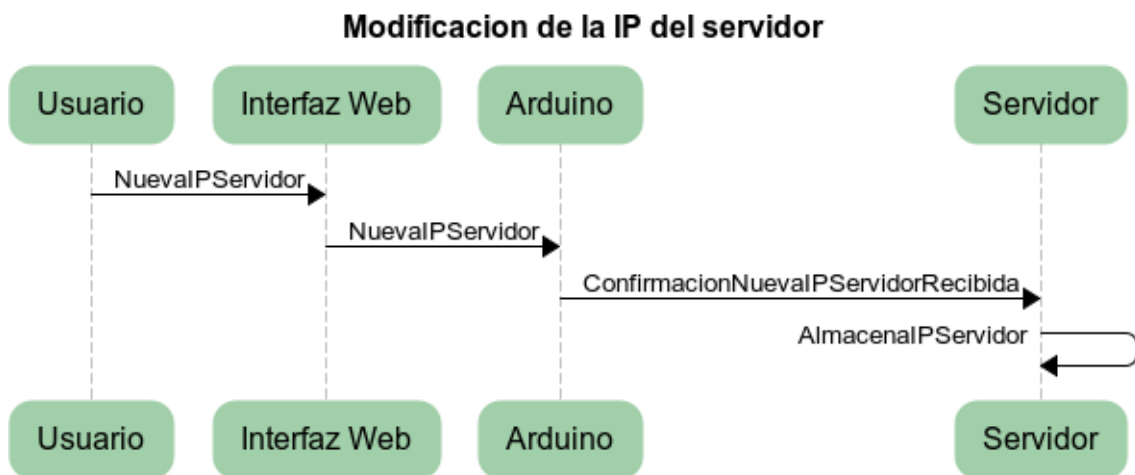


**Ilustración 34. Diagrama de secuencia. Activar/Desactivar el sistema de seguridad**



Para activar o desactivar el sistema de alarma el usuario seleccionará la opción en la interfaz web. La acción se le enviará al Arduino y la procesará realizando las acciones oportunas. Tras esto el Arduino enviará la confirmación al servidor de que ha recibido y procesado correctamente la petición de activación o desactivación. El servidor almacenará el estado en la BBDD.

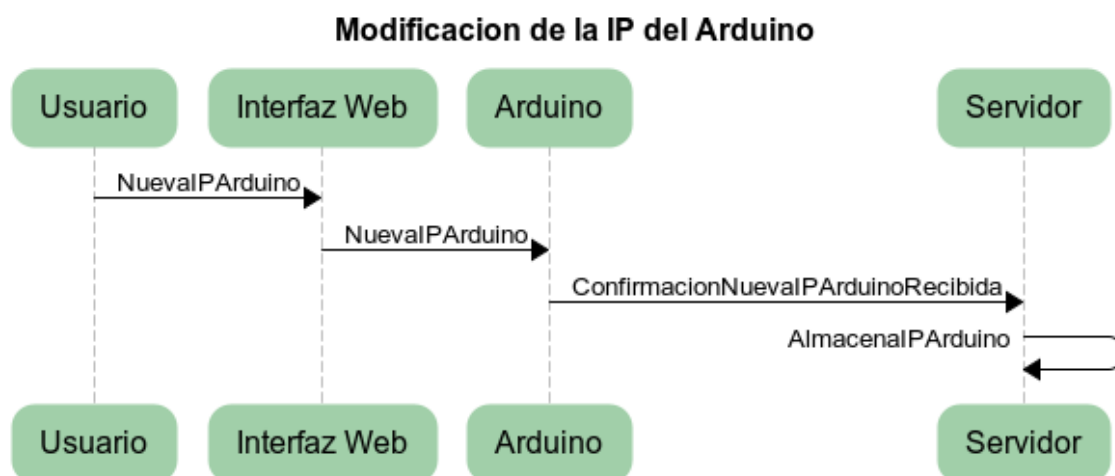
#### 4.4.5. Modificación de la IP del servidor



**Ilustración 35. Diagrama de secuencia. Modificación de la IP del servidor.**

Si se desea modificar la IP del servidor, el usuario introducirá en la interfaz la nueva IP. Esta IP se enviará al Arduino que la almacenará en su memoria y la empleará para acceder al servidor. Tras esto enviará un mensaje de confirmación de recepción de la nueva IP al servidor y este la almacenará en la BBDD.

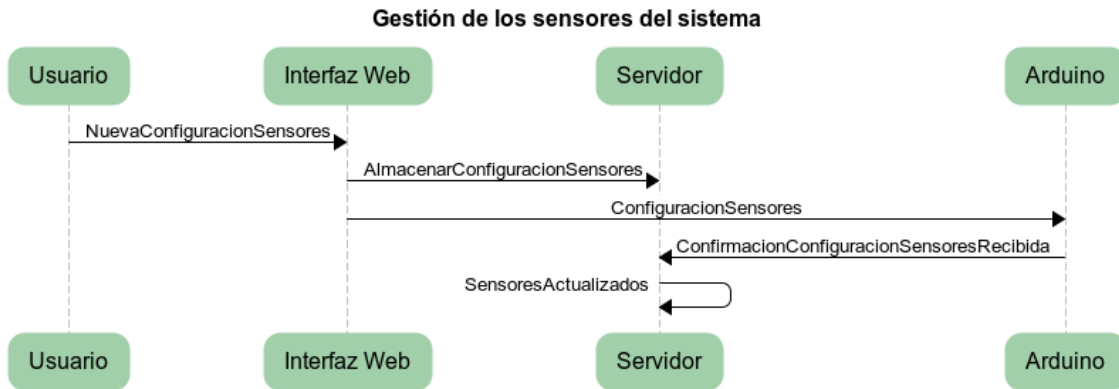
#### 4.4.6. Modificación de la IP del sistema Arduino



**Ilustración 36. Diagrama de secuencia. Modificación de la IP del sistema Arduino.**

Para modificar la IP del Arduino el usuario introducirá la nueva IP en la interfaz web y esta la enviará al Arduino. El Arduino actualizará su propia IP y enviará la confirmación al servidor de IP recibida. Tras recibir el mensaje el servidor actualizará la IP del Arduino en la base de datos.

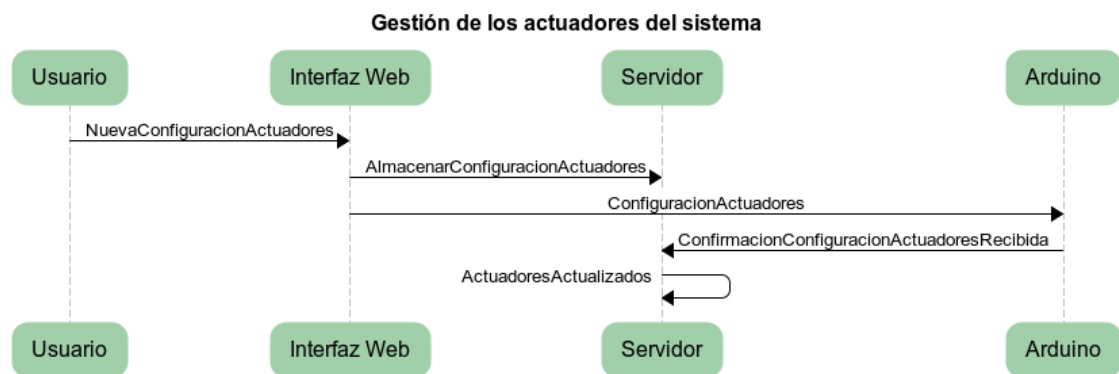
#### 4.4.7. Gestión de los sensores del sistema



**Ilustración 37. Diagrama de secuencia. Gestión de los sensores del sistema.**

Para añadir, modificar o eliminar un sensor del sistema, el usuario accederá a la sección de gestión de sensores en la interfaz. Aquí realizará las acciones convenientes y enviará la nueva configuración al Arduino y al servidor que la almacenará en la BBDD. El Arduino actualizará sus sensores según la configuración recibida y enviará un mensaje de confirmación de que ha recibido y configurado correctamente los sensores.

#### 4.4.8. Gestión de los actuadores del sistema

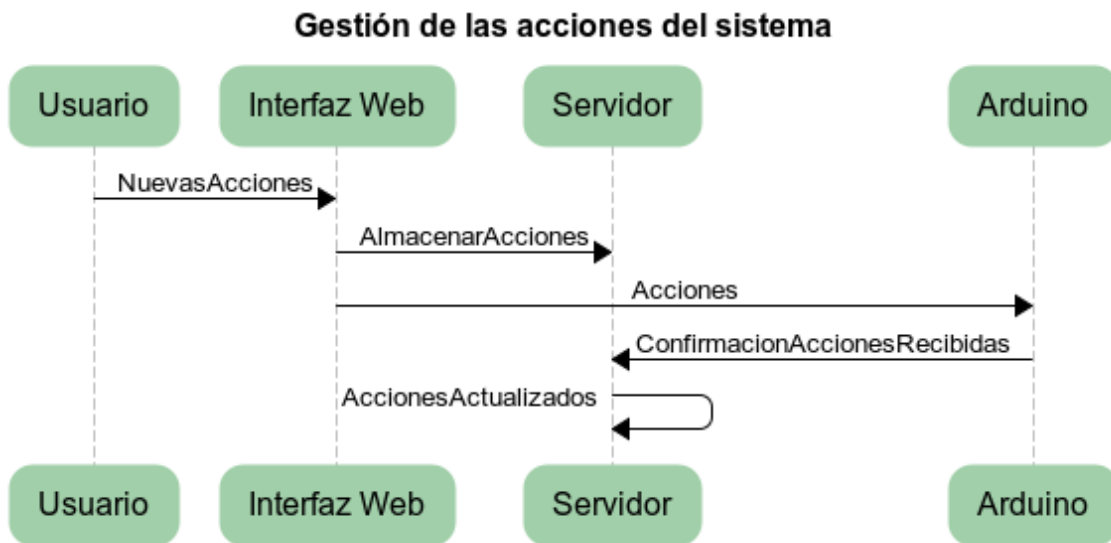


**Ilustración 38. Diagrama de secuencia. Gestión de los actuadores del sistema.**

Para modificar la configuración de los actuadores el usuario realiza los cambios pertinentes a través de la interfaz web. Después se envía la nueva configuración al servidor, para que la almacene en la base de datos, y al Arduino para que cambie su configuración de los actuadores. Para verificar que el Arduino ha recibido la configuración este envía un mensaje de confirmación al servidor.



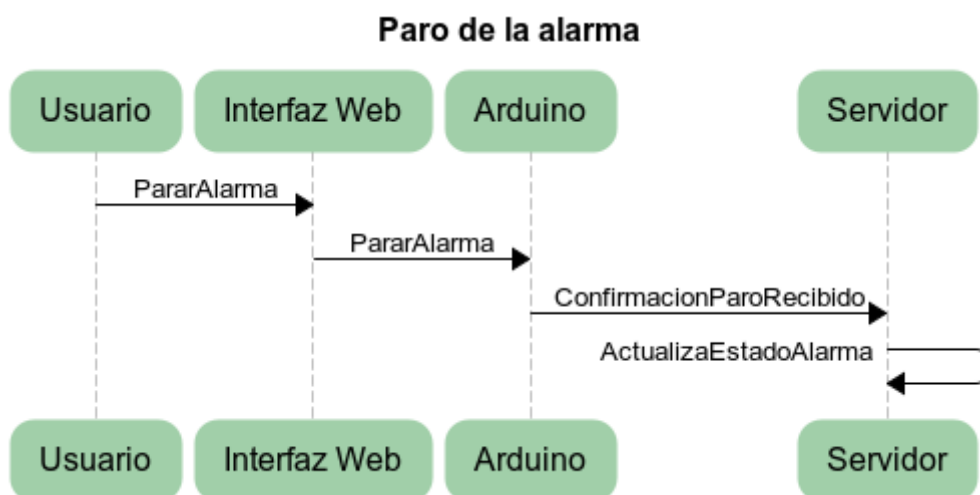
#### 4.4.9. Gestión de las acciones del sistema



**Ilustración 39. Diagrama de secuencia. Gestión de las acciones del sistema.**

Para configurar las acciones que hay que realizar cuando se activa un determinado sensor el usuario realizará la asociación de sensores y actuadores en la interfaz. Después se enviará esta configuración al Arduino para que la almacene en su memoria. Tras esto enviará el mensaje de confirmación de la recepción de las acciones al servidor.

#### 4.4.10. Paro de la alarma



**Ilustración 40. Diagrama de secuencia. Paro de la alarma.**

El paro de la alarma comienza cuando el usuario selecciona la opción en la interfaz web. Después se envía la orden de paro de la alarma al Arduino que desactiva todos los actuadores activados. A continuación envía la confirmación de alarma parada al servidor que actualiza el estado en la base de datos.



## 4.5. Diagrama UML del Arduino

A continuación se muestra el diagrama de flujo del programa que ejecuta el Arduino:

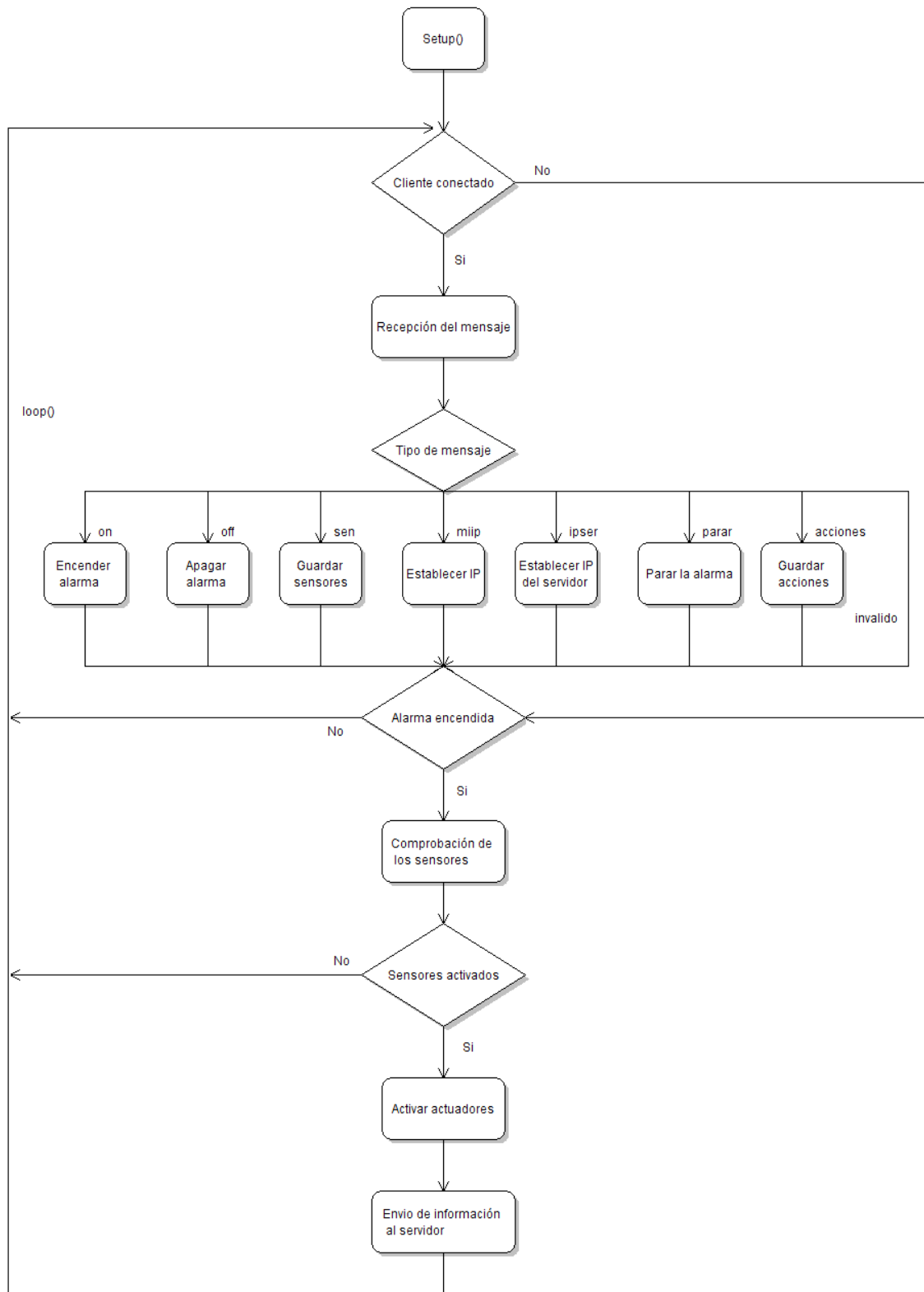


Ilustración 41. Diagrama de flujo del Arduino



El diagrama de flujo muestra el proceso que lleva a cabo el microcontrolador Arduino. El flujo comienza con la ejecución de un primer método de configuración del sistema (setup), donde se cargan de la memoria los distintos parámetros de sistema y configura las entradas y salidas del microcontrolador. Una vez se ha realizado la configuración se entra en el bucle principal del programa que se repetirá continuamente. La primera parte del bucle se encarga de comprobar si se ha recibido alguna petición GET al servidor que implementa el Arduino. Si ha recibido alguna petición este se encarga de leer los parámetros y comprueba si coinciden con alguno de los previstos. Si es así obtiene la información, la procesa y ejecuta las acciones correspondientes. Por ejemplo si se le envía una petición con el parámetro *ipser* indica que se le está enviando la IP del servidor. Entonces el Arduino obtendrá la IP y la actualizará en su memoria para así poder hacer referencia a la nueva IP recibida.

Luego en la segunda parte de bucle comprueba el estado de los sensores conectados, para ello comprobará cada uno de los distintos sensores. Si alguno de ellos está activado se pasará al estado de activación de los actuadores asociados a los sensores que hayan sido activados. Después se enviará un mensaje al servidor con la información del sensor que está activo. Tras estas acciones comenzaría de nuevo la ejecución del bucle. Si no hay ningún sensor activado directamente se volvería al comienzo del bucle.

## 4.6. Conclusiones

Los esquemas analizados en esta sección han tenido como objetivo describir de una forma visual el diseño del sistema de alarma así como también mostrar el flujo de la aplicación y de las funcionalidades mediante diagramas UML. Una vez conocido como será el sistema físico podremos pasar a ver como se ha realizado la implementación.

# 5. Implementación

## 5.1. Introducción

A continuación se va a analizar cómo se ha implementado desde un punto de vista software el sistema de alarma que hemos visto en las secciones anteriores.

Comenzaremos con la implementación de los diferentes elementos que lo componen como la base de datos, la programación del Arduino, el servidor y la aplicación web. Entraremos en detalle en algunas de las partes que se consideren de mayor interés con pequeños fragmentos de código e ilustraciones.

Después veremos los prototipos que se ha empleado durante todo el desarrollo. Todo esto servirá para hacer un seguimiento de las diferentes etapas de desarrollo.

Por último veremos los aspectos necesarios para realizar la implementación del sistema de forma real.

## 5.2. Implementación

En esta sección explicaremos como se ha realizado la implementación de los módulos más importantes del proyecto.

### 5.2.1. Base de datos

El sistema de gestión de bases de datos empleado en el proyecto ha sido MySQL. Se trata de un gestor de base de datos de software libre muy empleado en aplicaciones web. Para el acceso y manejo de la base de datos se ha utilizado la herramienta phpMyAdmin también con licencia GPL.

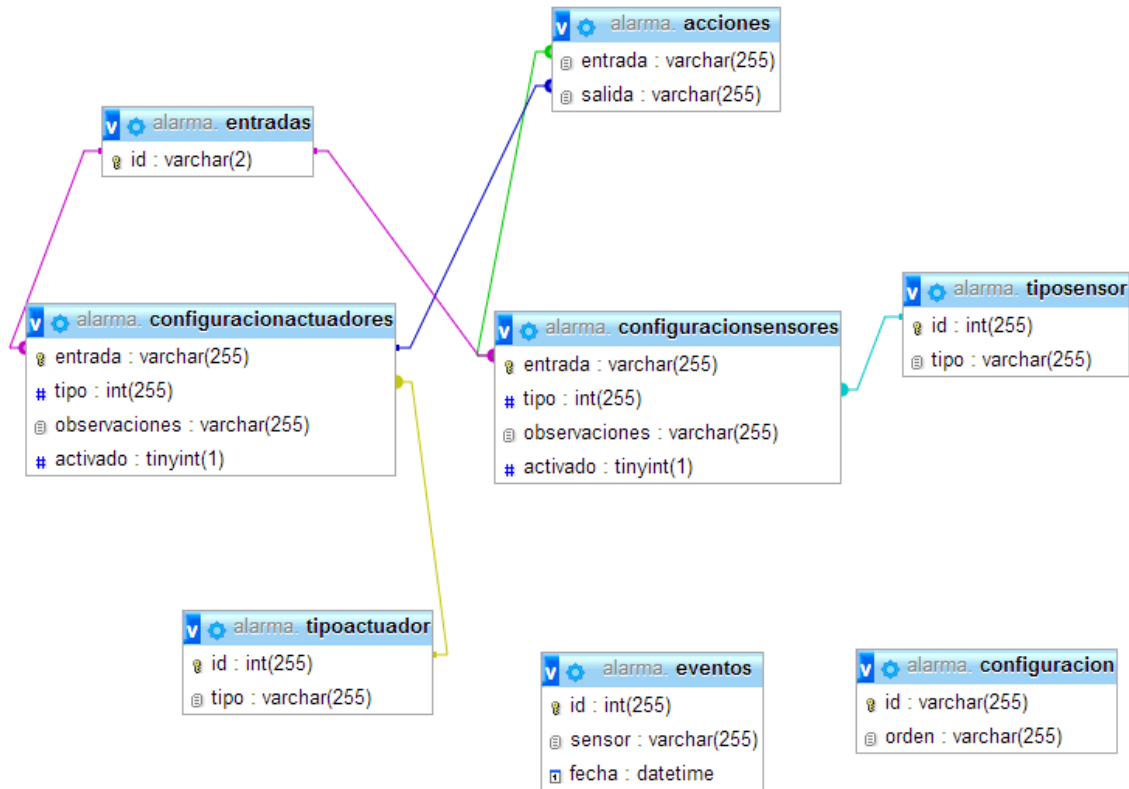
La base de datos se ha empleado en este proyecto para dos tareas principalmente:

- Almacenamiento de los parámetros de la aplicación y del Arduino como son las entradas, salidas, acciones y configuración de los distintos sensores y actuadores.
- Registro de eventos del sistema y trazabilidad de los mismos.

La base de datos es sencilla, contiene pocas tablas y estas a su vez contienen pocos datos en su interior ya que la aplicación no requiere un gran uso de datos almacenados.

La siguiente ilustración muestra gráficamente las tablas que componen la base de datos con las relaciones existentes entre ellas.





**Ilustración 42. Esquema de las tablas que componen la BBDD**

Vamos a ver con más detenimiento que contiene cada una de las tablas:

### ***Entradas***

Contiene todo el conjunto de pines del Arduino identificados con el número que aparece en la placa.

### ***Tiposensor***

Contiene los distintos tipos de sensores que hay disponibles en el sistema.

### ***Tipoactuador***

Contiene los distintos tipos de actuadores que hay disponibles en el sistema.

### ***Configuracionactuadores***

Contiene los actuadores que hay en el sistema. Cada entrada de la tabla contiene el pin del Arduino donde está conectado, el tipo de actuador, unas observaciones y un indicador para saber si está activo.

### ***Configuracion sensores***

Contiene los sensores que hay en el sistema. Cada entrada de la tabla contiene el pin del Arduino donde está conectado, el tipo de sensor, unas observaciones y un indicador para saber si está activo.

### ***Acciones***

Relaciona los sensores con los actuadores para saber que actuadores hay que activar cuando se activa un sensor determinado. Un sensor puede tener muchos actuadores relacionados.

### **Configuración**

Contiene parámetros de configuración de la aplicación como la IP del servidor, la IP del Arduino o el estado de la alarma, si está activada o desactivada.

### **Eventos**

Contiene el registro de los eventos de activación de la alarma que se han producido en el sistema. Se almacena el sensor implicado y la fecha y hora.

Esta base de datos será empleada por el servidor web almacenar información que introduzcamos en la aplicación web y que posteriormente se enviará al Arduino para que realice las acciones oportunas.

## **5.2.2. Programación del Arduino**

En esta sección vamos a ver la implementación del diagrama de flujo mostrado en la Ilustración 41. Diagrama de flujo del Arduino donde se muestra todo el proceso que lleva a cabo el Arduino.

El programa desarrollado en el Arduino cuenta con 2 partes principales: la inicialización (Setup) y el bucle (Loop). Vamos a ver con más detenimiento cada una de estas partes:

### **Setup**

Esta parte se ejecuta una única vez al comienzo de la ejecución del Arduino. Es la encargada de realizar la inicialización de diferentes parámetros y módulos que se emplearán más adelante.

En el proyecto esta parte será la encargada de la inicialización del servidor web que implementa el Arduino y de la carga de parámetros almacenados en la memoria EEPROM del Arduino. Los datos que se almacenan en la memoria son:

- IP del Arduino
- IP del servidor
- Estado del sistema (ON/OFF)
- Sensores PIR conectados
- Sensores de ultrasonidos conectados
- Sensores de infrarrojo conectados
- Actuadores a activar según el sensor activado

Para saber que sensores hay en el sistema se almacena el pin donde están conectados para cada tipo. Para saber las acciones que hay que realizar se almacenan parejas de pines donde el primero es el sensor activo y el segundo el actuador que hay que activar.

Con todos estos datos se inicializa el servidor y se configuran los diferentes pines del Arduino según su función.

### **Loop**



Esta parte se ejecuta dentro de un bucle infinito, será la parte encargada de controlar el sistema de alarma y de tomar acciones a diferentes sucesos. A su vez esta parte se compone de dos partes, la primera será la encargada de recibir y tratar los mensajes que le llegan al servidor web que implementa Arduino. La segunda parte del bucle será la encargada de detectar si alguno de los sensores de la alarma se ha activado.

### **Servidor**

La función del servidor web que implementa Arduino será recibir una serie de parámetros y órdenes del servidor como pueden ser activar o desactivar la alarma, la IP del servidor, su propia IP o la configuración de los sensores y actuadores.

Esta información será enviada por el servidor principal del sistema de alarma mediante peticiones http de tipo GET, por lo que el Arduino recibirá la información a través de parámetros en la URL. Por cada petición que llegue se realizará una búsqueda de los parámetros de la URL para obtener su contenido y poder almacenarlo en la memoria EEPROM. Por ejemplo una petición para encender la alarma sería:

*IPdelArduino?accion=ON*

Al realizar esta petición el servidor Arduino buscaría si existe el parámetro *acción* y obtendría su valor para analizar si es una instrucción conocida. El siguiente fragmento es el encargado de realizar esta tarea:

```
int accion = readString.indexOf("accion=");
if(accion!=-1){
  if(readString.substring(accion,accion+9)=="accion=ON")
  {
    Serial.println("Alarma ON");
    EEPROM.write(8, 1);
    state="ON";
    respuesta("?accion=ON");
  }
  else if (readString.substring(accion,accion+10)=="accion=OFF")
  {
    Serial.println("Alarma OFF");
    EEPROM.write(8, 0);
    state="OFF";
    respuesta("?accion=OFF");
  }
}
```

**Código 1. Recepción de las acciones en Arduino.**

Si analizamos las acciones que realiza al recibir una petición de este tipo vemos que primero notifica por el puerto serial que ha recibido la petición. Después actualiza el estado y lo almacena en la EEPROM. Por último lanza una respuesta http al servidor para notificar que ha recibido y procesado la petición correctamente.

Otro ejemplo información que podemos enviarle al Arduino sería la configuración de los sensores:

*IPdelArduino?pir=3629*

En este caso se le ha enviado la configuración de los sensores PIR donde el primer número, el 3, indica el número de sensores PIR que hay en el sistema. Los siguientes

números indican el pin del Arduino donde irán conectados los sensores, en este caso irán al pin 6, 2 y 9. El código encargado de recibir la petición es el siguiente:

```
int posicion = readString.indexOf("pir=");

if(posicion!=-1){
  string=readString.substring(posicion+4);
  byte aux;
  byte i=0;
  Serial.println("\nPIR");
  aux=string.charAt(0)-48;
  EEPROM.write(40, aux);

  while(i<aux){
    Serial.print(string.charAt(i+1));
    i++;
    EEPROM.write(40+i,((String) string.charAt(i)).toInt());
  }
  cargarDatos();
}
```

### **Código 2. Recepción de los sensores PIR.**

Siempre que se reciba una petición que modifique parámetros de la aplicación se realizaran las acciones oportunas para que los cambios se tengan en cuenta en la siguiente ejecución del bucle del programa.

#### **Alarma**

La segunda parte del bucle será la encargada de comprobar el estado de cada uno de los sensores que componen el sistema. Para ello se ha implementado un método de comprobación del estado del sensor distinto según el tipo de sensor disponible, en este caso se ha implementado uno para sensor PIR, otro para el sensor de infrarrojos y otro para el sensor de ultrasonidos. El siguiente código muestra el método de comprobación para los sensores de ultrasonidos:

```
void compruebaUltrasonidos(byte pin){

  while(j<10){

    pinMode(pin, OUTPUT); // ponemos el pin como salida
    digitalWrite(pin, HIGH); // lo activamos
    delayMicroseconds(10); // esperamos 10 microsegundos
    digitalWrite(pin, LOW); // lo desactivamos
    pinMode(pin, INPUT); // cambiamos el pin como entrada
    pulso = pulseIn(pin, HIGH,500000); // medimos el pulso de salida del sensor
    distancia = ((float(pulso/1000.0))*34.32)/2;
    distanciaTotal=distanciaTotal+distancia;
    j++;
  }

  if((distanciaTotal/10)<5){
    comprobarAccion(pin);
    if(enviarServidor(String(pin))){
      activado[pin]=true;
    }
  }
  distanciaTotal=0;
  j=0;
}
```



### Código 3. Método que comprueba el estado de los sensores de ultrasonidos

Si analizamos el código vemos que recibe como parámetro el pin donde está conectado el sensor. El método comienza realizando un bucle diez veces donde se calcula la distancia devuelta por el sensor de ultrasonidos y se almacena. Tras obtener diez medidas de distancia se calcula la distancia media, esto se realiza para obtener más precisión en la medida y evitar errores. Si la distancia media es menor que el umbral definido quiere significar que se ha detectado presencia y por lo tanto hay que activar la alarma. Para ello lo primero que se realiza es llamar al método `comprobarAccion(pin)` que se encarga de activar los actuadores relacionados con la activación de dicho sensor. Tras esto se envía el evento al sensor para que lo almacene y muestre el estado de la alarma en la aplicación web.

Esta acción se realizará para cada sensor que esté conectado al Arduino llamando a su método según el tipo de sensor.

### 5.2.3. Aplicación Web

Para el acceso al sistema se alarma se ha desarrollado una aplicación web que permite consultar y configurar la alarma. Esta aplicación será la interfaz entre el usuario y sistema de alarma. Desde aquí podrá realizar acciones como encender y apagar la alarma, parar la alarma en caso de que se haya activado, visualizar el registro de eventos y configurar los distintos parámetros de la alarma.

Para implementar la aplicación web se ha empleado `html5` y `css3`, para la maquetación y visualización, y `javascript` y `jquery` para dar funcionalidad a la página. Por detrás de esta aplicación se encuentra el servidor que se encarga de enviar toda la información que introduzcamos o seleccionemos en la aplicación al Arduino. La comunicación entre la aplicación y el servidor se ha realizado a través peticiones GET utilizando `jquery`.

La aplicación consta de 3 páginas seleccionables mediante el menú superior. Estas páginas son:

- Estado
- Registro
- Configuración

A continuación vamos a analizar cada una de las páginas que componen la aplicación.

#### Estado

Está en la página encargada de mostrar el estado de la alarma, es decir desde aquí podemos ver si la alarma está encendida o apagada y si alguno de los sensores está activado. La siguiente imagen muestra la interfaz:





**Ilustración 43. Interfaz de la aplicación. Estado de la alarma.**

En la parte superior se encuentran los botones para encender o apagar el sistema de alarma. También se encargan de mostrar el estado cambiando su color.

Después se sitúa una tabla que contiene la información de cada uno de los sensores que hay instalados actualmente en el sistema. Para cada sensor se muestra la entrada donde está conectado, el tipo de sensor, las observaciones que hayamos introducido y el estado. En el estado nos indica si está activado, es decir ha detectado una amenaza, o si esta desactivado.

Si alguno de los sensores estuviera activo el sistema de alarma se activaría mostrando un mensaje en la interfaz. La siguiente ilustración muestra el estado de la alarma activada.



**Ilustración 44. Interfaz de la aplicación. Alarma activada.**

En la parte inferior vemos que se muestra el mensaje de alarma activada. Esto quiere decir que algún sensor está activo por eso si nos fijamos en la tabla superior podemos ver que sensores se han activado, en este caso vemos que ha sido el sensor de la entrada 2.

Para poder parar la alarma y que el Arduino desactive los actuadores que haya encendido se ha situado un botón en la parte inferior que para toda la alarma. Tras parar la alarma desaparecerán todos los mensajes y se volverá al estado normal.

## Registro

La página de registro muestra el historial de eventos de alarma que se han registrado. Cada vez que se activa un sensor el Arduino envía el evento al servidor y este lo almacena. La siguiente ilustración muestra la interfaz del registro:



# ARDUENTORNO SEGURIDAD

ESTADO

REGISTRO

CONFIGURACIÓN

## Registro de eventos

Identificador	Sensor	Fecha y hora
1026	Puerta principal (Sensor PIR)	2014-05-22 21:19:45
1025	Garaje (Sensor de ultrasonidos)	2014-05-22 21:19:42
1024	Piscina (Sensor PIR)	2014-05-22 21:18:35
1023	Garaje (Sensor de ultrasonidos)	2014-05-22 21:17:54
1022	Jardin (Barrera de infrarrojos)	2014-05-22 21:17:50
1021	Puerta principal (Sensor PIR)	2014-05-22 21:17:45
1020	Piscina (Sensor PIR)	2014-05-22 20:58:30
1019	Puerta principal (Sensor PIR)	2014-05-22 20:56:27
1018	Piscina (Sensor PIR)	2014-05-22 20:56:08

**Ilustración 45. Interfaz de la aplicación registro de eventos**

Como vemos aparece una tabla donde cada fila es una activación de un sensor. La primera columna muestra un identificador numérico, la segunda muestra las observaciones del sensor y el tipo de sensor. La última columna muestra la fecha y hora en la que se produjo el evento.

## Configuración

La página de configuración ofrece una interfaz para configurar el sistema de alarma. Desde aquí se pueden realizar estas acciones:

- Cambiar IP Arduino
- Cambiar IP Servidor
- Configurar entradas (Sensores)
- Configurar salidas (Actuadores)
- Configurar acciones

La siguiente ilustración muestra la configuración de las IP's:



The screenshot shows a web application interface with a green header. On the left is a logo of a padlock inside a circle. To the right of the logo, the text "ARDUENTORNO SEGURIDAD" is displayed in bold black letters. Below the header, there are three navigation tabs: "ESTADO", "REGISTRO", and "CONFIGURACIÓN", with "CONFIGURACIÓN" being the active tab. The main content area is titled "Configuración" and contains two sections for IP configuration. The first section, "CONFIGURAR IP ARDUINO", has four input fields containing the values "169", "254", "165", and "132", followed by an "Enviar" button. The second section, "CONFIGURAR IP SERVIDOR", has four input fields containing the values "169", "254", "165", and "127", followed by an "Enviar" button.

**Ilustración 46.** Interfaz aplicación. Configuración de las IP's

Para configurar la IP del Arduino y del servidor se han dispuesto 4 campos de texto donde introducir cada campo que forma la IP. También se puede ver en cada campo el valor actual de la IP. Para enviar la IP al Arduino se realiza desde el botón Enviar asociado a cada IP.

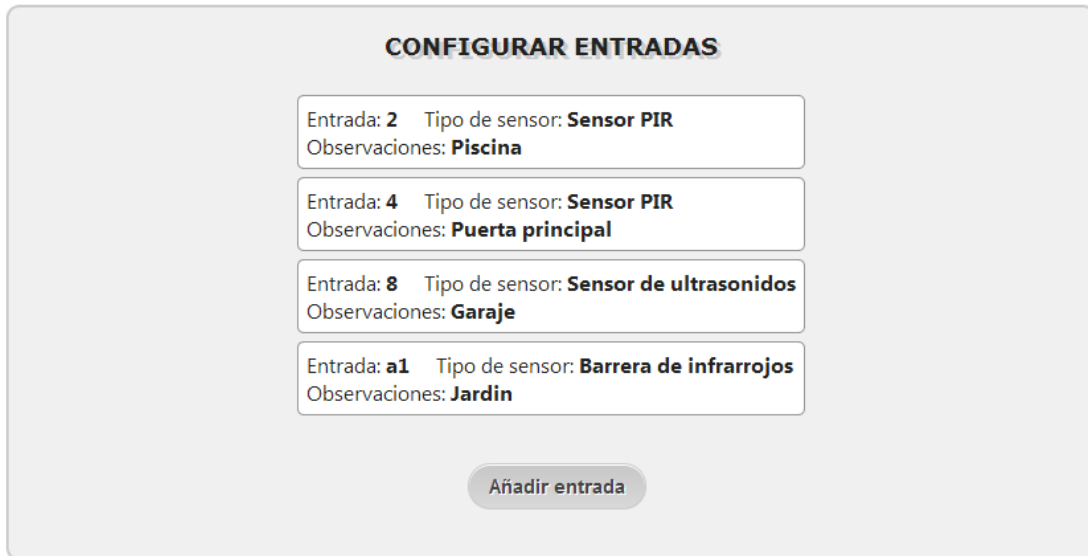
Para cargar las direcciones IP actuales se realiza una petición de tipo GET al servidor para cada una de las IP's. Después se extrae cada uno de los 8 bits que forman la IP y se visualiza por separado en cada textbox. El siguiente fragmento de código muestra el método encargado de realizar esta tarea.

```
function cargarIP(){
    $.get( "php/cargaIP.php", function( data ){
        data=data.concat('.');
        for(var i=1;i<5;i++){
            var pos=data.indexOf('.');
            var subs=data.substring(0,pos);
            $('#ip'.concat(i)).attr('placeholder',subs);
            data=data.substring(pos+1);
        } });

    $.get( "php/cargaIPser.php", function( data ){
        data=data.concat('.');
        for(var i=1;i<5;i++){
            var pos=data.indexOf('.');
            var subs=data.substring(0,pos);
            $('#ipser'.concat(i)).attr('placeholder',subs);
            data=data.substring(pos+1);
        } });
}
```

**Código 4.** Función para cargar las IP's

Para configurar los sensores que están conectados al sistema iremos a la sección *Configurar Entradas* dentro de la pestaña *Configuración*:



The screenshot shows a web interface titled "CONFIGURAR ENTRADAS". It contains four configuration boxes, each with an input field for the entry name, a dropdown for the sensor type, and a text field for observations. The configurations are as follows:

Entrada	Tipo de sensor	Observaciones
2	Sensor PIR	Piscina
4	Sensor PIR	Puerta principal
8	Sensor de ultrasonidos	Garaje
a1	Barrera de infrarrojos	Jardin

At the bottom of the interface is a button labeled "Añadir entrada".

**Ilustración 47. Interfaz aplicación. Configurar entradas**

Desde aquí podremos visualizar los sensores que hay configurados actualmente. También podremos añadir nuevos y modificar o borrar los existentes. Al pulsar sobre el botón *Añadir entrada* aparecerá debajo un recuadro donde configuraremos la entrada empleada, el tipo de sensor y unas observaciones. Al guardar se enviará al Arduino la nueva configuración de entradas.



**CONFIGURAR ENTRADAS**

Entrada: **2** Tipo de sensor: **Sensor PIR**  
Observaciones: **Piscina**

Entrada: **4** Tipo de sensor: **Sensor PIR**  
Observaciones: **Puerta principal**

Entrada: **8** Tipo de sensor: **Sensor de ultrasonidos**  
Observaciones: **Garaje**

Entrada: **a1** Tipo de sensor: **Barrera de infrarrojos**  
Observaciones: **Jardin**

Añadir entrada

**Añadir nueva entrada**

Entrada

Tipo de sensor

Observaciones

Guardar

**Ilustración 48. Interfaz aplicación. Configurar nueva salida.**

Para modificar o borrar una entrada existente deberemos pulsar sobre la entrada y nos aparecerá el siguiente recuadro donde realizaremos las acciones que deseemos.

**Modificar Entradas**

Entrada

Tipo de sensor

Observaciones

Modificar      Borrar

**Ilustración 49. Interfaz aplicación. Modificar salida.**

Si los que se quiere configurar son los actuadores deberemos ir a la sección *Configurar Salidas*. La forma de añadir, modificar o borrar actuadores será la misma que para los sensores.

### CONFIGURAR SALIDAS

Entrada: **6** Tipo de actuador: **Luz**  
 Observaciones: **Luz alarma**

Entrada: **a0** Tipo de actuador: **Sirena**  
 Observaciones: **Sirena**

Añadir salida

**Ilustración 50. Interfaz aplicación. Configurar salidas.**

La última sección de la pestaña Configuración será la encargada de configurar que salidas hay que activar cuando se activa una determinada entrada. La siguiente ilustración muestra la interfaz encargada de realizar esta acción.

### CONFIGURAR ACCIONES

Entradas	Salidas
Entrada: <b>2</b> Tipo: <b>Sensor PIR</b>	Salida: <b>6</b> Tipo: <b>Luz</b>
Entrada: <b>4</b> Tipo: <b>Sensor PIR</b>	Salida: <b>a0</b> Tipo: <b>Sirena</b>
Entrada: <b>8</b> Tipo: <b>Sensor de ultrasonidos</b>	
Entrada: <b>a1</b> Tipo: <b>Barrera de infrarrojos</b>	

**Ilustración 51. Interfaz aplicación. Configurar acciones.**



En la parte izquierda aparecen las entradas del sistema y en la derecha las salidas. Si se pulsa sobre una entrada aparecerán en la parte derecha las salidas relacionadas de color verde como se muestra en la anterior imagen. Para seleccionar o deseleccionar una salida solo habrá que pulsar sobre él y este irá cambiando de color indicando si está relacionado o no. Si se realiza una modificación aparecerá un mensaje indicando que debemos enviar la nueva configuración al Arduino. Pulsando el botón *Enviar* se enviarán las acciones al Arduino y este devolverá una respuesta al servidor indicando que ha recibido correctamente la nueva configuración.

### CONFIGURAR ACCIONES

Entradas	Salidas
Entrada: <b>2</b> Tipo: <b>Sensor PIR</b>	Salida: <b>6</b> Tipo: <b>Luz</b>
Entrada: <b>4</b> Tipo: <b>Sensor PIR</b>	Salida: <b>a0</b> Tipo: <b>Sirena</b>
Entrada: <b>8</b> Tipo: <b>Sensor de ultrasonidos</b>	
Entrada: <b>a1</b> Tipo: <b>Barrera de infrarrojos</b>	

Acciones desactualizadas en Arduino. Pulse el botón enviar para enviar las acciones a Arduino

**Ilustración 52. Interfaz aplicación. Configurar acciones desactualizadas.**

El siguiente fragmento muestra el código javascript encargado de manejar la selección de las salidas al pulsar sobre ellas y almacenar en la base de datos las relaciones con las entradas.

```
function guardaBorraAccion(id){  
  
    if(global!=""){  
        $("#mensajeAcciones").html("<p style='color:#ff0000 ;font-weight:  
bold'>Acciones desactualizadas en Arduino. Pulse el botón enviar para  
enviar las acciones a Arduino</p>");  
  
        $("#mensajeAcciones").append('<br><input type="button"  
id="botonenviaracciones" value="Enviar" class="botonenviar"  
onClick="enviarAccionesArduino()"><br><br>');  
  
        id=id.substr(1);  
        var a=new String($('.accionActuador#'+id).css('background-  
color'));  
  
        if(a.charAt(4)=='2'){  
            $('.accionActuador#'+id).css('background-  
color','rgb(170,235,120)');  
        }  
        else{  
            $('.accionActuador#'+id).css('background-  
color','white');  
        }  
    }  
}
```



```
    $.get( "php/guardarBorrarAccion.php",{entrada:global, salida:id});
  }else{alert("Debes seleccionar una entrada antes");}
}
```

#### Código 5. Función encargada de asociar las acciones.

El método comprueba primero si hay alguna entrada seleccionada, si no es así muestra un mensaje de aviso. Después muestra el mensaje que indica que hay que enviar las actualizaciones al Arduino y obtiene el color de fondo de la salida seleccionada. Esto nos servirá para saber si ya está seleccionado o no. Por último enviaremos al servidor la relación entre la entrada y la salida para que almacene la acción o la borre.

### 5.2.4. Servidor Web

Para la implementación del servidor se ha utilizado un servidor web Apache, en este caso se ha empleado el software libre XAMPP ya que facilita la instalación del servidor Apache. El lenguaje que se ha empleado en el servidor ha sido PHP.

La función del servidor en el proyecto ha sido la siguiente:

- Alojar la aplicación web de control de la alarma.
- Comunicar la aplicación web con el sistema Arduino.
- Acceso a la base de datos.
- Recibir eventos del sistema de Alarma enviados por el Arduino.

Implementa una serie de funciones PHP para realizar una serie de acciones, algunas de estas funciones se encargan de:

- Añadir, modificar y borrar entradas del sistema en la BBDD
- Añadir, modificar y borrar salidas del sistema en la BBDD
- Cargar IP de servidor y del Arduino.
- Generar las entradas, salidas y acciones del sistema que se le envían al Arduino.
- Obtener las entradas, salidas y acciones para visualizarlas en la aplicación web.
- Almacenar el estado de la alarma
- Recibir eventos y respuestas de confirmación del Arduino

El siguiente fragmento muestra el código PHP encargado de almacenar las acciones en la BBDD.

```
<?php
  if(isset($_GET['entrada']) && !empty($_GET['entrada']) &&
  isset($_GET['salida']) && !empty($_GET['salida'])) {

      $mysqli = new mysqli("localhost", "root", "", "alarma");

      $consulta="SELECT * FROM acciones where
  entrada='".$_GET['entrada']."' and salida='".$_GET['salida']."'";

      $resultado = $mysqli->query($consulta);

      if($resultado->num_rows>0){
```



```

        $mysqli->query("DELETE FROM acciones where
entrada='".$_GET['entrada']."' and salida='".$_GET['salida']."'");

    }
    else{
        $mysqli->query("INSERT INTO acciones (entrada,salida) VALUES
('".$_GET['entrada']."', '".$_GET['salida']."'");
    }
    $consulta="UPDATE configuracion SET orden='0' WHERE
id='accionesActualizadas'";
    $mysqli->query($consulta);

    $mysqli->close();
}

?>

```

**Código 6. Código encargado de almacenar las acciones en la BBDD**

Si analizamos el código vemos que la función recibe como parámetros la entrada y la salida que se quiere relacionar. Con estos parámetros se realiza una consulta para saber si existe ya esta relación. Si no existe se añade a la base de datos y si existe entonces se elimina.

Otro ejemplo de código PHP implementado muestra cómo se genera el mensaje que contiene las acciones que debe de realizar el Arduino cada vez que se activa un sensor.

```

<?php

    $mysqli = new mysqli("localhost", "root", "", "alarma");

    $consulta="SELECT * FROM acciones";

    $resultado = $mysqli->query($consulta) ;

    $parametros="acciones=";
    if($resultado->num_rows==0) $parametros=$parametros."00";
    else{
        if($resultado->num_rows*2>9){
            $parametros=$parametros.$resultado->num_rows*2;
        }
        else{
            $parametros=$parametros."0";
            $parametros=$parametros.$resultado->num_rows*2;
        }
        while($registro = $resultado->fetch_assoc()){
            $parametros=$parametros.parsearPIN($registro['entrada']);
            $parametros=$parametros.parsearPIN($registro['salida']);
        }
    }

    echo $parametros;

    function parsearPIN($pin){
        if($pin=='a0'){
            return 'a';
        }
    }

```

```
    }
    else if($pin=='a1'){
        return 'b';
    }
    else if($pin=='a2'){
        return 'c';
    }
    else if($pin=='a3'){
        return 'd';
    }
    else if($pin=='a4'){
        return 'e';
    }
    else if($pin=='a5'){
        return 'f';
    }
    return $pin;
}
?>
```

**Código 7. Método que se encarga de generar el mensaje con las acciones**

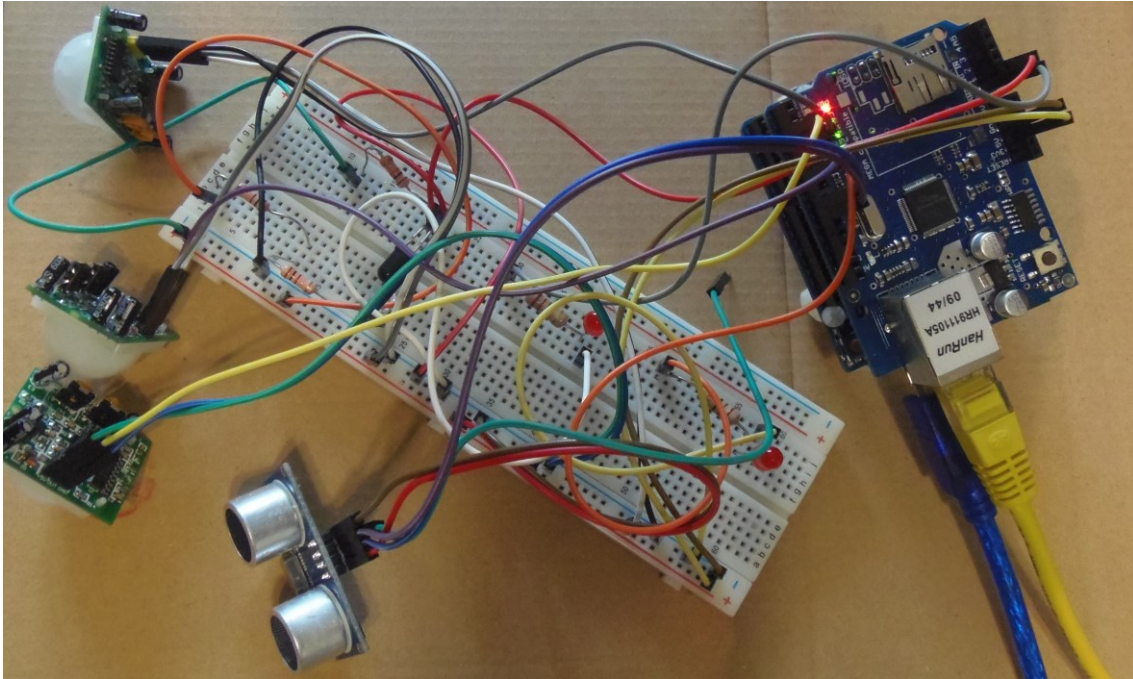
Primero se realiza una consulta para obtener todas las acciones. Cada entrada devuelta en la consulta contendrá un pin de entrada y uno de salida que indicará que están relacionados entre sí.

Con estas acciones se empezará a construir el mensaje donde los dos primeros dígitos indicaran el número de pines relaciones que contiene el mensaje. Después de estos dos dígitos se comienza a concatenar parejas de pines, que representaran las acciones, donde el primer pin es la entrada y el segundo la salida relacionada. Para los pines analógicos, que tienen una letra 'a' al inicio, se realizará una conversión de nombre para que solo tenga un dígito.

### 5.3.Prototipo

La siguiente ilustración muestra cómo ha sido el prototipo empleado durante el desarrollo del proyecto.





**Ilustración 53. Prototipo empleado en el desarrollo y pruebas**

Como se aprecia en la imagen tenemos el Arduino con el shield Ethernet incrustado sobre él. El Arduino recibe la alimentación a través del USB que también se ha empleado para enviar mensajes por el puerto serie al ordenador y así poder depurar y ver el estado del Arduino mediante un visor del puerto serie. La comunicación Ethernet se ha realizado con un cable conectado directamente al ordenador que contiene el servidor web.

Para la realización de los circuitos y facilitar las conexiones se ha empleado una protoboard. Desde esta protoboard se suministra alimentación a todos los componentes del sistema. Sobre ella se han realizado los circuitos que simulan el encendido de una luz mediante el uso de resistencias y led's. También se ha realizado el circuito que simula una barrera de infrarrojos con led's emisores y receptores de infrarrojos y resistencias.

Se han conectado varios sensores PIR y de ultrasonidos a las diferentes entradas para realizar las pruebas. La alimentación de estos sensores también ha sido a través de la protoboard.

Los esquemas de conexión de los distintos componentes se pueden encontrar en la sección 4.3 de la presente memoria.

## 5.4. Conclusiones

En este capítulo se ha pretendido mostrar cómo se ha implementado el sistema que en los capítulos anteriores se ha realizado la especificación y el diseño. La idea no era mostrar todo el código escrito sino explicar que tecnologías se han empleado y como ha sido la implementación de los puntos más importantes del proyecto.

Se ha hecho un repaso de los componente principales, como han sido la base de datos, la programación del Arduino, la aplicación web y el servidor web, intentando explicar el funcionamiento de cada uno de ellos.

Los fragmentos de código han buscado mostrar algunas implementaciones de ciertos componentes del sistema para así comprender mejor las explicaciones y poder ver como se ha implementado una determinada funcionalidad.

El prototipo muestra físicamente como ha sido el modelo empleado para el desarrollo y para las pruebas realizadas durante todo el desarrollo del proyecto.



## 6. Conclusiones

### 6.1. Trabajo realizado

Después de varios meses se ha concluido la realización del sistema de seguridad programable objetivo de este proyecto. En el hemos conseguido realizar el sistema de seguridad mediante el empleo de diferentes tecnologías de software libre, una de las más importantes ha sido el uso del hardware Arduino. Este hardware ha sido fundamental en la consecución del proyecto ya que ha sido el cerebro que nos ha permitido poder interconectar los distintos sensores y darle funcionalidad al sistema pudiendo conectarlo a un servidor.

La primera fase del proyecto ha constado de una búsqueda de información relacionada con la tarea que se pretendía realizar para poder situar el proyecto y marcar objetivos. Esto ha llevado a realizar una especificación de requisitos donde de modo formal se ha descrito el comportamiento del sistema que se iba a desarrollar. Posteriormente guiados por esta especificación se ha realizado el diseño de todo el sistema.

La parte más extensa y complicada del proyecto ha sido la implementación de todas las características con las que debía contar el sistema. La fase de implementación también ha contado con una extensa fase de pruebas para constatar que el funcionamiento del sistema era el correcto.

Otra parte interesante del proyecto ha sido el desarrollo de la aplicación web y del servidor encargado de proporcionar una interfaz al usuario desde donde manejar el sistema, ofrecer integridad en los datos gracias a la base de datos y ofrecer una forma de comunicación con el Arduino a través de Ethernet. Esto ha hecho que aumentara el número de tecnologías empleadas en la realización del proyecto.

Una de las mayores dificultades ha sido encontrar sistemas similares donde se emplee Arduino ya que solo se ha podido encontrar dos sistemas que lo utilizaran. Por eso se han analizado también sistemas que no emplearan Arduino. Otro aspecto a la hora que analizar sistemas similares ha sido la poca información que se puede encontrar debido obviamente a que es un campo donde es muy importante la propia seguridad y donde la hermeticidad de los sistemas es imprescindible.

Desde el punto de vista de la implementación uno de los problemas encontrados ha sido realizar la sincronización del servidor con el Arduino para saber si este último ha recibido y procesado correctamente las solicitudes. Para resolverlo se han implementado mensajes de confirmación que envía el Arduino al servidor cada vez que este envía un mensaje para así sincronizar correctamente la información del servidor con la del Arduino.

### 6.2. Aportaciones

En primer lugar se ha realizado un estudio de varios sistemas similares que existen en la actualidad. Este estudio ha permitido analizar qué aspectos importantes incorporan

cada uno de los sistemas y así poder decidir cuáles son las características con las que debe contar el sistema que se va a desarrollar.

Después se ha realizado una especificación completa del sistema de seguridad basado en Arduino. La estructura, los conceptos y el análisis realizado en esta especificación pueden ser trasladables a otros proyectos.

Luego se ha realizado el diseño del sistema de seguridad donde haciendo uso de diferentes esquemas y diagramas UML se muestra cómo va a ser el sistema y que es lo que va a hacer. Una primera descripción conceptual permite tener una visualización de del sistema, después mediante esquemas se muestran los elementos hardware que se van a emplear y su conexión. Para la especificación software se emplean diagramas de secuencia UML que permiten modelar la interacción entre los objetos del sistema.

Por último se ha aportado la implementación de un sistema de seguridad con Arduino el cual se comunica a través del shield Ethernet con un servidor que contiene una aplicación web para la gestión y el manejo del sistema. En esta parte se muestra tanto la programación del Arduino como de la aplicación web y el servidor que además dispone de una base de datos MySQL. Los fragmentos de código pretenden mostrar la implementación de algunos de los módulos funcionales del sistema.

## 6.3. Ampliaciones

El sistema desarrollado contiene varios puntos donde se podrían realizar ampliaciones que permitirían mejorar y aumentar la potencia del sistema. A continuación vamos a ver algunas posibles ampliaciones:

- Comunicación mediante tecnología GSM. El uso de tecnología GSM para el acceso a internet permitiría poder comunicar el Arduino con el servidor sin la necesidad de cables desde cualquier punto que dispusiera de cobertura. Para realizar esta ampliación se podría hacer uso de la shield GSM específica para Arduino.
- Envío de mensajes SMS cuando se produjera un evento en la alarma.
- Nuevos sensores y actuadores. Añadir nuevos sensores como detectores de humo o de gas permitirían detectar otro tipo de amenazas. También podrían emplearse sensores de inundación o de sonido.
- Aplicación de móvil para controlar el sistema de alarma. El acceso al sistema de alarma mediante una aplicación específica mejoraría la facilidad de manejo.
- Conexión de varios Arduinos entre ellos para aumentar la red y poder añadir un mayor número de sensores y actuadores y que todos los Arduinos se comunicaran entre sí.
- Acceso al sistema de alarma desde el exterior de la red local, para ello habría que implementar protocolos de seguridad.



# Referencias

---

- [1] Página principal del proyecto Arduino. [www.arduino.cc](http://www.arduino.cc)
- [2] Wikipedia, la enciclopedia libre [es.wikipedia.org](http://es.wikipedia.org)
- [3] Functiodomo [www.functiodomo.com](http://www.functiodomo.com)
- [4] Esquemas con Fritzing. <http://fritzing.org/home/>
- [5] MARD. Sistema de Alarma con Arduino MEGA  
<http://blog.bricogeek.com/noticias/arduino/mard-sistema-de-alarma-con-arduino-mega/>
- [6] Violet UML Editor <http://violet.sourceforge.net/>
- [7] Página web de creación de diagramas de secuencia  
<https://www.websequencediagrams.com>
- [8] W3Schools. Página web de información para desarrolladores web.  
<http://www.w3schools.com/>
- [9] Página jQuery. Librería javascript. <http://jquery.com/>
- [10] Teoría de la seguridad <http://definicion.de/seguridad/>
- [11] Información sobre sensores PIR <http://www.opiron.com/portfolio/todo-sobre-los-sensores-pir>
- [12] Información sobre varios tipos de sensores <http://libro-sensores.blogspot.com.es/>