

Sistema IoT de control de acceso inteligente basado en Arduino y servicios web

IoT system for intelligent access control based on Arduino and web services

Ing. Pedro Díaz Cruz^{1*}, Dr.Sc. Dioen Biosca Rojas², Ing. Guillermo González Jiménez³

Recibido: 11/2017 | Aceptado: 02/2018

PALABRAS CLAVE

Arduino
Control de acceso
RFID
Aplicación web
IoT

RESUMEN

La presente investigación propone un sistema de control de acceso inteligente con funcionalidades similares a los profesionales existentes en el mercado y una aplicación de gestión web multiplataforma. La solución es adaptable a cualquier tipo de entorno sobre el cual pueda desplegarse. Las necesidades de seguridad, control y monitoreo del acceso a una instalación siempre han estado presente y el sistema de llaves tradicional ha quedado obsoleto. Por este motivo se propone el desarrollo de un sistema propio alternativo que solucione esta problemática. Que se basa en la ideología del Internet of Things (IoT) utilizando hardware libre de Arduino, que dota de inteligencia a todas las puertas de una instalación y monitoriza parámetros de cualquier habitación de la misma.

ABSTRACT

The present work proposes an intelligent access control system with similar functionalities to the professionals existing in the market and a multiplatform web management application. The solution adaptable to any type of environment on which it may be deployed. The needs for security, access control & monitoring to a facility have always been present, and the traditional key system has become obsolete. In the world there are many suppliers but their solutions are expensive, limited by commercial restrictions, and often the full set of functionalities is unknown. Therefore, it is proposed to develop an alternative system of our own that solves this problem. This system is based on the ideology of IoT using Arduino free hardware, which provides intelligence to all the doors of an installation and monitors parameters of any room. The work ranges from the design process to its implementation and scale-up on a model built for this purpose. It achieves a functional and fast product, with a user-friendly interface and various levels of authentication.

KEYWORDS

Arduin
Access control
RFID
Web application
IoT

^{1*} Empresa Mixta Copal S.A. La Habana, Cuba. pedro.diaz@copalnet.co.cu

² Universidad Tecnológica de La Habana CUJAE, Cuba. biosca@copalnet.co.cu

³ Universidad de Ciencias Informáticas. La Habana, Cuba. gjimenez@uci.cu

Introducción

Desde la antigüedad, el ser humano ha necesitado protegerse a sí mismo y su confidencialidad. A mediados de los ochenta (Norman, 2011) comienzan a reemplazar las llaves mecánicas y comienzan a aparecer tipos de autenticación basados en componentes electrónicos y microprocesadores. Desde entonces, muchas compañías se han establecido y explotan un mercado bastante abundante, ya que el control de acceso es necesario en todo el mundo. Estas compañías ofrecen variedad de equipamiento y funcionalidades, pero las grandes compañías, muchas veces son conservadoras en cambios tecnológicos y siguen usando lo que ya tienen probado como seguro, hecho que limita la adaptación a nuevas tecnologías y arquitecturas y no abarata los precios que siguen siendo elevados.

El kit Arduino fue creado con propósitos educativos, pensado para artistas, diseñadores y otros que deseen incorporar computación a elementos físicos sin convertirse en un ingeniero electrónico (Banzi y Michael Shiloh, 2015). Arduino más que una marca, es una plataforma física de código y hardware abiertos para crear objetos interactivos que funcionan independientemente o se comunican con un software en cualquier computador e incluso directamente en la web y a bases de datos. Debido a la escala de producción y los avances tecnológicos actuales, el costo de una placa Arduino es muy bajo, lo cual lo hace atractivo a todos los públicos. Una de las grandes bondades de esta plataforma Arduino es la disponibilidad de pines de entrada y salida, tanto analógica como digital, que permiten la interacción con prácticamente todos los sensores disponibles en el mercado sin importar fabricante, modelo y protocolo. El desarrollo de plataformas de esta índole ha abaratado en gran medida el costo de los sensores (Kusriyanto y Putra, 2016). La comunidad ofrece librerías para la manipulación de sensores más complejos, por ejemplo, aquellos que necesitan protocolos como I2C, SPI, etc.

Otros proyectos han estudiado la factibilidad de soluciones de control de acceso basadas en microcontroladores Arduino (Saravia, 2015), pero no desarrollan un sistema completo integrador y proponen una solución muy simple y no muy desarrollada para su implementación.

El presente estudio se propone diseñar y probar sobre una maqueta un sistema de control de acceso inteligente que dotaría de inteligencia a cualquier punto de acceso a una instalación, ya sea una vivienda, parqueo o una gran edificación, basado en Arduino y gestionado por una plataforma web; que regule ciertas características propias de un sistema de ese tipo como es la gestión de alarmas o el tiempo permitido para la apertura de una puerta, entre otros. El sistema incluye la bonificación de leer parámetros de su entorno a través de sensores comerciales, por lo que es capaz de detectar presencias y la ocurrencia de incendios.

Materiales y métodos

Se emplearon los métodos científicos de observación y experimentación que realizan una intervención iterativa hasta lograr los objetivos propuestos. Se realizó un estudio sobre la teoría del control de acceso y el estado del arte en la materia. Luego se pasó a la primera etapa de diseño que se fue complementando con sucesivas iteraciones retroalimentada por las pruebas en una maqueta construidas a ese fin. Por último, se depura la integración del software y la plataforma de gestión WEB con el hardware disponible que está compuesto por las placas de Arduino y los sensores.

Se utilizaron varios materiales en la construcción del prototipo:

En el apartado de hardware (Figura 1):

Kit Arduino Mega 2560 con Ethernet Shield V1.

Integrado SM130 de MIFARE, como módulo lector RFID con sus respectivas tarjetas.

Placa de evaluación para el chip SM130 de *sparkfun* (fabricante) con antena incorporada.

Se utilizó una variedad de sensores: proximidad, pulsación, magnético.

Para simular un actuador a la salida de un relay se utiliza un LED, y varios con un código de colores para señalizar el estado de las puertas.

Juegos de cables, placa de prototipado, botones e interruptores electrónicos.

Se implementó un teclado táctil que se comunica en serie para complementar la obtención de la identidad de los usuarios.

Una computadora personal en la cual se ejecutaba el servidor del sistema.

Todos los componentes se montaron en una maqueta de madera a escala (1m por 1.30m) de una vivienda genérica.

En el apartado de Software:

Se diseñó una base de datos SQLite.

Los scripts que contienen la integración entre los controladores, la base de datos y el entorno gráfico se escriben en PHP.

El entorno visual se estructura sobre Symfony2, un marco de trabajo para PHP que además maneja la interacción de los scripts con las bases de datos.

El entorno visual se apoya en plantillas de *bootstrap* para facilitar su diseño y para garantizar que sea *responsive*.

Para la modelación de los sistemas y de la base de datos se utiliza Visual Paradigm y Microsoft Visio.



Figura 1. Leds indicadores de estado (izquierda), maqueta funcional con todo conectado (derecha), plataforma de control (abajo derecha).

Resultados y discusión

El paso preliminar constituye realizar un análisis de las diferentes variables que componen los sistemas de control de acceso y hacer un estudio para seleccionar la mejor opción. Luego, para la construcción del sistema propuesto se consideraron varios aspectos. La

parte electrónica, que involucra a las placas Arduino y sus Shields, microprocesadores asociados, sensores, actuadores y la señalización del estado del sistema y de las puertas. La segunda parte comprende el diseño del software de gestión del sistema, en base a criterios de diseño profesionales con múltiples funciones equivalentes a los sistemas comerciales actuales. La tercera fase se dedica al diseño de un escenario de prueba sobre una maqueta para la depuración del sistema y los protocolos de comunicación entre los dispositivos. Luego en un cuarto momento se hace un análisis económico aproximado en base a variantes del prototipo manufacturado.

Consideraciones a los mecanismos de autenticación

Los sistemas de autenticación actuales (Clarke, 2011) se agrupan basados en algo que se posee, algo que se conoce o en una característica biométrica. Lo ideal, descrito por todos los autores es combinar las mejores fuentes de autenticación posibles para aumentar la robustez del sistema. El sistema propuesto se basa fundamentalmente en la autenticación mediante tarjetas RFID –*Radio Frequency IDentification*–, que se considera el más eficiente entre las tecnologías de autenticación hasta el momento (Tabla 1), aunque también se emplean teclados digitales y es posible integrar módulos biométricos en el futuro.

Consideraciones al diseño del hardware

El hardware del sistema está compuesto por las placas microcontroladoras Arduino, tanto el software y hardware de esta plataforma son totalmente abiertos y cuentan con una inmensa comunidad de aprendizaje que socializa sus conocimientos y actualizan a los usuarios en el estado del arte constantemente. La amplia red de contribuyentes explica desde tutoriales básicos de cómo comenzar con la plataforma hasta el posteo de librerías y ayudas especializadas para hardware específico.

A pesar de que el diseño original de la plataforma no es el soporte a sistemas profesionales y si bien sus especificaciones no son gama alta en capacidad de procesamiento, dada su versatilidad, alta durabilidad, protección contra errores de conexión y cortos circuitos y su bajo precio, se ha convertido en plataforma ideal para el desarrollo de prototipos y su uso en sistemas fiables y de altas prestaciones está siendo altamente investigado

por la comunidad. Como las placas Arduino comerciales son destinadas a ambientes y situaciones muy variadas, los fabricantes no brindan un estudio detallado del parámetro MTBF. Sin embargo, algunos profesionales y científicos han realizado análisis de este tipo sobre placas Arduino (Novacek, 2016), dándole en el peor de los casos un MTBF de aproximadamente 8 años.

Existe una amplia gama de complementos electrónicos y placas compatibles con el controlador Arduino (denominadas Shields por el fabricante), que los proyectos pueden ser particularizados a la funcionalidad deseada y expandirse modularmente a otras si es necesario.

¿Por qué elegir Arduino ante otras placas microprocesadoras igual de populares y mucho más potentes como Raspberry?

El hecho que un corte de energía en el sistema no dañe el funcionamiento del mismo y requiera reconfiguración como es el caso en PI.

Un sistema de control de acceso tiene que poseer respaldo energético para garantizar la total funcionalidad de las puertas y la placa Arduino es mucho más fácil de alimentar.

A pesar de no ser una diferencia significativa, el precio es algo que siempre hay que considerar.

El hecho de que Raspberry PI tenga un OS es su mayor fortaleza, pero en este contexto es su mayor debilidad. Si posee un OS y además está conectado a la red significa que es muy vulnerable a atacantes y programas maliciosos mientras que en Arduino es imposible la reconfiguración a través de la red.

Parámetros del sistema	Código de barras	Reconocimientos de caracteres Ópticos	Reconocimiento de Voz	Biométricas	Tarjetas (con contactos)	Sistemas RFID
Cantidad típica de datos	1-100	1-100	-	-	16-64k	16-64k
Densidad de datos	Baja	Baja	Alta	Alta	Muy Alta	Muy Alta
Facilidad de lectura por las máquinas	Buena	Buena	Costosa	Costosa	Buena	Buena
Facilidad de lectura por las personas	Limitada	Simple	Simple	Difícil	Imposible	Imposible
Influencia del polvo/humedad	Muy Alta	Muy Alta	-	-	Posible	No influencia
Influencia de cobertura óptica	Fallo total	Fallo total	-	Posible	-	No influencia
Influencia de la dirección y de la posición	Baja	Baja	-	-	Unidireccional	No influencia
Degradación por uso	Limitada	Limitada	-	-	Contactos	No influencia
Relación costo-eletrónica de lectura	Muy baja	Medio	Muy alta	Muy alta	Baja	Media
Probabilidad de réplica o modificación no autorizada	Ligera	Ligera	Posible (grabación)	Posible	Imposible	Imposible
Velocidad del proceso de lectura	Baja ~4s	Baja ~3s	Muy Baja >5s	Muy baja >5-10s	Baja ~4s	Muy Alta ~0.5s
Distancia máxima entre la fuente de datos y el identificador	0-50cm	<1cm Scanner	0-50cm	Contacto Directo(solo para huellas dactilares)	Contacto directo	0-5m,

Tabla 1. Ventajas de los sistemas de acceso RFID sobre otros sistemas.

Como módulo lector se utiliza el integrado SM130 de MIFARE con sus tarjetas correspondientes, acoplado a una placa de evaluación de *sparkfun*, que tiene una antena integrada (Figura 2).



Figura 2. Chip SM130 sobre una placa de evaluación como lector RFID.

Consideraciones al desarrollo de la plataforma de gestión web

Para gestionar el sistema se escoge un ambiente web, producto a que la mayoría de dispositivos de uso personal y corporativo cuentan con un explorador que permitiría el acceso al mismo. Además, se construye *responsive*, término del inglés que implica que el sitio se ajusta a todas las resoluciones y dimensiones de pantalla. Se escriben los scripts en lenguaje Hypertext Preprocessor (PHP) y se construye la aplicación web utilizando Symfony2.

PHP es uno de los lenguajes del lado del servidor más usados en el mundo, muy adecuado para desarrollo web y puede ser embebido en HyperText Markup Language (HTML). El principal objetivo del lenguaje es desarrollar rápidamente páginas webs dinámicas. Como es tan popular funciona en todos los sistemas operativos de cierta relevancia, es compatible con la mayoría de los servidores web, incluyendo Apache e IIS y muchos más. El lenguaje permite escoger entre programación orientada a objetos y procedural o una mezcla de ambos. Es compatible con casi todas las bases de datos conocidas y está específicamente diseñado para ser más seguro que otros lenguajes Common Gateway Interface (CGI) como C y Perl.

Teniendo en cuenta lo descrito anteriormente, se decide utilizar el lenguaje PHP para la realización de la plataforma, ya que conviene que sea del lado del servidor por los objetivos que se persiguen de versatilidad y seguridad. Ahora la cuestión reside en decidir

si es eficiente construir una aplicación desde cero o utilizar herramientas y ambientes de desarrollo existentes.

En la actualidad, no se crea una página desde cero, existen muchas herramientas disponibles que facilitan e incluso hacen mucho más robusta y segura la página web. Este es el caso de los *Frameworks* que no son más que un conjunto de herramientas previamente diseñadas y un mecanismo de interacción que permite enfocarse más en el desarrollo del contenido y no en las herramientas para que su contenido sea seguro, adaptable y con todas las características actuales. Para PHP, existe una gran variedad de ambientes de este tipo como es el caso de Yii, CodeIgniter, Symfony, Laravel (Esaú, 2015), de hecho, cualquiera puede modificarlas y crearse sus propias herramientas. Dada una búsqueda exhaustiva a través de los foros y páginas especializadas de desarrollo web, el *Framework* de PHP más reconocido y para el que la comunidad tiene más soporte es sin duda la herramienta Symfony.

Consideraciones a la arquitectura del sistema de control de acceso

El mercado ha sentado las pautas y a través de los años ha predominado una arquitectura basada en un bus RS485 (Figura 3), que enlaza a todos los controladores a un controlador maestro y este se comunica con una base de datos en un servidor mediante RS232. La distancia del cableado RS485 máxima es de 1,22 km con una distancia máxima entre controladores de 450m.

Como se muestra en la figura 3, se está migrando hacia una arquitectura más distribuida basada en las redes Internet Protocol (IP), siguiendo la filosofía “todo sobre IP”, que implica un solo controlador por puerta conectada a la red, normalmente justo encima de la puerta.

Diseño y dimensionamiento del hardware

El hardware a diseñar tiene que proveer una forma de comunicación entre los sensores y actuadores de las puertas y los servidores de toma de decisión que se encuentran en algún lugar remoto. Para la comunicación entre los módulos de las puertas y los controladores se elige una comunicación serie, que incluye al lector RFID y a los sensores de estado. Esta conexión para alcanzar mayores distancias podría ser convertida a otro protocolo más resistente a las distancias como RS485. El intercambio de mensajes entre el servidor de toma de decisión se realiza mediante un Shield de Arduino (Ethernet Shield) sobre Ethernet y con un protocolo propio, Figura 4.

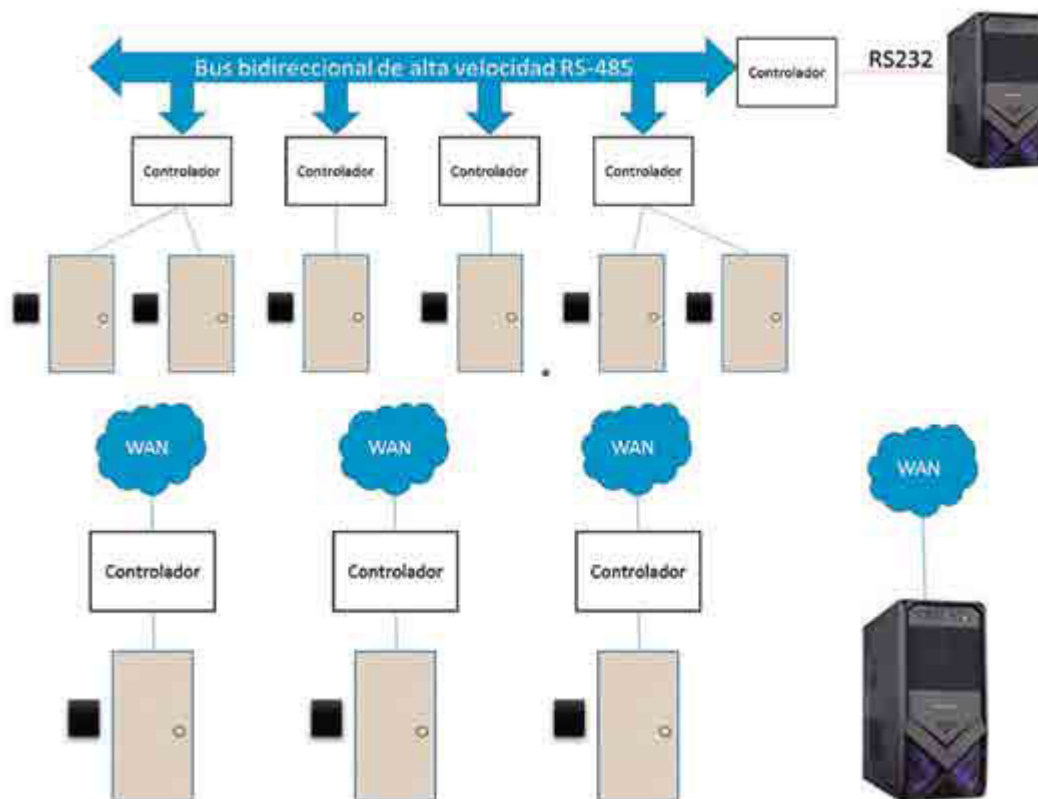


Figura 3. Arquitectura tradicional sobre bus RS-485 (Arriba) versus Arquitectura distribuida sobre IP (Abajo).

Se tienen en consideración posibles variantes de alimentación (Figura 5) que pueden ser implementadas, mediante una fuente local o Power over Ethernet (PoE) según convenga, en un análisis posterior se tienen en cuenta las variaciones posibles para ambos casos.

Diseño y dimensionamiento del software

El software comprende un conjunto de scripts que se comunican con una base de datos que contiene toda la información de usuario, instalación y permisos co-

respondientes y de un conjunto de páginas web que permiten la gestión gráfica del sistema.

Para el desarrollo del sistema de gestión se tienen en cuenta los siguientes requisitos funcionales:

Monitorear en tiempo real y llevar un registro de sucesos de todo tipo lo más detallado posible:

Diferenciar por un código de colores los tipos de eventos ocurridos que pueden ser:

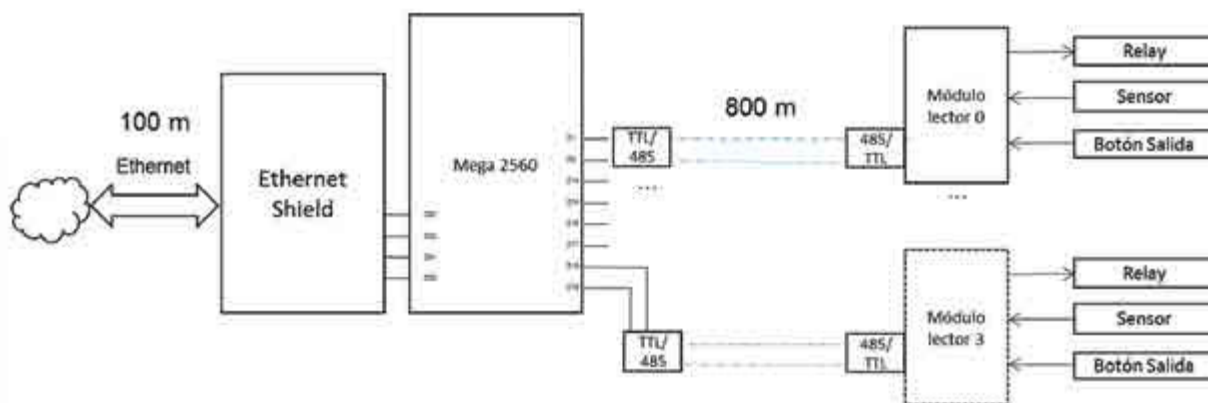


Figura 4. Esquemático del controlador.

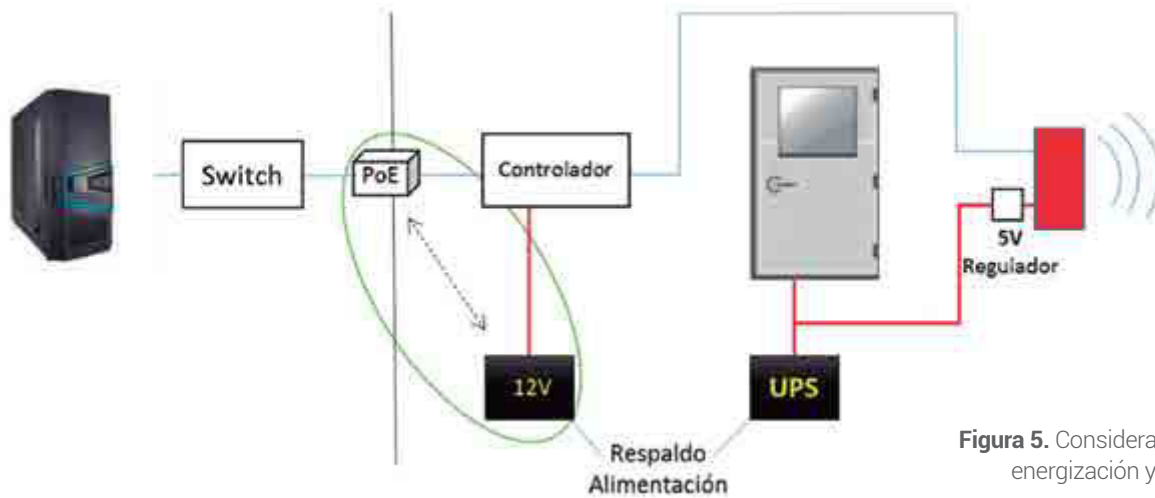


Figura 5. Consideraciones de energización y respaldo.

- I. Acciones de puertas
- II. Alertas
- III. Conexión
- IV. Configuración

Darle un tratamiento especial a cada una de las alertas que son:

Controlador desconectado.

- II. Puerta Forzada
- III. Puerta Abierta
- IV. Alarma de Incendio/Intruso
- V. Otros posibles
- C. Poder filtrar el registro de sucesos por el parámetro que se desee
- D. Eliminar las alertas una vez solucionadas
- E. Activar la alarma de puerta abierta cuando esté abierta más tiempo del previamente especificado a los controladores
- F. Las alarmas tienen que brindar una notificación en tiempo real

Controlar Manualmente el sistema:

- A. Abrir y cerrar puertas
- B. Identificar cualquier testigo y comprobar sus permisos
- C. Definir acciones a tomar en caso de eventos extraordinarios como un incendio
- D. Definir parámetros de configuración del sistema y de los controladores

Definir los elementos del sistema y hacerlo adaptable a cualquier instalación:

- A. Soportar la adición modificación, visualización de los siguientes elementos del sistema:

- I. Puertas
- II. Habitaciones
- III. Testigos

- B. Las habitaciones pueden bloquearse por decisión de los administradores del sistema
- C. Los testigos deben añadirse a través del lector del puesto de trabajo y no manualmente
- D. Permitir la visualización, y modificación sólo de la descripción y el tipo, a las listas (ya que el controlador es el que envía esos datos que no se deberían modificar):

- I. Lectores
- II. Sensores

Configurar los Usuarios del Sistema con todos los datos correspondientes y sus informaciones de acceso:

- A. Definir operadores/administradores de la plataforma de gestión, campos específicos:
 - I. Usuario y contraseña
 - II. Rol
- B. Definir empleados de la institución con los siguientes campos específicos:
 - I. Fecha del contrato
 - II. Cargo
- C. Definir Clientes/Visitantes con los siguientes campos específicos:
 - I. Fecha de entrada
 - II. Fecha de salida

A la hora de añadir un usuario nuevo o modificar su identificador debe ser suficiente con pasar el identificador por el lector de operación, pero también se puede hacer manual de una lista de testigos disponibles.

Un apartado que permita la especificación de los planes de acceso:

- A. Permitir la creación, modificación, visualización y eliminación de los planes de acceso
- B. Los planes de acceso deben definir los horarios individuales para cada día de la semana a las habitaciones deseadas
- C. Soportar la visualización de un plano de planta de la instalación para un mejor trabajo de los administradores del sistema
- D. Definir si el perfil de acceso puede ser visible por un usuario de menor permiso (operador)

Es necesaria la integración de la plataforma con lectores que no serán utilizados en las puertas, sino como lectores de operación que al acercar la tarjeta permita:

- A. Añadir testigos a la base de datos y ponerlos disponibles
- B. Agregando un usuario nuevo le asigne identificador automáticamente
- C. En la sección de Identificar TAG que solo con acercar el identificador a un lector de servicio inmediatamente devuelva la identidad del usuario asociado al mismo

D. Para acceder al sistema de gestión es necesaria una estricta autenticación

1. La plataforma debe poseer dos roles de actores:
 - A. Administrador
 - I. Todos los permisos
 - B. Operador
 - I. Monitorizar todos los sucesos
 - II. Asignar los perfiles de acceso definidos por el administrador
2. Permitir la salida del usuario actual y que regrese al nivel de autenticación.

La pantalla inicial de la aplicación se muestra en la figura 6, que lista todos los sucesos en la plataforma en orden cronológico, lo que permite el filtrado por diferentes parámetros acorde a los requisitos funcionales. En la figura 7 se muestra otro apartado de la aplicación donde se pueden definir perfiles de acceso.

Diseño de la maqueta.

Como base del control en el otro lado del servidor se utilizaron los siguientes componentes (Figura 8) sobre una maqueta correspondiente a un plano de planta de una instalación genérica:

- Arduino Mega2560 como controlador principal.

Monitoreo 6

Control

Plataforma

Usuarios

Acceso

Log

Alertas 1

Fecha	Tipo	Descripción
2017-05-09 22:24:40	Acciones de puerta	Se ha desbloqueado la puerta P_H2
2017-05-09 22:24:35	Acciones de puerta	Se ha cerrado la puerta P_H2
2017-05-09 22:24:20	Acciones de puerta	Se ha desbloqueado la puerta P_H2
2017-05-09 22:05:50	Conexion	06CC/03D5/12345678 VIRE_1/VIRE_2/VIRE_3/VIRE_4/VIRE_5/VIRE_6/VIRE_7/VIRE_8/VIRE_9 Fin Conexion al Controlador
2017-05-09 17:34:33	Configuracion	Usen 3A/Adress:4 en emag40ed id 8
2017-05-09 17:45:59	Configuracion	Cambio de configuracion
2017-05-09 17:26:01	Acciones de puerta	01/9P/11115910/50/250/Ping correcto
2017-05-09 17:04:57	Alerta	SGR Controlador desconectado
2017-05-09 17:34:37	Conexion	06CC/03D5/12345678 VIRE_1/VIRE_2/VIRE_3/VIRE_4/VIRE_5/VIRE_6/VIRE_7/VIRE_8/VIRE_9 Fin Conexion al Controlador
2017-05-09 17:26:57	Conexion	06CC/03D5/12345678 VIRE_1/VIRE_2/VIRE_3/VIRE_4/VIRE_5/VIRE_6/VIRE_7/VIRE_8/VIRE_9 Fin Conexion al Controlador
2017-05-09 17:23:48	Acciones de puerta	01/9P/11115910/50/250/Ping correcto

Figura 6. Pestaña de monitoreo de la Plataforma.

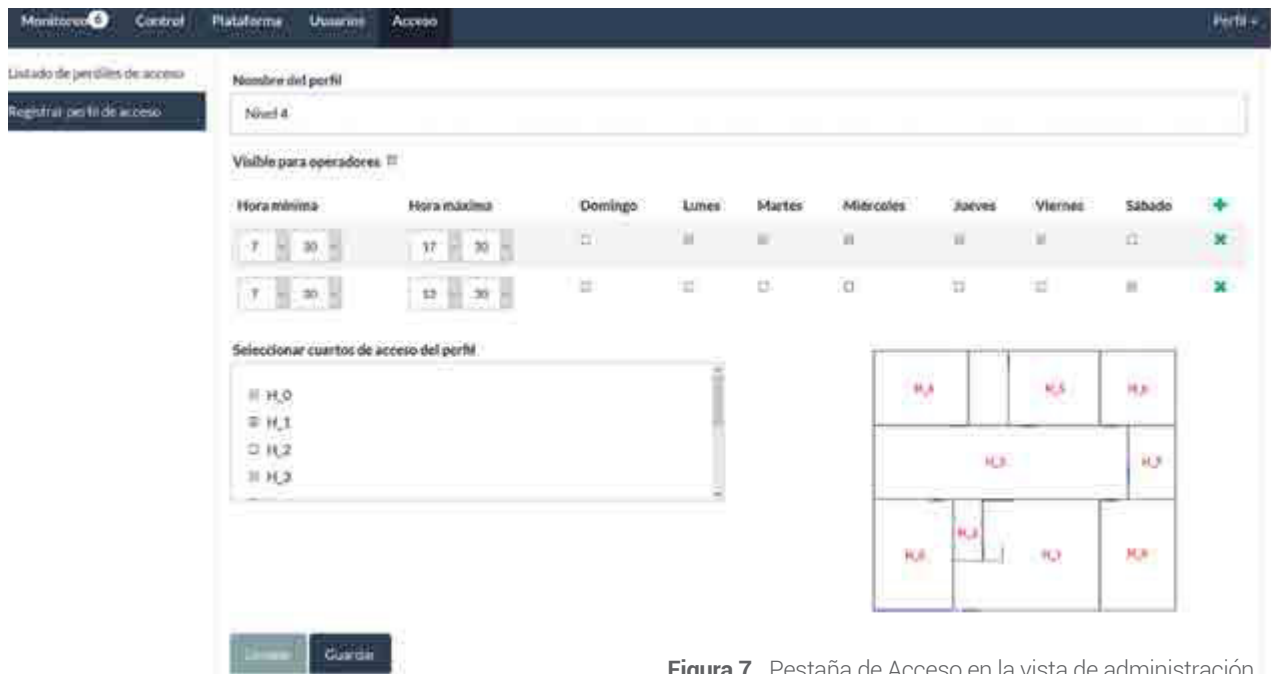


Figura 7. Pestaña de Acceso en la vista de administración.

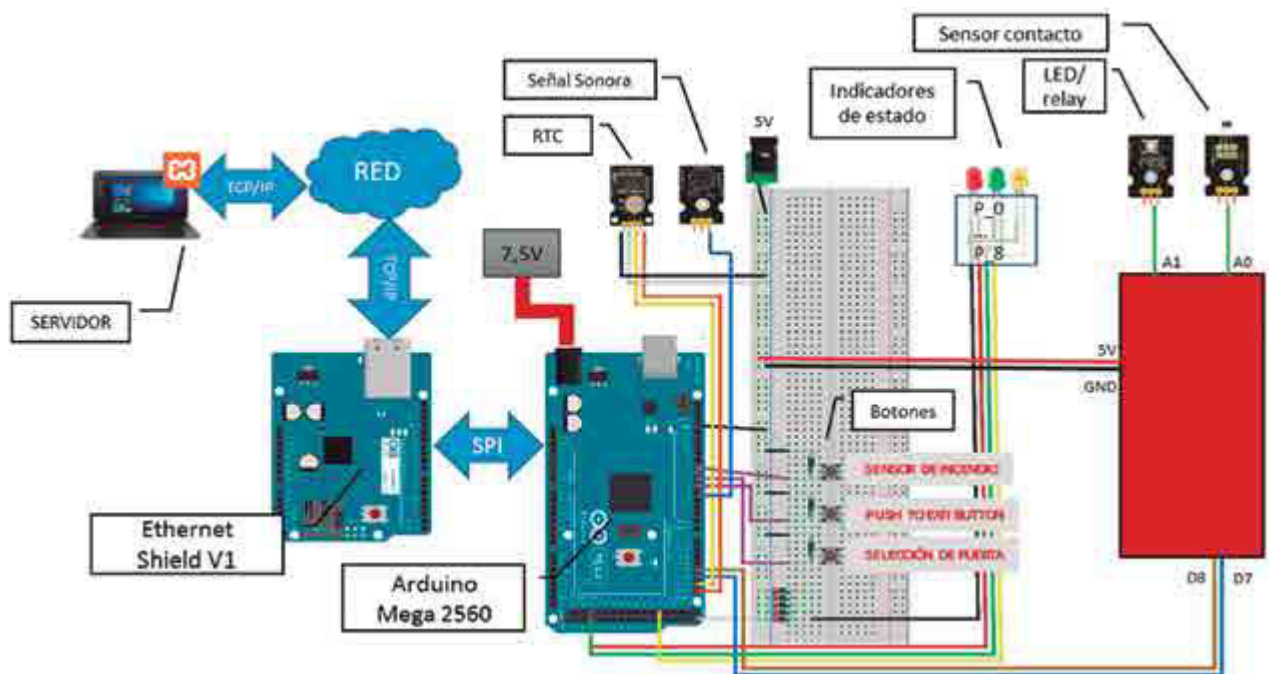


Figura 8. Componentes que integran la maqueta.

- Ethernet Shield V1 para la comunicación con la inteligencia en el Servidor.
- SM130 con la placa de evaluación como lector de las tarjetas.
- 9 juegos de leds compuestos por los colores rojo, amarillo y verde para representar el estado de las puertas y uno blanco más brillante para representar

el actuador de la puerta con sus respectivas resistencias de 220Ω .

- 3 Botones que permiten cambiar la puerta activa, simular un sensor de incendio y actuar como botón para la salida libre, con sus respectivas resistencias de $1K\Omega$.
- Un zumbador para las señales audibles.

- Dos fuentes de alimentación y respaldo, una de 7.5 v para el controlador y otra de 5v para el circuito de la puerta incluyendo el lector.

Integración entre los componentes del sistema

Aunque el controlador que está más cercano a las puertas posee cierto nivel de inteligencia, la decisión de apertura de puerta y la notificación inmediata al usuario recae sobre el servidor que es el que posee comunicación directa con la base de datos que contiene la información necesaria, de identificación, permisos, horarios y días que se puede acceder, si la habitación se encuentra bloqueada por un administrador del sistema, etc. lo que permite aceptar o denegar la apertura de una puerta.

El servidor de decisión está compuesto por dos secciones que se aprecian en la figura 10 y que pueden ser implementados por dos servidores independientes físicamente. Una parte atiende la interfaz gráfica para los usuarios programada en PHP haciendo uso de la herramienta Symfony 2, que a través de uno de sus módulos es posible la comunicación con la base de datos (BD) y de dos *scripts* principales que se ejecutan constantemente: “ForSocket.php” y “ForWebSocket.php”. El primero chequea el estado

de los controladores, responde a las peticiones de acceso y modifica los parámetros de los controladores basándose en consultas a la Base de Datos que puede encontrarse en otro sitio, y a través del segundo (ForWebSocket) se mantiene actualizada en tiempo real la plataforma de interfaz de usuario ya que es un sitio web en PHP.

Comunicación entre los módulos

Para la comunicación entre los controladores y la inteligencia global del sistema se define un protocolo de comunicaciones para cada evento, basándose en un resumen de mensajes (Tabla 2).

Estos mensajes se estructuran en una trama sencilla, sin agregar demasiados campos que de momento no sean necesarios ya que resulta importante el aprovechamiento del ancho de banda para prevenir latencias en las respuestas de los servidores ante una petición de acceso que pueden ser provocadas por el tiempo de procesamiento y por el de conmutación en los elementos de red.

Breve análisis de costo

Conociendo los costos de soluciones similares de distintos proveedores, se escoge KANTECH proveedor canadiense con presencia en Cuba, se hace un

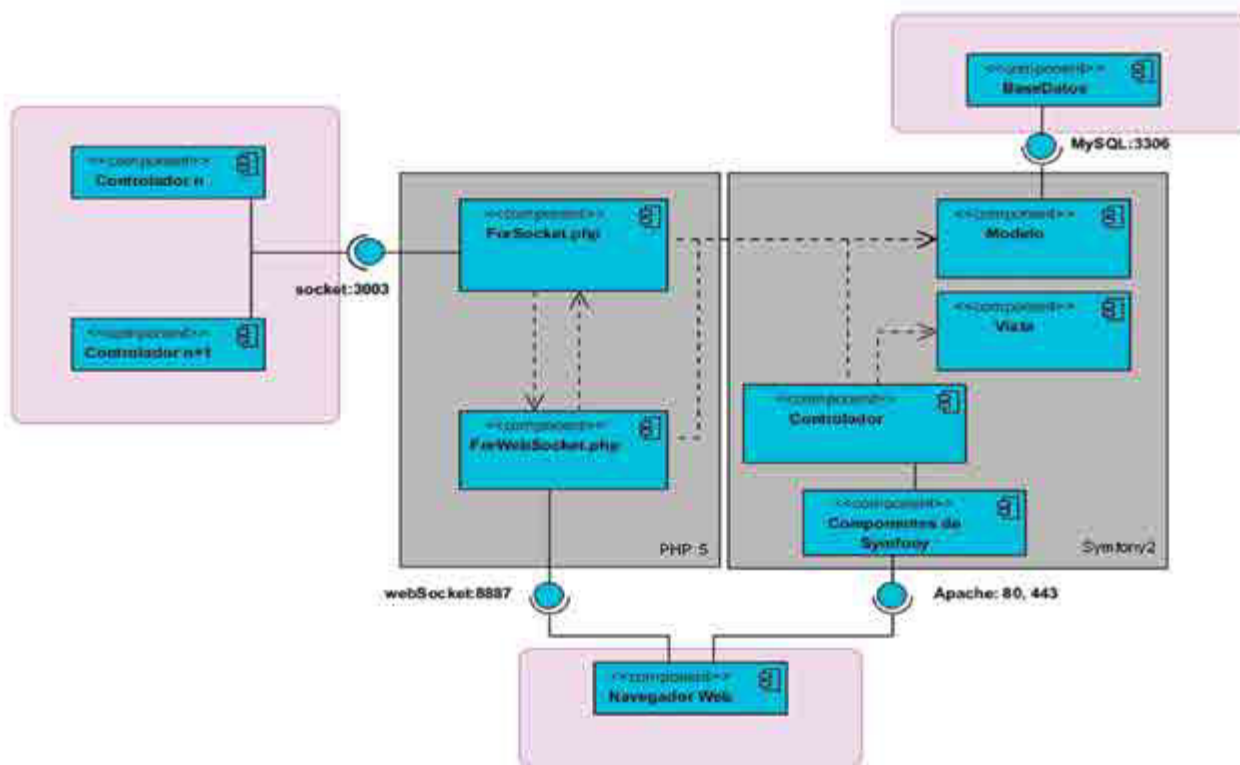


Figura 10. Distribución y conexiones entre los servidores.

No.	Descripción	Controlador	Plataforma
01	Petición de apertura de puerta	Envía REQUEST	Responde REPLY
02	Estado de puerta	Envía MSG	-
03	Heart beat	Envía HEART BEAT	Responde ACK
04	Señal de alarma	Envía WARNING	Responde COMANDOS
05	Comandos	Ejecuta	Envía COMANDOS
06	Hand shake	Envía CONTROLLER CONFIG	Responde SERVER CONFIG
07	Cambio de configuración	Responde CONTROLLER CONFIG	Envía NEW CONTROLLER CONFIG
08	Manejo de listas de acceso	Envía FULL LIST REQUEST	Responde FULL LIST
		Responde ACK	Envía FULL LIST
		Responde ACK	Envía UPDATE
		Responde ACK	Envía DELETE
09	Manejo de registros	Envía LOGS	Responde ACK

Tabla 2. Resumen de mensajes empleados.

Mensaje

MSG_ID /TYPE/ TAG_ID/PIN/CONTROLADOR_ID/ READER_ID/

Ejemplo de mensaje 01

01/RQ/13B4A545/4321/41/1/

TAG_ID: Es el código del testigo.

PIN: Número personal de identificación de 4 cifras

CONTROLADOR_ID: Es el identificador del controlador formado por los últimos dígitos de su ip.

READER_ID: Es el identificador del lector RFID determinado por el puerto serie al que está conectado, cómo solo son cuatro este número va de 0 a 3.

Tabla 3. Ejemplo de mensaje 01 tipo REQUEST.

análisis económico comparativo entre ambas soluciones para la maqueta diseñada. La tabla 4 contiene una comparación entre la solución KANTECH y dos variantes propuestas en la tesis de grado del autor principal, una alimentada y respaldada localmente -variante 2- y otra -variante 1- con alimentación y respaldo centralizado mediante PoE. La comparación tiene en cuenta la variación de los costos indirectos (transportación, salarios, almacenamiento...) en un porcentaje sobre el costo de producción (costo indirecto 0%). El análisis arroja que aún en el caso de que el costo indirecto sea igual al precio de costo del sistema, aún en la variante más costosa, se ahorra

30% con respecto a la solución que se implementa con KANTECH.

Conclusiones

Se diseñó un sistema de control de acceso basado en Arduino y sensores comerciales, con una arquitectura distribuida sobre IP utilizando e implementando el protocolo TCP/IP. El sistema posee numerosas funcionalidades que son accesibles a través de una plataforma web *responsive* por lo que se ejecuta en cualquier dispositivo con un navegador web sin importar la resolución. Se tiene en cuenta a la hora del diseño el despliegue físico en una instalación real, así como

Componente de costo indirecto (%)	0	40	50	80	100
Variante 1	\$8.545,13	\$9.172,68	\$9.329,57	\$9.800,23	\$10.114,01
Relación Costo V1/KANTECH	59,98%	64,39%	65,49%	68,79%	70,99%
Variante 2	\$7.850,85	\$8.178,57	\$8.260,50	\$8.506,29	\$8.670,15
Relación Costo V2/KANTECH	55,11%	57,41%	57,98%	59,71%	60,86%
KANTECH					\$14.246,31

Tabla 4. Comparación entre losN precios de las soluciones.

la ubicación espacial de los componentes del controlador dentro de un *Housing* determinado.

El sistema es modular y adaptable a cualquier instalación. La aplicación de gestión web funciona acorde a una lista de requerimientos funcionales bastante abundante y variada, es bastante rápida, amigable al usuario y permite la configuración desde cero de cualquier instalación. El registro de los sucesos es detallado y es posible el filtrado lo que resulta necesario para cualquier sistema de control de acceso. En base al análisis y diseño previo se implementa en un modelo a escala de una instalación genérica,

y construido con ese propósito. Se simulan los sensores y actuadores que no se poseen y se señalizan por medio de leds los estados de las puertas para comprobar el funcionamiento. Se prueban todos los sistemas y la integración del software web con los sensores de las puertas y los controladores en varios casos de uso.

Queda demostrado que es posible una solución de control de acceso basada en Arduino, que se puede desarrollar una solución propia en un tema tan sensible, a mejores precios incluso que los sistemas comerciales disponibles.

Referencia

- Banzi, M. y Michael Shiloh. (2015). *Make: Getting Started with Arduino* (3rd Edition). Maker Media, SEBASTOPOL. CA.
- Clarke, N. (2011). *Transparent User Authentication: Biometrics, RFID and Behavioural Profiling* (2011 edition). London : New York: Springer.
- Díaz, P. (2017). *Propuesta de Sistema de Control de Acceso Basado en Arduino y RFID*. Universidad Tecnológica de La Habana «José Antonio Echeverría», La Habana, Cuba.
- Esaú A. (2015, septiembre 28). Los 10 Frameworks PHP que solicitan las empresas. Recuperado 30 de mayo de 2017, de <https://openwebinars.net/blog/los-10-mejores-frameworks-php-que-solicitan-las-empresas/>
- Kusriyanto, M. y Putra, B. D. (2016). Smart home using local area network (LAN) based arduino mega 2560. En *2016 2nd International Conference on Wireless and Telematics (ICWT)* (pp. 127-131). <https://doi.org/10.1109/ICWT.2016.7870866>
- Norman, T. L. (2011). *Electronic Access Control* (1 edition). Waltham, MA: Butterworth-Heinemann.
- Novacek, G. (2016, septiembre 13). Tips for Predicting Product Reliability [Revista Científica]. Recuperado 2 de junio de 2016, de <http://circuitcellar.com/cc-blog/tips-for-predicting-product-reliability>
- Saravia, M. W. D. (2015). Access control system using NFC and Arduino. En *2015 IEEE Thirty Fifth Central American and Panama Convention (CONCAPAN XXXV)* (pp. 1-6). <https://doi.org/10.1109/CONCAPAN.2015.7428472>