INGENIERÍAS USBMed

https://revistas.usb.edu.co/index.php/ingUSBmed

ARTÍCULO DE INVESTIGACIÓN

# Módulo de control de acceso a zonas restringidas de la granja, Román Gómez Gómez del Politécnico Colombiano Jaime Isaza Cadavid sede Marinilla Antioquia, Colombia

## Access control module to restricted areas of the farm, Román Gómez Gómez of the Politécnico Colombiano Jaime Isaza Cadavid headquarters Marinilla Antioquia, Colombia

Juan Pablo Pizarro Duque[1]
William S Puche P[2]

[1]Ingeniería Informática, Facultad de Ingeniería, Politécnico Colombiano Jaime Isaza Cadavid, Medellín, Colombia. Email: juan_pizarro82141@elpoli.edu.co

[2]Tecnología en Infraestructura de Telecomunicaciones, Politécnico Colombiano Jaime Isaza Cadavid, Medellín, Colombia. Email: wspuche@elpoli.edu.co



**Tipo de artículo:** Investigación.
**Recibido:** 27-10-2020.
**Revisado:** 19-03-2021.
**Aprobado:** 19-07-2021.
**Doi:** 10.21500/20275846.5071

**Referenciar así:**
Juan P Pizarro and William S Puche, "Módulo de control de acceso a zonas restringidas de la granja, Román Gómez Gómez del Politécnico Colombiano Jaime Isaza Cadavid sede Marinilla Antioquia, Colombia," *Ingenierías USBMed*, vol. 12, n.° 31, pp. 17-32, 2021.

**Disponibilidad de datos:**
todos los datos relevantes están dentro del artículo, así como los archivos de soporte de información.

**Conflicto de intereses:**
los autores han declarado que no hay conflicto de intereses.

**Editor:** Andrés Felipe Hernández. Universidad de San Buenaventura, Medellín, Colombia.

**Resumen.** Este articulo plantea la implementación prototipo de un módulo de seguridad y control de acceso en zonas restringidas, donde se realizan procesos de investigación y producción en la granja Román Gómez Gómez del Politécnico Colombiano Jaime Isaza Cadavid, para mejorar el sistema actual de control de acceso basado en tecnología QR, aplicado en la entrada principal de la granja. Se caracterizará, todos los elementos de hardware y software de la aplicación existente; a partir de esta, se diseña un protocolo de seguridad para autenticar y controlar el acceso de personas; posteriormente, se construye el módulo que contiene el protocolo, donde se interviene el código fuente de la aplicación, para autorizar y registrar las visitas en cada una de las zonas visualizando tiempo real, finalmente se validan las mejoras sobre el sistema actual mediante una prueba de funcionamiento.

**Palabras Clave.** Protocolo, IoT, Modulo de seguridad, Tecnología QR, Control de acceso, Hardware, Aplicación en tiempo real.

**Abstract.** This article proposes the prototype implementation of a security and access control module in restricted areas, where research and production processes are carried out at the farm Roman Gómez Gómez of the Politécnico Colombiano Jaime Isaza Cadavid Jaime Isaza Cadavid, to improve the current access control system based on in QR technology, applied at the main entrance of the farm. All the hardware and software elements of the existing application will be characterized; Based on this, a security protocol is designed to authenticate and control people's access; Subsequently, the module containing the protocol is built, where the source code of the application is intervened, to authorize and record visits in each of the areas viewing real time, finally the improvements over the current system are validated by means of a test of functioning.

**Keywords.** Protocol, IoT, Security module, QR technology, Access control, Hardware, Real-time application.

## I. Introducción

Actualmente los centros de investigación agrícolas buscan mejorar los procesos de seguridad en sus recintos, con la finalidad de mitigar riesgos y afectación en sus proyectos. De igual forma, ocurre en las empresas muchos de los procesos que se manejan hoy en día y que otorgan seguridad en las organizaciones han avanzado en optimizar y mejorar cada vez más las vulnerabilidades y facilitando el control de los procesos. Muchos de los procesos se rigen bajo algún protocolo de seguridad, siendo así, indispensable que algún ente o empresa sigan ciertos protocolos que agilicen y brinden seguridad en todo momento, esto permite la implementación de hardware y software logrando que los protocolos puedan articularse de manera efectiva, mitigando así, el riesgo y vulnerabilidades expuestas que pueden no ser tenidas en cuenta por descuido, olvido y error de los seres humanos, al aplicarse de forma tradicional sin emplear estas herramientas tecnológicas [1], [2]. Los sistemas de seguridad han evolucionado adaptándose al medio tecnológico actual, específicamente los sistemas de control de acceso modernos, del mismo modo, asignan un amplio espectro el cual genera seguridad y confianza a la hora de proteger y restringir el ingreso de personas en ciertas zonas específicas [2], por consiguiente este articulo abarca una estrategia o protocolo de seguridad que permite autenticar las persona que desean acceder a una zona específica de la granja Román Gómez Gómez del Politécnico Jaime Isaza Cadavid del municipio de Marinilla Antioquia Colombia, implementado en un módulo adicional para la aplicación que actualmente se emplea para el acceso a la granja, pero, que carece de seguridad y control sobre las zonas restringidas que visitan los individuos una vez están dentro de esta. Por lo tanto este artículo, busca caracterizar los dispositivos de hardware con los que ya cuenta la institución y la aplicación existente e identificar los requisitos para su correcto funcionamiento, con el fin de llevar a cabo la solución adecuada del problema, diseñando un protocolo de seguridad de ingreso a las zonas de la granja, para evitar riesgos de seguridad en las zonas visitadas y de salud en los animales, de igual forma se construirá un prototipo del módulo de seguridad para autenticar y controlar el acceso adecuado de personas en las zonas de la granja permitiendo visualizar registro de visitas en tiempo real, esto se lograra verificar validando el prototipo a partir de una prueba que demuestre la mejora efectiva en el proceso de seguridad.

## II. Problemática

Actualmente, *la mayor parte de las organizaciones apoyan fuertemente su actividad en las tecnologías de la información y de las comunicaciones, componentes esenciales hoy en día de sus sistemas de información* [3], es por ello, que la seguridad en el contexto tecnológico tiende a evolucionar, buscando que cada vez se otorgue un nivel de confianza más alto respecto a la protección y aseguramiento datos, así como también, diferentes recursos físicos, personas y animales, de los cuales, es indispensable tener debido control y el acceso en todo momento. Las instituciones universitarias buscan una modernización en los procesos de control de acceso que agilicen métodos tradicionalmente empleados [4], [5], es por esto que el Politécnico Colombiano Jaime Isaza Cadavid - PCJIC presenta la necesidad de mejorar los procesos de ingreso donde la seguridad es el principal motivo para controlar y autenticar las personas en zonas restringidas al interior de todas sus sedes en el departamento de Antioquia. El PCJIC cuenta con una granja ubicada en el municipio de Marinilla, la cual está dividida por zonas de producción e investigación en las que se encuentran animales cultivos de diferentes especies, estas zonas son visitadas por estudiantes, docentes y diferentes entes interesados, sin restricción alguna dentro de la granja. La granja Román Gómez Gómez del PCJIC, actualmente dispone de un sistema de control de acceso que utiliza tecnología QR [6], [7], mediante un dispositivo que interactúa con una aplicación web, solo para limitar el ingreso general a la granja, con fallas de seguridad y de recopilación de información con riesgos que permiten una vez las personas están dentro de la granja, ingresan a todas las zonas sin ningún control ni registro de los lugares o zonas visitadas dentro de la granja. Esta falta de control, genera inseguridad y preocupación en la granja, ya que las personas no autorizadas en ciertas zonas, están ingresando y manipulando los elementos o herramientas, así como las especies que allí se encuentran, por el inadecuado manejo, lo que aumenta el riesgo de salud en las criaturas y afectación en los procesos de investigación que se esté realizando sobre estas, por lo que se requiere prevenir dichas circunstancias y a su vez controlar los riesgos que comprometan los recursos y bienes del PCJIC [8], [9]. Es importante denotar que la granja del municipio de Marinilla es certificada como *Granja Alta Calidad por sus buenas prácticas* ante el ICA (Instituto Colombiano Agropecuario).

Es por esto, que se identifica la necesidad de mejorar la seguridad respecto a los procesos de ingreso en zonas restringidas al interior de la granja Román Gómez Gómez, ajustando los procesos actuales como base hacia un protocolo que conceda control y autenticación de usuarios cuando ingresen a una zona o en el sentido contrario, que deniegue el acceso a la zona cuando la persona no esté autorizada y así, llevar un respectivo registro de visitas en todo momento [10], [11]. Esto permitirá al PCJIC mejorar sus procesos de seguridad de las zonas dentro de la granja; beneficiando el control sobre la cantidad adecuada de personas en los diferentes espacios restringidos, del mismo modo, un bienestar

para los animales, mejora el cumplimiento de control de asepsia y sanidad ya que se autentican personas con el conocimiento adecuado del manejo en la zona, mejora la seguridad de la institución y así mismo, se verán beneficiados los miembros pertenecientes a la facultad de ciencias agrarias de la institución, con una mejor experiencia en sus labores. Apoyado de un módulo de seguridad adicional, se actualiza y mejora la aplicación existente como el proceso de ingreso, este se ajustará a un protocolo que siempre se cumpla de manera efectiva con el fin de autenticar igualmente con códigos QR a los usuarios y para el control de acceso, ahora dirigido a las zonas restringidas y que permita mediante este módulo, recopilar los registros de las zonas visitadas en tiempo real, con información tangible para un mayor propósito.

## II. Procedimiento 1

En esta sección se detalla el proceso realizado desde la caracterización de los dispositivos de hardware con los que ya cuenta la institución y la aplicación existente [6] e identificar los requisitos para su correcto funcionamiento, con el fin de llevar a cabo la solución adecuada del problema [12]-[14].

### A. Análisis del funcionamiento de la aplicación existente

Se compone de una aplicación web que permite a un usuario mediante un formulario realizar una solicitud donde, diligencia sus datos personales, selecciona una fecha de visita y un redacta motivo de visita, para el ingreso a la granja Román Gómez Gómez del PCJIC, las solicitudes de ingreso son respondidas por un administrador que acepta o rechaza la solicitud del usuario. Si la solicitud es aceptada se le envía un email al usuario que contiene un código QR de ingreso a la granja y la fecha única de valides del mismo. Este código posteriormente se valida en la entrada de la granja donde es escaneado y verificado en la base de datos, para conceder o denegar el acceso a la granja por medio de un módulo scanner QR.

*Las tecnologías y dispositivos que usan:*

El desarrollo fue construido en su mayor parte en el lenguaje de programación orientado a objetos JavaScript [10], [11].

**Arquitectura:** Implementan la arquitectura de software Cliente-Servidor [15], [11], como se muestra en la Figura 1. Esta arquitectura es de gran ventaja y permite que se pueda intervenir en el código fácilmente y realizar cambios que comprenden el módulo de seguridad a implementar.

**Cliente:** Esta construido en React Js, [15], la gran ventaja es que permite un mayor dinamismo en la página por los diferentes componentes que se reutilizan.

**Servidor:** Esta construido en Node Js [16], [11], para el desarrollo simple del servidor utilizan el framework Express [17], que proporciona un fácil desarrollo de la APIRESTful.

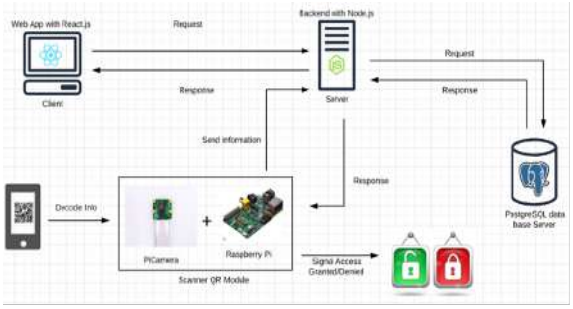


**Figura 1.** Diagrama de arquitectura de software implementada

**Base de datos:** Una base de datos de tipo relacional, Implementada en el sistema gestor de PostgreSQL, construida en lenguaje PSQL, que es similar y basado al muy conocido SQL (Structured Query Language). Con únicamente tres tablas las cuales no tienen relación y las consultas se efectúan en alguna tabla, por lo que abunda redundancia.

**Autenticación:** Se basan en la tecnología QR que significa (Quick Response), o código de respuesta rápida, donde utilizan códigos QR para almacenar información de la solicitud de un visitante que tiene acceso a la granja. La información se codifica en una matriz de puntos bidireccional conformada por módulos de color blancos y negros. En la Figura 2. se presenta la estructura y características generales de los códigos QR.

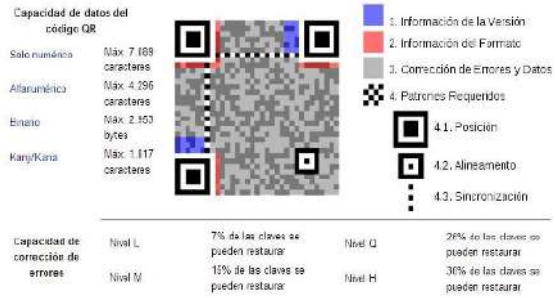


**Figura 2.** Estructura y características de los códigos QR [18]

**AWS:** Utilizan el servicio de almacenamiento en la nube llamado S3 donde almacenan códigos QR.

**Modulo Scanner QR:** Esta desarrollado en Python, [19]. También Utiliza librería OpenCV para el manejo de datos multimedia al momento de leer el código QR, la plataforma o dispositivo físico en donde se corre este módulo es una Raspberry pi 3 modelo B (micro procesador o mini pc) y para la lectura de los códigos se utiliza una pi camera v1.3, (un dispositivo de hardware adicional propio de la raspberry).

**Marco Legal:** Se sujeta a la Ley 1581 de 2012 artículo 189 de la constitución política de Colombia, para

el tratamiento, manejo adecuado y la protección de los datos personales de los usuarios, la cual se identifica en [20].

### B. Resultados del análisis procedimiento 1

Teniendo en cuenta lo anterior, se identificaron varios aspectos a corregir [6] y tener en cuenta para el posterior diseño del módulo de seguridad, esto comprende de requisitos para el correcto funcionamiento y cumplimiento de la solución al problema.

- El proceso de solicitud de visitas es ambiguo, de esta forma se necesita que las zonas y su disponibilidad dependan de una fecha visita, por si en algún momento la fecha de visita no es posible para alguna zona debido a algún evento fortuito o simplemente no es posible el ingreso de más visitantes.
- Se necesita establecer cierta cantidad tope de visitantes por zonas, esto evitaría un desborde de visitantes además de facilitar el control y seguridad de cada zona.
- En el formulario de solicitud, es necesario que el tipo de persona se obtenga directamente de la base de datos para posteriores cambios, no establecidos de forma estática.
- El motivo de visita debe ser más preciso donde el usuario indique específicamente a cuáles zonas son las que desea visitar.
- Al momento de que el usuario envía el formulario al servidor, la respuesta se muestra con un mensaje estático de la vista, capturando las respuestas desde el servidor para mostrar al usuario lo ocurrido.
- En la sección de solicitudes pendientes, cada card, con la solicitud debe de contener qué zonas fueron las solicitadas por el usuario, y el administrador debe de elegir cuáles se van a autorizar y cuáles no, con el fin de generar el código con las zonas que realmente tiene permiso autorizado y la autenticación del usuario.
- Para usuarios de tipo empleado, quienes deben estar en constante ingreso en las zonas, deben llevar el control adecuado y registro, ya sean docentes investigadores de alguna zona o personal de aseo, vigilancia y mantenimiento, es necesario generar un tipo de permiso especial donde el código QR de ingreso en las zonas sea siempre vigente sin fecha de caducidad, hasta una nueva orden, dirigido únicamente a las zonas permitidas o que tienen a su cargo.
- El administrador antes de aceptar, rechazar o dar algún permiso especial de una solicitud, debe de pronunciar una observación al usuario, esto como campo adicional en la sección de solicitudes pendientes además adicionar un botón para el permiso especial.
- En la sección de solicitudes aprobadas se debe modificar la manera en que opera ya que no se debe eliminar en ningún momento los registros, pues es sumamente importante que los datos no sean eliminados de una base de datos por buenas prácticas y para llevar un control de todas las acciones en la misma, además de todas las solicitudes y permisos.
- Es necesario diseñar una base de datos nueva ya que la base de datos actual solo comprende tres tablas (*solicitud_ingreso*, *solicitud_aprobada*, *usuario_admin*) y no hay un registro de usuarios adecuado, además hay información redundante por lo que en la tabla solicitud aprobada se insertan los mismos campos de *solicitud_ingreso*, adicional a eso no hay una relación entre las tablas que pueda aportar información útil.
- Los códigos QR no se deben de almacenar en ningún, por lo tanto, el servicio de AWS S3 no será tenido en cuenta en la implementación.
- Los códigos QR se generan al instante de realizar la solicitud y este proceso está mal, ya que se deben generar una vez se apruebe la solicitud, por lo que se están almacenando códigos que nunca serán utilizados.
- Los emails no solo deben de informar si la solicitud del usuario fue aceptada, también debe informar si la solicitud fue rechazada.
- La estructura del email enviado debe contener las zonas a las que ha sido autorizado el visitante y observaciones si es necesario, por el administrador para informarle de manera formal de qué requiere al momento de ingresar a la zona.
- Es necesario implementar una sección de visitas, en la cual se registren las visitas en tiempo real con la información de fecha y hora que ingresa un usuario.
- El módulo de decodificación y validación de códigos QR solo está disponible para la entrada principal.
- La arquitectura es tolerante a cambios y se puede añadir el módulo sin conflictos en el funcionamiento.
- Es necesario seguir con el mismo lenguaje JavaScript ya que es una ventaja poder desarrollar backend (Node js) y frontend (React js) sobre una misma tecnología web además de poder hacer la aplicación más dinámica, mejorando los tiempos empleados en la construcción.

## Procedimiento 2

En esta sección se detalla el proceso de diseño para el protocolo de seguridad del ingreso a las zonas de la granja, con base en el proyecto existente [6] para evitar riesgos de seguridad en las zonas visitadas y de salud en los animales.

### A. Diseño de la nueva base de datos

Como punto de partida se establece un nuevo modelo de base de datos completo el cual describe cómo será el comportamiento y almacenamiento de los datos, fue necesario reemplazar la base de datos existente por problemas como redundancia de datos, información ineficiente, falta de relaciones, falta de tablas, ya que solo contaba con tres tablas. Las tablas *solicitud_ingreso* y *solicitud_aprobada* guardaban los mismos datos, no es una buena práctica, además, de que no brinda información de utilidad, por lo que se mezclaba la información

de la solicitud y el usuario, la tabla *usuario_admin* que contenía las credenciales del administrador no posee problemas. Por lo tanto, se realiza un nuevo diseño del modelo de base de datos, el cual se diseña estratégicamente para cumplir todas las necesidades que debe abordar el módulo de seguridad. Esto se evidencia en la Figura 3.

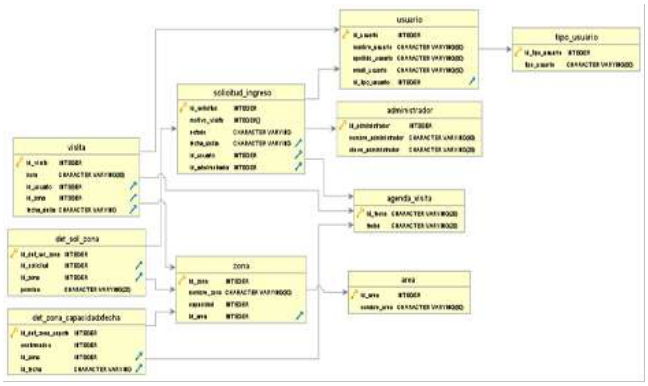


**Figura 3.** Modelo entidad relación de la base de datos

Se establecieron diez (10) tablas las cuales se relacionan para brindar información de utilidad, eficiencia en las consultas, además eliminamos la redundancia, y se añadieron tablas con el fin de poder dar una vía para el correcto cumplimiento del protocolo que constantemente requiere de información específica y de fácil acceso Tabla 1.

**Tabla 1.** Descripción de las tablas del modelo de datos

| Nombre tabla | Función |
|---|---|
| administrador | Contendrá las credenciales para el ingreso a la plataforma. |
| agenda_visita | Contendrá las fechas con visitas agendadas. |
| área | Contiene las áreas de programa que ofrece la granja. |
| det_sol_zona | Contiene datos del permiso de cada zona con relación a una solicitud. (Authorized o Denied). |
| det_zona_capacidadxfecha | Contiene el número de visitantes confirmados para una zona ligada a una fecha. |
| solicitud_ingreso | Contiene los datos de la solicitud con relación a un usuario y un administrador que resuelve la solicitud. |
| tipo_usuario | Contiene los perfiles de ingreso permitidos en la grana. |
| usuario | Contiene datos personales del usuario. |
| visita | Contiene registros de cada usuario que ingresa a una zona ligada a la fecha y hora de cada ingreso. |
| zona | Contiene las zonas con relación a un área perteneciente a un programa de la granja. |

## B. Diseño del protocolo

Una vez realizado el diseño de la base de datos, se procede con el diseño del protocolo, este se desarrolla basado en varias etapas las cuales tanto el administrador, el usuario, el cliente, el servidor y el módulo scanner cumplen roles y actividades que se deben lograr para poder garantizar el control y la seguridad en las zonas, ya que el ingreso será especifico de personas autorizadas y la cantidad permitida de ingreso por zona en todo momento será controlada.

## C. Etapa 1 – Generar solicitud:

El proceso de generar solicitudes se realiza mediante un formulario donde el usuario diligencia sus datos personales y demás datos sobre la visita, fue modificado especialmente para brindar mayor interacción entre componentes del cliente y el servidor, con fin de eliminar los objetos estáticos del lado del cliente, esto quiere decir que todos los recursos se consumen de la API-RESTful y de forma dinámica se actualizan los datos del componente según la elección del usuario, Se garantiza una elección especifica de zonas disponibles según la fecha. En la Figura 4 se presenta de forma resumida y clara el funcionamiento de la Etapa 1 del protocolo, en un esquema, donde los recuadros de color verde encierran los componentes del cliente añadidos y modificados los cuales están en constante interacción con la API del servidor y se observa la dependencia entre el componente zona respecto a fecha.

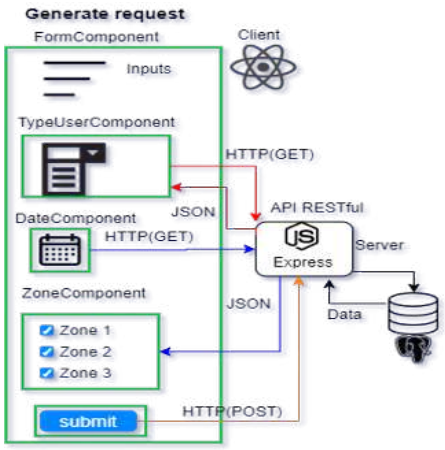


**Figura 4.** Esquema de funcionamiento del protocolo etapa 1

## D. Etapa 2 - Gestión de solicitudes

Esta etapa internamente se dividió en dos subetapas, puesto que se van a gestionar solicitudes pendientes y solicitudes que ya fueron aprobadas.

### 1) Etapa 2.1 - Solicitudes pendientes

En el proceso de gestión de solicitudes pendientes se realizan modificaciones a los cards donde se despliega la información de cada solicitud que contendrá datos personales del usuario, visita y específicamente zonas solicitadas por el usuario, además, un espacio de observaciones. El administrador podrá realizar acciones como aceptar solicitud, permiso especial y rechazar la solicitud. Para los casos "Aceptar" y "Especial", se debe seleccionar cuales zonas se van a autorizar y dejar una observación si es necesario, se envía la respuesta al servidor, se actualiza la base de datos y genera un código QR de autenticación para las zonas autorizadas, que posteriormente se envía por correo al usuario, la única diferencia es que un permiso especial no tiene fecha de caducidad para el ingreso hasta nueva orden. Ahora, para el caso de "Rechazar" no se debe seleccionar ninguna zona, pero sí dejar una observación, se envían los datos al servidor y se actualiza la base de datos para luego enviar un correo de respuesta al usuario donde indique que ha sido rechazada la solicitud. A continuación, se muestra el esquema de funcionamiento resumido para la etapa 2.1 Figura 5.

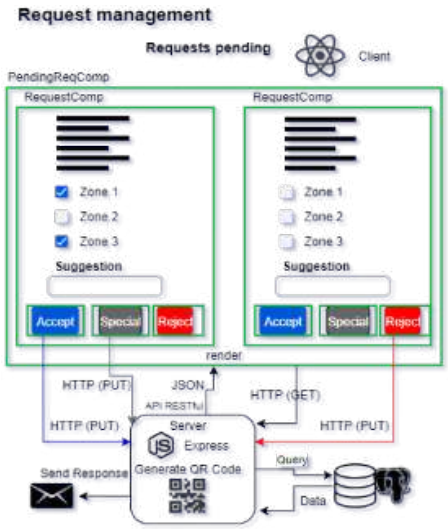


**Figura 5.** Esquema de funcionamiento del protocolo etapa 2.1

### 2) Etapa 2.2 – Solicitudes aprobadas

El proceso de gestión de solicitudes aprobadas fue modificado para no eliminar ningún registro en la base de datos, pero sí alterarlo en su estado, se basa en la acción "Rechazar" de la subetapa anterior, pero la diferencia es que se anulan solicitudes o permisos que antes fueron aceptados, esto es necesario por motivos de seguridad y control sobre las personas que dejan de tener acceso en las zonas. El administrador envía los datos al servidor y se realizan las modificaciones en la base de datos, luego el usuario recibe un correo donde se le informa que sus permisos fueron anulados.

A continuación, se muestra de forma resumida el esquema de funcionamiento para la anulación de solicitudes aprobadas. Figura 6.

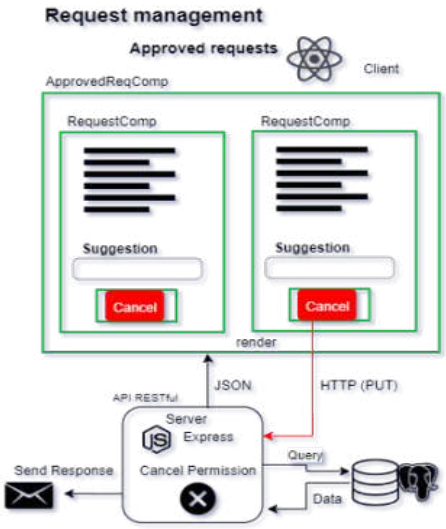


**Figura 6.** Esquema de funcionamiento del protocolo etapa 2.2

### E. Etapa 3 - Validación de ingreso

El proceso de validación de ingreso de usuarios fue modificado para que un usuario se dirija a cada zona donde presentara el código QR de autenticación, funciona mediante el módulo scanner que estará ubicado en cada zona, fue intervenido para la manipulación de los datos una vez son decodificados, para convertirlos en formato JSON [17], [19] junto con la zona donde se encuentra realizando la lectura, para luego enviar la petición HTTP de tipo POST al servidor, puesto que ahora se debe registrar la visita en la base de datos, solo si la validación es correcta, una vez el servidor comprueba los datos de autenticación del usuario en la zona, envía una respuesta para autorizar o denegar el acceso. A continuación, en la Figura **??**, se muestra de forma resumida el funcionamiento de la etapa 3 mediante un esquema.

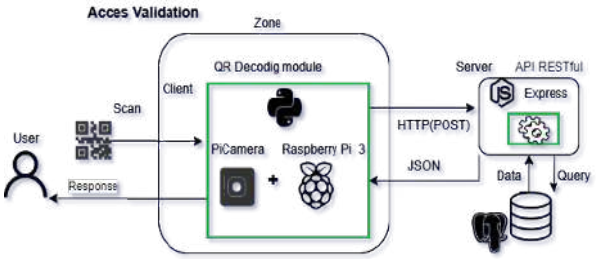


**Figura 7.** Esquema de funcionamiento del protocolo etapa 3

### F. Etapa 4 Control

El proceso de control de vistas se añadió como necesidad de visualizar lo que ocurre en tiempo real en

cada zona, su funcionamiento se basa en la tecnología WebSocket, usando la dependencia Socket.io en ambos sentidos, tanto el lado del cliente como del servidor, se basa en eventos que se emiten y se escuchan por un canal de intercambio de datos bidireccional, lo que permite la visualización en tiempo real de los registros de ingreso que se presentan en las zonas, así mismo, se añadieron componentes para la interacción con un filtro especial por zonas y por fecha, para un mejor enfoque de control en las zonas, los datos se visualizan en una tabla. Se puede observar el funcionamiento en un esquema de la Figura 8.

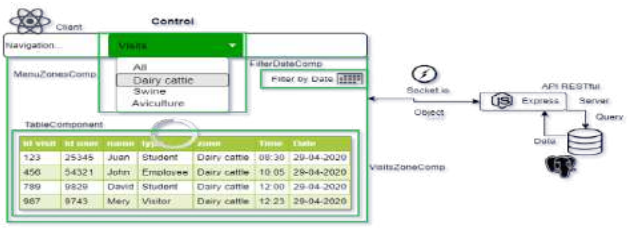


**Figura 8.** Esquema de funcionamiento del protocolo etapa 4

## IV. Procedimiento 3

En esta sección detalla la construcción del prototipo módulo de seguridad implementando el protocolo de ingreso, para autenticar y controlar el acceso adecuado de personas en las zonas de la granja y permitir visualizar registro de visitas en tiempo real.

### A. Implementación de la base de datos

Anteriormente se había mencionado, la base de datos estaba implementada bajo el motor de base de datos relacional PostgreSQL, por lo que se opta continuar con este gestor, puesto que nos provee buenas herramientas tanto gráfica y de consola para crear la nueva base de datos, realizar pruebas de las consultas y las respectivas modificaciones, además, es gratuita y de código abierto. Según el modelo de datos creado en el procedimiento 2, se procede a realizar la creación de la base de datos y posteriormente las tablas por la terminal de comandos, se utiliza lenguaje PSQL y como resultado obtenemos las siguientes tablas que se muestran en la Figura 9.

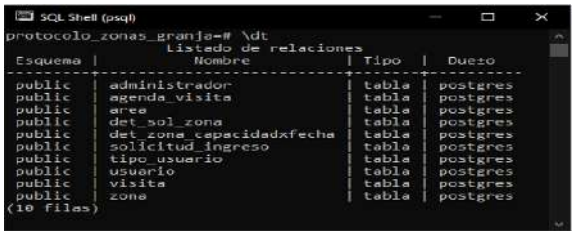


**Figura 9.** Base de datos y lista de tablas creadas en PostgreSQL

En la Figura 10, se muestra un ejemplo de la herramienta utilizada para la administración de la base de datos y así mismo, para realizar las pruebas de consultas SQL.

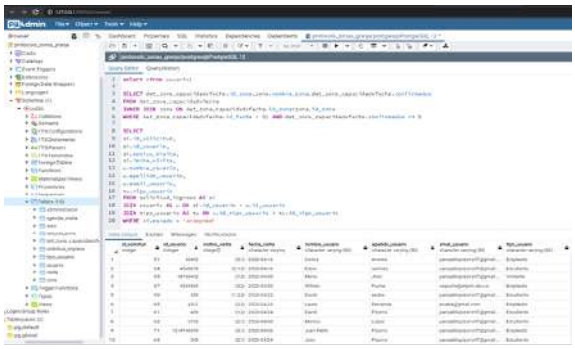

**Figura 10.** Herramienta PgAdmin

### B. Cambios en el Backend

El backend fue intervenido en muchos de los archivos realizando modificaciones (M de color amarillo) adicionándose otros archivos con el fin de mejorar la lógica y de implantar el protocolo (U de color verde), esto se puede observar en la Figura 11,además, la estructura del proyecto sigue siendo la misma con leves cambios y adiciones que no afectan en el funcionamiento, como los scripts de la base de datos, el socket de conexión y el protocolo donde se maneja todo lo que tiene que ver con las solicitudes, con el objetivo de reutilizar todas las funciones fácilmente y abstraer los métodos para añadir nuevas rutas que serán gestionadas desde la API RESTFUL.


**Figura 11.** Estructura y modificaciones en el Backend

A continuación, se explican los cambios y mejoras en cada uno de los directorios de la estructura del proyecto:

• **Controllers:** Esta carpeta contiene todos los archivos o clases donde se definen los controladores, para

resolver cada una de las rutas de la API. Cada controlador ejecuta métodos necesarios para tratar y manipular la información recibida y a entregar al cliente. Se modificaron y añadieron características a los controladores, siguiendo el protocolo que se encuentra en la clase "protocoloSolicitud", que es donde se constata cada método que será utilizado por los controladores y además, se añadió el controlador "visitasController" para manejar los métodos referentes a el registro y visualización de visitas, también, se comunica el "clienteController" con la clase "socket", que se encuentra en el directorio "plugins", para manejar los métodos de validación del QR leído en una zona y avisar que hubieron cambios, para que el socket pueda emitir al cliente más adelante.

• **Plugins:** Los directorios contienen los métodos utilizados para enviar emails a el usuario, para generar códigos QR [21] y además de establecer sockets de conexión.

• **Mail_sender:** Fue modificado para añadir ciertas características, con el fin de mejorar el entendimiento por parte del usuario, puesto que solo se enviaban emails con el id, código QR y fecha en el caso de que la solicitud fuera aceptada. Las modificaciones fueron para emails de: Permiso especial, Solicitud rechazada y Permiso anulado. Todos con cambios en la estructura y contenido del email según el caso. Cabe resaltar que se continúa empleando la librería Nodemailer, [22] y no fue necesario cambiar la librería porque, es fácil de usar y cumple con las necesidades de enviar los emails correctamente, bajo los parámetros modificados que se necesitan.

• **Qr_Code:** Se modificó la forma en generar tanto el id del código QR como la codificación del mismo con la información adicional. La información que se codifica fue modificada añadiendo campos nuevos para detallar y saber que permisos tiene un usuario, en que zonas y cuando, en la Figura 12 y Figura 13 muestran los dos casos de códigos que se generan.


**Figura 12.** Estructura de código QR para solicitud aceptada

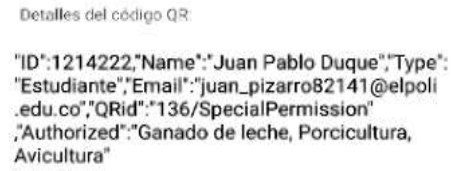


**Figura 13.** Estructura de código QR para permiso especial

Se cambia la librería para generar los códigos QR "qr-image" por "qrcode", la razón es que para "qrcode" hay un mayor seguimiento y desarrollo de actualizaciones, lo que permite un respaldo y fuente de información para posibles cambios futuros, este cambio no afecta en el proceso de generar códigos. Esta librería proporciona un método para corregir el nivel de error [23], teniendo en cuenta de que la información es poca para codificar y se desea realizar una lectura rápida, se eligió un nivel Medio "M", con la finalidad de garantizar lectura rápida en el módulo scanner, puesto que con un nivel alto "H" tarda un poco más de tiempo en reconocer el código y se vuelve un poco tedioso a la hora de agilizar el proceso de lectura, además, garantizar de que si el usuario presenta el código digital o impreso sea posible su lectura, donde se corre el riesgo medio de pérdida de legibilidad en una superficie vs una pantalla donde no hay pérdida. En la Tabla 2, se observa los niveles y porcentaje de resistencia a error de lectura.

**Tabla 2.** Niveles y resistencia al error de qrcode [21]

| Nivel | Resistencia al error |
|---|---|
| $\mathbf{L}_{(\text{bajo})}$ | $\sim 7\,\%$ |
| $\mathbf{M}_{(\text{medio})}$ | $\sim 15\,\%$ |
| $\mathbf{Q}_{(\text{cuartil})}$ | $\sim 25\,\%$ |
| $\mathbf{M}_{(\text{alto})}$ | $\sim 30\,\%$ |

Por último, el método "qrUpload" fue eliminado, puesto que los códigos se estaban almacenando y muchos nunca se estaban usando, por lo que al momento de que el usuario enviaba la solicitud, el código se generaba sin que el administrador aceptara la solicitud de ingreso, así que se almacenaba en el servicio AWS S3, lo que resulta totalmente innecesario, porque es información que ya existe en la base de datos lo que causa redundancia, más consumo de recursos y por ende más tiempo de ejecución. Además, cabe resaltar que, al momento de enviar un email con el código, este queda almacenado en el correo electrónico del usuario y de la administración de la granja.

Socketio: Se agregó este directorio o capa para establecer websockets [24], esta herramienta es necesaria para establecer un canal de transferencia de datos bidireccional entre el cliente y el servidor, puesto que para realizar la sección de visitas es importante mantener un intercambio de datos en tiempo real, sobre lo que ocurre en cada una de las zonas al momento que ingresa un usuario, para que dicho evento se refleje en el mismo instante. La forma de implementar los WebSockets de manera eficiente se hará utilizando la librería Socket.io [24], además, "Socket.IO permite la comunicación en tiempo real, bidireccional y basada en eventos. Funciona en todas las plataformas, navegadores o dispositivos, centrándose igualmente en la confiabilidad y la velocidad" [25], es por ello que se decide utilizar

esta librería. En primera instancia fue necesario hacer unos cambios en la configuración del servidor, por lo que Express directamente no funciona con "socket.io". Se importó la librería integrada de Node js "http" que permite levantar el servidor y usar los sockets, luego, se crea una variable "server" donde se define el servidor usando "http", se parametriza con la constante "app" de Express, puesto que Express está basado en "http", posteriormente, se reemplaza app por "server", donde se llama a la función "listen" y finalmente, requerimos el archivo donde se va a inicializar el socket. Se puede observar la configuración en la Figura 14.

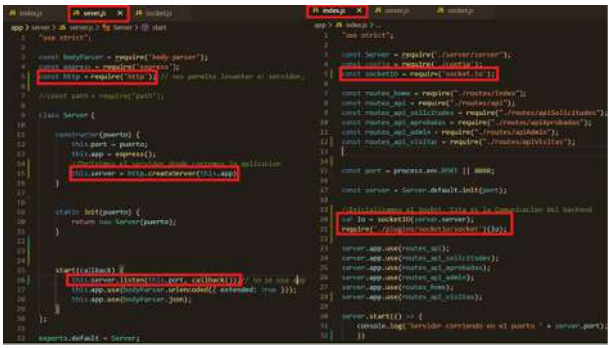

**Figura 14.** Configuración del servidor Express para el uso de Socket.io

Se crea la clase "socket.js" y se inicializa el socket de conexión donde llamara funciones con eventos que se estarán escuchando y emitiendo según el caso, posteriormente, se debe exportar para su uso en las diferentes rutas de la API. Finalmente, Se crea un evento particular que no hace parte de "socket.io", pero si hace parte de una dependencia propia de Node llamada events, esta será utilizada para eventos que se necesitan al interior de la clase, como: "newVisitRecord" que se emite cuando la función "sendNewVisit" es ejecutada, (recibe como parámetro las visitas nuevas) por el controlador "clientController", quien a su vez fue ejecutado mediante la ruta de la API que usa el módulo QR, para enviar nuevas visitas al momento de ser validado un ingreso. Cuando esto ocurre se puede emitir el evento "getNewVisit", que se dispara por el socket y que envía las nuevas visitas al cliente, que por supuesto, estará escuchando dicho evento. De esta manera, es que se logra generar un registro de visitas en tiempo real, donde se visualiza en todo momento en el Frontend.

**Routes:** Contiene todas las rutas del servidor de Express, las rutas o direccionamiento [26], por lo que a cada ruta se le asignó un método que deriva del protocolo HTTP y que define el tipo de petición a resolver, se generaron nuevas rutas y se hicieron cambios en el tipo de método, según la petición y la acción que el servidor debe cumplir para resolver las peticiones que recibe. Las rutas se manejan de la siguiente manera: http://URL_BASE/END_POINT, donde la URL_BASE será la URL del servidor en donde se encuentre desplegado o en su defecto, la ruta local y el END_POINT es el punto final, la URL o ruta específica de la API, que responde a una petición que el servidor atiende.

## V. Procedimiento 4

En esta sección se realiza una prueba del funcionamiento completo de todos factores intervenidos durante el proceso, donde se podrá observar el resultado final del módulo de seguridad y donde se verifica el cumplimiento del protocolo en todas sus etapas.

A continuación, se encierra la interacción correspondiente (cliente-servidor), en recuadros DE SIMULACION REALES de colores para propósitos ilustrativos.

### A. Etapa 1 – Generar solicitud de ingreso:

En la Figura 15 y Figura 16 se observa el siguiente proceso, Inicialmente el cliente carga el formulario con los datos de tipo usuario, obtenidos desde servidor, (Rojo), el usuario completa sus datos personales. A medida que interactúa con los componentes fecha y zonas, el servidor realiza las validaciones para encontrar las zonas disponibles según la fecha seleccionada, (Amarillo, para objeto de prueba el tope de confirmados por zona es igual a 5). Luego, el usuario presiona Enviar y espera la validación (Azul), donde el servidor pronuncia los nuevos confirmados en las zonas y vemos que la solicitud ha sido exitosa.


**Figura 15.** Envió de solicitud de ingreso a las zonas

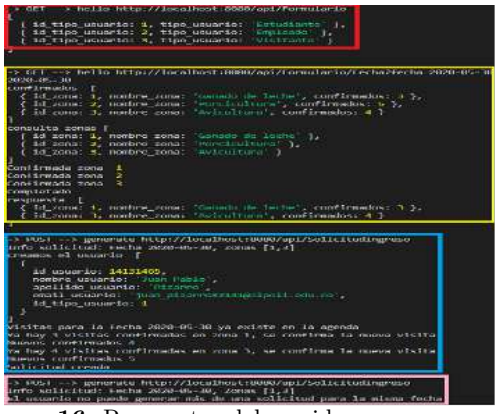

**Figura 16.** Respuestas del servidor para comprobar disponibilidad y generar solicitud

Si un usuario intenta solicitar una visita para la misma fecha, se lanza un mensaje de validación (Rosado), como se observa en las Figura 16 y Figura 17.

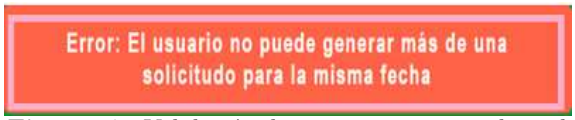


**Figura 17.** Validación de error en generar solicitud

## B. Etapa 2 - Gestión de solicitudes

### 1) Etapa 2.1 - Solicitudes Pendientes

El Administrador ingresa a la sección de solicitudes pendientes, donde inicialmente se renderiza la vista con los datos obtenidos desde el servidor, como se observa en la Figura 18 y Figura 19 (Azul).

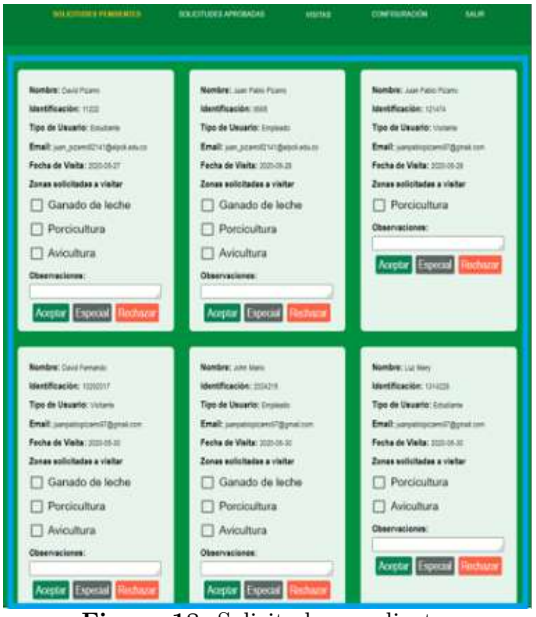

**Figura 18.** Solicitudes pendientes

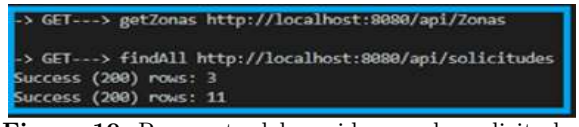


**Figura 19.** Respuesta del servidor con las solicitudes pendientes

**Aceptar solicitud:** El administrador selecciona las zonas que va a autorizar, redacta una observación y luego presiona "Aceptar". Podemos observar el comportamiento y resultado en las Figura 20 y Figura 21, donde las que no fueron seleccionadas, automáticamente quedan denegadas (Verde). Nuevamente se obtienen las solicitudes pendientes restantes (Azul).

El usuario recibe un correo electrónico con el código QR de autenticación y la información clara del uso del mismo, donde especifican las zonas permitidas y la fecha única de validez y demás detalles de la visita


**Figura 20.** Aceptar solicitud con permisos de ingreso a zonas específicas

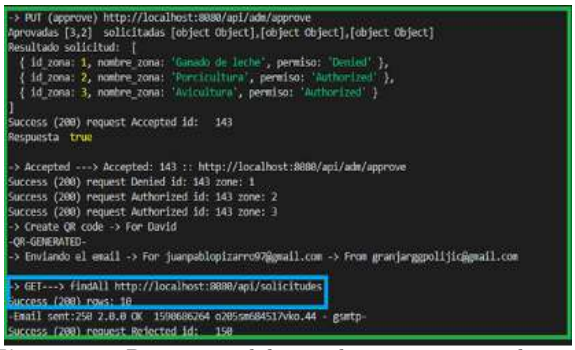


**Figura 21.** Respuesta del servidor en aceptar solicitud

(Verde), como se puede observar en las Figura 21 y Figura 22.


**Figura 22.** Email con código de autenticación de usuario

**Permiso especial:** El administrador selecciona las zonas que va a autorizar, redacta una observación y luego presiona "Especial". Podemos observar el comportamiento y resultado en las Figura 23 y Figura 24, donde todas las zonas seleccionadas fueron autorizadas con un permiso especial (Gris). Nuevamente se obtienen las solicitudes pendientes restantes (Azul).


**Figura 23.** Permiso especial de acceso en zonas específicas

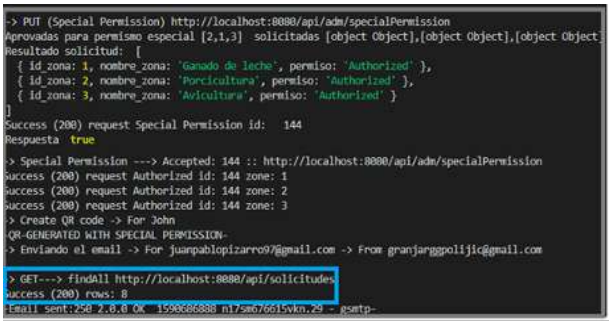


**Figura 24.** Respuesta del servidor para permiso especial

El usuario recibe un correo electrónico con el código QR de autenticación y la información clara del uso del mismo, donde especifican las zonas permitidas y que el tipo de permiso no tiene fecha de caducidad, hasta una nueva orden (Gris), como se puede observar en las Figura 24 y Figura 25.


**Figura 25.** Email con código de autenticación de usuario con permiso especial

Rechazar solicitud: El administrador deja una observación y presiona "Rechazar", automáticamente se deniegan los permisos en todas las zonas, se liberan los cupos donde se había confirmado (Rojo) y luego se obtienen las solicitudes pendientes restantes (Azul). El proceso se observa en las Figura 26 y Figura 27.


**Figura 26.** Rechazo de solicitud de ingreso


**Figura 27.** Respuesta del servidor rechazo de solicitud

El usuario recibe un correo electrónico con detalles del rechazo de la solicitud (Rojo), se puede observar la respuesta en las Figura 27 y Figura 28.

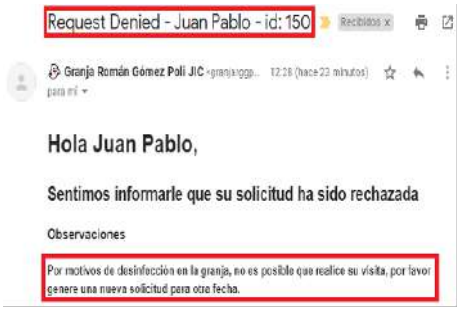


**Figura 28.** Email de rechazo de solicitud

## 2) Etapa 2.2 Solicitudes aprobadas

En esta sección el administrador puede observar todas las solicitudes aceptadas y con permiso especial, tiene la posibilidad de anular los permisos, según sea necesario (Por motivos de seguridad en la granja). Procede a dejar una observación y presiona "Cancelar" (Rojo). Nuevamente se obtienen las solicitudes aprobadas restantes (Azul). Este proceso se refleja en las Figura 29 y Figura 30.

El usuario recibe un correo electrónico con la información detallada, sobre la anulación de los permisos


**Figura 29.** Solicitudes aprobadas

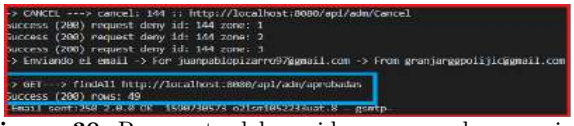


**Figura 30.** Respuesta del servidor para anular permisos en solicitudes aprobadas

que anteriormente teína (Rojo). En este caso un empleado. Se observa en las Figura 30 y Figura 31.


**Figura 31.** Email de Anular permisos

## C. Etapa 3 – Validación de acceso

El usuario se dirige a una zona de la granja y presenta el código QR de autenticación para ser validado, el módulo scanner decodifica el código, luego manda los datos al servidor especificando la zona de lectura y recibe una respuesta para autorizar o denegar el ingreso, esta prueba se realiza en 2 zonas, donde a cada una le corresponde una cámara, ambas conectadas a la misma Raspberry [27], pero ejecutando procesos independientes para la lectura. De igual forma, se establece un montaje básico de Leds, en donde se puede indicar una respuesta para el usuario, como se observa en las Figura 32 y Figura 33.

En la Figura 33. Se observa la respuesta del servidor cuando un usuario es autorizado en este caso con permiso especial, se registra la visita en la base de datos, con hora y fecha, se emite el evento para que el cliente web pueda escucharlo, además, retorna una respuesta de la petición que se hizo desde el cliente modulo scanner, para autorizar el ingreso en la zona, se observa un mensaje por consola y se enciende un led de color verde (Figura 32), que indica que el usuario es autorizado en la zona Porcicultura.

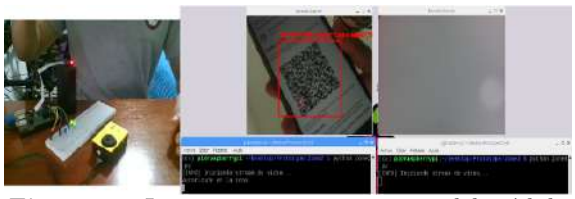

**Figura 32.** Lectura y respuesta exitosa del módulo scanner QR en la zona Porcicultura

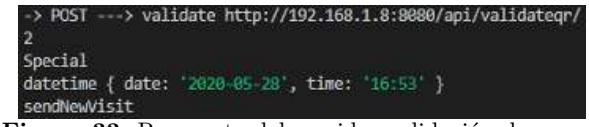


**Figura 33.** Respuesta del servidor validación de acceso exitoso y registro de visita en la zona Porcicultura

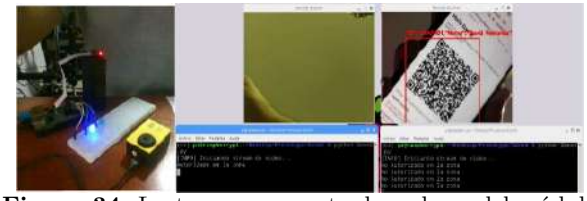

**Figura 34.** Lectura y respuesta de rechazo del módulo scanner QR en la zona Avicultura

En la Figura 35, se observa la respuesta del servidor cuando un usuario no es admitido y se le niega el acceso, retornando la respuesta al módulo scanner donde se le niega el ingreso, este proceso se visualiza por la consola y se enciende un led de color azul Figura 34, que indica que el usuario no está autorizado para la fecha en la zona de Avicultura.

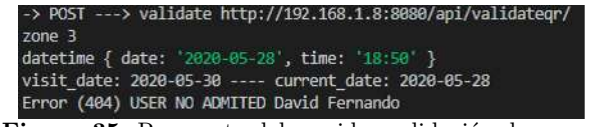


**Figura 35.** Respuesta del servidor validación de acceso de rechazo en zona Avicultura por fecha no valida

Ahora un usuario con código de autenticación únicamente valido en la zona Avicultura, intenta ingresar a la respectiva zona y fecha, donde la respuesta es exitosa como se muestra en las Figura 36 y Figura 37. Se puede ver el mensaje por consola y se enciende el led verde donde se le indica que el acceso es autorizado.

Luego el mismo usuario intenta ingresar a la zona Porcicultura presentando el código QR, en este caso como el usuario solo tiene autorización en Avicultura, se muestra un mensaje por consola e igualmente se enciende el led azul indicando una respuesta negativa

**Figura 36.** Lectura y respuesta exitosa de usuario que ingresa a la zona y fecha autorizada, Avicultura

```
zone 3
datetime { date: '2020-06-25', time: '20:17' }
visit_date: 2020-06-25 ---- current_date: 2020-06-25
Accepted
Success (200) USER ADMITED Mery Duque
```

**Figura 37.** Respuesta del servidor cuando un usuario con autorización es válido en la zona y fecha

donde el usuario no está autorizado para la zona, evitando que se violen los permisos y que los usuarios no ingresen a zonas en las que no tienen autorización. Los resultados se muestran en las Figura 38 y Figura 39.

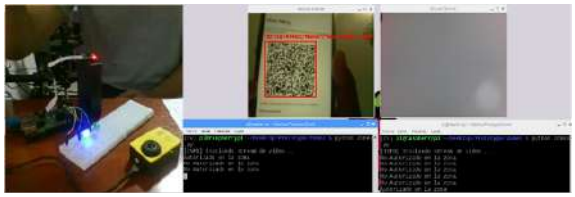

**Figura 38.** Lectura y respuesta negativa de ingreso a usuario en cuya zona no tiene autorización, Porcicultura

```
zone 2
datetime { date: '2020-06-25', time: '20:18' }
visit_date: 2020-06-25 ---- current_date: 2020-06-25
Error (404) USER NO ADMITED Mery Duque
```

**Figura 39.** Respuesta del servidor cuando un usuario intenta ingresar a una zona no autorizada

En las Figuras 40, 41, 42 y 43, se observa como posterior a la anulación de permisos (como se aprecia en la Figura 38), el usuario no pude ingresar de nuevo a las zonas, en este caso igualmente se valida en Porcicultura y Avicultura, donde ahora el mensaje por consola es negativo y así mismo se visualiza la respuesta en el montaje donde se enciende el led azul.

**Figura 40.** Lectura y respuesta negativa en zona Porcicultura de usuario con permisos anulados

### D. Etapa 4 – Control

Una vez que el administrador entra a la sección de visitas, puede seleccionar como filtrarlas (Rojo), se es-

```
zone 2
Special
Error (404) USER NO AUTHORIZED John Mario
```

**Figura 41.** Respuesta del servidor cuando un usuario intenta ingresar a la zona Porcicultura y los permisos fueron anulados

**Figura 42.** Lectura y respuesta negativa en zona Avicultura de usuario con permisos anulados

```
zone 3
Special
Error (404) USER NO AUTHORIZED John Mario
```

**Figura 43.** Respuesta del servidor cuando un usuario intenta ingresar a la zona Avicultura y los permisos fueron anulados

tablece un socket con el servidor (Amarillo), como se observa en las Figura 44 y Figura 45, si por algún motivo sale de la sección de visitas, se desconecta del socket.


**Figura 44.** Visualización de visitas en tiempo real

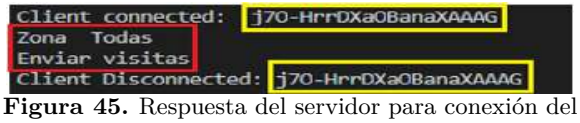


**Figura 45.** Respuesta del servidor para conexión del cliente y visualización de visitas

Cuando se escucha el evento que recibe las nuevas visitas, se renderiza el componente de la tabla, según la ruta o la información filtrada en ese momento, para una correcta actualización. En la Figura 46 se observa cómo se filtra por zona Porcicultura y se actualiza una nueva visita para esa misma zona.

La respuesta y emisión del evento se puede observar en la Figura 36 donde se realizó la acción desde la zona Porcicultura (id 2 en este caso). Al igual se puede filtrar por fechas anteriores, pero no se visualiza ningún cambio en el componente si ocurre una visita, para no alterar la vista donde el administrador realiza su


**Figura 46.** Visualización de visitas filtradas por zona en tiempo real

consulta. En la Figura 47 y Figura 48 se observa el resultado en el cliente y el servidor respectivamente.


**Figura 47.** Visualización de visitas filtradas por fecha y zona

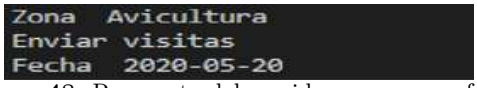


**Figura 48.** Respuesta del servidor con zona y fecha específica

Finalmente, se realiza la prueba en las dos zonas Avicultura y Porcicultura, con un usuario que tiene el acceso en ambas, en la Figura 49 se visualiza el registro en tiempo real.


**Figura 49.** Visualización de visitas en tiempo real para un usuario que ingresa en las zonas Avicultura y Porcicultura

## VII. Trabajos futuros

Se sugiere la implementación del prototipo en todas las sedes del PCJIC, por lo que se puede adecuar fácilmente y expandir el control de acceso en cualquier dependencia, añadiendo reportes, haciendo uso de la información disponible para la toma de decisiones importantes en la institución, incluso, dada la situación actual mundial por el covid-19, aportando en el correcto control de flujo de personas en un ambiente cerrado, como laboratorios, aulas, oficinas, etc. Y así, velar por la salud y bienestar de la comunidad académica y en general.

## VIII. Conclusiones

Los métodos y procesos empleados basados en la literatura dieron resultados positivos en el protocolo diseñado, pues cada etapa se pudo ejecutar con precisión para efectuar funciones específicas que aportaron sustancialmente una mejora a la seguridad del sistema intervenido. Se logró diseñar e implementar la estructura apropiada del módulo de seguridad, donde se puede concluir que gracias a la arquitectura cliente-servidor, se permite adicionar dicho módulo positivamente y mejorar las características de la aplicación, incluso para posibles mejoras a futuro de manera fácil sin perjudicar el funcionamiento. Se logra el control de acceso en cada una de las zonas, de acuerdo a la capacidad permitida, esto funciona como medida de protección para la salud de los animales y personas, puesto que hay un correcto flujo dentro de la granja, permitiendo el ingreso de personas adecuadas con autorización.

Para finalizar, se logra establecer una visualización de registro de visitas en tiempo real, que aporta a la seguridad y facilita el control en las instalaciones de la granja.

## IX. Agradecimientos

Este trabajo es soportado y adscrito: PROYECTOS DE INVESTIGACIÓN SEDE CENTRAL MICROCUANTIA SMC 7224 AÑO 2019 denominado “Empleo de las Tics para el monitoreo térmico avícola en la granja Román Gómez Gómez del municipio de marinilla PCJIC”, financiado por la Vicerrectoría de Docencia e Investigación del Politécnico colombiano Jaime Isaza Cadavid.

## Referencias

[1] M. I. Guzmán Méndez and A. M. Pineda Hernández and Z. Carbonell Muñoz and A. Ganem Pareja, A., “Elaboración de un protocolo institucional de seguridad del paciente en la facultad de Odontología de la Universidad de Cartagena,” Ph.D. dissertation, Universidad de Cartagena, 2017. http://repositorio.unicartagena.edu.co/bitstream/handle/11227/6335/Informe%20FINAL%2C%20FINAL.pdf?sequence=1&isAllowed=y.

[2] S. López and A. Bishara, A. and J. E. Muguerza Jaramillo, “Diseño e implementación de un sistema de control de acceso y monitoreo de sensores para data center de la empresa Quifatez. SA, utilizando hardware libre,” Ph.D dissertation, 2018. https://dspace.ups.edu.ec/handle/123456789/16451.

[3] C. D. P. H. de Pablos and J. J. L. H. Agius and S. M. R. Romero and S. M. Salgado, *Organización y transformación de los sistemas de información en la empresa.* Esic, 2018. https://editorial.tirant.com/es/libro/organizacion-y-transformacion-de-los-sistemas-de-informacion-en-la-empresa-9788473568142.

[4] C. D. C. de Bogotá, *Seguridad de la información como medida para la protección de los datos.* 2017. https://bibliotecadigital.ccb.org.co/handle/11520/19326.

[5] L. C. Marín Martínez and N. Campuzano and J. C. Ocampo, "Diseño de protocolo de ingreso a obra a través de listas de chequeo para el proyecto Terranova apartamentos," Ph.D. dissertation, Universidad Libre Seccional Pereira, 2017. http://repositorio.unilibrepereira.edu.co:8080/pereira/handle/123456789/747.

[6] S. Ríos Díaz and C. Garcés Agudelo and W. S. Puche, *Prototipo basado en la tecnología de códigos QR para limitar el acceso a las personas que visiten la granja Román Gómez Gómez del Politécnico ubicada en el sector de marinilla.* Medellín, Colombia: Politécnico Colombiano Jaime Isaza Cadavid, 2019.

[7] J. S.Martínez Villarreal, "Diseño de protocolos de seguridad y salud ocupacional en el Laboratorio de Petróleos de la Escuela Politécnica Nacional EPN en función de los requerimientos de la unidad institucional," M.S. thesis, Quito, Universidad de las Américas, 2018. http://dspace.udla.edu.ec/handle/33000/9475.

[8] C. Martínez-Troncoso, "Protocolo de gobierno y gestión de identidades digitales y de control de acceso en el contexto de una institución de educación superior," M.S. thesis, Universidad del Norte, 2018. http://manglar.uninorte.edu.co/bitstream/handle/10584/8326/133657.pdf?sequence=1.

[9] M. E. A. López y L. J. Rios, "Gestión del control de acceso con tecnología open source en proyectos de domótica," in XX Workshop de Investigadores en Ciencias de la Computación (WICC 2018, Universidad Nacional del Nordeste), 2018. http://sedici.unlp.edu.ar/handle/10915/68037.

[10] M. Gans, T. Hodges y G. Wilson, *JavaScript for Data Science.* CRC Press, 2020. https://js4ds.org/.

[11] Q. Luong, Web application development with Reactjs framework; Case: Web application for an association, 2019 [Online]. Available: https://www.theseus.fi/handle/10024/166801.

[12] A. F. Garay Flórez and J. D. Gallego Giraldo, *Diseño e implementación de un sistema para controlar el acceso a zonas restringidas en la clínica de la Universidad Cooperativa de Colombia utilizando internet de las cosas,* 2018, https://repository.ucc.edu.co/handle/20.500.12494/4193.

[13] E. D. Prado Cañal, "Sistema de gestión y control de acceso basado en IoT y Smartphones," Ph.D. thesis, Universitat Politècnica de Catalunya, 2017. https://upcommons.upc.edu/handle/2117/108916.

[14] F. D. Guano Quimbita and W. E. Sandoval Amagua, "Implementación de un sistema de acceso y seguridad para el Laboratorio de Tecnología Industrial," Ph.D. thesis, Quito, 2019. https://bibdigital.epn.edu.ec/handle/15000/20690.

[15] S. U. Jan y F. Qayum, "A Robust Authentication Scheme for Client-Server Architecture with Provable Security Analysis," *Network and Communication Technologies*, vol. 3, n.° 1, 2018. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3331985.

[16] G. Born, Kompendium HTML: con XHTML, DHTML, CSS, XML, XSL y WML. Markt + Tecnik. Berlin, 2001.

[17] M. Abernethy, ¿Simplemente qué es node.js? un ser-vidor listo para codificar. IBM developer Works. Cali-fornia, 2011 [Online]. Available: https://www.ibm.com/developerworks/ssa/opensource/library/os-nodejs/index.html.

[18] Sos para alumnos, Códigos QR, que son [Online]. Available: https://sites.google.com/site/sosparaalumnos/que-es/el-codigo-qr.

[19] Express-Infraestructura de aplicaciones web Node.js. Retrieved May 27, 2020 from: https://expressjs.com/es/.

[20] Marco general de la protección de los datos personales en Colombia. Numeral 11 del artículo 189 de la Constitución Política y la Ley 1581 de 2012 [Online]. Available: https://www.mintic.gov.co/portal/604/articles4274_documento.pdf.

[21] Information Capacity and Versions of the QR Code. Retrieved June 1, 2020 from: https://www.qrcode.com/en/about/version.html#versionPageMixed.

[22] Nodemailer. Retrieved April 15, 2020 from: https://nodemailer.com/about/.

[23] qrcode - npm. Retrieved April 7, 2020 from: https://www.npmjs.com/package/qrcode.

[24] Composites 1: An Exploration into Real-Time Animated Notation in the Web Browser. Retrieved June 2, 2020 from: https://hal.archives-ouvertes.fr/hal-02382500/document#page=389.

[25] Socket.io 2.0 is here. Retrieved June 2, 2020 from: https://socket.io/.

[26] Direccionamiento básico de Express [Online]. Available: https://expressjs.com/es/starter/basic-routing.html.

[27] G. I. B. G. Carlos Patricio Chavez Ñauñay, "Estudio comparativo de las tecnologías Python y

Perl para desarrollar aplicaciones web implementando al programa de alfabetización del Consejo Provincial de Chimborazo," Riobamba, 2011 [Online]. Available: http://dspace.espoch.edu.ec/handle/123456789/463.