

Universidad Autónoma Metropolitana

Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería
Licenciatura en Ingeniería en Computación

**Sistema de administración y configuración de
dispositivos en una red de sensores, mediante Bluetooth.**

Modalidad: Proyecto de investigación
Trimestre 2017 – Otoño

Erick Iván Medina Salas
2133035302
eims@azc.uam.mx

Asesor:

Dr. Leonardo Daniel Sánchez Martínez
Profesor Asociado
Departamento de Sistemas
ldsm@azc.uam.mx

Co-Asesor:

Dr. José Alejandro Reyes Ortiz
Profesor Asociado
Departamento de Sistemas
jaro@azc.uam.mx

15 de noviembre de 2017

Declaratoria

Yo, Leonardo Daniel Sánchez Martínez, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Leonardo Daniel Sánchez Martínez

Asesor

Yo, José Alejandro Reyes Ortiz, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Dr. José Alejandro Reyes Ortiz

Co-Asesor

Yo, Erick Iván Medina Salas, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Erick Iván Medina Salas

Alumno

Resumen

En este documento, se presenta el proceso de desarrollo de una aplicación Web para un Sistema de Información. Este sistema es capaz de administrar y configurar una red de sensores, en un espacio inteligente, de manera inalámbrica, utilizando la tecnología Bluetooth. El Sistema de Información está compuesto por los siguientes módulos:

- Módulo de alta, eliminación o modificación: encargado de administrar la Base de Datos del Sistema.
- Módulo de procesamiento de información: funciona como intermediario entre los demás módulos del sistema, así como de las interfaces gráficas. También se encarga de realizar las peticiones hacia los demás módulos.
- Módulo para la administración de la interfaz Bluetooth: administra los recursos necesarios para manejar la interfaz Bluetooth.
- Módulo para la detección de nodos: se encarga de detectar nuevos dispositivos que contengan una interfaz Bluetooth encendida.
- Módulo para el envío de información: se encarga de enviar información de configuración a dispositivos con interfaz Bluetooth.

Además, se describe también las bibliotecas, externas de Java, utilizadas para la creación del proyecto, como *Hibernate* y *Bluecove*. Para finalizar se describe los resultados y conclusiones obtenidos, como realización de este proyecto.

Tabla de contenido

| | |
|---|----|
| 1. Introducción..... | 1 |
| 2. Antecedentes | 1 |
| 3. Justificación..... | 3 |
| 4. Objetivos..... | 3 |
| 4.1 Objetivo general..... | 3 |
| 4.2 Objetivos particulares | 3 |
| 5. Marco Teórico | 3 |
| 5.1 Interfaz Bluetooth en Java | 3 |
| 5.1.1 Bluecove | 4 |
| 5.1.2 Paquete javax.bluetooth..... | 4 |
| 5.1.3 Paquete javax.obex | 5 |
| 5.2 Mapeo Objeto Relacional (ORM) | 6 |
| 5.2.1 Hibernate | 6 |
| 6. Desarrollo del proyecto | 6 |
| 6.1 Módulo de alta, eliminación y modificación de datos. | 7 |
| 6.1.1 Estructura de la base de datos | 7 |
| 6.1.2 Clases Bean | 8 |
| 6.1.3 Archivos de configuración y mapeo de Hibernate | 11 |
| 6.1.4 Clase CrearConexion de Hibernate | 12 |
| 6.1.5 Clases Operaciones | 12 |
| 6.1.6 Administrador de la Base de Datos..... | 13 |
| 6.2 Módulo para administración de la interfaz Bluetooth..... | 14 |
| 6.3 Módulo para la detección de nodos. | 14 |
| 6.4 Módulo para el envío de información por Bluetooth | 15 |
| 6.5 Módulo para el procesamiento de información. | 15 |
| 6.5.1 Peticiones de Alta | 16 |
| 6.5.2 Peticiones de Modificación o Eliminación..... | 18 |
| 6.5.3 Peticiones a los módulos Bluetooth | 22 |
| 6.6 Interfaces | 23 |
| 7. Resultados..... | 28 |
| 8. Conclusiones y trabajo a futuro | 48 |
| 9. Referencias Bibliográficas..... | 49 |
| 10. Apéndices..... | 49 |

Tabla de Figuras

| | |
|---|----|
| Figura 1. Diagrama Entidad-Relación del SI..... | 7 |
| Figura 2. Clase DispositivoAmbiental..... | 8 |
| Figura 3. Clase DispositivoComunicacion..... | 8 |
| Figura 4. Clase DispositivoElectronico..... | 9 |
| Figura 5. Clase Espacio..... | 9 |
| Figura 6. Clase NodoConcentrador..... | 9 |
| Figura 7. Clase Nodoldentificador..... | 10 |
| Figura 8. Clase NodoSensor..... | 10 |
| Figura 9. Clase Red..... | 10 |
| Figura 10. Clase Usuario..... | 11 |
| Figura 11. Clase CrearConexion..... | 12 |
| Figura 12. Clase AdministradorBD..... | 13 |
| Figura 13. Clase AdministradorBluetooth..... | 14 |
| Figura 14. Clase DetectorNodos..... | 14 |
| Figura 15. Clase EnviInformacion..... | 15 |
| Figura 16. Clase Procesador..... | 15 |
| Figura 17. Formulario para dar de alta un espacio..... | 16 |
| Figura 18. Diagrama de secuencia DarAltaNodo..... | 17 |
| Figura 19. ComboBox tipo de dispositivo..... | 18 |
| Figura 20. Formulario para buscar un espacio..... | 18 |
| Figura 21. ComboBox para modificar un espacio..... | 19 |
| Figura 22. Opciones en modificar espacio..... | 19 |
| Figura 23. Formulario para modificar un espacio..... | 19 |
| Figura 24. Diagrama de secuencia ModificarNodo..... | 20 |
| Figura 25. Diagrama de secuencia ModificarRed..... | 21 |
| Figura 26. Opciones Bluetooth..... | 22 |
| Figura 27. Opción Dispositivo Electrónico..... | 23 |
| Figura 28. Opción Dispositivo de Comunicación..... | 23 |
| Figura 29. Caja de texto para dirección MAC..... | 23 |
| Figura 30. Caja de texto para Voltaje..... | 24 |
| Figura 31. Caja de texto para Primer Nombre del usuario..... | 24 |
| Figura 32. Caja de búsqueda por Espacio..... | 24 |
| Figura 33. Botón "Guardar" deshabilitado..... | 24 |
| Figura 34. Opción Alta Nodo Concentrador..... | 25 |
| Figura 35. Opción Alta Nodo Identificador..... | 25 |
| Figura 36. Opción Alta Nodo Sensor..... | 26 |
| Figura 37. Dispositivos Wifi no asociados..... | 26 |
| Figura 38. Dispositivos Bluetooth no asociados..... | 26 |
| Figura 39. Mensaje de alerta Dispositivos Wifi..... | 27 |
| Figura 40. Información dispositivo Bluetooth ya registrado..... | 27 |
| Figura 41. Dispositivo Bluetooth no registrado..... | 27 |
| Figura 42. Interfaz de Inicio..... | 28 |
| Figura 43. Opciones Alta..... | 28 |
| Figura 44. Alta Dispositivo Ambiental..... | 29 |
| Figura 45. Alta Dispositivo Electrónico/Comunicación..... | 29 |
| Figura 46. Alta Espacio..... | 30 |

| | |
|--|----|
| Figura 47. Alta Nodos..... | 30 |
| Figura 48. Alta Redes..... | 31 |
| Figura 49. Alta Usuarios | 31 |
| Figura 50. Mensaje "Datos Almacenados Correctamente" | 32 |
| Figura 51. Mensaje "Espacio repetido" | 32 |
| Figura 52. Opciones Modificación..... | 33 |
| Figura 53. Búsqueda de Dispositivos Ambientales para modificar..... | 33 |
| Figura 54. Resultado de búsqueda Dispositivo Ambiental..... | 34 |
| Figura 55. Interfaz para modificar un dispositivo ambiental..... | 34 |
| Figura 56. Búsqueda de Dispositivos Electrónicos/Comunicación para modificar..... | 35 |
| Figura 57. Resultado de búsqueda Dispositivo Electrónico/Comunicación | 35 |
| Figura 58. Interfaz para modificar un Dispositivo Electrónico..... | 36 |
| Figura 59. Búsqueda de Espacios para modificar | 36 |
| Figura 60. Resultado de búsqueda Espacio | 37 |
| Figura 61. Interfaz para modificar un Espacio | 37 |
| Figura 62. Búsqueda de Nodos para modificar..... | 38 |
| Figura 63. Resultado de búsqueda Nodo | 38 |
| Figura 64. Interfaz para modificar un Nodo | 39 |
| Figura 65. Mensaje "Datos actualizados correctamente" | 39 |
| Figura 66. Mensaje "Datos actualizados correctamente, información de configuración enviada" | 40 |
| Figura 67. Mensaje "Datos actualizados correctamente, error al enviar información de configuración" | 40 |
| Figura 68. Búsqueda de Redes para modificar | 41 |
| Figura 69. Resultado de búsqueda Red..... | 41 |
| Figura 70. Interfaz para modificar una Red | 42 |
| Figura 71. Búsqueda de Usuarios para modificar | 42 |
| Figura 72. Resultado de búsqueda Usuario..... | 43 |
| Figura 73. Interfaz para modificar un Usuario..... | 43 |
| Figura 74. Mensaje "Usuario eliminado correctamente"..... | 44 |
| Figura 75. Opciones Bluetooth | 44 |
| Figura 76. Búsqueda de dispositivos Bluetooth cercanos, no asociados | 45 |
| Figura 77. Interfaz para agregar un nuevo nodo, por medio de la detección..... | 45 |
| Figura 78. Mensaje "Datos almacenados correctamente, información de configuración no pudo ser enviada" | 46 |
| Figura 79. Mensaje "¡No se encontraron nodos disponibles!"..... | 46 |
| Figura 80. Búsqueda de dispositivos Bluetooth cercanos, sin configurar | 47 |
| Figura 81. Mensaje "Información de configuración enviada" | 47 |

1. Introducción

Los sistemas de información (SI) permiten almacenar y procesar información, a través de la entrada y salida de datos. Estos sistemas ayudan a tener una mejor organización de la información siendo auxiliares para el manejo y administración de empresas u organizaciones en la actualidad. Los SI utilizan diversos componentes para la persistencia de datos, tales como archivos o bases de datos.

Una Base de Datos (BD) es una estructura de datos que nos permite almacenar una gran cantidad de datos de manera persistente, los datos se encuentran organizados en tablas y estas a su vez en registros. Las tablas se encuentran relacionadas entre sí a través de índices, posibilitando una rápida recuperación de información.

De la misma forma, los SI pueden utilizar diferentes tecnologías para comunicarse con otros sistemas o dispositivos, por ejemplo, Bluetooth y Wifi. Si bien, el propósito de ambas tecnologías es el mismo, estas tienen características que las hacen distintas y por ende más o menos atractivas para ciertas aplicaciones. La principal diferencia entre Bluetooth y Wifi es el tipo de comunicación, ya que el primero siempre es punto a punto, mientras que último puede operar punto a punto o multipunto.

Hoy en día, la mayoría de los dispositivos electrónicos existentes, como celulares (*Smartphone*) y computadoras, integran ambas tecnologías. Sin embargo, la mayoría de los dispositivos móviles mínimos, como relojes y pulseras inteligentes, integran únicamente Bluetooth.

Así, en este proyecto se desarrolla un SI para administrar la información de los dispositivos asociados a diversos espacios físicos, así como configurar los mismos mediante Bluetooth. En particular, se considera que un espacio tiene asociado una red de sensores que está integrada por: un nodo identificador (NI), uno o más nodos sensores (NS) y un nodo concentrador (NC). En este sentido, el SI administrará la información de los nodos asociados a un espacio en particular y enviará información a cada dispositivo para su configuración al momento del despliegue de la red en el mismo. Información como el nombre y la contraseña de la red son datos necesarios para integrarse a esta y compartir información en la misma. El SI utilizará una BD para almacenar información de los dispositivos configurados, así como de las personas y espacios asociados a estos, permitiendo obtener información de manera rápida y precisa.

2. Antecedentes

Monitoreo de temperatura y humedad en un campo de cultivo utilizando el protocolo bluetooth [1]

En el trabajo presentado en el Congreso de Instrumentación, se diseñó un sistema de monitoreo de temperatura y humedad en un campo de cultivo usando como medio de transmisión el protocolo Bluetooth, utilizando microcontroladores y sensores. Es similar al proyecto debido a que se utilizarán sensores establecidos en un espacio, y a diferencia, en el proyecto de monitoreo utilizan otro dispositivo con interfaz Bluetooth para

recolectar información de los sensores, mientras que, en este proyecto se utilizará la tecnología Bluetooth para configurar cada dispositivo asociado a un espacio.

Monitoreo y control por Bluetooth [2]

En este proyecto de integración se trabajó con un sensor de temperatura y un microcontrolador, utilizando la tecnología Bluetooth se establecía una conexión punto a punto enviando información de la señal producida por el sensor hacia una computadora donde la señal es monitoreada. Es semejante a este proyecto debido a que también se trabajará con sensores vía Bluetooth, y a diferencia, no solo permitirá establecer una conexión sino también su configuración de cada dispositivo de manera automática.

Configuración e implementación de un sistema de control aplicando la tecnología bluetooth con PLC. Caso práctico: Laboratorio de Automatización (EIS). [3]

En el proyecto de grado de ingeniería, se equipó, en un laboratorio de automatización industrial, circuitos lógicos programables (PLC) para controlar y monitorear la producción dentro del laboratorio utilizando la tecnología Bluetooth. Es similar a este proyecto debido a que se establecerán diferentes sensores para monitorear un espacio, y a diferencia, se utilizará el Bluetooth para su configuración y WiFi para su comunicación.

Diseño e implementación de una aplicación para teléfonos celulares que permita la configuración de equipos cisco utilizando un enlace Bluetooth [4]

En el proyecto de grado de ingeniería, se diseñó e implementó una aplicación para teléfonos celulares permitiendo la configuración de equipos cisco vía inalámbrica Bluetooth, ofreciendo así una alternativa para configurar este tipo de equipos. Es semejante a este proyecto debido a que también se configurará por medio de Bluetooth dispositivos como sensores dando una alternativa para su configuración, y a diferencia, será un sistema donde la información que se obtendrá de cada nodo configurado será guardada en una BD para su posterior uso.

Sistemas Inteligentes de Transporte [5]

En el proyecto la compañía Abertis autopistas trabaja con la tecnología Bluetooth, utiliza diferentes sensores para la detección de dispositivos Bluetooth en autopista permitiendo elevar el nivel de información disponible para la toma de decisiones en cuestión de seguridad y en los clientes. Es semejante a este proyecto debido a que también se utilizará Bluetooth, y a diferencia, los dispositivos detectados se podrán configurar automáticamente, así como administrar.

Desarrollo e implementación de una red inalámbrica de sensores de temperatura y humedad [6]

En el proyecto de grado de ingeniería, se desarrolló e implemento una red inalámbrica de sensores (WSN) de temperatura y humedad. Es semejante a este proyecto debido a que se manejarán sensores y otros dispositivos vía inalámbrica a través de Wifi, mientras que la interfaz Bluetooth servirá para configurará automáticamente los dispositivos de manera que estén conectados a una misma red. De esta manera se podrá administrar la información de cada nodo configurado de manera automática.

3. Justificación

La administración de información siempre ha sido y será una necesidad primordial en el diseño de sistemas. En este caso en particular, el área de investigación en sistemas de información inteligentes (AISII) del Departamento de Sistemas de la Universidad Autónoma Metropolitana - Azcapotzalco, en el marco del Proyecto de Investigación titulado "Arquitectura Orientada a Servicios para Aplicaciones Sensibles al Contexto Utilizando Ontologías", iniciado el 6 de septiembre de 2016 y con número de acuerdo de aprobación por el Consejo Divisional 577.3.6.4.1, tiene diversos espacios de los que desea obtener información. Para ello, cada espacio tiene una red de sensores asociada, que a su vez está conformada por un conjunto particular de nodos. Cada nodo está integrado por un conjunto de componentes electrónicos específicos, que fueron adquiridos por la Universidad Autónoma Metropolitana - Azcapotzalco. El buen uso, manejo y seguimiento de estos componentes es responsabilidad del AISII. La cantidad de información generada por la asociación de una red a un espacio es considerable, y más aún si se considera que el número de espacios puede aumentar. Por otro lado, el despliegue de una red de sensores no es una tarea sencilla, ya que se deben considerar las características de los nodos para dicha actividad. Además, se debe considerar que una vez que se ha realizado el despliegue, la mayoría de las veces resulta inconveniente su modificación o alteración.

Así, resulta evidente crear un SI que permita administrar la información de las redes de sensores asociadas a varios espacios, así como la configuración a distancia de sus componentes.

4. Objetivos

4.1 Objetivo general

- Diseñar e implementar un sistema de información que permita administrar y configurar la información de los dispositivos de una red de sensores, a través de Bluetooth.

4.2 Objetivos particulares

- Diseñar e implementar el modelo de datos para el soporte del sistema.
- Diseñar e implementar módulo de alta, eliminación y modificación de datos.
- Diseñar e implementar un módulo para el procesamiento de información.
- Diseñar e implementar un módulo para la detección de nodos.
- Diseñar e implementar un módulo para el envío de información por Bluetooth

5. Marco Teórico

5.1 Interfaz Bluetooth en Java

El Bluetooth es una tecnología que se utiliza para la comunicación inalámbrica de dos o más dispositivos. Se utiliza principalmente en dispositivos móviles, como, por ejemplo: laptops, audífonos o manos libres, pulseras, relojes y celulares inteligentes, entre otros.

Esta tecnología, trabaja como un protocolo de comunicación, donde existe un cliente y un servidor. Quien inicia la conexión (el cliente) es el maestro, y quien recibe la conexión (el servidor) es el esclavo.

Debido a la facilidad de utilizar esta tecnología, cada vez es más común empezar a programar aplicaciones en dispositivos móviles con soporte para Bluetooth.

5.1.1 Bluecove

Bluecove es una biblioteca, en Java, que permite manejar la interfaz Bluetooth. Proporciona la interfaz Java JSR-82¹ para seguir los perfiles de Bluetooth:

- *SDP*² – Perfil de aplicación de detección de servicio
- *RFCOMM*³ – Protocolo de emulación de cable en serie
- *L2CAP*⁴ – Protocolo de adaptación y control de enlace lógico
- *OBEX*⁵ – Perfil de intercambio genérico de objetos (*GOEP*⁶) sobre *RFCOMM* y *TCP*⁷.

JSR-82 proporciona dos paquetes: *javax.bluetooth* y *javax.obex*. El primero permite realizar búsquedas de dispositivos, ya sea que estén encendidos y en un radio alcanzable por la interfaz Bluetooth o emparejados con el dispositivo Local; además, permite la búsqueda de servicios que ofrecen dispositivos cercanos y encendidos; y también, permite la comunicación mediante flujos de datos. El segundo permite realizar una conexión a través del protocolo OBEX, para la comunicación entre dos dispositivos.

5.1.2 Paquete javax.bluetooth

Para inicializar una conexión, es necesario tener un cliente Bluetooth. Un cliente Bluetooth necesita realizar las siguientes operaciones para lograr la comunicación:

- Buscar dispositivos.
- Buscar servicios de los dispositivos encontrados.
- Intentar establecer una comunicación, a través de un protocolo.
- Enviar y recibir mensajes.

El paquete contiene la clase *LocalDevice*, esta clase define a la interfaz Bluetooth del dispositivo local. Se obtiene un objeto de esta clase a través del método *LocalDevice.getLocalDevice()*, permitiendo obtener información de él, como, por ejemplo: dirección MAC Bluetooth, nombre del dispositivo y modo de conectividad.

Para iniciar el proceso de búsqueda, se tiene a la clase *DiscoveryAgent*. Esta clase, proporciona métodos para realizar descubrimientos de dispositivos y servicios. Un dispositivo local debe tener solo un objeto *DiscoveryAgent*. Este objeto debe ser recuperado mediante el método *getDiscoveryAgent()* del objeto *LocalDevice*.

¹ JSR-82 es un estándar definido por El Proceso de la Comunidad Java para proporcionar un estándar para desarrollar aplicaciones Bluetooth en Java.

² SDP es la abreviatura del término inglés “*Session Description Protocol*”

³ RFCOMM es la abreviatura del término inglés “*Radio Frequency Communication*”

⁴ L2CAP es la abreviatura del término inglés “*Logical Link Control and Adaptation Protocol*”

⁵ OBEX es la abreviatura del término inglés “*Object Exchange*”

⁶ GOEP es la abreviatura del término inglés “*Generic Object Exchange Profile*”

⁷ TCP es la abreviatura del término inglés “*Transmission Control Protocol*”

Hay dos formas de descubrir dispositivos: primero, se puede usar *startInquiry()* para iniciar una búsqueda de dispositivos cercanos del dispositivo local. Los dispositivos detectados, por *DiscoveryAgent*, se devuelven mediante el método *deviceDiscovered()* de la interfaz *DiscoveryListener*.

La interfaz *DiscoveryListener* permite descubrir dispositivos y servicios. Esta interfaz proporciona cuatro métodos:

- *DiscoveryListener.deviceDiscovered*
- *DiscoveryListener.inquiryCompleted*
- *DiscoveryListener.servicesDiscovered*
- *DiscoveryListener.serviceSearchCompleted*

Los dos primeros se utilizan para descubrir dispositivos y los dos últimos se utilizan para descubrir servicios. El primer método se invoca cuando un nuevo dispositivo es descubierto. El segundo método se invoca cuando se ha terminado la búsqueda de dispositivos, El tercer método se invoca cuando se ha descubierto uno o más servicios. El último método se invoca cuando se termina la búsqueda de servicios.

La segunda forma de descubrir dispositivos es a través del método *retrieveDevices()*. Este método devolverá los dispositivos que se han descubierto a través de una consulta previa o devolverá aquellos dispositivos que se tratan como preconocidos. Los dispositivos conocidos son aquellos que se encuentran emparejados con el dispositivo local.

El método *retrieveDevices()* no realiza alguna consulta, pero proporciona una forma rápida de obtener una lista de los dispositivos cercanos. Devuelve una lista de objetos *RemoteDevice*. La clase *RemoteDevice* define un dispositivo remoto, con él que se puede obtener su nombre a través del método *getFriendlyName()* y su dirección Bluetooth a través del método *getBluetoothAddress()*.

La clase *DiscoveryAgent* también encapsula la funcionalidad proporcionada por el perfil de aplicación de descubrimiento de servicio. La clase proporciona una interfaz para que una aplicación busque y recupere atributos para un servicio en particular. Hay dos formas de buscar servicios. Para buscar un servicio en un solo dispositivo, se debe usar el método *searchServices()*. Por otro lado, no importa en qué dispositivo está el servicio, el método *selectService()* realiza una búsqueda de servicio en un conjunto de dispositivos remotos. Cada servicio es identificado numéricamente por su *UUID (Universal Unique Identifier)* y cada servicio contendrá sus atributos propios.

5.1.3 Paquete *javax.obex*

El protocolo *OBEX* se trata de un protocolo de comunicación, parecido a *HTTP*, debido a que se envían y reciben mensajes que contiene cabeceras y puede o no contener un cuerpo de mensaje. También se manejan códigos de respuesta.

Los métodos que existen en *OBEX* son:

- *CONNECT*. Para iniciar una sesión en el servidor.
- *PUT*. Para enviar un archivo al servidor.
- *GET*. Para solicitar un archivo al servidor.
- *DELETE*. Para solicitar la eliminación de un archivo del servidor.

- *SETPATH*. Para solicitar el cambio de directorio actual dentro del sistema de archivos *del* servidor.
- *DISCONNECT*. Para finalizar la sesión en el servidor.

Las cabeceras son definidas en un objeto de tipo *HeaderSet*. Las más comunes son *count*, *name* y *length*. Existe también la clase *Operation* que permite enviar mensajes, a través de un flujo de salida (*DataOutputStream*); y recibir mensajes, a través de un flujo de entrada (*DataInputStream*), de cualquier tipo, con o sin cuerpo.

5.2 Mapeo Objeto Relacional (ORM⁸)

El ORM consiste en vincular o mapear entidades de una base de datos relacional con clases de un lenguaje de programación orientado a objetos. Es decir, vincular registros de una BD con objetos del lenguaje.

Muchas de las aplicaciones utilizadas hoy en día, están basadas en un lenguaje de programación orientado a objetos y utilizan además una base de datos relacional, dando la posibilidad de utilizar ORM. Existen diferentes *frameworks* comerciales o de uso libre, como *Hibernate*, que nos ayudan a utilizar esta técnica de programación.

Ventajas de utilizar ORM:

- Se reduce el tiempo de desarrollo: muchas herramientas ORM utilizan la ingeniería inversa para crear archivos de configuración, así como las clases necesarias, para hacer un mapeo a partir de una base de datos relacional.
- Se hace una abstracción en cuanto a consultas SQL⁹: debido a que se trabajan con objetos y no con registros, permitiendo adaptarse a cualquier manejador de base de datos.
- Se tienen métodos y clases ya definidos en muchas herramientas ORM, permitiendo extraer, insertar, modificar o eliminar objetos de manera más fácil.

5.2.1 Hibernate

Hibernate se encarga de asignar clases de Java a las tablas de la base de datos, y de asignar los tipos de datos de Java a los tipos de datos de SQL. Además, proporciona servicios de búsqueda y recuperación de datos. Su principal objetivo es reducir significativamente el tiempo de desarrollo.

6. Desarrollo del proyecto

Los módulos con los que cuenta el sistema son:

- Módulo de alta, eliminación y modificación de datos.
- Módulo para el procesamiento de información.
- Módulo para administración de interfaz Bluetooth
- Módulo para la detección de nodos.
- Módulo para el envío de información por Bluetooth

⁸ ORM es la abreviatura del término en inglés "*Object-Relational mapping*".

⁹ SQL es la abreviatura del término en inglés "*Structured Query Language*".

El módulo de administración de interfaz Bluetooth, únicamente, tiene comunicación directa con los módulos: para la detección de nodos y para el envío de información por Bluetooth. Los módulos: de alta, eliminación y modificación de datos, detección de nodos y envío de información, únicamente, tienen comunicación directa con el módulo de procesamiento de información.

6.1 Módulo de alta, eliminación y modificación de datos.

Este módulo se encarga de administrar la base de datos (escritura y lectura) además de permitir al usuario poder dar de alta, eliminar o modificar espacios, así como de las personas o nodos asociados a estos. Las acciones de este módulo dependerán completamente de las instrucciones del módulo de procesamiento, lo que implica que existe una comunicación bidireccional entre estos.

6.1.1 Estructura de la base de datos

En la Figura 1 se presenta el modelo de datos para el soporte del SI.

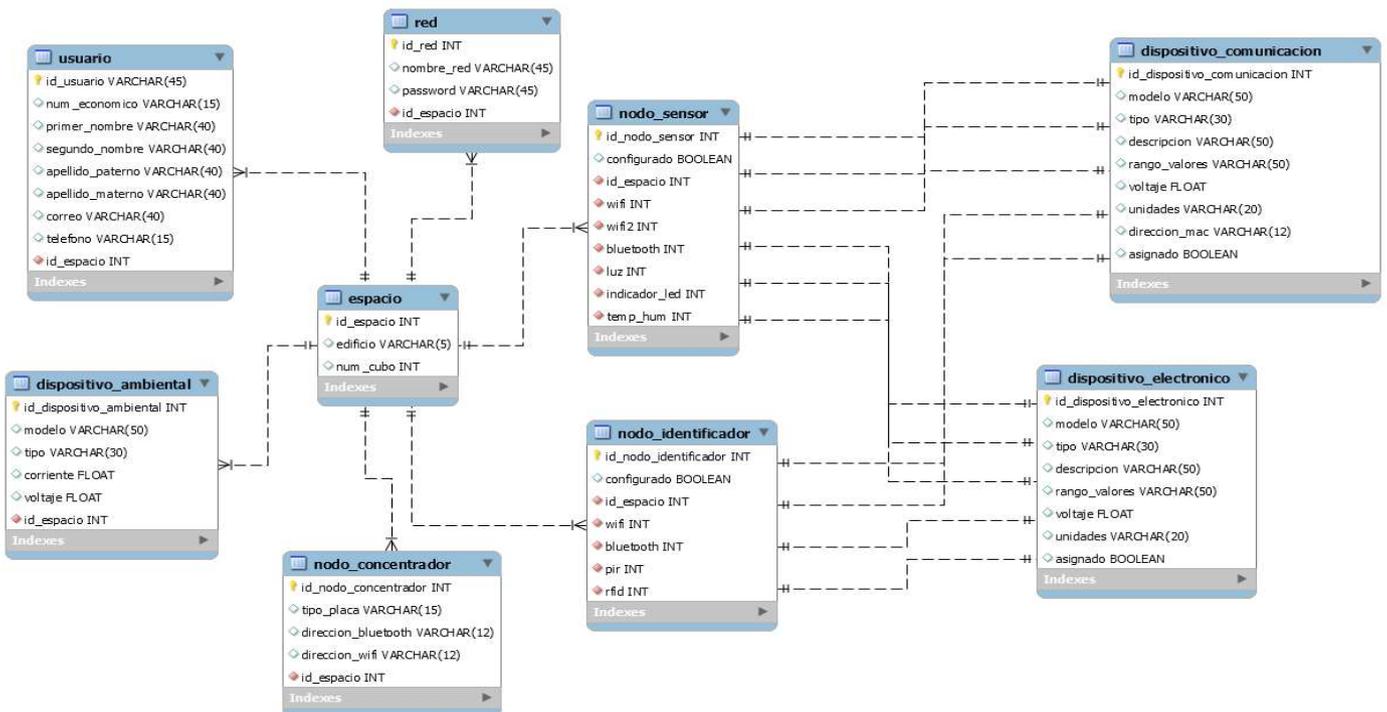


Figura 1. Diagrama Entidad-Relación del SI

6.1.2 Clases Bean

Una vez obtenido el Modelo E-R, es necesario crear las clases *Bean*, las cuales serán necesarias para poder realizar el mapeo objeto-relacional hacia la base de datos del sistema.

Se implementan los siguientes *Bean*:

DispositivoAmbiental: esta clase, define objetos de tipo Dispositivo Ambiental tal y como se muestra en la Figura 2.

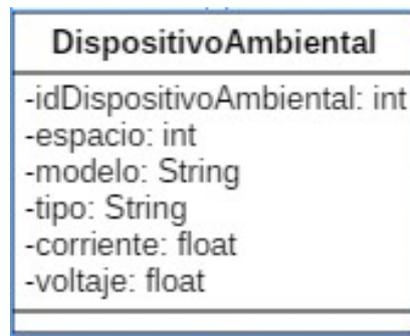


Figura 2. Clase DispositivoAmbiental

DispositivoComunicacion: esta clase, define objetos de tipo Dispositivo de Comunicación tal y como se muestra en la Figura 3.



Figura 3. Clase DispositivoComunicacion

DispositivoElectronico: esta clase, define objetos de tipo Dispositivo Electrónico tal y como se muestra en la Figura 4.



Figura 4. Clase DispositivoElectronico

Espacio: esta clase, define objetos de tipo Espacio tal y como se muestra en la Figura 5.

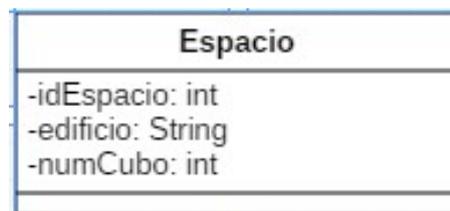


Figura 5. Clase Espacio

NodoConcentrador: esta clase, define objetos de tipo Nodo Concentrador tal y como se muestra en la Figura 6.

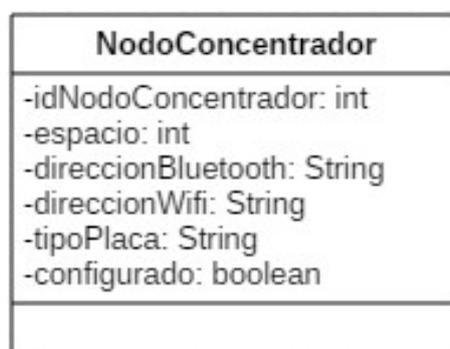


Figura 6. Clase NodoConcentrador

NodoIdentificador: esta clase, define objetos de tipo Nodo Identificador tal y como se muestra en la Figura 7.

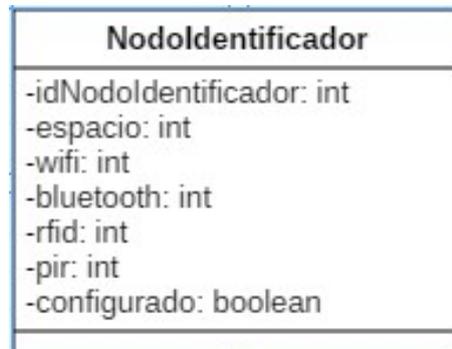


Figura 7. Clase NodoIdentificador

NodoSensor: esta clase, define objetos de tipo Nodo Sensor tal y como se muestra en la Figura 8.

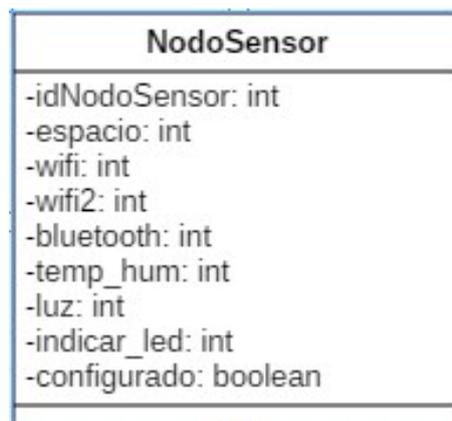


Figura 8. Clase NodoSensor

Red: esta clase, define objetos de tipo Red tal y como se muestra en la Figura 9.

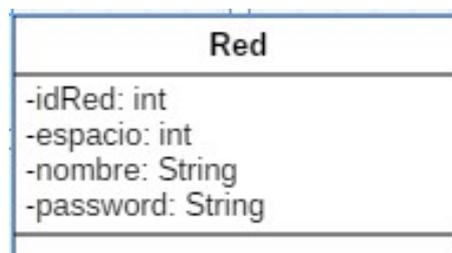


Figura 9. Clase Red

Usuario: esta clase, define objetos de tipo Usuario tal y como se muestra en la Figura 10.

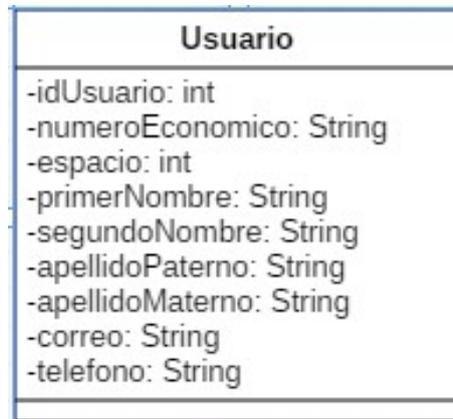


Figura 10. Clase Usuario

6.1.3 Archivos de configuración y mapeo de Hibernate

Las clases *Bean* son semejantes a las entidades de la BD, lo cual nos permite realizar el mapeo de una forma más simple. La biblioteca *Hibernate* necesita de archivos de configuración (XML), así como de mapeo, para poder realizar la conexión a la base de datos y realizar el mapeo objeto-relacional.

Se implementa el archivo de configuración principal (hibernate.cfg.xml): este archivo contiene los datos de acceso a la base de datos. Además, se indica la ubicación de los archivos de mapeo (XML).

Archivos de mapeo

Estos archivos contienen los datos necesarios para relacionar (mapear) una clase *Bean* con una entidad de la BD, se hace referencia a cada atributo de cada clase y a cada atributo de la entidad, indicando el tipo de dato entre ellos.

Se implementan los siguientes archivos de mapeo:

- DispositivoAmbiental.hbm.xml
- DispositivoComunicacion.hbm.xml
- DispositivoElectronico.hbm.xml
- Espacio.hbm.xml
- NodoConcentrador.hbm.xml
- NodoIdentificador.hbm.xml
- NodoSensor.hbm.xml
- Red.hbm.xml
- Usuario.hbm.xml

6.1.4 Clase CrearConexion de Hibernate

Una vez realizado los archivos de configuración y mapeo, así como de las clases *Bean*, se realiza una clase la cual establecerá una conexión con la base de datos. En la Figura 11, se muestra la clase CrearConexion.

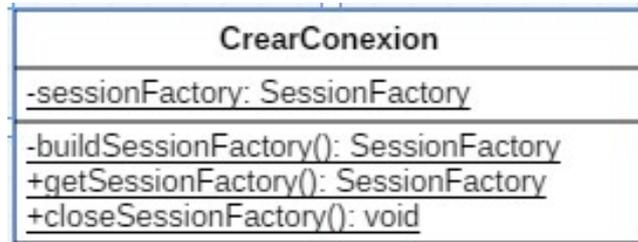


Figura 11. Clase CrearConexion

6.1.5 Clases Operaciones

Cada clase *Bean* estará asociada a una clase operación, la cual, será la encargada de insertar, eliminar, modificar u obtener registros de su misma entidad. Estas clases instanciarán a la clase CrearConexion por lo que abrirán la conexión hacia la BD, se obtendrá un objeto de tipo *Session*.

```
Session session = null;
try{
    session = CrearConexion.getSessionFactory().openSession();
}catch(HibernateException ex){
    System.err.println("No se pudo abrir la conexion");
    throw new ExceptionInInitializerError(ex);
}
```

A partir de aquí se pueden realizar diferentes operaciones con el objeto *session*, como, por ejemplo: guardar, eliminar o actualizar un espacio.

```
session.save(espacio); //guardar
session.delete(espacio); //eliminar
session.update(espacio); //actualizar
```

De la misma, el objeto *session* puede ejecutar sentencias tipo SQL

```
session.createQuery("SELECT e FROM Espacio e WHERE (e.numCubo IS NOT NULL AND e.edificio IS NOT NULL)").list();
```

Al momento de crear una sentencia SQL se utilizan las clases *Bean*, mapeadas, más no las entidades de la base de datos.

De esta forma se tienen las siguientes clases:

- OperacionesDispositivoAmbienta
- OperacionesDispositivoComunicacion
- OperacionesDispositivoElectronico
- OperacionesEspacio
- OperacionesNodoConcentrador
- OperacionesNodoIdentificador
- OperacionesNodoSensor
- OperacionesRed
- OperacionesUsuario

6.1.6 Administrador de la Base de Datos

Se implementa la clase *AdministradorBD* para el módulo tal y como se muestra en la Figura 12.

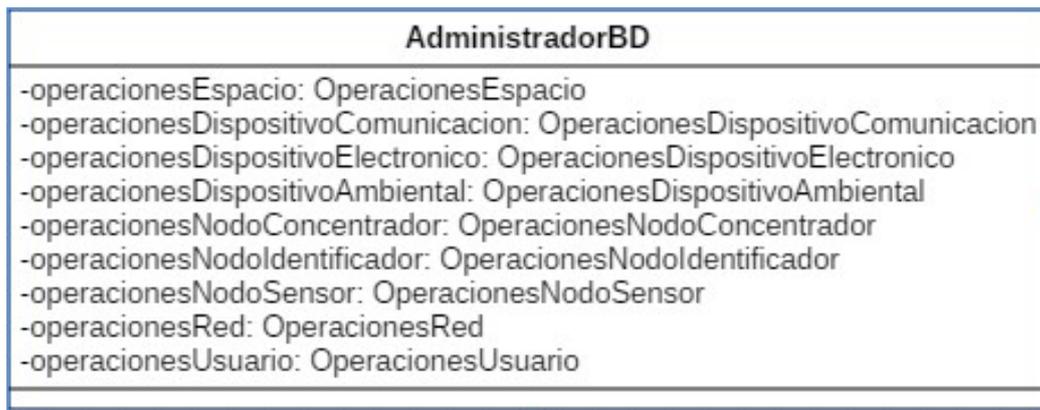


Figura 12. Clase *AdministradorBD*

La clase *AdministradorBD* interactúa con las clases *Operaciones* de Hibernate, debido a que las clases *Operaciones* son las que realizan las inserciones, modificaciones, eliminaciones u obtención de registros de cada entidad. La clase *AdministradorBD* recibe siempre parámetros de tipo *entero*, *flotante*, *String* o *boolean*, debido a que construye, dentro de cada método, los objetos necesarios que las clases *Operaciones* necesitan para trabajar.

Esta clase, además, se encarga de hacer distintas validaciones en la BD, como, por ejemplo no insertar registros o información duplicada, manteniendo así las distintas restricciones que se tienen en la BD.

Siguiendo el patrón de diseño MVC (Modelo-Vista-Controlador), esta clase se comunica con la clase Procesador, la cual será la que hará peticiones hacia la BD.

6.2 Módulo para administración de la interfaz Bluetooth

Este módulo se encarga de administrar la interfaz Bluetooth, implementa la biblioteca Bluecove.

Se implementa la clase AdministradorBluetooth para el módulo tal y como se muestra en la Figura 13.

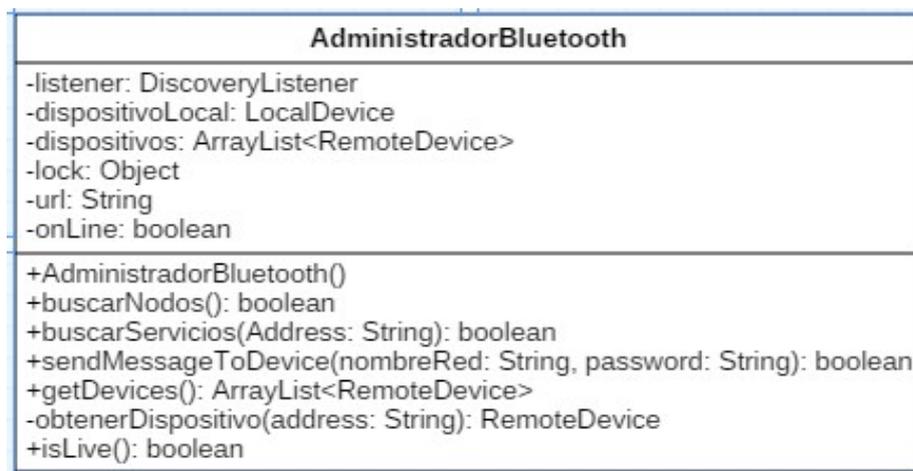


Figura 13. Clase AdministradorBluetooth

En el constructor de la clase, se implementa, también, la interfaz *DiscoveryListener* sobrescribiendo sus métodos.

6.3 Módulo para la detección de nodos.

Este módulo se encarga de detectar a todos los nodos dentro de un rango de cobertura, el cual está determinado por el alcance de la tecnología Bluetooth, para posteriormente pasarle una lista de estos al módulo de procesamiento de información. El módulo podrá detectar los dispositivos a petición del módulo de procesamiento.

Se implementa la clase DetectorNodos para el módulo tal y como se muestra en la Figura 14.

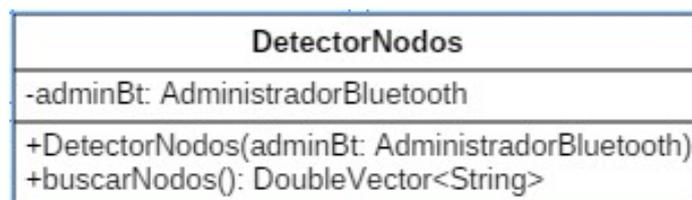


Figura 14. Clase DetectorNodos

La clase `DetectorNodos` necesita de la clase `AdministradorBluetooth`, por lo que existe una relación de composición entre estas clases, esto se ve reflejado en el constructor de la clase `DetectorNodos`.

6.4 Módulo para el envío de información por Bluetooth

Este módulo se encarga de establecer una conexión con un nodo especificado por el módulo de procesamiento para que posteriormente se envíe información de configuración, por ejemplo, el nombre y contraseña de una red Wifi para agregar al nodo a dicha red.

Se implementa la clase `EnviInformacion` para el módulo tal y como se muestra en la Figura 15.

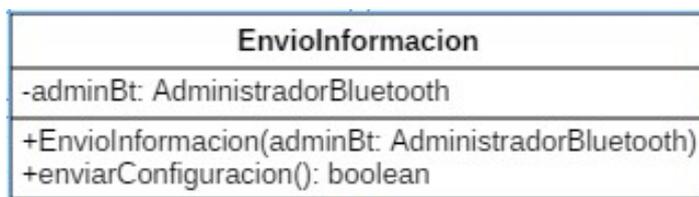


Figura 15. Clase `EnviInformacion`

La clase `EnviInformación` necesita de la clase `AdministradorBluetooth`, por lo que existe una relación de composición entre estas clases, esto se ve reflejado en el constructor de la clase `EnviInformación`.

6.5 Módulo para el procesamiento de información.

Este módulo se encargará de recibir o enviar información desde y hacia la interfaz de usuario. Además, puede mandar peticiones al módulo de alta, eliminación y modificación, por ejemplo, para solicitar que se añada un dispositivo o borrarlo del sistema. De la misma forma, puede enviar peticiones a los módulos de detección de nodos y envío de información por Bluetooth, por ejemplo, solicitar información de los nodos disponibles para posteriormente enviar alguna información de configuración. Esto implica que este módulo funciona como intermediario entre los distintos módulos del sistema.

Se implementa la clase `Procesador` para el módulo tal y como se muestra en la Figura 16.

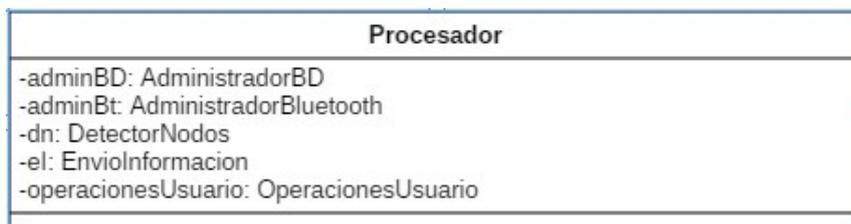


Figura 16. Clase `Procesador`

La clase `Procesador`, además, implementa diferentes métodos para mandar peticiones hacia los demás módulos.

Además, el módulo contiene *Servlets*, que funcionan como controladores, estos reciben información de los JSPs y envían o reenvían respuesta al mismo o a otros JSPs. Los

campos recibidos por los *Servlets* son utilizados por la clase Procesador, con la cual inicia las peticiones hacia los distintos módulos.

Los *Servlets* implementados son los siguientes:

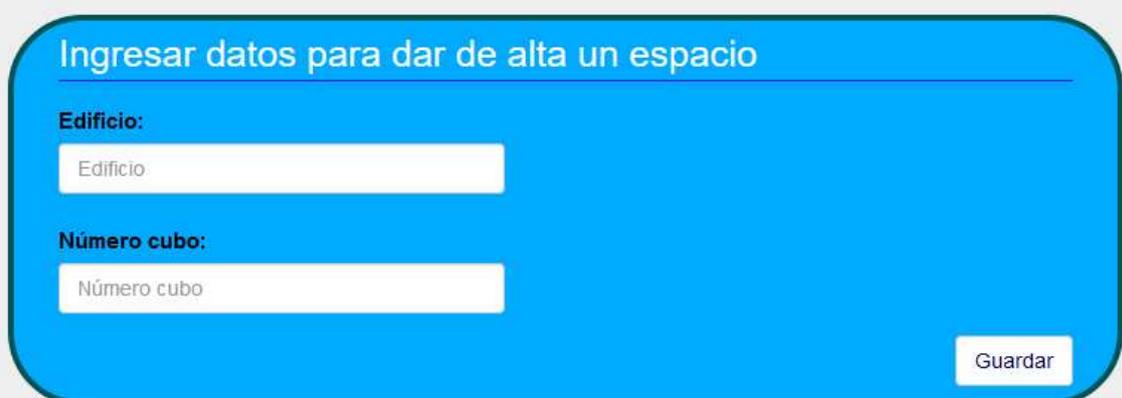
- AltaDispositivoAmbiental
- AltaDispositivoElectronico
- AltaEspacio
- AltaNodo
- AltaRed
- AltaUsuario
- EnviarInformacion
- ModificarDispositivoAmbiental
- ModificarDispositivoElectronico
- ModificarEspacio
- ModificarNodoConcentrador
- ModificarNodoIdentificador
- ModificarNodoSensor
- ModificarRed
- ModificarUsuario
- OpcionesModificarNodo

Este módulo interactúa directamente con el módulo de alta, eliminación y modificación (AdministradorBD), inicializa el módulo para administrar la interfaz Bluetooth (AdministradorBluetooth), el cual es necesario para los módulos de detección de nodos y de envío de información.

6.5.1 Peticiones de Alta

El módulo recibirá la información necesaria, a través de diferentes *Servlets*, como, por ejemplo: AltaEspacio, con el cual se procederá realizar una petición, por medio de la clase Procesador, hacia el módulo de alta, eliminación o modificación.

El usuario introduce información en un formulario tal y como se muestra en la Figura 17.



El formulario tiene un fondo azul con un título "Ingresar datos para dar de alta un espacio" en blanco. Hay dos campos de entrada de texto: "Edificio" y "Número cubo", ambos con sus respectivos labels en azul. Un botón "Guardar" está en la esquina inferior derecha.

Figura 17. Formulario para dar de alta un espacio

El formulario envía la información hacia el *Servlet* AltaEspacio, el cual obtiene cada campo enviado y este a su vez utiliza a la clase Procesador:

```
p.darAltaEspacio(edificio, numero_cubo)
```

El método *darAltaEspacio()* realiza una petición al módulo de Alta, Eliminación y Modificación, aquí es donde la clase AdministradorBD realiza la validación de no volver a insertar, en este caso, un espacio ya registrado. Dependiendo si se insertó o no, se devuelve una respuesta al módulo de procesamiento y éste a su vez regresa la respuesta al *Servlet*. El *Servlet* envía la respuesta al JSP correspondiente, indicando si se insertó el espacio, o si el espacio ya estaba registrado, o si hubo error al insertar.

También, se repetirá el mismo procedimiento, como el anterior, para dar de alta a:

- Dispositivos Ambientales
- Dispositivos Electrónicos
- Dispositivos de Comunicación
- Nodos (Concentrador, Identificador o Sensor)
- Redes
- Usuarios

Existen excepciones para:

- Dar de alta un Nodo: el *Servlet* AltaNodo diferencia y obtiene la información necesaria, dependiendo del tipo de nodo a registrar. Manda la información requerida a la clase Procesador. La clase Procesador envía una petición al Administrador de BD y obtiene una respuesta de este. Si se registra el nodo en la base de datos, la clase Procesador intentará enviar la información de configuración de Red, esto a través del módulo para el envío de información.

Se muestra el diagrama de secuencia para dar de alta un Nodo Concentrador en la Figura 18.

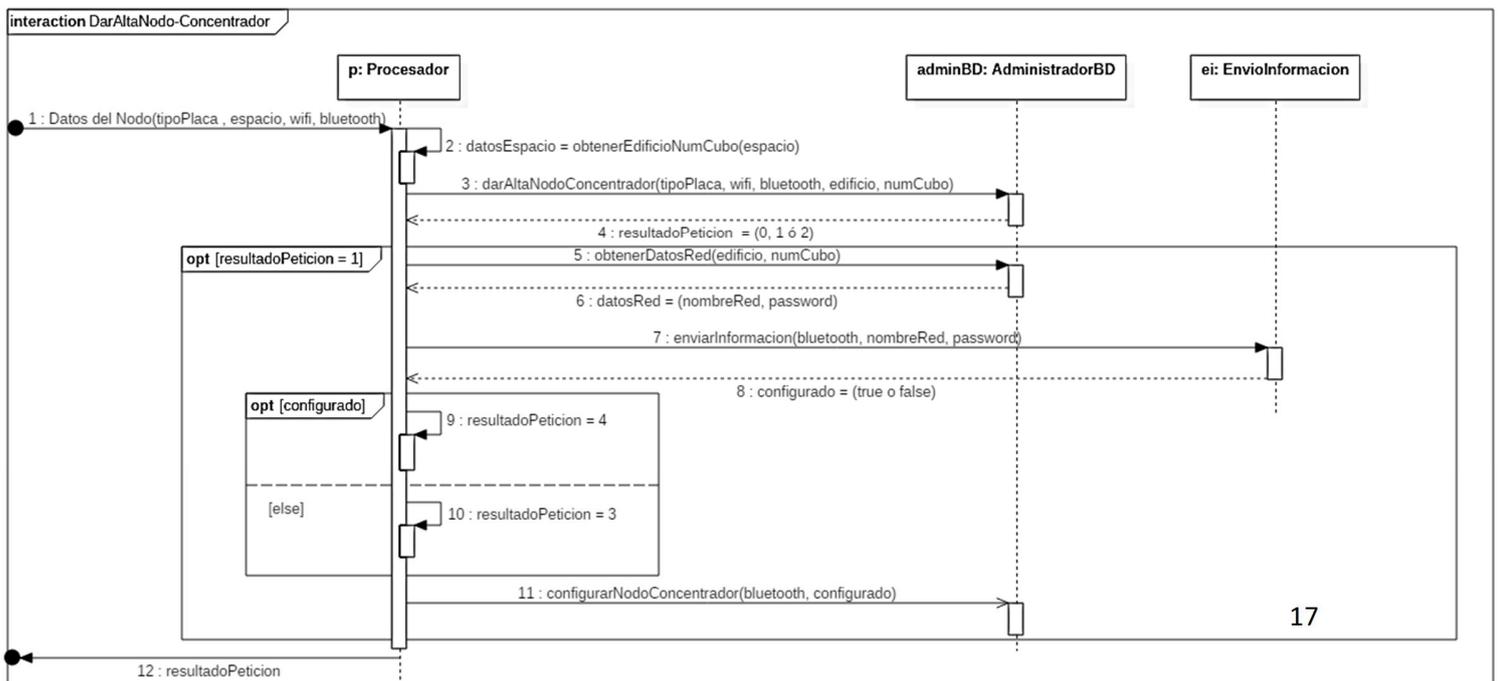


Figura 18. Diagrama de secuencia DarAltaNodo

Se repite el mismo procedimiento para dar de alta un Nodo Sensor o un Nodo Identificador.

- Dar de alta un Dispositivo Electrónico o un Dispositivo de Comunicación: el *Servlet* *AltaDispositivoElectronico* no solo obtiene información de un dispositivo electrónico, sino que también puede obtener y diferenciar información de un dispositivo de comunicación. El JSP, de donde se extraen los datos, contiene un combo box como el que se muestra en la Figura 19, en el cual se elige que tipo de dispositivo se registrará.



Figura 19. ComboBox tipo de dispositivo

- Dar de alta Redes, Usuarios, Dispositivos Ambientales y Nodos: será necesario insertar antes Espacios, ya que las entidades: redes, usuarios, dispositivos ambientales y cualquier tipo de nodo, se encuentra asociado a un espacio.

6.5.2 Peticiones de Modificación o Eliminación

Para poder modificar una entidad de la BD, se cuenta con una opción de búsqueda tal y como se muestra en la Figura 20.

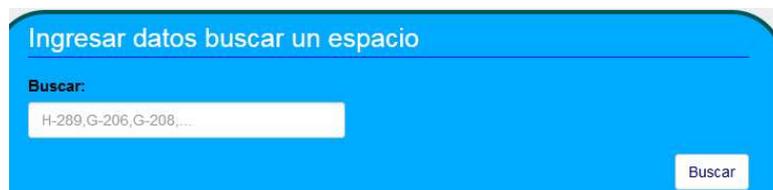


Figura 20. Formulario para buscar un espacio

La búsqueda se realiza a través del espacio, con el formato (edificio – númeroCubículo). Del mismo modo se hace la búsqueda para:

- Dispositivos Ambientales
- Dispositivos Electrónicos o de Comunicación
- Nodos (Concentrador, Identificador o Sensor)
- Redes

Para Usuarios se realiza la búsqueda a través del nombre, ya sea nombre(s) o apellido o cualquier combinación de estos.

Una vez que se obtiene una respuesta del Administrador de la BD, en la Figura 21 se mostrará una serie de opciones que el usuario puede elegir.

Figura 21. ComboBox para modificar un espacio

Al oprimir modificar, se redirigirá a otra interfaz, donde se muestran los atributos del registro que se desea modificar. Se tendrá la opción de guardar y cancelar en todos los casos. Excepto que para usuarios, redes y espacios existirá también la posibilidad de eliminar tal y como se muestra en la Figura 22.

Figura 22. Opciones en modificar espacio

En caso de guardar

El módulo de procesamiento de información, recibirá la información necesaria, a través de diferentes *Servlets*, como, por ejemplo: *ModificarEspacio*, con el cual se procederá realizar una petición, por medio de la clase *Procesador*, hacia el módulo de alta, eliminación o modificación.

El usuario modifica la información obtenida en un formulario tal y como se muestra en la Figura 23.

Figura 23. Formulario para modificar un espacio

El formulario envía la información modificada, previamente se obtiene el ID del registro a modificar. La clase *Procesador* utiliza los campos enviados como parámetros:

```
p.modificarEspacio(idEspacio, edificio, numero_cubo, opcion)
```

El método *modificarEspacio()* realiza una petición al módulo de Alta, Eliminación y Modificación. El parámetro opción deberá contener la cadena “guardar”, aquí es donde la clase AdministradorBD realiza la validación para no insertar un registro duplicado. Dependiendo si se modificó o no, se devuelve una respuesta al módulo de procesamiento y éste a su vez regresa la respuesta al *Servlet*. El *Servlet* envía la respuesta al JSP correspondiente, indicando si se actualizó el espacio, o si el espacio ya estaba registrado, o si hubo error al actualizar el registro.

También, se repetirá el mismo procedimiento, como el anterior, para modificar a:

- Dispositivos Ambientales
- Dispositivos Electrónicos
- Dispositivos de Comunicación
- Nodos (Concentrador, Identificador o Sensor)
- Redes
- Usuarios

Existen excepciones para:

- modificar un Nodo Concentrador: primero el *Servlet* OpcionesModificarNodo reenviará, dependiendo del tipo de nodo a modificar, la información al JSP correspondiente. Una vez modificado, el Servlet ModificarNodo manda la información requerida a la clase Procesador. La clase Procesador envía una petición al Administrador de BD y obtiene una respuesta de este. Si se modifica el espacio del nodo, la clase Procesador intentará enviar la nueva información de configuración de Red, esto a través del módulo para el envío de información.

En la Figura 24, se muestra el diagrama de secuencia, para modificar un Nodo Concentrador.

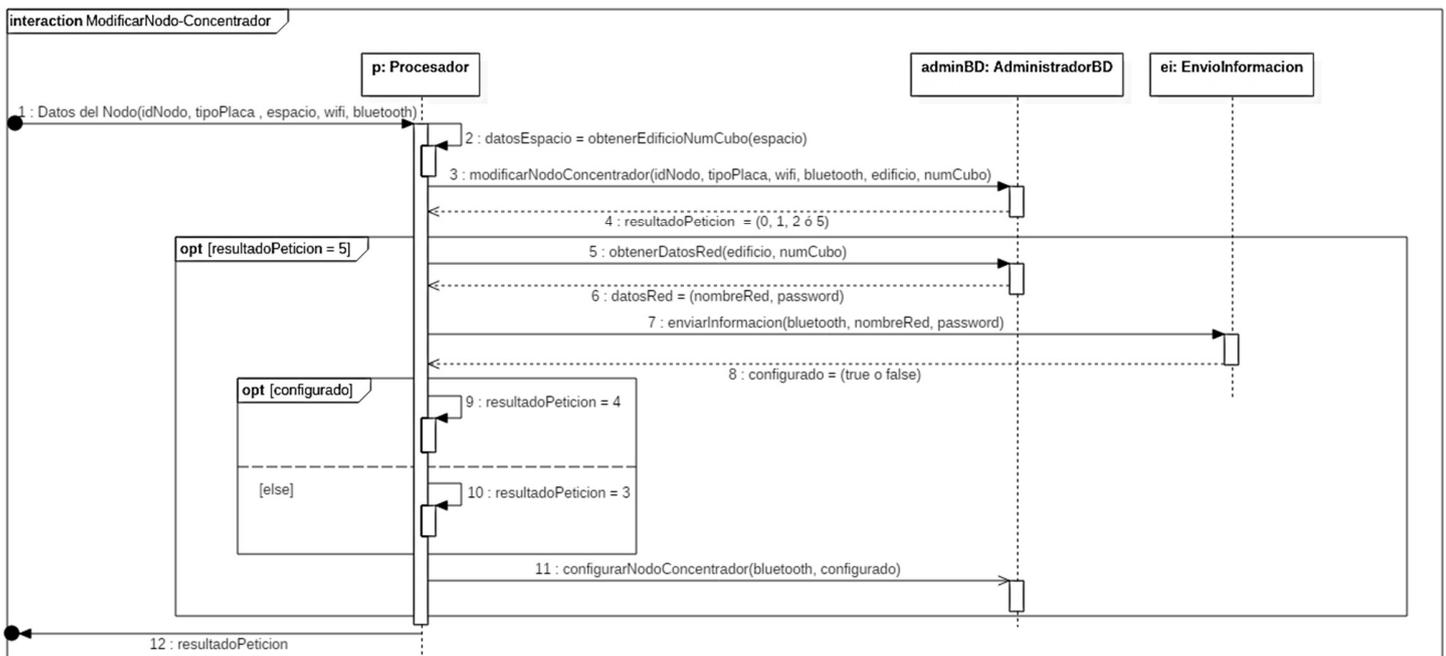


Figura 24. Diagrama de secuencia ModificarNodo

Se repite el mismo procedimiento para modificar un Nodo Sensor o un Nodo Identificador.

- modificar una Red: primero el *Servlet* ModificarRed manda la información requerida a la clase Procesador. La clase Procesador envía una petición al Administrador de BD y obtiene una respuesta de este. Una vez actualizada la Red, la clase Procesador intentará enviar la nueva información de configuración de Red, a todos los nodos que estén asociados al mismo espacio que la Red, esto a través del módulo para el envío de información. En la Figura 25, se muestra el diagrama de secuencia, para modificar una Red.

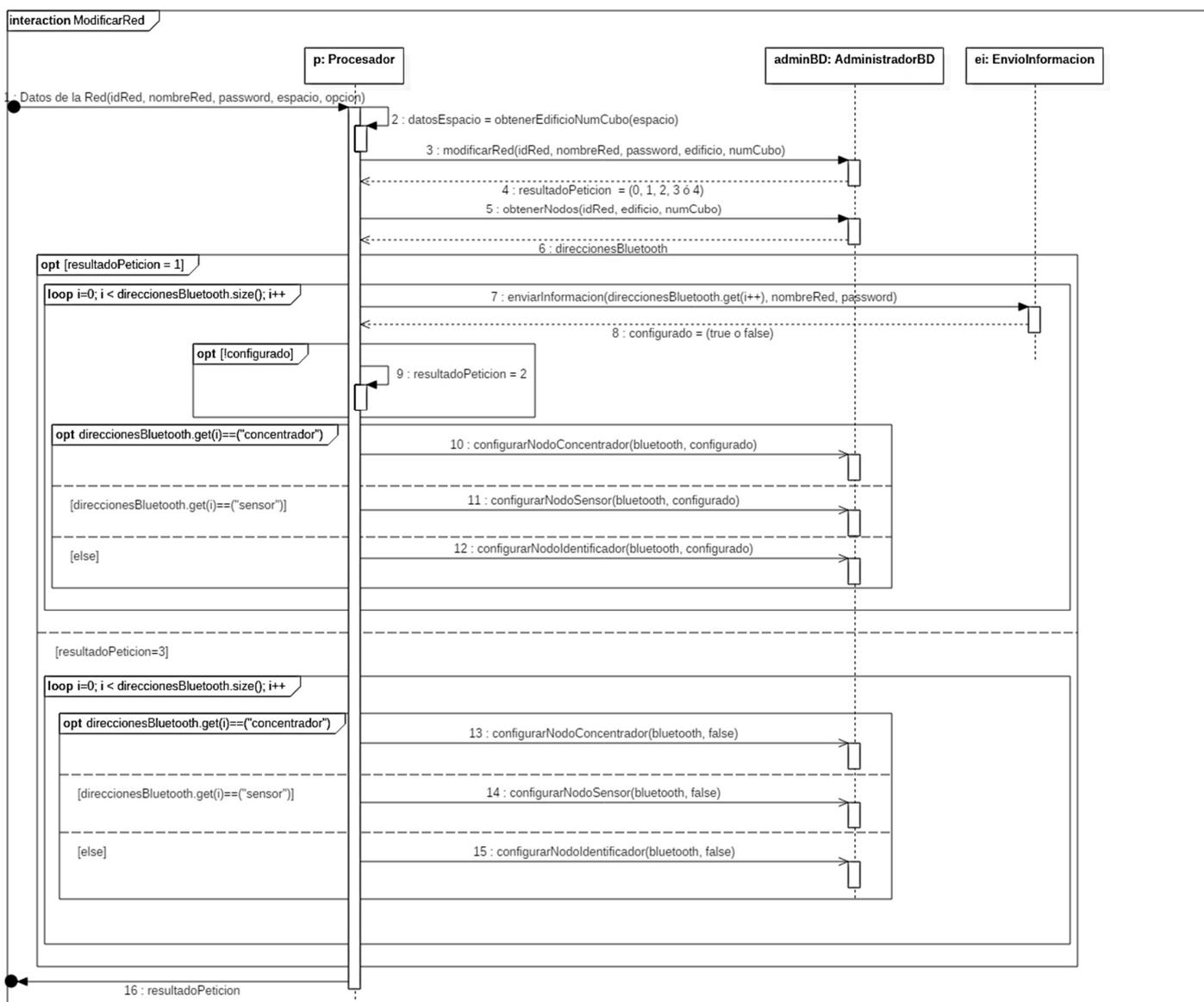


Figura 25. Diagrama de secuencia ModificarRed

En caso de eliminar

El módulo de procesamiento de información, recibirá la información necesaria, a través de diferentes *Servlets*, como, por ejemplo: *ModificarEspacio*, con el cual se procederá realizar una petición, por medio de la clase *Procesador*, hacia el módulo de alta, eliminación o modificación.

El usuario elige la opción Eliminar.

El formulario envía la información modificada, previamente se obtiene el ID del registro a modificar. La clase *Procesador* utiliza los campos enviados como parámetros:

```
p.modificarEspacio(idEspacio, edificio, numero_cubo, opcion)
```

El método *modificarEspacio()* realiza una petición al módulo de Alta, Eliminación y Modificación. El parámetro opción deberá contener la cadena "eliminar". Se devuelve una respuesta al módulo de procesamiento y éste a su vez regresa la respuesta al *Servlet*. El *Servlet* envía la respuesta al JSP correspondiente, indicando si se borro el espacio, o si hubo error al borrar el registro.

Se repite el mismo procedimiento para Usuario y Red. Hay una excepción, debido a que es posible que se encuentren dispositivos (ambientales, electrónicos y de comunicación), redes y usuarios asociados al espacio que se desea borrar, todos estos se actualizarán con un valor de 1 en el espacio, indicando que no hay un espacio asociado. De esta forma se podrá volver a asignar un espacio, en las opciones de modificar de cada entidad. Es importante que se actualicen todas las entidades con 1, de lo contrario, si al menos uno no se actualiza, no se podrá borrar el espacio y se volverá a asociar cada entidad con el espacio.

6.5.3 Peticiones a los módulos Bluetooth

Para hacer peticiones de búsqueda a través de la interfaz Bluetooth, se cuenta con dos opciones tal y como se muestra en la Figura 26.

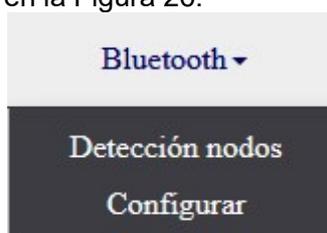


Figura 26. Opciones Bluetooth

En *Detección nodos*, se podrá realizar una búsqueda de dispositivos Bluetooth encendidos y, aparecerán aquellos que no estén asociados a un nodo ya dado de alta, es decir, que su dirección MAC Bluetooth no esté asociado a un nodo.

En *Configurar*, se podrá realizar una búsqueda de dispositivos Bluetooth encendidos y solamente se mostrará aquellos que no han sido configurados, es decir, que no hayan recibido la información de configuración de la red asociada al espacio.

En ambos casos, el módulo de procesamiento realiza peticiones al módulo de detección de nodos y, este, regresa una lista con todos los dispositivos Bluetooth que se encuentren encendidos. Además, el módulo de procesamiento hace una petición, al administrador de la BD, obteniendo así listas de direcciones MAC Bluetooth, ya sea que las direcciones estén asociadas a un nodo o que ya se encuentren configuradas, de esta forma es como el módulo de procesamiento realiza un filtrado.

6.6 Interfaces

Las interfaces para el SI se realizaron en JSP, utilizando HTML5, *Bootstrap*¹⁰ y Java. Se tuvo la necesidad hacer validaciones importantes, en ocasiones se utilizó *JavaScript*. Algunas validaciones son, por ejemplo:

- Al dar de alta de alta un Dispositivo Electrónico o de Comunicación: se encuentra un *ComboBox*, donde el usuario tiene que elegir qué tipo de dispositivo desea dar de alta, y dependiendo el tipo que seleccione, el campo Dirección MAC podrá aparecer o no, tal y como se muestra en las Figuras 27 y 28.

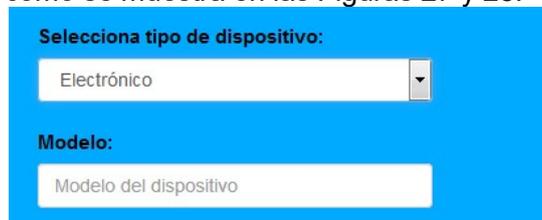
Este formulario tiene un fondo azul. En la parte superior, hay un encabezado que dice "Selecciona tipo de dispositivo:". Debajo de esto hay un menú desplegable con el texto "Electrónico" y una flecha hacia abajo. Abajo del menú, hay un encabezado "Modelo:" seguido de un campo de texto con el texto "Modelo del dispositivo".

Figura 27. Opción Dispositivo Electrónico

Este formulario tiene un fondo azul. En la parte superior, hay un encabezado que dice "Selecciona tipo de dispositivo:". Debajo de esto hay un menú desplegable con el texto "Comunicación" y una flecha hacia abajo. Abajo del menú, hay un encabezado "Dirección MAC:" seguido de un campo de texto con el texto "Dirección MAC del dispositivo". Abajo de este campo, hay un encabezado "Modelo:" seguido de un campo de texto con el texto "Modelo del dispositivo".

Figura 28. Opción Dispositivo de Comunicación

- Al dar de alta o modificar un Dispositivo de Comunicación: en la Figura 29, se encuentra un campo de texto donde se ingresa o modifica su dirección MAC. Para este tipo de campos se hace una validación para que solo se permita escribir caracteres hexadecimales.

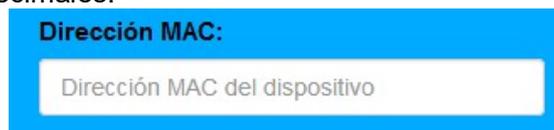
Este formulario tiene un fondo azul. En la parte superior, hay un encabezado que dice "Dirección MAC:". Debajo de esto hay un campo de texto con el texto "Dirección MAC del dispositivo".

Figura 29. Caja de texto para dirección MAC

¹⁰ Bootstrap es un framework para diseñar sitios y aplicaciones Web.

- Campos de texto donde solo se aceptan números. En la Figura 30, se muestra un ejemplo.

Figura 30. Caja de texto para Voltaje

- Campos de texto donde solo se aceptan letras. En la Figura 31, se muestra un ejemplo.

Figura 31. Caja de texto para Primer Nombre del usuario

- Al realizar una búsqueda por Espacio, solo se acepta el formato “edificio-númeroCubículo”. En la Figura 32, se muestra el campo de texto donde se realiza la búsqueda.

Figura 32. Caja de búsqueda por Espacio

- Al dar de alta dispositivos ambientales, nodos, usuarios y redes: en caso de no existir espacios registrados, el botón “Guardar” estará deshabilitado tal y como se muestra en la Figura 33.



Figura 33. Botón "Guardar" deshabilitado

- Al dar de alta una red: en el *ComboBox* para elegir el espacio, solo estarán aquellos espacios que no tengan ya una red asociada.
- Al dar de alta un nodo: al elegir qué tipo de nodo se quiere dar de alta, aparecerán o no, aquellos campos que cada tipo de nodo requiere. En caso de no existir algún campo se informará al usuario. En las Figuras 34 a 36, se muestran los campos dependiendo el tipo de nodo.

Selecciona el tipo de nodo:

Tipo de placa:

Dirección MAC WiFi del nodo:

Dirección MAC Bluetooth del nodo:

Selecciona un Espacio:

Figura 34. Opción Alta Nodo Concentrador

Selecciona el tipo de nodo:

Selecciona un Espacio:

Dispositivo WiFi:

Dispositivo Bluetooth:

Dispositivo RFID:

Dispositivo PIR:

Figura 35. Opción Alta Nodo Identificador

Selecciona el tipo de nodo:
 Sensor

Selecciona un Espacio:
 G-206

Dispositivo WiFi:
 Dirección MAC: 60E3AC719CE1, Descripción: Modelo3 - Dispositivo para nodo Ide
 Dirección MAC: A039F7719CE1, Descripción: Modelo4 - Dispositivo para nodo Sen

Dispositivo WiFi2:
 Dirección MAC: 60E3AC719CE1, Descripción: Modelo3 - Dispositivo para nodo Ide
 Dirección MAC: A039F7719CE1, Descripción: Modelo4 - Dispositivo para nodo Sen

Dispositivo Bluetooth:
 Dirección MAC: A039F7EFC0C7, Descripción: Modelo1 - Dispositivo para nodo Ser
 Dirección MAC: 60E3AC719CE2, Descripción: Modelo1 - Dispositivo para nodo Ide

Dispositivo Sensor Temperatura Humedad:
 Descripción: Modelo1 - Descripcion1
 Descripción: Modelo2 - Descripcion2

Figura 36. Opción Alta Nodo Sensor

- Al dar de alta un nodo Identificador o Sensor: al elegir los dispositivos Wifi, Bluetooth, RFID o PIR para un nodo Identificador, o al elegir los dispositivos Wifi, Wifi 2, Bluetooth, Sensor de Temperatura-Humedad, Sensor de Luz o indicador Led para un nodo Sensor; solo aparecerán aquellos dispositivos que no se encuentren asociados a un nodo tal y como se muestra en las Figuras 37 y 38.

Dispositivo WiFi:
 Dirección MAC: 60E3AC719CE1, Descripción: Modelo3 - Dispositivo para nodo Ide
 Dirección MAC: A039F7719CE1, Descripción: Modelo4 - Dispositivo para nodo Sen

Figura 37. Dispositivos Wifi no asociados

Dispositivo Bluetooth:
 Dirección MAC: A039F7EFC0C7, Descripción: Modelo1 - Dispositivo para nodo Ser
 Dirección MAC: 60E3AC719CE2, Descripción: Modelo1 - Dispositivo para nodo Ide

Figura 38. Dispositivos Bluetooth no asociados

- Al dar de alta un nodo Sensor: no se puede escoger el mismo dispositivo Wifi, en las opciones Wifi y Wifi 2. En la Figura 39, se muestra un mensaje de alerta.

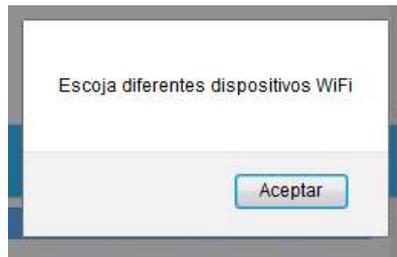


Figura 39. Mensaje de alerta Dispositivos Wifi

- Si se agrega un nuevo dispositivo por búsqueda Bluetooth, al dar de alta un nodo, si la dirección MAC Bluetooth ya estaba registrada en un dispositivo de comunicación, solo podrá agregar un nodo Identificador o Sensor, de lo contrario solo podrá agregar a un nodo Concentrador tal y como se muestra en las Figuras 40 y 41.

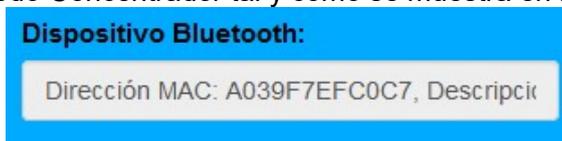


Figura 40. Información dispositivo Bluetooth ya registrado



Figura 41. Dispositivo Bluetooth no registrado

7. Resultados

El desarrollo del Sistema de Información obtuvo como resultado una aplicación Web. En las Figuras 42 a 81, se muestran las interfaces implementadas.



Figura 42. Interfaz de Inicio



Figura 43. Opciones Alta

Ingresar datos para dar de alta un dispositivo ambiental

Modelo:

Tipo:

Corriente:
 A

Voltaje:
 V

Selecciona un Espacio:

Guardar

Figura 44. Alta Dispositivo Ambiental

Inicio / Alta / Disp. Electrónicos

Ingresar datos para dar de alta un dispositivo electrónico o de comunicación

Selecciona tipo de dispositivo:

Modelo:

Tipo:

Descripción:

Rango valores:

Unidades:

Voltaje:
 V

Guardar

Figura 45. Alta Dispositivo Electrónico/Comunicación

Inicio

Alta ▾

Modificar ▾

Bluetooth ▾

Inicio / Alta / Espacios

Ingresa datos para dar de alta un espacio

Edificio:

Número cubo:

Guardar

Proyecto Terminal

Erick Iván Medina Salas
eims@azc.uam.mx

Figura 46. Alta Espacio

Inicio

Alta ▾

Modificar ▾

Bluetooth ▾

Inicio / Alta / Nodos

Agregue el nuevo nodo, seleccione las diferentes opciones

Selecciona el tipo de nodo:

Tipo de placa:

Dirección MAC WiFi del nodo:

Dirección MAC Bluetooth del nodo:

Selecciona un Espacio:

Guardar

Figura 47. Alta Nodos

Ingresar datos para dar de alta una red

Nombre de la Red:

Password:

Selecciona un Espacio:

Guardar

Proyecto Terminal

Erick Iván Medina Salas
eims@azc.uam.mx

Figura 48. Alta Redes

Ingresar datos para dar de alta un usuario

Número económico:

Primer Nombre:

Segundo Nombre(opcional):

Apellido Paterno:

Apellido Materno:

Correo:

Teléfono:

Selecciona un Espacio:

Guardar

Figura 49. Alta Usuarios

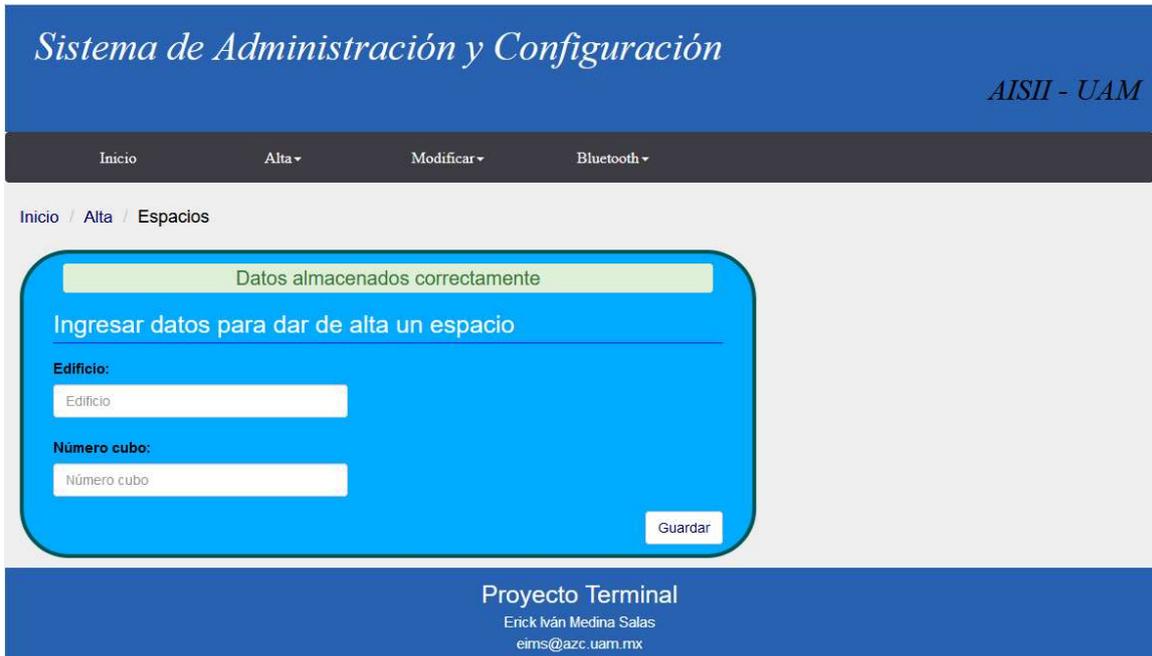


Figura 50. Mensaje "Datos Almacenados Correctamente"



Figura 51. Mensaje "Espacio repetido"

Elija una de las siguientes opciones

Dispositivos Ambientales

Dispositivos Electrónicos

Espacios

Nodos

Usuarios

Figura 52. Opciones Modificación

Ingresar datos buscar un dispositivo ambiental

Buscar:

H-289,G-208,F-203,...

Buscar

Elija un dispositivo de la siguiente lista

Seleccione una Opción:

Modificar

Proyecto Terminal

Erick Iván Medina Salas
eims@azc.uam.mx

Figura 53. Búsqueda de Dispositivos Ambientales para modificar

Inicio

Alta

Modificar

Bluetooth

Inicio / Modificar / Disp. Ambientales

Ingresar datos buscar un dispositivo ambiental

Buscar:

H-289,G-208,f-203,...

Buscar

Elija un dispositivo de la siguiente lista

Seleccione una Opción:

Modelo: BBB-C-5, Tipo: Luz, Ubicación: H-289
Modelo: AAA-A-6, Tipo: Ventilador, Ubicación: H-289
Modelo: AAA-B-7, Tipo: Ventilador, Ubicación: H-288
Modelo: Prueba1, Tipo: prueba1, Ubicación: Sin espacio

Modificar

Proyecto Terminal

Erick Iván Medina Salas

Figura 54. Resultado de búsqueda Dispositivo Ambiental

Inicio

Alta

Modificar

Bluetooth

Inicio / Modificar / Disp. Ambiental / Cambiar

Modificar datos del dispositivo ambiental

Modelo:

BBB-C-5

Tipo:

Luz

Corriente:

10 A

Voltaje:

5 V

Selecciona un Espacio:

H-289

Cancelar

Guardar

Figura 55. Interfaz para modificar un dispositivo ambiental

Ingresar datos buscar un dispositivo electrónico

Buscar:
H-289,G-208,f-203,...

Selecciona tipo de dispositivo: Electrónico
Electrónico
Comunicación

Buscar

Elija un dispositivo de la siguiente lista

Seleccione una Opción:

< >

Modificar

Figura 56. Búsqueda de Dispositivos Electrónicos/Comunicación para modificar

Ingresar datos buscar un dispositivo electrónico

Buscar:
H-289,G-208,f-203,...

Selecciona tipo de dispositivo: Electrónico

Buscar

Elija un dispositivo de la siguiente lista

Seleccione una Opción:

Modelo: Modelo1, Descripción: Descripción1, Tipo: rfid, Tipo Nodo: Identificador, Espac
Modelo: Modelo3, Descripción: Descripción3, Tipo: pir, Tipo Nodo: Identificador, Espac

< >

Modificar

Proyecto Terminal
Erick Iván Medina Salas
eims@azc.uam.mx

Figura 57. Resultado de búsqueda Dispositivo Electrónico/Comunicación

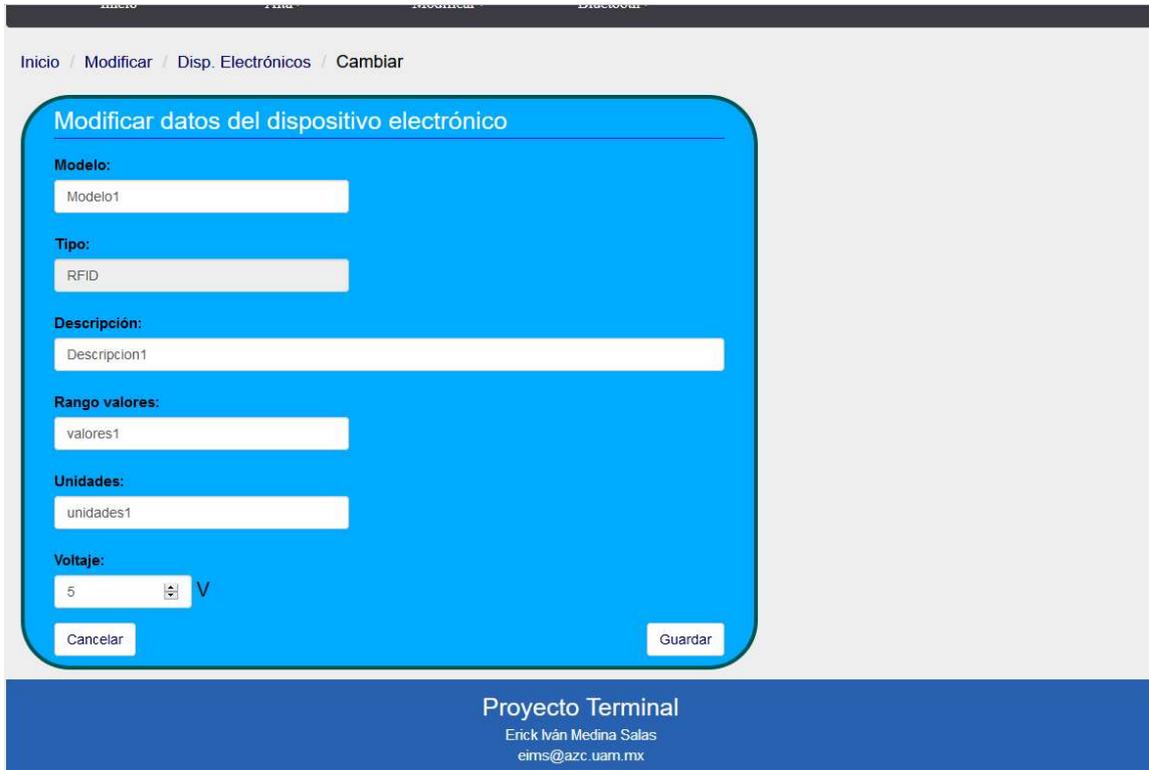


Figura 58. Interfaz para modificar un Dispositivo Electrónico

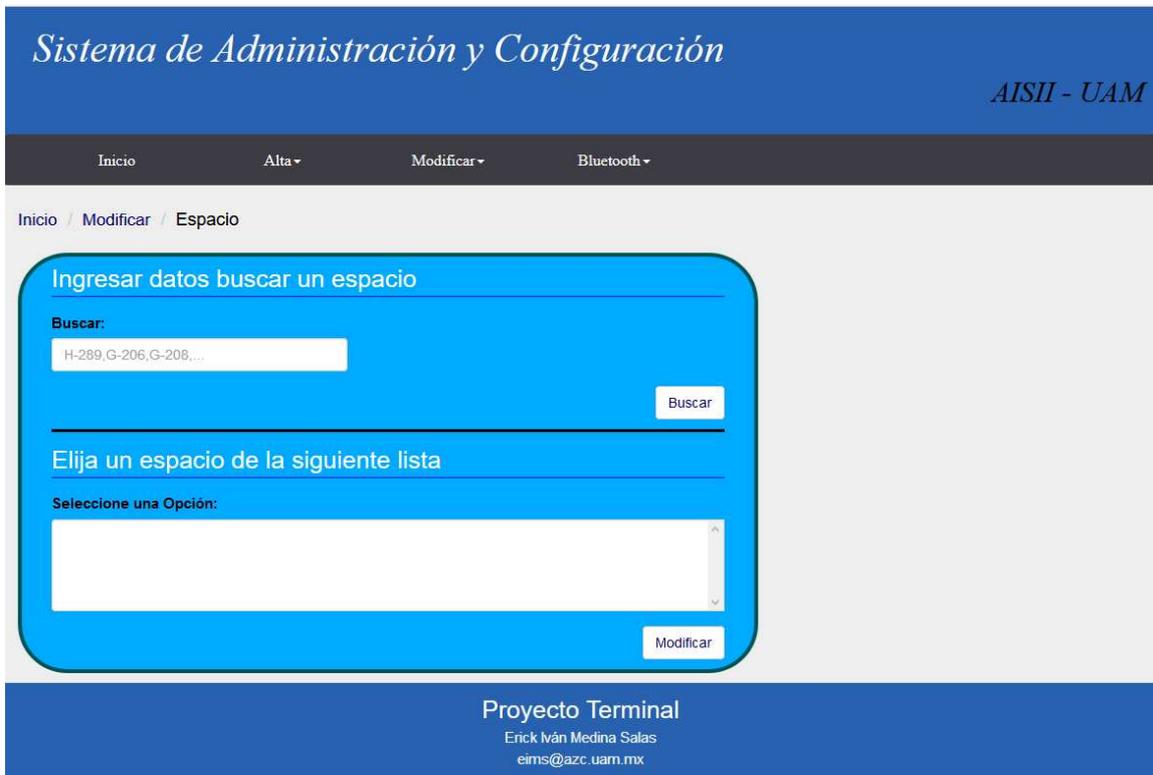


Figura 59. Búsqueda de Espacios para modificar

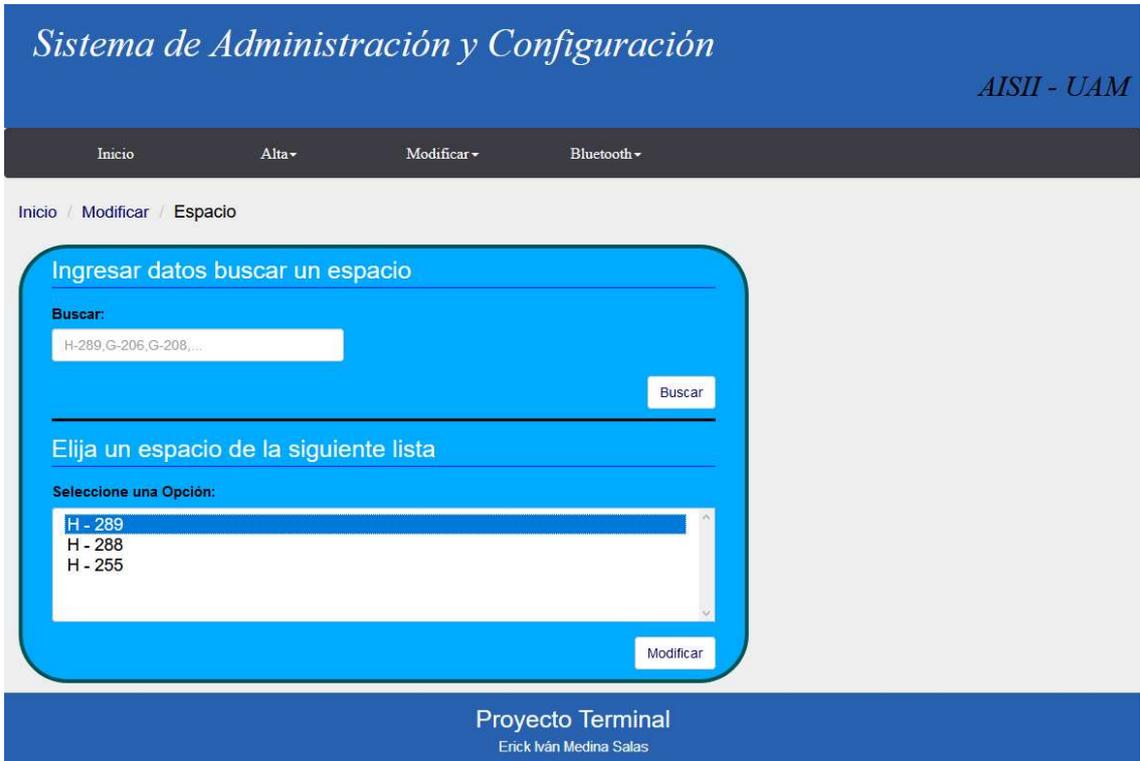


Figura 60. Resultado de búsqueda Espacio



Figura 61. Interfaz para modificar un Espacio

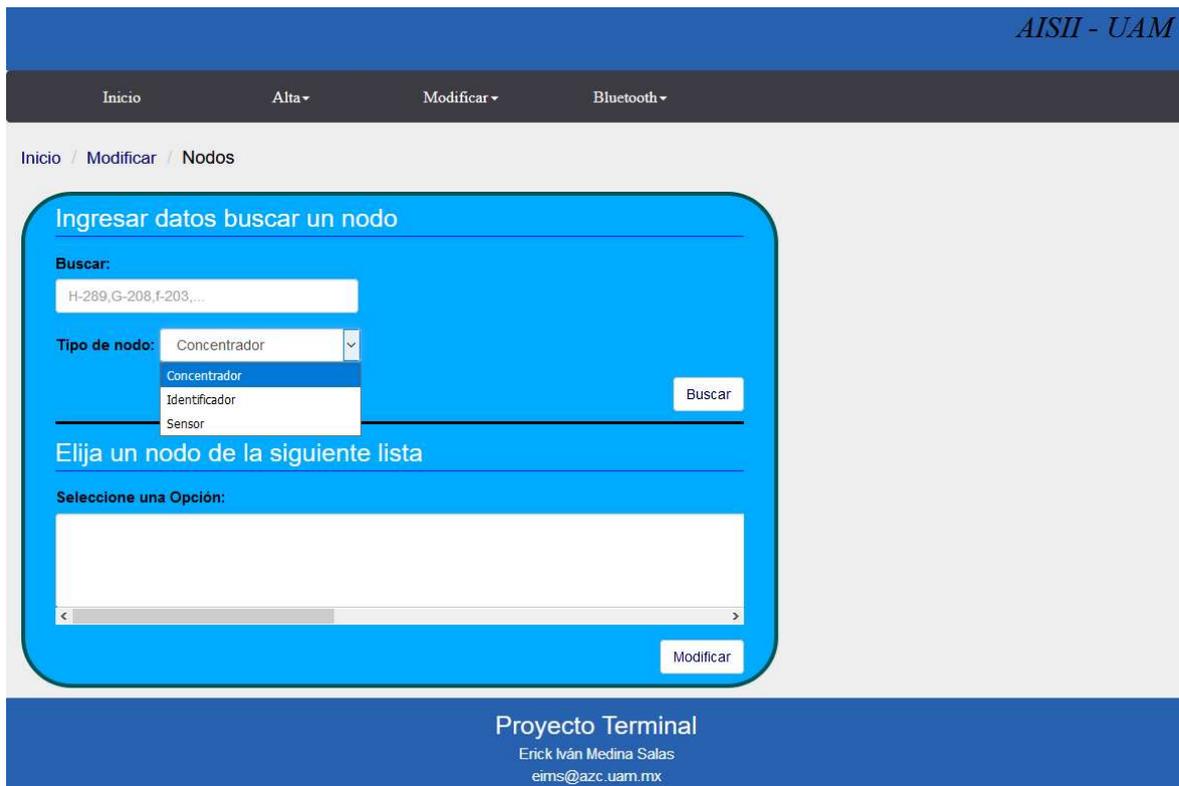


Figura 62. Búsqueda de Nodos para modificar

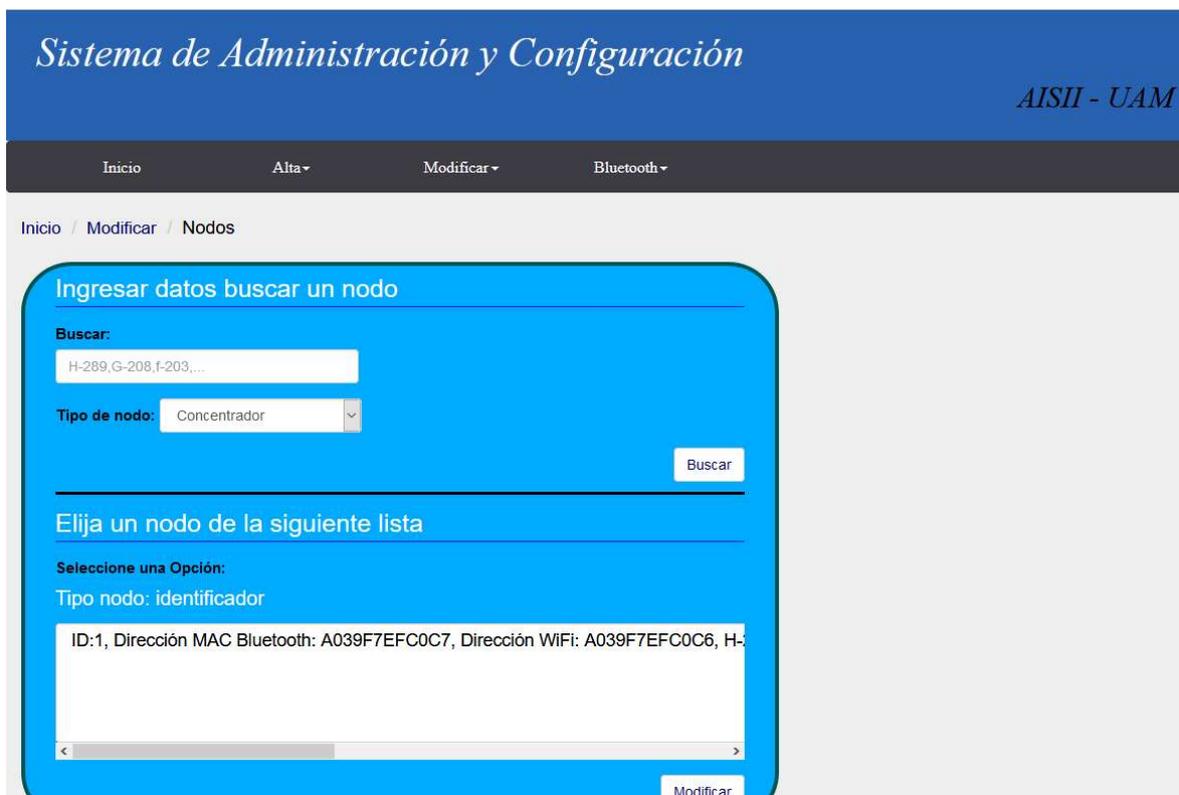


Figura 63. Resultado de búsqueda Nodo

Modificar datos del nodo identificador

Dispositivo WIFI:
 Dirección MAC: A039F7EFC0C6, Descripción: Modelo2 - Dispositivo para nodo S

Dispositivo Bluetooth:
 Dirección MAC: A039F7EFC0C7 - Modelo1

Dispositivo RFID:
 Descripción: Modelo2 - Descripción2
 Descripción: Modelo1 - Descripción1

Dispositivo PIR:
 Descripción: Modelo4 - Descripción4
 Descripción: Modelo3 - Descripción3

Selecciona un Espacio:
 H-288

Cancelar Guardar

Figura 64. Interfaz para modificar un Nodo

Sistema de Administración y Configuración AISII - UAM

Inicio Alta Modificar Bluetooth

Inicio / Modificar / Nodos

Datos actualizados correctamente

Ingresa datos buscar un nodo

Buscar:
 H-289,G-208,F-203,...

Tipo de nodo: Concentrador

Buscar

Elija un nodo de la siguiente lista

Seleccione una Opción:

Modificar

Figura 65. Mensaje "Datos actualizados correctamente"

Datos actualizados correctamente, información de configuración enviada

Ingresa datos para buscar un nodo

Buscar:

H-289,G-208,F-203,...

Tipo de nodo:

Concentrador ▾

Buscar

Elija un nodo de la siguiente lista

Seleccione una Opción:

[Empty list box]

Modificar

Figura 66. Mensaje "Datos actualizados correctamente, información de configuración enviada"

Datos actualizados correctamente, error al enviar información configuración

Ingresa datos para buscar un nodo

Buscar:

H-289,G-208,F-203,...

Tipo de nodo:

Concentrador ▾

Buscar

Elija un nodo de la siguiente lista

Seleccione una Opción:

[Empty list box]

Figura 67. Mensaje "Datos actualizados correctamente, error al enviar información de configuración"

Ingresar datos buscar una red

Buscar:

H-289,G-208,f-203,...

Buscar

Elija una red de la siguiente lista

Seleccione una Opción:

Modificar

Proyecto Terminal

Erick Iván Medina Salas
eims@azc.uam.mx

Figura 68. Búsqueda de Redes para modificar

Ingresar datos buscar una red

Buscar:

H-289,G-208,f-203,...

Buscar

Elija una red de la siguiente lista

Seleccione una Opción:

Nombre: Laboratorio G-208, Tipo: passwordg206, Ubicación: G-208

Modificar

Proyecto Terminal

Erick Iván Medina Salas
eims@azc.uam.mx

Figura 69. Resultado de búsqueda Red

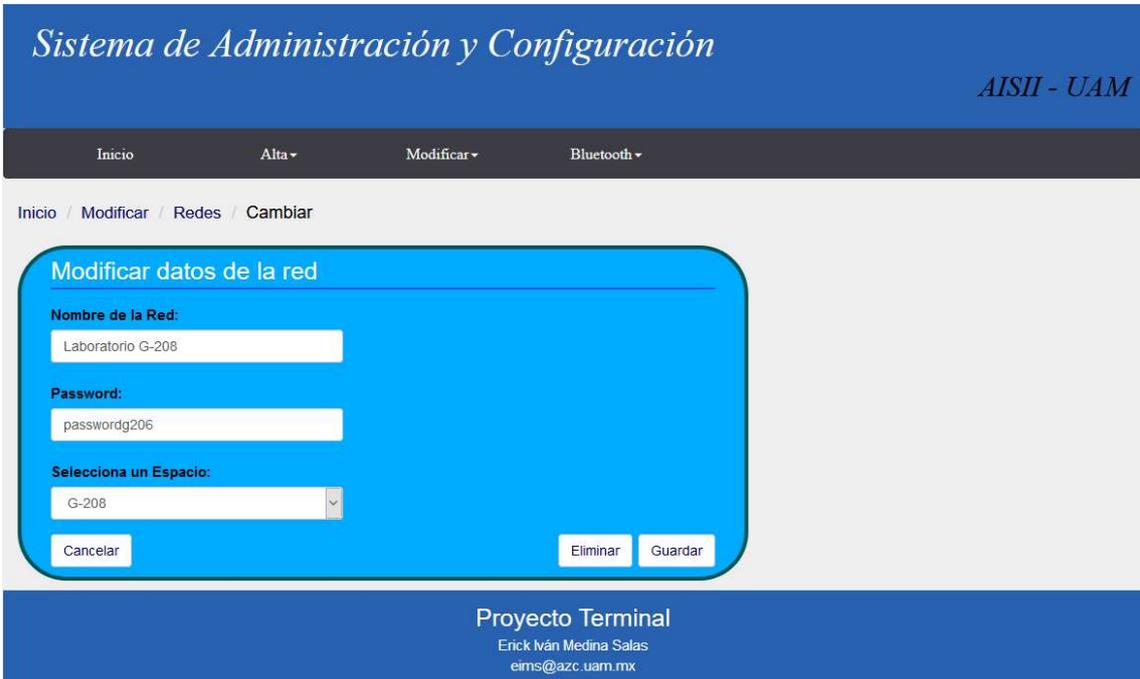


Figura 70. Interfaz para modificar una Red

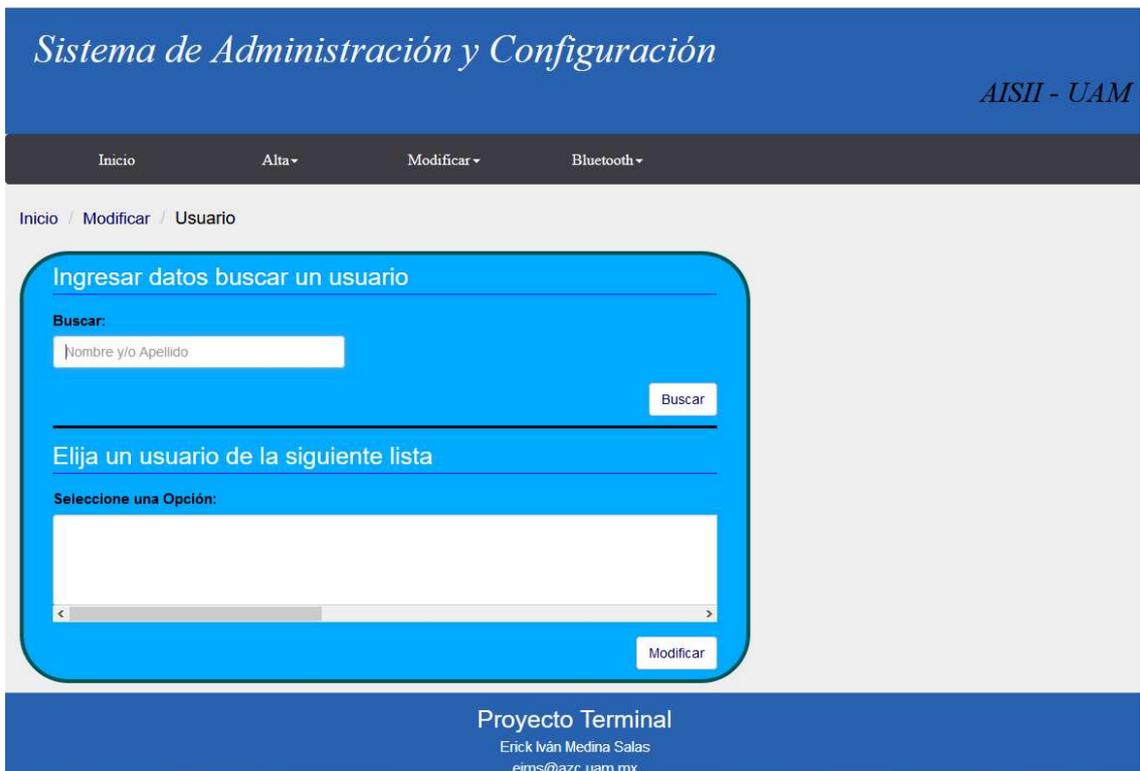


Figura 71. Búsqueda de Usuarios para modificar

Inicio Alta Modificar Bluetooth

Inicio / Modificar / Usuario

Ingresar datos buscar un usuario

Buscar:

Elija un usuario de la siguiente lista

Seleccione una Opción:
Nombres: Leonardo Daniel, Apellidos: Sánchez Martínez, Número Económico: 456789

Proyecto Terminal
Erick Iván Medina Salas

Figura 72. Resultado de búsqueda Usuario

Modificar datos del usuario

Número económico:

Primer Nombre:

Segundo Nombre(opcional):

Apellido Paterno:

Apellido Materno:

Correo:

Teléfono:

Selecciona un Espacio:

Figura 73. Interfaz para modificar un Usuario

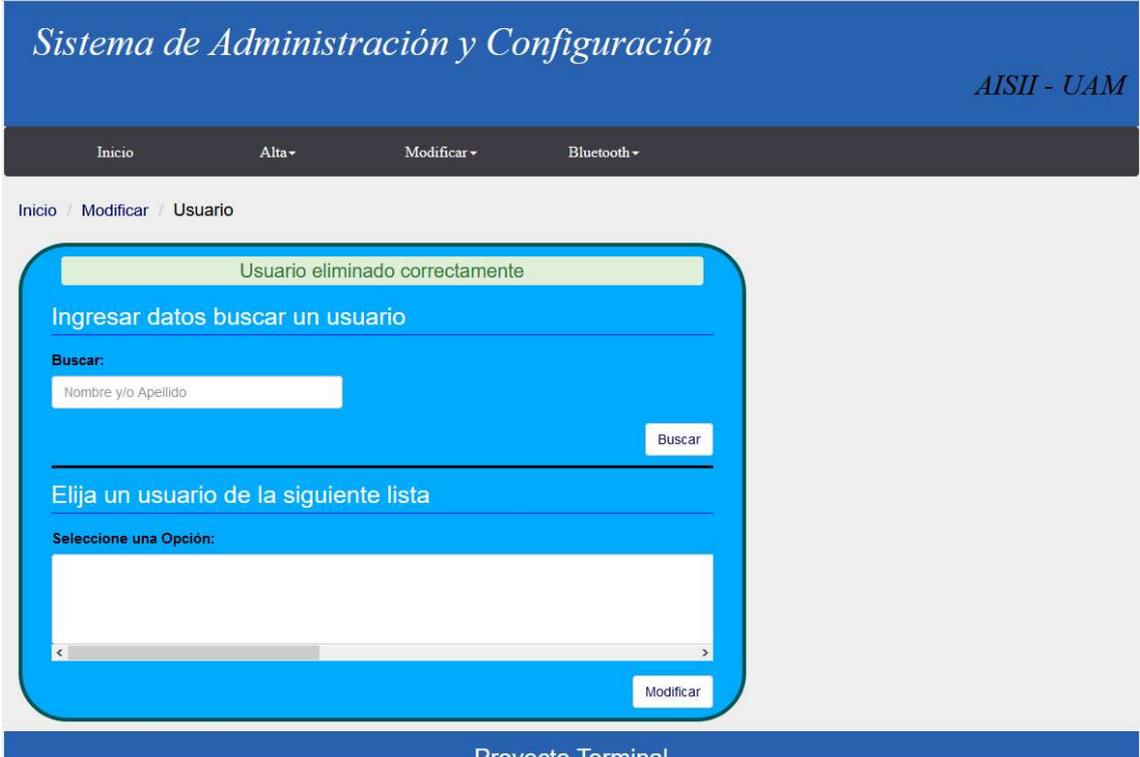


Figura 74. Mensaje "Usuario eliminado correctamente"



Figura 75. Opciones Bluetooth

Inicio / Detección nodos

Dispositivos Bluetooth disponibles cercanos

| Dirección MAC | Nombre | Agregar |
|---------------|------------------|---------|
| F0D7AA7B452C | Dispositivo G(5) | Agregar |
| AC37438EF18E | Dispositivo H | Agregar |
| 60E3AC719CE2 | LG ZONE | Agregar |

Actualizar

Proyecto Terminal

Erick Iván Medina Salas
eims@azc.uam.mx

Figura 76. Búsqueda de dispositivos Bluetooth cercanos, no asociados

Inicio / Bluetooth / Detectar Nodos / Nuevo nodo

Agregue el nuevo nodo, seleccione las diferentes opciones

Selecciona el tipo de nodo:

Concentrador

Tipo de placa:

Arduino

Dirección MAC WiFi del nodo:

Dirección MAC WiFi

Dirección MAC Bluetooth del nodo:

F0D7AA7B452C

Selecciona un Espacio:

G-206

Cancelar

Guardar

Proyecto Terminal

Erick Iván Medina Salas
eims@azc.uam.mx

Figura 77. Interfaz para agregar un nuevo nodo, por medio de la detección



Figura 78. Mensaje "Datos almacenados correctamente, información de configuración no pudo ser enviada"

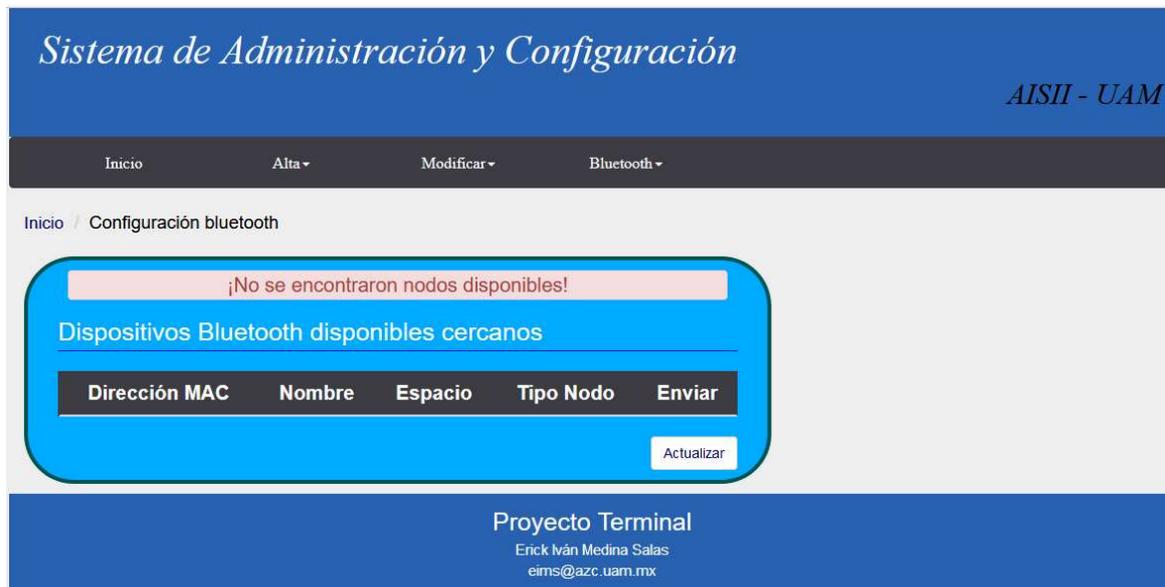


Figura 79. Mensaje "¡No se encontraron nodos disponibles!"



Figura 80. Búsqueda de dispositivos Bluetooth cercanos, sin configurar



Figura 81. Mensaje "Información de configuración enviada"

8. Conclusiones y trabajo a futuro

Para este proyecto, se desarrolló una aplicación Web para un Sistema de Información, logrando cumplir los objetivos propuestos para el mismo. El SI es capaz de administrar y configurar una red de sensores de manera inalámbrica, dentro de un espacio inteligente, utilizando Bluetooth.

Actualmente existen sistemas de configuración inalámbricos que realizan la misma tarea, como *Cisco Secure Access Control Server* [7] o el Sistema de Distribución Inalámbrica (WDS) [8], pero se necesita de otros componentes para la realización de la misma tarea, por lo que resultaría más costoso económicamente. El SI implementado está adaptado a las necesidades requeridas por la red de sensores y el espacio inteligente.

La implementación del este proyecto con lleva a las siguientes ventajas:

- La configuración de los nodos se hace de manera automática, pues una vez obtenido los datos necesarios, se puede enviar la información de configuración con solo dar clic en un botón, las veces que se requieran. Reduciendo tiempo, debido a que, de no existir el proyecto, se tendría que configurar nodo por nodo de manera manual.
- Se administra de una manera más fácil cada espacio, incluyendo los nodos, dispositivos, usuarios y redes asociados a cada uno de ellos.
- En caso de extender el proyecto, es decir, aumentar más “entidades” al espacio inteligente, se puede hacer de una manera más rápida debido a la organización que el SI implementa, pues gracias al diseño MVC junto con el *framework Hibernate*, es más fácil trabajar.

En cuanto a usabilidad de la aplicación, se pudiera mejorar las interfaces de manera que sea más amigable e intuitiva para el usuario. En cuanto al manejo de la interfaz Bluetooth, se pudiera mejorar los tiempos de respuesta por la biblioteca Bluecove. También, se pudiera utilizar algún *framework* o biblioteca para remarcar mejor el diseño MVC implementado por la aplicación.

9. Referencias Bibliográficas

- [1] J. Ignacio Vega, G. Salgado, M. A. Lagos y V. N. Tapia, F. J. Sánchez, "Monitoreo de temperatura y humedad en un campo de cultivo utilizando el protocolo bluetooth," en SOMI Congreso de Instrumentación, Jalisco, México, Oct. 2014. [En línea] Disponible: <http://somi.ccadet.unam.mx/somi29/memoriassomi29/PDFS/electronica/100-SGSOMI-66-100.pdf>
- [2] J. W. Nicols Rafael y P. E. Rojas López, "Monitoreo y control por Bluetooth," Proyecto terminal, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco, México, 2006.
- [3] M. I. Tixi Paucar, "Configuración e implementación de un sistema de control aplicando la tecnología Bluetooth con PLC, caso práctico laboratorio de automatización (EIS)," Tesis de licenciatura, 2009. [En línea] Escuela Superior Politécnica de Chimborazo, Rio Bamba, Ecuador. Disponible: <http://dspace.esPOCH.edu.ec/bitstream/123456789/112/1/18T00400.pdf>.
- [4] C. E. Ortiz Sampedro, "Diseño e implementación de una aplicación para teléfonos celulares que permita la configuración de equipos Cisco utilizando un enlace Bluetooth," Tesis de licenciatura, 2010. [En línea] Escuela Politécnica Nacional, Quito. Disponible: <http://bibdigital.epn.edu.ec/bitstream/15000/2675/1/CD-3360.pdf>.
- [5] "Sistemas Inteligentes de Transportes," Autopistas particulares, 2017. [En línea]. Disponible: <https://www.autopistas.com/es/sistemas-inteligentes-de-transportes>.
- [6] J. Iraceburu González y J. Goicoechea Fernández, "Desarrollo e implementación de una red inalámbrica de sensores de temperatura y humedad," Tesis de licenciatura, 2014. [En línea] Universidad Pública de Navarra. Disponible: http://academica-e.unavarra.es/bitstream/handle/2454/11846/TFG_IraceburuGonzalezJulen2014.pdf?sequence=1.
- [7] "Estaciones base Wi-Fi: Instalación y configuración de un sistema de distribución inalámbrico (802.11b/g)", Apple Support, 2017. [En línea]. Disponible: <https://support.apple.com/es-mx/HT202076>.
- [8] "La administración inalámbrica del sistema de control y del Sistema de control de redes con el ejemplo de configuración ACS 5.x", Cisco, 2017. [En línea]. Disponible: https://www.cisco.com/c/es_mx/support/docs/wireless/wireless-control-system/116004-ncs-technote-00.html.

10. Apéndices

Se anexa código fuente de cada módulo integrado del SI, JavaDoc y script de generación de la BD en un disco CD.