

Universidad Autónoma Metropolitana Unidad Azcapotzalco
División de Ciencias Básicas e Ingeniería
Licenciatura en Ingeniería en Computación

Administración de información de un espacio inteligente

Modalidad: Proyecto de investigación
Trimestre 2017 Primavera

Alumna: Guadalupe Aguilera Cortés
Matrícula: 2171998924
Correo: guadalupaguilera@gmail.com

Asesor: José Alejandro Reyes Ortiz
Profesor Asociado
Núm. Económico: 37847
Correo: jaro@correo.azc.uam.mx
Departamento de Sistemas

Asesor: Leonardo Daniel Sánchez Martínez
Núm. Económico: 39768
Correo: lds@correo.azc.uam.mx
Departamento de Sistemas

21 de julio de 2017

Declaratoria

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.



José Alejandro Reyes Ortiz

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

Yo, Guadalupe Aguilera Cortés, 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.



Guadalupe Aguilera Cortés

Resumen

El siguiente trabajo describe el comportamiento de sensores basado en comunicación inalámbrica desarrollado con el fin de controlar variables ambientales existentes en un espacio inteligente y de esta manera a se pueda llevar un control de los eventos que suceden dentro de este.

Dentro de lo anterior un espacio inteligente es un lugar físico como habitaciones, lugares de trabajo, hogares entre otros que se componen de diversos sensores con el fin de brindar servicios y facilitar el trabajo pesado por un sistema tecnológico que se adapte a las necesidades del usuario.

Es por ello que se desarrollará un espacio inteligente para que este se adapte de manera automática con el fin de mejorar la calidad de este y para que brinde satisfacción al usuario sustituyendo el trabajo peligroso, trabajo físico y tareas repetitivas sin necesidad de perder tiempo realizando este tipo de actividades.

El espacio inteligente se desarrollará a través de una red de sensores las cuales medirán las variables de temperatura, luminosidad, presencia y humedad para posteriormente poder administrar la información obtenida por los sensores y hacer un análisis de las preferencias del usuario.

Para la realización del espacio inteligente se consideraron tres módulos: Modulo físico de sensores, módulo de toma de decisiones y el módulo de base de datos. El modulo físico de sensores es el encargado de capturar datos de dicho espacio. El módulo de toma de decisiones es el encargado de determinar la difusión de los datos adquiridos por el módulo físico de sensores y de la ejecución de una acción para regular el ambiente la información obtenida se administrará en una base de datos por lo cual se consideró capturar información de las variables ambientales de temperatura, humedad, luminosidad y presencia que provienen de la red de para que de esta manera se lleve un control de los eventos que sucedan en dicho espacio.

Tabla de contenido

1	Introducción	7
2	Antecedentes.....	7
2.1	Proyectos de Integración o terminales	7
2.1.1	Aplicación concurrente para la obtención y almacenamiento de datos a partir de sensores [3].	7
2.1.2	Red de monitoreo remoto inalámbrico de sensores con dispositivos ZIGBEE [4].	8
2.2	Tesis.....	8
2.2.1	Red de sensores para el control de contextos usando contextos web [5]. ...	8
2.2.2	Estudio de la plataforma de Android para dispositivos móviles y desarrollo de aplicación para la administración de redes de sensores [6].	8
2.3	Artículos	8
2.3.1	Control Automático de Condiciones Ambientales en Domótica usando Redes Neuronales Artificiales [7].	9
2.3.2	Sistemas de sensores inalámbricos para la implementación de espacios inteligentes [8].	9
3	Justificación	9
4	Objetivos.....	9
4.1	Objetivo General.....	9
4.2	Objetivos Particulares.....	9
5	Marco teórico.....	10
5.1	Entornos inteligentes	10
5.2	Administración de la información	10
5.3	Sockets.....	10
5.3.1	Arquitectura de Cliente-Servidor	11
5.3.2	Socket Servidor	11
5.3.3	Socket Cliente	11
5.4	Concurrencia	11
5.5	Sensor	12
5.6	Netbeans	12
5.6.1	Lenguaje de programación Java.....	12

5.7	MySQL	13
6	Desarrollo del proyecto	13
6.1	Tecnologías usadas.....	13
6.2	Implementación	13
6.2.1	Módulo físico de sensores.	15
6.2.2	Módulo de base de datos	16
6.2.3	Módulo de Toma de decisiones	18
7	Resultados.....	23
8	Análisis y discusión de resultados.....	27
9	Conclusiones	27
10	Referencias bibliográficas	28
11	Apéndice	29
11.1	Script de SQL de la Base de Datos	29
11.2	Código Fuente para la toma de decisiones	33

Índice de Figuras

Figura 1. Arquitectura Cliente Servidor	11
Figura 2. Interfaz de desarrollo de Netbeans	12
Figura 3. Módulo del proyecto.....	13
Figura 4. Diagrama de comunicación entre los dispositivos participantes del entorno inteligente	15
Figura 5. Diagrama Entidad Relación	16
Figura 6. Base de datos en MySQL	17
Figura 7. Diagrama de clases de Servidor	19
Figura 8. Simulación de 45 Clientes.....	22
Figura 9. Clase para crear números aleatorios	22
Figura 10. Tiempo de envío de datos.....	22
Figura 11. Datos generados aleatoriamente	23
Figura 12. Datos aleatorios almacenados en la tabla nodo sensor	24
Figura 13. Cambio de las variables ambientales durante la ejecución	24
Figura 14. Variación de la temperatura	25
Figura 15. Variación de la humedad	25
Figura 16. Variación de la luminosidad	26
Figura 17. Variación de la presencia.....	26
Figura 18. Mensaje de error al ejecutar más de 45 clientes	27

Índice de tablas

Tabla 1. Paquete-Inf. Contiene los valores leídos de los sensores. Considerar que no hay presencia.....	14
Tabla 2. Ejemplo de un Paquete-Inf.....	14
Tabla 3. Paquete-Event. Contiene el tipo de evento generado	14
Tabla 4. Modelo de la base de datos	17
Tabla 5. Variables ambientales de acuerdo a las normas de la Secretaría del Trabajo y Previsión Social	19
Tabla 6. Rangos a consideración cambio de variable ambiental.....	20

1 Introducción

La evolución de las tecnologías inalámbricas ha permitido la integración del mundo físico en el ámbito del mundo digital de manera que cualquier elemento que se encuentre en el ambiente tiene la capacidad de proporcionar información de los eventos que suceden en su entorno a través de diversos medios de comunicación [1]. Uno de los que más ha llamado la atención son los sensores inalámbricos [2], los cuales requieren la interconexión hacia redes externas con el fin de gestionar los datos medidos y controlar el comportamiento de los mismos de forma remota. Además, se requiere que la información medida pueda ser entregada de manera fiable y segura y que los valores obtenidos puedan ser útiles en la toma de decisiones sobre el ambiente dado [1] [2].

Dentro de lo anterior, están los espacios inteligentes los cuales son los lugares físicos como un hogar o un área de trabajo que están compuestos para un conjunto de sensores y elementos computacionales interactuando entre sí que tienen la capacidad de adaptarse a las preferencias del usuario y llevar un control de automatización de los servicios del entorno. Por ejemplo, se pueden automatizar actividades como el control de energía eléctrica, el control ambiental y el control de acceso a un espacio físico con el fin de brindar comodidad y seguridad al usuario [3].

Los componentes de los espacios inteligentes se pueden distribuir geográficamente y comunicarse a través de Internet por medio de protocolos de comunicación capturando información de las variables ambientales para así ajustar el entorno de acuerdo a las preferencias del usuario.

Es por eso, que para llevar un control de un espacio inteligente se propone realizar un componente de software que permita administrar la información capturada por una red de sensores para regular el ambiente de manera inteligente en dicho espacio.

2 Antecedentes

2.1 Proyectos de Integración o terminales

2.1.1 Aplicación concurrente para la obtención y almacenamiento de datos a partir de sensores [3].

El objetivo de este proyecto terminal es el monitoreo de un espacio inteligente en el cual se pretende monitorear la temperatura de diversos sensores. Los parámetros a medir son evento de la detección de puerta abierta en una habitación, humedad, luminosidad y presencia de un objeto, toda la información adquirida mediante el monitoreo se almacenará en una ontología a través de servicios web con el fin de realizar pruebas de funcionamiento, el proyecto mencionado y la propuesta presente tienen un objetivo en común cuatro

variables a medir, luminosidad, temperatura, humedad y presencia, la diferencia es que la información obtenida de este proyecto se almacenará en una base de datos para utilizarla en la toma de decisiones.

2.1.2 Red de monitoreo remoto inalámbrico de sensores con dispositivos ZIGBEE [4].

El objetivo de este proyecto terminal se basa en el monitoreo remoto para campos de temperatura y humedad. El sistema que propone este proyecto terminal está constituido por sensores con dispositivos ZigBee los cuales servirán para que cada nodo de la red pueda tener comunicación de manera inalámbrica y de este modo se podrá monitorear y almacenar la información de temperatura y humedad. El proyecto mencionado y la propuesta planteada comparten un objetivo en común que es el almacenamiento de datos de la temperatura y la humedad las cuales son las variables en común y se hará mediante un protocolo de comunicación. La diferencia es que las tecnologías a utilizar para lograr el desarrollo de esta propuesta son distintas.

2.2 Tesis

2.2.1 Red de sensores para el control de contextos usando contextos web [5].

El objetivo principal de este trabajo es demostrar la adquisición de información a través de sensores y de un protocolo de comunicación para que de esta forma pueda brindar servicios adecuados a los usuarios dependiendo las necesidades que cada uno presente. La diferencia del trabajo presente es que se va a tener un control de información llevándolos a una base de datos para su posterior análisis y la toma de decisiones en un ambiente dado.

2.2.2 Estudio de la plataforma de Android para dispositivos móviles y desarrollo de aplicación para la administración de redes de sensores [6].

Este proyecto de tesis propone desarrollar un prototipo de una aplicación que administre redes de sensores inalámbricos, la información se obtendrá mediante un monitoreo, se almacenará en una base de datos y se podrá obtener la información por medio de internet mediante coordenadas geográficas para poder acceder a ellos y poder monitorear diferentes parámetros. Esta tesis tiene un objetivo similar a la presente propuesta ya que se llevará una administración de información obtenida por los sensores mediante un gestor de base de datos como lo es MySQL. La diferencia es que en esta propuesta habrá un módulo de toma de decisiones que determinará si el valor de alguna de las variables se está excediendo.

2.3 Artículos

2.3.1 Control Automático de Condiciones Ambientales en Domótica usando Redes Neuronales Artificiales [7].

El objetivo fundamental de este artículo es la adquisición de información en un entorno como lo es un hogar o una oficina llevando el control de automatización en donde se determina el control de la luminosidad, temperatura, humedad entre otros valores. A diferencia con el proyecto a desarrollar es que la información del espacio inteligente se almacenará en un gestor de base de datos a través de un protocolo de comunicación y mediante inferencias mediante el módulo de toma de decisiones.

2.3.2 Sistemas de sensores inalámbricos para la implementación de espacios inteligentes [8].

El objetivo de este artículo es optimizar el uso y el ahorro de energía eléctrica de un espacio inteligente, mediante un sistema de sensores inteligentes basado en comunicación inalámbrica. De la misma manera el sistema ayudará en el aprovechamiento de ahorro de energía eléctrica, optimizando el uso de los sistemas de iluminación. En general, puede ser utilizado para el monitoreo y supervisión de las variables físicas ambientales con las que cuenta dicho espacio. Este artículo es similar a esta propuesta ya que se obtendrá información de variables físicas con las que cuenta un espacio inteligente mediante sensores, pero con la diferencia que la información obtenida se almacenará en un gestor de base de datos.

3 Justificación

El entorno inteligente se desarrollará para que este se adapte de manera automática con el fin de mejorar la calidad de este y para que brinde satisfacción al usuario sustituyendo el trabajo peligroso, trabajo físico y tareas repetitivas sin necesidad de perder tiempo realizando este tipo de actividades. Teniendo una red de sensores que midan variables como la temperatura, luminosidad, presencia y humedad de un entorno se crea la necesidad de administrar la información obtenida por los sensores para que después se puedan extraer los datos de los parámetros de cada una de las variables ambientales y realizar tomas de decisiones con el fin de automatizar los dispositivos existentes de un entorno.

4 Objetivos

4.1 Objetivo General

Diseñar e implementar un protocolo de comunicación para el registro de eventos y toma de decisiones en un entorno inteligente.

4.2 Objetivos Particulares

- Modelar las entidades participantes en el proyecto para almacenar los datos relacionados con las variables ambientales en un espacio inteligente.

- Diseñar e implementar un módulo para la toma de decisiones en un espacio inteligente.
- Implementar un protocolo de aplicación para la comunicación de una red de sensores orientada a espacios inteligentes.

5 Marco teórico

5.1 Entornos inteligentes

Los entornos inteligentes son espacios físicos como hogares, lugares de trabajo, un edificio entre otros, en donde existen sistemas de sensores y elementos computacionales capaces de adquirir información para que de este modo pueda llevar un control de dicho espacio y de esta manera obtener la comodidad del usuario. Los entornos inteligentes cuentan con un protocolo de comunicación ya que este nos permite la interacción de un sistema de sensores y elementos computacionales con el fin de controlar variables ambientales como lo es la detección de presencia, la temperatura/humedad y luminosidad con el propósito de que la tecnología interactúe con el usuario de una manera transparente. Para que este objetivo se lleve a cabo, es necesario desarrollar un protocolo de comunicación con el fin de recuperar información y de este modo poder detectar eventos importantes por medio de los sensores y con la ayuda de un gestor de base de datos poder almacenar los eventos sucedidos.

5.2 Administración de la información

En la actualidad, el crecimiento acelerado de la información dentro de las organizaciones ha provocado que sea complejo gestionar toda esa información en un entorno o espacio. Es por ello que se debe gestionar, procesar y recabar información para transformarlo en conocimiento.

De acuerdo con Escamilla “la información es parte fundamental para que los individuos de una organización alcancen un desarrollo óptimo, esto a través del apoyo que aporta en el proceso de toma de decisiones sustentadas en la adquisición de nuevos conocimientos” [14].

Se distinguen tres etapas para llegar a tomar una decisión: recuperación de la información, análisis de la información y toma de decisiones.

5.3 Sockets

Un socket es un punto de acceso a los servicios de comunicación entre procesos. Cada Socket tiene asociada una dirección que lo identifica. Conociendo esta última, se puede

establecer una comunicación con un socket para que actúe como extremo de un canal bidireccional.

Los sockets son el método más utilizado para la comunicación entre procesos, los cuales se ejecutan en una misma computadora o en diferentes computadoras conectadas a través de una red.

5.3.1 Arquitectura de Cliente-Servidor

Consiste básicamente en un cliente que realiza peticiones a otro programa (el servidor) que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras (figura 1).



Figura 1. Arquitectura Cliente Servidor

5.3.2 Socket Servidor

Para que un socket pueda tener comunicación es necesario asignarle un número de puerto y dirección única para que de este modo los sockets clientes puedan tener comunicación. Teniendo en cuenta que algunos puertos no se pueden utilizar porque están reservados por otros servicios estándares.

5.3.3 Socket Cliente

El cliente debe de tener una dirección para poder tener comunicación con el servidor para que de este modo poder solicitar información.

5.4 Concurrencia

Hace referencia a la propiedad que tiene los sistemas de cómputo donde sus procesos se pueden ejecutar de manera simultánea.

Dentro de la programación hace referencia en donde se implementa la programación multihilo, un programa multihilo contiene secciones de instrucciones que se pueden ejecutar de forma simultánea.

Los hilos a menudo son conocidos como procesos ligeros o subprocessos. Un hilo es similar a un proceso con la diferencia de que un hilo se ejecuta dentro del contexto de un programa.

5.5 Sensor

Un sensor o captador, como prefiera llamársele, no es más que un dispositivo diseñado para recibir información de una magnitud del exterior y transformarla en otra magnitud, normalmente eléctrica, que seamos capaces de cuantificar y manipular.

5.6 Netbeans

Es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto libre y gratuito sin restricciones de uso. La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (**manifest file**) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos (figura 2).

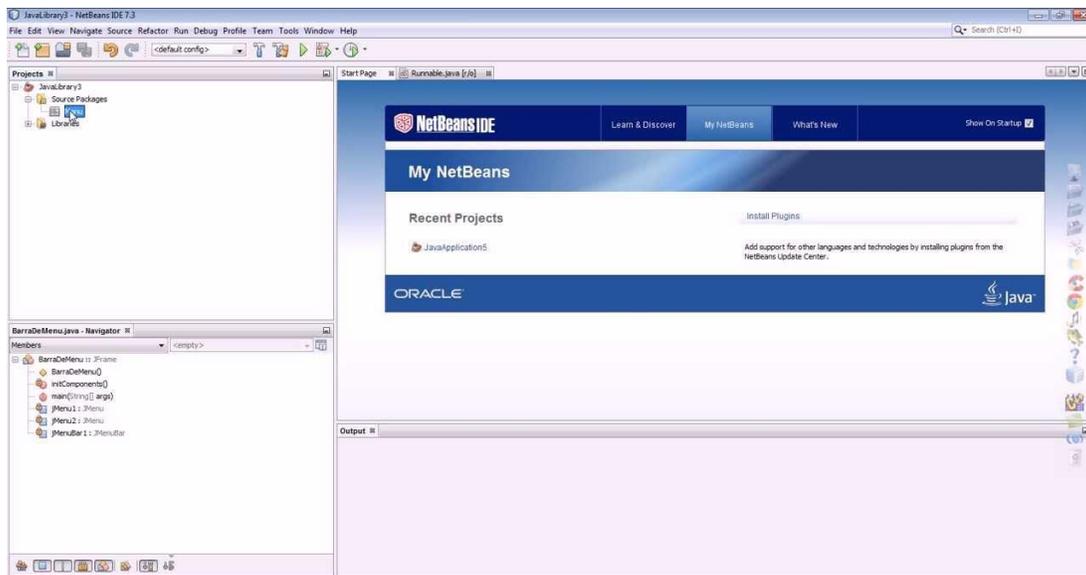


Figura 2. Interfaz de desarrollo de Netbeans

5.6.1 Lenguaje de programación Java.

Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no

funcionarán a menos que tenga Java instalado y cada día se crean más. Java es rápido, seguro y fiable. Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes.

5.7 MySQL

Es un sistema de administración de bases de datos relacionales de código abierto, basado en el lenguaje de consulta estructurado.

6 Desarrollo del proyecto

En esta sección se describe cómo fue implementado cada uno de los módulos de los que se compone el proyecto los cuales son encargados de la administración de información de variables ambientales de un espacio inteligente basándose en la toma de decisiones.

6.1 Tecnologías usadas

Para la elaboración del proyecto terminal se utilizó:

- MySQL WorkBench es una herramienta para diseñar e implementar bases de datos y el cual es el entorno para la creación de la base de datos a utilizar en este proyecto.
- Java es un lenguaje de programación que se utilizó para hacer la conexión del servidor con la base de datos, además fue utilizado para realizar la comunicación entre el cliente y servidor y para poder desarrollar el módulo de toma de decisiones.
- Sistema operativo Raspbian es una distribución libre de Linux basado en Debian para la manipulación de la placa Raspberry PI.

6.2 Implementación

Este presente proyecto se desarrolló por tres módulos de tres módulos los cuales se presentan en la figura 3.

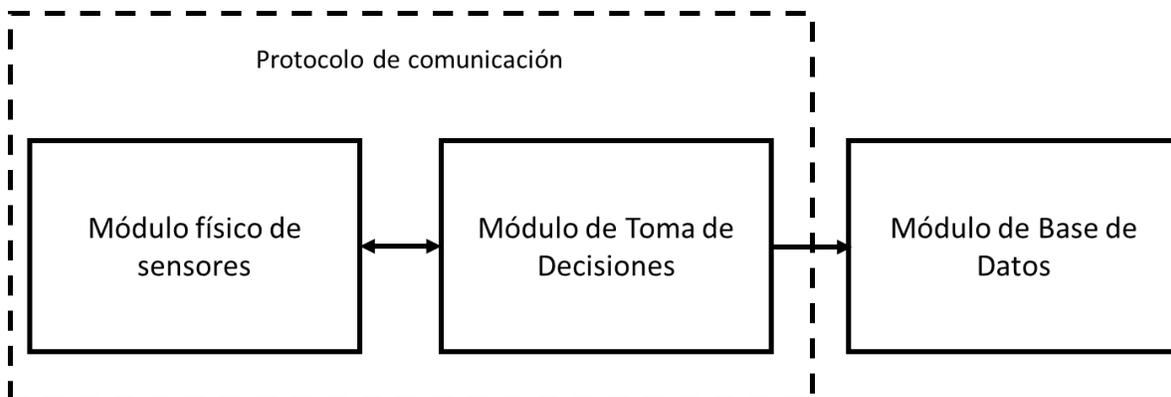


Figura 3. Módulo del proyecto

En la siguiente sección se describe la estructura de los paquetes así como el flujo de mensajes que se envían en cada uno de los módulos.

En la tabla 1 se define el paquete para la comunicación entre un NS y un NC.

Cabecera	Temperatura	Humedad	Luminosidad	Presencia
01	T	H	L	P

Tabla 1. Paquete-Inf. Contiene los valores leídos de los sensores. Considerar que no hay presencia.

Los valores que contiene el Paquete-Inf van separados entre comas, un ejemplo se muestra en la tabla 2:

Paquete-Inf
22,0.4,250,0

Tabla 2. Ejemplo de un Paquete-Inf

En el ejemplo podemos observar que:

Temperatura (T) = 22 °C

Humedad (H) = 0.4

Luminosidad (L) = 250

Presencia (P) = 0

El Paquete-Inf se envía cada cierto tiempo desde el NS al NC. Se hace el envío para conocer el estado del ambiente. El envío se realiza mediante WiFi.

En la tabla 3 se muestra en Paquete-Event el cual se envía desde el NC a la base de datos cada que ocurre un evento de interés.

Cabecera	tipo_evento	Variación	Cantidad_variación	tiempo_evento	IdEstacio
001	01-Temperatura 02-Luminosidad 03-Humedad 04-Presencia	0-Aumentar 1-Disminuir	C	T	E

Tabla 3. Paquete-Event. Contiene el tipo de evento generado

La Figura 4 ilustra la comunicación entre los dispositivos participantes en un espacio inteligente utilizando los paquetes definidos anteriormente. Todo paquete de tipo 01 siempre se almacena en la base de datos y un paquete 001 se almacena siempre y cuando exista una variación del valor actual con respecto al valor anterior. Por el momento no se consideran los mensajes de error ni mensajes de confirmación al momento de efectuar una acción.

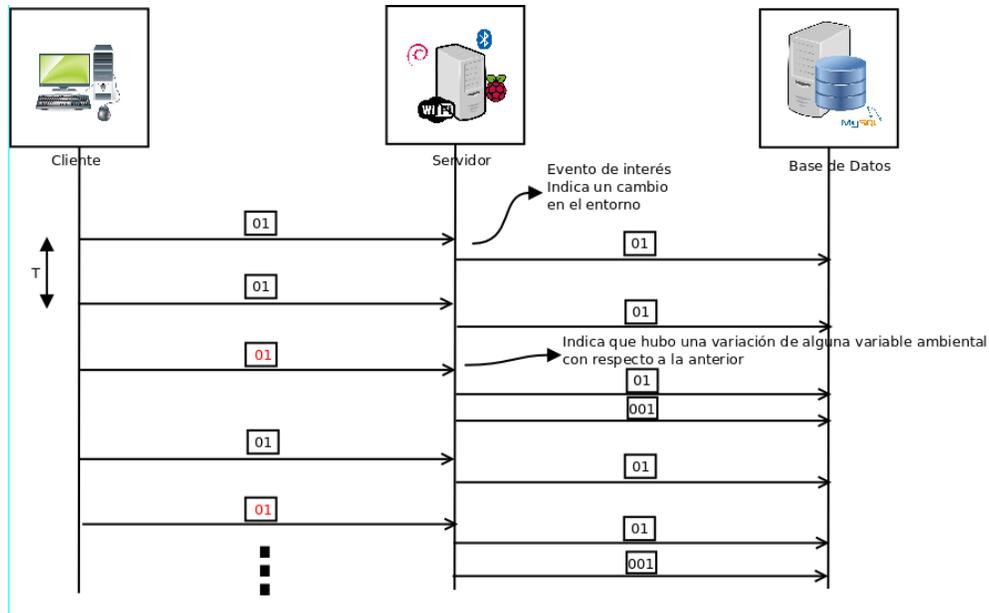


Figura 4. Diagrama de comunicación entre los dispositivos participantes del entorno inteligente

6.2.1 Módulo físico de sensores.

En este módulo existen tres tipos de nodos en un espacio: nodo sensor (NS), nodo identificador (NI) y nodo concentrador (NC).

- Un NS es una placa Arduino con módulos WiFi y Bluetooth, y sensores de temperatura/humedad y luz, este nodo genera eventos los cuales son reportados al NC.
- Un NI es una placa Arduino con módulos WiFi, Bluetooth y NFC y que de la misma manera que el NS genera eventos que son reportados al NC cada determinado tiempo.
- Un NC es una placa Raspberry Pi 3. Este NC debe ser capaz de almacenar eventos y asociarlos al NI o NS del que proviene. Posterior al almacenamiento, el NC debe tomar una decisión: no hacer nada, responder al NS o NI del que provino el evento o notificar a otra entidad (denominada servidor) del evento ocurrido.

Entonces el módulo de sensores es el encargado de la adquisición de los datos relacionados al ambiente mediante sensores que estarán comunicados por medio de un protocolo de comunicación. Cada "T" segundos obtiene los valores de las variables relacionadas al ambiente y las difunde por un canal de comunicación inalámbrico. Dichos datos serán de utilidad para evaluar el entorno, tomar una decisión en base a las características del lugar y tomar una acción para ajustar el ambiente.

Cada tabla implementada en el diagrama cuenta con una funcionalidad como se muestra en la siguiente descripción **Tabla 4**.

Tabla	Descripción
Espacio	Está destinada para almacenar los espacios del departamento de sistemas.
Espacio profesor	Está destinada para relacionar os espacios existentes con los profesores-
Profesores	Almacena a los profesores son parte de la comunidad de la universidad Autónoma Metropolitana.
División	Son las áreas de las que se compone la universidad.
Departamento	Son las áreas de las que se compone una división.
Nodo sensor	Almacena los valores de la variable ambientales: temperatura, humedad, luminosidad y presencia.
Nodo Identificador	Almacena los identificadores del WI-FI bluetooth.
Nodo Concentrador	Es la placa de Rapsberry el cual es el encargado de almacenar la identificación de cada uno de los arduinos los cuales están asociados a un espacio.
Eventos	Es el encarado de almacenar los eventos provenientes de cada uno de los sensores en base al módulo de toma de decisiones.

Tabla 4. Modelo de la base de datos

Con el diagrama anterior de entidad-relación (Figura 5) se realizó *Reverse Engineer* (Ingeniería inversa) para la creación de la base de datos en MySQL.

La base de datos se le determinó como nombre **proyecto** el cual contiene las tablas y atributos mencionados anteriormente.



Figura 6. Base de datos en MySQL

6.2.3 Módulo de Toma de decisiones

Es el encargado de determinar la difusión de los datos adquiridos por el módulo físico de sensores y de la ejecución de una acción para regular el ambiente. Para ello, el módulo está a la espera de las condiciones actuales del ambiente y las compara con el histórico anterior para determinar un cambio radical en este y así disparar una acción para regular el mismo.

Para poder implementar el módulo de toma de decisiones se realizó un servidor que fuera capaz de servir múltiples clientes de manera concurrente y que siempre estuviera prendido. Esto se logró a través de la implementación de un servidor con hilos en java.

La comunicación entre el cliente y el servidor se realizó mediante sockets utilizando el puerto 8000 de manera local para que de esta forma se enlace una comunicación entre ellos y de esta manera puedan interactuar intercambiando mensajes que se encuentran en la tabla 1, 2 y 3.

Además, dentro del servidor se creó una clase llamada ConexiónBase con el fin de que el servidor pudiera comunicarse con la base de datos y de esta manera se pudieran almacenar los eventos generados por NI y NS.

Para realizar dicha comunicación del cliente y servidor se realizó lo siguiente:

- Se hizo un proyecto (ClienteEspacio), dentro de este contiene un paquete (clienteconhilos) en el cual se encuentra una clase (Hilo) la cual hereda Thread y esto hace que se convierta en un hilo y se pudiera ejecutar de manera concurrente con el método run().
- Posteriormente dentro de un try- catch se controló todo tipo de errores durante la ejecución, cabe mencionar que dentro del try se codificaron todas las sentencias necesarias para la conectividad mientras en el catch se ejecutan los mensajes si estas producen algún error.
- En el proyecto (ServidorEspacio) se creó una clase (ServidorEcoconHilos) la cual contiene la conexión con el Cliente de manera local utilizando el puerto 8000, además existen los siguientes paquetes:
 - com.uam.espacioi.datos. Contiene las clases para la conectividad con la base de datos, así como los métodos para la persistencia de datos.
 - com.uam.espacioi.dto. Contiene el mapeo de las entidades de la base de datos que se utilizan en el proyecto, los cuales son: NodoSensor y Evento.
 - com.uam.espacioi.negocio. Contiene una clase en donde se realizan las operaciones necesarias para la toma de decisiones de los datos generados por NI y NS.

En la figura 7 se muestra el Diagrama de Clases del proyecto y el código fuente para la toma de decisiones se muestra en el apéndice 11.2.

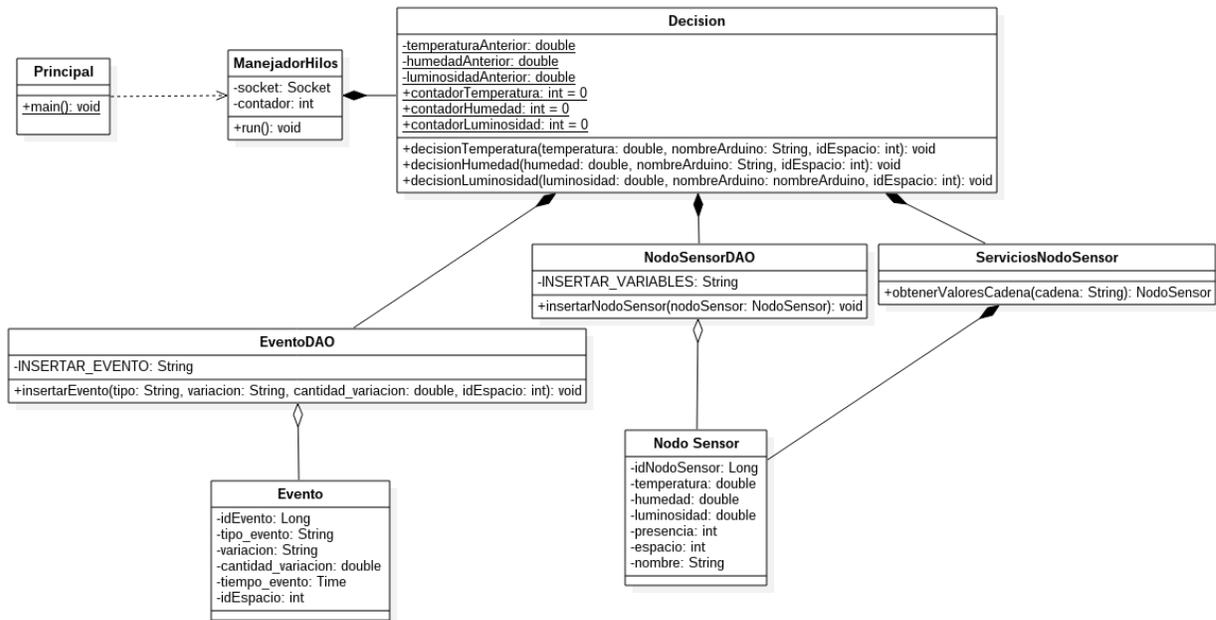


Figura 7. Diagrama de clases de Servidor

6.2.3.1 Toma de decisiones de las variables ambientales.

Es el módulo encargado de determinar la difusión de los datos adquiridos por el módulo físico de sensores y de la ejecución de una acción para regular el ambiente. Para ello, el módulo está a la espera de las condiciones actuales del ambiente y las compara con el histórico anterior para determinar un cambio radical en este y así disparar una acción para regular el mismo.

De acuerdo a las normas de la Secretaría del Trabajo y Previsión Social (STPS) se muestra en la siguiente tabla información de los rangos mínimos y máximos de las variables ambientales para centros de trabajo, tales como oficinas, cuartos de control, centros de cómputo y laboratorios, entre otros, en los que se disponga de ventilación artificial para confort de los trabajadores o por requerimientos de la actividad en el centro de trabajo, se recomienda tomar en consideración la humedad, la temperatura y luminosidad de preferencia en los términos siguientes:

Variables ambientales	Mínimo	Máximo
Temperatura	15 °C	30 °C
Humedad	20%	60%
Luminosidad	300 (Luxes)	400 (Luxes)

Tabla 5. Variables ambientales de acuerdo a las normas de la Secretaría del Trabajo y Previsión Social

Considerando los parámetros de la tabla 2 se determinó una variación específica para cada variable ambiental, considerando 7 intervalos entre los rangos mínimos y máximos:

Por lo tanto para obtener la variación de la temperatura, humedad y luminosidad se consideró la siguiente fórmula:

$$v = \frac{va_{max} - va_{min}}{7}$$

Donde:

v= variación

va=variable ambiental (temperatura, humedad o luminosidad).

Sustituyendo los valores mencionados en la tabla 2 por cada variable ambiental se obtuvieron los siguientes resultados:

- Para la temperatura:

$$v = \frac{30^{\circ}C - 15^{\circ}C}{7} = 2.14^{\circ}C$$

- Para la humedad:

$$v = \frac{60\% - 20\%}{7} = 5.71\%$$

- Para la Luminosidad:

$$v = \frac{400 Lux - 300 Lux}{7} = 14.28 Lux$$

En la tabla 6 se muestran los resultados de la variación de cada una de las variables ambientales.

VARIABLES AMBIENTALES	VARIACIÓN
Temperatura	2.14 °C
Humedad	5.71 %
Luminosidad	14.28 (Luxes)

Tabla 6. Rangos a consideración cambio de variable ambiental

- **Toma de decisión Temperatura.**

Para realizar la toma de decisión de la variable ambiental temperatura se consideró lo siguiente:

- Si la temperatura es menor a 15 grados, es decir, no se encuentra dentro de los rangos definidos en la tabla 2 se mandará un mensaje al nodo final.
- Si la temperatura se sobrepasa a los 30 grados es decir no se encuentra dentro de los rangos definidos se mandará un mensaje al nodo final.

Dentro de este módulo de toma de decisiones también se consideró que una variación de temperatura podría ser algún dato como los siguientes:

- Si la temperatura anterior con respecto a la temperatura actual hay una diferencia de 2.14°C o más se considera una variación importante por lo tanto se almacenará en la base de datos dentro de la tabla eventos de los contrario no se hará nada al respecto.
- **Toma de decisión Humedad.**

Consideración de toma de decisiones para la variable ambiental de humedad:

- Si la humedad es menor al 40% o superior al 60% es decir, no se encuentra en los rangos definidos, se mandara un mensaje al nodo final.

Dentro de este módulo de toma de decisiones también se consideró que una variación de humedad podría ser algún dato como los siguientes:

- Si la humedad es menor o inferior a 0.053 % a la humedad actual ese dato se va almacenar en la tabla eventos de la base de datos ya que se considera una variación importante, de lo contrario no se considera variación y no se almacena en la tabla eventos.
- **Toma de decisión Luminosidad.**

Para la toma de decisión de la variable ambiental de luminosidad se consideró lo siguiente:

- Si la luminosidad es menor o superior a 300 Luxes es decir no se encuentra dentro de definidos se mandará un mensaje al nodo final.

Dentro de este módulo de toma de decisiones también se consideró que una variación de luminosidad podría ser algún dato como los siguientes:

- Si la luminosidad es menor o superior a 13.33 luxes a la luminosidad actual ese dato se va almacenar en la tabla eventos de la base de datos proyecto ya que se considera una variación importante, de lo contrario no se considera variación y no se almacena en la tabla eventos.
- **Toma de decisión Presencia**

Por último, para detectar si en el espacio había presencia había dos consideraciones las cuales son las siguientes:

- Si es el espacio existía presencial de algún individuo el evento que se consideraba como un 1 y de la misma manera se almacenaba en la base de datos en la tabla eventos.
- De lo contrario si en el espacio no existía presencial el evento que se consideraba como un 0 y de la misma manera se almacenaba en la base de datos en la tabla eventos.

Para realizar la toma de decisiones de la temperatura, humedad y luminosidad fue necesario declarar una variable inicializada en 0 por cada una de las variables ambientales para verificar que sea el primer valor que se inserta en la base de datos y llevar un conteo de cuantas variables ambientales se han almacenado en la tabla eventos.

En el entorno de desarrollo de Netbeans se implementó la clase cliente y la clase servidor en donde se simularon 45 clientes y se observó que el servidor no soporta más de esta cantidad (figura 8).

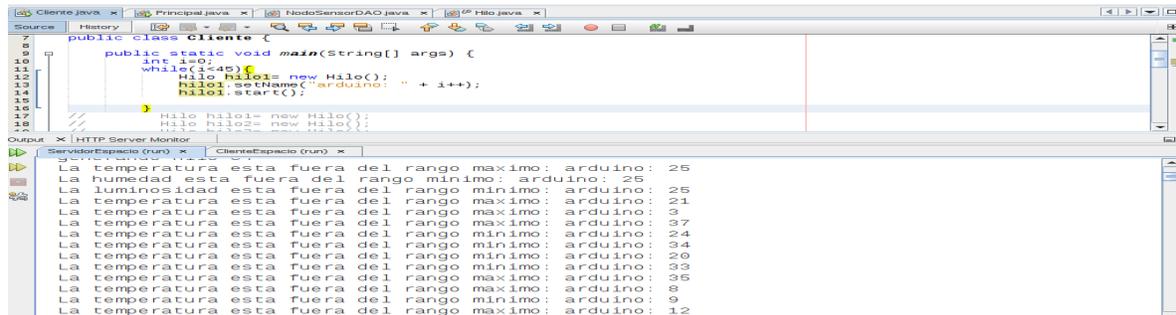


Figura 8.Simulación de 45 Clientes

Para la simulación de los datos de las variables ambientales las cuales se almacenan en la tabla NodoSensor y Eventos de las base de datos, se realizó un método en donde se generan números aleatorios. Ver figura 9.

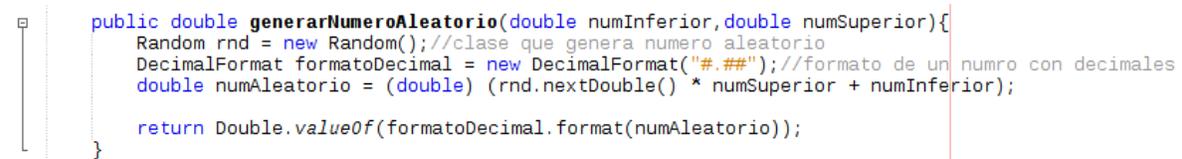


Figura 9. Clase para crear números aleatorios

Además, cada cliente envía 10 datos cada 10 segundos por lo tanto se registraran 450 datos insertados NodoSensor.

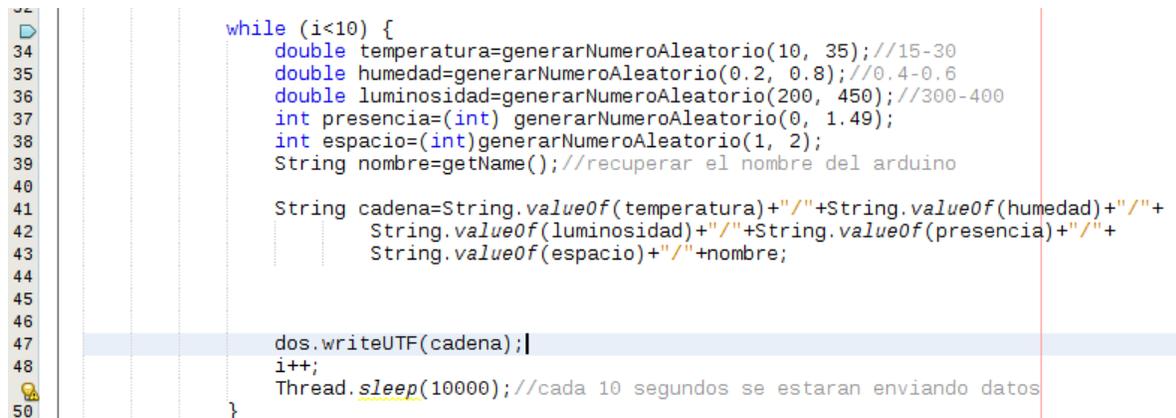


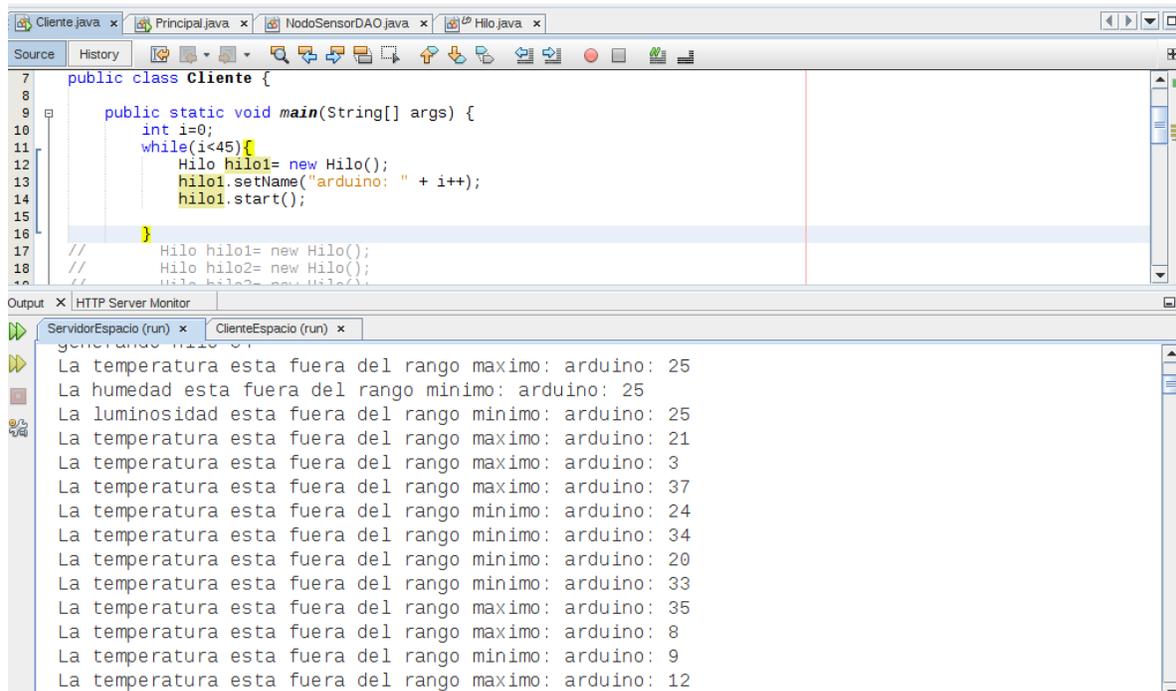
Figura 10.Tiempo de envío de datos

7 Resultados

Este proyecto fue simulado en una memoria portable con 16 GB de almacenamiento en el cual se encontraba instalado el sistema operativo Raspbian. La computadora donde se realizaron las pruebas tenía las siguientes características

- Laptop Samsung NP300E4C-A0KMX:
- Procesador Intel Celeron B820 (1.7 GHz),
- Memoria de 2 GB DDR3
- Disco Duro de 320 GB
- Pantalla HD LED de 14"
- Video Intel HD Graphics

Al momento de ejecutar el servidor y el cliente en la parte inferior de la figura 11 se muestran los datos que se están mandando de manera aleatorio de las variables ambientales, estas variables se almacenan en la base de datos Nodo Sensor y si existe alguna variación de alguna variable ambiental se van almacenando en la tabla eventos de la base de datos del proyecto.



The screenshot shows an IDE with several tabs: 'Cliente.java', 'Principal.java', 'NodoSensorDAO.java', and 'Hilo.java'. The 'Cliente.java' tab is active, displaying the following code:

```
7 public class Cliente {
8
9     public static void main(String[] args) {
10         int i=0;
11         while(i<45){
12             Hilo hilo1= new Hilo();
13             hilo1.setName("arduino: " + i++);
14             hilo1.start();
15         }
16     }
17     // Hilo hilo1= new Hilo();
18     // Hilo hilo2= new Hilo();
19     // Hilo hilo3= new Hilo();
20 }
```

The 'Output' window at the bottom shows the following text:

```
ServerEspacio (run) x ClienteEspacio (run) x
generando hilo 01
La temperatura esta fuera del rango maximo: arduino: 25
La humedad esta fuera del rango minimo: arduino: 25
La luminosidad esta fuera del rango minimo: arduino: 25
La temperatura esta fuera del rango maximo: arduino: 21
La temperatura esta fuera del rango maximo: arduino: 3
La temperatura esta fuera del rango maximo: arduino: 37
La temperatura esta fuera del rango minimo: arduino: 24
La temperatura esta fuera del rango minimo: arduino: 34
La temperatura esta fuera del rango minimo: arduino: 20
La temperatura esta fuera del rango minimo: arduino: 33
La temperatura esta fuera del rango maximo: arduino: 35
La temperatura esta fuera del rango maximo: arduino: 8
La temperatura esta fuera del rango minimo: arduino: 9
La temperatura esta fuera del rango maximo: arduino: 12
```

Figura 11. Datos generados aleatoriamente

Los datos generados que se muestran en la figura 11 se almacenaron en la tabla Nodosensor la cual almacena los valores de las variables ambientales el número de Arduino o cliente del cual proviene la información y el espacio en el cual ocurren estos eventos figura 12.

Query 1 x | Nodo_sensor x | Eventos x | Nodo_sensor x | Eventos x

Limit to 1000 rows

```
1 • SELECT * FROM proyecto.Nodo_sensor;
```

2

#	idNodo_sensor	temperatura	humedad	presencia	arduino	Espacio_idEspacio
1	1	33.48	0.33	0	arduino: 25	2
2	2	25.85	0.33	0	arduino: 26	1
3	3	14.16	0.37	1	arduino: 33	1
4	4	35.66	0.88	1	arduino: 6	2
5	5	25.97	0.91	1	arduino: 31	2
6	6	31.43	0.96	1	arduino: 22	1
7	7	15.73	0.48	0	arduino: 1	2

Nodo_sensor 3 x

Action Output

	Time	Action	Message	Dur	
✓	1	17:21:03	SELECT * FROM proyecto.Nodo_sensor LIMIT 0, 1000	450 row(s) returned	0.0

Figura 12. Datos aleatorios almacenados en la tabla nodo sensor

Como se muestra en la siguiente figura 13 en donde se ejecutaron 45 clientes (conexiones) solo fueron capturados 1218 eventos los cuales se consideran cambios de las variables ambientales y se almacenan en la tabla eventos de la base de datos.

Los eventos almacenados se consideran una variación fuera del rango (Ver tabla 3) de alguna variables ambiental.

Limit to 50000 rows

```
1 • SELECT * FROM proyecto.Eventos;
```

2

#	idEventos	tipo_evento	variacion	cantidad_variacion	tiempo_evento	idEspacio
1	1	temperat...	Disminuyó	19.32	16:25:22	1
2	2	temperat...	Aumentó	2.18	16:25:22	2
3	3	temperat...	Disminuyó	18.7	16:25:22	2
4	4	temperat...	Aumentó	4.86	16:25:22	2
5	5	temperat...	Disminuyó	22.61	16:25:22	1
6	6	temperat...	Disminuyó	11.86	16:25:22	2
7	7	temperat...	Disminuyó	20.28	16:25:22	2

Eventos 4 x

Action Output

	Time	Action	Message	Dura	
✓	1	17:47:10	SELECT * FROM proyecto.Eventos LIMIT 0, 50000	1218 row(s) returned	0.00

Figura 13. Cambio de las variables ambientales durante la ejecución

Para la temperatura hubo un total de 405 variaciones tal como se muestra en la figura 14.

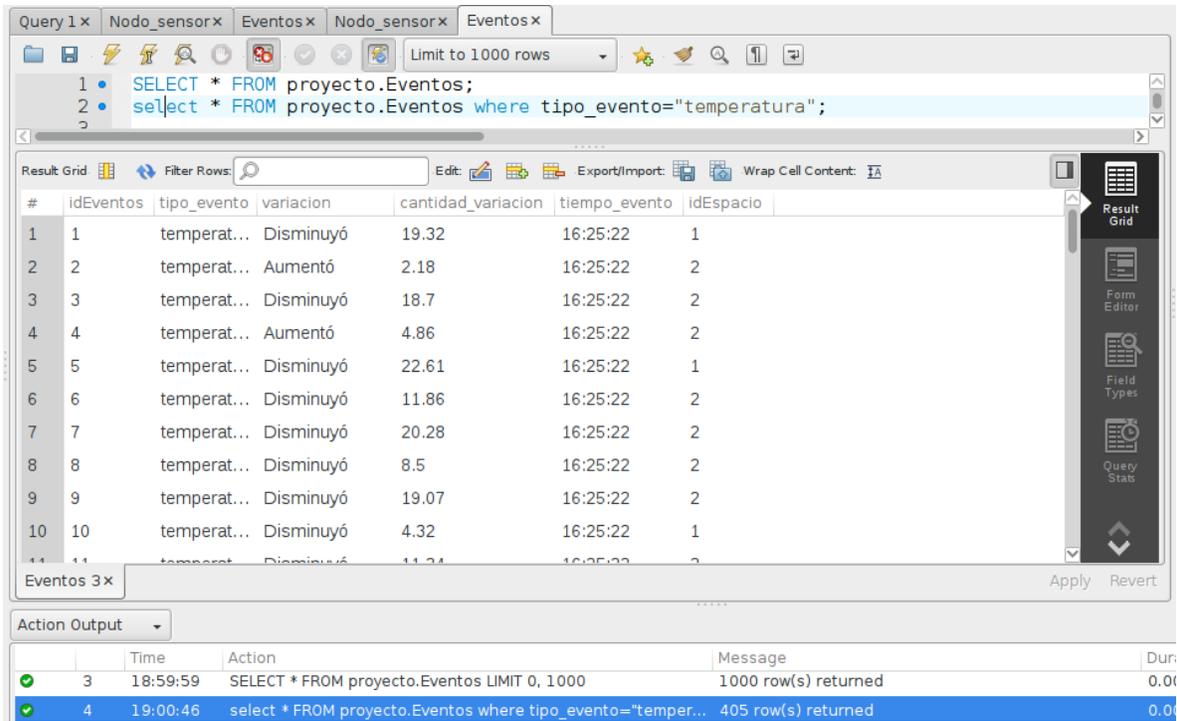


Figura 14. Variación de la temperatura

Para la humedad hubo un total 391 variaciones tal como se muestra en la figura 15.

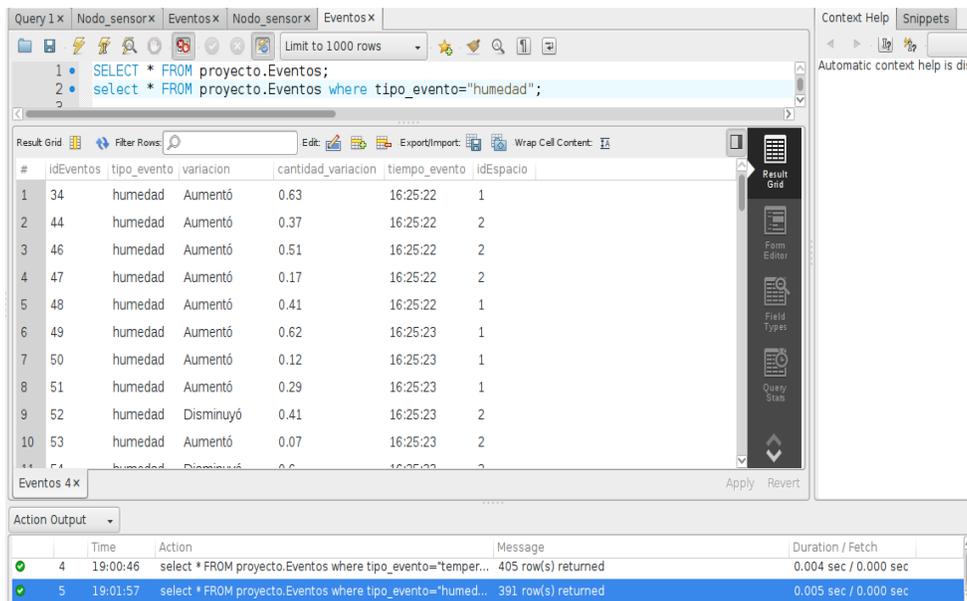


Figura 15. Variación de la humedad

Para la luminosidad hubo un total de 422 variaciones tal como se muestra en la figura 16.

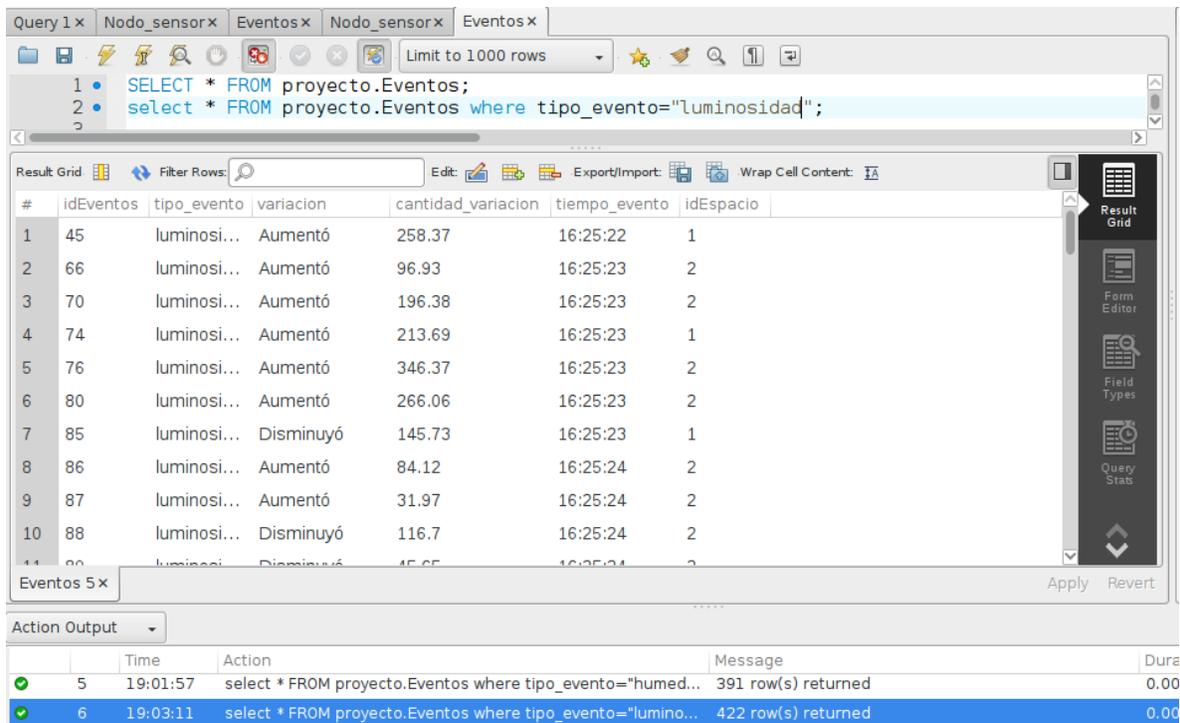


Figura 16. Variación de la luminosidad

Finalmente para la presencia también es generada por un número aleatorio ya sea 1 que significa presencia o 0 que no existe presencia en un espacio. Durante la simulación de los 45 clientes (conexiones) se detectaron 143 presencia en un espacio figura 17.

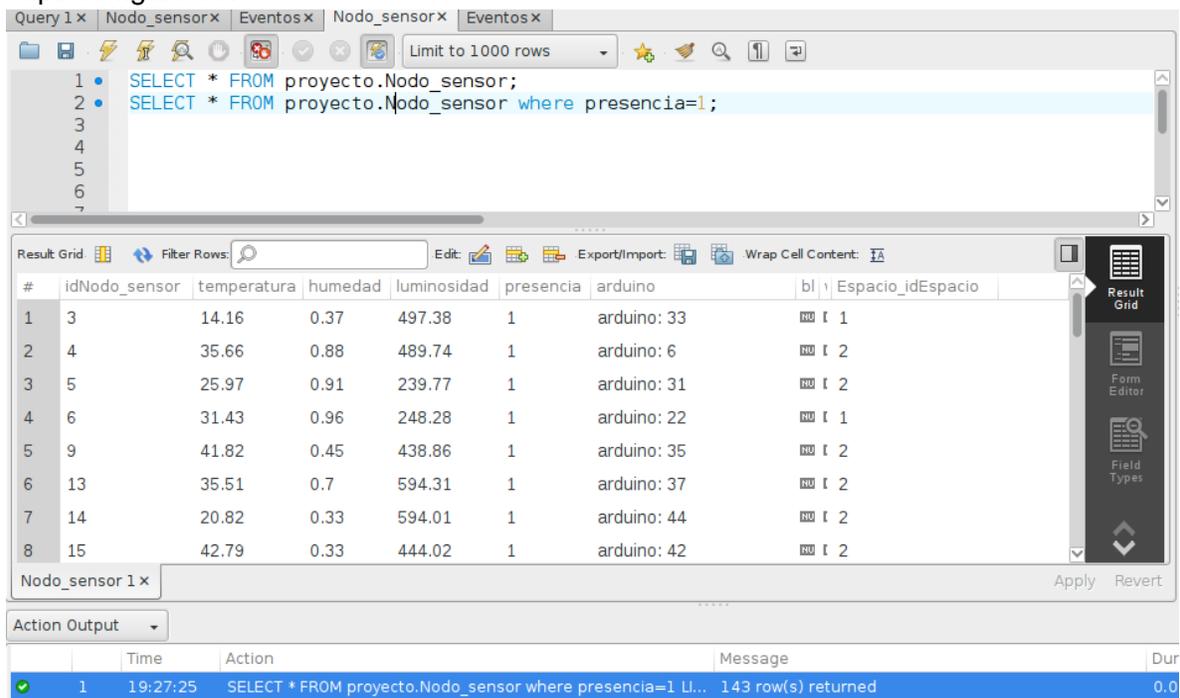
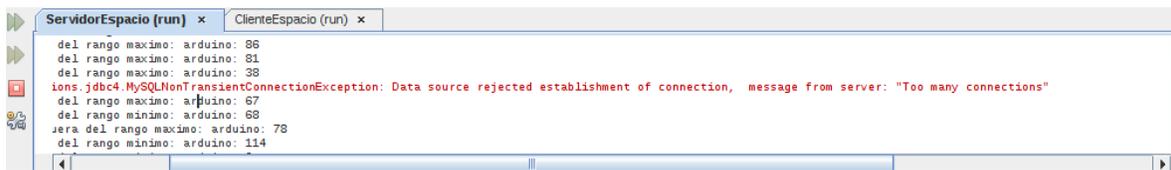


Figura 17. Variación de la presencia

8 Análisis y discusión de resultados

Al hacer pruebas en un escenario simulado en donde se ejecutaron distintos números de clientes, se obtuvo que el máximo número de clientes que soportaba el servidor eran 45, al momento de ejecutar más clientes el servidor mostraba mensajes de error donde decía que había muchos clientes y no podía insertar información en la base de datos por lo mismo, tal como se muestra en la figura 18.

Las pruebas que se realizaron en la sección anterior se ejecutaron de forma correcta porque, a pesar de que es un escenario simulado, se pudo obtener todos los datos que un sensor enviaría tal como el tipo de evento, la hora, el lugar, todo el evento.



```
ServidorEspacio [run] x ClienteEspacio (run) x
del rango maximo: arduino: 86
del rango maximo: arduino: 81
del rango maximo: arduino: 38
ions.jdbc4.MySQLNonTransientConnectionException: Data source rejected establishment of connection, message from server: "Too many connections"
del rango maximo: arduino: 67
del rango minimo: arduino: 68
Jera del rango maximo: arduino: 78
del rango minimo: arduino: 114
```

Figura 18. Mensaje de error al ejecutar más de 45 clientes

De todas las pruebas realizadas todas salieron bien excepto una (Ver figura 18). Cabe señalar que esto pasó porque el servidor se vuelve inoperable debido a las limitaciones del equipo ya que el sistema operativo se está ejecutando desde una memoria portable de 16 GB con 2 GB de memoria RAM y cuando múltiples clientes quieren enviar información cada uno consume un poco de memoria RAM lo que ocasiona que se empiece a alentar el servidor y algunos datos se guarden y otros no.

De acuerdo a la figura 13 en donde se muestra la información obtenida por los sensores la cual es almacenada correctamente, se puede observar que en la tabla eventos se guarda las variaciones que existieron cuando se fueron leyendo cada una de las variables ambientales, los datos almacenados pueden variar dependiendo de que si hubo o no variación pero para las pruebas existieron una gran número de variaciones debido a que los valores que se enviaban eran aleatorios.

9 Conclusiones

En este trabajo se puede mostrar que la tecnología computacional y la electrónica pueden facilitar las actividades cotidianas de los humanos, una aplicación clara sería un espacio inteligente el cual se desarrolló en este trabajo terminal.

Con la ayuda de un protocolo de comunicación se logró que los sensores pudieran interactuar y enviar información al nodo concentrado. Un nodo concentrador se hizo responsable de almacenar los eventos, en este caso era la placa raspberry Pi el cual contiene su propio sistema operativo que es una distribución de linux. En este proyecto nos ayudó a realizar las pruebas y a tomar decisiones de los datos que se reciben del nodo sensor y el nodo identificador.

En el desarrollo de este proyecto fue necesario modelar las entidades del proyecto para así poder almacenar los datos de las variables ambientales existentes en un espacio inteligente para ello se realizó un diagrama entidad relación con los campos correspondientes para que posteriormente se pudiera realizar la base de datos la cual fue desarrollada utilizando en el gestor de base de datos relacional MySQL.

Posteriormente se desarrolló la comunicación entre el cliente y servidor por medio de sockets, para el desarrollo del servidor se utilizó el lenguaje de programación Java y la comunicación con la base de datos fue de manera local.

Netbeans el cual es un entorno de desarrollo fue necesario para poder realizar el servidor y la conexión de la base de datos, para que el cliente y servidor se pudieran comunicar por medio de sockets de manera local utilizando el puerto 8000 y lo más importante fue necesario para realizar la toma de decisiones del espacio inteligente. Fue aquí en donde se declararon ciertos parámetros para que de esta manera se tomarán las decisiones correspondientes a cada variable ambiental.

10 Referencias bibliográficas

[1] A. Peña Ayala, *Sistemas Basados en Conocimiento: Una Base para su Concepción y Desarrollo*. México, D.F.: Instituto Politécnico Nacional, 2006.

[2] C. Casanova, "El rol de los sistemas de información en la gestión del conocimiento en las empresas", *Petrotecnia*, pp. 40-52, 2009.

[3] D. Heredia Vizcaíno, "Common mistakes in software development project management", *Universidad Simón Bolívar*. Barranquilla, Colombia, pp. 25-31, 2013.

[4] P. Ramírez Comonfort, "Prototipo web para gestionar proyectos de software", proyecto terminal, División de CBI, Universidad Autónoma Metropolitana Azcapotzalco, México, 2011.

[5] J. M. Rincón Barrón y A. Navarrete Hinojosa, "Implementación de una base de conocimiento con objetos multimedia para Java básico", proyecto terminal, División de CBI, Universidad Autónoma Metropolitana Azcapotzalco, México, 2009.

[6] J. A. Aguirre López, "Implementación de un árbol de decisiones con extracción automática del modelo a partir de datos estadísticos de cáncer de mama", proyecto terminal, División de CBI, Universidad Autónoma Metropolitana Azcapotzalco, México, 2012.

[7] D. Barrios Crespo, "Desarrollo e implementación de la base de conocimiento del modelo electrónico integrado de ingeniería de empresas Y&V", Licenciatura, Universidad Simón Bolívar, 2008.

[8] A. Castaño Vélez, "Providers Selection Optimization Integrating a Decision Tree in a Business Process", ResearchGate, vol. 10, no. 5, pp. 24-38, 2011.

[9] K. Gilbert, "Técnicas híbridas de Inteligencia Artificial y Estadística para el descubrimiento de conocimiento y la minería de datos", Tendencias de la minería de datos en España, pp. 119-130, 2004.

[10] Abud Figueroa, M. and Soriano Zárate, O. (n.d.). Gestión del Conocimiento. Utilización de XML para Bases de Conocimiento.

[11] Osamu Mizuno, T. and Keishi Sakamoto, Y. (2000). Characterization of Risky Projects based on Project Managers' Evaluation. ACM 23000, pp.387-395.

[12] Wallance, L. and Keil, M. (2004). Software Project Risk and their Effect on outcomes. Communications of the ACM, 47(4), pp.68-73.

[13] Heredia Vizcaíno, D. (2013). Errores comunes en la gestión de proyectos de software. Investigación-Innovación, (Universidad Simón Bolívar), pp.25-31.

[14] Escamilla GG. Información. En: La Biblioteca del futuro. México: UNAM; Secretaría de Servicios Académicos; DGB,1996. p. 258.

11 Apéndice

11.1 Script de SQL de la Base de Datos

```
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';

-- -----
-- Schema mydb
-- -----
-- Schema proyecto
-- -----

-- -----
-- Schema proyecto
-- -----

CREATE SCHEMA IF NOT EXISTS `proyecto` DEFAULT CHARACTER SET utf8 ;
USE `proyecto` ;
```

```
-----  
-- Table `proyecto`.`Division`  
-----
```

```
CREATE TABLE IF NOT EXISTS `proyecto`.`Division` (  
  `idDivision` INT(11) NOT NULL AUTO_INCREMENT,  
  `nombre_div` VARCHAR(45) NULL DEFAULT NULL,  
  PRIMARY KEY (`idDivision`))  
ENGINE = InnoDB  
AUTO_INCREMENT = 4  
DEFAULT CHARACTER SET = utf8;
```

```
-----  
-- Table `proyecto`.`Departamento`  
-----
```

```
CREATE TABLE IF NOT EXISTS `proyecto`.`Departamento` (  
  `idDepartamento` INT(11) NOT NULL AUTO_INCREMENT,  
  `nombre_dep` VARCHAR(45) NULL DEFAULT NULL,  
  `Division_idDivision` INT(11) NOT NULL,  
  PRIMARY KEY (`idDepartamento`),  
  INDEX `fk_Departamento_Division1_idx` (`Division_idDivision` ASC),  
  CONSTRAINT `fk_Departamento_Division1`  
    FOREIGN KEY (`Division_idDivision`)  
    REFERENCES `proyecto`.`Division` (`idDivision`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB  
AUTO_INCREMENT = 6  
DEFAULT CHARACTER SET = utf8;
```

```
-----  
-- Table `proyecto`.`Espacio`  
-----
```

```
CREATE TABLE IF NOT EXISTS `proyecto`.`Espacio` (  
  `idEspacio` INT(11) NOT NULL AUTO_INCREMENT,  
  `edificio` VARCHAR(45) NULL DEFAULT NULL,  
  `numero` INT(11) NULL DEFAULT NULL,  
  PRIMARY KEY (`idEspacio`))  
ENGINE = InnoDB  
AUTO_INCREMENT = 3  
DEFAULT CHARACTER SET = utf8;
```

```
-----  
-- Table `proyecto`.`Profesores`  
-----
```

```
CREATE TABLE IF NOT EXISTS `proyecto`.`Profesores` (  
  `idProfesores` INT(11) NOT NULL AUTO_INCREMENT,  
  `numero_economico` VARCHAR(45) NULL DEFAULT NULL,  
  `nombre_prof` VARCHAR(45) NULL DEFAULT NULL,  
  `apellido_paterno` VARCHAR(45) NULL DEFAULT NULL,  
  `apellido_materno` VARCHAR(45) NULL DEFAULT NULL,  
  `Departamento_idDepartamento` INT(11) NOT NULL,  
  PRIMARY KEY (`idProfesores`),
```

```

INDEX `fk_Profesores_Departamento1_idx` (`Departamento_idDepartamento`
ASC),
CONSTRAINT `fk_Profesores_Departamento1`
  FOREIGN KEY (`Departamento_idDepartamento`)
  REFERENCES `proyecto`.`Departamento` (`idDepartamento`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB
AUTO_INCREMENT = 3
DEFAULT CHARACTER SET = utf8;

```

```

-----
-- Table `proyecto`.`Espacio_Profesor`
-----

```

```

CREATE TABLE IF NOT EXISTS `proyecto`.`Espacio_Profesor` (
  `idEspacio_Profesor` INT(11) NOT NULL AUTO_INCREMENT,
  `Espacio_idEspacio` INT(11) NOT NULL,
  `Profesores_idProfesores` INT(11) NOT NULL,
  PRIMARY KEY (`idEspacio_Profesor`),
  INDEX `fk_Espacio_Profesor_Espacio1_idx` (`Espacio_idEspacio` ASC),
  INDEX `fk_Espacio_Profesor_Profesores1_idx` (`Profesores_idProfesores`
ASC),
  CONSTRAINT `fk_Espacio_Profesor_Espacio1`
    FOREIGN KEY (`Espacio_idEspacio`)
    REFERENCES `proyecto`.`Espacio` (`idEspacio`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Espacio_Profesor_Profesores1`
    FOREIGN KEY (`Profesores_idProfesores`)
    REFERENCES `proyecto`.`Profesores` (`idProfesores`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
AUTO_INCREMENT = 3
DEFAULT CHARACTER SET = utf8;

```

```

-----
-- Table `proyecto`.`Eventos`
-----

```

```

CREATE TABLE IF NOT EXISTS `proyecto`.`Eventos` (
  `idEventos` INT(11) NOT NULL AUTO_INCREMENT,
  `tipo_evento` VARCHAR(45) NULL DEFAULT NULL,
  `variacion` VARCHAR(45) NULL DEFAULT NULL,
  `cantidad_variacion` DOUBLE NULL DEFAULT NULL,
  `tiempo_evento` TIME NULL DEFAULT NULL,
  `idEspacio` INT(11) NULL DEFAULT NULL,
  PRIMARY KEY (`idEventos`),
  INDEX `fk_Eventos_1_idx` (`idEspacio` ASC),
  CONSTRAINT `fk_Eventos_1`
    FOREIGN KEY (`idEspacio`)
    REFERENCES `proyecto`.`Espacio` (`idEspacio`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB

```

```
DEFAULT CHARACTER SET = utf8;
```

```
-----  
-- Table `proyecto`.`Nodo_concentrador`  
-----
```

```
CREATE TABLE IF NOT EXISTS `proyecto`.`Nodo_concentrador` (  
  `idNodo_concentrador` INT(11) NOT NULL AUTO_INCREMENT,  
  `bluetooth_concentrador` DOUBLE NULL DEFAULT NULL,  
  `Espacio_idEspacio` INT(11) NOT NULL,  
  PRIMARY KEY (`idNodo_concentrador`),  
  INDEX `fk_Nodo_concentrador_Espaciol_idx` (`Espacio_idEspacio` ASC),  
  CONSTRAINT `fk_Nodo_concentrador_Espaciol`  
    FOREIGN KEY (`Espacio_idEspacio`)  
    REFERENCES `proyecto`.`Espacio` (`idEspacio`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8;
```

```
-----  
-- Table `proyecto`.`Nodo_identificador`  
-----
```

```
CREATE TABLE IF NOT EXISTS `proyecto`.`Nodo_identificador` (  
  `idNodo_identificador` INT(11) NOT NULL AUTO_INCREMENT,  
  `wifi_identificador` VARCHAR(45) NULL DEFAULT NULL,  
  `bluetooth_identificador` DOUBLE NULL DEFAULT NULL,  
  `Espacio_idEspacio` INT(11) NOT NULL,  
  PRIMARY KEY (`idNodo_identificador`),  
  INDEX `fk_Nodo_identificador_Espaciol_idx` (`Espacio_idEspacio` ASC),  
  CONSTRAINT `fk_Nodo_identificador_Espaciol`  
    FOREIGN KEY (`Espacio_idEspacio`)  
    REFERENCES `proyecto`.`Espacio` (`idEspacio`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8;
```

```
-----  
-- Table `proyecto`.`Nodo_sensor`  
-----
```

```
CREATE TABLE IF NOT EXISTS `proyecto`.`Nodo_sensor` (  
  `idNodo_sensor` INT(11) NOT NULL AUTO_INCREMENT,  
  `temperatura` DOUBLE NULL DEFAULT NULL,  
  `humedad` DOUBLE NULL DEFAULT NULL,  
  `luminosidad` DOUBLE NULL DEFAULT NULL,  
  `presencia` INT(11) NULL DEFAULT NULL,  
  `arduino` VARCHAR(45) NULL DEFAULT NULL,  
  `bluetooth_nodo` DOUBLE NULL DEFAULT NULL,  
  `wifi` VARCHAR(45) NULL DEFAULT NULL,  
  `Espacio_idEspacio` INT(11) NOT NULL,  
  PRIMARY KEY (`idNodo_sensor`),  
  INDEX `fk_Nodo_sensor_Espaciol_idx` (`Espacio_idEspacio` ASC),  
  CONSTRAINT `fk_Nodo_sensor_Espaciol`
```

```

    FOREIGN KEY (`Espacio_idEspacio`)
    REFERENCES `proyecto`.`Espacio` (`idEspacio`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

11.2 Código Fuente para la toma de decisiones

- **Clase: Decisión**

```

static double temperaturaAnterior;
static int contadorTemperatura = 0;
static double humedadAnterior;
static int contadorHumedad = 0;
static double luminosidadAnterior;
static int contadorLuminosidad = 0;

public void decisionTemperatura(double temperatura, String nombreArduino, int
idEspacio) {
    EventoDAO eventoDAO = new EventoDAO();

    if (temperatura < 15) { //compara los rangos
        System.out.println("La temperatura esta fuera del rango minimo: " +
nombreArduino);
    } else if (temperatura > 30) {
        System.out.println("La temperatura esta fuera del rango maximo: " +
nombreArduino);
    }

    if (contadorTemperatura == 0) {
        contadorTemperatura++;
    } else {

        double resultado = temperatura - temperaturaAnterior;
        DecimalFormat formatoDecimal = new DecimalFormat("#.##");
        double r=Double.valueOf(formatoDecimal.format(resultado));

        if (r <= -2.14) {
            eventoDAO.insertarEvento("temperatura", "Disminuyó", r * -1,
idEspacio); //para que el numero sea positivo
        } else if (r >= 2.14) {
            eventoDAO.insertarEvento("temperatura", "Aumentó", r, idEspacio);
        }
        contadorTemperatura++;
    }

    temperaturaAnterior=temperatura;
}

```

```

public void decisionHumedad(double humedad, String nombreArduino, int idEspacio)
{
    EventoDAO eventoDAO = new EventoDAO();

    if (humedad < 0.4) {
        System.out.println("La humedad esta fuera del rango minimo: " +
nombreArduino);
    } else if (humedad > 0.6) {
        System.out.println("La humedad esta fuera del rango maximo: " +
nombreArduino);
    }
    if (contadorHumedad == 0) {
        contadorHumedad++;
    } else {

        double resultado = humedad -humedadAnterior;
        DecimalFormat formatoDecimal = new DecimalFormat("#.##");
        double r=Double.valueOf(formatoDecimal.format(resultado));

        if (r <= -0.053) {
            eventoDAO.insertarEvento("humedad", "Disminuyó", r * -1,
idEspacio);//para que el numero sea positivo
        } else if (r >= 0.053) {
            eventoDAO.insertarEvento("humedad", "Aumentó", r, idEspacio);

        }
        contadorHumedad++;

    }
    humedadAnterior=humedad;
}

```

```

public void decisionLuminosidad(double luminosidad, String nombreArduino, int
idEspacio) {
    EventoDAO eventoDAO = new EventoDAO();

    if (luminosidad < 250) {//compara los rangos
        System.out.println("La luminosidad esta fuera del rango minimo: " +
nombreArduino);
    } else if (luminosidad > 400) {
        System.out.println("La luminosidad esta fuera del rango maximo: " +
nombreArduino);
    }
    if (contadorLuminosidad==0) {
        contadorLuminosidad++;
    }
    else{
        double resultado = luminosidad - luminosidadAnterior;
        DecimalFormat formatoDecimal = new DecimalFormat("#.##");
        double r=Double.valueOf(formatoDecimal.format(resultado));

        if (r <= -13.33) {
            eventoDAO.insertarEvento("luminosidad","Disminuyó",r*-1,
idEspacio);//para que el numero sea positivo
        }
        else if ( r >= 13.33) {
            eventoDAO.insertarEvento("luminosidad","Aumentó",r, idEspacio);

        }
        contadorLuminosidad++;
    }
}

```

```
}  
luminosidadAnterior=luminosidad;  
}
```