

IoT para el monitoreo de la concentración de los estudiantes en un aula de clases y su relación con los factores ambientales.

Eder Samir Correa Acosta

Universidad de Antioquia - GITA
Medellín, Colombia
eder.correa@udea.edu.co

1. INTRODUCCIÓN

Para este trabajo se ha diseñado un prototipo simulado en la plataforma Node-Red, en el cual el escenario simulado es un aula educativa monitoreado por una red IoT, en donde hay estudiantes y docentes en un encuentro académico. El sistema diseñado busca notificar al docente cuando los estudiantes no están atentos en su clase, a causa de factores ambientales (ruido, temperatura y luminosidad) y de esta forma él pueda realizar cambios en dichos factores, los cuales son sugeridos por medio de un sistema de alerta que puede estar en dos estados: estado de alerta y estado de operación normal; en el primer caso hay necesidad de hacer cambios en los factores ambientales y en el segundo caso no se generará ninguna recomendación o acción por parte de dicho sistema.

Así mismo, se crea un módulo de análisis de datos y de recomendación, en el cual se implementa un modelo predictivo que permite anticipar el estado del sistema basado en los datos obtenidos en la experimentación. Es implementado usando el algoritmo “decision tree” (Safavian *et. al.*, 1991). Esto puede ser útil, pues el sistema diseñado que realiza las recomendaciones, solo lo puede hacer cuando estas están disponibles en una base de datos, en el caso de que no exista, no podría sugerir los cambios.

La implementación de un proyecto de este tipo, puede ser útil en un ambiente escolar en que se quiera que la atención de los estudiantes no se vea afectada por factores externos, para así obtener mejores resultados educativos. También se fijan las bases, para realizar un trabajo futuro que pueda implementar técnicas de Inteligencia Artificial (IA) para el modelamiento de estudiantes y el análisis de datos educativos.

2. OBJETIVO

General

- Automatizar el monitoreo de la concentración de los estudiantes en un aula de clases usando IoT, determinando los factores ambientales que influyen en este.

Específicos

- Determinar el estado general de concentración de los estudiantes en un aula de clases, por medio de sensores no invasivos.
- Analizar la información obtenida de los sensores, para generar alertas en un sistema de monitoreo central.

3 - ARQUITECTURA

A continuación, se describe la arquitectura del sistema simulado. En la figura 1 se muestra de forma global, la arquitectura del sistema y luego se describen los componentes individuales.

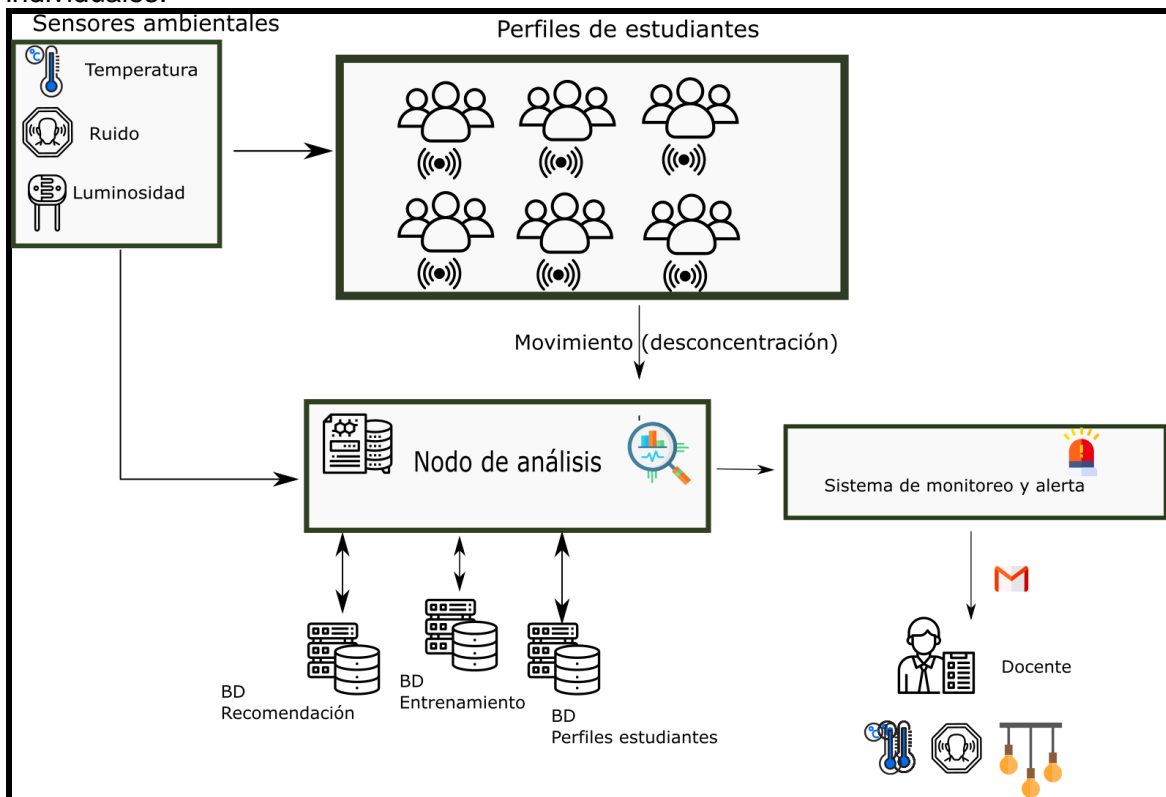


Figura 1. Arquitectura del sistema simulado.

Sensores: Miden las condiciones ambientales del aula. Para este proyecto se consideraron tres tipos de sensores: temperatura, ruido y luminosidad. A su vez, cada estudiante posee un sensor de movimiento, que se activará cuando los sensores reporten datos que estén fuera del límite de tolerancia definidos en el perfil del estudiante.

Perfil de estudiante: Representa el comportamiento de los estudiantes con respecto a las condiciones ambientales. En la búsqueda bibliográfica no se detectó un dataset que representara este comportamiento. Por lo tanto, se crean de forma manual perfiles que representan diferentes condiciones. Es así, como se definieron 6 comportamientos diferentes. Cada perfil tiene un nivel mínimo y máximo de tolerancia y cuando este es

sobrepasado por alguna de las condiciones ambientales, este envía la señal de movimiento representando incomodidad del estudiante, y, por lo tanto, falta de concentración. El perfil en un ambiente real representaría una combinación de variables que influyen en el comportamiento del estudiante (estrato, género, ciudad de nacimiento, promedio, edad, entre otros).

En la tabla 1 se muestran los perfiles escogidos para esta simulación.

Perfil	Temperatura mínima	Temperatura máxima	Ruido Mínimo	Ruido máximo	Luminosidad Mínima	Luminosidad Máxima	Descripción
1	10	30	1	5	1	5	Muy tolerantes
2	17	25	1	2	3	4	Poco tolerantes
3	12	28	1	4	2	5	Tolerantes
4	15	27	1	3	3	5	Medio
5	25	35	3	5	3	4	De clima calido
6	1	12	1	1	2	4	De clima frio

Tabla 1. Perfiles de usuario.

Bases de Datos: Para asegurar la persistencia de la información se implementan bases de datos. Se consideraron 2 bases de datos principales que se irán alimentando mientras la simulación está en curso.

- *Base de datos de recomendación:* Esta base de datos se irá actualizando a medida que exista la condición “estado de operación normal”, es decir, cuando la mayoría de estudiantes están concentrados en las instrucciones del docente. Será guardado el valor registrado por los sensores y el número de estudiantes de cada perfil. Esta base de datos sirve para consultas cuando está activo el estado de alerta.
- *Base de datos de entrenamiento.* Aquí se guardan el estado del sistema y los valores que tenían los sensores cuando se reportaba dicho estado. Esta base de datos se usará para implementar algoritmos de aprendizaje de máquina que puedan predecir futuros estados del sistema.

Sistema de análisis: Encargado de realizar un análisis general del estado de concentración de los estudiantes, basado, en los datos provenientes de los sensores. Cuando más del 25% de los estudiantes del aula estén en estado de movimiento (representando falta de concentración), este notifica que existe la necesidad de hacer cambios en las condiciones ambientales. De esta forma, se envía una señal al sistema de monitoreo y alerta. También interactúa constantemente con las bases de datos, para el correcto funcionamiento del sistema.

Sistema de Monitoreo y alerta: Encargado de notificar al docente que existe la necesidad de hacer cambios en las condiciones dentro del aula. Además, si existen recomendaciones previamente guardadas en la base de datos “recomendación”, esté envía un e-mail indicando los cambios que son necesarios en las condiciones del aula.

Sistema de IA: Este sistema funciona fuera de la arquitectura anteriormente presentada, pues se usa posterior a la obtención de los datos. Este sistema obtiene la información recibida de la base de datos de entrenamiento y crea un modelo predictivo que sirve para

generar recomendaciones. El modelo se crea usando algoritmos de IA con los datos guardados en la base de datos anteriormente mencionada. Este tiene la ventaja con respecto al sistema implementado en la arquitectura original, de que es capaz de generar recomendaciones que el sistema no necesariamente conoce.

4- IMPLEMENTACIÓN:

En esta sección se presentarán los elementos principales que fueron necesarios para la implementación de la simulación.

Software necesario: Los siguientes componentes de software fueron necesarios para la implementación de la propuesta:

- *Sistema Operativo:* El proyecto se realizó en el sistema operativo Ubuntu 18.04.
- *Node-red:* Es una herramienta de programación en IoT desarrollada por IBM. Proporciona un editor de flujo visual, accedido mediante un navegador que facilita la conexión del flujo (Kim, S., et al., 2019). El sistema contiene nodos que están representados por los iconos apropiados y el lenguaje de programación de la plataforma es JavaScript.
- *MQTT:* Se instala el servicio MQTT. Este servicio permite la implementación del protocolo del mismo nombre. Este es uno de los protocolos de mensajería livianos que sigue el paradigma de publicación-suscripción (Dizdarević, et al 2019). Los clientes se conectan con un servidor central denominado “broker” que corre en el sistema operativo.
- *MYSQL:* Se implementa la base de datos MYSQL para guardar de forma permanente o temporal datos de la simulación, estas bases de datos interactúan de forma continua con el software Node-red mediante la implementación de los nodos MYSQL.
- *Servicio de mensajería:* Para enviar los mensajes de alerta se usa el sistema de mensajería de Google (Gmail). Para esto se crea una cuenta de correo electrónico y se configuran los parámetros necesarios que permiten la ejecución del correo electrónico desde Node-red. Para este trabajo se creó un usuario llamado “noderedsamir@gmail.com”. Desde esta cuenta se crean los mensajes.

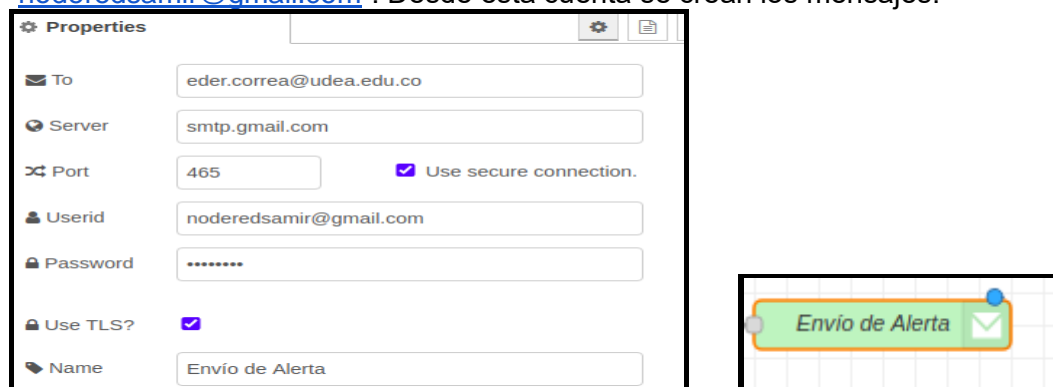


Figura 2. Usando gmail desde la plataforma Node-red.

Sensores: Los sensores que captan las condiciones ambientales son representados en la implementación por clientes “Publish” del servicio MQTT (Figura 3) y publican en el “broker” con el tópico: “topic/sensor”, cada vez que se registra un cambio en las condiciones ambientales se crean los nodos en el programa Node Red de tipo “MQTT IN”.

Receptores MQTT: Los nodos receptores reciben las condiciones ambientales enviados por los sensores mediante el protocolo MQTT y son implementados en el flujo de Node-red como nodos “Subscribe” (Figura 3). Estos se suscriben bajo el tópico: “topic/sensor” y reciben mensajes del “broker” cada vez que se registra un cambio en las condiciones ambientales.

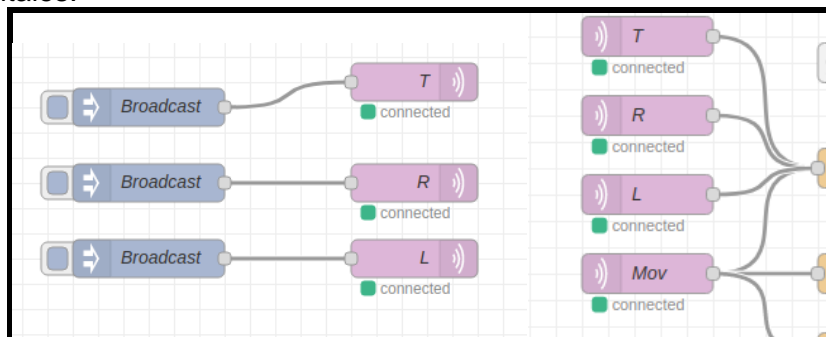


Figura 3. En la izquierda Nodos “Publish” y en la Derecha Nodos “Subscribe”.

Perfiles de los usuarios: Para la implementación de estos perfiles se usaron los bloques “Function” de Node-red (Figura 4). Las entradas de estos nodos, son del tipo “Subscribe” que reciben las condiciones climáticas, y la salida son sensores “publish” que publican bajo el tópico: “topic/movimiento”.

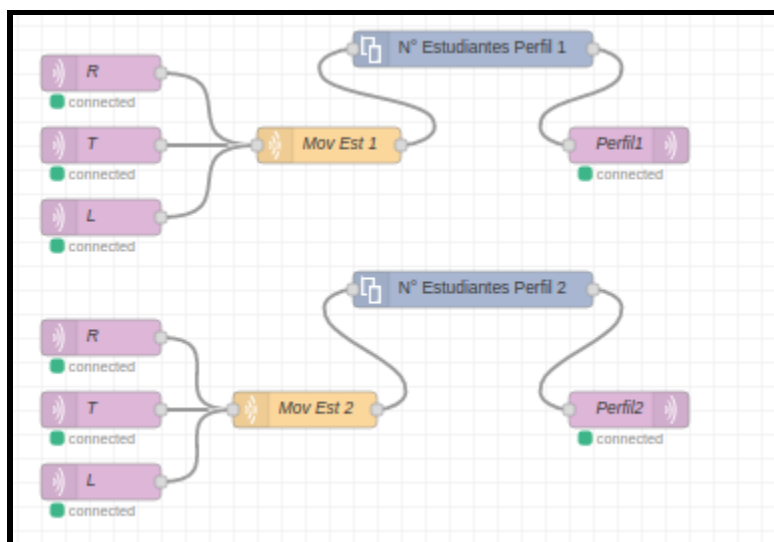


Figura 4. Perfiles de los estudiantes.

Nodos de procesamiento. Los nodos de procesamiento están distribuidos en toda la simulación y son implementados con el bloque de Node-red “Function” (Figura 5). Estos

son bloques de código escritos en el lenguaje de programación JavaScript. Estos procesan los datos, envían y reciben peticiones de las bases de datos, pre procesan los datos para enviarlos al servicio de mensajería y todas las tareas asociadas a la transformación de los datos.

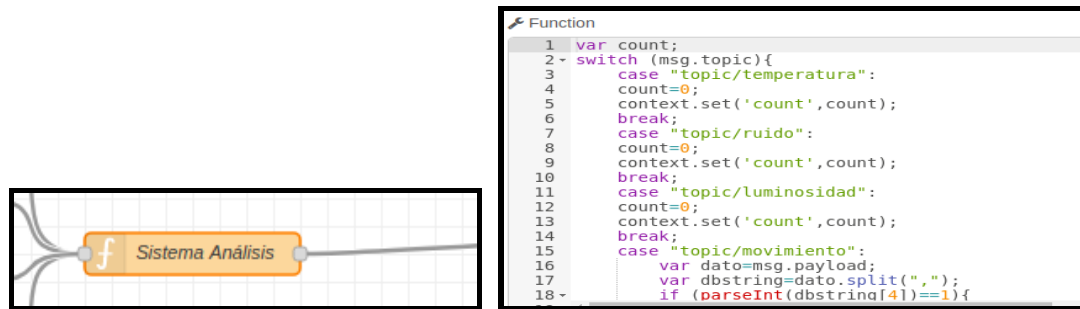


Figura 5. Nodo Sistema de análisis tipo "Function", usando el lenguaje de programación Java Script.

Sistema Inteligente: Se implementa el flujo de la figura 6 en Node-red con el fin de crear una metodología que pueda generar un sistema predictivo, para optimizar la recomendación del cambio de las condiciones que se le envía al docente. El siguiente flujo muestra la implementación hecha en Node-red:

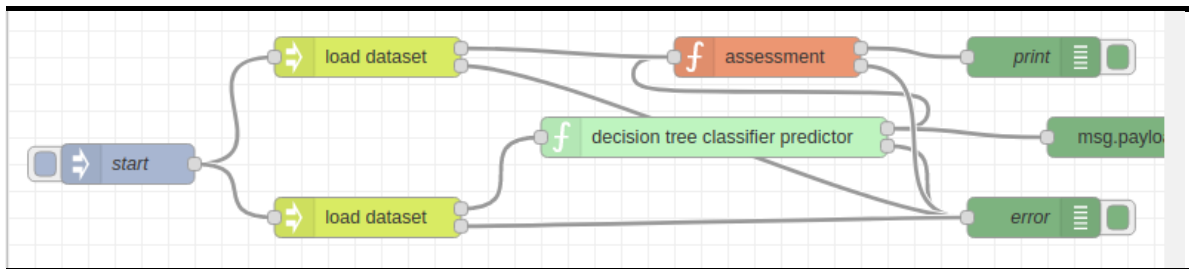


Figura 6. Nodos "Functions" usando el lenguaje de programación Java Script.

5- FUNCIONAMIENTO Y RESULTADOS.

Escenario de simulación: El escenario que se simula en este proyecto es un aula de clases en donde hay 40 estudiantes definidos por los perfiles de la tabla 1. El número de estudiantes se mantiene fijo, aunque la combinación de perfiles puede ser diferente (representando cambios de clase en una institución educativa). En caso de que más del 25% de estudiantes esté registrando movimiento el sistema enviará una alerta vía e-mail con las sugerencias (si las hay).

Prueba de funcionamiento: Para realizar la validación del funcionamiento del sistema se crea la siguiente configuración de perfiles de estudiantes:

Perfil 1	Perfil 2	Perfil 3	Perfil 4	Perfil 5	Perfil 6
20	5	10	3	1	1

Tabla 2. Perfiles para la simulación.

Después de la configuración anterior, se ingresan valores al azar entre rangos funcionales de los factores ambientales, con el fin de validar su funcionamiento. La tabla 3 extraída de MYSQL muestra los resultados de este experimento, en el cual el estado 0 indica estado de operación normal y el estado 1 indica estado de alerta:

Temperatura	Ruido	Iluminacion	Estado	Temperatura	Ruido	Iluminacion	Estado
26	2	3	0	26	2	3	0
32	2	1	1	32	2	1	1
11	4	2	1	11	4	2	1
30	2	1	1	30	2	1	1
9	2	4	1	9	2	4	1
9	2	4	1	9	2	4	1
7	3	2	1	7	3	2	1
7	2	1	1	7	2	1	1
28	3	4	0	28	3	4	0
32	3	2	1	32	3	2	1
7	1	1	1	7	1	1	1

Tabla 3. Resultados Base de datos “Entrenamiento” obtenida desde MYSQL

En el caso representado en la primera fila de la tabla anterior, el sistema no generó ninguna alerta. En cambio, en el caso representado por la segunda fila de la tabla 3 el sistema genera alerta y se recibieron mensajes como el mostrado en la siguiente figura:

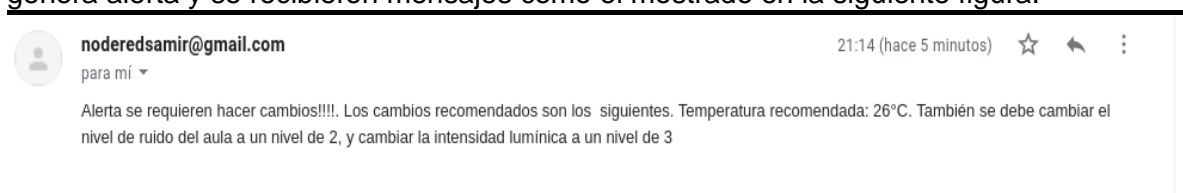


Figura 7. Notificación obtenida desde nodered al correo electrónico.

En el caso de la figura 7, se obtuvo una sugerencia, pues en la base de datos “recomendaciones” se almacena la combinación de perfiles y los cambios sugeridos, como se puede ver en la tabla 4.

p1	p2	p3	p4	p5	p6	temperatura	ruido	luminosidad
30	3	1	1	2	3	20	2	3
10	10	5	5	2	8	20	1	4
10	10	5	5	2	8	20	1	4
10	10	5	5	2	8	20	1	4
10	10	5	5	2	8	20	1	4
10	10	5	5	2	8	20	1	4
10	10	5	5	2	8	20	1	4
10	10	5	5	2	8	17	1	3
20	15	2	1	1	1	18	2	3
30	5	2	1	1	1	29	4	2
30	5	2	1	1	1	29	4	2
30	5	2	1	1	1	22	2	4
21	3	11	2	2	1	20	1	2
20	5	10	3	1	1	26	2	3

Tabla 4. Base de datos recomendación

El sistema de recomendación escoge la primera opción disponible en la base de datos “recomendaciones” para las configuraciones disponibles. En la figura anterior se resalta la opción que el sistema detectó y son los valores que finalmente envía al usuario final. En este sentido el sistema también fue capaz de realizar las acciones de forma correcta.

Entrenamiento del sistema y modelo de IA: En la última fase se busca establecer un modelo que permita realizar cambios de forma predictiva y de esta forma implementar un sistema recomendador basado en técnicas de aprendizaje de máquina. Para esto se utilizó el algoritmo “Decision tree”. Uno de los índices de desempeño importantes cuando se crea un modelo predictivo, es la precisión con que este es capaz de realizar esta tarea. La precisión indica las veces que el algoritmo acierta en la clasificación con respecto a datos de prueba o validación.

Para lo anterior, en este trabajo inicialmente se estudia cuantos datos se necesitan para obtener una buena precisión. Entonces se ejecuta la simulación 126 veces, obteniendo los resultados de precisión que se muestran en la siguiente tabla.

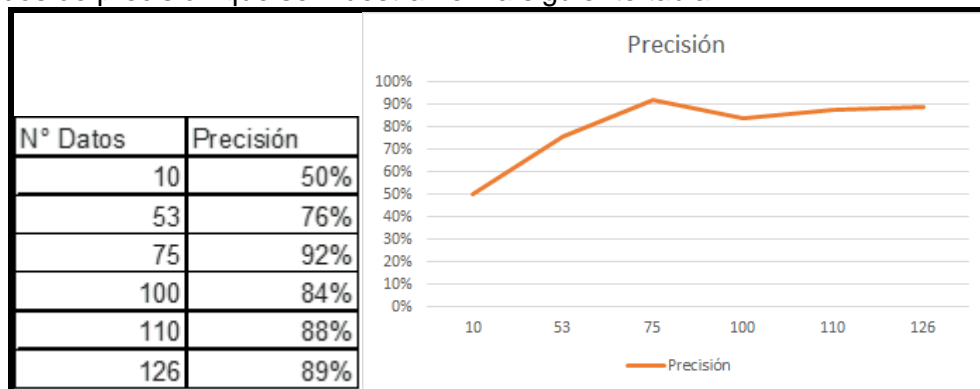


Figura 6. N° de Datos vs precisión del Sistema de IA.

Se puede observar en la anterior figura que la precisión del sistema cuando se tienen 126 datos es del 89%. En 75 datos se presenta un caso particular y es que el modelo aparentemente tiene una precisión del 92%, lo cual no es un valor esperado. Esto se puede dar porque la cantidad de datos de validación aún son insuficientes para generalizar el desempeño del modelo.

Para validar el funcionamiento del modelo recomendador se ingresan 4 datos que el algoritmo nunca ha visto. Estos mismos valores son simulados en el sistema y el resultado se muestra en la siguiente tabla:

T	R	L	Clasificación hecha por el algoritmo	Resultado de la simulación.
29.3	2.5	3.8	1	1
13.4	1.2	2.2	0	0
33.5	4.2	4.2	1	1
26.2	1.5	4.3	0	0

Tabla 5. Resultado de la clasificación del estado del sistema predictivo.

Vemos que el algoritmo acierta el 100 % de las veces del ejemplo propuesto. Aunque la medida de precisión indica que el 12 % de las veces no recomendará la opción correcta. Estos datos insertados representan recomendaciones que el sistema usando el modelo predictivo, es capaz de proporcionar al docente y que son aprendidos por el sistema de acuerdo a los datos que se producen durante la simulación.

6- DISCUSIÓN, CONCLUSIÓN Y TRABAJO FUTURO.

En este trabajo se pudo simular un sistema de IoT capaz de monitorear y notificar el estado general de concentración de los estudiantes en un aula de clases. Mediante la plataforma de desarrollo Node-red, fue posible la interacción entre varios servicios como MQTT, MYSQL y GMAIL que facilitan la implementación y el correcto funcionamiento del sistema.

Aunque se realizó un estudio de cómo se comportaba la precisión de un sistema de IA, estos resultados deben ser considerados como preliminares, pues se debe estudiar qué algoritmo de IA se puede adaptar mejor al contexto y además se deben tener más datos para mejorar la confiabilidad y la precisión del sistema. Se observó en general que cuando los datos aumentaban, mejoraba la precisión. Esta metodología permitiría establecer recomendaciones al docente, cuando uno de los datos no pueda ser cambiado e incluso predecir datos que nunca se han ingresado al sistema.

Los perfiles de estudiantes usados en este proyecto se hicieron con datos no basados en la realidad. Para el futuro se pretende usar datos reales y de esta forma mediante técnicas de aprendizaje no supervisado, detectar automáticamente los perfiles de los estudiantes que fueron creados aquí manualmente. Finalmente, el sistema se puede fortalecer buscando que este automáticamente cambie las condiciones ambientales del aula, pues en un ambiente real, no es conveniente que el docente tenga esta carga de trabajo.

Referencias:

Kim, S., Park, J., Jeong, J., Yun, Y. S., Eun, S., & Jung, J. (2019, September). Survey of IoT platforms supporting artificial intelligence. In Proceedings of the Conference on Research in Adaptive and Convergent Systems (pp. 65-66).

Lekić, M., & Gardašević, G. (2018, March). IoT sensor integration to Node-RED platform. In 2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH) (pp. 1-5). IEEE

Dizdarević, J., Carpio, F., Jukan, A., & Masip-Bruin, X. (2019). A survey of communication protocols for internet of things and related challenges of fog and cloud computing integration. ACM Computing Surveys (CSUR), 51(6), 1-29.

Safavian, S. R., & Landgrebe, D. (1991). A survey of decision tree classifier methodology. IEEE transactions on systems, man, and cybernetics, 21(3), 660-674.