

PROTOTIPO DE SIRENA AUTOMATIZADA UTILIZANDO TECNOLOGÍAS INTERNET DE
LAS COSAS (IOT) COMO ESTRATEGIA PARA OPTIMIZAR EL CUMPLIMIENTO DE LA
JORNADA ESCOLAR EN INSTITUCIONES EDUCATIVAS.

JUAN SEBASTIÁN BADILLO RINCÓN

UNIVERSIDAD POPULAR DEL CESAR
FACULTAD DE INGENIERÍAS Y TECNOLÓGICAS
PROGRAMA INGENIERÍA DE SISTEMAS
LÍNEA DE INVESTIGACIÓN TELEMÁTICA
SECCIONAL AGUACHICA, CESAR

2024

PROTOTIPO DE SIRENA AUTOMATIZADA UTILIZANDO TECNOLOGÍAS INTERNET DE
LAS COSAS (IOT) COMO ESTRATEGIA PARA OPTIMIZAR EL CUMPLIMIENTO DE LA
JORNADA ESCOLAR EN INSTITUCIONES EDUCATIVAS.

JUAN SEBASTIÁN BADILLO RINCÓN

Propuesta proyecto de grado para optar el título de Ingeniero de sistemas, modalidad
presencial.

Director(a)

Ing. Darwin Navarro Pino

UNIVERSIDAD POPULAR DEL CESAR
FACULTAD DE INGENIERÍAS Y TECNOLÓGICAS
PROGRAMA INGENIERÍA DE SISTEMAS
LÍNEA DE INVESTIGACIÓN TELEMÁTICA
SECCIONAL AGUACHICA, CESAR

2024



**UNIVERSIDAD POPULAR
DEL CESAR**
Seccional Aguachica

NOTA DE ACEPTACIÓN

FIRMAS

DIRECTOR

JURADO

JURADO

Aguachica, 2024

Dedicatoria

Quiero comenzar expresando mi profundo agradecimiento a Dios, quien ha sido mi guía constante en este viaje de crecimiento personal y profesional. Agradezco a Dios por haber puesto en mi vida a las personas adecuadas, los pensamientos oportunos y las circunstancias apropiadas que han contribuido a mi formación.

A mis padres, Sandra Isabel Rincón Medina y Jaime Badillo Cáceres, les debo todo lo que soy y lo que llegaré a ser. Su influencia ha moldeado mi esencia y mis valores, y mi mayor anhelo es hacer realidad sus sueños como muestra de agradecimiento por el amor infinito e incondicional que me han brindado. Este logro no es solo mío, sino también de ellos.

A mi hermanita, Andrea Camila Badillo Rincón, quien desde su nacimiento ha añadido a mi vida un profundo sentimiento de responsabilidad. En ella encuentro un reflejo de mí mismo, y estoy seguro de que, en muchos aspectos, podrá superarme.

A mi novia, Melisa Alexandra Carrascal Sanjuan, le agradezco haber llegado a mi vida y por su apoyo en cada una de mis decisiones y circunstancias. Gracias a ella, he logrado salir de mi zona de confort y enfrentar nuevos desafíos con valentía. Su impulso constante me ha llevado a superarme día a día y a cumplir con cada una de mis responsabilidades. Juntos hemos compartido el camino hacia nuestros objetivos académicos, superando obstáculos y desafíos con determinación.

A mis mascotas, Molly y Lilly, les expreso mi amor eterno y mi más profundo agradecimiento porque a través de ellas, he aprendido que el amor va más allá de lo tangible, y que con un simple ladrido son capaces de hacerme sentir lo importante que soy para ellas.

A todos ellos, desde lo más profundo de mi ser, los amo con todas mis fuerzas.

Quiero agradecer también a todos mis profesores y, sobre todo, al Ing. Darwin Navarro Pino, que es para mí un ejemplo a seguir, por su paciencia, profesionalismo y calidad de persona.

Con cariño, Juan Sebastian Badillo Rincón.

Contenido

1. PROBLEMA	7
1.1 FORMULACIÓN DEL PROBLEMA	9
1.2 JUSTIFICACIÓN	9
1.3 OBJETIVOS	11
1.3.1. <i>Objetivo General</i>	11
1.3.2. <i>Objetivos específicos</i>	11
1.4 DELIMITACIONES.....	11
1.4.1 <i>Temporal</i>	11
1.4.2 <i>Espacial</i>	11
1.4.3 <i>Contextual</i>	12
2. MARCO REFERENCIAL	12
2.1 MARCO HISTÓRICO.....	12
2.2 MARCO TEÓRICO.....	14
2.2.1 <i>Internet de las cosas</i>	14
2.2.2 <i>Impacto y beneficio del IoT</i>	16
2.2.1 <i>ISO/IEC 30141</i>	17
2.2.2 <i>Arquitectura de 5 Capas</i>	18
2.3.3 <i>Cloud Computing para Internet de las cosas (IoT)</i>	19
2.3.3.1 <i>Arquitectura de Cloud Computing</i>	20
2.3.3.2 <i>Modelos de servicios de Cloud Computing</i>	21
2.3.3.3 <i>Modelos de Implementación de Cloud Computing</i>	22
2.3.5 <i>Metodología de prototipado</i>	23
2.3 MARCO LEGAL.....	24
2.4 MARCO CONCEPTUAL.....	25
3. DISEÑO METODOLÓGICO PRELIMINAR.....	27
3.1 TIPO DE INVESTIGACIÓN:.....	27
3.2 POBLACIÓN Y MUESTRA	28
3.2.1 <i>Población</i>	28
3.2.2 <i>Muestra</i>	29
3.3 INSTRUMENTOS DE RECOLECCIÓN DE DATOS	29
3.4 PROCEDIMIENTOS O MÉTODOS DE DESARROLLO	29
4. RESULTADOS.....	31
4.1 IDENTIFICACIÓN DE LOS REQUERIMIENTOS Y LAS TECNOLOGÍAS DEL INTERNET DE LAS COSAS APLICABLES AL PROTOTIPO	31
4.2 DISEÑO DEL PROTOTIPO HARDWARE Y SOFTWARE SEGÚN LOS REQUERIMIENTOS DEFINIDOS Y LAS TECNOLOGÍAS IoT SELECCIONADAS EN LA FASE DE ANÁLISIS.....	36
4.3 CONSTRUIR Y CODIFICAR EL PROTOTIPO IoT, INCLUYENDO LA APLICACIÓN MÓVIL DE CONTROL LOCAL Y LA APLICACIÓN MULTIPLATAFORMA DE GESTIÓN DE HORARIOS	38
4.4 PRUEBA PILOTO.....	42

4.4.1 Evidencias de la prueba de concepto o piloto.....	42
4.4.2 Evidencias de las pruebas de aceptacion	45
CONCLUSIONES.....	47
ANEXOS	48
ANEXO A.....	48
CÓDIGO DEL PROGRAMA DE LA ESP32 DEV-KIT.....	48
ANEXO B	66
ANDROID STUDIO.....	66
ANEXO C	67
ESTRUCTURA DEL JSON.....	67
ANEXO D.....	68
MANUAL DE USUARIO	68
BIBLIOGRAFIA.....	72

1. PROBLEMA

A lo largo del tiempo, en las instituciones educativas se han implementado procedimientos manuales que con el tiempo han perdido su efectividad y eficiencia debido a una variedad de factores. Como resultado, se hace necesario realizar una evaluación y actualización constante de los procesos educativos para garantizar una experiencia de aprendizaje efectiva y de alta calidad para los estudiantes. Acosta-Díaz et al. (2020) expresa lo siguiente:

En el sistema educativo nacional se asocia la eficiencia con los niveles de logro de indicadores que se alcanzan en un periodo determinado y, por tanto, la eficiencia en las operaciones que estén encaminadas a lograr dichos objetivos educativos. La medición tendrá que realizarse de acuerdo con las condiciones dadas en cada caso; sin embargo, al hablar del uso de las tecnologías de información y comunicación (TIC) en este ámbito y particularmente con referencia a Internet de las cosas (IoT, por sus siglas en inglés), se presenta un panorama muy extenso en cuanto a las posibilidades de automatizar procesos e incrementar la eficiencia de las operaciones en las escuelas y universidades en general. (p. 39)

A diario en las Instituciones Educativas se desarrollan jornadas académicas donde el cumplimiento del tiempo en las sesiones de clases es relevante para el correcto desarrollo de las mismas, evitando invadir tiempo de otros docentes o asignaturas, que los estudiantes pierdan tiempo esperando el docente, que los espacios físicos se encuentren ocupados en otras clases, que se altere el desarrollo ordenado de las clases por cambios de horarios, son algunos de los inconvenientes que se presentan a menudo por no tener una herramienta para el manejo óptimo del cumplimiento de los horarios en las sesiones de clases.

La problemática radica en que existe un actor encargado de dar cumplimiento a las sesiones de clases y esta persona es encargada de realizar el toque de un timbre en las horas estipuladas por la institución, cuando se habla de ello hacemos referencia a los períodos de

las sesiones de clases, para indicar la terminación y el inicio de la siguiente. El actor en mención es llamado comúnmente celador.

Además de la función descrita anteriormente, el celador también es encargado de múltiples tareas que contribuyen al funcionamiento eficiente y seguro de la institución educativa. Su principal función es mantener la seguridad en el colegio, controlando el acceso de personas ajenas a la institución y supervisando las instalaciones y equipos. Además, debe vigilar la disciplina y hacer cumplir las normas y reglamentos establecidos por la institución. El celador también es el primer contacto que tienen los visitantes al llegar al colegio, por lo que es responsable de atenderlos, proporcionarles información y guiarlos hacia las personas o lugares adecuados dentro de la institución. Asimismo, puede asistir en la organización y desarrollo de actividades extracurriculares, mantener la limpieza y el orden en las instalaciones del colegio, y apoyar en tareas administrativas. En definitiva, el papel del celador en un colegio es fundamental para garantizar la seguridad, el bienestar y el éxito de los estudiantes y el personal.

Tal como se menciona en (Peleteiro & Seferian, 2006), esta persona cuenta con múltiples funciones dentro de la institución, por consiguiente se dificulta la posibilidad de elaborarlas todas al tiempo establecido, muchas veces al momento exacto para el toque del timbre no se encuentre en el lugar donde está ubicado el dispositivo, o puede olvidar realizar el toque para el cambio. En los centros educativos, pueden surgir otras situaciones que afectan a la organización de las actividades escolares, y una de ellas es la modificación de la duración de las clases debido a actividades extracurriculares o imprevistos. Cuando esto ocurre, es necesario hacer ajustes para que los estudiantes y el personal docente puedan seguir el ritmo del calendario escolar. Sin embargo, estos ajustes pueden tener efectos secundarios, como el uso del timbre para señalar el inicio y el final de las clases. En algunos casos, el timbre debe generar niveles de ruido muy elevados para que sea perceptible en todo el colegio, lo que puede generar contaminación auditiva y afectar negativamente la salud

auditiva de las personas. Los trastornos auditivos asociados a la contaminación son la pérdida auditiva permanente y la elevación del umbral auditivo. Además, los efectos psicológicos y sociales del ruido pueden provocar cansancio, tensión y enfermedades nerviosas o cardiovasculares. Además, para las personas con limitaciones auditivas, el timbre puede ser un dispositivo excluyente, ya que les resulta difícil escucharlo si se encuentran alejados o si el nivel de ruido ambiental es alto.

1.1 Formulación del problema

¿De qué manera se puede ayudar en la gestión del cumplimiento de la jornada académica escolar?

1.2 Justificación

El termino Internet de las Cosas (IoT) es utilizado para describir la necesidad de conectar todas las cosas a la red de internet convirtiéndose en parte fundamental para el desarrollo de sistemas de tecnologías de la información y las telecomunicaciones (TIC), IoT utiliza las tecnologías emergentes para integrar objetos conectándolas por medio de internet en cualquier lugar y hora, buscando tener en cuenta la compatibilidad de sus aplicaciones entre dispositivos y sus tecnologías (Pinzón Niño, 2016). Sin embargo, el IoT va más allá; es el pilar sobre el que sienta sus bases el paradigma de la industria 4.0. La ejecución del proyecto tiene dos propósitos, desarrollar un prototipo IoT que permita optimizar el control y el cumplimiento de la jornada escolar. El otro sin ser menos importante en el marco de la investigación formativa, es aportar al cuerpo del conocimiento de IoT para ser aplicado en el contexto del currículo del programa académico de ingeniería de sistemas de la Universidad Popular del Cesar Seccional Aguachica.

Las Instituciones Educativas en su rutina diaria llevan a cabo diferentes actividades que en muchos casos son programadas en último momento, en las cuales los estudiantes deben ser partícipes de ellas, sin olvidar que toman sus clases en sus respectivas aulas. Diariamente el estudiantado atiende al toque de un timbre que les indica un cambio de hora,

un evento propuesto, terminación de la jornada escolares, son diferentes los motivos a los que los estudiantes están atentos al toque del timbre o sirena.

Analizando las diferentes situaciones presentes en el desarrollo de la jornada escolar de las instituciones educativas, relacionadas con el cumplimiento del tiempo de las sesiones de clases, se observa la necesidad de automatizar este proceso y eliminar la intervención de una persona para su operación.

El prototipo IoT debe disponer de diferentes formas de ser configurado, mediante un panel standalone y mediante una aplicación, el cual permitirá realizar actualizaciones o modificaciones en el horario establecido y además de ello manipular los toques a través de la aplicación en tiempo real.

La gestión del sistema sería llevada a través de una aplicación multiplataforma, la cual notificaría a los docentes de la institución de los cambios de horas y eventualidades como lo son inicio, cambio y finalización de clases o imprevistos en el momento; a través de una página web se realizará la configuración de los componentes del sistema como lo son carga, modificación y/o actualización de horarios de la institución y el comportamiento de la sirena, llevara registro del tiempo de las jornadas laboradas para generar estadísticas sobre el tiempo escolar.

Para la ejecución del proyecto se seguirán las fases de la metodología de prototipado, cuyo objetivo es construir un prototipo de manera rápida para experimentar, retroalimentar y posteriormente hacer ajustes. Las fases propuestas son; especificaciones y requerimientos, diseño, construcción, pruebas y optimización.

1.3 Objetivos

1.3.1. Objetivo General

Desarrollar un prototipo de sirena automatizada utilizando tecnologías IoT como estrategia para optimizar el cumplimiento de la jornada escolar en instituciones educativas.

1.3.2. Objetivos específicos

El objetivo general se alcanzará mediante la elaboración de los siguientes objetivos específicos:

- Identificar los requerimientos y las tecnologías del Internet de las Cosas que pueden ser aplicables al prototipo.
- Diseñar los prototipos Hardware y Software según los requerimientos definidos y las tecnologías IoT seleccionadas en la fase de análisis.
- Construir y codificar el prototipo IoT, incluyendo la aplicación móvil de control local y la aplicación multiplataforma de gestión de horarios.
- Realizar una prueba piloto del prototipo IoT.

1.4 Delimitaciones

1.4.1 Temporal

El desarrollo de este proyecto tendrá como periodo de duración 6 meses, a partir de la aprobación del formato 1.

1.4.2 Espacial

El proyecto de investigación se desarrollará en el municipio de Aguachica departamento del Cesar y se tomarán dos instituciones educativas de carácter oficial en las cuales se realizará una implementación de prueba.

1.4.3 Contextual

El municipio de Aguachica está ubicado al sur del Departamento del Cesar de la geografía colombiana, con una población de 93.917 habitantes. Aguachica está en una posición geográfica estratégica. Conecta el Caribe colombiano con los Santanderes a través de la Ruta del Sol y es un punto clave de comunicación para la región del Catatumbo por la Ruta Nacional 70. La ciudad también cuenta con un aeropuerto, transporte ferroviario y fluvial a través del puerto de Gamarra, lo que la hace accesible desde diferentes puntos de Colombia.

2. Marco referencial

2.1 Marco histórico

Hoy en día la IoT está tomando posesión de todos los artefactos que cumplen con algún tipo de funcionalidad con el fin de potencializarlo y mejorar su productividad en diversos sectores de la industria y en la vida cotidiana. A razón de esto vemos prototipos automatizados que permiten la manipulación de este desde cualquier parte del mundo, brindando accesibilidad para la configuración del componente de manera virtual (Bonilla-Fabela et al., 2016).

A nivel internacional el concepto de IoT se ha adoptado rápidamente en las instituciones educativas, esto permite automatizar procesos en las instituciones y dar visión a los estudiantes para que adquieran conocimientos de IoT. En Perú, Catacora et al. (2020) desarrollaron un sistema IoT que permite gestionar el consumo de energía en la universidad privada de Tacna mediante dispositivos inteligentes que controlan la luminosidad en las instalaciones de acuerdo con el movimiento y la cantidad de luz natural; según el informe esto permite un ahorro de un 44% de gastos energéticos en dichas instalaciones.

Asimismo, en el ámbito nacional las escuelas han optado por incluir tecnologías IoT que le permitan facilitar el proceso de enseñanza y aprendizaje de sus estudiantes. En la

Universidad Autónoma de Bucaramanga el ingeniero de sistemas Rafael David Vergara Herrera desarrollo una investigación y prototipo para optar por el título de maestría titulada “ASISTENTE EDUCATIVO INTELIGENTE PARA LA UTILIZACIÓN DE TECNOLOGÍAS IOT APLICADAS EN LAS AULAS DE CLASE DE LAS INSTITUCIONES EDUCATIVAS DE COLOMBIA” (Vergara, 2021). El objetivo de este proyecto fue determinar y analizar la influencia de un Asistente Educativo Inteligente (AEI) en el mejoramiento de la competencia razonamiento lógico en estudiantes de un plantel educativo del departamento de Córdoba; hicieron parte 22 estudiantes de Grado 11º de educación secundaria, mediante la metodología cuantitativa, con diseño preexperimental.

Es importante resaltar que no solamente se requiere implementar IoT en los procesos objetivos en las escuelas sino en aquellos que generan beneficios que impulsen el buen desarrollo de las jornadas. En la universidad Piloto de Colombia, el estudiante de ingeniería de sistemas Santiago Amézquita Ayure para optar por su título profesional desarrollo una investigación y prototipo titulado “Prototipo de sistema de monitorización y regulación automática de material particulado basado en enfoques de IoT para entidades educativas ubicadas en entornos hostiles de alto flujo vehicular. Hacia el aumento de la calidad del aire en las instituciones educativas” (Amézquita Ayure et al., 2022). El objetivo de este proyecto es realizar un prototipo que permita la visualización y regulación de concentraciones indoor de material particulado en instituciones educativas.

En el ámbito local en el departamento del Cesar se ha adoptado rápidamente la utilización de tecnologías IoT para automatizar procesos; en la ciudad de Aguachica, el Centro Agroempresarial SENA en conjunto con la Universidad Popular del Cesar Seccional Aguachica trabajaron sobre un prototipo que permite monitorear variables agroambientales con el fin de mejorar la productividad agropecuaria (Trillos et al., 2022).

Teniendo en cuenta los proyectos mencionados se demuestra el gran potencial que tiene el uso de IoT para la automatización de procesos a nivel general. Esto servirá como punto apoyo para lograr la creación del prototipo automatizado de sirena para el cumplimiento de la jornada escolar en instituciones educativas.

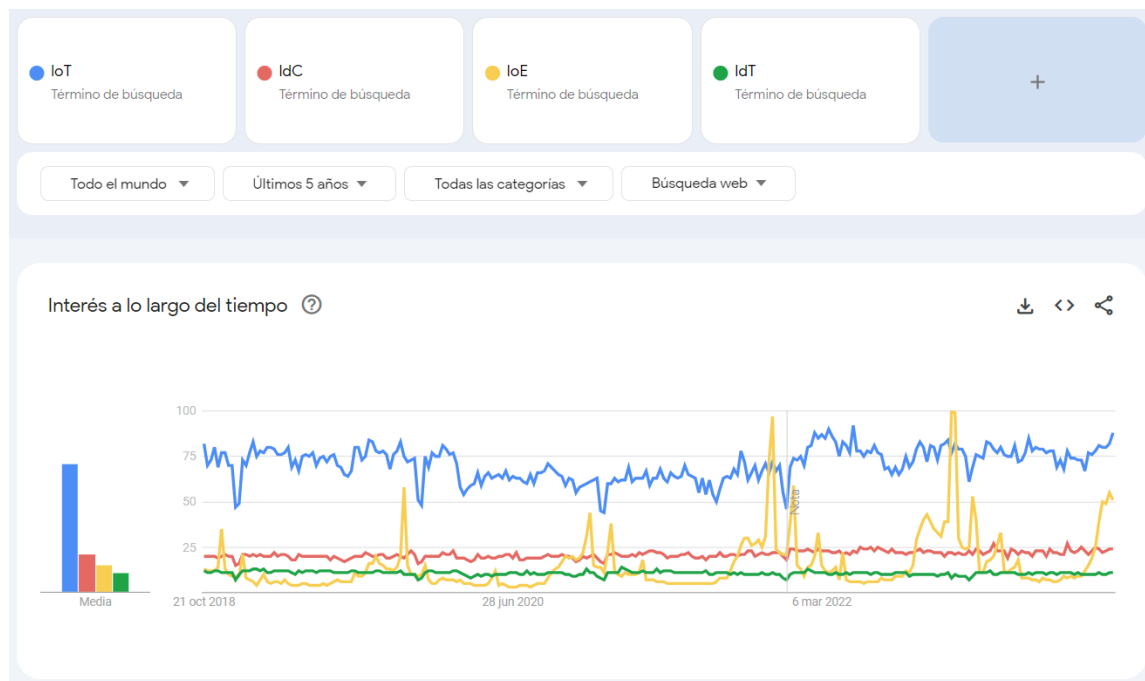
2.2 Marco teórico

Con el fin de identificar y conocer los diferentes elementos teóricos requeridos para el desarrollo del proyecto y los conceptos técnicos que permitieron su diseño y construcción, se hace necesario la contextualización del paradigma IoT, las arquitecturas, y los elementos que componen un objeto IoT, las cuales generan bases sólidas para la investigación.

2.2.1 Internet de las cosas

Para responder a una definición sobre que es Internet de las cosas es recomendado establecer la denominación a utilizar, iniciemos con que Internet de las Cosas y su acrónimo (*IdC*), es la traducción de *Internet of Things* y su acrónimo (*IoT*), de igual manera se presenta el termino relacionado Internet de Todo (*IdT*), traducido de *Internet of Everything* (*IoE*), con los cuales haciendo una revisión de los diferentes términos en busca del más utilizado, en la figura 1 generada con la herramienta *Google Trends* muestra que el uso del término IoT está mayormente generalizado, por lo tanto de igual manera, aunque el documento esta escrito en español utilizaremos como acrónimo para referirnos a Internet de las cosas **IoT**.

Figura 1. Comparación del uso de los términos IdC, IoT, IoE, IdT en el mundo entre el 2018 y 2023



Fuente: Generada por el autor en Google Trends

Debido a la evolución actual del IoT, se presenta una definición mucho más amplia apoyada en la definición dada por uno de los altos ejecutivo de Cisco Systems en el 2012 en la que aclar las diferencias con el IoE.

Internet de las Cosas (IoT) es un conglomerado de tecnologías que permiten conectar a internet objetos físicos, con la capacidad de interactuar con el entorno, dichas capacidades permiten tomar decisiones y comunicarse con otros objetos y sistemas inclusive personas y seres vivos.

Internet de Todo (IoE) se puede decir que define el ecosistema completo para IoT y, según Evans, (2012) “*IoE reúne personas, procesos, datos y cosas para conseguir que las conexiones en red sean más relevantes y valiosas que nunca. convertir la información en acciones que creen nuevas capacidades y experiencias más ricas,*

oportunidades económicas sin precedentes para empresas, personas y países”.
(pág. 3)

2.2.2 Impacto y beneficio del IoT

Desde la aparición por primera vez del término IoT en 1999 por Kevin Aston, Se puede afirmar que empiezan a hacerse notorios los cambios en la industria clásica lo que empieza a sentar las bases para la transformación social incorporando nuevos métodos de producción apoyados en las tecnologías de la información y la comunicación, propiciando la llegada de la cuarta revolución industrial o más llamada *Industria 4.0*.

Los cambios que promulga y que evidentemente está produciendo vivir en la era de la cuarta revolución industrial, en todos los ámbitos de la sociedad; empresarial, gubernamental y en la vida cotidiana de las personas, en gran parte son consecuencia de la integración de las tecnologías IoT en la realización de las actividades, que con lleva a generar nuevos modelos de negocio concluyendo con una real transformación digital, donde siempre al final el principal beneficiado será el ciudadano.

El IoT ha cambiado nuestros hábitos, nos hace más productivos y eficientes, lo cual por razones intrínsecamente relacionadas también aumenta la productividad del sector empresarial. hoy podemos decir que las cosas han evolucionado, desde la aparición de la electrónica, se inicia una avalancha por la construcción de artefactos y maquinas que nos permitieran tener procesos de supervisión, automatización y control más autónomos, para finalmente hoy en día hablar de artefactos inteligentes. A manera de ejemplo mencionar el caso de la **domótica**, la cual consiste en la automatización y control remoto del hogar con la intención de tener medidas de supervisión con sensores para detectar intrusos, situaciones de riesgo como incendios, inundaciones, control sobre la iluminación, artefactos, accesos de puertas ventanas, sistemas de jardinería, entre otras posibilidades para mejorar el confort de la vivienda. Con las tecnologías IoT se empieza acuñar el termino de casas inteligentes,

en las cuales se pueden agregar innovaciones donde el sistemas de riego del jardín, aunque, se encuentre programado para activarse a determinada hora, sean capaces de terminar si este realmente es necesario en situaciones como la posibilidad de que ocurran lluvias, su intensidad, la humedad de la tierra, información obtenida a través de sensores que envían información a aplicaciones y confrontarla con la información de los servicios meteorológicos disponibles en Internet. Estas innovaciones generan efectos positivos al medio ambiente, optimizan recursos energéticos y económicos.

Un ejemplo más, es el uso de dispositivos wearables como las Smart band, hace algunos años era necesario acudir a un centro de salud o disponer de equipos médicos específicos para supervisar o medir nuestros signos vitales y conocer que tan bien está funcionando nuestro cuerpo, sin embargo, hoy ya es posible llevar registros en tiempo real de nuestra frecuencia cardiaca, presión arterial, niveles de oxígeno en la sangre, los cuales son enviados a internet, para que puedan ser consultado por nuestro médico, inclusive generar alertas que permitan tomar medidas de precaución, incluso, hay sistemas que pueden realizar llamadas a los sistemas de urgencias, y enviar la información para que el personal de salud se aliste según la situación presentada.

2.2.1 ISO/IEC 30141

La presente norma ISO/IEC del 2018 fue realizada estrictamente para el diseño y desarrollo de prototipos IoT. ISO (Organización Internacional de Normalización) es una organización para la creación, gestión y diseño de normativas internacionales. En este caso en particular no solo ISO aportó metodológicamente para el desarrollo de la norma sino también IEC. Tal como lo indica la norma ISO en su prefacio: La Organización Internacional de Normalización (ISO) y la Comisión Electrotécnica Internacional (IEC) trabajan juntas para establecer normas mundiales. Los países que son miembros de ISO o IEC participan en la creación de estas normas a través de comités técnicos especializados en áreas específicas. Los comités técnicos de ISO e IEC trabajan juntos en temas de interés mutuo. Otras organizaciones,

tanto gubernamentales como no gubernamentales, también colaboran con ISO y IEC. En el ámbito de la tecnología de la información, ISO e IEC tienen un comité técnico conjunto llamado ISO/IEC JTC 1 (ISO, 2018).

Profundizando en las normas IEC, ellas establecen reglas y definiciones para que los ingenieros y diseñadores puedan comunicarse eficazmente hablando todo el mismo idioma. Estas normas se aplican en la fabricación, prueba, ensayo y certificación de productos eléctricos y tecnológicos. De acuerdo con lo descrito anteriormente, es importante el uso de sus reglas para la creación del prototipo electrónico.

2.2.2 Arquitectura de 5 Capas

La arquitectura de sistemas IoT (Internet de las cosas) se utiliza para diseñar y desarrollar soluciones que permiten la conexión, interacción y comunicación entre dispositivos y servicios de manera eficiente y segura. Esta arquitectura es esencial para construir sistemas IoT escalables y confiables, ya que se trata de una red compleja de dispositivos y servicios interconectados que deben trabajar juntos de manera efectiva (González, 2013).

Algunas de las funciones que puede desempeñar la arquitectura de sistemas IoT incluyen:

- Gestión de dispositivos: La arquitectura de sistemas IoT permite la conexión y la gestión de dispositivos, incluyendo sensores, actuadores y dispositivos de red.
- Recopilación de datos: Los sistemas IoT pueden generar grandes cantidades de datos, y la arquitectura de sistemas IoT ayuda a recopilar, almacenar y procesar estos datos de manera efectiva.
- Análisis de datos: La arquitectura de sistemas IoT puede ayudar a analizar los datos recopilados para extraer información valiosa y mejorar la eficiencia de los sistemas.
- Seguridad: La arquitectura de sistemas IoT puede ayudar a garantizar la seguridad y la privacidad de los datos y los dispositivos.
- Interoperabilidad: La arquitectura de sistemas IoT permite que los dispositivos y los servicios interactúen y se comuniquen entre sí de manera efectiva, lo que puede mejorar la eficiencia y la eficacia de los sistemas.

Teniendo en cuenta lo anterior, se podría destacar que la arquitectura de sistemas IoT es esencial para diseñar, implementar y mantener sistemas IoT confiables y escalables. Por ello consta de 5 capas, como lo son: la capa de percepción, capa de red, capa de procesado, capa de aplicación y capa de negocio. Las anteriores capas permiten mantener bien estructurada la arquitectura IoT (Ortiz, 2019).

2.3.3 Cloud Computing para Internet de las cosas (IoT)

La computación en la nube, también conocida como Cloud Computing, es una tecnología que se utiliza para ofrecer servicios a través de Internet. Esta tecnología se basa en servidores que están disponibles desde cualquier lugar y en cualquier momento, lo que facilita el acceso a servicios y datos (Ortiz Monet, 2019b). Ese grupo de servidores están alojados en lugares específicos en todo el mundo llamados *Datacenters*, la ubicación y el tipo de servicio dependen del proveedor de servicio en la nube.

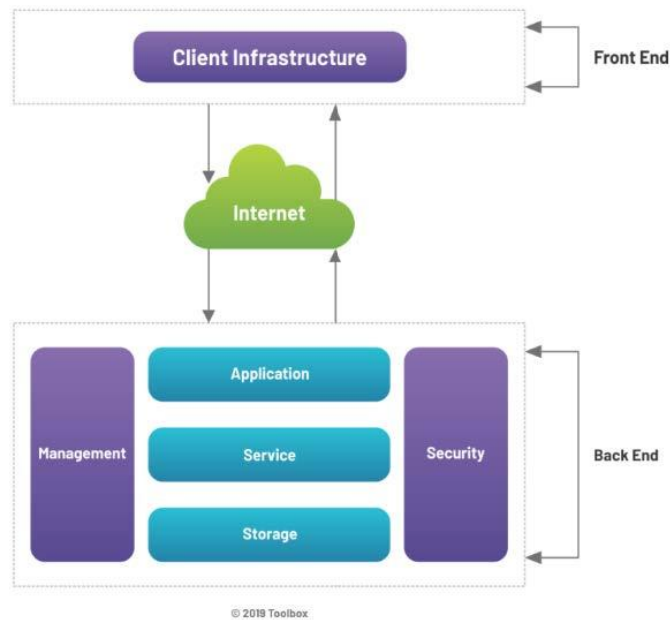
En el contexto del Internet de las Cosas (IoT), la computación en la nube se ha convertido en una herramienta fundamental. Gracias a ella, se pueden almacenar y procesar grandes cantidades de datos generados por los dispositivos IoT, lo que permite realizar análisis más precisos y tomar decisiones más informadas. Además, la computación en la nube ofrece una gran flexibilidad y escalabilidad, lo que significa que los servicios y aplicaciones pueden adaptarse a las necesidades cambiantes de los usuarios y de los dispositivos IoT, sin requerir una gran inversión en infraestructura y recursos de TI.

Otra ventaja importante de la computación en la nube para el IoT es la posibilidad de acceder a los datos y servicios desde cualquier lugar y en cualquier momento. Esto facilita el desarrollo de soluciones IoT para diferentes industrias y ámbitos, desde la agricultura y la industria manufacturera hasta la salud y el hogar inteligente.

2.3.3.1 Arquitectura de Cloud Computing

La arquitectura de Cloud Computing se divide en dos partes fundamentales las cuales se conectan mediante internet, esas partes son el Front End y el Back End como se aprecia en la Figura 2.

Figura 2. Arquitectura de Cloud Computing



Fuente: (Ortiz, 2019b)

Arquitectura front-end: se refiere a la parte de la arquitectura centrada en el usuario final, que incluye:

- Software e interfaz de usuario que se ejecutan en el dispositivo del cliente y son visibles y utilizados por el usuario.
- El dispositivo del cliente no requiere una gran capacidad de procesamiento, ya que la mayoría del procesamiento se realiza en la nube.

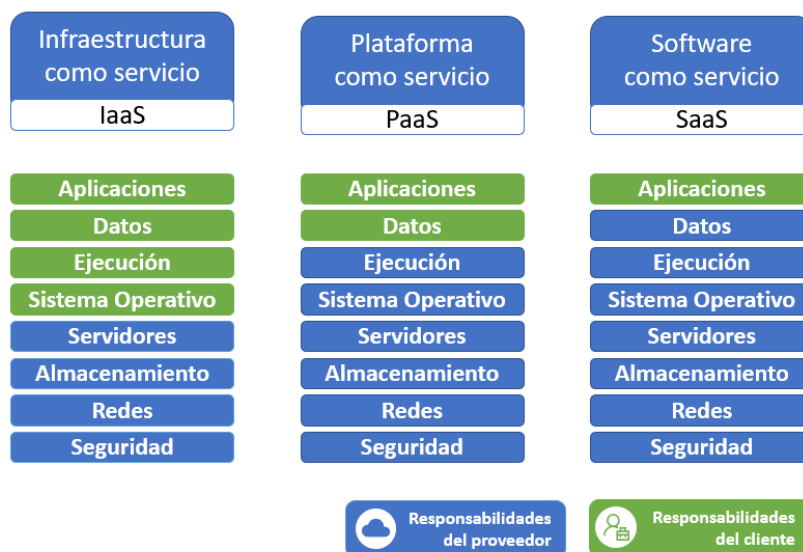
Arquitectura back-end: es la responsabilidad del proveedor del servicio en la nube y respalda la arquitectura front-end. Está compuesta por varios elementos:

- La aplicación proporcionada al usuario final, donde se coordinan las necesidades del cliente con los recursos en la parte back-end.
- Los servicios que realizan tareas de computación en la nube, como almacenamiento, servicios web o entornos de desarrollo de aplicaciones.
- El almacenamiento de datos necesario para la ejecución del software en la nube.
- Los servicios de administración de recursos que aseguran un funcionamiento fluido del sistema de manera eficiente.
- Las medidas de seguridad para proteger al servidor contra pérdidas de datos o ataques, incluyendo copias de seguridad del almacenamiento para garantizar el correcto funcionamiento del sistema.

2.3.3.2 Modelos de servicios de Cloud Computing

Cloud Computing se compone de diferentes capas y componentes que trabajan juntos para ofrecer servicios y recursos de TI de manera eficiente y escalable (Manuel et al., 2010). Como lo indica la Figura 3, Estas capas se dividen en 3 infraestructuras las cuales indican cual es la responsabilidad del proveedor de servicios en la nube y las responsabilidades del administrador de la nube.

Figura 3. Infraestructuras en la nube



2.3.3.3 Modelos de Implementación de Cloud Computing

En el mundo del Cloud Computing, existen varios modelos de implementación que ofrecen diferentes enfoques para satisfacer las necesidades específicas según los requerimientos del proyecto. Estos modelos incluyen la nube pública, la nube privada y la nube híbrida. Cada modelo de implementación tiene sus propias ventajas y consideraciones, y la elección adecuada depende de factores como la seguridad, el control, la escalabilidad y los requisitos específicos de la organización.

- **Nube pública (Public Cloud):** En este modelo, los proveedores de servicios en la nube ofrecen recursos informáticos, como servidores virtuales, almacenamiento y aplicaciones, a través de Internet de forma pública. Los usuarios pueden acceder a estos recursos de manera compartida, lo que implica que los servidores y la infraestructura subyacente son gestionados por el proveedor de la nube. La nube pública es altamente escalable, flexible y rentable, ya que los usuarios pagan únicamente por los recursos que consumen. Ejemplos populares de proveedores de nube pública incluyen Amazon Web Services (AWS), Microsoft Azure y Google Cloud Platform.
- **Nube privada (Private Cloud):** En este modelo, la infraestructura en la nube es exclusiva de una sola organización. La nube privada puede ser gestionada y mantenida por la organización misma o por un proveedor de servicios externo. Los recursos en la nube privada pueden estar ubicados en las instalaciones de la organización (on-premises) o en un centro de datos externo. La nube privada ofrece un mayor control, seguridad y personalización, ya que los recursos no se comparten con otros usuarios o empresas. Esto es especialmente relevante para organizaciones que tienen requisitos estrictos de cumplimiento normativo o necesitan mantener datos sensibles en un entorno altamente controlado.
- **Nube híbrida (Hybrid Cloud):** Este modelo combina la nube pública y la nube privada, permitiendo a las organizaciones utilizar ambos entornos de manera integrada. La

nube híbrida proporciona flexibilidad y escalabilidad, ya que las organizaciones pueden aprovechar los recursos de la nube pública para cargas de trabajo variables o de pico, mientras mantienen datos sensibles o aplicaciones críticas en la nube privada. Además, la nube híbrida facilita la migración gradual de aplicaciones y datos entre los entornos de nube, lo que permite una transición más suave y controlada.

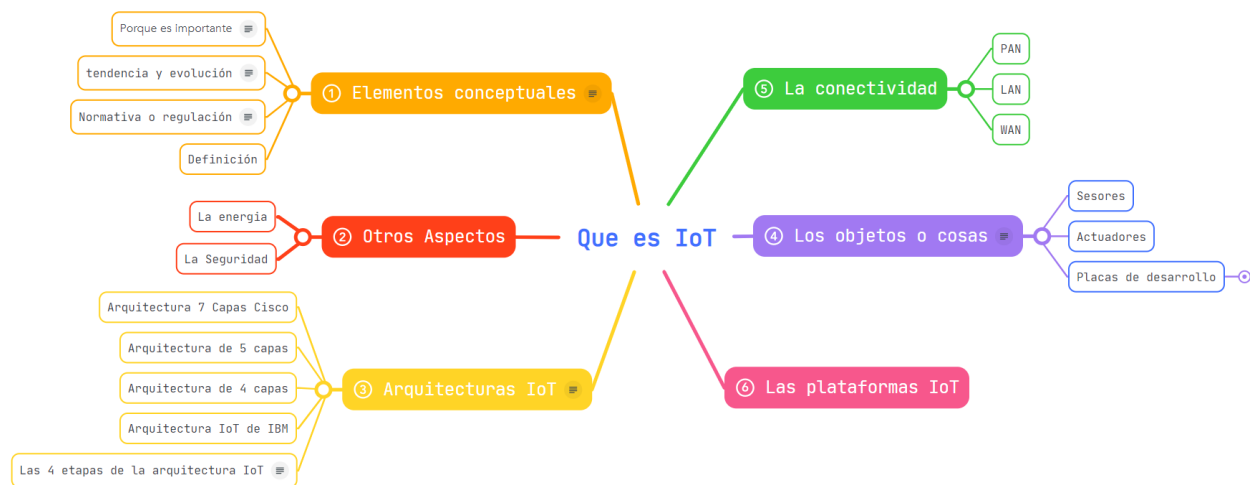
Estos modelos de implementación en el Cloud Computing ofrecen diferentes opciones para satisfacer las necesidades y requisitos específicos de cada proyecto u organización (Microsoft, 2023).

2.3.5 Metodología de prototipado

La metodología se enfoca en la generación rápida de modelos iniciales con el fin de explorar diversas alternativas de diseño y recibir comentarios de los usuarios. El libro presenta varios ejemplos que ilustran cómo se aplica esta metodología y brinda consejos prácticos para su ejecución (Bill Buxton, 2007).

Finalmente, con la figura 4 se pretende orientar al lector sobre la estructura general utilizada para abordar los aspectos teóricos necesarios en el desarrollo del presente trabajo de investigación, así mismo para aquellos que deseen incorporar en la construcción de prototipos IoT.

Figura 4. Diagrama marco teórico



2.3 Marco legal

Para la construcción del prototipo IoT es necesario tener en cuenta de la normativa Colombia las siguientes leyes:

Constitución Política de Colombia.

Ley 1581 de 2012: La ley de Habeas Data es la principal norma que regula la protección de datos personales y la privacidad en Colombia. En el contexto de un proyecto de grado que involucre el desarrollo de un software y un prototipo IOT, es importante cumplir con las disposiciones legales correspondientes en relación con la recolección, uso y almacenamiento de datos personales.

Ley 1273 de 2009: Establece las normas y regulaciones que se aplican a la seguridad informática y la protección de la información. Es importante garantizar que el software y el prototipo IOT desarrollados cumplan con estas disposiciones legales para proteger los datos y prevenir posibles ciberataques.

Ley 603 de 2000: La ley de licencias de Software establece las normas y procedimientos para el licenciamiento de software en Colombia, con el objetivo de proteger los derechos de autor y fomentar el uso de software legal en el país. Ya que el prototipo contará con una

aplicación móvil que controlará la información es necesario conocer los procedimientos para el licenciamiento de software.

2.4 Marco conceptual

En este marco se encuentran definidos un conjunto de términos en los cuales tienen importancia en el proyecto.

Industria 4.0: El término industria 4.0 se refiere a un nuevo modelo de organización y de control de la cadena de valor a través del ciclo de vida del producto y a lo largo de los sistemas de fabricación apoyado y hecho posible por las tecnologías de la información. **(del Val Román, 2016)**

IoT: Es la interconexión digital de objetos cotidianos con internet. Permite el cambio automático de información con otros dispositivos o centros de control sin intervención humana, capturando gran cantidad de información clave sobre uso y rendimiento. **(Garzón, 2019)**

Protocolo MQTT: El protocolo MQTT (Message Queue Telemetry Transport) se trata de un protocolo de mensajería asíncrona, usado en el ámbito de las comunicaciones machine-to-machine (M2M), en el campo de IoT(Internet de las cosas). Su funcionamiento se basa en el intercambio de mensajes mediante el modelo de publicación y suscripción. **(Mahedero Biot, 2020)**

Broker MQTT: El broker MQTT se trata del servidor encargado de la distribución de los mensajes a los receptores. El broker recibe los mensajes, hace un chequeo del topic al que están suscritos y los encamina hacia los clientes suscritos a este topic. El broker nos brinda la posibilidad de que los mensajes sean persistentes, es decir, que se guarden los mensajes hasta que el cliente al que va dirigido se conecte. **(Mahedero Biot, 2020)**

Topic: El topic es el tema del mensaje, es decir, a quien va dirigido ese mensaje o de quien queremos recibir tal mensaje. El topic sirve como identificador al broker para poder encaminar cada uno de los mensajes recibidos hacia los clientes adecuados que también se hayan suscrito al topic. **(Mahedero Biot, 2020)**

Hosting: El alojamiento web o hosting web es el servicio que permite cargar los archivos de un determinado sitio web en un servidor y proveer los recursos necesarios para ponerlo en marcha y disposición del público en general las 24 horas del día. **(Arias, 2023)**

ESP32: El ESP32 es un microcontrolador de bajo coste creado por Espressif Systems. Destaca debido a que posee tecnologías WiFi y Bluetooth integradas, dos núcleos de procesamiento y tecnología de bajo consumo, que, en relación con su precio, lo hacen un dispositivo muy recomendable para desarrollar aplicaciones enfocadas a Internet de las Cosas. **(Benito Herranz, 2019)**

Puerto Serial: Es una interfaz de comunicación que permite la transferencia de datos de manera secuencial, es decir, bit por bit, a través de un solo cable.

Bluetooth: Bluetooth es una propuesta de especificación de radiofrecuencia por transmisión de corto alcance, transmisión de datos de punto a multipunto. Bluetooth puede transmitir a través de objetos sólidos no metálicos. Su cable de unión nominal es de 10 cm a 10 m en teoría, pero se puede extender a 100 m mediante el incremento de transmisión de energía. **(Vergara, 2008)**

Sensor: Un sensor es un dispositivo capaz de detectar magnitudes físicas o químicas, que son llamadas variables de instrumentación y transformarlas a señales o variables eléctricas, las variables de instrumentación pueden ser: temperatura, intensidad,

distancia, aceleración, inclinación, desplazamiento, fuerza, humedad, presión, torsión, etc. (Reyes-Flores, 2019)

Nube: la nube es un relativamente nuevo modelo que consiste en el acceso a través de Internet a todo tipo de servicios y recursos informáticos ofrecidos por una red de servidores destinados a ese fin. (Gutiérrez, 2018)

Edge Compute: La computación de borde o Edge Computing apareció alrededor de 2002 y se refiere al paradigma que permite ubicar los recursos de computación en el borde de la red, de modo que la computación ocurra cerca de las fuentes de datos. (Mela, 2021)

SBC (single board computer) o placa de circuitos reducida: Es un dispositivo el cual integra todas sus partes en una sola placa procesador, memoria RAM, conectores, etc (Chicaiza Pareja, 2017)

Protocolo de comunicación: Son un conjunto de reglas y pautas que permiten a los usuarios de la informática enviar un bloque de datos de una ubicación a otra. Se utilizan fundamentalmente para determinar el formato y la transmisión correcta de los datos, tanto en las comunicaciones analógicas como digitales (Lifeder, 2020)

Fog Computing: Es una arquitectura emergente para computar almacenamiento, control y servicio de red que distribuye estos servicios a usuarios finales entre Cloud y las cosas (Rodríguez et al., 2017)

3. Diseño metodológico preliminar

3.1 Tipo de Investigación:

Para el desarrollo del proyecto se plantea un enfoque cualitativo, debido a la intención de los autores de resolver una problemática concreta en las instituciones educativas, tomando

como caso de estudio una institución educativa de Aguachica Cesar, en la cual mediante métodos inductivos se buscan identificar los requerimientos del prototipo IoT los cuales se convertirán en el insumo para su construcción. La información recolectada no requiere ser de tipo estadística, sin embargo, la experiencia de los autores jugará un papel importante en el análisis, reflexión e interpretación de la información por su relación con el contexto objeto de estudio.

Como el objetivo del proyecto es desarrollar un prototipo tanto hardware y software de sirena automatizada utilizando tecnologías internet de las cosas (IoT) para lo cual es necesario aplicar los conocimientos recibidos en asignaturas del plan de estudio del programa de ingeniería de sistemas de la Universidad Popular del Cesar Seccional Aguachica, además de incursionar en la tendencia del IoT. Contribuyendo con el crecimiento del cuerpo del conocimiento base del desarrollo curricular, y queriendo estar a la vanguardia de nuevas tecnologías, y ser capaz de adaptar cualquier amenaza, debilidad o riesgo que se presente, con el fin de gestionar estrategias para mantenerse y proyectarse en la sociedad. (Molero et al., 2019). Por lo anterior se enmarca en la investigación aplicada con el propósito de fortalecer y consolidar el cuerpo del conocimiento del área implicada en el estudio, con fines inmediatos y determinados en búsqueda de nuevos conocimientos técnicos con una aplicación inmediata a cierto problema ya definido.

3.2 Población y Muestra

3.2.1 Población

El desarrollo del prototipo está dirigido a las instituciones educativas de carácter pública en los niveles de básica primaria, secundaria y media técnica de Colombia, las cuales se rigen por lineamientos del Ministerio de Educación Nacional (MEN), lo que permite en marcarlas en una población con similares características, donde los resultados del proyecto aportaran beneficios al control de la jornada escolar.

3.2.2 Muestra

Para el caso de los estudios con enfoque cualitativo, las muestras no se consideran de tipo estadísticos, la recolección de la información se realiza de manera focalizada por la experticia del investigador u orientada por las dinámicas del desarrollo del proyecto, por tal motivo se consideran inicialmente a los coordinadores y rector de la institución educativa tomada como caso de estudio.

3.3 Instrumentos de recolección de datos

En este proyecto se recurre a las técnicas descriptivas como la entrevista y el cuestionario para la recolección de los datos necesarios para establecer los requerimientos y especificaciones para la construcción del prototipo IoT, su aplicación estará dirigida los sujetos responsables por el proceso de inspección y vigilancia de la jornada escolar.

Los resultados serán consolidados en un documento llamado requerimientos y especificaciones para el prototipo IoT, y diagramados con la técnica de casos de usos, mediante un análisis descriptivo de los datos recopilados.

3.4 Procedimientos o métodos de desarrollo

La investigación de enfoque cualitativo es bastante rigurosa en los métodos y teorías que orientan su desarrollo, en la búsqueda de un método apropiado para el desarrollo del proyecto, se selecciona y adapta la metodología de prototipado experimental mediante tareas y productos para cada una de las 5 fases mostradas en la figura 5 (Navarro et al., 2021)

Figura 5. Metodología de Prototipado Evolutivo Experimental



Fuente: (Navarro Pino et al., 2021)

Fase de Identificación y Requerimiento: En esta etapa se define el alcance del prototipo hardware y software, elaborando una lista de funcionalidades donde se detallan los requisitos funcionales como no funcionales, además, se describen las restricciones.

Fase Diseño: En la etapa de selección del modelo o arquitectura para proyectos de IoT, se suelen estructurar los modelos en niveles o capas, similar al modelo OSI o TCP/IP, con el objetivo de establecer los elementos que lo conforman y lograr la interoperabilidad. Algunos modelos de referencia ampliamente reconocidos son: ITU, Pragma, ARM del Foro Internacional de IoT, Microsoft Azure, Google Cloud IoT y IoT Intel. Sin embargo, no hay consenso sobre qué modelo utilizar, por lo tanto, estos se pueden generalizar en arquitecturas de 3, 4 o 5 capas, siendo la de 4 capas la más comúnmente empleada. En esta fase también se debe definir los componentes del prototipo IoT y, a partir de ello, seleccionar los protocolos y tecnologías apropiadas para su construcción.

Fase Construcción: se construyen y programan los dispositivos, se configuran las herramientas y equipos para la conectividad, se realiza el montaje o la configuración de la

plataforma IoT, programar las aplicaciones y funcionalidades para los clientes o usuarios finales.

Fase Pruebas: estas son realizadas por parte del usuario final sobre el prototipo en un entorno de producción controlado, con el propósito de validar los requerimientos iniciales y concretar nuevos requerimientos que permitan realizar ajustes para optimizar el funcionamiento.

Fase Optimización: se inicia un proceso iterativo propuesto por la metodología, que parte del rediseño para refinar los requerimientos iniciales o nuevos, como resultado del testeo realizado por el usuario final en la fase de pruebas, adicional a esto le permite al equipo desarrollador mejorar el conocimiento acerca del sistema.

Fase Prueba de Ingeniería: en la adaptación de la metodología del ciclo de vida del prototipo, se considera como una fase de una única tarea o actividad, la cual consiste en entregar el producto final de ingeniería refinado por parte del cliente para satisfacer sus necesidades y que es susceptible para iniciar su proceso de producción en serie. Esta es considerada relevante en el desarrollo de prototipos IoT que definen una arquitectura de 5 capas. Debido al alcance del proyecto esta fase no será tomada en cuenta.

4. Resultados

4.1 Identificación de los requerimientos y las tecnologías del Internet de las Cosas aplicables al prototipo

La identificación de Tecnologías IoT consolidadas, en el marco del primer objetivo Específico, se llevó a cabo mediante una búsqueda sistemática de literatura en bases de datos académicas como Google Académico, ScienceDirect, Dialnet, SClelo, repositorios de trabajos de grado de diversas universidades, de igual manera se revisaron informes técnicos

y documentación de Fabricantes. Para organizar y clasificar los resultados se propuso la estructura mostrada en la figura 6 de los componentes de un sistema IoT basado y ajustado de la propuesta de Romo & Villalobo (2019), tambien se consideraron las sugerencias de Navarro et al., (2022).

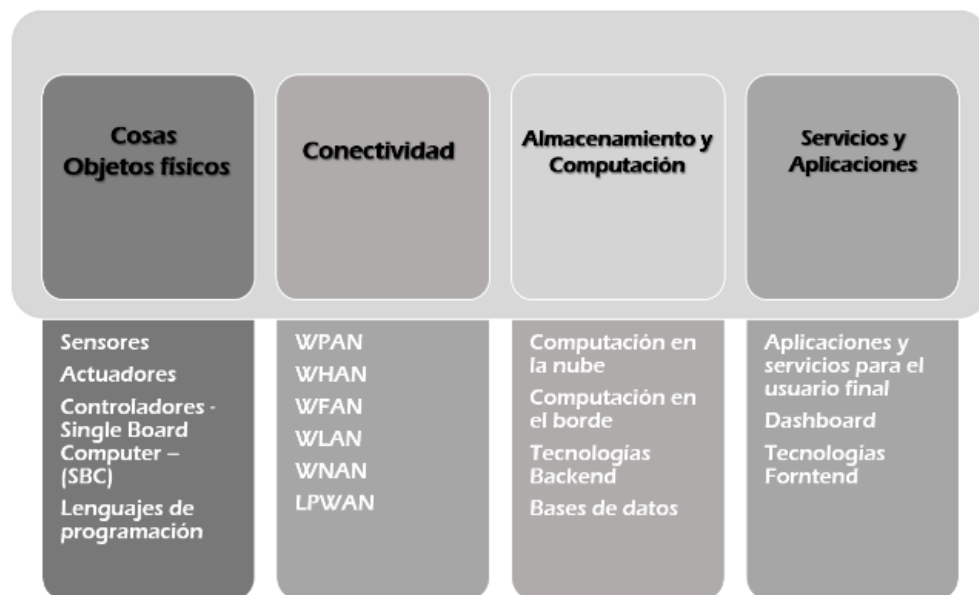


Figura 6. Componentes de un sistema IoT

Tecnologías para el desarrollo de los objetos físicos: aquí se agrupan las tecnologías que permiten la construcción y programación del hardware que permite la adquisición de los datos y la ejecución de ordenes o presentación de la información al entorno o al usuario final. Estas tecnologías pueden variar o subclasificarse según el ámbito para el cual se va a esta dirigida la solución (industrial, domestico o cotidiano y maker).

Placas de desarrollo Single Board Computer (SBC) y unidades de microcontrolador (MCU): es una placa de circuito electrónico con un microprocesador que incorpora internamente componentes similares a un computador en un solo chip, aunque de menores prestaciones, dicha placa facilita la conexión de sensores, actuadores y tarjetas de comunicación, entre las más utilizadas para prototipar dispositivos IoT a

bajos costos se encuentra; Arduino, Raspberry Pi, DevKit Esp32, NodeMCU, entre muchas.

Sensores: son elementos hardware que se conecta a la placa de desarrollo por medio de sus pines de entrada para recolectar datos de variables del entorno, entre ellos se encuentran sensores de proximidad, acelerómetro, giroscopio, sensores de humedad, temperatura, presión, entre muchos, inclusive existen una categoría llamada sensores inteligentes los cuales integran un procesador.

Actuadores: son elementos hardware que se encargan de realizar las acciones siguiendo las ordenes de control enviadas desde el microprocesador de la placa de desarrollo, en la mayoría de los casos los actuadores no se pueden conectar directamente a la placa de desarrollo, por lo que se requiere que las señales eléctricas de control enviadas por el microcontrolador sean adaptadas a una señal de mayor potencia y así controlar actuadores como motores, válvulas, indicadores luminosos, entre muchos.

Lenguajes de programación: en este caso específico, es el lenguaje utilizado para indicarle al hardware (Placa de desarrollo, sensores, actuadores), cual debe ser su comportamiento o funcionamiento de los circuitos electrónicos, entre ellos están lenguajes de alto nivel como Python, C, C++, Processing, sin embargo, al momento de escribir el código es recomendado utilizar los entornos de desarrollo integrado (IDE), uno de los mas populares es el IDE de Arduino por ser un ecosistema completo para desarrollo de prototipos IoT, pero existen otras alternativas como Visual Studio Code que es un editor de código fuente multilenguaje y multiplataforma.

Tecnologías para la conectividad de los objetos: en relación a los componentes necesarios para establecer la conectividad a internet y con otros dispositivos cercanos, se revisaron tecnologías de redes de área personal inalámbricas (**WPAN**), redes área local inalámbricas (**WLAN**), las redes inalámbricas área amplia y de baja potencia (**LPWAN**), entre otras. Aunque como ejemplo solo se han mencionado tecnologías inalámbricas, no quiere decir

que las tecnologías cableadas sean excluidas del IoT. Los estándares inalámbricos existente en el mercado son abundantes, aunque dirigidos a usuarios con necesidades específicas, y que son seleccionadas basadas en criterios como cobertura, alcance, tasa de transferencia o ancho de banda, consumo energético, banda de frecuencia, costo y disponibilidad embebida en SBC, (Ulloa-Vásquez, Carrizo, & García-Santander, 2022). Entre los cuales se identifican los siguientes estándares:

Wireless Personal Area Network (WPAN): Bluetooth, MiWi, ANT+

Wireless Home Area Network (WHAN): ZigBee, Z-Wave.

Wireless Local Area Network (WLAN): Wi-Fi (802.11 a/b/g/n/ac/ax).

Wireless Neighborhood Area Network (WNAN): Wi-SUN, ZigBee NAN.

Wireless Wide Area Network (WWAN): 3G, 4G, 5G, LTE, WiMAX

Low Power Wide Area (LPWAN): SIGFOX, LoRa.

Tecnologías para el almacenamiento y procesamiento de los datos: Dentro de los componentes para el almacenamiento y procesamiento de los datos, lo más común es que esto se haga en la nube, ya sea pública o privada. Aquí cobran relevancia tecnologías como el cloud computing y Edge computing. Esta última se ha convertido en una tendencia dentro del IoT al permitir minimizar la carga de datos enviada hacia internet y de esta manera optimizar el procesamiento en la nube y la tasa de transferencia, que permiten bajar los costos económicos al contratar servicios bajo demanda, al mismo tiempo que generan beneficios al medio ambiente (Hewlett Packard Enterprise).

Dentro de tecnologías y herramientas identificadas para ser utilizadas como componente de almacenamiento y procesamiento, enmarcada en la capa intermediaria se revisaron: Mosquitto como broker MQTT, Firebase, Arduino IoT Cloud, MyQttHub, Google cloud, Amazon Web Services, inclusive lenguajes de programación para el desarrollo del Backend.

Finalmente se revisaron las tecnologías y herramientas con las cuales se desarrollan las aplicaciones para que el sistema interactúe con el usuario final, aunque algunas veces se tienen prototipo M2M que no requieren de la interacción con un humano para la toma de decisiones. Aquí se identificaron lenguajes de programación, framework para el desarrollo de frontend y dashboard entre los cuales se tienen: Java, NodeJS, ThingsBoard, NodeRed, Arduino IoT Cloud.

Seguidamente se realizó la evaluación de tecnologías y herramientas con el propósito de recomendar y seleccionar a través de la construcción del prototipo de prueba utilizado para validar su viabilidad, para lo cual se consideraron criterios como usabilidad, el costo, curva de aprendizaje, accesibilidad y la capacidad para resolver problemas cotidianos. En la tabla 1 se muestra sintetizado la recolección de los requerimientos funcionales obtenido de la aplicación de instrumentos de recolección como as historias de usuario aplicadas a los coordinadores y rector de institución educativa seleccionada para la prueba piloto del prototipo. La tabla 1 muestra los requisitos funcionales definidos para el prototipo.

Tabla 1: Requerimientos funcionales del prototipo

Cod	Descripción del requerimiento
RQ1	El dispositivo debe controlar el toque de la sirena en cada cambio de clase
RQ2	El dispositivo debe tener un botón para seleccionar horarios pre configurados
RQ2	El dispositivo funciona con un horario por defecto
RQ2	El dispositivo debe tener un pulsador para actualizar la fecha y hora desde internet
RQ2	El dispositivo debe detectar los días de fines de semana, sábados y domingos y no activar la sirena
RQ2	Modificar el horario en forma standalone por Wifi
RQ2	El dispositivo debe tener una pantalla que muestre (Tableta): Fecha y hora, Intervalo de clase actual, configuración seleccionada

RQ2	El dispositivo debe tener funcionamiento silencioso
RQ2	El dispositivo debe activación manual
RQ2	Aplicación de Android, debe mostrar: horario activo del día, el intervalo de hora, programar horarios (Rol coordinador), Recibir notificaciones, enviar notificaciones (rol coordinador), permitir activar la sirena (rol coordinador), modo silencioso (rol coordinador)

4.2 Diseño del prototipo Hardware y Software según los requerimientos definidos y las tecnologías IoT seleccionadas en la fase de análisis

para iniciar el proceso de diseño antes de construir cualquier producto (en este caso un dispositivo inteligente), como buena práctica siempre diversos expertos como Gámez López y Sánchez (2018), recomiendan tener un marco de referencia en este contexto definir la arquitectura a utilizar. Aunque Sethi y Sarangi (2017), resaltan que no existe un consenso en el modelo de arquitectura utilizar, sin embargo, se realizó una revisión y selección basados en el número de capas 3, 4, 5, 6 y hasta 7 entre ellas la propuesta de El Hakim (2018). Además, se tuvo en cuenta el ámbito de su aplicación; sector salud, transporte, agricultura, educación entre otros. Finalmente, se optó por utilizar y recomendar el modelo de 4 capas propuesto por Boujrad, Lazaar, y Hassine (2020), el cual emplea la tecnología Wireless Body Area Network (WBAN) y presenta similitudes con la arquitectura de Ortiz (2019).

Uno de los motivos por los cuales se seleccionó la arquitectura de 4 capas es que presenta mucha relación y es coherente con la estructura de división de los componentes en los que podemos dividir un sistema IoT. Sin embargo, para el desarrollo de prototipos IoT en otros ámbitos, se propone hacer una revisión de los resultados presentados por Buitrón Ruiz (2022).

Para sintetizar los resultados de la evaluación y selección de las herramientas a utilizar se muestra en la figura 7 el esquema final del prototipo IoT para automatizar el funcionamiento de una sirena escolar utilizada para validar los resultados.

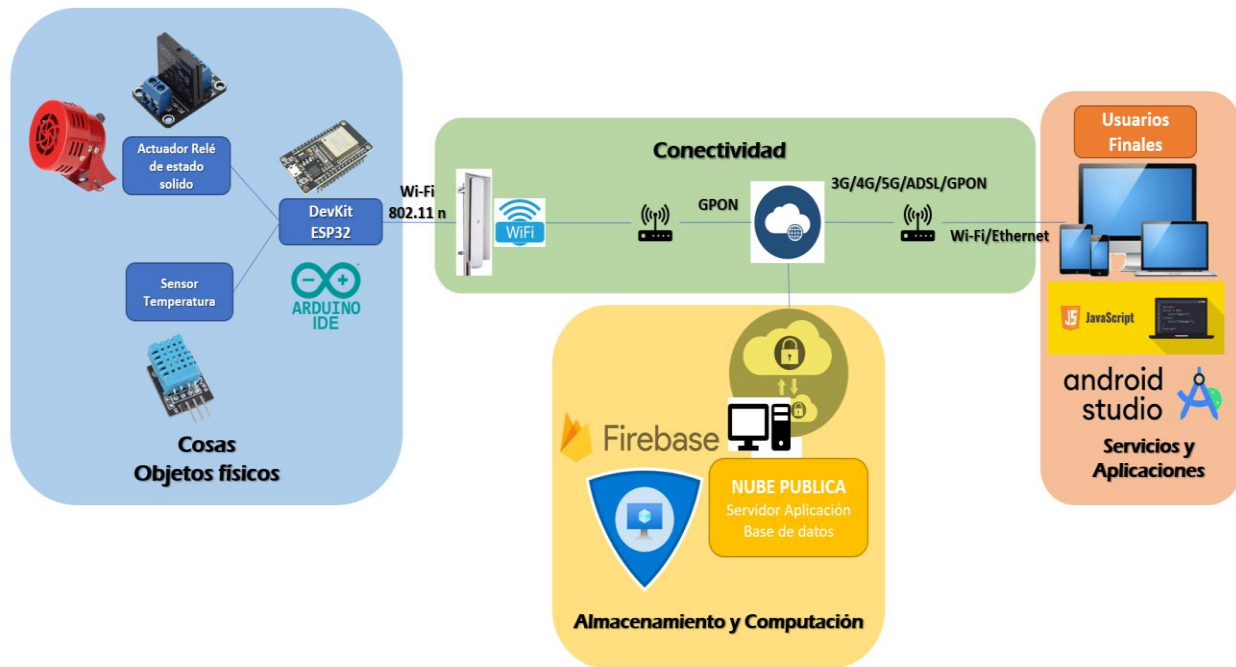


Figura 7. componentes del prototipo IoT de Sirena Escolar

Teniendo como insumo los requerimientos definidos y mostrados en la tabla 1, la figura 8 muestra el diseño del panel del prototipo IoT, para el cual se seleccionó utilizar la placa DevKit ESP32, y para programarla utilizar el IDE de Arduino, la selección de esta placa fue orientada por los criterios de costos, por tener incorporada conectividad Wi-Fi y Bluetooth, además cuenta con un RTC que facilita el desarrollo de objetos que requieren precisión en la hora.

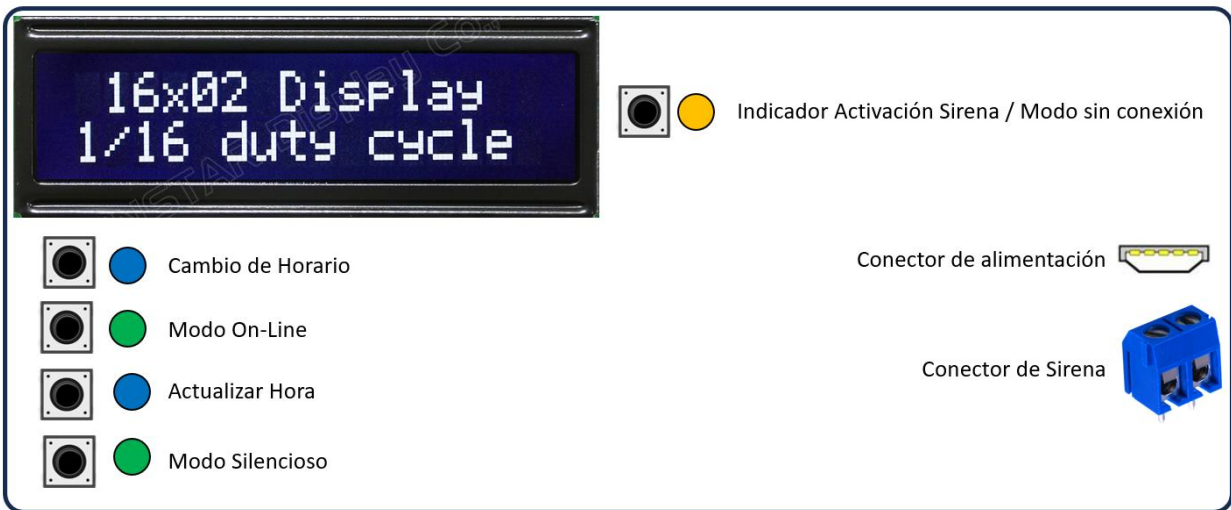


Figura 8. panel del prototipo IoT de Sirena Escolar

El prototipo se conecta a internet utilizando la WLAN de tecnología Wi-Fi de la institución Educativa, que a su vez tiene un canal de acceso a internet con tratado con un ISP que utiliza tecnologías de acceso a internet GPON.

El propósito de tener conectado el prototipo hardware a internet con una base de datos **Firestore** es registrar el tiempo escolar trabajado a diario según el horario programado, además de notificar a los usuarios finales mediante una APP de cambios de horarios y los avisos de cambios de sesiones de clase, salida, entre otras eventualidades, de la misma manera se pueden almacenar configuraciones en la base de datos para que pueden ser cargadas al prototipo hardware. Los usuarios pueden conectarse al aplicativo desde cualquier tipo de acceso a internet.

4.3 Construir y codificar el prototipo IoT, incluyendo la aplicación móvil de control local y la aplicación multiplataforma de gestión de horarios

El Objetivo Específico 3 se alcanzó mediante la construcción exitosa de un prototipo de IoT utilizando las tecnologías y herramientas identificadas en los objetivos anteriores. El prototipo

desarrollado demostró la viabilidad y utilidad del IoT en la vida diaria al abordar desafíos específicos en entornos cotidianos y laborales.

Una vez analizados los requerimientos y elaborado los diseños conceptuales se procede a la construcción del circuito electrónico del objeto físico el cual se puede detallar en el diagrama de la figura 9, se puede evidenciar el uso de las tecnologías seleccionadas en los objetivos anteriores, la figura 10 se presenta el prototipo real.

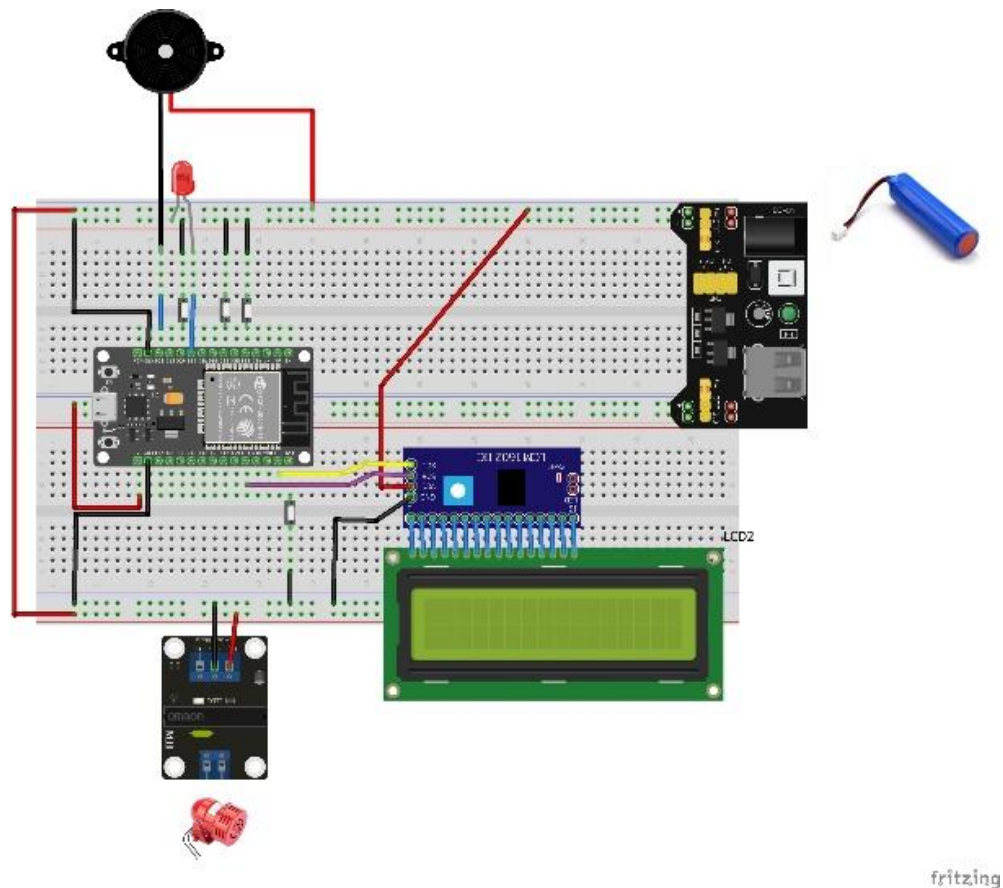


Figura 9. Diagrama de conexiones del circuito electrónico.

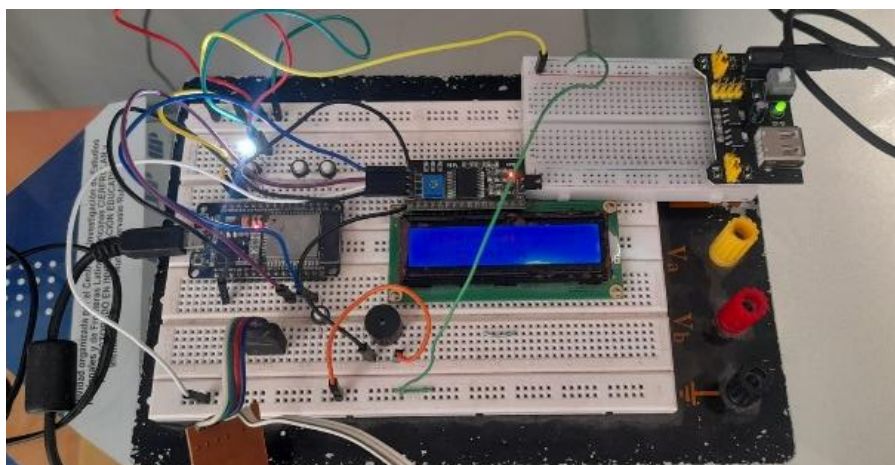


Figura 10. Prototipo real de prueba.

De igual manera en figura 11 se muestra la aplicación de usuario final, con la que se interactúa con el dispositivo físico. Al mismo tiempo las figuras 12 y 13 muestran una porción del código desarrollado y la base de datos utilizadas para almacenar los datos respectivamente.

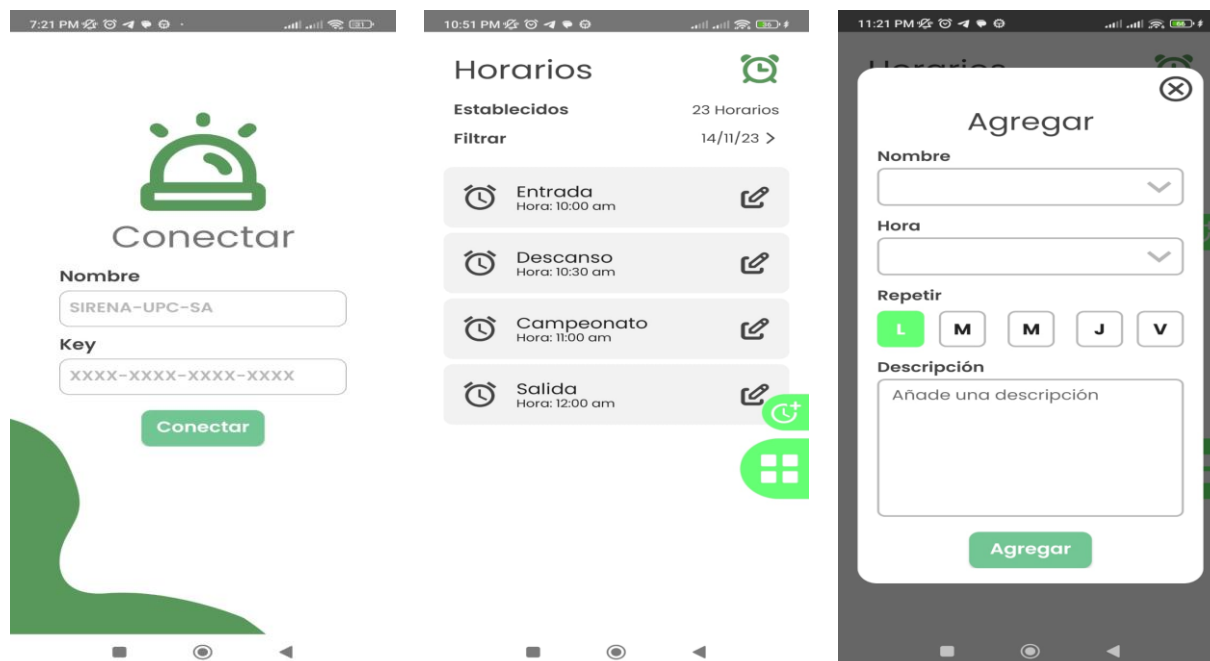


Figura 11. APP para Smartphone con OS Android.



```
SIRENA_IOT_V5.1.1 | Arduino IDE 2.3.2
File Edit Sketch Tools Help

SIRENA_IOT_V5.1.1.ino
1 #include <ArduinoJson.h>
2 #include <ArduinoJson.hpp>
3 #include <ESP32Time.h>
4 #include <Arduino.h>
5 #include <WiFi.h>
6 #include <FirebaseESP32.h>
7 #include <addons/TokenHelper.h>
8 #include <addons/RTDBHelper.h>
9 #include <Wire.h>
10 #include <LiquidCrystal_I2C.h>
11
12 // Dirección I2C de la pantalla LCD1602A (puede variar)
13 #define LCD_ADDR 0x27
14
15 // Define las dimensiones de la pantalla LCD1602A
16 #define LCD_COLS 16
17 #define LCD_ROWS 2
18
19 // Crea un objeto de la clase LiquidCrystal_I2C
20 LiquidCrystal_I2C lcd(LCD_ADDR, LCD_COLS, LCD_ROWS);
21
22 #define WIFI_SSID "SIRENA-ITI"
23 #define WIFI_PASSWORD "SIRENAIOT2024"
24
25 #define API_KEY "AIzaSyAQCW-bNwy5RmZ0lB5W8EHwWiOwrlA_OEE"
26 #define DATABASE_URL "esp32-ab14f-default-rtdb.firebaseio.com"
27 #define USER_EMAIL "control_sirena@gmail.com"
28 #define USER_PASSWORD "12345678abc"
29
30 FirebaseData fbdo;
31 FirebaseAuth auth;
32 FirebaseConfig config;
33
```

Figura 12. Código de la placa DevKit-1.

Los resultados obtenidos resaltan la importancia de la selección cuidadosa de tecnologías y herramientas para el desarrollo de prototipos de IoT, así como su impacto en la solución de tareas cotidianas. Estos hallazgos respaldan la idea de que el IoT tiene el potencial de mejorar significativamente la calidad de vida y la eficiencia en entornos cotidianos.

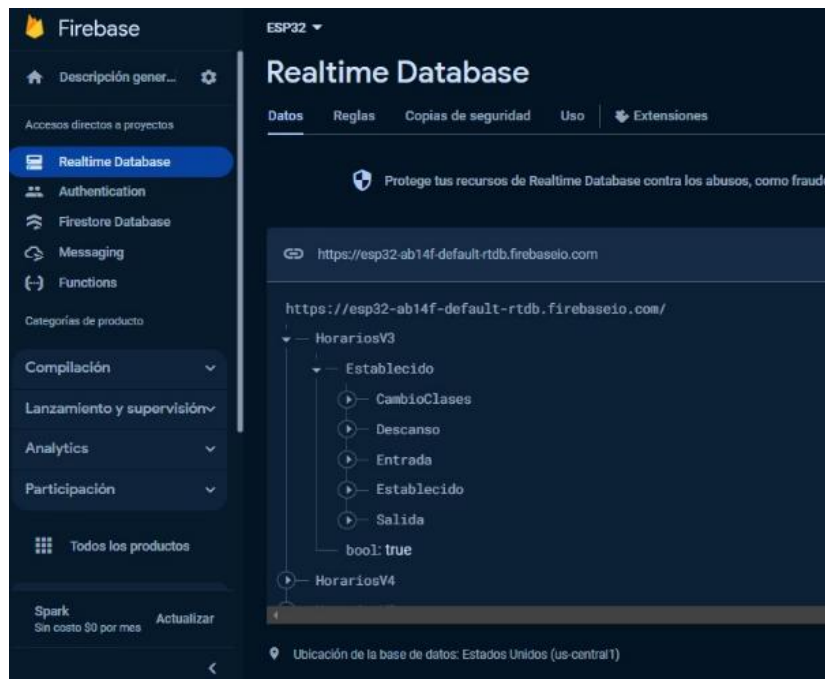


Figura 13. Base de datos en Firebase.

4.4 Prueba piloto

4.4.1 Evidencias de la prueba de concepto o piloto.

Para la realización de la prueba de concepto o piloto se selecciono a la sede principal de la Institución Educativa Tecnico Industrial Laureano Gomez Castro de Aguachica Cesar, en la cual se instaló el prototipo de sirena IoT y que fue asignada al coordinador academico para su gestión. La figura 14 muestra el prototipo final funcionando en su contexto.

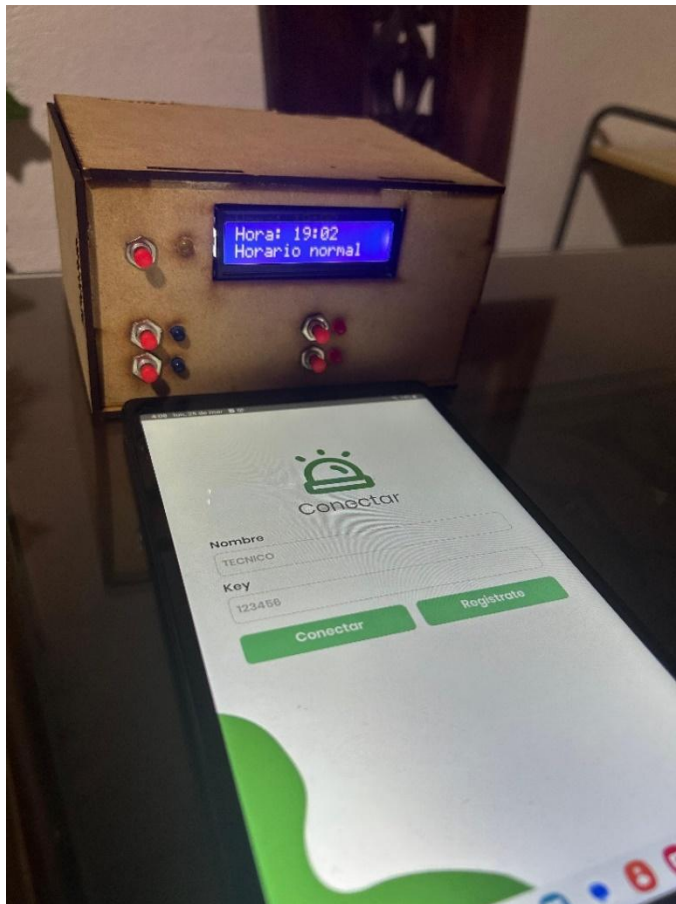


Figura 14. Prototipo final instalado y funcionando.

Se debe resaltar que al prototipo final fue necesario realizar ajuste al circuito, los cuales pueden verse reflejado en la figura 15, dichos ajustes estan relacionados con la optimización del suministro de energia, debido a que el RTC interno de la ESP32 cada vez que se interrumpia el suministro de energia, este regresaba a su valores predeterminado de

fabrila, lo que ocasionaba la necesita de realizar la sincronización de la fecha y hora de la RTC. Con la intención de minimizar la intervención en la operación del prototipo IoT se le adiciono un modulo para cargar una bateria de Li-Ion como respaldo cuando la red de suministro electrico falla, con una sola bateria de 1800 mAh se tiene una autonomia de 4 horas aproximadamente.

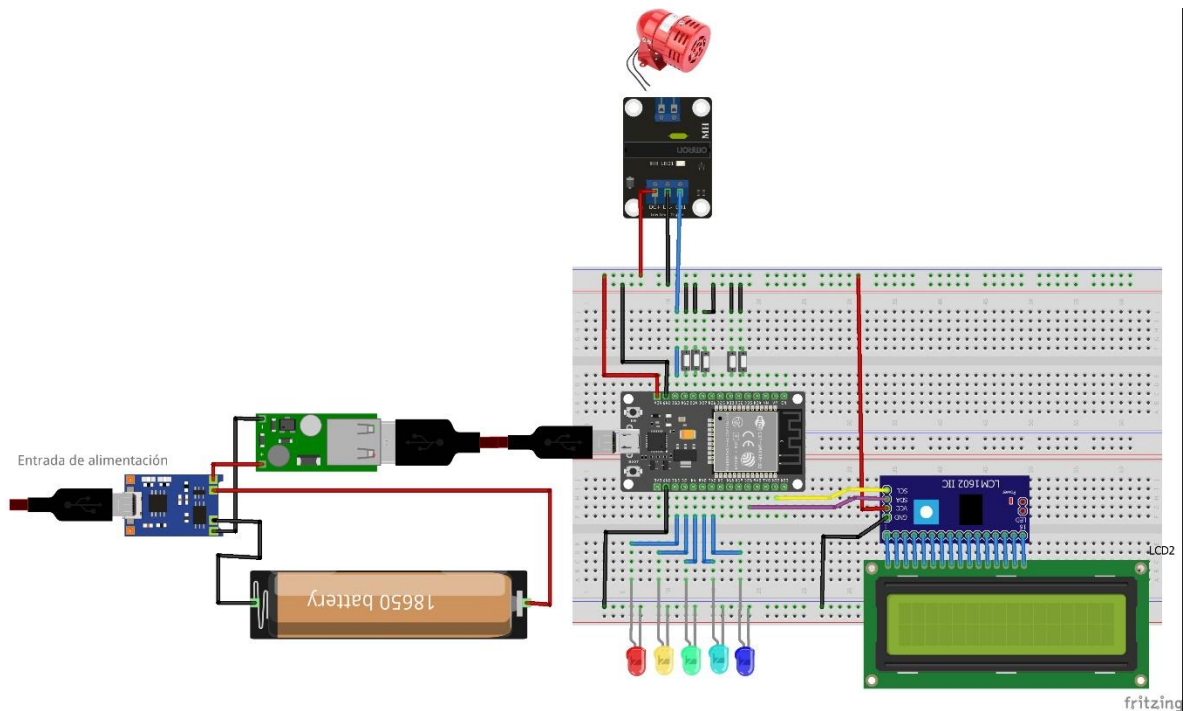


Figura 15. Diagrama del circuito final del Prototipo.

De la misma manera en el anexo A se presenta el código final cargado en la ESP32 Dev-Kit, en el anexo B se presenta la versión final de APP, de la cual se muestran las vistas en la figura 16. Finalmente en el anexo C se puede observar la estructura completa de la base de datos utilizadas para almacenar los horarios la cual fue construida en Firebase.

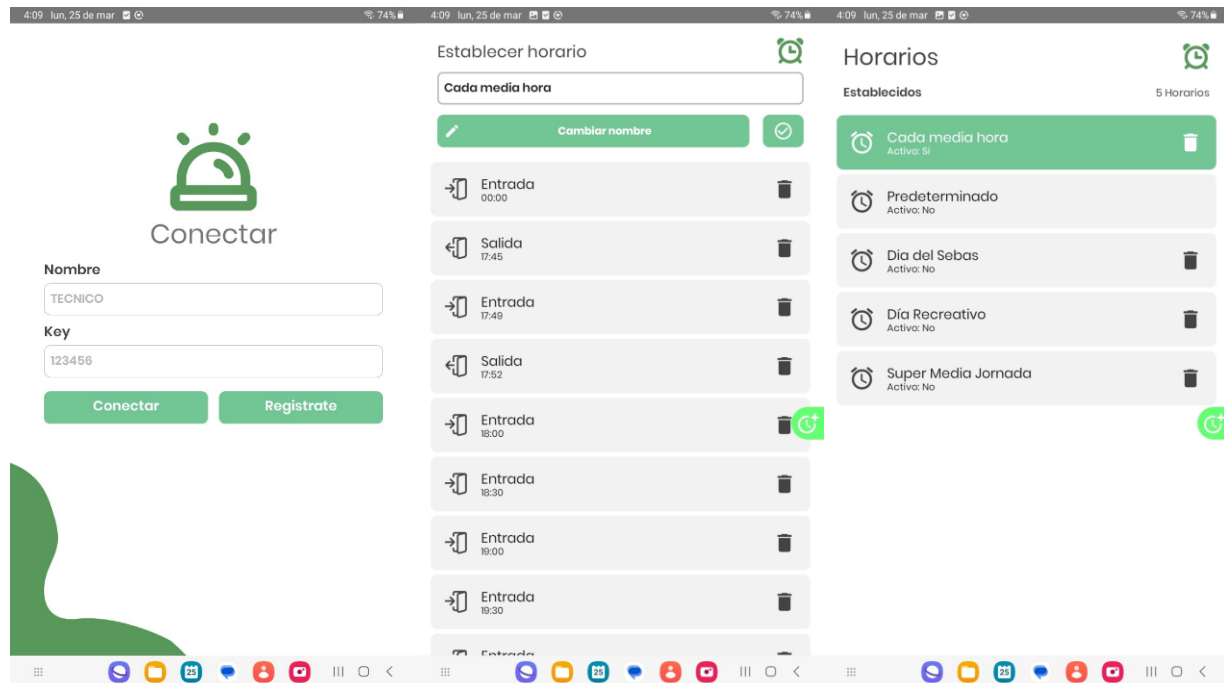


Figura 16. Versión final del APP.

4.4.2 Evidencias de las pruebas de aceptación

Las pruebas de aceptación realizadas por los usuarios finales del prototipo, se convirtieron en una actividad de crucial importancia en la mejora y optimización de las funcionalidades, y garantizar el cumplimiento de los requerimientos establecidos en terminos de usabilidad. Las prueba de funcionamiento por parte del usuario final en este caso, estuvieron a cargo del coordinador de la Institución educativa y el docente de Informatican.

Las pruebas realizadas por parte de los usuarios les permitieron evaluar y hacer comentarios objetivos y pertinentes que contribuyeron a mejorar la programación de la ESP32, la APP con la que interactua de manera remota el usuario y el circuito electronico, estos quedaron plasmados en el diagrama de la figura 15, y los anexos A, B y C. La tabla 2, muestra el instrumento utilizado para orientar las pruebas y la evaluación del prototipo.

Tabla 2: Instrumento para pruebas de aceptación

Pruebas de aceptación operativas Prototipo IoT de Sirena Escolar El objetivo de esta prueba es garantizar que el producto funcione de manera efectiva y cumpla con los requisitos del usuario.					
Descripción del Criterio	1	2	3	4	5
El prototipo ha permitido desatender la labor de estar pendiente de activar la sirena para los cambios de clase. (1=desacuerdo 5=totalmente de acuerdo)					
El prototipo es preciso para activar las indicaciones de cambio de clases. (1=desacuerdo 5=totalmente de acuerdo)					
El prototipo hace que el control de cambio de clase y otras relacionadas a la activación de la sirena escolar, se ha convertido en una tarea desatendida. (1=desacuerdo 5=totalmente de acuerdo)					
Como considera el nivel de dificultad para realizar la modificación de los horarios de cambio de clases. (1=difícil 5=muy fácil)					
Consideras que el prototipo ayuda a optimizar el cumplimiento de la jornada escolar y mejora la disciplina. (1=desacuerdo 5=totalmente de acuerdo)					
Como consideras el nivel de dificultad del uso y la configuración del prototipo de manera remota mediante la app. (1=desacuerdo 5=totalmente de acuerdo)					
El prototipo ha presentado fallos en el funcionamiento que hayan entorpecido el desarrollo de la jornada escolar. (1=ninguna 5=falla permanentemente)					
Como evalúas la confiabilidad del prototipo para indicar los cambios de clase. (1=poco confiable 5=totalmente confiable)					
¿Alguna sugerencia, recomendación o comentario para la mejora del prototipo IoT de sirena escolar? 					
Nombre y Apellidos:					
Cargo:					

CONCLUSIONES

De manera concluyente el proyecto permite afirmar que la aplicación de tecnologías y herramientas del Internet de las Cosas (IoT) en el desarrollo de prototipos para la solución de tareas cotidianas se puede usar como una estrategia acertada para mejorar la eficiencia y la comodidad en entornos domésticos e inclusive laborales.

La construcción exitosa del prototipo de IoT y su posterior implementación en entornos cotidianos han revelado el potencial transformador del IoT para mejorar la calidad de vida y la eficiencia en la realización de tareas habituales. La automatización de procesos, la iteración M2M, Maquinas con personas, el ambiente y seres vivos, la capacidad de monitoreo y toma de decisiones o ejecución de acciones de manera autónoma en tiempo real, han demostrado ser herramientas valiosas para optimizar el funcionamiento y la gestión de recursos en entornos cotidianos. Además, de generar efectos positivos al medio ambiente, optimizar recursos energéticos y económicos

Otro hallazgo importante de resaltar dentro de las tecnologías de los componentes de conectividad, es la relevancia y el posicionamiento de tecnologías como Bluetooth Smart Ready por su bajo consumo energía (BLE) y Wi-Fi la cuales son cruciales para el desarrollo de prototipos de IoT en entornos cotidianos. Estas tecnologías permiten una conectividad eficiente y confiable entre dispositivos, facilitando y agilizando la construcción de artefactos inteligentes. Teniendo en cuenta lo anterior, como futuros estudios de investigación, es recomendable profundizar en el desarrollo Prototipos IoT con criterios Eficiencia Energética.

Igualmente, la evaluación de herramientas y plataformas de código abierto como Arduino permitió establecer su importancia en el desarrollo de soluciones de IoT accesibles y adaptables. Estas plataformas ofrecen flexibilidad y funcionalidad, lo que ayuda a los Maker a implementar prototipos de IoT personalizados y efectivos para abordar una variedad de tareas cotidianas.

Anexos

Anexo A

Código del programa de la ESP32 Dev-Kit

```
#include <ArduinoJson.h>
#include <ArduinoJson.hpp>
#include <ESP32Time.h>
#include <Arduino.h>
#include <WiFi.h>
#include <FirebaseESP32.h>
#include <addons/TokenHelper.h>
#include <addons/RTDBHelper.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Dirección I2C de la pantalla LCD1602A (puede variar)
#define LCD_ADDR 0x27

// Define las dimensiones de la pantalla LCD1602A
#define LCD_COLS 16
#define LCD_ROWS 2

// Crea un objeto de la clase LiquidCrystal_I2C
LiquidCrystal_I2C lcd(LCD_ADDR, LCD_COLS, LCD_ROWS);

// #define WIFI_SSID "SIRENA-ITI"
// #define WIFI_PASSWORD "SIRENAIOT2024"

#define WIFI_SSID "SinInternet"
#define WIFI_PASSWORD "SININTERNET"

#define API_KEY "AIzaSyAQCW-bNwy5RmZO1B5W8EHwWiOwr1A_OEE"
```

```
#define DATABASE_URL "esp32-ab14f-default-rtdb.firebaseio.com"
#define USER_EMAIL "control_sirena@gmail.com"
#define USER_PASSWORD "12345678abc"
```

```
FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;
```

```
unsigned long sendDataPrevMillis = 0;
unsigned long count = 0;
```

```
//Configuración del RTC-----INICIO
ESP32Time rtc;
const char *ntpSever = "pool.ntp.org";
const long gmtOffset_sec = -5 * 3600;
const int daylightOffset_sec = 0;
//Configuración del RTC-----FIN
```

```
DynamicJsonDocument doc(1024);
```

```
String macAddress; //Dirección MAC de la ESP
```

```
String Predeterminado = "{\"cambioHora\":
[\"06:55\", \"08:05\", \"09:00\", \"10:10\", \"11:05\", \"13:25\", \"14:35\",
\"15:30\", \"16:40\", \"17:35\"], \"descanso\": [\"07:50\", \"09:55\",
\"14:20\", \"16:25\"], \"entrada\": [\"06:00\", \"12:30\"], \"salida\":
[\"12:00\", \"18:30\"]}";
String Predeterminado2 = "{\"cambioHora\":
[\"06:55\", \"08:05\", \"09:00\", \"10:10\", \"11:05\", \"13:25\", \"14:35\",
\"15:30\", \"16:40\", \"17:35\"], \"descanso\": [\"07:50\", \"09:55\",
```

```
\\"14:20\\", \\"16:25\\"],\\"entrada\\": [\\"06:00\\", \\"12:30\\"], \\"salida\\":  
[\\"12:00\\", \\"18:30\\"]}";
```

```
String Predeterminado3 = "{\\"cambioHora\\":  
[\\"06:55\\",\\"08:05\\",\\"09:00\\",\\"10:10\\",\\"11:05\\",\\"13:25\\",\\"14:35\\"  
,\\"15:30\\",\\"16:40\\",\\"17:35\\"], \\"descanso\\": [\\"07:50\\", \\"09:55\\",  
\\"14:20\\", \\"16:25\\"],\\"entrada\\": [\\"06:00\\", \\"12:30\\"], \\"salida\\":  
[\\"12:00\\", \\"18:30\\"]}";
```

```
String ultimoOnline = Predeterminado;
```

```
int currentScheduleIndex = 0; // Índice del horario actual
```

```
bool modeOnline = false;
```

```
void setup() {  
    Serial.begin(115200);  
  
    pinMode(13, OUTPUT);  
    pinMode(15, OUTPUT);  
    pinMode(2, OUTPUT);  
    pinMode(4, OUTPUT);  
    pinMode(16, OUTPUT);  
    pinMode(17, OUTPUT);  
  
    pinMode(14, INPUT_PULLUP); //Mode online  
    pinMode(12, INPUT_PULLUP);  
    pinMode(27, INPUT_PULLUP);  
    pinMode(33, INPUT_PULLUP);  
    pinMode(32, INPUT_PULLUP); //Activar alarma
```

```
lcd.init();
lcd.backlight();
lcd.setCursor(0, 0);
lcd.print("Bienvenido!!!");
lcd.setCursor(0, 1);
lcd.print("Sirena IoT :");

leds();

macAddress = WiFi.macAddress(); //Guardar la MAC de la ESP en la
variable
Serial.println("Esta es la MAC: " + macAddress);

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("MAC:");
lcd.setCursor(0, 1);
lcd.print(macAddress);

delay(5000);

configTime();

connectToWiFiWithTimeout(); // Intenta conectarse al WiFi con un
límite de 20 segundos
if (WiFi.status() == WL_CONNECTED) {
    connectFirebaseIfNeeded();
}
}
```

```
void leds() {  
  for (int i = 0; i < 10; i++) {  
    digitalWrite(15, HIGH);  
    delay(100);  
    digitalWrite(15, LOW);  
    digitalWrite(2, HIGH);  
    delay(100);  
    digitalWrite(2, LOW);  
    digitalWrite(4, HIGH);  
    delay(100);  
    digitalWrite(4, LOW);  
    digitalWrite(16, HIGH);  
    delay(100);  
    digitalWrite(16, LOW);  
    digitalWrite(17, HIGH);  
    delay(100);  
    digitalWrite(17, LOW);  
    delay(100);  
  }  
}
```

// Función para intentar conectarse al WiFi por un máximo de 20 segundos

```
void connectToWiFiWithTimeout() {  
  unsigned long startTime = millis(); // Tiempo de inicio de la  
  conexión  
  bool connected = false;  
  
  // Intenta conectarse al WiFi durante un máximo de 20 segundos  
  while (millis() - startTime < 10000) {
```

```

connectWifi();
if (WiFi.status() == WL_CONNECTED) {
    connected = true;
    break;
}
delay(1000); // Espera 1 segundo antes de intentar nuevamente
}

if (connected) {
    Serial.println("Conexión WiFi establecida");
    modeOnline = true; // Activa el modo online

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Conexion wifi");
    lcd.setCursor(0, 1);
    lcd.print("Establecida :)");
} else {
    Serial.println("No se pudo conectar al WiFi en 20 segundos");
    modeOnline = false;

    digitalWrite(4, LOW);

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Wifi:");
    lcd.setCursor(0, 1);
    lcd.print("Sin conexion :(");
}
}

```

```
// Función para conectar al wifi
// Método para conectar al WiFi con un límite de tiempo de 20 segundos
void connectWifi() {
    unsigned long startTime = millis(); // Guarda el tiempo de inicio
    de la conexión

    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

    Serial.print("Connecting to Wi-Fi");

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Wifi:");
    lcd.setCursor(0, 1);
    lcd.print("Conectando");

    while (WiFi.status() != WL_CONNECTED && millis() - startTime <
10000) {
        lcd.print(".");

        Serial.print(".");
        delay(300);
    }

    if (WiFi.status() == WL_CONNECTED) {
        Serial.println();
        Serial.print("Connected with IP: ");
        Serial.println(WiFi.localIP());
        Serial.println();
    }
}
```

```

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Conexion wifi");
    lcd.setCursor(0, 1);
    lcd.print("Establecida :)");

} else {
    Serial.println();
    Serial.println("Failed to connect to Wi-Fi within 20 seconds.");

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Wifi:");
    lcd.setCursor(0, 1);
    lcd.print("Sin conexion :(");
}
}

void disconnectWifi() {
    // Verifica si está conectado antes de desconectar
    if (WiFi.status() == WL_CONNECTED) {
        // Desconecta WiFi después de 10 segundos (por ejemplo)
        delay(10000);
        WiFi.disconnect();
        Serial.println("WiFi desconectado");
    } else {
        Serial.println("No hay conexión WiFi");
    }
}

```

```
}
```

```
// Función para conectar a la base de datos de Firebase
```

```
void connectFirebaseIfNeeded() {
```

```
    if (!Firebase.ready()) {
```

```
        connectFirebase();
```

```
    }
```

```
}
```

```
// Función para conectar a Firebase
```

```
void connectFirebase() {
```

```
    Serial.printf("Firebase Client v%s\n\n", FIREBASE_CLIENT_VERSION);
```

```
    config.api_key = API_KEY;
```

```
    auth.user.email = USER_EMAIL;
```

```
    auth.user.password = USER_PASSWORD;
```

```
    config.database_url = DATABASE_URL;
```

```
    config.token_status_callback = tokenStatusCallback;
```

```
    Firebase.reconnectNetwork(true);
```

```
    fbdo.setBSSLBufferSize(4096, 1024);
```

```
    Firebase.begin(&config, &auth);
```

```
    Firebase.setDoubleDigits(5);
```

```
    digitalWrite(4, HIGH);
```

```
}
```

```
// Función para actualizar la hora
```

```
void configTime() {
```

```
    connectWifi();
```

```
    configTime(gmtOffset_sec, daylightOffset_sec, ntpSever);
```

```
    struct tm timeinfo;
```

```

    if (getLocalTime(&timeinfo)) {
        rtc.setTimeStruct(timeinfo);
    }
}

```

// Función para obtener el nombre del horario que está funcionando y armar la ruta

```

String scheduleRoute;
String scheduleCurrent;

```

```

void getScheduleCurrent() {

```

```

    currentScheduleIndex = 3;

```

```

    String path = "/devices/" + macAddress + "/scheduleCurrent";
    if (Firebase.getString(fbdo, path)) {
        scheduleCurrent = fbdo.stringData();
        scheduleRoute = "/devices/" + macAddress + "/schedules/" +
scheduleCurrent;
        getJsonSchedules();
    } else {
        modeOnline = false;
        Serial.println("Error al obtener el valor de Firebase: " +
fbdo.errorReason());
    }
}

```

// Función para obtener el json de una ruta especifica, deserializarlo y almacenarlo en un array

```
String jsonString; //Aquí se almacena el horario que esta funcionando
en el momento
```

```
std::vector<String> horasArray;
```

```
void getJsonSchedules() {
    if (Firebase.ready()) {
        if (Firebase.getJSON(fbdo, scheduleRoute)) {
            FirebaseJson &json = fbdo.jsonObject();
            json.toString(jsonString);
            horasArray.clear();
```

```
            ultimoOnline = jsonString; //Guardamos ultimo horario que se
bajó de la base de datos y lo guardamos para poder usarlo en el modo
offline
```

```
        } else {
            Serial.println("Error al obtener el JSON de Firebase");
            Serial.println(fbdo.errorReason());
        }
    }
    deserialize();
}
```

```
// Función para deserializar el JSON y extraer cada hora para poder
comparar
```

```
void deserialize() {
    deserializeJson(doc, jsonString);
    const char *categorias[] = { "cambioHora", "descanso", "entrada",
"salida" };
    for (const auto &categoria : categorias) {
        JsonArray array = doc[categoria];
```

```

    for (const auto &value : array) {
        String hora = value.as<String>();
        if (rtc.getTime("%H:%M") == hora) {
            ringAlarm(categoria, hora);
        }
        horasArray.push_back(hora);
    }
}
}

```

// Función para activar la alarma según la categoría y hora especificadas

```

unsigned long lastAlarmTime = 0;
bool silentMode = false;

```

```

void ringAlarm(String categoria, String hora) {
    unsigned long currentTime = millis();
    if (rtc.getTime("%H:%M") == hora && currentTime - lastAlarmTime >=
60000) {
        Serial.print("Alarma activada para la categoría ");
        Serial.println(categoria);

        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Alarma de ");
        lcd.print(categoria);

        int repetitions = (categoria == "entrada") ? 1 : (categoria ==
"cambioHora") ? 1

```

```

: (categoria ==
"descanso")    ? 1
: (categoria ==
"salida")      ? 3
: 0;

if (silentMode) {
    for (int i = 0; i < repetitions; ++i) {
        digitalWrite(15, HIGH);
        delay(3000);
        digitalWrite(15, LOW);
        delay(2000);
    }
} else {
    for (int i = 0; i < repetitions; ++i) {
        digitalWrite(15, HIGH);
        digitalWrite(13, HIGH);
        delay(3000);
        digitalWrite(15, LOW);
        digitalWrite(13, LOW);
        delay(2000);
    }
}

lastAlarmTime = currentTime;
Serial.println("Alarma desactivada");
}
}

```

```

// Arreglo de horarios
String schedules[] = { Predeterminado, Predeterminado2,
Predeterminado3, ultimoOnline };

String scheduleNames[] = { "Horario normal", "Dia del jean", "Media
Jornada", "Ultimo Online" };

// Función para cambiar al siguiente horario
void changeSchedule() {
    currentScheduleIndex++; // Incrementar el índice
    if (currentScheduleIndex >= sizeof(schedules) /
sizeof(schedules[0])) {
        // Si el índice supera el tamaño del arreglo, vuelve al principio
        currentScheduleIndex = 0;
    }
    // Actualizar el jsonString con el nuevo horario
    jsonString = schedules[currentScheduleIndex];

    // Imprimir el nombre del horario en la pantalla LCD
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(scheduleNames[currentScheduleIndex]);

    Serial.println(scheduleNames[currentScheduleIndex]);
}

void loop() {
    if (digitalRead(14) == LOW) {

```

```

    if (!modeOnline) {
        digitalWrite(4, HIGH);
    } else {
        digitalWrite(4, LOW);
    }

    modeOnline = !modeOnline;
    Serial.println(modeOnline ? "Modo online activado" : "Modo offline
activado");

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(modeOnline ? "Modo online activado" : "Modo offline
activado");

    delay(1000);
    if (modeOnline) {
        connectToWiFiWithTimeout(); // Intenta conectarse al WiFi con
un límite de 20 segundos
        if (WiFi.status() == WL_CONNECTED) {
            connectFirebaseIfNeeded();
        } // Intenta establecer la conexión a Firebase cuando se activa
el modo online
    }
}

if (digitalRead(12) == LOW) {
    digitalWrite(2, HIGH);
    Serial.println("Entrando al cambio de horario");
}

```

```

    if (!modeOnline) {
        Serial.println("Horario cambiado");
        changeSchedule(); //AQUI VAMOS A PONER EL CODIGO DE CAMBIAR DE
HORARIOS
    } else {
        Serial.println("No se pudo cambiar horario");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("No se puede");
        lcd.setCursor(0, 1);
        lcd.print("cambiar el horario");
        delay(3000);
    }
} else {
    digitalWrite(2, LOW);
}

if (digitalRead(27) == LOW) {
    digitalWrite(17, HIGH);

    silentMode = !silentMode;
    Serial.println(silentMode ? "Modo silencio activado" : "Modo
silencio desactivado");

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Modo silencioso");
    lcd.setCursor(0, 1);
    lcd.print(silentMode ? "activado" : "desactivado");
}

```

```
    delay(5000);
} else {
    if (silentMode) {
        digitalWrite(17, HIGH);
    } else {
        digitalWrite(17, LOW);
    }
}

if (digitalRead(33) == LOW) {
    digitalWrite(16, HIGH);

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Sincronizando");
    lcd.setCursor(0, 1);
    lcd.print("hora reloj");

    delay(5000);

    configTime();

} else {
    digitalWrite(16, LOW);
}

if (modeOnline) {
    getSchedCurrent();
} else {
    digitalWrite(4, LOW);
}
```

```

        deserialize();
    }

    if (digitalRead(32) == LOW) {
        digitalWrite(15, HIGH);
        digitalWrite(13, HIGH);
        delay(3000);
        digitalWrite(15, LOW);
        digitalWrite(13, LOW);
    }

    Serial.println("La hora actual es: " + rtc.getTime("%H:%M"));

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Hora: ");
    lcd.print(rtc.getTime("%H:%M"));
    lcd.setCursor(0, 1);
    lcd.print(modeOnline ? scheduleCurrent :
scheduleNames[currentScheduleIndex]);

    delay(3000);
}

// Historial de actualizacion
// - El dispositivo se conecta y desconecta del wifi correctamente.
// - El dispositivo cambia de modo online y offline si se pierde la
conexion.
// - El dispositivo cambia de horarios cuando se encuentra en el modo
offline.

```

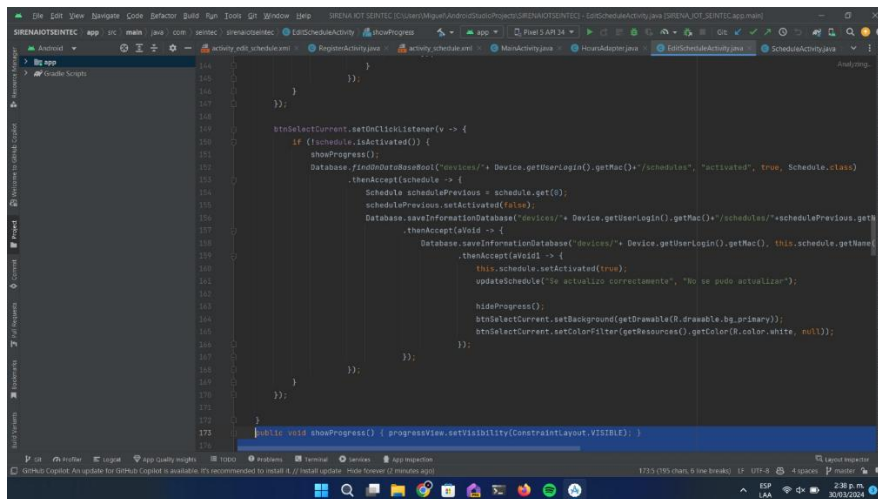
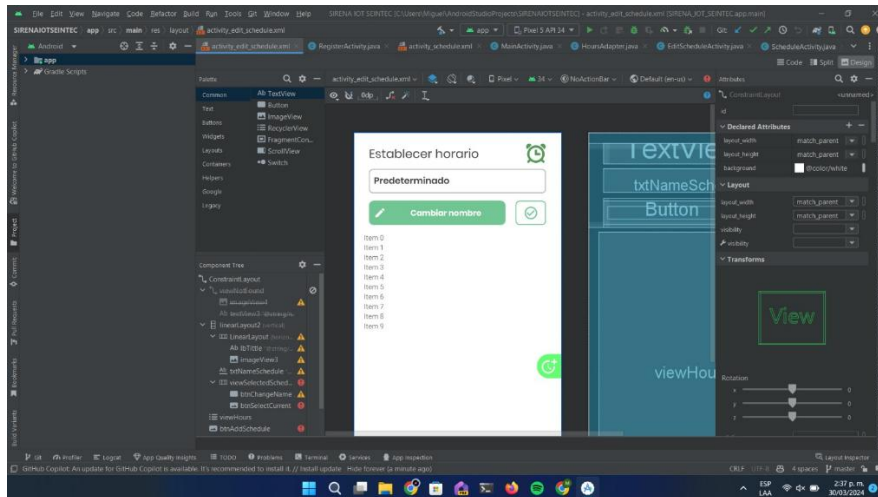
// - El dispositivo rescata el ultimo horario online y lo guarda para poder usarlo en el modo offline.

// - Los mensajes aparecen correctamente en la pantalla lcd.

// - El modo silencioso funciona correctamente.

Anexo B

Android Studio

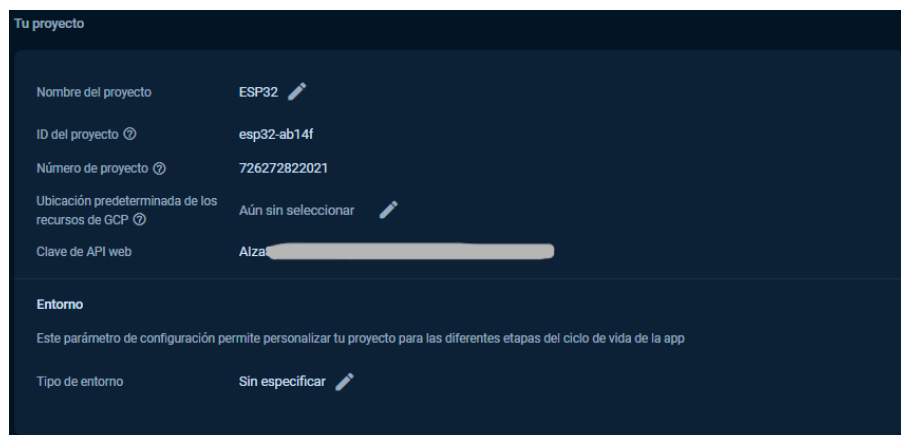


Anexo C

Estructura del Json



Configuración del Firebase:



Anexo D

Manual de usuario

¡Bienvenido al manual de usuario para el prototipo de sirena automatizada! Este manual te guiará a través de los pasos necesarios para utilizar el dispositivo de manera efectiva en tu colegio. Asegúrate de seguir cada paso cuidadosamente para aprovechar al máximo las funciones de la sirena automatizada.

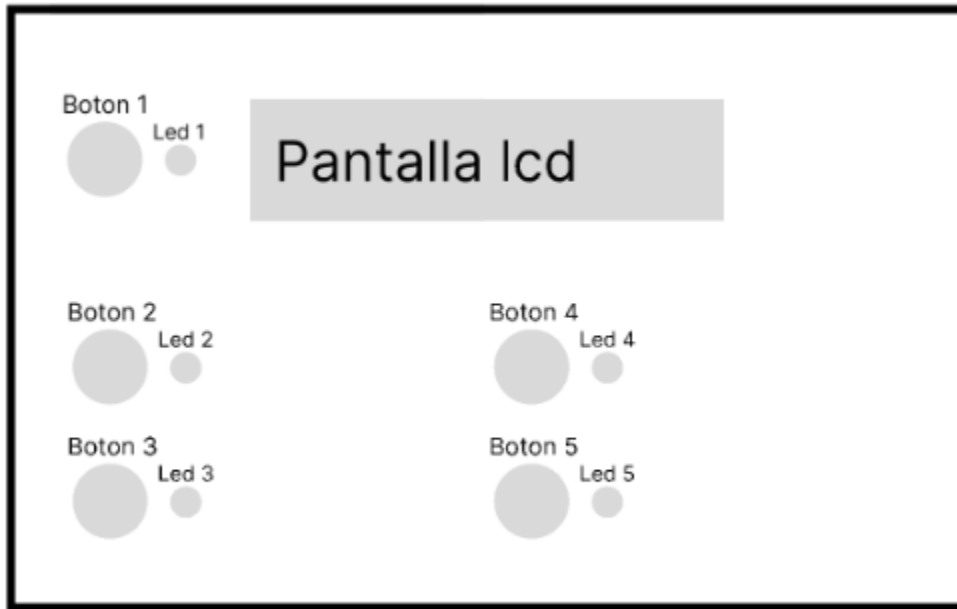


Imagen de referencia (Creación propia).

Encendido del Dispositivo

Conecta el cable micro USB proporcionado al dispositivo y enchúfalo a una toma de corriente.

Utiliza un cargador de carga normal o convencional para asegurar un suministro de energía estable, de esta forma el dispositivo cargará la batería interna.

El encendido tarda unos segundos mientras el dispositivo se inicializa.

Identificación de la MAC

Después de la inicialización, la dirección MAC del dispositivo aparecerá en la pantalla durante 5 segundos.

Toma nota de esta dirección MAC, ya que la necesitarás para registrar el dispositivo en la aplicación.

Registro en la Aplicación Android

Instala la aplicación proporcionada en tu dispositivo Android.

Abre la aplicación y selecciona la opción de registro.

Ingresa la dirección MAC del dispositivo y asigna una contraseña para tu cuenta.

Una vez registrado, tendrás acceso a todas las funciones de control del dispositivo desde la aplicación.

Configuración del Horario

El dispositivo cuenta con 3 horarios preestablecidos los cuales funcionan sin necesidad de conectarse a internet, los cuales son los siguientes:

Horario A				Horario B				Horario C			
Cambio Hora	Descanso	Entrada	Salida	Cambio Hora	Descanso	Entrada	Salida	Cambio Hora	Descanso	Entrada	Salida
6:55	7:50	6:00	12:00	6:35	7:45	6:00	9:55	8:35	9:45	8:00	12:00
8:10	10:00	12:30	18:30	7:10	13:40	12:30	16:30	9:10	13:40	12:30	16:30
9:05	14:20			8:10	15:05			10:10	15:05		
10:10	16:30			8:45				10:45			
11:05				9:20				11:20			
13:25				13:05				13:05			
14:40				13:55				13:55			
15:35				14:30				14:30			
16:40				15:20				15:20			
17:35				15:55				15:55			

Además de estos horarios preestablecidos, hay uno llamado "Horario Online", en el cual se almacena el horario configurado desde la aplicación.

Activación de la Sirena

Una vez que el dispositivo esté configurado y los horarios estén programados, la alarma se activará automáticamente, pero también puedes activar la sirena manualmente desde la aplicación y con el **botón 1**, revise la imagen de referencia del dispositivo.

Uso de los Botones

El dispositivo cuenta con 5 botones descritos y numerados de la siguiente manera:

BOTÓN 1 (Activación Manual): Mantén presionado durante unos segundos hasta que el indicador LED1 se encienda. Esto activará la alarma manualmente.

BOTÓN 2 (Cambio de horario manual): Mantén presionado durante unos segundos hasta que se active el LED2. Esto cambiará entre los horarios preestablecidos del dispositivo. Recuerda que hay tres horarios preestablecidos y uno adicional llamado "Horario Online".

BOTÓN 3 (Modo online/offline): Mantén presionado hasta que encienda el LED3. Esto cambiará el modo de online a offline y viceversa. Si el dispositivo se conecta a internet y el modo online queda activo, el LED3 permanecerá encendido; de lo contrario, se apagará.

BOTÓN 4 (Sincronizar hora actual): Mantén presionado hasta que encienda el LED4. El dispositivo intentará conectarse a internet si no lo está y actualizará la hora del dispositivo.

BOTÓN 5 (Modo silencioso): Mantén presionado hasta que encienda el LED5. Esto activará o desactivará el modo silencioso. Si el LED5 está encendido, significa que el modo silencioso está activo.

Revise la imagen de referencia.

Mantenimiento y Soporte

Para cualquier consulta adicional o asistencia técnica, no dudes en ponerte en contacto con nuestro equipo de soporte técnico.

Visualización de Funciones en la Pantalla del Dispositivo:

El prototipo de sirena automatizada está equipado con una pantalla que muestra información relevante sobre las funciones que se están ejecutando en tiempo real.

Desde la pantalla del dispositivo, podrás visualizar detalles como el estado de la conexión a internet, el modo de operación (online u offline), el estado de activación de la alarma, el horario actual y cualquier otra información importante relacionada con el funcionamiento del dispositivo.

Utiliza esta pantalla como una herramienta adicional para monitorear y verificar el estado y las configuraciones del dispositivo en cualquier momento.

Conexión de Sirena u Otros Sensores de Salida

Para conectar la sirena u cualquier otro sensor de salida, utilice la entrada del relé ubicada en el costado derecho del dispositivo.

Es importante buscar la ayuda de un técnico calificado para garantizar una conexión segura y adecuada de los dispositivos de salida.

El técnico podrá proporcionar orientación sobre la selección apropiada de cables y conexiones, así como sobre la configuración de los dispositivos de salida para su correcto funcionamiento con el prototipo de sirena automatizada.

BIBLIOGRAFIA.

- Acosta-Díaz, R., Aréchiga, M. A., Ramos-Michel, E. M., & Damián-Reyes, P. (2020). EL INTERNET DE LAS COSAS Y SU IMPACTO EN LA EDUCACIÓN. *EL INTERNET DE LAS COSAS*, 184.
- Alapizco Bobardel, S., & Vázquez Rodríguez, G. (2021). Las Metodologías De Innovación Para El Desarrollo De Nuevos Productos: Un Enfoque Al Consumidor. *Mundo, Arquitectura, Diseño gráfico y Urbanismo*, 4(8), 19-28. doi:<https://doi.org/10.36800/madgu.v3i5.40>
- Amézquita Ayure, S., & others. (2022). *Prototipo de sistema de monitorización y regulación automática de material particulado basado en enfoques de iot para entidades educativas ubicadas en entornos hostiles de alto flujo vehicular. Hacia el aumento de la calidad del aire en las instituciones educativas.*
- Arias, C. (2023). colombiahosting. Obtenido de <https://www.colombiahosting.com.co/blog/que-es-hosting-web/>
- Bill Buxton. (2007). *Sketching User Experiences: Getting the Design Right and the Right Design* - Bill Buxton - Google Libros.
[https://books.google.com.co/books?hl=es&lr=&id=2vfPxocmLh0C&oi=fnd&pg=PP1&dq=Buxton,+B.\(2007\).+%22Sketching+User+Experiences:+Getting+the+Design+Right+and+the+Right+Design.%22+Morgan+Kaufmann.&ots=07Stidi7XK&sig=xE0kzwX_v3dzCkj-JaMj2kzvJA8&redir_esc=y#v=onepage&q&f=false](https://books.google.com.co/books?hl=es&lr=&id=2vfPxocmLh0C&oi=fnd&pg=PP1&dq=Buxton,+B.(2007).+%22Sketching+User+Experiences:+Getting+the+Design+Right+and+the+Right+Design.%22+Morgan+Kaufmann.&ots=07Stidi7XK&sig=xE0kzwX_v3dzCkj-JaMj2kzvJA8&redir_esc=y#v=onepage&q&f=false)
- Bandyopadhyay, D., & Sen, J. (2011). Internet of Things: Applications and Challenges in Technology and Standardization. *Wireless Pers Commun*, 49-69. doi:<https://doi.org/10.1007/s11277-011-0288-5>
- Benito Herranz, Á. (2019). Desarrollo de aplicaciones para IoT con el módulo ESP32. Universidad de Alcalá. Escuela Politécnica Superior.
- Bonilla-Fabela, I., Tavizon-Salazar, A., Morales-Escobar, M., Guajardo-Muñoz, L. T., & Laines-Alamina, C. I. (2016). IoT, el internet de las cosas y la innovación de sus aplicaciones. *Vinculatégica efan*, 2(1), 2313–2340.
- Boujrad, M., Lazaar, S., & Hassine, M. (2020). Performance Assessment of Open Source IDS for improving IoT Architecture Security implemented on WBANs. *Proceedings of the 3rd International Conference on Networking* (págs. 1-4). New York: Information Systems & Security. doi:10.1145/3386723.3387892
- Buitrón Ruiz, D. F. (2022). Arquitecturas y modelos de referencias para sistemas IoT. Estado del arte de las arquitecturas para sistemas IoT. Escuela Politecnica Nacional. Trabajo de Grado. Quito: Escuela Politecnica Nacional.

- Catacora, C. A., Ale-Nieto, T., Valencia, E. L., & Deza, J. (2020). Desarrollo e implementación de un sistema inalámbrico (IoT) de gestión energética para la Universidad Privada de Tacna. *INGENIERÍA INVESTIGA*, 2(01), 186–198.
- Del Giorgio Solfa, F., & Lara Marozzi, S. (2020). Diseño y desarrollo de productos en base a prototipado rápido. XV Semana Internacional de Diseño en Palermo 2020, 15.
- Del Val Román, J. L. (2016). Industria 4.0: la transformación digital de la industria. Valencia: Conferencia de Directores y Decanos de Ingeniería Informática, Informes CODDII.
- El Hakim, A. (2018). Internet of Things (IoT) System Architecture and Technologies, White Paper. IoT World Forum Reference Model. Cisco.
- Evans, D. (2012). The Internet of Everything How More Relevant and Valuable Connections. pag 9. Obtenido de https://www.cisco.com/c/dam/global/en_my/assets/ciscoinnovate/pdfs/loE.pdf
- Gámez López, M., & Sánchez, O. (2018). Desarrollo de un prototipo electrónico e informático aplicando la tecnología del internet de las cosas para (1 ed.). La Libertad, El Salvador: ITCA Editores.
- Garzón, W. L., & López, J. C. (2019). Tecnología internet of things (IoT) y el big data. *Mare Ingenii*, 1(1), 73–79.
- González, D. R. (2013). Arquitectura y Gestión de la IoT. *Telemática*, 12(3), 49–60. <https://revistatelematica.cujae.edu.cu/tele/article/view/119>
- González, L., Osiris, S., Laguía, D., Esteban, G., & Karim, H. (2020). Internet del Futuro – Estudio de tecnologías IoT. Informe Científico Técnico UNPA, 12(3), 105-137.
- Gutiérrez, Á. a. (2018). Almacenamiento en la nube. España: Acta. España.
- Hernandez, R., Fernandez, C., & Batista, P. (2018). Metodología de la investigación (Vol. 6). Mexico: McGraw-Hill.
- Hewlett Packard Enterprise. (s.f.). What is Edge Computing? Recuperado el 31 de enero de 2023, de <https://www.hpe.com/us/en/what-is/edge-computing.html>
- ISO. (2018). *ISO/IEC 30141:2018(en), Internet of Things (IoT) — Reference Architecture*. <https://www.iso.org/obp/ui/#iso:std:iso-iec:30141:ed-1:v1:en>
- Lopez i Seuba, M. (2019). Internet de las Cosas, Latransformación Digital de la sociedad. Bogota: Rama.
- Mahedero Biot, F. (2020). Desarrollo de una aplicación IoT para el envío de imágenes mediante el protocolo MQTT. València: Universitat Politècnica de València.

Manuel, J., Navarro, A., & López Sanz, M. (2010). *TESIS FIN DE MASTER Cloud Computing: fundamentos, diseño y arquitectura aplicados a un caso de estudio*.

Mela, J. L. (2021). *Edge Computing: Aplicaciones y desafíos actuales*. Panamá: Visión Antataura.

Microsoft. (2023). *Descripción de los distintos tipos de modelos en la nube*.

<https://learn.microsoft.com/es-es/training/modules/fundamental-azure-concepts/types-of-cloud-computing>

Navarro Pino, D., Arango Trillos, J., Rincón Pinzón, M., & Ramirez Orellano, L. (2022). Prototipo iot para monitorear variables agroambientales determinantes en la productividad de fincas agropecuarias. *Revista Colombiana de Tecnología de Avanzada (RCTA)*, 1(39), 134-141.
doi:<https://doi.org/10.24054/rcta.v1i39.1415>

Navarro Pino, D., Guerrero Sumalave, D. F., Arango Trillos, J. E., Rincón Pinzón, M. A., & Ramírez Orellano, L. O. (2021). Sistema IOT para el monitoreo de variables físicas agroambientales determinantes en la producción agropecuaria. *Tendencias en la Investigación Universitaria.: Una visión desde Latinoamérica*. Volumen XV, 2021, ISBN 978-980-7857-52-9, pág. 249, 249.
<https://dialnet.unirioja.es/servlet/articulo?codigo=8644560&info=resumen&idioma=SPA>

Ortiz Monet, M. (2019a). *Implementación y Evaluación de Plataformas en la Nube para Servicios de IoT*.
<https://riunet.upv.es/handle/10251/127825>

Ortiz Monet, M. (2019b). *Implementación y Evaluación de Plataformas en la Nube para Servicios de IoT*.
<https://riunet.upv.es/handle/10251/127825>

Peleteiro, M., & Seferian, D. (2006). ¿Dónde están los libros?... El lugar de la biblioteca en las escuelas normales. *XIV Jornadas Argentinas de Historia de la Educación 9 al 11 de agosto de 2006 La Plata, Argentina. Habitar la escuela: producciones, encuentros y conflictos*.

Pinzón Niño, D. L. (2016). *Panorama de aplicación de internet de las cosas (IoT)*.
<https://hdl.handle.net/11634/672>

Ray, P. P. (2018). A survey on Internet of Things architectures. *Journal of King Saud University-Computer and Information Sciences*, Vol 30(n°3), 291-319.

Reyes-Flores, E. (2019). *Tipos de sensores*. México: Con-Ciencia Serrana Boletín Científico de la Escuela Preparatoria Ixthahuaco.

Romo Gonzalez, M., & Villalobo Alonso, M. (2019). Aplicaciones y desarrollo de prototipos con Internet de las Cosas. *Revista de Tecnologías Computacionales*, 7. doi: 10.35429/JOCT.2019.9.3.1.7

Sethi, P., & Sarangi, S. (2017). Internet of Things: Architectures, Protocols, and Applications. *Hindawi Journal of Electrical and Computer Engineering*, 25. doi:<https://doi.org/10.1155/2017/9324035>

Tamayo, M. (2009). El proceso de la investigación científica. Mexico: Editorial Mc Graw Hil.

Ulloa-Vásquez, F., Carrizo, D., & García-Santander, L. (2022). Estudio comparativo de costo de implementación de tecnologías inalámbricas para IoT y M2M para la última milla: Un estudio de caso. Ingeniare. Revista chilena de ingeniería, 30(3), .422-428.
doi:<http://dx.doi.org/10.4067/S0718-33052022000300422>

Vergara Herrera, R. D. (2021). *Asistente educativo inteligente para la utilización de tecnologías IOT aplicadas en las aulas de clase de las instituciones educativas de Colombia*.
<https://repository.unab.edu.co/handle/20.500.12749/15041>

Vergara, S. G. (2008). Tecnología Bluetooth. Mexico: Instituto Politécnico Nacional.

Zumba Gamboa, J. P., & León Arreaga, C. A. (2018). Evolución de las Metodologías y Modelos utilizados en el Desarrollo de . INNOVA Research Journal, 3(10), 20-33.