



UNIVERSIDAD TÉCNICA DE MACHALA
FACULTAD DE INGENIERÍA CIVIL

MAESTRÍA EN SOFTWARE

**DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE GESTIÓN DE
ACCESO A LABORATORIOS REMOTOS EN UN ENTORNO DE INTERNET
DE LAS COSAS**

KEVIN ADRIÁN VALAREZO PAZ

Propuesta Metodológica y Tecnológica Avanzada

TUTOR(A): ING. DIXYS LEONARDO HERNÁNDEZ ROJAS, PHD.
COTUTOR: ING. RAMIRO HERNÁN QUEZADA SARMIENTO, MGS.

Machala
2022

DEDICATORIA

A mi madre, por ser la primera persona en brindarme su apoyo incondicional. A mis hermanos, por ser aquellos a los que puedo recurrir para afrontar mis problemas y a mis sobrinos, por ser quienes me han inspirado a ser mejor persona. A mis amigos que han estado ahí para animarme en los momentos buenos y más aún en los malos.

- Ing. Kevin Adrián Valarezo Paz

AGRADECIMIENTOS

Primero, quiero agradecer a Dios por brindarme salud y sabiduría para afrontar este reto. A todos los docentes que han impartido sus cátedras durante las clases, ya que han sabido transmitir sus conocimientos para ayudar en el crecimiento de conocimiento.

Un agradecimiento especial para el Ing. Dixys Hernández y al Ing. Ramiro Quezada, por saberme orientar y ser mi guía durante el desarrollo de este proceso de titulación.

A la Universidad Técnica de Machala, a la Facultad de Ingeniería Civil y a la Maestría en Software por haberme acogido durante estos meses, siendo una grata experiencia.

- Ing. Kevin Adrián Valarezo Paz

RESPONSABILIDAD DE AUTORÍA

Yo, **KEVIN ADRIÁN VALAREZO PAZ** con C.I. **0704640333**, declaro que el trabajo de “**DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE GESTIÓN DE ACCESO A LABORATORIOS REMOTOS EN UN ENTORNO DE INTERNET DE LAS COSAS**”, en opción al título de Magister en Software, es original y auténtico; cuyo contenido: conceptos, definiciones, datos empíricos, criterios, comentarios y resultados son de mi exclusiva responsabilidad.

KEVIN ADRIÁN VALAREZO PAZ

C.I.: 0704640333

Machala, 2022/octubre/12

REPORTE DE SIMILITUD TURNITIN

Tesis de Kevin2

INFORME DE ORIGINALIDAD

1 %	1 %	1 %	0 %
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	repositorio.utmachala.edu.ec Fuente de Internet	1 %
2	hdl.handle.net Fuente de Internet	1 %

Excluir citas Activo

Excluir bibliografía Activo

Excluir coincidencias < 1%

CERTIFICACIÓN DEL TUTOR

Yo, **DIXYS LEONARDO HERNÁNDEZ ROJAS** con C.I.: **0923026298**; tutor del trabajo de “**DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE GESTIÓN DE ACCESO A LABORATORIOS REMOTOS EN UN ENTORNO DE INTERNET DE LAS COSAS**”, en opción al título de **Magister en Software**, ha sido revisado, enmarcado en los procedimientos científicos, técnicos, metodológicos y administrativos establecidos por el Centro de Posgrado de la Universidad Técnica de Machala (UTMACH), razón por la cual doy fe de los méritos suficientes para que sea presentado a evaluación.

DIXYS LEONARDO HERNÁNDEZ ROJAS

C.I: 0923026298

Machala, 2022/octubre/12

CESIÓN DE DERECHOS DE AUTOR

Yo, **KEVIN ADRIÁN VALAREZO PAZ** con C.I.: **0704640333**, autor del trabajo de titulación “**DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE GESTIÓN DE ACCESO A LABORATORIOS REMOTOS EN UN ENTORNO DE INTERNET DE LAS COSAS**”, en opción al título de **Magister en Software**, declaro bajo juramento que:

- El trabajo aquí descrito es de mi autoría, que no ha sido presentado previamente para ningún grado o calificación profesional. En consecuencia, asumo la responsabilidad frente a cualquier reclamo o demanda por parte de terceros de manera exclusiva.
- Cede a la Universidad Técnica de Machala de forma exclusiva con referencia a la obra en formato digital los derechos de:
 1. Incorporar la mencionada obra en el repositorio institucional para su demostración a nivel mundial, respetando lo establecido por la Licencia *Creative Commons Attribution-NoCommercial* – Compartir Igual 4.0 Internacional (CC BY NCSA 4.0); la Ley de Propiedad Intelectual del Estado Ecuatoriano y el Reglamento Institucional.
 2. Adecuarla a cualquier formato o tecnología de uso en INTERNET, así como correspondiéndome como Autor la responsabilidad de velar por dichas adaptaciones con la finalidad de que no se desnaturalice el contenido o sentido de la misma.

KEVIN ADRIÁN VALAREZO PAZ

C.I.: 0704640333

Machala, 2022/octubre/12

RESUMEN

Debido a la emergencia sanitaria ocasionada por el COVID-19 se tomó la decisión de suspender las clases presenciales en todos los niveles de estudio, lo que dio origen al despliegue de modalidades de aprendizaje a distancia. El distanciamiento social motivó a que se deban organizar las prácticas en grupos reducidos para que los estudiantes utilicen los dispositivos e instrumentos de forma correcta y segura, sumado a la falta de recursos para el desarrollo de prácticas académicas induce a que exista una reducción en el desarrollo de las habilidades de aprendizaje en los estudiantes de una asignatura. La ausencia de herramientas y recursos para la realización de prácticas académicas es uno de los motivos para que la experiencia de los estudiantes se vea limitada.

Los laboratorios remotos son una excelente respuesta a estas problemáticas, debido a que permiten a los estudiantes realizar experimentos con componentes en tiempo real de forma remota a través de internet. El presente trabajo muestra la implementación de una arquitectura IoT basada en microservicios, la cual despliega un sistema de gestión de acceso a laboratorios remotos a través de API-REST junto con el protocolo de comunicación MQTT y WebSockets. El desarrollo del sistema de gestión de acceso utilizó el enfoque metodológico DevOps. Para la automatización de la integración y despliegue continuo se emplearon las herramientas Gitlab, Docker, Jinkens, Slack.

El sistema implementado fue puesto a prueba con cuatro prácticas de laboratorio y 38 usuarios remotos concurrentes, con un bajo uso recursos de ancho de banda, memoria RAM y CPU del hardware utilizado, con una latencia menor a un segundo para las interacciones del sistema con dichos usuarios. Como el sistema es una aplicación web, se le aplicaron criterios de calidad de la norma ISO/IEC 9126 cuya calificación obtenida fue de 4.74/5, y una auditoria de calidad de rendimiento, accesibilidad, buenas prácticas y posicionamiento con la herramienta Lighthouse de Google cuyos criterios de calificación superaron el 90%.

PALABRAS CLAVES: Laboratorios Remotos, IoT, Microservicios, DevOps, MQTT

ABSTRACT

Due to the health emergency caused by COVID-19, the decision was made to suspend face-to-face classes at all levels of study, which led to the deployment of distance learning modalities. Social distancing led to the need to organize practices in small groups so that students could use the devices and instruments correctly and safely, and the lack of resources for the development of academic practices led to a reduction in the development of learning skills in the students of a subject. The absence of tools and resources for academic practices is one of the reasons for the limited experience of students.

Remote laboratories are an excellent answer to these problems, since they allow students to perform experiments with components in real time remotely through the Internet. This work shows the implementation of an IoT architecture based on microservices, which deploys an access management system to remote laboratories through API-REST together with the MQTT communication protocol and WebSockets. The development of the access management system used the DevOps methodological approach. For automation of continuous integration and deployment the tools Gitlab, Docker, Jinkens, Slack were used.

The implemented system was tested with four laboratory practices and 38 concurrent remote users, with a low use of bandwidth, RAM and CPU resources of the hardware used, with a latency of less than one second for the interactions of the system with these users. As the system is a web application, quality criteria of the ISO/IEC 9126 standard were applied, with a score of 4.74/5, and a quality audit of performance, accessibility, good practices and positioning with Google's Lighthouse tool, whose qualification criteria exceeded 90%.

KEYWORDS: Remote Labs, IoT, Microservices, DevOps, MQTT

ÍNDICE GENERAL

	Pág.
DEDICATORIA	2
AGRADECIMIENTOS	3
RESPONSABILIDAD DE AUTORÍA	4
REPORTE DE SIMILITUD TURNITIN	5
CERTIFICACIÓN DEL TUTOR	6
CESIÓN DE DERECHOS DE AUTOR	7
RESUMEN	8
ABSTRACT	9
ÍNDICE GENERAL	10
ÍNDICE DE ILUSTRACIONES	12
ÍNDICE DE TABLAS	14
LISTA DE ABREVIATURAS Y SÍMBOLOS	15
INTRODUCCIÓN	16
CAPÍTULO 1. MARCO TEÓRICO	19
1.1. ANTECEDENTES HISTÓRICOS	19
1.1.1. Historia de los laboratorios remotos	19
1.1.2. Revisión Sistemática de Literatura	22
1.2. ANTECEDENTES CONCEPTUALES Y REFERENCIALES	29
1.2.1. Hipótesis de la investigación	29
1.2.2. Matriz de Operacionalización de variables	29
1.2.3. Variable independiente	30
1.2.4. Variable dependiente	30
1.2.5. Fundamentación teórica	30
1.3. ANTECEDENTES CONTEXTUALES	33
1.3.1. Problema	33
1.3.2. Implementación	33
1.3.3. Escenario de implementación	34
1.3.4. Objetivo General	34
1.3.5. Objetivos Específicos	34
CAPÍTULO 2. MATERIALES Y MÉTODOS	35
2.1. TIPO DE ESTUDIO O INVESTIGACIÓN	35
2.2. PARADIGMA O ENFOQUE	36

2.3.	POBLACIÓN Y MUESTRA	36
2.4.	MÉTODOS TEÓRICOS CON LOS MATERIALES UTILIZADOS	37
2.5.	MÉTODOS EMPÍRICOS CON LOS MATERIALES UTILIZADOS	37
2.6.	TÉCNICAS ESTADÍSTICAS PARA EL PROCESAMIENTO DE DATOS	38
CAPÍTULO 3. DESARROLLO DEL PROTOTIPO		39
3.1.	IMPLEMENTACIÓN DE ARQUITECTURA IOT	39
3.1.1.	Arquitectura IoT	39
3.1.2.	Distribución de recursos y estructura de RED	41
3.1.3.	Enrutamiento	41
3.2.	DESARROLLO DE PROTOTIPO DE SOFTWARE	42
3.2.1.	Fase de planificación	43
3.2.2.	Fase de codificación	53
3.2.3.	Fase de construcción	61
3.2.4.	Fase de pruebas	62
3.2.5.	Fase de despliegue	64
3.2.6.	Fase de lanzamiento	66
3.2.7.	Fase de operación	68
3.2.8.	Fase de monitoreo	69
CAPÍTULO 4. EXPERIMENTACIÓN Y RESULTADOS		71
4.1.	CONFIGURACIÓN EXPERIMENTAL	71
4.1.1.	Objetivos de la experimentación	71
4.1.3.	Características de equipos	72
4.1.4.	Caso de estudio	73
4.2.	HALLAZGOS FUNDAMENTALES	73
4.2.1.	Prueba de concepto	74
4.2.2.	Comprobación de caso de estudio	82
4.2.3.	Evaluación de Calidad	83
4.2.4.	Latencia de comunicación	86
4.2.5.	Concurrencia de usuarios	90
CONCLUSIONES		94
RECOMENDACIONES		95
BIBLIOGRAFÍA		96

ÍNDICE DE ILUSTRACIONES

	Pág.
Ilustración 1: Campos del conocimiento de laboratorios remotos	21
Ilustración 2: Filtros de la Fase de Revisión	24
Ilustración 3: Causas y consecuencias del problema de investigación	33
Ilustración 4: Materiales y Métodos de la investigación	35
Ilustración 5: Arquitectura IoT basada en microservicios	40
Ilustración 6: Distribución de recursos y arquitectura de red	41
Ilustración 7: Ciclo de vida DevOps	42
Ilustración 8: Diagrama de Casos de uso de Administración	47
Ilustración 9: Diagrama de Casos de uso de Unidad Académica	48
Ilustración 10: Diagrama de Casos de uso de Académico	48
Ilustración 11: Diagrama de Casos de uso de Laboratorio	49
Ilustración 12: Diagrama de Casos de uso de Laboratorio	50
Ilustración 13: Cadena de valor del proyecto	50
Ilustración 14: Flujo de valor y capacidades del Módulo Administración	51
Ilustración 15: Flujo de valor y capacidades de Unidad Académica	51
Ilustración 16: Flujo de valor y capacidades del microservicio Laboratorio	52
Ilustración 17: Flujo de valor y capacidades del microservicio Laboratorio	52
Ilustración 18: Flujo de valor y capacidades del microservicio Reservación	53
Ilustración 19: Flujo de valor y capacidades del microservicio Comunicación	53
Ilustración 20: Diseño de Base de Datos Relacional de Administración	54
Ilustración 21: Diseño de Base de Datos Relacional de Unidad Académica	54
Ilustración 22: Diseño de Base de Datos Relacional de Académico	55
Ilustración 23: Modelo de documento No Relacional de Laboratorio	55
Ilustración 24: Modelo de documento No Relacional de Reservación	56
Ilustración 25: Prototipo de pantalla gestión de usuarios	56
Ilustración 26: Prototipo de pantalla de configuración de curso	57
Ilustración 27: Prototipo de pantallas de configuración de laboratorio	58
Ilustración 28: Prototipo de pantalla de listado de cursos disponibles	58
Ilustración 29: Prototipo de pantalla de acceso a Curso	59
Ilustración 30: Prototipo de pantalla de reserva de laboratorio	59
Ilustración 31: Prototipo de pantallas de escenario	60
Ilustración 32: Pycharm 2022.1.3 Community Edition	60
Ilustración 33: Visual Studio Code	61
Ilustración 34: Interfaz de Konga - Kong	61
Ilustración 35: Configuración de los Jenkinsfile	62
Ilustración 36: Configuración de pipeline de pruebas en Gitlab Runner	63
Ilustración 37: Pipelines ejecutados para la comprobación de una aplicación	64
Ilustración 44: Pipeline de despliegue del microservicio Laboratorio	65
Ilustración 45: Tareas de despliegue de los microservicios	66
Ilustración 40: Aplicaciones alojadas en GitLab.com	66
Ilustración 46: Historial de commits en el repositorio de GitLab	67

Ilustración 47: Versión de compilación de pipeline	68
Ilustración 43: Contenedores de los microservicios en máquina 2	68
Ilustración 44: Contenedores de los microservicios en máquina 2	69
Ilustración 45: Consumo de recurso del contenedor Laboratorio	69
Ilustración 46: Canal de comunicación Slack #remote-labs	70
Ilustración 46: Revisión de Logs Contenedor	70
Ilustración 48: Escenarios de experimentación	72
Ilustración 49: Maqueta de ESP32 adaptada con cámara de video	73
Ilustración 50: Inicio de sesión de un usuario administrador	74
Ilustración 51: Interfaz de gestión de usuarios del sistema	75
Ilustración 52: Interfaz de gestión de los cursos	75
Ilustración 53: Acceso a la configuración de Curso	76
Ilustración 54: Vista de participantes y recursos	76
Ilustración 55: Acceso a la gestión de laboratorios	77
Ilustración 56: Configuración de conexión de laboratorios remotos	77
Ilustración 57: Tipos de configuración de escenarios de laboratorios remotos	78
Ilustración 58: Inicio de sesión de usuario con rol docente	78
Ilustración 59: Inicio de sesión de usuario con rol estudiante	79
Ilustración 60: Proceso de automatriculación a los cursos	79
Ilustración 61: Proceso de automatriculación a los cursos	80
Ilustración 62: Sala de espera de laboratorios	80
Ilustración 63: Escenarios de prácticas de laboratorio	81
Ilustración 64: Escenarios de prácticas de laboratorio	81
Ilustración 65: Escenario de pruebas de caso de estudio	82
Ilustración 66: Estudiantes de la asignatura de IOT probando el sistema	82
Ilustración 67: Secuencia de pasos del escenario de prácticas del caso de estudio	83
Ilustración 68: Resultados de auditoría realizada con Lighthouse	85
Ilustración 69: Simulador de pruebas de transmisión de datos	86
Ilustración 70: Proceso de envío de tramas de datos	87
Ilustración 71: Diagrama de cajas de los datos de tramas enviadas	87
Ilustración 72: Histograma de frecuencias de datos de envío de tramas	88
Ilustración 73: Proceso de envío de tramas de datos	88
Ilustración 74: Diagrama de cajas de los datos de recepción de tramas	89
Ilustración 75: Histograma de frecuencias de datos de recepción de tramas	89
Ilustración 76: Historial de acceso a laboratorios proporcionado con el sistema	90
Ilustración 77: Usuarios con más conexiones en laboratorios	91
Ilustración 78: Utilización de laboratorios	91
Ilustración 79: Consumo de recursos con 21 usuarios conectados simultáneamente	92
Ilustración 80: Consumo de recursos con 38 usuarios conectados simultáneamente	93

ÍNDICE DE TABLAS

	Pág.
Tabla 1: Criterios de Inclusión y Exclusión	23
Tabla 2: Resultados de Cadena de Búsqueda	23
Tabla 3: Resumen de Fase de revisión	24
Tabla 4: Resultados de la Revisión Bibliográfica	28
Tabla 5: Matriz de Operacionalización de variables	29
Tabla 6: Características del Equipo Anfitrión	42
Tabla 7: Identificación de módulos del sistema	43
Tabla 8: Interesados del proyecto	43
Tabla 9: Funciones del Administrador	44
Tabla 10: Funciones del Docente	44
Tabla 11: Funciones del Estudiante	44
Tabla 12: Historias de usuario – Requisitos funcionales	46
Tabla 13: Requisitos No Funcionales	47
Tabla 14: Objetivos de los experimentos y métricas	71
Tabla 15: Características del Equipo Anfitrión del servidor	72
Tabla 16: Características de los equipos cliente	72
Tabla 17: Evaluación de calidad bajo la norma ISO/IEC 9126	84
Tabla 18: Estadística descriptiva del envío de tramas	87
Tabla 19: Estadística descriptiva del envío de tramas	88
Tabla 20: Registro de sesiones de los laboratorios remotos	92

LISTA DE ABREVIATURAS Y SÍMBOLOS

API: Interfaz de Programación de Aplicaciones

DevOps: Development and Operations

IDE: Integrated Development Environment

IES: Institución de Educación Superior

IoT: Internet de las Cosas

IP: Protocolo de Internet

MQTT: Message Queuing Telemetry Transport

RSL: Revisión Sistemática de la Literatura

SO: Sistema Operativo

INTRODUCCIÓN

Debido a la emergencia sanitaria ocasionada por el COVID-19, en el Ecuador, como en varios países de Latinoamérica, se decidió de suspensión de las clases presenciales en todos los niveles de estudio, lo que dio origen de modalidades de aprendizaje a distancia [1]. El distanciamiento social motivó a que se deban organizar las prácticas en grupos lo más reducido posible para que los estudiantes utilicen los dispositivos e instrumentos de forma correcta y segura. La ausencia de herramientas y recursos para la realización de prácticas académicas es otro de los motivos por que la experiencia de los estudiantes en la ejecución de prácticas académicas se ha visto limitada. Las experiencias de laboratorio juegan un papel considerable en la educación, son esenciales para que los estudiantes comprendan y consoliden los aprendizajes teóricos. Los laboratorios remotos ofrecen a los usuarios la flexibilidad de uso de los laboratorios físicos optimizando los recursos [2], son aplicaciones que permiten a los estudiantes trabajar de forma remota con instrumentos y hardware reales. Por lo general, estos laboratorios remotos proporcionan una cámara web que apunta al instrumento para aumentar la percepción de los estudiantes sobre el uso de equipos reales en lugar de simplemente usar una simulación [3]. La implementación de un sistema que gestione los laboratorios remotos es de gran importancia puesto que, permitiría el acceso a más artefactos electrónicos destinados a la realización de prácticas académicas a medida que se los vaya desarrollando.

En medio de la pandemia mundial desencadenada por el COVID-19, mantener a los estudiantes motivados sin el apoyo presencial de los maestros y la interacción con los compañeros de clase, llega a ser un gran desafío. La limitación de recursos y la falta de acceso a laboratorios especializados son otros de los motivos por los que los métodos de enseñanza-aprendizaje se han visto condicionados. Existen múltiples implementaciones de laboratorios remotos, la mayoría de ellos son enfocados a el desarrollo e implementación de los instrumentos usados en diferentes campos de estudio. El trabajo propuesto por Moussa [4], realiza el diseño de un modelo para compartir laboratorios electrónicos analógicos a los estudiantes; el trabajo de Nenov [5] se centra en desarrollar prácticas con los motores de corriente continua; mientras que, el trabajo de Guc [6] se centra en desarrollar una experiencia de aprendizaje de laboratorio remoto para la educación en el campo de la mecatrónica. Todas estas investigaciones ofrecen soluciones monolíticas en sus arquitecturas de red para brindar soluciones que cubren sus casos de estudio específicos. Trabajos como los de Fujii [7], se orientan a brindar una solución

orientada a Laboratorios como servicio, dando otro enfoque en la utilización de laboratorios remotos. En cambio, el trabajo de Krzysztof Tokarz et al. [8] propone un modelo de infraestructura para el uso de laboratorios remotos dentro del campo del Internet de las Cosas, en la que se discute el uso de tecnologías y plataformas relacionadas al IoT que se pueden aprovechar para el dictado de cursos a distancia.

La idea de tener estaciones experimentales remotas en las instituciones educativas es bastante atractiva. Cada experimento requiere un costo de esfuerzo inicial específico por parte del educador, sin embargo, esta cantidad de tiempo dedicado debe compararse con el tiempo necesario en las configuraciones experimentales habituales. Por lo tanto, registrar y cargar experimentos puede requerir un mayor costo de esfuerzo al principio, pero es muy probable que requiera menos tiempo y menor costo de mantenimiento posteriormente [9]. Ante esto, la pandemia mundial intensificó la necesidad de contar con herramientas que permitan una mejor adaptación de aprendizaje remoto. Campos de estudio como la electrónica, mecánica, robótica, incluso el Internet de las Cosas, pueden sacarle mayor provecho a una implementación de laboratorios remotos, debido a que, se pueden preparar experimentos para que los estudiantes realicen prácticas y así reforzar sus actividades académicas, este hecho hace necesario de contar con un sistema que gestione la comunicación con los experimentos y una arquitectura que permita la escalabilidad de estos laboratorios que aborden diversos campos de estudio.

A pesar de existir varios tipos de implementaciones para los laboratorios remotos, desde desarrollos sencillos como los propuestos por Moussa [4] o Guerra [10], o basados en servicios en nube como los de Vitliemov [11] o Fujii [7]. La naturaleza de esta clase de laboratorios hace pensar en que se pueda implementar una solución IOT, en la que se crea un smart traducer (mote), lo que permite conectar el laboratorio a una interfaz de forma automática, y de esta manera poder escalar en cantidad de laboratorios para una institución educativa. Para esto, se plantea desarrollar un sistema web que gestione el acceso a los artefactos electrónicos desarrollados en la asignatura de Internet de las Cosas, mismos que serán empleados como dispositivos de experimentación de los laboratorios remotos que se vayan implementando en una Institución de Educación Superior (IES).

Un entorno experimental que pueda manejar el desarrollo y los experimentos de un campo de estudio determinado, se vuelve importante, por el motivo que es requerido contar con una plataforma que cubra la necesidad de realizar prácticas experimentales dentro del desarrollo de las clases. Como se observa en el trabajo de Fujii [7], se implementa una solución en nube donde a través de laboratorios remotos buscan realizar experimentos de

prácticas en el campo de Internet de las Cosas. La presente investigación busca implementar un sistema que gestione el acceso a laboratorios remotos mediante una arquitectura IOT basada en microservicios para la realización de prácticas académicas. A diferencia del trabajo propuesto por Guan et. al [12] el cual trabajan únicamente con WebSockets para comunicarse con los laboratorios remotos, esta investigación añade el uso de microservicios para distribuir la carga del control de la disponibilidad de los laboratorios y también añade la capa de protocolos de telemetría para comunicarse con los dispositivos electrónicos, lo que permite que la capa de WebSockets y microservicios controlen las acciones realizadas por los usuarios en el escenario de prácticas, en cambio, el protocolo de telemetría recibe la información generada por los WebSockets y lo publica para los dispositivos electrónicos.

Esta investigación es realizada dentro de un ambiente de laboratorio, cuyo objeto de estudio es la implementación de un sistema de gestión de acceso a laboratorios remotos con una arquitectura IOT usando técnicas de microservicios en la asignatura de Internet de las Cosas, como parte de la malla curricular vigente de la carrera de Ingeniería en Tecnologías de la Información de una IES; asignatura que busca brindar el acceso a los recursos hardware que se posee y/o generan dentro de las actividades académicas, cuya población de pruebas se considera a los estudiantes que han cursado esta materia.

Por lo anteriormente expuesto, se busca resolver la hipótesis de que si un sistema de gestión de laboratorios remotos incrementa el acceso a recursos para la realización de prácticas académicas. Para cumplir con el objetivo planteado, se realiza la implementación de una arquitectura IoT basada en microservicios, donde se distribuye los contenedores en donde se despliegan los contenedores que albergan los servicios y aplicaciones desarrolladas, la comunicación entre los aplicativos se realiza mediante el API-REST mientras que la comunicación con los laboratorios es realizada con telemetría. El desarrollo del sistema de gestión de acceso fue realizado con el uso de la metodología DevOps, debido al ofrecimiento de herramientas que facilitan la integración y el despliegue continuo de la aplicación. El sistema de gestión de acceso, mediante un ambiente de bajos recursos, controla los tiempos de utilización y la disponibilidad de los laboratorios remotos para una o varias asignaturas, a su vez de gestionar la accesibilidad de los usuarios y la ejecución de comandos remotos en la realización de prácticas de laboratorio.

CAPÍTULO 1. MARCO TEÓRICO

En el presente capítulo se establecen los aspectos relevantes del marco teórico mediante tres secciones. La primera sección se basa en la descripción de los antecedentes históricos, luego en la segunda sección se abordan los antecedentes conceptuales y finalmente en la tercera sección se detallan los antecedentes contextuales de esta investigación.

1.1. ANTECEDENTES HISTÓRICOS

En este apartado se considera la técnica utilizada para la recopilación, análisis y comprensión de los temas investigados. Una Revisión Sistemática de la Literatura (RSL) es un método para evaluar e interpretar la investigación disponible relacionada con una pregunta de investigación, un tema o un fenómeno de interés [13]. Antes de realizar la RSL se revisa una revisión de la evolución en la historia de los laboratorios remotos.

1.1.1. Historia de los laboratorios remotos

La experimentación es un componente necesario para el aprendizaje de ciencia y tecnología. Los laboratorios tradicionales se utilizan para este propósito y son limitados en comparación con el número de estudiantes [14]. Dado que el desarrollo de estas instalaciones de laboratorio requiere una gran cantidad de recursos humanos y económicos, los laboratorios remotos proporcionan una solución alternativa aceptable para los estudiantes que se ven privados de acceso directo al laboratorio [15]. Desde 1996, junto a la educación a distancia aparecieron los primeros indicios de laboratorios remotos. Los cursos que contenían laboratorio a través de video bidireccional con conexión de audio bidireccional, los administradores consideraban el nivel de participación del instructor del sitio remoto para que supervisara la transmisión del curso y monitoreara el laboratorio [16], hasta un instructor que audita el curso y realiza los ejercicios antes de la hora programada.

El proyecto desarrollado por Virtual Instruments System In Reality (VISIR). Fue uno de los primeros pasos en laboratorios remotos, que comenzó como un proyecto de investigación en el Instituto de Tecnología Blekinge, Suecia por el equipo de Gustavsson en 1999 [17]. La idea base de los laboratorios remotos es compartir y poner en común recursos tecnológicos, esto implica la creación de plataformas que utilicen el mismo idioma, ofrezcan los mismos servicios e intercambien los mismos formatos de datos. Para la manipulación de dispositivos electrónicos, se encuentra LabView, creado en 1983. Un entorno de desarrollo para diseñar sistemas utilizando lenguaje de programación visual diseñado para sistemas de pruebas, de control y diseño, simulados o reales [18], su

licencia es propietaria, y en 1988 se realizaron las primeras implementaciones de esta plataforma en la educación superior. Este software fue muy utilizado en los primeros años para la implementación de laboratorios remotos, debido a que su naturaleza facilitaba su utilización.

Para el 2001, se desarrolló un prototipo de sistema cliente-servidor para realizar experimentos de forma remota en motores de corriente continua sin escobillas, incluido material didáctico basado en la web y otro software [19]. La computadora del servidor estaba conectada al laboratorio de motores, y la imagen visual junto los sonidos del experimento se transmitían a la computadora del cliente en tiempo real. Otro ejemplo de instrumentación remota para ejercicios de corriente continua proponía una arquitectura cliente-servidor [20], en la que se requería una interfaz de usuario para imitar el instrumento de medición real, ejecutándose en una red heterogénea de computadoras sin necesidad de una instalación previa de software especializado. También se proporcionaba una retroalimentación visual por medio de WebCams conectadas a un servidor de video, en las que las capturas de pantalla mostraban a los estudiantes lo que realmente sucedía en la pantalla resultando útiles para la resolución remota de mediciones de corriente continua y corriente alterna.

Por el 2002, mientras aún se seguía desarrollando interfaces de usuario con los applets de java [21], se lograba la conexión de laboratorios remotos a través del internet usando microcontroladores [22], en los que debido a la tecnología de conexión a internet en ese tiempo se consideraba experimentos sin transmisión de audio o video [23]. Muchas de las implementaciones de estos laboratorios orientaban sus experimentos en la educación en ingeniería eléctrica [24], donde buscaban la manera óptima de aprovechar los recursos limitados de transmisión de datos para una percepción adecuada de la práctica. En el 2004, con la utilización de la matriz de puertas programables (FPGA por sus siglas en inglés) en el desarrollo de laboratorios remotos, se introdujo la experimentación en los cursos de informática [25]. Los estudiantes pueden ir más allá de la simulación, probando sus diseños de microprocesador en hardware e interactuando con un sistema real. Luego en el 2005 se comienza a la web 2.0 [26] y en el 2008 se dan los primeros pasos en la utilización de dispositivos móviles para el acceso a los laboratorios remotos [27]. En el 2006, los laboratorios remotos comienzan a ser utilizados en entornos e-learning [28]. Las arquitecturas donde son implementados comienzan a tener diversas capas. La separación entre las funcionalidades específicas del equipo y las relacionadas con la medición y el control del acceso, la transferencia y la gestión de datos [29].

A medida que avanza el tiempo, se han desarrollado múltiples soluciones de laboratorios remotos para distintos campos del conocimiento, como se observa en la ilustración 1, tales como lo son: la electrónica [30]–[34], realidad virtual y aumentada [35][36], mecatrónica [6], [37], [38], robótica [39]–[41], física [42][43], Internet de las Cosas [44]–[48], entre otros. Las arquitecturas implementadas para la comunicación de estos laboratorios remotos con los usuarios se basan en el cliente-servidor, enfocándose más, en el desarrollo de los instrumentos para la realización de las practicas experimentales, en comparación de brindar soluciones para la gestión de acceso a los laboratorios remotos. Uno de los siguientes pasos es diseñar y evaluar una solución que garantice el acceso exclusivo al instrumento entre muchos estudiantes sin conflictos con un dispositivo de bajo costo [4].

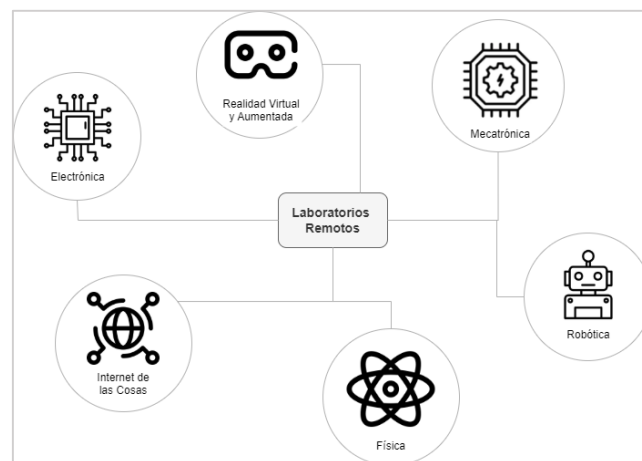


Ilustración 1: Campos del conocimiento de laboratorios remotos

En medio de la pandemia mundial desencadenada por el nuevo coronavirus también conocido como Sars-CoV-2, además de estar sujeta a la distancia social y al bloqueo de instalaciones como medio de contención, la población se ha enfrentado a numerosos problemas derivados de la gravedad de la crisis de salud, con implicaciones en diferentes áreas, como salud [49], economía [50], bienestar social [51], educación [52], entre otras. Debido al impacto de COVID-19, todas las experiencias de aprendizaje, incluidos los procesos de laboratorio, se debieron adaptar al acceso remoto. Los laboratorios remotos son alternativas para las instituciones educativas que no cuentan con laboratorios de aula, o que pretenden ampliar sus escasos recursos, o que deseen compartir equipos con otras instituciones, y así permitir que un mayor número de estudiantes obtenga conocimientos prácticos en diversas áreas del conocimiento [53]. En este sentido, es importante destacar la transformación que representó la educación remota de emergencia en los procesos de enseñanza y aprendizaje. Los estudiantes que no habían experimentado el aprendizaje a

distancia, totalmente mediados por las tecnologías, tuvieron que reajustarse repentinamente a este nuevo formato de aprendizaje [54].

El Internet de las Cosas (IOT) es una de las tendencias que va en aumento. Todo tipo de objetos cotidianos se pueden conectar a Internet desde un simple foco hasta los automóviles. El IOT es una evolución de la aplicación móvil integrada y todo lo que está conectado a Internet para integrar una mayor capacidad de comunicación y utilizar análisis de datos para extraer información significativa [55]. Controlar equipos eléctricos y electrónicos de manera remota, se deriva del concepto conocido como la Computación Ubicua, desde principios de la década de 1990 [56], siendo por primera vez en el año 1999 que Kevin Ashton utilizó el término “Internet de las Cosas” en el estándar de etiquetado de objetos usando identificación por radiofrecuencia en red (RFID) para aplicaciones logísticas [57]. Desde ese entonces, se ha trabajado en muchos sistemas que se centran en redes de sensores, tecnologías de middleware, nube y redes de comunicación.

1.1.2. Revisión Sistemática de Literatura

El método de Bárbara Kitchenham tiene como objetivo evaluar y analizar todas las fuentes relevantes en cuanto a las preguntas de investigación. Esta guía metodológica propone un conjunto de fases, desde la definición de preguntas de investigación hasta la publicación de resultados.

Preguntas de investigación

En esta fase se establecieron las siguientes preguntas para la búsqueda de información acerca de los laboratorios remotos, además de tecnologías empleadas y sus campos de estudio aplicado.

RQ1: ¿Qué prácticas experimentales académicas han sido implementadas en los laboratorios remotos?

RQ2: ¿Cómo son los laboratorios remotos orientados a realizar prácticas experimentales de IOT?

RQ3: ¿Qué características debe poseer la arquitectura IoT para proporcionar un enfoque de desarrollo basado en microservicios?

Definir criterios de inclusión y exclusión para la RSL

Para la selección de los estudios se consideraron los títulos y resúmenes que corresponden en los que se denotará los siguientes criterios de inclusión y exclusión:

Criterios de inclusión
<ul style="list-style-type: none"> • Escritos solamente en Ingles y de los últimos 5 años; • Describe la utilización de Internet de las Cosas en el desarrollo o aplicación de Laboratorios Remotos; • Fuentes primarias únicamente (Artículos de revista y conferencias) <p>Temas principales: Remote Labs AND Internet of Things;</p>
Criterios de exclusión
<ul style="list-style-type: none"> • Estudios no incluidos en las bases de datos seleccionadas y Estudios duplicados <p>Otros criterios: Para estudios duplicados se dará prioridad a los de revistas.</p>

Tabla 1: Criterios de Inclusión y Exclusión

Identificar las bases de datos y motores de búsqueda que se van a utilizar

Para el proceso de búsqueda se tomó en consideración varias bases de datos de referencias bibliográficas, citas y editoras académicas, tales como:

- IEEE Xplorer
- Springer
- Web of Science
- Science Direct
- MDPI

Definir los términos de búsqueda;

Se toma en consideración los vocablos o frases que expresan los criterios de búsqueda que se aplican en cada fuente o plataforma de base de datos científica, tales como, título, índice, palabras claves, búsquedas avanzadas, metadatos, citación. Como es lógico se requirió de la siguiente ecuación de búsqueda tomando en cuenta los operadores AND y OR junto a los términos en inglés:

Término 1: Remote Laboratory AND Término 2: Internet of Things

Cadena de búsqueda: "Remote Laboratory" AND ("Internet Of Things" OR "IOT") OR Término 3: B-Learning OR Término 4: Microservice

Buscar en bases de datos científicas

Se realizó mediante la indagación en fuentes aceptadas ampliamente en la comunidad científica, es por ello que se utilizan revistas científicas de alto impacto y actas de congresos de relevancia.

Cadena de búsqueda sugerida	Resultados	
"Remote Laboratory" AND ("Internet Of Things" OR "IOT " OR "B-Learning" OR "Microservice")	IEEE	286
	Springer	370
	Web Of Science	91
	MDPI	33
	Science Direct	170

Tabla 2: Resultados de Cadena de Búsqueda

Como se observa en la tabla 2, los resultados obtenidos mediante la cadena de búsqueda definida anteriormente arrojan una cantidad de estudios adecuados para ser considerados en la fase de revisión.

Fases de Revisión

En la fase de revisión, se realizó la selección de los estudios que cumplan con los criterios de inclusión y exclusión, y se fue aplicando los filtros respectivos hasta obtener un resultado de cantidad de artículos:

Fuentes	Resultados	quitando duplicados	por título	por resumen	todo artículo	% por Fuente
IEEE	286	261	165	88	19	30.11%
Springer	370	369	67	33	8	38.95%
Web Of Science	91	55	25	9	2	9.58%
MDPI	33	31	8	4	0	3.47%
Science Direct	170	19	19	4	3	17.89%
Total	950	735	284	138	32	100%

Tabla 3: Resumen de Fase de revisión

En la tabla 3, se observa cuáles fueron los criterios de selección para aplicar los filtros en el proceso de selección de las investigaciones encontradas según la cadena de búsqueda aplicada en las diferentes fuentes de bases de datos ya definidas.

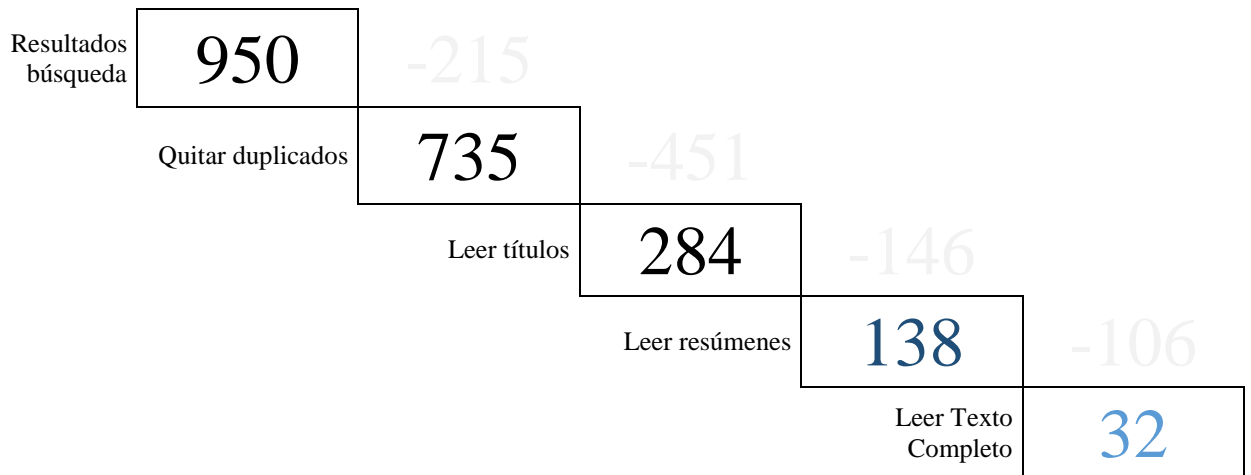


Ilustración 2: Filtros de la Fase de Revisión

Extracción de Datos

La extracción de los datos, se realizó llenando una matriz con los datos más relevantes de los estudios primarios. Considerando los siguientes atributos: título, base de datos, tipo de papers, revista, DOI, año, páginas.

Presentación de Resultados

De acuerdo con la revisión realizada a las diferentes fuentes, y la selección de los artículos que servirán como referencia para el desarrollo de este trabajo, en la Tabla 4 se exponen los siguientes resultados en base a las preguntas de investigación y a los criterios de inclusión y exclusión:

No.	TÍTULO	AÑO	AUTORES	BASE DE DATOS	TIPO	REVISTA	DOI	CITAS	PÁGINAS
1	Implementing a remote laboratory on a chip	2017	O. M. Machidon A. L. Machidon P. A. Cotfas D. T. Cotfas	IEEE	Conference Paper	2017 IEEE 23rd International Symposium for Design and Technology in Electronic Packaging (SIITME)	10.1109/SIITME.2017.8259880	1	155-158
2	IoT Remote Group Experiments in the Cyber Laboratory: A FPGA-based Remote Laboratory in the Hybrid Cloud	2017	N. Fujii N. Koike	IEEE	Conference Paper	2017 International Conference on Cyberworlds (CW)	10.1109/CW.2017.29	11	162-165
3	Design and implementation remote laboratory based on Internet of Things: Study case in diffraction grating experiment	2017	W. C. Karunianto A. H. Saputro	IEEE	Conference Paper	2017 International Conference on Computer, Control, Informatics and its Applications (IC3INA)	10.1109/IC3INA.2017.8251756	4	143-146
4	Proposal of Component Based Architecture for Internet of Things: online laboratory case study	2017	Miroslav Gula Katarína Žáková	Science Direct	Journal	IFAC-PapersOnLine	10.1016/j.ifacol.2017.08.153	4	337-342
5	Design and Performance Evaluation of Containerized Microservices on Edge Gateway in Mobile IoT	2018	Amit Singh Gaur Jyoti Budakoti Chung-Horng Lung	IEEE	Conference Paper	2018 IEEE International Conference on Internet of Things and IEEE Green Computing and Communications and IEEE Cyber, Physical and Social Computing and IEEE Smart Data	10.1109/Cybermatics_2018.2018.00055	4	138-145
6	Models and Smart Adaptive Interfaces for the Improvement of the Remote Laboratories User Experience in Education	2018	Luis Felipe Zapata Rivera Maria M. Larrondo Petrie	Springer	Conference Paper	Online Engineering & Internet of Things	10.1007/978-3-319-64352-6_39	1	416-423
7	Work-in-Progress: A Smart Scheduling System for Shared Interactive Remote Laboratories	2018	Mohammed Moussa Abdelhalim Benachenhou Abderrahmane Adda Benatia	Springer	Conference Paper	Teaching and Learning in a Digital World	10.1007/978-3-319-73204-6_65	1	601-606
8	Remote Laboratories: For Real Time Access to Experiment Setups with Online Session Booking, Utilizing a Database and Online Interface with Live Streaming	2018	B. Kalyan Ram S. Arun Kumar S. Prathap B. Mahesh B. Mallikarjuna Sarma	Springer	Conference Paper	Online Engineering & Internet of Things	10.1007/978-3-319-64352-6_19	0	190-204

No.	TÍTULO	AÑO	AUTORES	BASE DE DATOS	TIPO	REVISTA	DOI	CITAS	PÁGINAS
9	Enabling the Automatic Generation of User Interfaces for Remote Laboratories	2018	Wissam Halimi Christophe Salzmann Hagop Jamkojian Denis Gillet	Springer	Conference Paper	Online Engineering & Internet of Things	10.1007/978-3-319-64352-6_73	7	778-793
10	Improving the Scalability and Replicability of Embedded Systems Remote Laboratories Through a Cost-Effective Architecture	2019	A. Villar-Martínez L. Rodríguez-Gil I. Angulo P. Orduña J. García-Zubía D. López-De-Ipiña	IEEE	Journal	IEEE Access	10.1109/ACCESS.2019.2952321	11	164164-164185
11	Generation of Multiple Interfaces for Hybrid Online Laboratory Experiments based on Smart Laboratory Learning Objects	2019	L. F. Zapata-Rivera M. M. Larrondo-Petrie C. P. Weinthal	IEEE	Conference Paper	2019 IEEE Frontiers in Education Conference (FIE)	10.1109/FIE43999.2019.9028421	1	1-8
12	Using Internet of Things to Enhance Remote Experiments in Learning Management Systems	2019	Khaleel Mershad Ali Hamieh	IEEE	Conference Paper	2019 IEEE International Smart Cities Conference (ISC2)	10.1109/ISC246665.2019.9071786	0	458-464
13	Open Digital Lab for You – Laboratory-based learning scenarios in education, research and qualification	2019	Anke Pfeiffer Dieter Uckelmann	IEEE	Conference Paper	2019 5th Experiment International Conference (exp.at'19)	10.1109/EXPAT.2019.8876560	6	36-41
14	Internet of Thing (IoT) Based Remote Labs in Engineering	2019	Tareq S. El-Hasan	IEEE	Conference Paper	2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)	10.1109/CoDIT.2019.8820591	3	976-982
15	Virtual/Remote Labs for Automation Teaching: a Cost Effective Approach	2019	Ruben Morales-Menendez Ricardo A. Ramírez-Mendoza Antonio Jr. Vallejo Guevara	Science Direct	Journal	IFAC-PapersOnLine	10.1016/j.ifacol.2019.08.219	19	266-271
16	Effects of remote labs on cognitive, behavioral, and affective learning outcomes in higher education	2019	Post, Lysanne S. Guo, Pengyue Saab, Nadira Admiraal, Wilfried	Web Of Science	Journal	COMPUTERS & EDUCATION	10.1016/j.compedu.2019.103596	40	103596
17	An Application of IoT for Conduct of Laboratory Experiment from Home	2020	T. V. Tran H. Takahashi T. Narabayashi H. Kikura	IEEE	Conference Paper	2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)	10.1109/IEMTRONICS51293.2020.9216375	2	1-4

No.	TÍTULO	AÑO	AUTORES	BASE DE DATOS	TIPO	REVISTA	DOI	CITAS	PÁGINAS
18	An Approach to Use Virtual and Remote Labs in Mechatronics Education Based on Cloud Services	2020	P. Vitliemov D. Bratanov M. Marinov	IEEE	Conference Paper	2020 7th International Conference on Energy Efficiency and Agricultural Engineering (EE&AE)	10.1109/EEAE49144.2020.9279072	1	1-4
19	Internet of Things Network Infrastructure for The Educational Purpose	2020	K. Tokarz P. Czekalski G. Drabik J. Paduch S. Distefano R. Di Pietro G. Merlino C. Scaffidi R. Sell G. S. Kuaban	IEEE	Conference Paper	2020 IEEE Frontiers in Education Conference (FIE)	10.1109/FIE44824.2020.9274040	1	1-9
20	Design and Development of Remote Laboratory System to Facilitate Online Learning in Hardware Programming Subjects	2020	H. S. Jo R. S. Jo	IEEE	Conference Paper	2020 13th International UNIMAS Engineering Conference (EnCon)	10.1109/EnCon51501.2020.9299326	1	1-5
21	Remote Implementation of Microcomputers Laboratory Practices: A Case Study	2020	W. K. Wong F. H. Juwono W. N. Loh	IEEE	Conference Paper	2020 Sixth International Conference on e-Learning (econf)	10.1109/econf51404.2020.9385488	0	214-218
22	Smart-Lab: Design and Implementation of an IoT-based Laboratory Platform	2020	Sabrine Khriji Dhouha El Houssaini Rim Barioul Talha Rehman Olfa Kanoun	IEEE	Conference Paper	2020 IEEE 6th World Forum on Internet of Things (WF-IoT)	10.1109/WF-IoT48130.2020.9221143	2	1-5
23	Remote Controlled Laboratory Experiments for Engineering Education in the Post-COVID-19 Era: Concept and Example	2020	Amr Karam Mohammed Helmy M. El Zoghby Mahmoud M. Elmesalawy	IEEE	Conference Paper	2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)	10.1109/NILES50944.2020.9257888	7	629-634
24	Blended Learning in Control Engineering Teaching; an Example of Good Practice	2020	J.A.Rossiter	Science Direct	Journal	IFAC PapersOnLine	10.1016/j.ifacol.2020.12.1797	1	17252-17257
25	Design of a Low Cost Switching Matrix for Electronics Remote Laboratory	2020	Abderrahmane Adda Benattia Abdelhalim Benachenhou Mohammed Moussa Abdelhamid Mebrouka	Springer	Conference Paper	Cyber-physical Systems and Digital Twins	10.1007/978-3-030-23162-0_15	1	155-164
26	Digital Twins in Remote Labs	2020	Heinz-Dietrich Wuttke Karsten Henke René Hutschenreuter	Springer	Conference Paper	Cyber-physical Systems and Digital Twins	10.1007/978-3-030-23162-0_26	5	289-297

No.	TÍTULO	AÑO	AUTORES	BASE DE DATOS	TIPO	REVISTA	DOI	CITAS	PÁGINAS
27	A flexible remote laboratory with programmable device under test	2020	Angrisani, Leopoldo Bonavolonta, Francesco D'Arco, Mauro Liccardo, Annalisa	Web Of Science	Journal	MEASUREMENT	10.1016/j.measurement.2020.107584	35	107584
28	Application of Open-Hardware-Based Solutions for Rapid Transition From Stationary to the Remote Teaching Model During Pandemic	2021	A. Bekasiewicz B. Pankiewicz M. Wojcikowski M. Klosowski S. Koziel	IEEE	Journal	IEEE Transactions on Education	10.1109/TE.2020.3043479	0	299-307
29	Design and Evaluation of a User Experience Questionnaire for Remote Labs	2021	J. Cuadros V. Serrano J. García-Zubía U. Hernandez-Jayo	IEEE	Journal	IEEE Access	10.1109/ACCESS.2021.3069559	3	5022-50230
30	An Open Platform for Hardware-in-Loop Virtual Simulation via Internet of Things	2021	Ke Dong Tianle Yu Pengfei Wu	IEEE	Conference Paper	2021 2nd Information Communication Technologies Conference (ICTC)	10.1109/ICTC51749.2021.9441658	0	225-229
31	Remote Labs in Germany—An Overview About Similarities and Variations	2021	Tobias R. Ortelt Tobias Haertel Silke Frye	Springer	Conference Paper	Cross Reality and Data Science in Engineering	10.1007/978-3-030-52575-0_11	0	143-153
32	Impact of remote experimentation, interactivity and platform effectiveness on laboratory learning outcomes	2021	Krishnashree Achuthan Dhananjay Raghavan Balakrishnan Shankar Saneesh P. Francis Vysakh Kani Kolil	Springer	Journal	International Journal of Educational Technology in Higher Education	10.1186/s41239-021-00272-z	1	1-24

Tabla 4: Resultados de la Revisión Bibliográfica

1.2. ANTECEDENTES CONCEPTUALES Y REFERENCIALES

1.2.1. Hipótesis de la investigación

Para esta investigación se elaboró la hipótesis de investigación (H1) y la hipótesis nula (H0) que serán desarrolladas luego de la presentación y análisis de los resultados obtenidos en la investigación realizada.

H1: Un sistema para la gestión de laboratorios remotos incrementa el acceso a recursos para la realización de prácticas académicas de la asignatura de Internet de las Cosas.

H0: Un sistema para la gestión de laboratorios remotos no incrementa el acceso a recursos para la realización de prácticas académicas de la asignatura de Internet de las Cosas.

1.2.2. Matriz de Operacionalización de variables

VARIABLE	DEFINICIÓN CONCEPTUAL	MÉTRICAS, DIMENSIONES	INDICADOR
Sistema para la gestión de laboratorios Remotos (independiente)	Los Laboratorios Remotos tienen la finalidad de permitir a estudiantes, el acceso a realizar prácticas académicas en un entorno real, manipulando remotamente los objetos de laboratorio.	Módulo Académico	% usabilidad del módulo Académico
		Módulo Laboratorio	% usabilidad del módulo Laboratorio
Habilidades de aprendizaje (dependiente)	Son aquellas tareas involucradas en el proceso de aprendizaje que permiten a una persona aprender nuevo conocimiento.	Habilidades desarrolladas	Utilidad del sistema como herramienta de aprendizaje
Prácticas académicas (dependiente)	Es una actividad formativa desarrollada por el estudiante, en la que se constituyen como una herramienta en el aprendizaje, ya que brindan la posibilidad de entender cómo se construye el conocimiento dentro de la comunidad científica.	Número de usuarios que acceden a un recurso	Cantidad de usuarios que acceden a un recurso.
		Número de recursos disponibles	Cantidad de recursos a los que puede acceder un usuario.

Tabla 5: Matriz de Operacionalización de variables

1.2.3. Variable independiente

Sistema para la gestión de acceso a Laboratorios Remotos

1.2.4. Variable dependiente

Habilidades de aprendizaje a través de las prácticas académicas.

1.2.5. Fundamentación teórica

Para el presente estudio se consideran distintas herramientas que ayudan con el cumplimiento de los objetivos específicos.

Laboratorios remotos

Un laboratorio remoto permite a los estudiantes realizar experimentos con un dispositivo remoto en tiempo real a través de Internet [58]. Los laboratorios remotos son una alternativa interesante a los laboratorios tradicionales, ya que al tener en cuenta que cualquier simulación solo alcanza un cierto grado de aproximación a la experimentación real [59]. Los laboratorios remotos combinan los beneficios del enfoque de aprendizaje práctico con una comprensión detallada de los conceptos teóricos, sin el apoyo logístico que requiere el enfoque práctico de los laboratorios tradicionales [10]. Sin embargo, la efectividad del laboratorio remoto está íntimamente ligada a la capacidad del usuario para interactuar con el equipo y, por lo tanto, esta puede ser la mayor debilidad en comparación con otras modalidades [58]. La ayuda humana en los experimentos suele ser limitada, lo que lleva al denominado "aprendizaje centrado en el estudiante", en el que el proceso de aprendizaje y sus resultados están en manos de los estudiantes [60], lo que propicia a indicar que, los laboratorios remotos pueden ser tomados como herramientas útiles dentro del B-Learning. La implementación de laboratorios remotos genera expectación, interés y atención en los estudiantes, lo que lo hace adecuado para muchas materias diferentes [61].

En comparación con las prácticas en laboratorios tradicionales, las que se realizan por un laboratorio remoto aportan una serie de ventajas, tales como: pueden ser utilizadas por una gran cantidad de estudiantes distribuidos en múltiples instituciones; una amplia gama de equipos, se puede recopilar y analizar una gran cantidad de datos [62]. Para un correcto funcionamiento se debe considerar los siguientes criterios: la conectividad, la comunicación entre hardware y software, la retroalimentación de la información producida por el circuito solicitado por los estudiantes y las posibles interconexiones modulares desarrolladas [61]. Los estudiantes pueden realizar los experimentos desde

cualquier lugar controlando directamente el equipo en el laboratorio [63]. Estos experimentos no necesariamente requieren que los estudiantes se conviertan en expertos en la operación de equipos específicos, sino que los utilicen como herramientas para recopilar datos o realizar una tarea. El laboratorio remoto permite a los estudiantes solidificar sus conceptos teóricos realizando experimentos con compañeros [64].

Internet de las Cosas (IoT)

El IoT tiene un impacto social, tanto en nuestra forma de trabajar como de vivir. Además, abre una gama completa de nuevas posibilidades, como por ejemplo, nuevos modelos de negocio o ecosistemas [65][66][67]. El IoT es la red que permite a los dispositivos compartir datos entre ellos, ya sea la recabada por sensores o la que es efectuada por actuadores además que permite la manipulación remota de los mismos [68][69].

El desarrollo de IOT es inseparable del desarrollo de la tecnología inteligente, estos protocolos consienten el desarrollo de soluciones de IoT garantizando una baja latencia, bajo consumo de energía y la calidad de servicio [70][71][72]. Los sistemas IOT podrían encargarse de recolectar datos en diferentes entornos, desde ecosistemas naturales hasta edificios, debido a esto, se podrían utilizar para diversos campos.

Arquitecturas IoT

Dentro de una arquitectura IoT, según lo expresa [73], existen tres componentes básicos como lo son el hardware, middleware y la capa de presentación, conocida como la arquitectura de tres capas. Existe también la arquitectura de cinco capas. El papel de las capas de percepción y aplicación en esta arquitectura es el mismo que el de la arquitectura de tres capas [74]. El IOT tiene una amplia gama de objetos, por lo tanto, no existe una arquitectura particular para ello y se basa en diferentes dominios tales como:

- Identificación de frecuencia de radio (RFID) [75],
- Arquitectura orientada a servicios (SOA) [76],
- Red de sensores inalámbricos (WSN) [77] [78][79],
- Arquitectura IoT para Smart Cities [80],
- Arquitectura IoT-Fog [81] [82],
- Arquitectura basada en IOT para el desarrollo sanitario [83],
- Arquitectura de seguridad basada en objetos (OSCAR) [74],
- Arquitectura ciber-física y de seguridad social [84],
- Arquitectura jerárquica de seguridad [85],
- Arquitectura de seguridad de tráfico con reconocimiento de medios [86],
- Arquitectura de seguridad basada en HIMMO [74], y
- Arquitectura de seguridad basada en IoTNetWar [87].

Microservicios

Un microservicio se puede representar como una caja negra, que se divide en capas. Estas capas se organizan en función del tipo de tarea a realizar. La comunicación entre microservicios solo es posible a través de las interfaces expuestas por el propio microservicio [88]. Se desarrollan, implementan y mantienen por separado [89] [90]. Esto permite que los equipos sean autónomos donde pueden decidir qué tecnología utilizar que mejor se adapte a las necesidades actuales de una organización [91]. El lenguaje y la base de datos pueden ser diferentes de un microservicio a otro.

En el contexto de IOT, los microservicios aumentan la escalabilidad y la reutilización en este tipo de sistemas [88]. Los microservicios pueden compartir el núcleo del sistema operativo subyacente, lo que permite implementaciones más rápidas sin disminuir el rendimiento [92]. Al adoptar la “ideología” de DevOps, los equipos de software separados son responsables de diferentes aspectos de la aplicación final, lo que permite que tanto el equipo como el núcleo del software desarrollen, prueben, manejen fallas y escalen de forma independiente [93].

DevOps

Los métodos DevOps han reducido la brecha cultural y metodológica entre desarrolladores y operadores. El paradigma DevOps asigna más atención al relanzamiento continuo, las herramientas unificadas y los procesos organizacionales. Los avances comunes de DevOps incluyen, por ejemplo, canalizaciones de integración continua/entrega continua (CI/CD) y soluciones de orquestación altamente automatizadas para el entorno de tiempo de ejecución [94]. Dentro del ciclo de vida que maneja DevOps, se considera: Planificar, Desarrollar, Verificar, Probar, Desplegar, Funcionar y Monitorear.

La adopción de metodologías DevOps presenta las siguientes características: acelera la innovación, mejora la colaboración, aumentar la eficiencia, reduce fallas, acelera el tiempo de recuperación y mejora la satisfacción laboral [95]. Al adoptar DevOps en un entorno IOT, permite a los desarrolladores monitorear los sistemas que construyen, bajo demanda, en autoservicio y, por lo tanto, permiten una retroalimentación rápida y continua [96]. Cabe destacar que, existen desafíos como la falta de colaboración y comunicación, falta de habilidad y conocimiento, infraestructura complicada, falta de gestión, falta de enfoque DevOps y problemas de confianza [97], resultan complicaciones para la adopción de DevOps en una organización y/o equipo de desarrollo de software.

1.3. ANTECEDENTES CONTEXTUALES

1.3.1. Problema

Con el COVID-19 propagándose rápidamente en todo el mundo, muchos países han iniciado varias estrategias para prevenir la propagación de la infección, incluido el cierre temporal de las instituciones educativas [98]. Las clases virtuales sustituyeron a las presenciales, por lo tanto, las plataformas de enseñanza y las herramientas educativas lograron adaptarse para mantener la continuidad en el proceso educativo [99]. Ninguna institución educativa estuvo preparada para las clases virtuales, desde las escuelas, colegios, incluso las IES, incluyendo la ausencia de herramientas y recursos para el desarrollo de prácticas experimentales académicas.

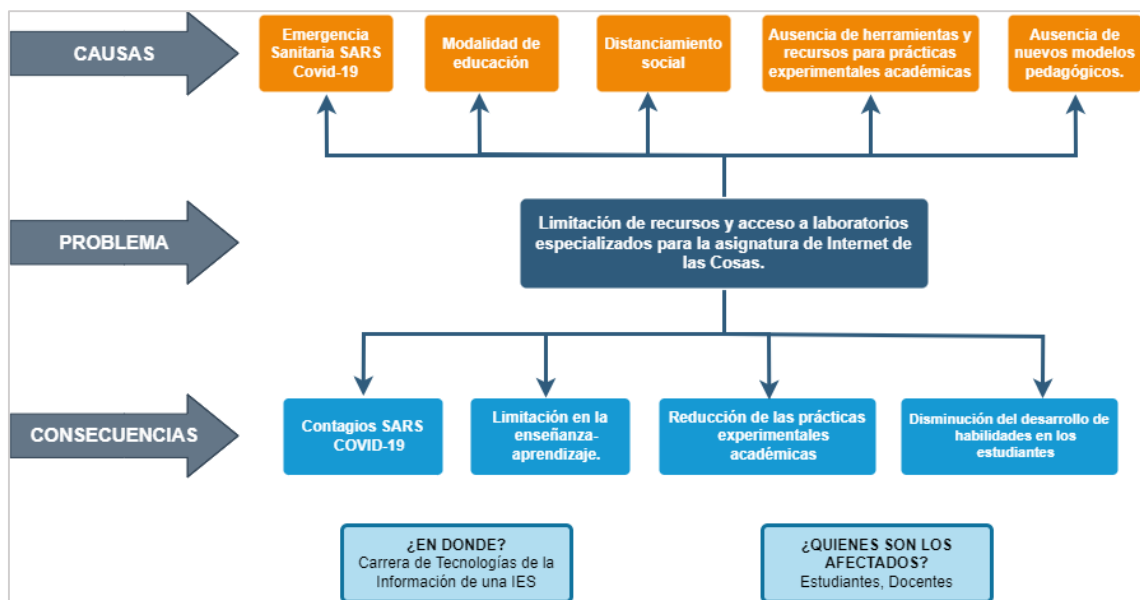


Ilustración 3: Causas y consecuencias del problema de investigación

Tal como se observa en la ilustración 3, existe una limitación de recursos y acceso a laboratorios especializados para la asignatura “Internet de las Cosas” de la carrera de Ingeniería en Tecnologías de la Información de una IES en la ciudad de Machala, lo que provoca limitaciones en la enseñanza-aprendizaje de parte de los docentes. La falta de recursos para el desarrollo de prácticas experimentales induce a que exista una reducción en el desarrollo de las habilidades de aprendizaje en los estudiantes de la asignatura.

1.3.2. Implementación

Se implementa una arquitectura IoT basada en microservicios que permita la conexión con los artefactos electrónicos desarrollados por los estudiantes de la asignatura, mismos que servirán como escenarios de ejecución de los laboratorios remotos de la asignatura.

El desarrollo del sistema de gestión de acceso a los laboratorios implica la implementación de una arquitectura IoT basada en microservicios, que permita la conexión con maquetas de artefactos electrónicos las cuales sirven como escenarios de las prácticas académicas. Dentro de la arquitectura se implementará un API-Gateway, microservicios Python y NodeJS conectados a bases de datos Postgres y MongoDB; un servidor de telemetría MQTT y un servidor de gestión de video.

1.3.3. Escenario de implementación

La implementación de esta investigación fue dentro de la infraestructura de Red de una IES en la ciudad de Machala en un entorno de pruebas con la participación de estudiantes de la asignatura “Internet de las Cosas” de la carrera de Tecnologías de la Comunicación.

1.3.4. Objetivo General

Implementar un sistema para la gestión de laboratorios remotos en la Carrera de Tecnologías de la Información de una IES, utilizando una arquitectura IOT basada en microservicios con un enfoque de metodológico DevOps para la realización de prácticas académicas de la asignatura Internet de las Cosas.

1.3.5. Objetivos Específicos

- Investigar las técnicas empleadas para el desarrollo de un sistema de gestión de laboratorios remotos mediante una Revisión Sistemática de la Literatura utilizando la guía metodológica de [13].
- Implementar una arquitectura IoT basada en microservicios para la comunicación con los dispositivos electrónicos usados para las prácticas académicas de la asignatura de Internet de las Cosas.
- Implementar un sistema para la gestión de laboratorios remotos en un entorno IoT mediante el uso de metodologías ágiles para incrementar el acceso a recursos que permitan la realización de prácticas académicas en la asignatura de “Internet de las Cosas” de una IES.
- Evaluar la gestión del acceso de los usuarios a los escenarios de prácticas para determinar los niveles de utilización de los laboratorios remotos.

CAPÍTULO 2. MATERIALES Y MÉTODOS

En este capítulo, se describe la metodología y todos los materiales utilizados para el desarrollo de este estudio. A continuación, se detallan el tipo de estudio y enfoque de investigación empleados para la consecución de los objetivos planteados.



Ilustración 4: Materiales y Métodos de la investigación

2.1. TIPO DE ESTUDIO O INVESTIGACIÓN

El presente estudio pretende establecer relaciones de causa y efecto que permiten el aumento de la comprensión sobre un tema específico [100]. El enfoque explicativo, utiliza como fuente de información la literatura de artículos publicados que se eligen de manera cuidadosa para tener una equilibrada comprensión del tema. Este enfoque se aplica para explicar las diversas implementaciones de laboratorios remotos en diferentes campos del conocimiento.

El enfoque descriptivo tiene como objetivo la descripción de la naturaleza de un segmento demográfico, sin concentrarse en las razones por las que se produce un determinado fenómeno [101]. Es por ello que se emplea este enfoque para especificar el motivo por el que un sistema para la gestión de laboratorios remotos incrementa el acceso a recursos para la realización de prácticas académicas.

2.2. PARADIGMA O ENFOQUE

Los paradigmas utilizados para esta investigación son las siguientes:

- **Cuantitativo:** Este tipo de paradigma es utilizado en este proyecto para describir la recolección de datos y explicar los resultados mediante el uso de herramientas de análisis de datos, matemático y estadístico.
- **Cuasi-Experimental:** el estudio se basa en la implementación de un sistema de gestión de acceso a laboratorios remotos por lo que se enfoca en identificar la forma en la que se relaciona la variable independiente sobre la dependiente y qué es lo que produce.

2.3. POBLACIÓN Y MUESTRA

Como este trabajo se enfoca en la asignatura de “Internet de las Cosas”, se considera a los estudiantes de la carrera de Tecnologías de la Información de una IES en la ciudad de Machala como la población que será afectada por la implementación de un sistema de gestión de laboratorios remotos. Por tal motivo, la muestra se focaliza en los estudiantes de la asignatura “Internet de las Cosas”, mismos que van a manipular el prototipo implementado.

El tipo de muestreo será probabilístico sistemático debido a que la muestra se obtendrá de una población finita, para eso se utilizó la fórmula de tamaño de muestra para poblaciones finitas, donde se estableció una población de estudiantes de la carrera de Tecnologías de la Información con un total de 396 estudiantes, un nivel de confianza del 99% con su respectivo valor z-score de 2.580, un margen de error del 15% y probabilidad de éxito y fracaso del 50%, los resultados de la aplicación de la fórmula se muestran a continuación:

n= muestra

N= población= 396

p= probabilidad a favor = 50%

q= probabilidad en contra = 50%

z= nivel de confianza = 2.580 (95%)

e= error de muestra = 15%

$$n = \frac{(N * z^2 * p * q)}{e^2 * (N - 1) + z^2 * p * q}$$

Una vez aplicada la fórmula, se obtuvo una muestra de 39 estudiantes, mismos que deberán utilizar la aplicación para la comprobación de la hipótesis.

2.4. MÉTODOS TEÓRICOS CON LOS MATERIALES UTILIZADOS

Los métodos teóricos para la recolección de datos tomados en cuenta para el desarrollo de este trabajo son los siguientes:

- Encuesta: es una técnica de recolección de datos que se refiere al uso de un cuestionario de preguntas diseñado previamente, sin alterar el entorno ni el fenómeno donde se recoge la información, mismo que servirá para medir los niveles de inmersión, utilidad, satisfacción y usabilidad del sistema de gestión de laboratorios remotos.
- Ejecución de artefactos con sensores: Debido a la naturaleza del proyecto, se utilizan dispositivos electrónicos para el funcionamiento del proyecto propuesto, por lo tanto, con la ejecución de los artefactos con sensores se pretende medir el rendimiento de la arquitectura IoT conjunto el prototipo de software desarrollado.

2.5. MÉTODOS EMPÍRICOS CON LOS MATERIALES UTILIZADOS

Los métodos empíricos utilizados para el desarrollo del proyecto propuesto son los siguientes:

- Revisión Sistemática de Literatura: para dar un mayor sustento de investigación en el desarrollo de este proyecto, en el Capítulo 1 se incluyó la guía metodológica para la

RSL de Bárbara Kitchenham, que otorga prioridad a las fuentes primarias y secundarias con validez científica desde las siguientes fuentes de datos como:

- ✓ IEEE Xplorer
 - ✓ Springer
 - ✓ Web of Science
 - ✓ Science Direct
 - ✓ MDPI
-
- Productos: se desarrollará tres productos entregables. El primero es el diseño e implementación de la arquitectura IoT basada en microservicios con contenedores Docker. El segundo producto serán los códigos fuentes de los microservicios desarrollados en Python y NodeJS. Finalmente, el tercer producto será la aplicación web desarrollada para testear las funcionalidades.

 - Desarrollo de un Prototipo de software: será el entregable, mismo que implica en la implementación de una arquitectura IoT, para posteriormente desarrollar y ejecutar el sistema de gestión de laboratorios remotos, dentro del cual se utilizaron las siguientes herramientas y tecnologías para su construcción:
 - ✓ Comunicación: API-REST, MQTT, WebSocket
 - ✓ Lenguajes de programación: Python – Django, Nodejs
 - ✓ Bases de Datos: Postgres, MongoDB, MySQL
 - ✓ Metodología: DevOps
 - ✓ Arquitectura: IoT 3 capas, Microservicios

2.6. TÉCNICAS ESTADÍSTICAS PARA EL PROCESAMIENTO DE DATOS

La técnica estadística para el procesamiento de los datos obtenidos es la siguiente:

- Estadística descriptiva: al utilizar esta técnica se pretende describir las frecuencias de los datos arrojados por la encuesta realizada a la muestra de estudiantes, a su vez, representar cuantitativamente los resultados de las métricas utilizadas para medir el rendimiento de la arquitectura IoT planteada, mediante medidas de tendencia central y medidas de dispersión.

CAPÍTULO 3. DESARROLLO DEL PROTOTIPO

En el presente capítulo se describen de manera detallada los productos generados en el proyecto investigativo. Este proyecto comprende de dos partes, el diseño de la arquitectura IoT y el desarrollo del prototipo del sistema de gestión de laboratorios remotos con la utilización de la metodología DevOps.

3.1. IMPLEMENTACIÓN DE ARQUITECTURA IOT

Uno de los objetivos para lograr la implementación de un sistema de gestión de acceso a laboratorios remotos es la implementación de una arquitectura IoT que permita la comunicación entre la aplicación con los dispositivos electrónicos usados para las prácticas académicas de la asignatura de Internet de las Cosas.

3.1.1. Arquitectura IoT

Se propone una arquitectura IoT de tres capas, debido a que describe la lógica de la implementación, en la que los datos obtenidos por los componentes reales que se transmiten hacia otros dispositivos o servidores y son procesados en las aplicaciones para un uso determinado.

Como se observa en la ilustración 5 se dispone de las capas Cloud, Cliente y laboratorios, mismas que tienen relación con las capas de aplicación, red y percepción de la arquitectura de tres capas. En la capa Cliente estarán ubicados todos los usuarios que utilicen la aplicación web desde computadoras portátiles o de escritorio, hasta los teléfonos móviles desde su navegador web. Dentro de la capa Cloud, se encuentran las herramientas que permiten la comunicación entre los dispositivos electrónicos. En esta capa se alojan los microservicios desarrollados conectados a un API-Gateway que permite la comunicación de la web-app con los microservicios, a su vez, esta capa contiene un microservicio que permite conectar el adaptador de protocolos con la web-app, este puente lo que permite es manipular en tiempo real los laboratorios, enviando señales de acción y dando respuesta al usuario que maneja la aplicación web. Todo esto es implementado bajo una infraestructura de máquinas virtuales en una red local.

Por último, la capa de laboratorios funge como la capa de percepción, puesto que en esta capa se localizan los dispositivos electrónicos con los cuales el usuario interactuará de manera remota.

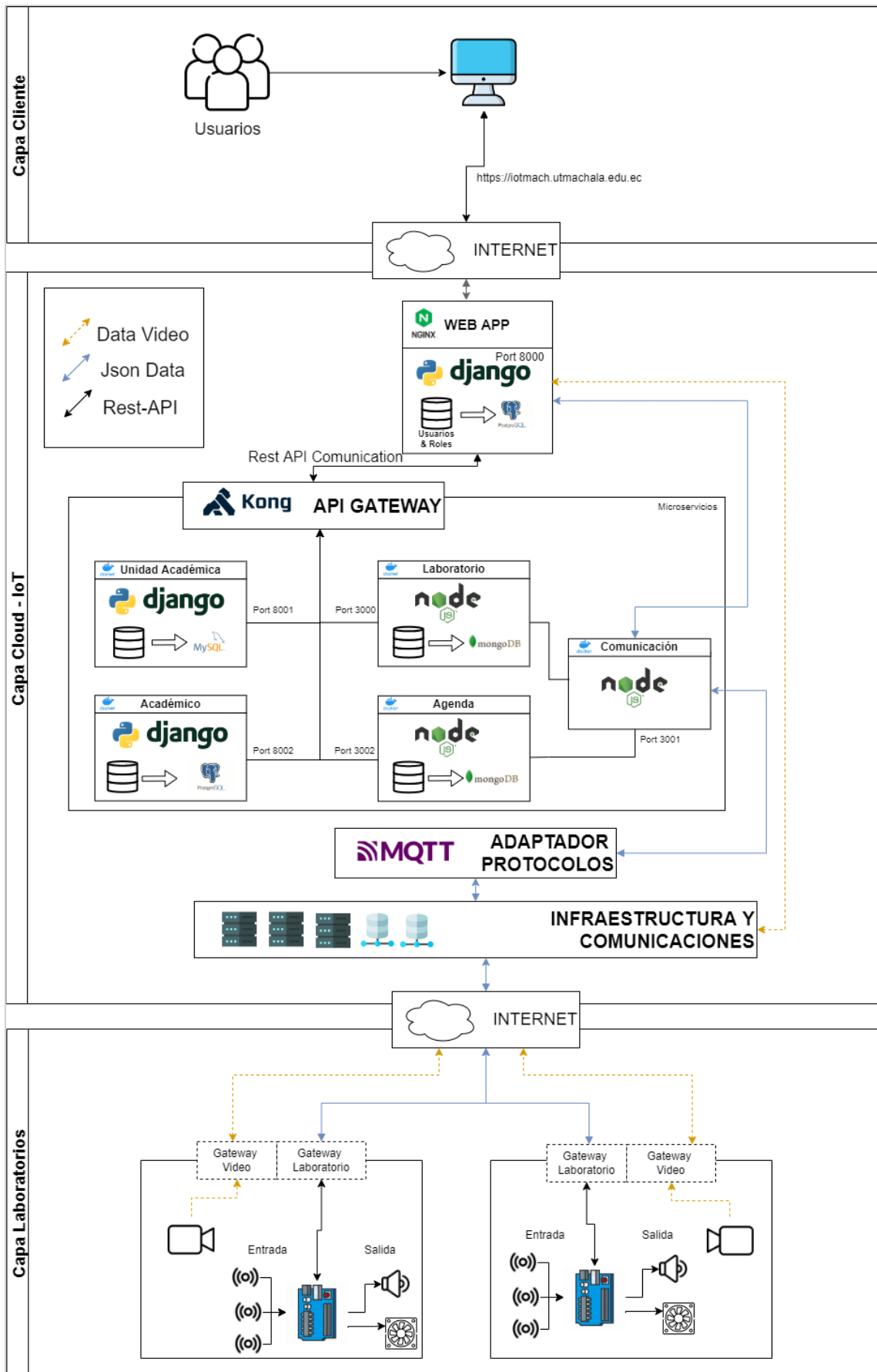


Ilustración 5: Arquitectura IoT basada en microservicios

3.1.2. Distribución de recursos y estructura de RED

La arquitectura IoT fue implementada dentro de la infraestructura de red de una IES ubicada en la ciudad de Machala. Para la implementación de esta arquitectura se utilizó una computadora personal. Esta computadora fungió como anfitrión de diversas máquinas virtuales, en las que se distribuyeron los diferentes servicios necesarios para el despliegue de la arquitectura IoT. Como se observa en la ilustración 6, se repartieron los servicios en cuatro máquinas virtuales. En la máquina 1 se alojan los servicios para el despliegue de la aplicación web y el API-GATEWAY, mientras que en la máquina 2 se encuentran los microservicios que utilizan una base de datos para almacenar información.

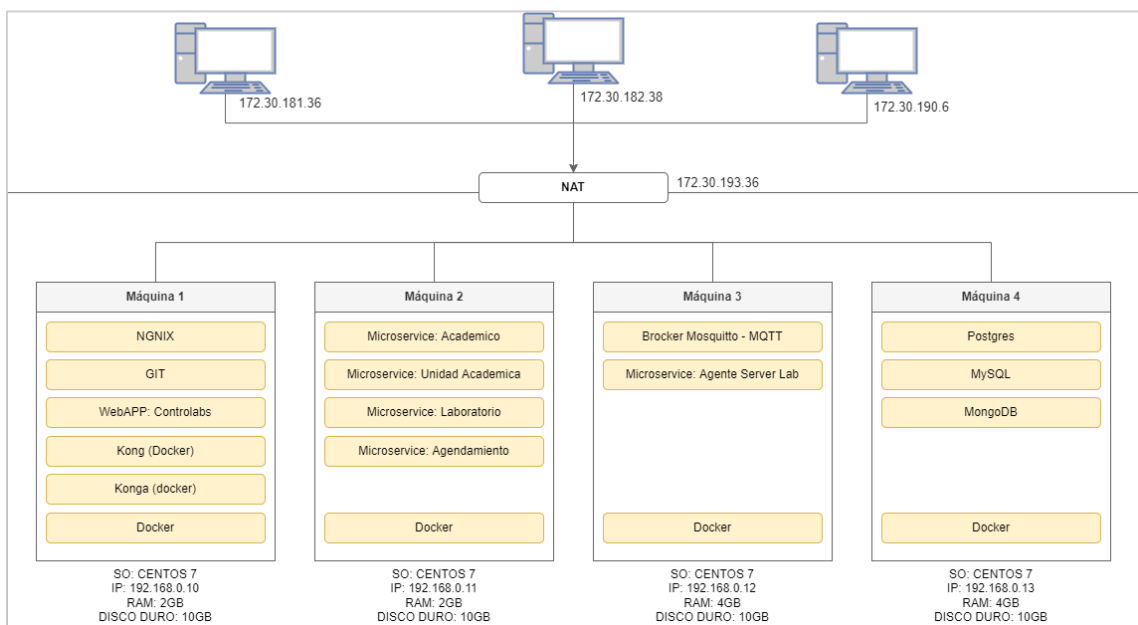


Ilustración 6: Distribución de recursos y arquitectura de red

En la máquina 3 corre el microservicio encargado de comunicar la aplicación web con el adaptador de protocolos, el bróker MOSQUITTO que maneja el protocolo MQTT, mismos que se encuentran alojados en esta máquina virtual; en cambio en la máquina 4 se encuentran las bases de datos. Todos estos servicios son desplegados mediante contenedores Docker y ejecutándose sobre sistemas operativos Centos 7.

3.1.3. Enrutamiento

En vista de que la implementación de la arquitectura será realizada en una red privada de máquinas virtuales, es necesario realizar un enrutamiento a los servicios que se verán reflejados en la infraestructura de red de la IES.

Servicio	Aplicación	Puerto salida	IP Máq. Virtuales	Puerto privado
Web	ControlLabs	80	192.168.0.10	8080
API-REST	API Gateway	8081	192.168.0.10	8081
Websocket	Comunicación	3000	192.168.0.12	3003
MQTT	Servidor Mosquitto	1883	192.168.0.12	1883

Tabla 6: Características del Equipo Anfitrión

En la tabla 6, se detallan los servicios que tendrán salida en la infraestructura de red. Estos servicios son los que se necesitan para que el sistema funcione correctamente, el servicio de MQTT es requerido para que los dispositivos se comuniquen con la aplicación web. Tanto el API-REST como los WebSockets sirven para que la aplicación web se comunique con los microservicios y los dispositivos electrónicos.

3.2. DESARROLLO DE PROTOTIPO DE SOFTWARE

El sistema de gestión de laboratorios remotos está basado en microservicios que utilizan protocolos de comunicación el API-REST y WebSockets. Para el desarrollo del prototipo, se empleó el ciclo de vida de DevOps como metodología.

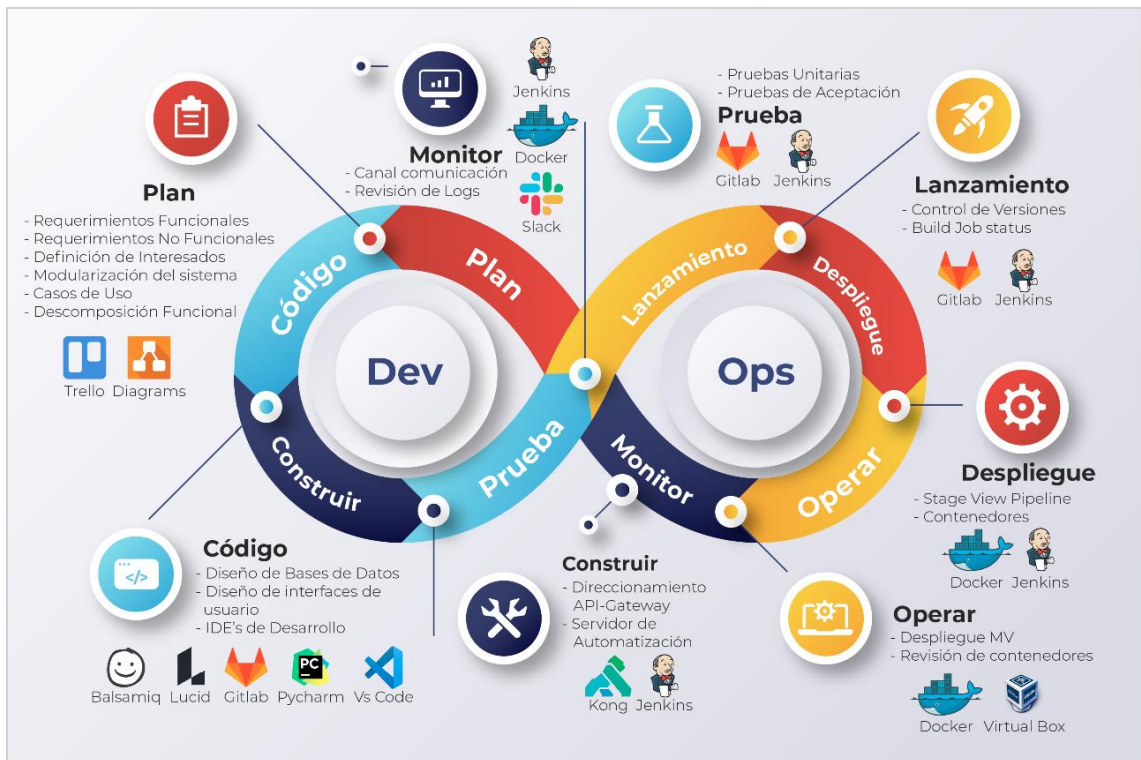


Ilustración 7: Herramientas utilizadas en el ciclo de vida DevOps

Como se puede observar en la ilustración 7, se muestra un desglose de las actividades realizadas en las distintas fases propuestas por DevOps para la realización del sistema. Estas fases están descritas en los siguientes puntos de este capítulo.

3.2.1. Fase de planificación

En esta etapa se realiza diseño del software, se definieron los requerimientos del sistema, se identifican a los interesados del proyecto, se especifica el comportamiento de los usuarios y la descomposición de las funcionalidades del sistema.

Identificación de módulos del sistema

El sistema propuesto está basado en microservicios los mismos que manipularán la información con el fin de gestionar los laboratorios remotos. Los módulos se detallan en la tabla 7.

Módulo	Detalle
Administración	Permite el acceso y direccionamiento a los usuarios mediante la gestión de: Usuarios, Roles, Permisos y Menú
Unidad Académica	Permite el registro de la información de las asignaturas, gestionando lo siguiente: Facultad, Carrera, Malla, Nivel y Asignatura
Académico	Se registra la información de los cursos en los que los estudiantes tienen acceso a los laboratorios mediante: Periodo, Curso, Recurso y Matriculado
Laboratorio	Se registra y configura los dispositivos electrónicos que servirán como herramientas para prácticas académicas, mediante: Laboratorio, Configuración y Escenario
Reservaciones	Permite realizar las reservaciones de los laboratorios, mediante: Reservación y Participantes
Comunicación	Permite la comunicación entre la aplicación web y los dispositivos electrónicos, a través del protocolo MQTT y WebSockets.

Tabla 7: Identificación de módulos del sistema

Interesados del proyecto

Se definen quienes serán los interesados del proyecto en la tabla 8, mismos que gestionarán el proceso de implementación del sistema.

Nombre	Rol	Profesión	Responsabilidades
Dixys Leonardo Hernández Rojas	Gerente del proyecto	Ingeniero Eléctrico	Revisión de documentación, entregables y aprobación de pruebas.
Kevin Adrián Valarezo Paz	Desarrollador	Ingeniero de Sistemas	Desarrollo e Implementación del sistema, conjunto a las pruebas de la aplicación.

Tabla 8: Interesados del proyecto

Usuarios del sistema

Se definen los tipos de usuario que tendrá el sistema:

Administrador: tiene acceso a la configuración y registros de la mayoría de entidades del sistema:

Usuario	Administrador
Actividades	Gestiona a los usuarios Gestiona las facultades, mallas, carreras, niveles y asignaturas Gestiona los cursos Configura los cursos Gestiona los laboratorios Configura a los laboratorios
Contenido de interacción	Módulo Administración Módulo Unidad Académica Módulo Cursos Módulo Laboratorios

Tabla 9: Funciones del Administrador

Docente: tiene como función configurar los cursos y laboratorios, además de gestionar las reservaciones de los laboratorios:

Usuario	Docente
Actividades	Configura los cursos Acceso a los cursos Configura los laboratorios Acceso a los laboratorios Gestionar Reservaciones
Contenido de interacción	Módulo Cursos Módulo Laboratorios Módulo Reservaciones

Tabla 10: Funciones del Docente

Estudiante: tiene acceso a los cursos y laboratorios, realiza las reservas de laboratorio:

Usuario	Estudiante
Actividades	Acceso a los cursos Acceso a los laboratorios Reservar laboratorios Manipulación de los componentes de comunicación.
Contenido de interacción	Módulo Cursos Módulo Laboratorios Módulo Reservaciones Módulo Comunicación

Tabla 11: Funciones del Estudiante

Especificación de Requisitos Funcionales

Los requisitos funcionales del sistema propuesto especifican las funciones que dispone, mismas que son especificadas a través de las historias de usuario, detalladas en el listado presentado en la tabla 12.

ID Historia	Enunciado de la Historia	Alias	Módulo	Prioridad
RF-ADM-001	Como un Administrador, necesito crear roles y permisos, con la finalidad de restringir funcionalidades de los usuarios.	Creación de roles y permisos	Administración	ALTA
RF-ADM-002	Como un Administrador, necesito registrar a los usuarios que utilizarán el sistema asignándoles el rol que le corresponde.	Creación de usuarios	Administración	ALTA
RF-ADM-003	Como un Administrador, necesito dar de baja a los usuarios, con la finalidad de que no tengan acceso al sistema.	Dar de baja a los usuarios	Administración	MEDIO
RF-ADM-004	Como un Administrador, necesito registrar a los menús que permitan a la navegabilidad hacia nuevas funcionalidades.	Creación de menús	Administración	MEDIO
RF-ADM-005	Como cualquier usuario, necesito actualizar mi contraseña de acceso al sistema	Actualización de contraseña	Administración	MEDIO
RF-ADM-006	Como un Estudiante, Docente o Administrador, necesito actualizar mis datos del perfil.	Actualización del perfil de usuario	Administración	BAJO
RF-ADM-007	Como un Administrador, necesito registrar de manera masiva a los usuarios con rol estudiante al sistema.	Carga masiva de los usuarios.	Administración	BAJO
RF-UA-008	Como un Administrador, necesito registrar a las facultades.	Creación de facultades	Unidad Académica	MEDIO
RF-UA-009	Como un Administrador, necesito registrar a las carreras pertenecientes a una facultad.	Creación de carreras	Unidad Académica	MEDIO
RF-UA-010	Como un Administrador, necesito registrar a las mallas académicas pertenecientes a una carrera.	Creación de mallas	Unidad Académica	MEDIO
RF-UA-011	Como un Administrador, necesito registrar a los niveles pertenecientes a las mallas.	Creación de niveles	Unidad Académica	MEDIO
RF-UA-012	Como un Administrador, necesito registrar a las asignaturas dentro de un nivel.	Creación de asignaturas	Unidad Académica	MEDIO
RF-UA-013	Como un Administrador, necesito realizar la búsqueda de facultades.	Búsqueda de facultades	Unidad Académica	ALTO
RF-UA-014	Como un Administrador, necesito realizar la búsqueda de carreras filtrando por facultad.	Búsqueda de carreras	Unidad Académica	ALTO
RF-UA-015	Como un Administrador, necesito realizar la búsqueda de asignaturas filtrando por carrera.	Búsqueda de asignaturas	Unidad Académica	ALTO
RF-ACA-016	Como un Administrador, necesito registrar un período académico para restringir los acceso a los cursos.	Creación de períodos académicos	Académico	ALTO

ID Historia	Enunciado de la Historia	Alias	Módulo	Prioridad
RF-ACA-017	Como un Administrador, necesito registrar los cursos que permitan el acceso al laboratorio.	Creación de cursos	Académico	ALTO
RF-ACA-018	Como un Administrador o Docente, necesito configurar los cursos para asignar los laboratorios y/o dar de baja a los participantes.	Configuración de los cursos	Académico	ALTO
RF-ACA-019	Como un Estudiante o Docente, necesito acceder a los cursos mediante una contraseña.	Acceso a cursos	Académico	ALTO
RF-ACA-020	Como un Estudiante, necesito darme de baja de un curso.	Darse de baja en los cursos	Académico	MEDIO
RF-LAB-021	Como un Administrador, necesito crear los laboratorios que serán utilizados en los cursos	Creación de laboratorios	Laboratorio	ALTO
RF-LAB-022	Como un Administrador o Docente, necesito configurar los laboratorios para ajustar los parámetros de configuración y el escenario donde será ejecutado.	Configuración de los laboratorios	Laboratorio	ALTO
RF-LAB-023	Como un Administrador o Docente, necesito realizar una búsqueda de los laboratorios para ser asignados a los cursos donde sean utilizados.	Búsqueda de laboratorios	Laboratorio	MEDIO
RF-LAB-024	Como un Estudiante o Docente, necesito acceder a una sala de espera para observar el estado del laboratorio antes de su uso.	Acceso a sala de espera de laboratorio	Laboratorio	ALTO
RF-LAB-025	Como un Estudiante o Docente, necesito acceder a un escenario de prácticas académicas.	Acceso a los laboratorios	Laboratorio	ALTO
RF-RES-026	Como un Estudiante, necesito realizar una reservación de un laboratorio en una fecha y hora determinada.	Reservación de laboratorios	Reservación	MEDIO
RF-RES-027	Como un Estudiante o Docente, necesito realizar una cancelación a una reservación de laboratorio realizada con anterioridad antes de que se cumpla la fecha establecida.	Cancelación de reserva de laboratorio	Reservación	BAJO
RF-COM-028	Como un Estudiante o Docente que ha ingresado a un laboratorio, necesito realizar una acción dentro de un dispositivo electrónico.	Enviar datos a un dispositivo electrónico.	Comunicación	ALTO
RF-COM-029	Como un Estudiante o Docente que ha ingresado a un laboratorio, necesito recibir información emitida por un dispositivo electrónico.	Recibir datos emitidos por un dispositivo electrónico.	Comunicación	ALTO
RF-LAB-030	Como un Estudiante o Docente, necesito salir del escenario de prácticas, cerrando las actividades realizadas.	Salir de escenario de laboratorio	Comunicación	MEDIO

Tabla 12: Historias de usuario – Requisitos funcionales

Especificación de Requisitos No Funcionales

Estos requisitos especifican los criterios que se van a usar en la operación del sistema:

CÓDIGO	DESCRIPCIÓN
RNF-1	Disponibilidad: El sistema debe operar a través de la infraestructura de red de la IES en Machala.
RNF-2	Interoperabilidad: El aplicativo debe comunicarse con diversos microservicios indistintamente en el lenguaje que se encuentren desarrollados.
RNF-3	Integridad: El sistema debe almacenar los datos en las bases de datos establecidas por los microservicios.
RNF-4	Eficiencia: El sistema debe soportar el acceso concurrente de usuarios conforme a su rol previamente asignado.
RNF-5	Usabilidad: Los formularios de ingreso de información y las interfaces de manipulación de objetos serán de fácil entendimiento, con la finalidad de realizar un trabajo rápido y coherente.

Tabla 13: Requisitos No Funcionales

Diagramas de Caso de Uso

Los diagramas de caso de uso permiten la representación de manera gráfica a los requisitos funcionales del sistema, en los que se detalla el comportamiento e interacción de los actores con los objetos de la aplicación. A continuación, se mostrarán los escenarios de los módulos en los que los actores intervienen.

Administración: en este módulo se registrarán los usuarios con sus roles, manejando los permisos junto a los menús para la navegabilidad del sistema.

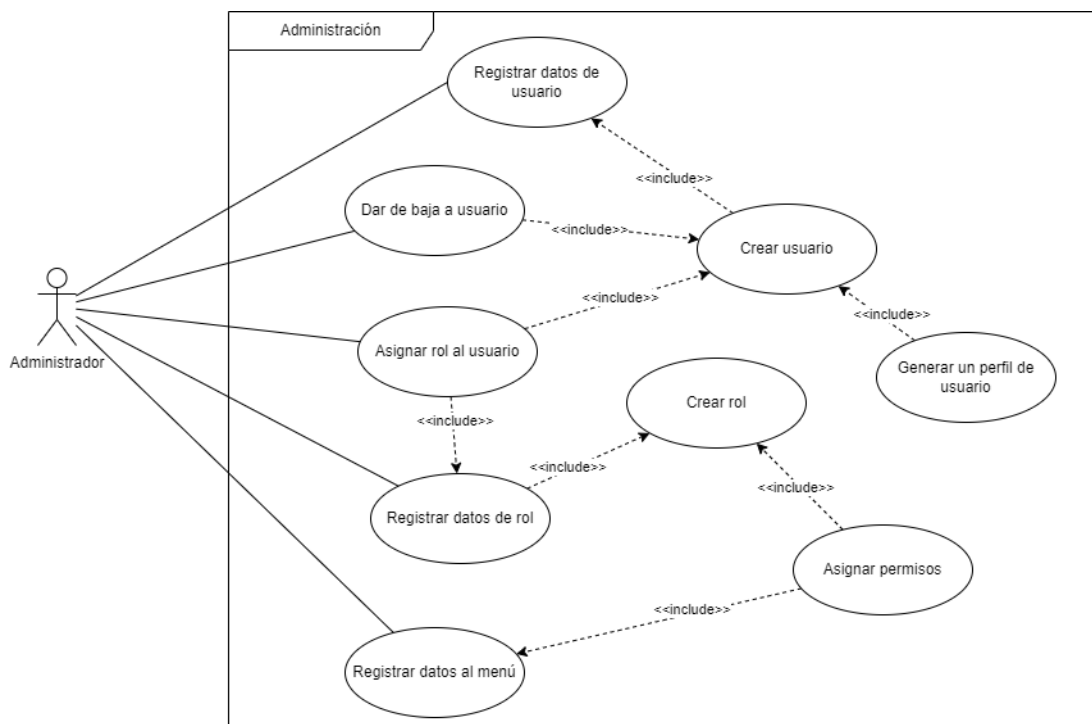


Ilustración 8: Diagrama de Casos de uso de Administración

En la ilustración 8, se observa las interacciones que tiene el Administrador al gestionar los accesos que tendrán los usuarios, la creación de roles y permisos, y gestionar los menús del sistema.

Unidad Académica: en este microservicio se guardarán los registros de las asignaturas, carreras y facultades.

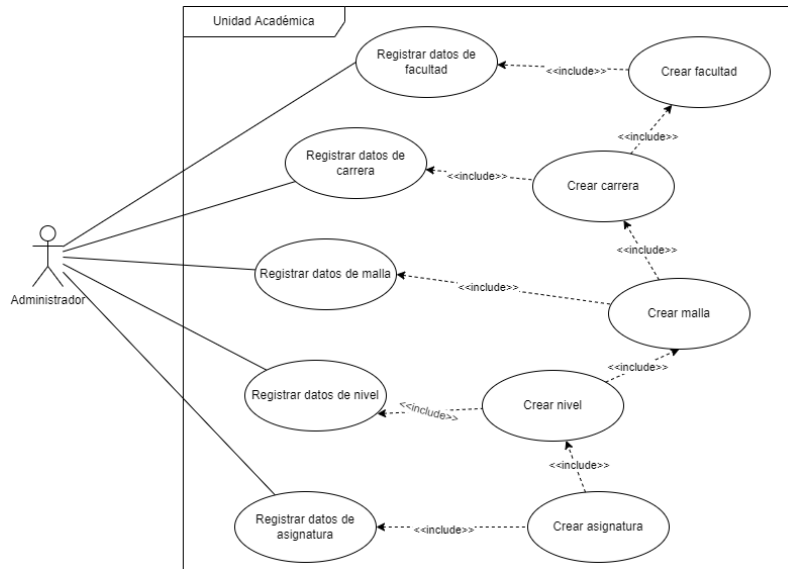


Ilustración 9: Diagrama de Casos de uso de Unidad Académica

En la ilustración 9, se observa que el Administrador es el encargado de crear las facultades, carreras, mallas, niveles y asignaturas.

Académico: en este microservicio se manejan los accesos a los recursos a través de los cursos en lo que solo tendrán acceso los que tengan la credencial de acceso.

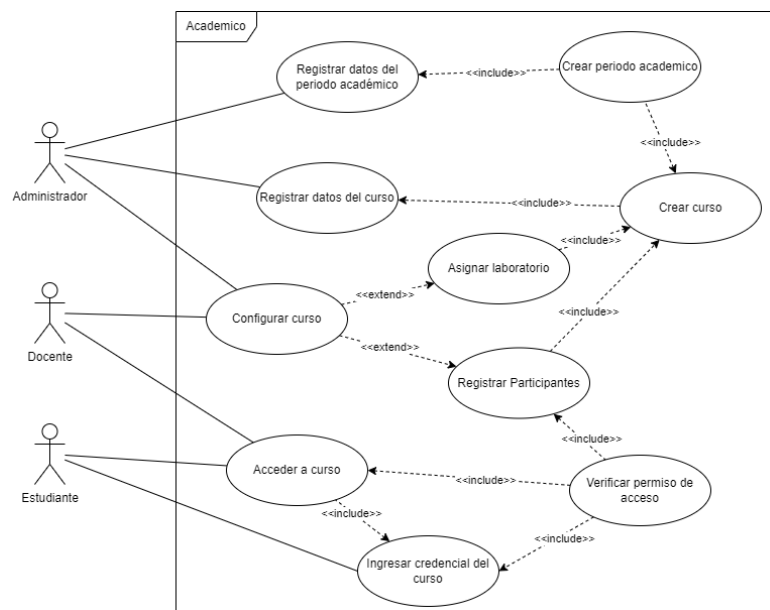


Ilustración 10: Diagrama de Casos de uso de Académico

Como se visualiza en la ilustración 10, están presentes tres tipos de actores: el Administrador, Docente y Estudiante. El Administrador está encargado de la creación del periodo académico que se rigen los cursos, y la creación de estos últimos. La configuración de los cursos también puede ser realizada por un Docente, aquí se asignan los laboratorios y se administra quien tiene acceso al curso. El Estudiante para acceder a un curso debe ingresar la credencial que le permita registrarse y tener camino libre a los recursos asignados al curso.

Laboratorio: en este microservicio se manejan los registros y el control de accesos a los dispositivos electrónicos.

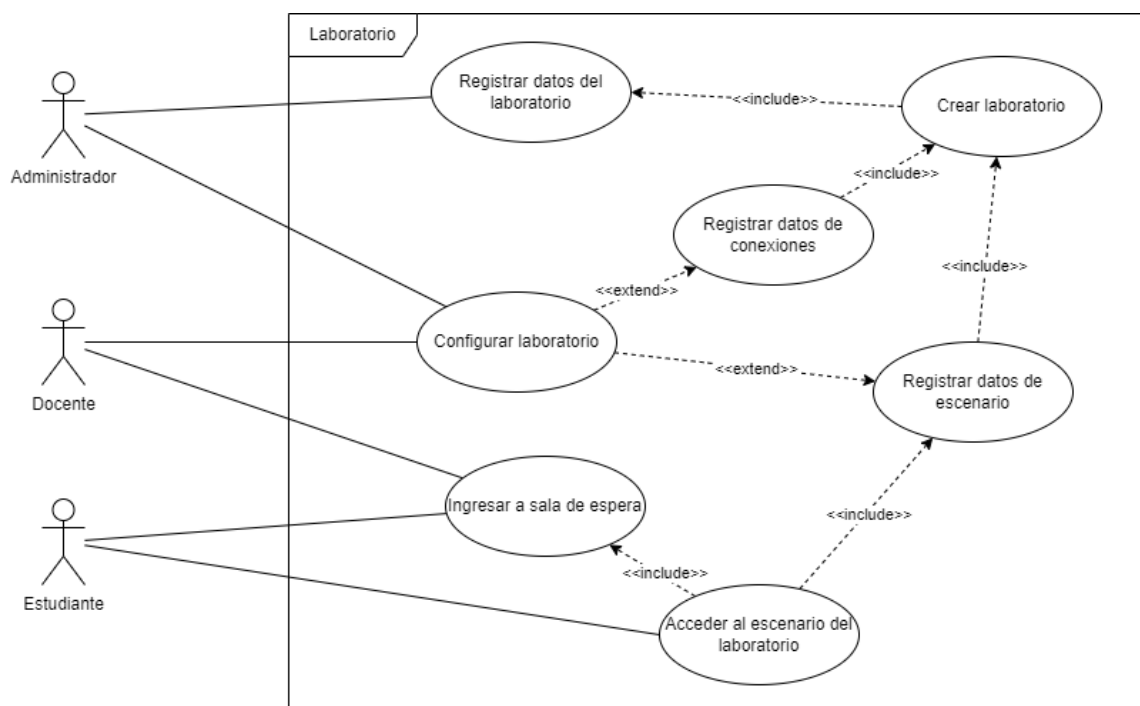


Ilustración 11: Diagrama de Casos de uso de Laboratorio

En la ilustración 11, están presentes tres tipos de usuarios del sistema. El Administrador está encargado de la creación de los laboratorios y su configuración, misma que también puede ser realizada por un Docente, en esta se ajustan las credenciales de acceso para la comunicación con los dispositivos electrónicos y se registra el tipo de escenario con el que trabajará. El Estudiante para acceder a un laboratorio debe ingresar previamente a una sala de espera.

Comunicación: este microservicio sirve para el envío de datos hacia los dispositivos electrónicos a través de un dashboard, y estos dispositivos responderán con información.

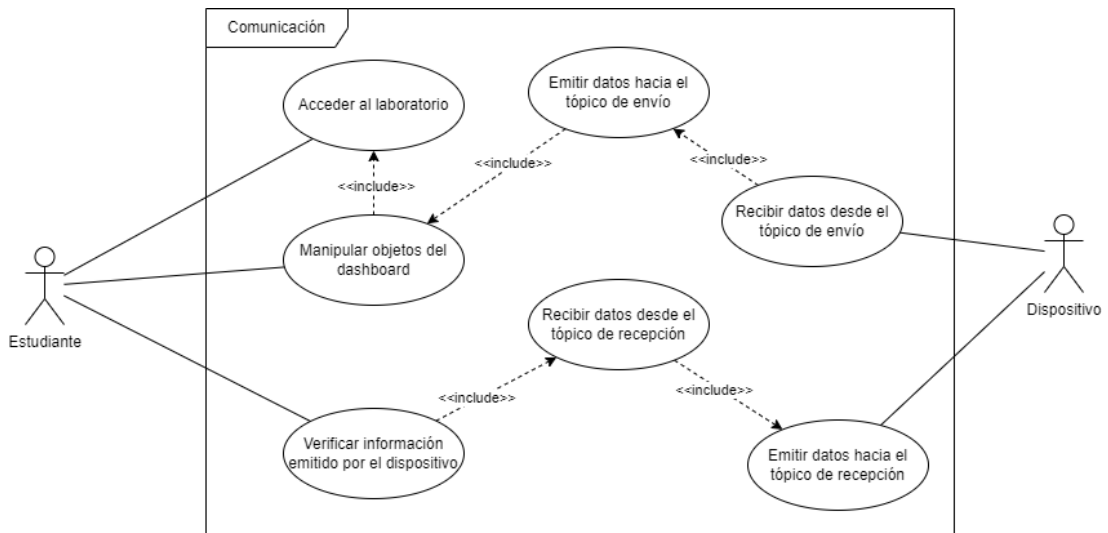


Ilustración 12: Diagrama de Casos de uso de Laboratorio

En la ilustración 12 se observa como el Estudiante accede al laboratorio y al manipular los objetos del dashboard emite información dirigida hacia un dispositivo electrónico. Este dispositivo recibe la información realiza su función y emite datos hacia el servidor, mismos que son receptados por el dashboard, lo que permite la visualización de la respuesta al estudiante.

Descomposición Funcional

La descomposición funcional al ser un procedimiento de análisis que descompone un proceso complejo, se utiliza para examinar los elementos individuales de un sistema, mediante la cadena de valor se representa las actividades del sistema, en el que se descompone en sus partes constitutivas.

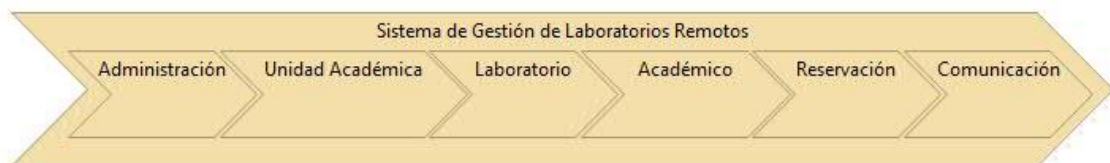


Ilustración 13: Cadena de valor del proyecto

Como se puede observar en la ilustración13, el sistema planteado se descompone en sus distintos módulos de manera secuencial, es decir, para su correcto funcionamiento deberían ejecutarse primero el módulo de Administración, seguido del microservicio de Unidad Académica, posterior a ello, se deben tener listos los Laboratorios para que los microservicios Académico, Reservación y Comunicación se ejecuten de la manera correcta.

Flujo de valor y capacidades

Los flujos de valor junto a sus capacidades, permite descomponer a mayor detalle la cadena de valor mostrada en la ilustración 13. Esta descomposición permite visualizar las distintas funcionalidades de los microservicios a través de las capacidades del flujo de valor analizado.

En la ilustración 14, se observa el flujo de valor junto a las capacidades del módulo de Administración, estas capacidades comprenden la gestión de Usuarios, Roles y Menú. En la gestión de usuarios, se controla los usuarios a crearse y quienes pueden acceder al sistema, mediante los roles, que también se gestionan de manera independiente. La gestión de Menú permite al usuario la navegabilidad dentro del sistema por lo que es necesario tener los permisos y roles ya establecidos antes de crear los menús del sistema.

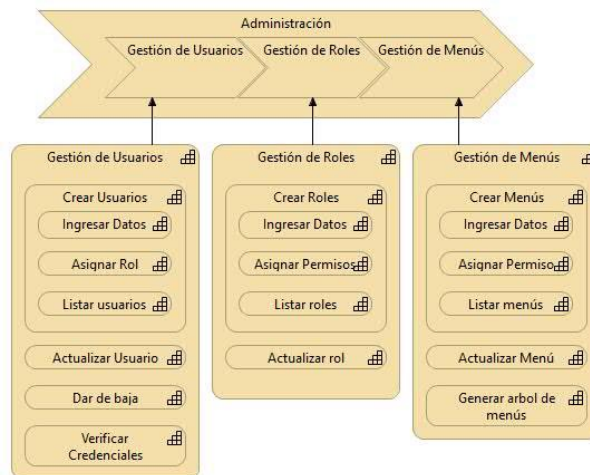


Ilustración 14: Flujo de valor y capacidades del Módulo Administración

En la ilustración 15, en el microservicio de Unidad Académica, se puede observar cómo trabajan de manera secuencial las funciones desde la gestión de Facultades hasta la gestión de Asignaturas.

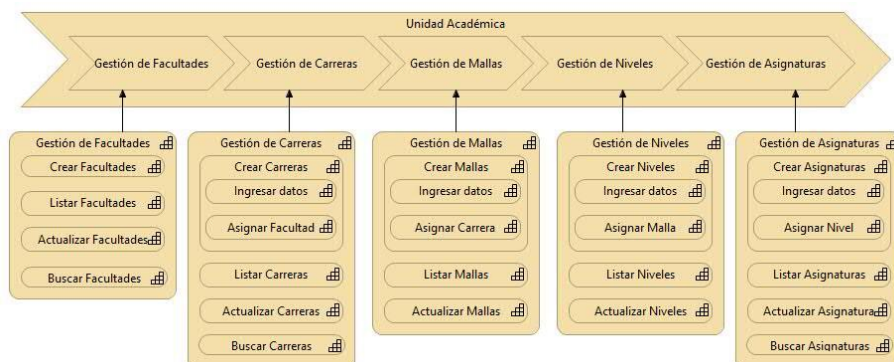


Ilustración 15: Flujo de valor y capacidades de Unidad Académica

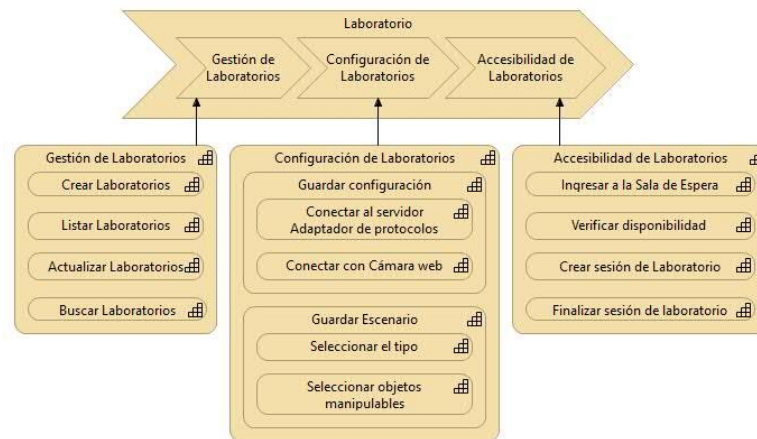


Ilustración 16: Flujo de valor y capacidades del microservicio Laboratorio

El microservicio Laboratorio se descompone con la gestión de laboratorios en donde se crean los registros de los laboratorios; la configuración de laboratorios implica realizar la conexión con el servidor adaptador de protocolos y la configuración del tipo de escenario; y el control a la accesibilidad de los laboratorios en la que se verifica la disponibilidad y los permisos para conceder el acceso.

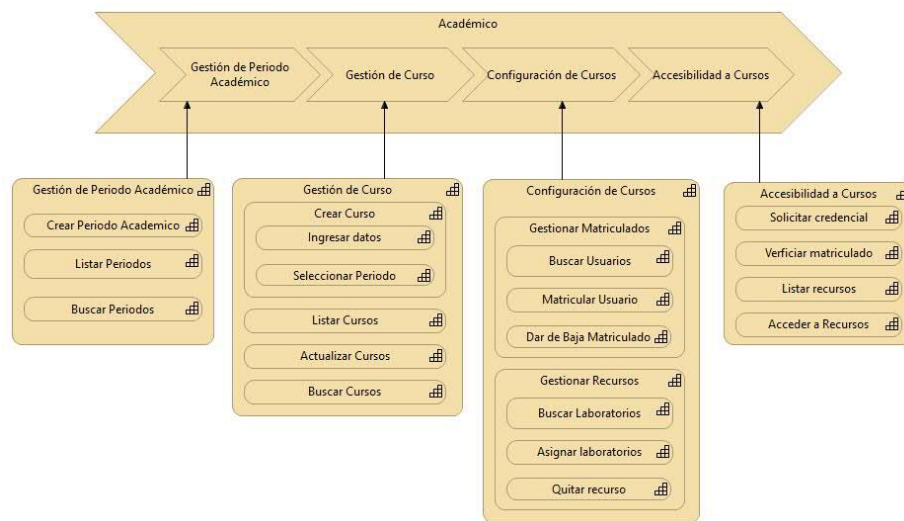


Ilustración 17: Flujo de valor y capacidades del microservicio Laboratorio

El microservicio Académico comprende la gestión de Período Académico, Curso, Configuración de Curso y Accesibilidad. El periodo académico permite controlar el tiempo de disponibilidad de los cursos. En la Gestión de Cursos se registran las asignaturas a las cuales van a hacer uso de los Laboratorios, mismos que son asignados como recursos en la Configuración de Cursos, así como se visualiza en la ilustración 17. En la accesibilidad a los Cursos se controla quienes son los usuarios que tienen acceso al Curso mediante la solicitud de una credencial, matriculando a los usuarios que la posean.

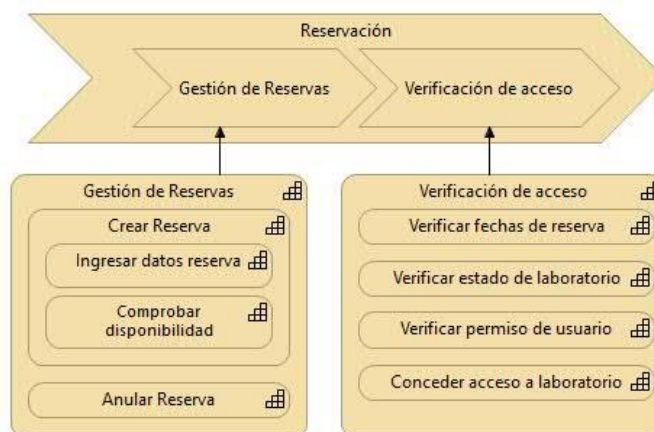


Ilustración 18: Flujo de valor y capacidades del microservicio Reservación

La Reservación permite la separación de uso de un laboratorio en una fecha estimada, como se observa en la ilustración 18, este microservicio comprende la gestión de Reservas y a su vez realizar la verificación de acceso de las reservas de los laboratorios.

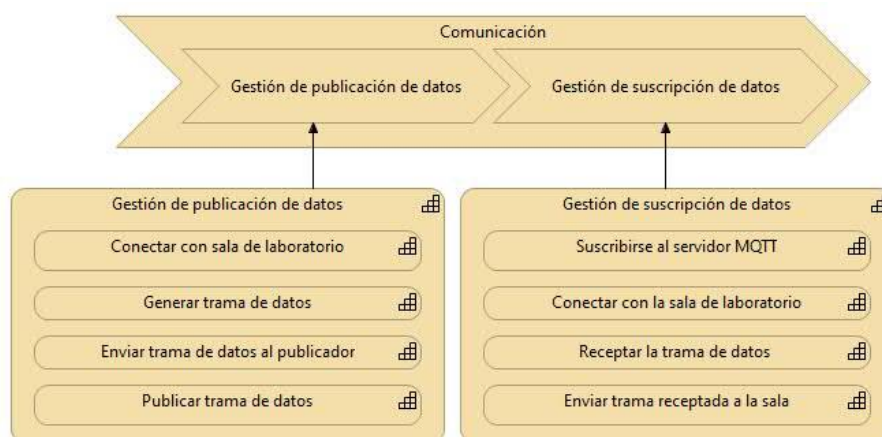


Ilustración 19: Flujo de valor y capacidades del microservicio Comunicación

El microservicio de Comunicación parte de las sesiones de laboratorios, encargado de realizar la publicación y suscripción de datos generados por la manipulación de los laboratorios, tal y como se lo indica en la ilustración 19. Este microservicio, controla el flujo de las tramas de datos generados tanto por la interfaz de usuario como por los dispositivos electrónicos.

3.2.2. Fase de codificación

En la presente fase se procedió a realizar la codificación de los procesos descritos en el análisis de los requerimientos, el diseño del software y la codificación, se establece las herramientas y componentes que permitan el proceso de codificación, tanto como el sistema de gestión de versiones como los IDE's de desarrollo empleado.

Diseño de las bases de datos

En este proyecto cada microservicio, a excepción de Comunicación, cuenta con una base de datos, ya sea relacional o no relacional, como se muestra a continuación.

Administración: utiliza una base de datos relacional debido al sistema de autenticación de usuarios que utiliza Django, que maneja cuentas de usuario, grupos, permisos y sesiones de usuario basadas en cookies [102].

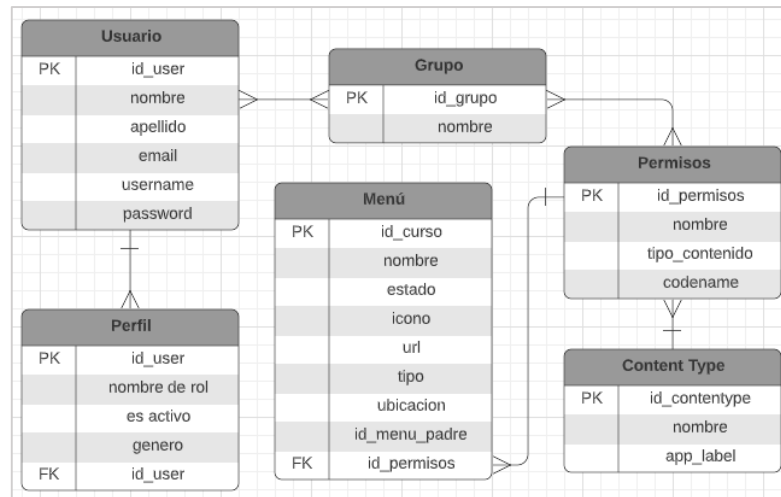


Ilustración 20: Diseño de Base de Datos Relacional de Administración

En la ilustración 20, se muestran las tablas que Django proporciona para la adaptación de su sistema de autenticación, en las cuales se adapta las tablas de Perfil y Menú. Con estas tablas se puede brindar los accesos y navegabilidad a los usuarios dependiendo de los roles y permisos a los que pertenecen.

Unidad Académica: este microservicio utiliza una base de datos relacional debido a las relaciones que existen entre las entidades, desde la facultad hasta llegar a la asignatura.

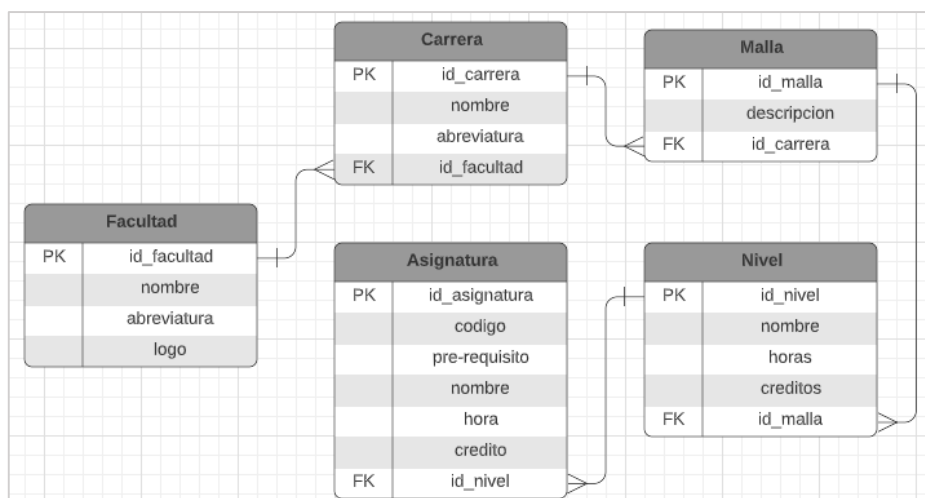


Ilustración 21: Diseño de Base de Datos Relacional de Unidad Académica

Académico: en este microservicio se registran los cursos, en los cuales dependerán del periodo académico para que los matriculados puedan acceder a los recursos que maneja cada curso. Los recursos son los laboratorios que serán manejados por cada curso.

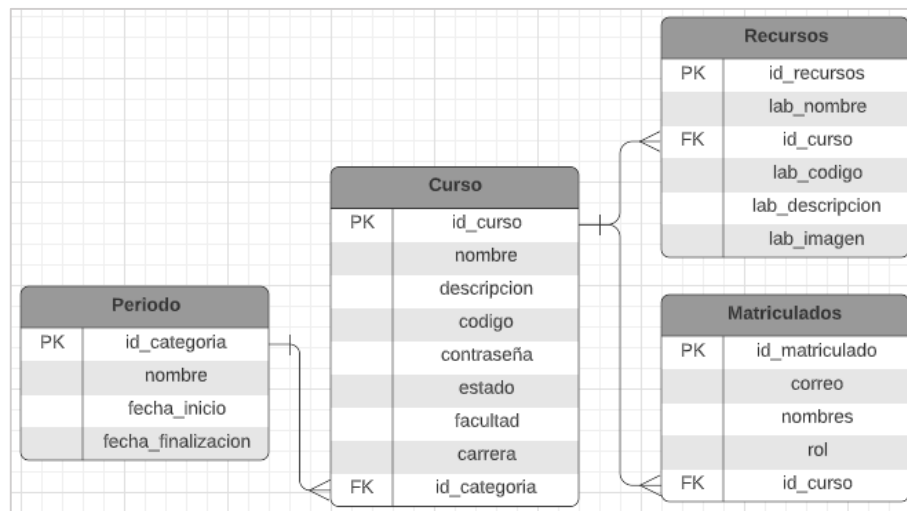


Ilustración 22: Diseño de Base de Datos Relacional de Académico

Laboratorio: este microservicio se maneja con una base de datos no relacional basada en documentos, mismos que tendrán el registro de los laboratorios sumado a la información proveniente del dispositivo electrónico que funge como escenario del laboratorio para las prácticas académicas.

```

1  {
2  {
3    "nombre": "nombre laboratorio",
4    "informacion_general": {
5      "cod_laboratorio": "código identificador del laboratorio",
6      "descripcion": "descripción del laboratorio remoto",
7      "tutorial": "pasos a seguir para desarrollar la práctica en archivo pdf",
8      "imagen_circuito": "una imagen del circuito",
9      "disponible": true
10   },
11   "escenario": {
12     "tipo": "dashboard/compilador",
13     "objetos": [
14       {
15         "widget": "switch/slide/knob/stat/onoff",
16         "interfaz": "identificador de la interfaz",
17         "nombre": "un nombre de objeto",
18         "topico": "publicacion/recepcion",
19         "minimo": "un valor minimo",
20         "maximo": "un valor maximo"
21       }
22     ],
23     "compilacion": {
24       "artefacto": "arduino/esp32/raspberry",
25       "lenguaje": "python,c,c++,javascript"
26     }
27   },
28   "conexion": {
29     "topico": {
30       "publicacion": "un tópico publicador",
31       "recepcion": "un tópico suscriptor"
32     },
33     "camara": "una dirección de acceso a una cámara IP",
34     "host": "una dirección para bróker MQTT",
35     "port": "puerto para bróker MQTT",
36     "user": "usuario de conexión a MQTT",
37     "password": "contraseña de conexión a MQTT",
38     "idDispositivo": "identificador de dispositivo electrónico (laboratorio)"
39   }
40 }
  
```

Ilustración 23: Modelo de documento No Relacional de Laboratorio

En la ilustración 23, refleja el cómo está distribuida el formato de almacenamiento de la información, los parámetros de conexión y la configuración del escenario de laboratorio. Esta estructura de datos permitirá identificar a los dispositivos electrónicos a su vez, se vincula con los cursos que los utilizarán para las asignaturas en los que los estudiantes se han matriculado.

Reservación: con una base de datos no relacional basada en documentos, se registra las reservaciones de los laboratorios, lo que permite verificar su estado. En la ilustración 24, se muestra la estructura de datos que tendrá este microservicio.

```

1  {
2  "nombre_laboratorio": "nombre del laboratorio extraído del módulo laboratorio remoto",
3  "cod_laboratorio": "código identificador del laboratorio extraído del módulo laboratorio remoto",
4  "hora_inicio": "hora inicio de agendamiento",
5  "hora_final": "hora finalización de agendamiento",
6  "fecha": "fecha de agendamiento",
7  "personas": [
8  {
9  "first_name": "nombre del usuario",
10 "last_name": "apellido del usuario",
11 "username": "nombre identificador de usuario",
12 "email": "correo electrónico del usuario"
13 }
14 ],
15 "nombre_reservacion": "nombre del evento de reservación",
16 "estado": "RESERVADO(POR DEFECTO), EN EJECUCION, FINALIZADO"
17 }

```

Ilustración 24: Modelo de documento No Relacional de Reservación

Diseño de interfaces de usuario

Dentro del proceso de codificación de software, el diseño de las interfaces de usuario es necesario para esquematizar la manera como se interactuará con el sistema, para ello un wireframe es una representación visual de la estructura y funcionalidad de un sitio web o una aplicación móvil [103]. Por tal motivo, se utilizó la herramienta Balsamiq Mockups para la realización del maquetado de las interfaces de usuario del sistema.

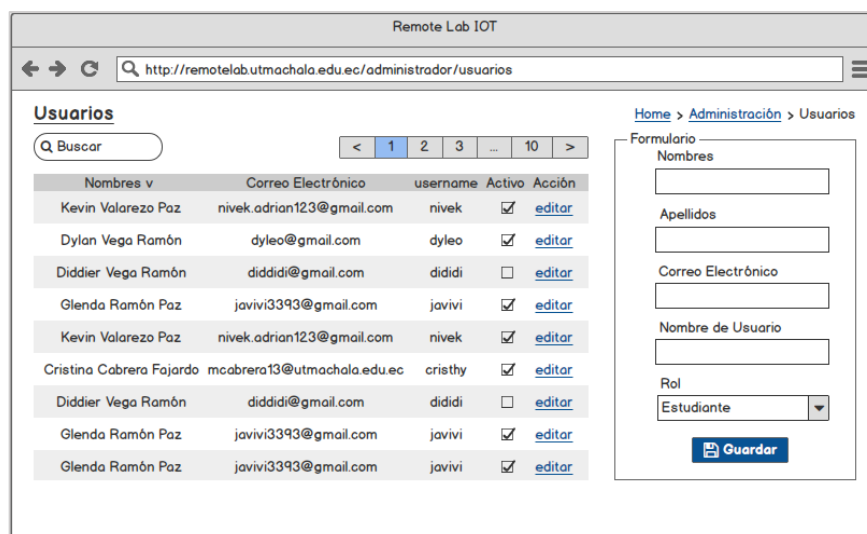


Ilustración 25: Prototipo de pantalla gestión de usuarios

Para los formularios donde se generan las funciones de operación de la información se diseñó un prototipo de interfaz con el estilo que se observa en la ilustración 25, en la que se muestra un listado de los datos en la parte izquierda conjunto a un formulario de datos en la parte de ella, cuenta con paginación y un cuadro de búsqueda. Este prototipo se aplica para la gestión de Usuarios, Facultades, Carreras, Mallas, Asignaturas, Cursos y Laboratorios.

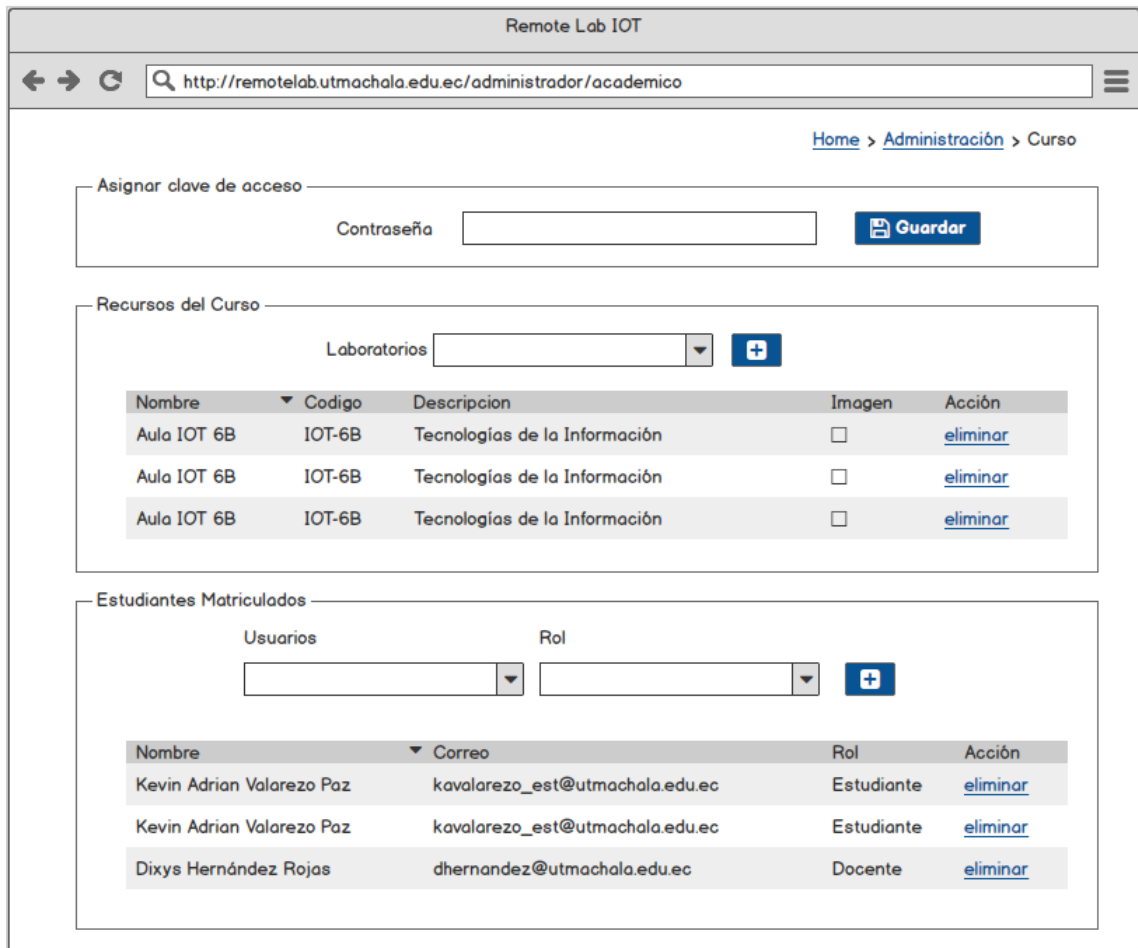


Ilustración 26: Prototipo de pantalla de configuración de curso

Cada curso debe configurarse, en la configuración debe agregarse una contraseña, gestionar los estudiantes matriculados que tendrán acceso en el curso y gestionar los laboratorios que se van a utilizar, para ello se ingresa al formulario de configuración de cursos que se observa en la ilustración 26.

Así mismo cada laboratorio debe configurarse, lo que comprende de registrar los parámetros de conexión hacia el bróker MQTT y la conexión a la cámara, además de seleccionar el tipo de escenario a utilizar, tal como se representa a la izquierda de la ilustración 27, en donde se observa el formulario de configuración de un escenario

compilador, mientras que en la derecha de la misma ilustración se visualiza la configuración de un escenario dashboard.

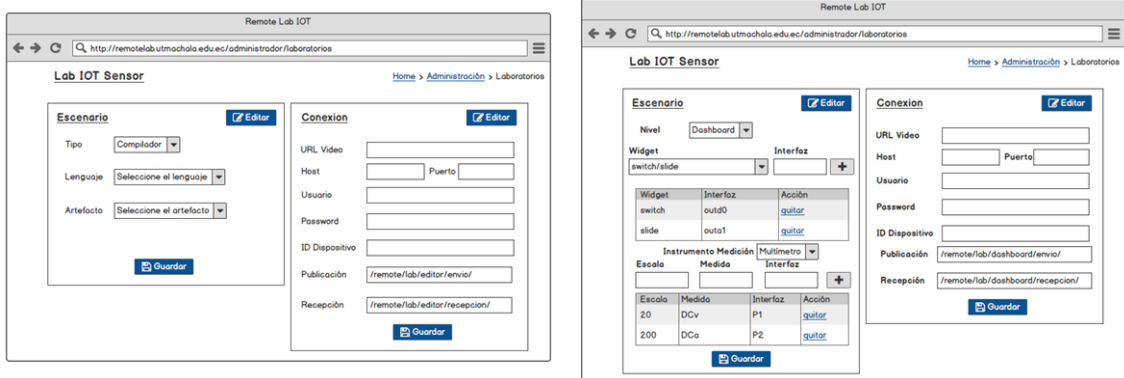


Ilustración 27: Prototipo de pantallas de configuración de laboratorio

Un usuario estudiante y docente, puede visualizar los cursos disponibles que se encuentran en un periodo académico activo, así como se visualiza en la ilustración 28, al ingresar a un curso el sistema pedirá una contraseña para que el estudiante se matricule en el curso seleccionado.

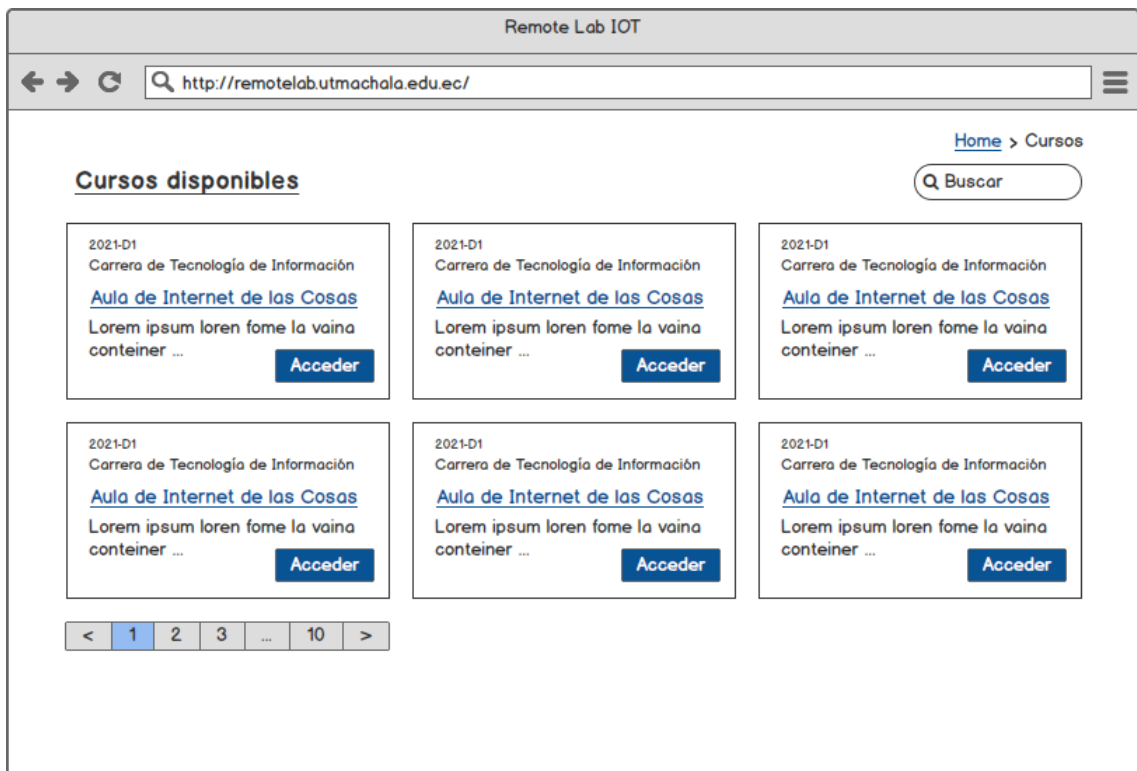


Ilustración 28: Prototipo de pantalla de listado de cursos disponibles

Una vez que se haya ingresado la contraseña correcta, se visualizarán los laboratorios remotos que fueron asignados a ese curso (ilustración 29). Cada ocasión que se acceda a un laboratorio habrá una sala de espera, en la que se podrá agendar el laboratorio (ilustración 30) o esperar que el laboratorio esté disponible para ser utilizado.

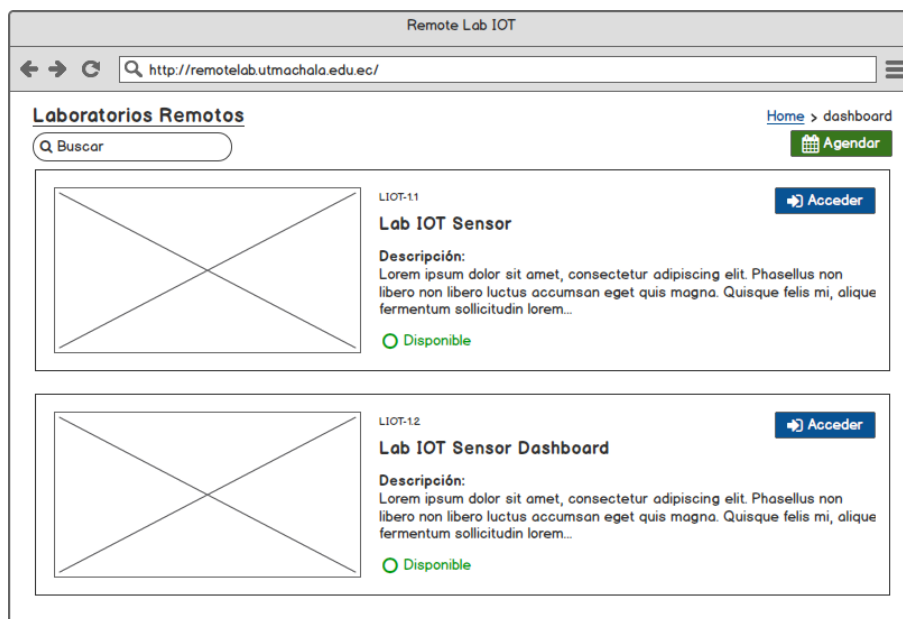


Ilustración 29: Prototipo de pantalla de acceso a Curso

Para el agendamiento del laboratorio se diseñó un prototipo en el que permita ingresar la fecha y hora de inicio y finalización, seleccionar el laboratorio del cual se desea ocupar para realizar una práctica. Una vez registrado la reserva, se podrá visualizar dicha reservación en la agenda mensual, esto representa que a cierta fecha y hora un laboratorio fue reservado.

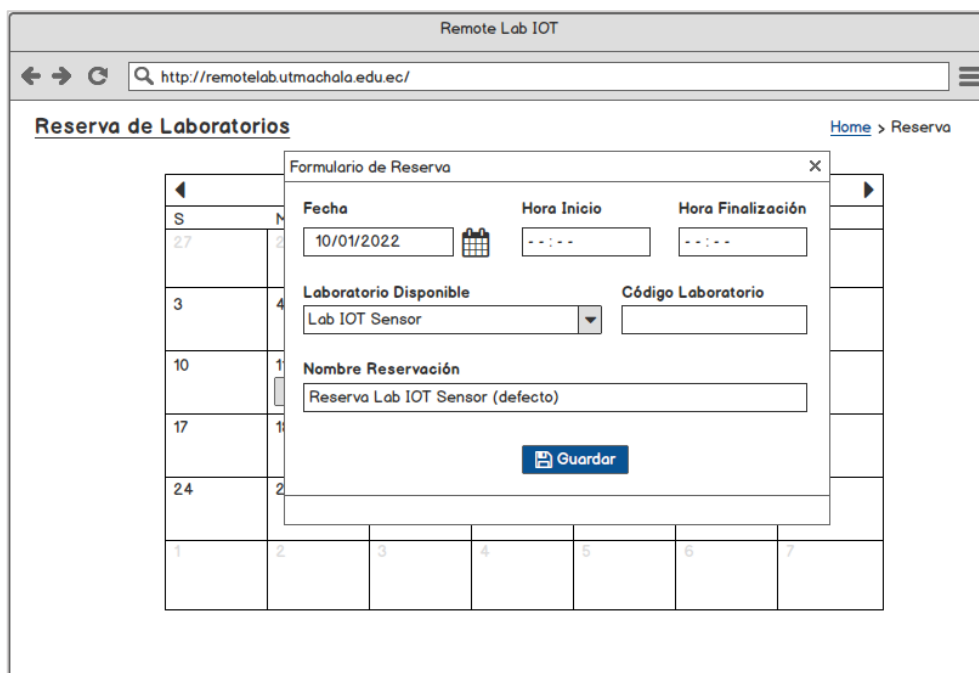


Ilustración 30: Prototipo de pantalla de reserva de laboratorio

Existen dos tipos de escenarios para los laboratorios los que utilizan editor de código y los que utilizan un dashboard, como se puede ver en la ilustración 31 donde se observa los tipos de escenarios. En ambos escenarios se cuenta con una cámara para visualizar el comportamiento del hardware según se manipule las opciones.

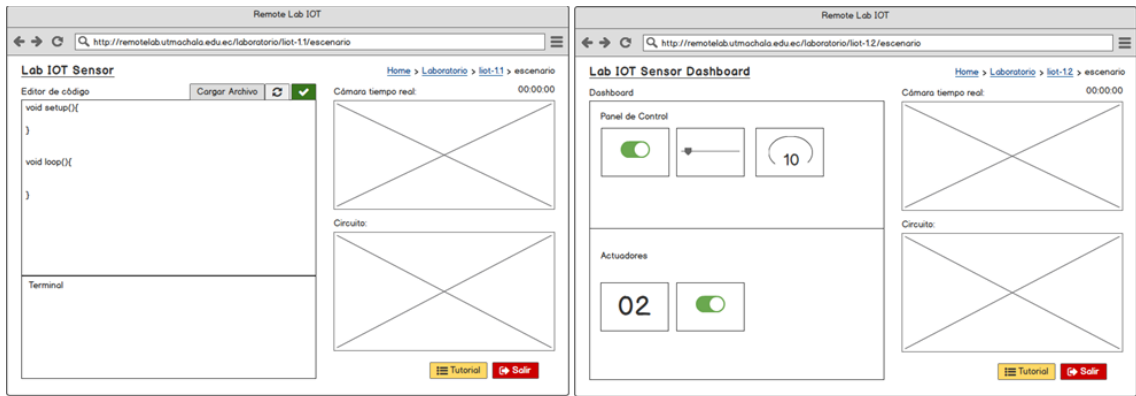


Ilustración 31: Prototipo de pantallas de escenario

IDE de Desarrollo

Un entorno de desarrollo integrado facilita al programador la escritura de su código, la depuración o incluso la compilación de sus aplicaciones en un lenguaje de máquina.

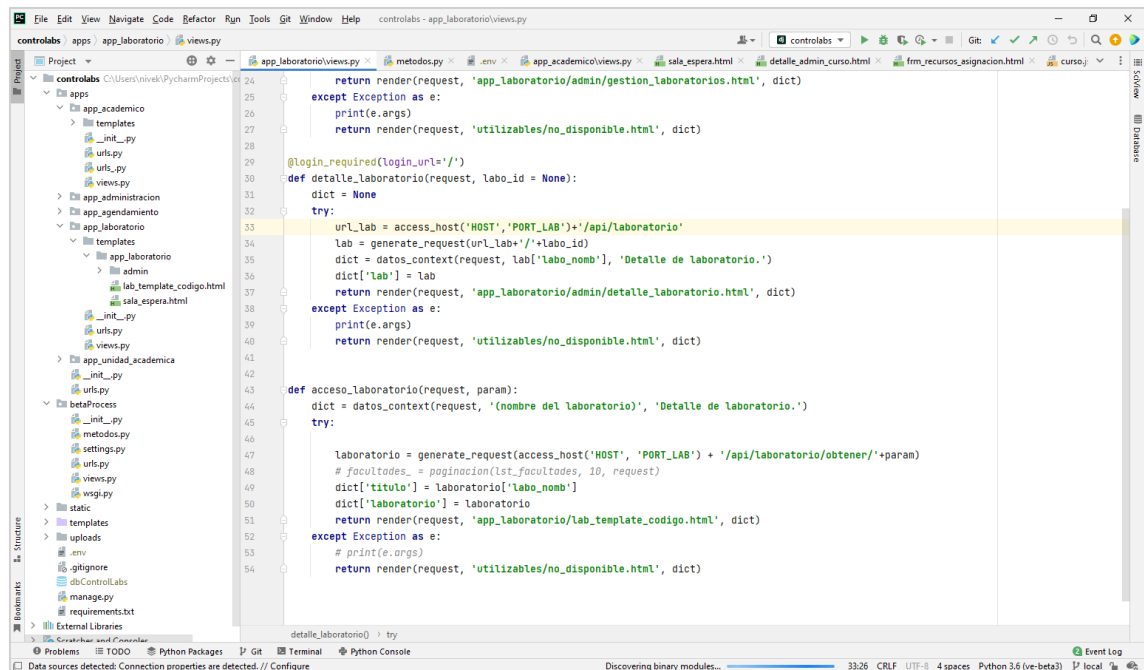


Ilustración 32: Pycharm 2022.1.3 Community Edition

El sistema de gestión de laboratorios, tanto el frontend como los microservicios, fueron desarrollados con diversos lenguajes de programación, los mismos que son: Python y Javascript. Se puede utilizar el VS Code como IDE de desarrollo para ambos lenguajes, pero para el caso de Python, utilizar Pycharm ahorra tiempos de configuración y puesta en marcha, debido a la optimización de sus plugins para el desarrollo en Django.

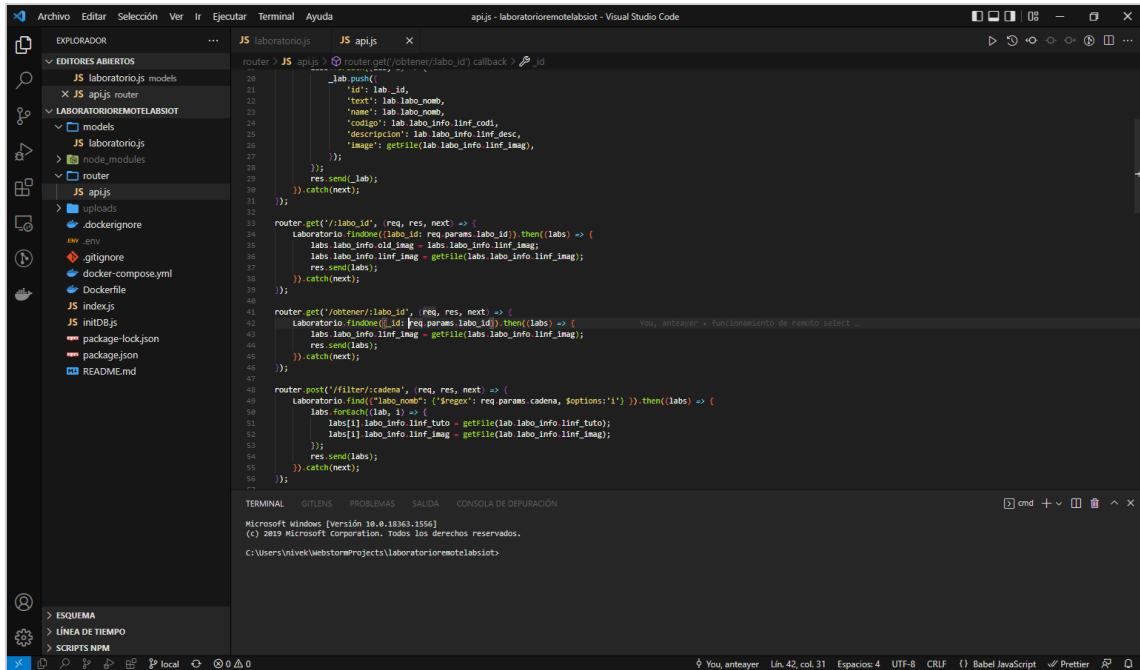


Ilustración 33: Visual Studio Code

3.2.3. Fase de construcción

En esta fase se concreta las etapas que se realizó en la codificación de los microservicios, para posteriormente integrarlos en un API-Gateway, además de establecer las funciones que podrán usar los usuarios del sistema y el diseño de las interfaces.

Direccionamiento de API

Cada microservicio proporciona sus datos, cada uno de ellos tiene su propio enrutamiento, es por ello que fue necesaria la implementación de un API-Gateway en el cual se establecieron el enrutamiento a cada microservicio.

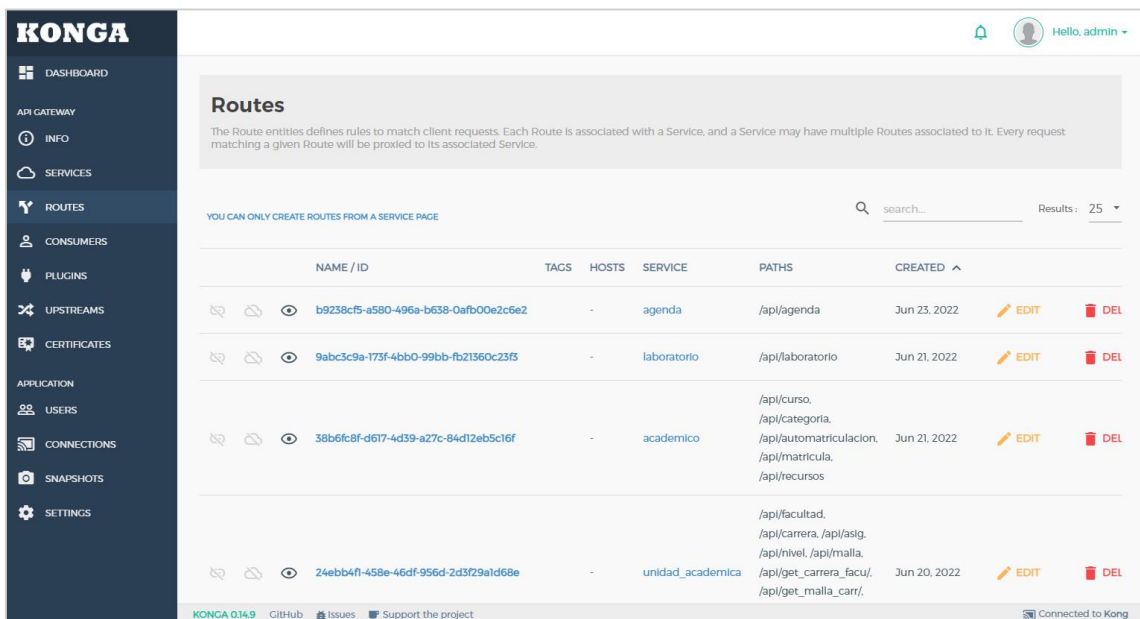


Ilustración 34: Interfaz de Konga - Kong

Como se visualiza en la ilustración 28, se ha definido las rutas de los microservicios en una sola dirección, utilizando Kong a través de su gestor visual Konga. La implementación de este gateway facilita el acceso al frontend ya que interactúa como un único punto de entrada en el conjunto de microservicios.

Servidor de Automatización

Es una herramienta para crear y probar continuamente proyectos de software, lo que facilita la integración de cambios en sus proyectos y la entrega de nuevas versiones a los usuarios. Jenkins permite implementar diferentes “stages” o etapas, en donde se definen los parámetros de configuración para cada etapa. Estas etapas pueden comprender desde el testing hasta el despliegue. En cada aplicación se define un jenkinsfile, que es un archivo de configuración donde se estable los pasos a seguir para construir el servicio. En estos archivos se puede incluir las pruebas del código para verificar el funcionamiento antes de pasar a una etapa de despliegue.

```
1  #!/usr/bin/env groovy
2  pipeline {
3    agent any
4    stages {
5      stage('build') {
6        steps {
7          sh 'docker-compose build'
8        }
9      }
10     stage('deploy') {
11       steps {
12         sh 'docker-compose up -d'
13       }
14     }
15   }
16   post {
17     success {
18       slackSend(
19         color: "good",
20         message: "*Microservicio:* Unidad Académica\n*${currentBuild.currentResult}:*
21         Job ${env.JOB_NAME} build ${env.BUILD_NUMBER}\n More info at: (<${env.BUILD_URL}|Here>)"
22       )
23     }
24     failure {
25       slackSend(
26         color: "danger",
27         message: "*Microservicio:* Unidad Académica\n*${currentBuild.currentResult}:* Job ${env.JOB_NAME}
28         build ${env.BUILD_NUMBER}\n More info at: (<${env.BUILD_URL}|Here>)"
29       )
30     }
31   }
32 }
33 }
```

Ilustración 35: Configuración de los Jenkinsfile

En la ilustración 35 se puede observar la estructura de los jenkinsfiles empleados en las aplicaciones. En este archivo de configuración se establecen las etapas que realizará Jenkins en el momento del despliegue, al finalizar las etapas notifica vía Slack el estado de la compilación del servicio.

3.2.4. Fase de pruebas

La fase de pruebas que propone el ciclo de vida de DevOps, fue realizada en cada ocasión que se generó una nueva versión de los microservicios y de la aplicación. Las pruebas

comprenden en la verificación del funcionamiento sin errores y el cumplimiento de los requerimientos planteados en cada revisión efectuada.

Pruebas Unitarias

Con las pruebas unitarias se comprueba el funcionamiento de partes del código para validar su funcionamiento. Las pruebas fueron diseñadas tanto para Python como para NodeJS, puesto que son los lenguajes principales de las aplicaciones desarrolladas, gracias a estas pruebas, se pudo comprobar que no exista problemas en el funcionamiento ni problemas de sintaxis en el código fuente.

Se puede configurar Jenkins para ejecutar etapas de prueba y se notifique el estado de dichas pruebas. Al utilizar Gitlab como repositorio para la gestión de versiones, se utiliza la herramienta de CI/CD en la que se configura un pipeline para la realización de pruebas al código generado en cada versión. En la ilustración 36 se observa la configuración para realizar pruebas de revisión del código Python utilizando Django. Esta revisión consiste en dos fases la de revisión de migraciones y marco de comprobación que sirve para la validación de un proyecto. Estas comprobaciones permiten detectar problemas en las ramas antes de combinar una versión estable.

```
1  ---
2  variables:
3    MYSQL_DATABASE: db_test
4    MYSQL_ROOT_PASSWORD: mysql
5    MYSQL_USER: user
6    MYSQL_PASSWORD: dbpass
7    MYSQL_HOST: mysql
8  default:
9    image: ubuntu:20.04
10   services:
11     - mysql:8.0
12   cache:
13     paths:
14       - ~/.cache/pip/
15   before_script:
16     - apt -y update
17     - apt -y install apt-utils
18     - apt -y install net-tools python3.8 python3-pip mysql-client libmysqlclient-dev
19     - apt -y upgrade
20     - pip3 install -r requirements.txt
21   migrations:
22     stage: build
23     script:
24       - python3 manage.py makemigrations
25       - python3 manage.py migrate
26   django-tests:
27     stage: test
28     script:
29       - echo "GRANT ALL on *.* to '${MYSQL_USER}';"| mysql -u root --password='${MYSQL_ROOT_PASSWORD}'
30       -h mysql
31       - python3 manage.py check
```

Ilustración 36: Configuración de pipeline de pruebas en Gitlab Runner

Al tener tres aplicaciones desarrollados en lenguaje Python con el framework Django, a las que se aplicaron estas configuraciones para revisar el estado del código. Las pruebas

se realizan en un ambiente con base de datos MySQL en una imagen Docker de Ubuntu. Este ambiente es configurado en el flujo de trabajo con los pipelines de CI/CD que proporciona GitLab como se puede observar en la ilustración 37, en la que la ejecución dio por válida las correcciones del código fuente.

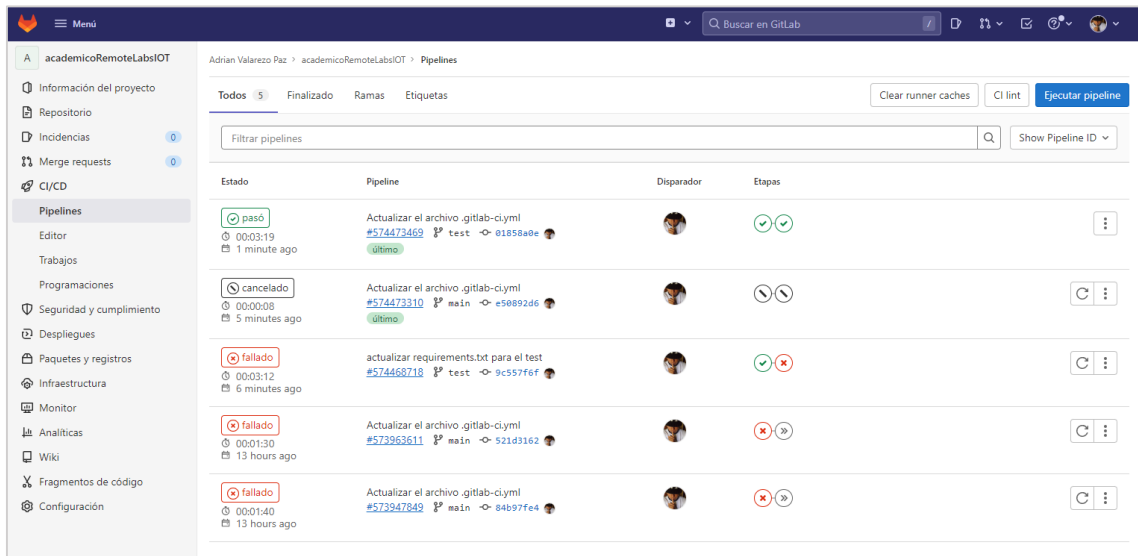


Ilustración 37: Pipelines ejecutados para la comprobación de una aplicación

Pruebas de Aceptación

Son las pruebas manuales en las que se comprueba el cumplimiento de los requisitos. Estas pruebas permiten la detección de errores en el funcionamiento en conjunto de los módulos antes de la aprobación del almacenamiento de algún cambio en el repositorio.

El proceso de estas pruebas arrancó desde el comienzo del proyecto, cada ocasión donde se tenía un cambio antes de ser mezclado con la rama principal del repositorio, primero se corren las pruebas de revisión y luego se procede a revisar manualmente el cumplimiento de las funciones. En caso de haber alguna incidencia en el funcionamiento del código, se procedió a ser corregido antes de cargar los cambios en la rama principal.

3.2.5. Fase de despliegue

Una vez que las aplicaciones pasaron las pruebas, se procede a crear una versión del software de cada una de ellas. El objetivo es indicar que ha pasado con todas las validaciones previas. Las validaciones son usadas para detectar cuál es la versión estable en caso de que exista algún conflicto. Esta fase se integra con el código existente, por lo que el proceso de implementación tiene lugar constantemente. La implementación es realizada de tal manera que cualquier cambio efectuado en el código, no afecta en el funcionamiento del sistema. En esta fase se determinó el ambiente en el que fue

desplegado el sistema, dicho ambiente es de contenedores Docker usando Jenkins. Este proyecto fue desplegado en un ambiente de máquinas virtuales en la herramienta VirtualBOX.

Con Jenkins se estableció el flujo de integración continua para desplegar las aplicaciones. Desde que se realiza un commit en Gitlab, Jenkins detecta los cambios que se han realizado. Compila el código y prepara un “build” en base del jenkinsfile realizado. Si el “build” falla, envía una notificación al equipo de desarrollo. Si resulta exitoso, despliega el servicio. Después de la prueba, Jenkins genera un feedback y notifica al equipo el build y los resultados del testeo.

Al utilizar Jenkins se configuró los respectivos *jenkinsfiles* para cada microservicio a desplegar. Dentro de este archivo de configuración se definió el pipeline de integración y despliegue de las aplicaciones.

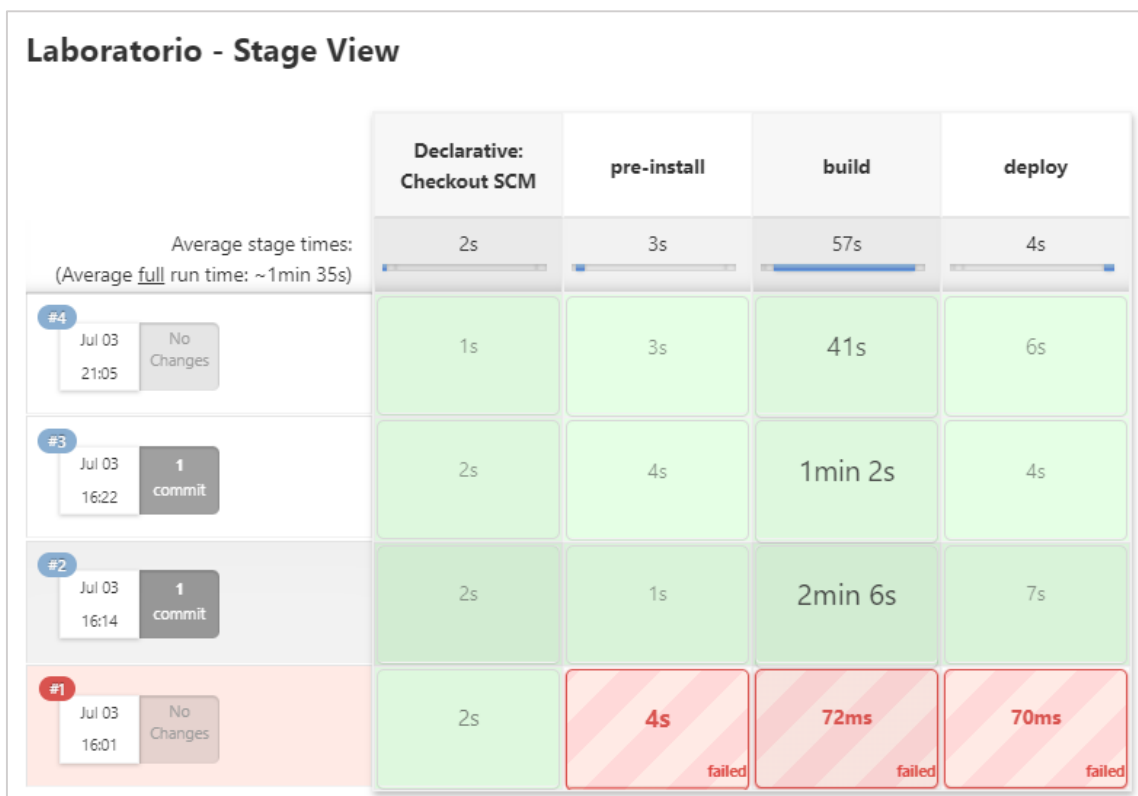


Ilustración 38: Pipeline de despliegue del microservicio Laboratorio

Con Jenkins el trabajo de despliegue para las aplicaciones se ahorra tiempo en el proceso de realizar una secuencia de pasos para conseguir la ejecución de los microservicios. Cada implementación de Jenkins es realizada para ejecutar tareas de despliegue de los servicios a construir, como el caso del sistema frontend que está alojado en la máquina 1 de la infraestructura y del microservicio de comunicación que se aloja en la máquina 3. En la

ilustración 45 se verifica la definición de las tareas para ejecutar los procesos de despliegue de los microservicios alojados en la máquina 2.

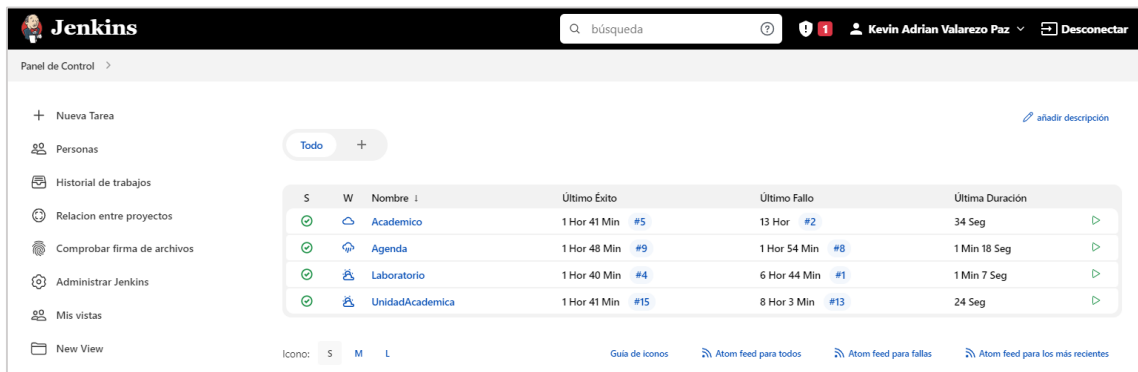


Ilustración 39: Tareas de despliegue de los microservicios

3.2.6. Fase de lanzamiento

La fase de lanzamiento consiste en poner en marcha la versión estable de las aplicaciones, para ello el sistema de gestión de versiones ayuda a gestionar las versiones que fueron lanzadas y brinda la retroalimentación necesaria para conocer cuáles han sido los cambios realizados desde el comienzo del proyecto.

Control de versiones

Es una práctica que sirve para buscar y gestionar los cambios generados en el código fuente. Los sistemas de control de versiones son herramientas que ayudan a gestionar los cambios en el código fuente en el tiempo de desarrollo.

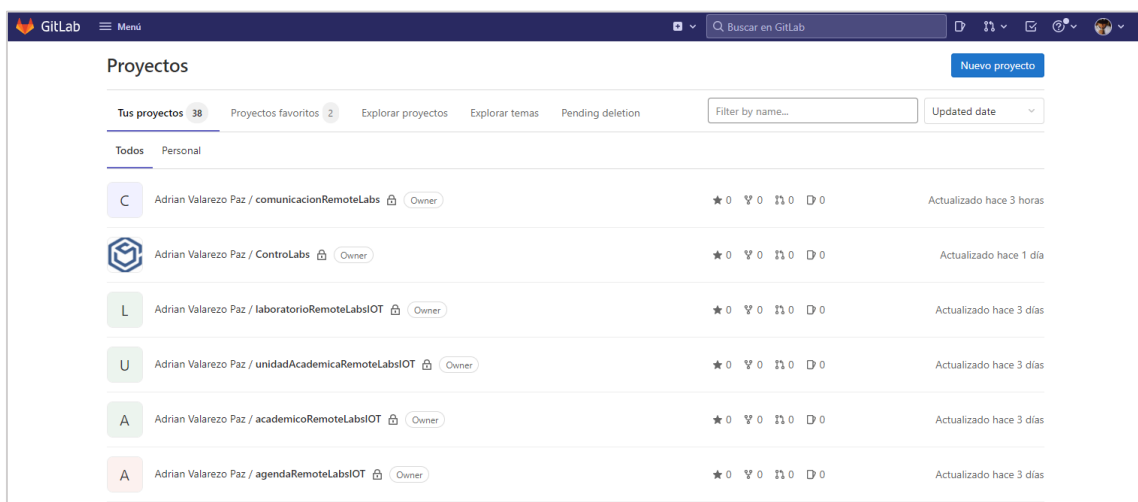


Ilustración 40: Aplicaciones alojadas en GitLab.com

El control de versiones realiza un seguimiento de las modificaciones realizadas en el código fuente a través de un tipo especial de base de datos. Dicha base de datos es gestionada con git y el repositorio que se utilizó fue Gitlab, tal como se observa en la

ilustración 40, en la cual se visualizan los microservicios que forman parte de este sistema de gestión de laboratorios remotos.

Para cada aplicación se establecieron dos ramas de desarrollo:

- **Main o master:** es la rama principal en las aplicaciones, en esta se carga la última versión estable con los cambios aprobados en la fase de pruebas. Las configuraciones de los servicios de las aplicaciones se dirigen a un entorno local.
- **Server:** o rama de lanzamiento, aquí se cargan las versiones que son desplegadas con los cambios derivados de la rama principal, ya que cada vez que la rama main es actualizada, se procede a realizar un “merge” a la rama server. Las configuraciones son establecidas con el direccionamiento del ambiente de producción establecido por la arquitectura IoT.

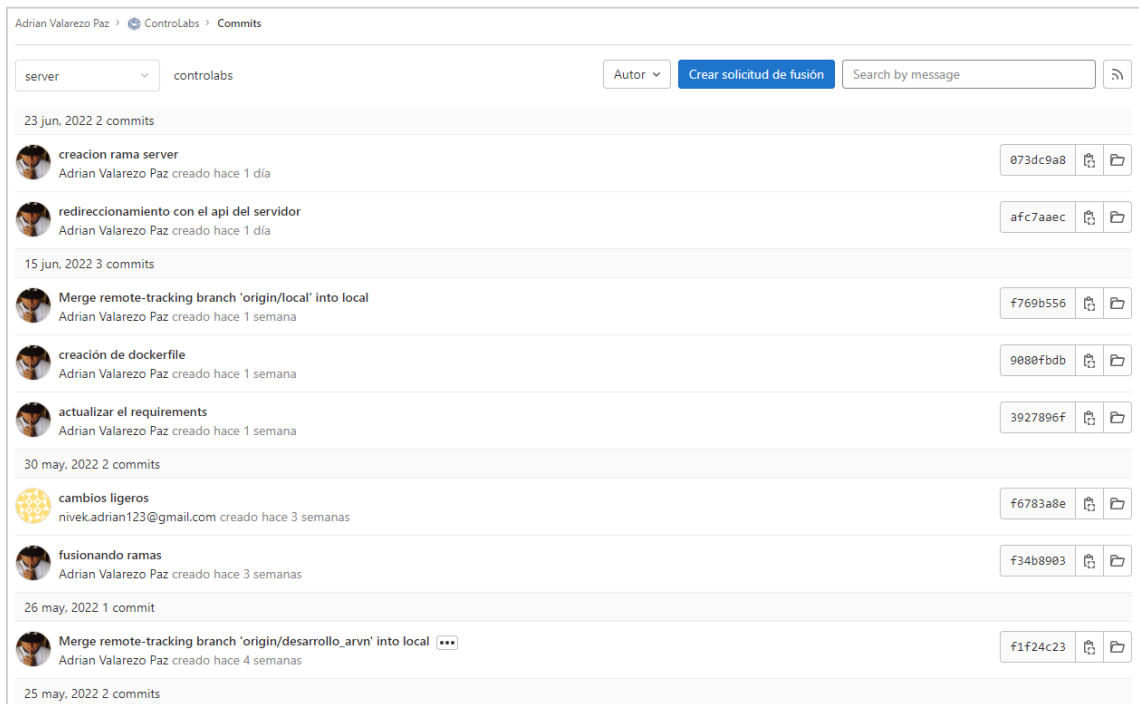


Ilustración 41: Historial de commits en el repositorio de GitLab

En la ilustración 41 se observa el historial de cambios realizados en el proyecto denominado “controlabs”, estos cambios son los que se han realizado en la rama server. Así como esta aplicación, el resto de aplicaciones cuentan con su rama server de la cual despliegan sus servicios. Al desplegar las aplicaciones, Jenkins genera una versión de “build” con información del estado de la ejecución de despliegue, como se observa en la ilustración 47, en donde se indica quien desplegó la tarea, el código de revisión, la información del repositorio junto con la hora y fecha del despliegue.

Build #5 (3 jul. 2022 21:05:06) Conservar esta ejecución para siempre

añadir descripción Started 1 Hor 44 Min ago
Took 34 Seg

Iniciado por el usuario [Kevin Adrian Valarezo Paz](#)

git **Revision:** 9fc6df542f14d721e8befab1f7c94faac9b764ca
Repository: https://gitlab.com/nivek_adrian/academicoremotelabsiot.git

- refs/remotes/origin/server

Ilustración 42: Versión de compilación de pipeline

3.2.7. Fase de operación

La fase de operación tiene como objetivo minimizar las interrupciones o el tiempo de inactividad del servicio. Esta fase se centra en la optimización de las aplicaciones y los entornos para lograr un buen rendimiento alimentado la fase de planificación del desarrollo continuo con informes de errores y comentarios para la realización de retroalimentaciones.

Despliegue máquinas virtuales

Para el alojamiento de la arquitectura IoT se utilizó una computadora cuyas características fueron mencionadas en la tabla 6. En este equipo anfitrión se instalaron cuatro máquinas virtuales de VirtualBox, como se observa en la ilustración 43, en donde se refleja las características de las máquinas implementadas.





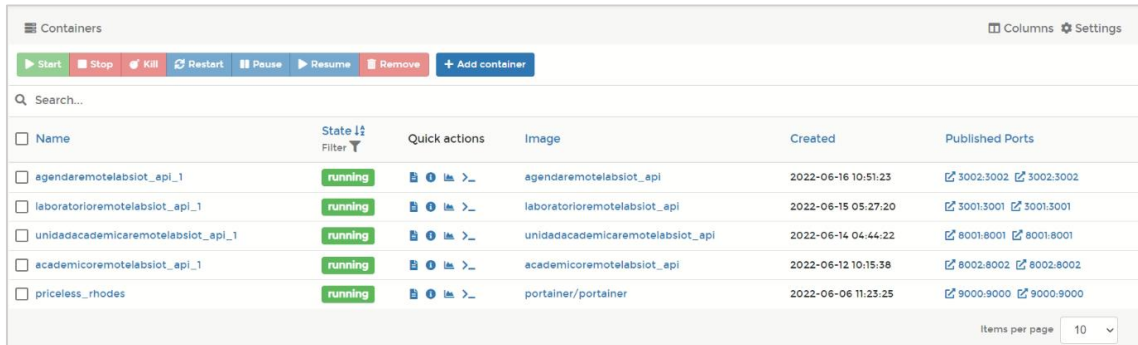
<p>General</p> <p>Nombre: remotelab-maq01 Sistema operativo: Red Hat (64-bit) Grupo: remotelab</p> <p>Sistema</p> <p>Memoria base: 2048 MB Orden de arranque: Disquete, Óptica, Disco duro Aceleración: VT-x/AMD-V, Paginación anidada, PAE/NX, Paravirtualización KVM</p>	<p>Previsualización</p> 
<p>General</p> <p>Nombre: remotelab-maq02 Sistema operativo: Red Hat (64-bit) Grupo: remotelab</p> <p>Sistema</p> <p>Memoria base: 2048 MB Orden de arranque: Disquete, Óptica, Disco duro Aceleración: VT-x/AMD-V, Paginación anidada, PAE/NX, Paravirtualización KVM</p>	<p>Previsualización</p> 
<p>General</p> <p>Nombre: remotelab-maq03 Sistema operativo: Red Hat (64-bit) Grupo: remotelab</p> <p>Sistema</p> <p>Memoria base: 2048 MB Orden de arranque: Disquete, Óptica, Disco duro Aceleración: VT-x/AMD-V, Paginación anidada, PAE/NX, Paravirtualización KVM</p>	<p>Previsualización</p> 
<p>General</p> <p>Nombre: remotelab-maq04 Sistema operativo: Red Hat (64-bit) Grupo: remotelab</p> <p>Sistema</p> <p>Memoria base: 2048 MB Orden de arranque: Disquete, Óptica, Disco duro Aceleración: VT-x/AMD-V, Paginación anidada, PAE/NX, Paravirtualización KVM</p>	<p>Previsualización</p> 

Ilustración 43: Contenedores de los microservicios en máquina 2

Debido a que se cuenta con un host disponible dentro de la infraestructura de red de la IES, para el acceso a los servicios se lo realiza mediante la dirección iotmach.utmachala.edu.ec. En este enlace se despliegan los servicios para que el sistema pueda funcionar de manera correcta tanto dentro de la IES como en el exterior.

Revisión de Contenedores

Las aplicaciones desarrolladas fueron desplegadas en contenedores Docker mismos que se pueden gestionar mediante el uso de Portainer, como en la ilustración 44 se visualiza la lista de despliegue de los microservicios en la máquina 2, al igual que en las demás máquinas virtuales, los servicios desplegados fueron mediante el uso de contenedores.



Name	State	Quick actions	Image	Created	Published Ports
agenderemotelabslot_api_1	running	[Stop] [Kill] [Restart] [Pause] [Resume] [Remove]	agenderemotelabslot_api	2022-06-16 10:51:23	3002:3002 3002:3002
laboratorioremotelabslot_api_1	running	[Stop] [Kill] [Restart] [Pause] [Resume] [Remove]	laboratorioremotelabslot_api	2022-06-15 05:27:20	3001:3001 3001:3001
unidadacademicaaremotelabslot_api_1	running	[Stop] [Kill] [Restart] [Pause] [Resume] [Remove]	unidadacademicaaremotelabslot_api	2022-06-14 04:44:22	8001:8001 8001:8001
academicoremotelabslot_api_1	running	[Stop] [Kill] [Restart] [Pause] [Resume] [Remove]	academicoremotelabslot_api	2022-06-12 10:15:38	8002:8002 8002:8002
priceless_rhodes	running	[Stop] [Kill] [Restart] [Pause] [Resume] [Remove]	portainer/portainer	2022-06-06 11:23:25	9000:9000 9000:9000

Ilustración 44: Contenedores de los microservicios en máquina 2

El uso de Portainer permitió la constatación del funcionamiento de cada servicio desplegado. En la ilustración 45 se puede constatar que cada contenedor ofrece una revisión del estado del consumo de recursos y a su vez realizar determinadas acciones para atender algún problema que surja.

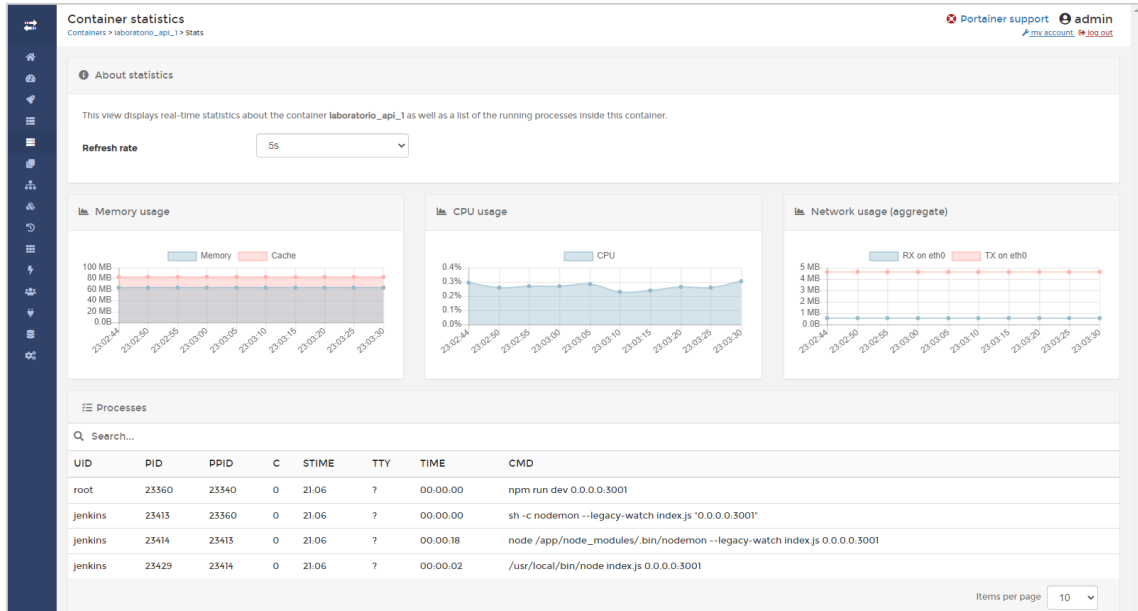


Ilustración 45: Consumo de recurso del contenedor Laboratorio

3.2.8. Fase de monitoreo

En la fase de monitoreo se realiza la supervisión del sistema una vez aprobadas las pruebas y el despliegue de las aplicaciones se notifica que no existe novedad alguna. Una de las herramientas empleadas para el monitoreo del sistema es Slack, como en la

ilustración 46 en la que Jenkins está conectado con un canal Slack y comunica el estado de las tareas de pruebas, lanzamiento y despliegue de los microservicios.

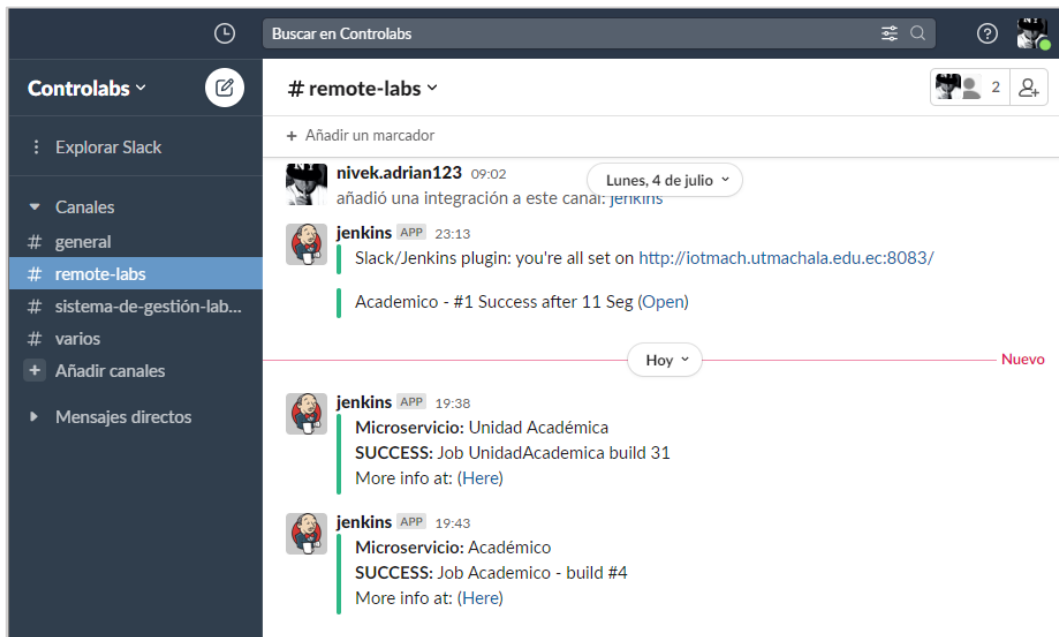


Ilustración 46: Canal de comunicación Slack #remote-labs

Otra forma de monitorear el estado de funcionamiento es usando la herramienta Portainer, debido a que, al ser un gestor visual de Docker, permite acceder a los logs de los servicios desplegados. Con esta herramienta se facilita el acceso a la detección de fallas, puesto que evita ingresar a los logs mediante comandos en la terminal de la máquina virtual, tal como se visualiza en la ilustración 47. Solo basta con dirigirse al enlace de la herramienta y revisar el estado de los contenedores.

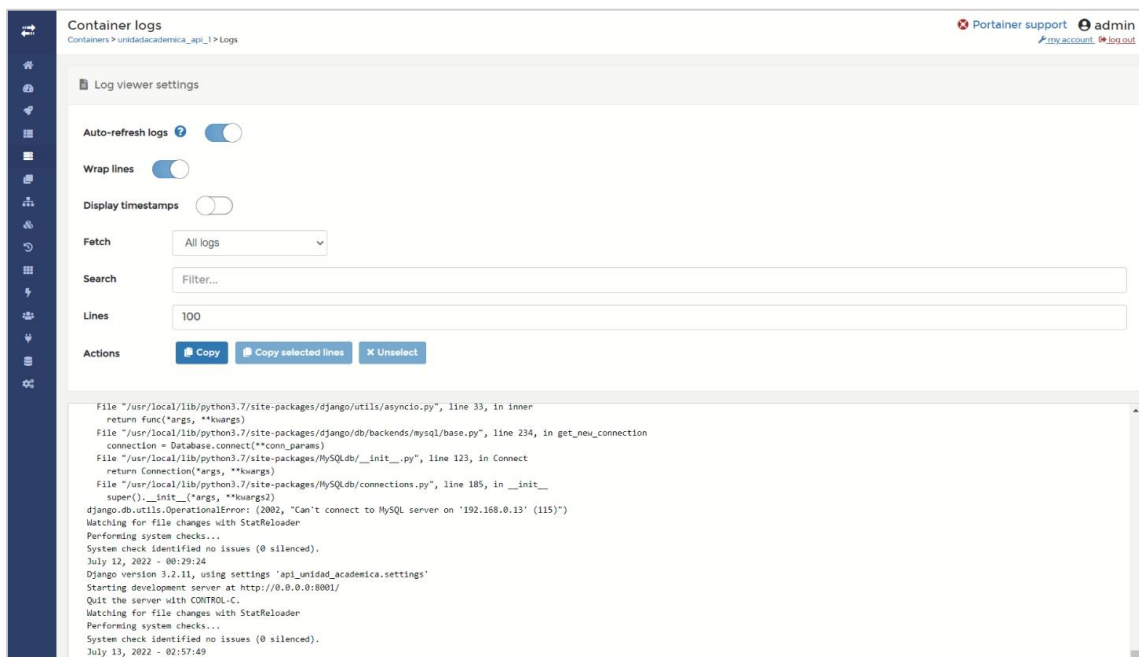


Ilustración 47: Revisión de Logs Contenedor

CAPÍTULO 4. EXPERIMENTACIÓN Y RESULTADOS

En este capítulo se presentará los resultados obtenidos de las experimentaciones realizadas al sistema. En la sección 4.1 se define las pruebas que se realizarán al sistema mediante una configuración experimental, luego en la sección 4.2, trata de los principales hallazgos encontrados, seguido de la sección 4.3 que abarca la relación de los resultados obtenidos con trabajos previos.

4.1. CONFIGURACIÓN EXPERIMENTAL

4.1.1. Objetivos de la experimentación

Luego de realizar el desarrollo del prototipo, se realizaron experimentos para la comprobación de los objetivos planteados en esta investigación. Las pruebas experimentales realizadas fueron de calidad, latencia, concurrencia de usuarios y de rendimiento, cuyos objetivos y métricas se detallan en la tabla 14.

Nro.	Prueba	Objetivo	Métricas
1	Calidad	Evaluar la calidad de uso y el rendimiento de la aplicación web.	Niveles de rendimiento del aplicativo web Valoración de criterios de la norma ISO/IEC 9126
2	Latencia	Medir los tiempos de respuesta de la transferencia de tramas de datos	Tiempos promedio de envío y recepción de tramas
3	Concurrencia de acceso a los laboratorios	Conocer las conexiones de acceso a los laboratorios	Número de usuarios con accesos a laboratorios Número de laboratorios accedidos por usuarios
4	Rendimiento	Medir el consumo de los recursos hardware utilizados	Cantidad de recursos utilizados

Tabla 14: Objetivos de los experimentos y métricas

4.1.2. Escenario de pruebas

Las pruebas se efectuaron en un escenario que comprende el servidor donde estuvieron desplegadas las herramientas y los microservicios desarrollados. Este servidor se conectó a la infraestructura de red de una IES en la ciudad de Machala y se utilizó el dominio <http://iotmach.utmachala.edu.ec/> que se encontraba disponible, también se contó con una maqueta que constaba de un microcontrolador ESP32 y varios periféricos junto a una cámara IP. Además de contar con un cliente MQTT cuya función es simular el proceso

de envío/recepción de los laboratorios remotos. Estas condiciones se observan de manera gráfica en la ilustración 48

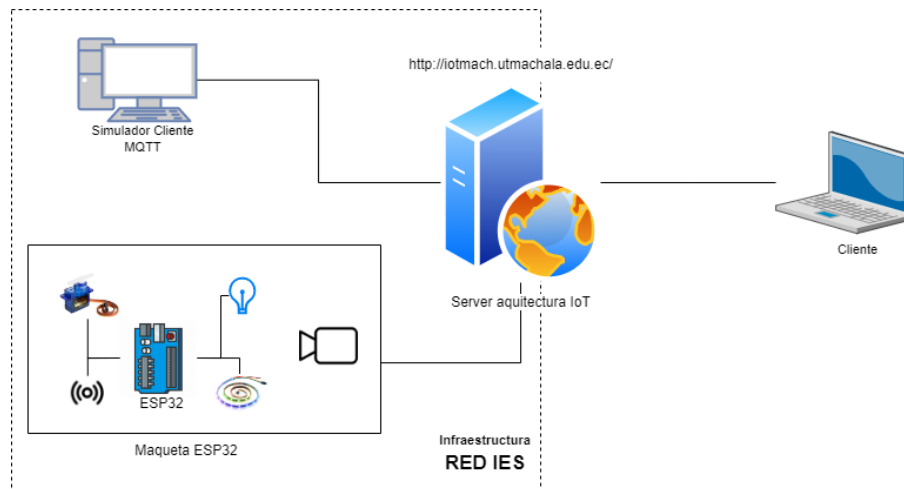


Ilustración 48: Escenarios de experimentación

4.1.3. Características de equipos

El despliegue de la arquitectura IoT y de la aplicación desarrollada se realizó en un computador personal que efectuó las funciones de servidor. Este equipo cuenta con las características descritas en la tabla 15, en el mismo se albergaron las máquinas virtuales donde se desplegó las herramientas y servicios del sistema.

Computadora Anfitriona			
Procesador	Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz		
Arquitectura	x64	RAM	12 GB
Sistema Operativo	Windows 10 Pro	Disco Duro	1 TB

Tabla 15: Características del Equipo Anfitrión del servidor

Las máquinas virtuales que se montaron en el equipo de la tabla 15 se le configuró a cada una de ellas 4GB de memoria RAM con 10GB de espacio en el almacenamiento y Centos7 como sistema operativo. En cuanto al equipo que albergó el simulador de cliente MQTT y del computador que fue utilizado como cliente del sistema, ambos equipos cuentan con las características del procesador, la memoria RAM y el sistema operativo que se detallan en la tabla 16.

Computador Simulador MQTT y Computador Cliente			
Procesador	Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz		
Arquitectura	x64	RAM	8 GB
Sistema Operativo	Windows 10 Pro	Disco Duro	1 TB

Tabla 16: Características de los equipos cliente

Los equipos anteriormente mencionados se los utilizaron para realizar las pruebas experimentales definidas en los objetivos de experimentación.

4.1.4. Caso de estudio

En la asignatura de Internet de las Cosas, los estudiantes han desarrollado maquetas como parte de la evaluación final en la materia. Una de las maquetas que se adaptó para la realización de las pruebas del sistema es la observada en la ilustración 49, la cual se trata de un proyecto que utiliza un microcontrolador ESP32 con periféricos de entrada y salida. A esta maqueta se le agregó un soporte que sujeta una cámara IP. Esta maqueta fue configurada para ejecutar el plan piloto de pruebas para comprobar del control y la gestión de acceso a los recursos de los laboratorios remotos.



Ilustración 49: Maqueta de ESP32 adaptada con cámara de video

El plan piloto consiste en que los usuarios acceden a un escenario de prácticas de laboratorio, el cual contiene un editor de código, el mismo que compila el código escrito en lenguaje C para Arduino a través de WebSockets con el microservicio laboratorio. Una vez compilado, el microservicio permite descargar el archivo de configuración que es quemado en el firmware del ESP32, y de esta manera, ejecutar acciones desde el encendido de luces hasta el movimiento de un servomotor.

4.2. HALLAZGOS FUNDAMENTALES

Al finalizar esta investigación, se logró la implementación de manera exitosa la arquitectura IoT basada en microservicios. Fueron cuatro las máquinas virtuales utilizadas para el despliegue de la arquitectura en donde se implementó el sistema de gestión de acceso a los laboratorios remotos. Esta aplicación fue desarrollada en Python 3.7 con la utilización del framework Django 3.2.1, cuya comunicación con los microservicios fue mediante el uso de API-REST proporcionado por un API-GATEWAY y para la comunicación con los laboratorios fue con el uso de Websockets conectado a un microservicio que publica y recibe data vía MQTT. Los microservicios y bases de datos

se alojaron en contenedores Docker. Tras la correcta implementación de la arquitectura y de la aplicación antes mencionadas, se obtuvieron los siguientes hallazgos fundamentales:

4.2.1. Prueba de concepto

El sistema de gestión de acceso a los laboratorios remotos ofrece la creación de cursos para las asignaturas de las diferentes facultades de la IES en la ciudad de Machala, dentro de los cursos se puede gestionar los usuarios participantes del curso y los recursos que dispondrá para su utilización, los recursos son los laboratorios remotos. El sistema permite gestionar la creación y configuración de los laboratorios, tanto su conexión como el tipo de escenario con el que trabajan. Los usuarios disponen también de una agenda en la que pueden reservar el uso del laboratorio remoto.

Al ingresar a la dirección <http://iotmach.utmachala.edu.ec/>, el sistema muestra una interfaz donde se ingresan las credenciales para tener acceso al sistema. Como fue definido en el desarrollo del prototipo, existen tres roles de usuarios: Estudiantes, Docentes, Administradores. En la ilustración 50 se puede observar como un usuario con rol de administrador hace ingreso al sistema. Una vez que sus credenciales fueron válidas se le permite el ingreso al sistema en donde la pantalla principal muestra los menús de acceso a los módulos de administración.

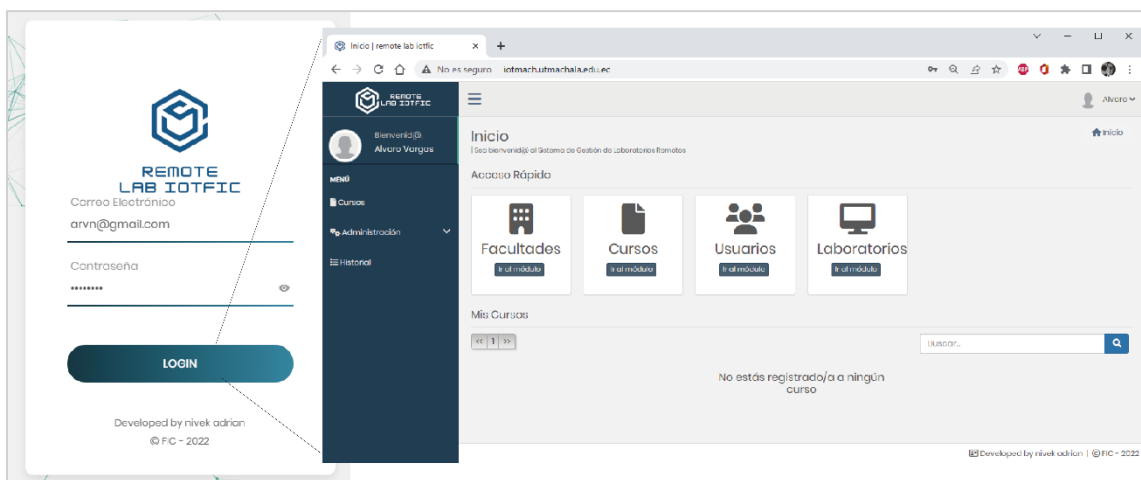


Ilustración 50: Inicio de sesión de un usuario administrador

Un usuario administrador tiene acceso al módulo de facultades, cursos, usuarios y laboratorios. En el módulo de usuarios se listan los usuarios del sistema, además de contar con un formulario de registro y edición de la información de los usuarios, como se visualiza en la ilustración 51. En esta interfaz también permite el bloqueo de acceso al sistema a cualquier usuario. Los datos necesarios para el registro de un usuario son los

nombres, apellidos, correo electrónico y el rol que se les asigna, cuando se crea un nuevo registro de usuario se genera una contraseña por defecto, la misma que tendrá que ser cambiada cuando el usuario ingrese al sistema por primera vez.

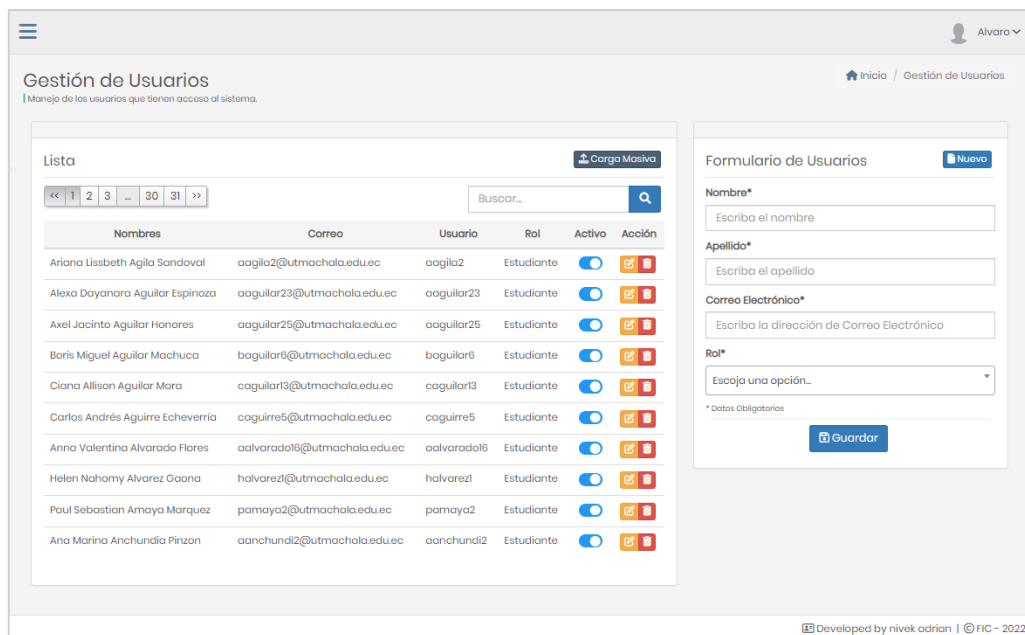


Ilustración 51: Interfaz de gestión de usuarios del sistema

El administrador tiene en la interfaz de gestión de cursos la opción de crear cursos relacionados con la asignatura de una carrera. En la ilustración 52, se visualiza el listado de los cursos creados además del formulario en donde se agrega la información descriptiva de cada curso.

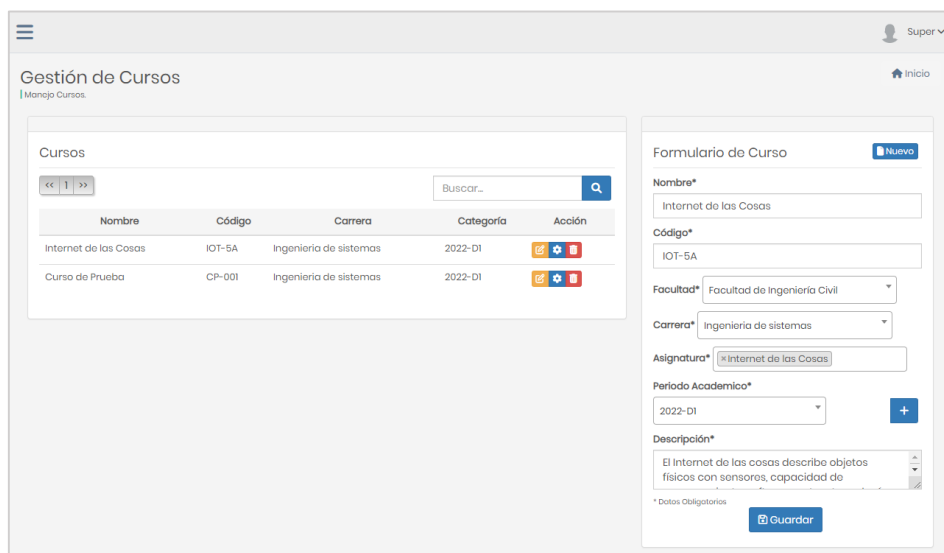


Ilustración 52: Interfaz de gestión de los cursos

Además de crear los cursos, también hay que realizar la configuración de los mismos. Para ingresar a la interfaz de configuración se puede acceder a través de la lista de cursos o dentro de la vista del curso, como se observa en la ilustración 53. En la interfaz de

configuración de curso existen tres pestañas la de configuración de clave, la gestión de participantes y la gestión de recursos. La pestaña clave se encarga de administrar una contraseña de matriculación con la que los estudiantes deben registrarse para tener acceso a los laboratorios.

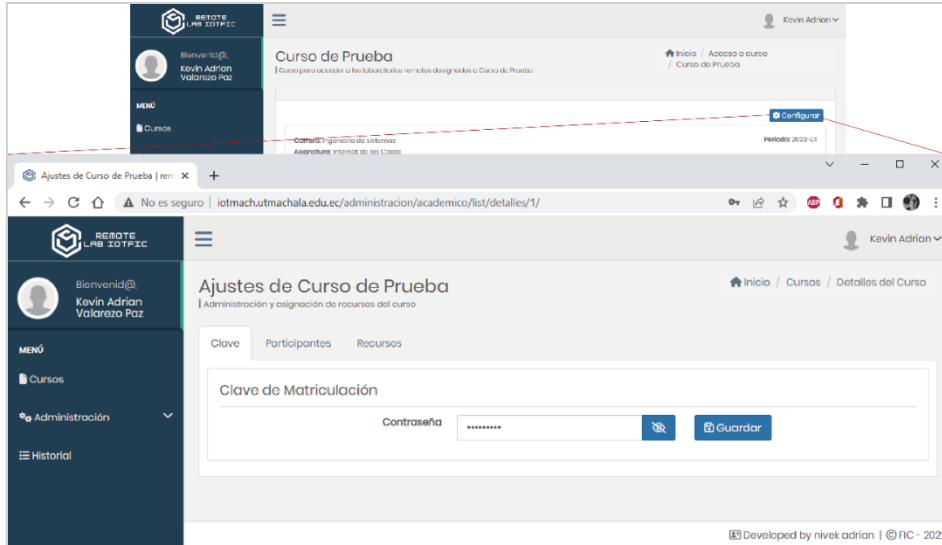


Ilustración 53: Acceso a la configuración de Curso

En la ilustración 54 se muestran las pestañas de participantes y recursos que también forman parte de la configuración del curso. En la pestaña participante se gestiona a los usuarios que tienen acceso al curso, en esta opción se puede dar de baja a los usuarios que están listados y agregar más participantes de ser necesario. En cambio, en la pestaña de recursos se gestiona los laboratorios que dispondrá el curso, en este apartado se agrega o se quita estos recursos.

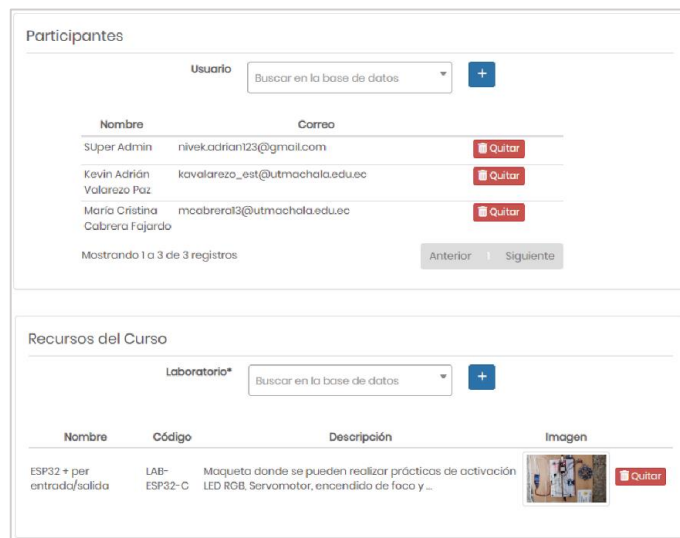


Ilustración 54: Vista de participantes y recursos

En la gestión de laboratorios, se registran los datos informativos del laboratorio, una imagen representativa y el manual de utilización. En la interfaz que se observa en la

ilustración 55, se listan los laboratorios creados y a través de ellos se accede a la configuración de estos, y en la parte derecha está el formulario de creación y edición de la información de laboratorios.

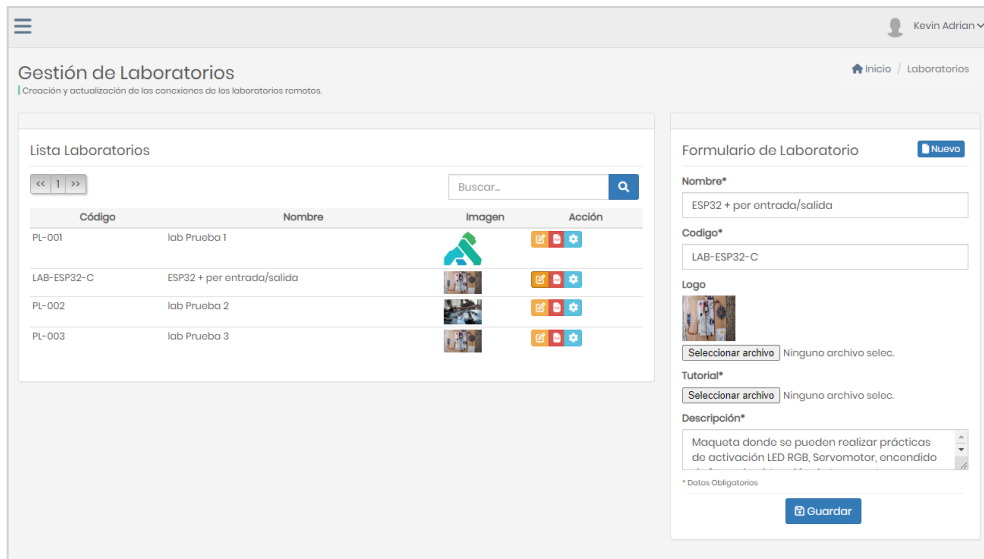


Ilustración 55: Acceso a la gestión de laboratorios

En la configuración de laboratorios existen dos pestañas, la pestaña denominada como configuración y la siguiente pestaña es denominada como escenario. La ilustración 56 describe como es la pestaña configuración, la cual se trata de un formulario para ingresar datos de conexión al bróker Mosquitto en el que el laboratorio está conectado y los tópicos donde circula el tráfico de tramas de comunicación, se ingresa también la URL de la cámara que sirve para monitorear la maqueta.

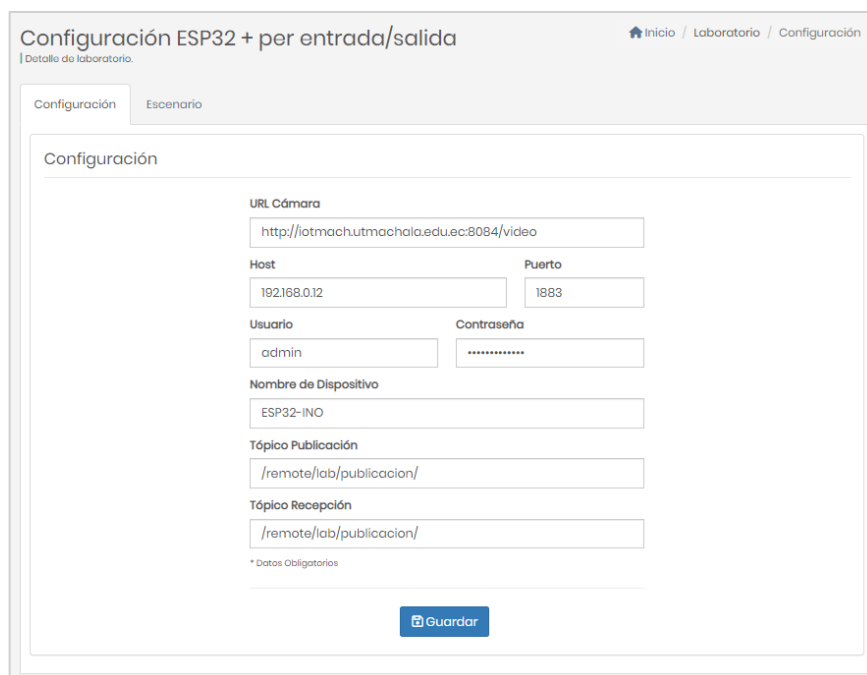


Ilustración 56: Configuración de conexión de laboratorios remotos

En la pestaña escenario, se configura el nivel de escenario con el que el laboratorio remoto trabaja. En la ilustración 57, se observan los niveles de escenarios con los que cuenta el sistema que son dashboard, compilador y el denominado como web embebida. En esta sección se configuran los recursos que cuenta el laboratorio para realizar la práctica desde identificar los periféricos o definir el lenguaje de programación para trabajar la práctica.

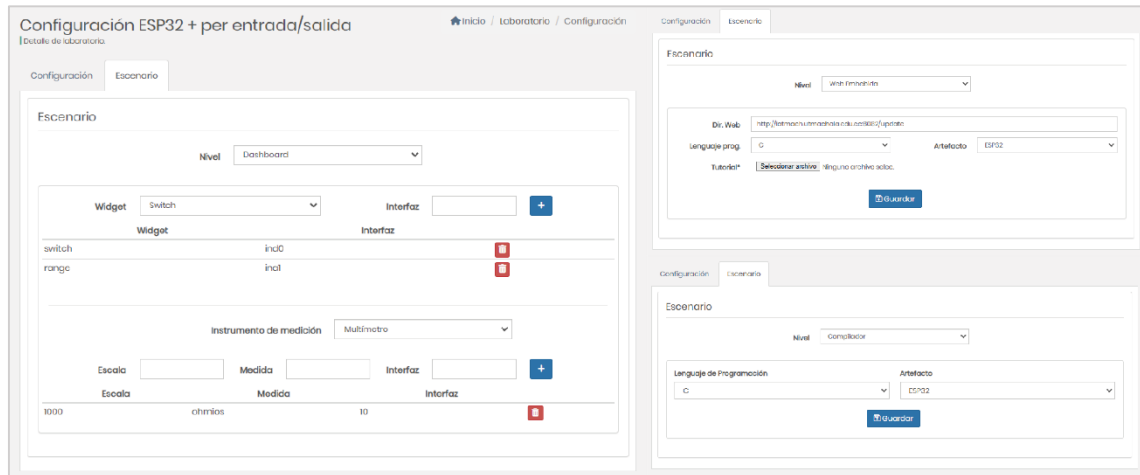


Ilustración 57: Tipos de configuración de escenarios de laboratorios remotos

Además del administrador, un usuario con rol docente tiene acceso a la configuración de un curso y la configuración de laboratorios. Una vez que ingresa sus credenciales, el sistema solo les da acceso a las opciones mencionadas, tal como se observa en la ilustración 58.

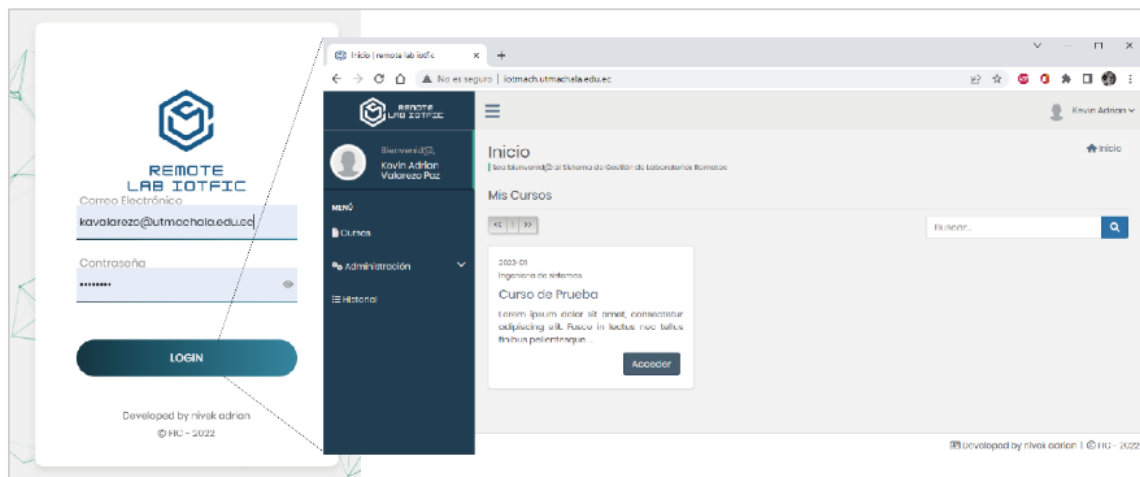


Ilustración 58: Inicio de sesión de usuario con rol docente

En cambio, un usuario con rol de estudiante puede ver los cursos disponibles del sistema, automatriculase a los cursos, acceder a los laboratorios y realizar reservaciones. En la ilustración 59, se observa la interfaz principal del sistema luego de iniciar sesión.

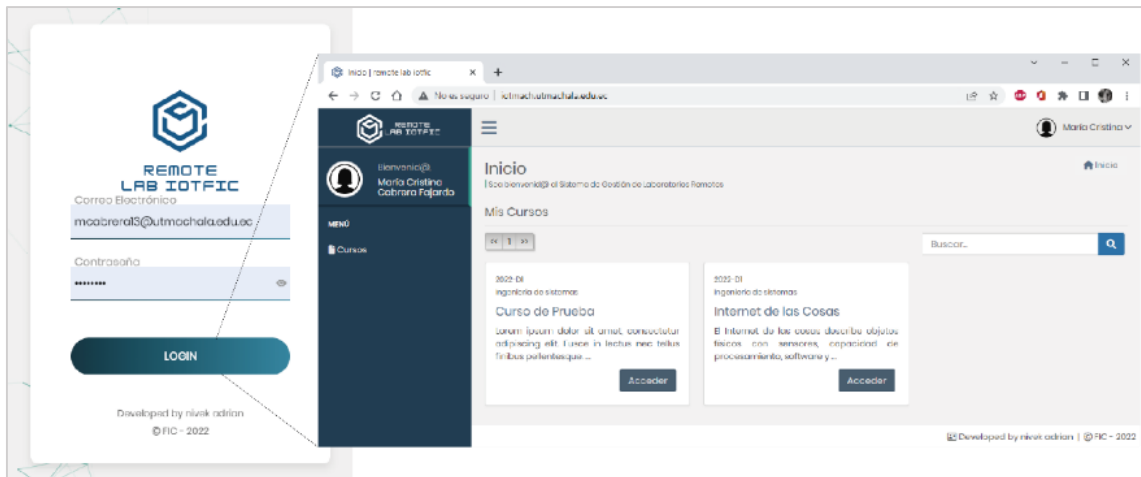


Ilustración 59: Inicio de sesión de usuario con rol estudiante

Los estudiantes pueden acceder a los cursos a través del menú lateral, el cual los redirecciona a una lista de los cursos disponibles. Una vez que se ha seleccionado el curso, el sistema le solicita al estudiante la contraseña de automatriculación, proceso que se puede observar en la ilustración 60.

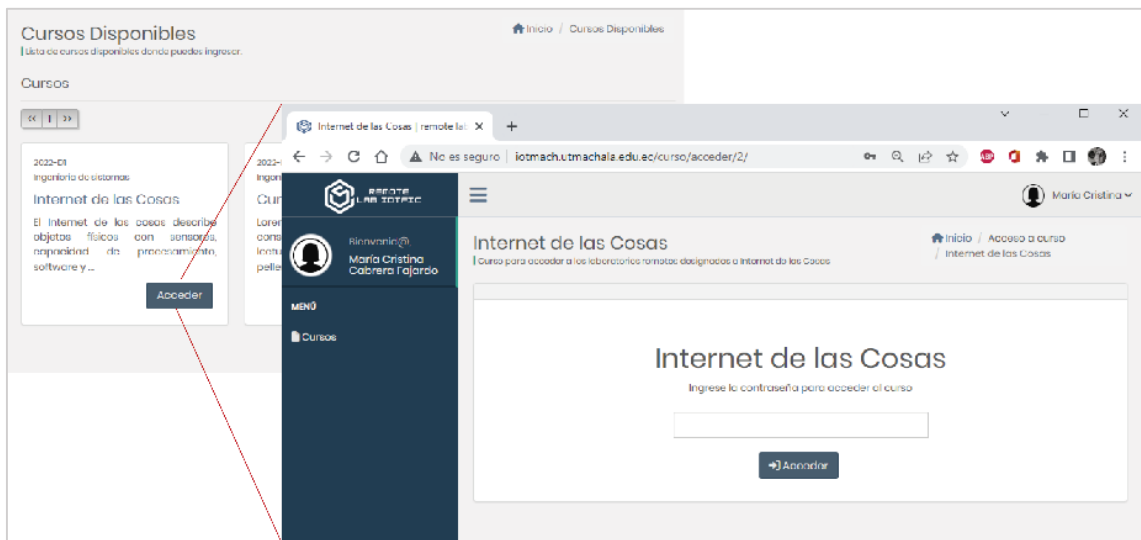


Ilustración 60: Proceso de automatriculación a los cursos

Si la credencial es correcta el estudiante se convierte en participante del curso y tendrá acceso a los laboratorios que están disponibles dentro del curso. El estudiante observa una interfaz como la que visualiza en la ilustración 61, la cual muestra los datos informativos del curso y los laboratorios asignados que el estudiante puede hacer uso.

Internet de las Cosas Inicio / Acceso a curso / Internet de las Cosas

Curso para acceder a los laboratorios remotos designados a Internet de las Cosas

Carrera: Ingeniería de sistemas **Periodo:** 2022-D1
Asignatura: Internet de las Cosas

Descripción:
 El Internet de las cosas describe objetos físicos con sensores, capacidad de procesamiento, software y otras tecnologías que se conectan e intercambian datos con otros dispositivos y sistemas a través de internet u otras redes de comunicación.

Laboratorios Disponibles

 **Código Laboratorio:** LAB-ESP32-C **Acceder**

ESP32 + per entrada/salida

Descripción
 Maqueta donde se pueden realizar prácticas de activación LED RGB, Servomotor, encendido de foco y la obtención de temperatura

Darse de Baja

Ilustración 61: Proceso de automatriculación a los cursos

Cuando se pulsa el botón acceder de cualquier laboratorio, el estudiante es llevado a la sala de espera, como en la parte superior de la ilustración 62, donde muestra la cantidad de usuarios que están utilizando el escenario de prácticas, si el contador está en cero significa que el laboratorio está disponible siempre y cuando no tenga una reserva activa, de no haber reservación se le concede el acceso al usuario. En cambio, si un usuario desea reservar el laboratorio para un horario posterior e invitar a más participantes en ese mismo horario, el estudiante tiene la opción de agendar.

Sala de Espera - ESP32 + per entrada/salida Inicio / Curso / Sala de espera

ESP32 + per entrada/salida **Código:** LAB-ESP32-C

0 usuarios

Acceder **Agendar**

Agendamiento
 octubre de 2022

dom	lun	mar	mié	jue	vie	sáb

Ilustración 62: Sala de espera de laboratorios

Cuando un usuario accede al laboratorio, según el escenario de prácticas configurado se visualizará en la pantalla. En la ilustración 63 se observa los elementos que forman parte de los distintos escenarios de prácticas de laboratorio, en donde cada escenario contiene una cámara y elementos de manipulación de datos, desde codificación hasta botones que realizan acciones.

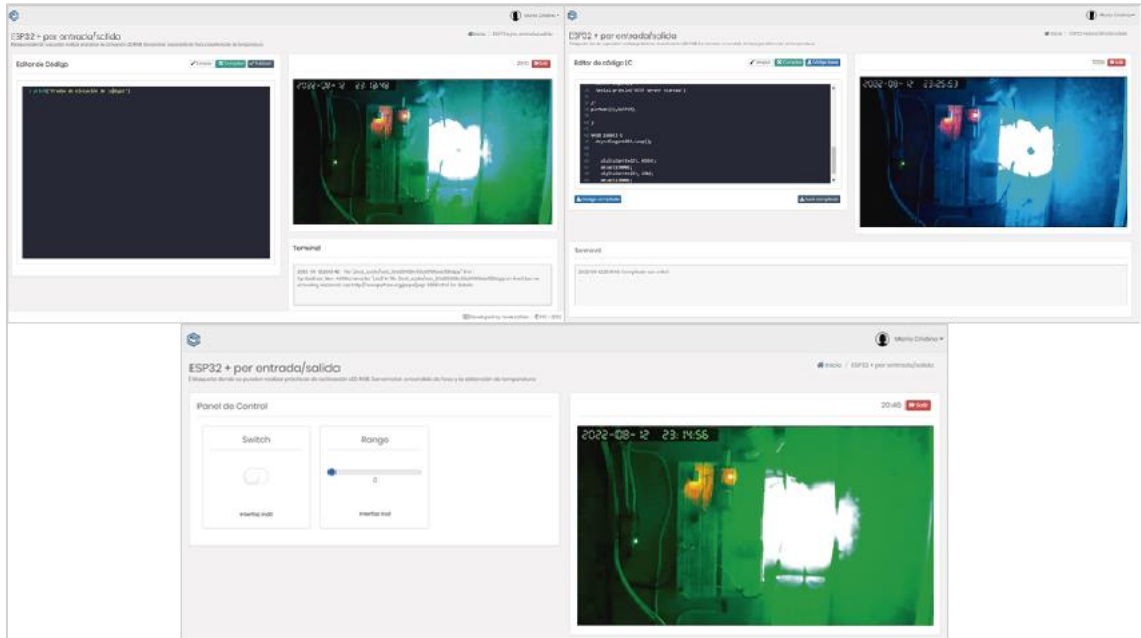


Ilustración 63: Escenarios de prácticas de laboratorio

Si un estudiante desea reservar un laboratorio, desde la sala de espera accede a la interfaz de agendamiento, selecciona una fecha y le aparecerá una ventana donde configura la fecha y hora de reserva, además de agregar a más personas en la reserva en la opción que indica la ilustración 63.

+ Agenda
✕

Laboratorio*

Nombre*

Hora inicio*

Usuario

Cod. Lab*

Fecha*

Hora fin*

Nombre	Apellido	Correo	Username
María Cristina	Cabrera Fajardo	mcabrera13@utmachala.edu.ec	mcabrera13
Cabrera Tigrero	Mayker Xavier	mcabrera14@utmachala.edu.ec	mcabrera14
Rodriguez Zumba	Erick Steeven	erodrigue2@utmachala.edu.ec	erodrigue2

* Datos Obligatorios

Cancelar
Guardar

Ilustración 64: Escenarios de prácticas de laboratorio

Como se demuestra con la prueba de concepto, al crear nuevos laboratorios se generan nuevos accesos a recursos para la realización de prácticas académicas de una asignatura, lo que permite a los docentes tener a disposición más herramientas para el desarrollo de sus clases, y a los estudiantes les permite acceder a los laboratorios en cualquier instante.

4.2.2. Comprobación de caso de estudio

Para la comprobación del caso de estudio, se creó el escenario “web embebida”, el mismo que permitió comprobar el sistema de gestión de acceso al laboratorio remoto. La prueba consiste en que mediante el escenario un estudiante ingrese el código necesario para ejecutar una acción, este código será compilado y quemado en el firmware de un microcontrolador ESP32, este escenario se puede ver reflejado en la ilustración 65.

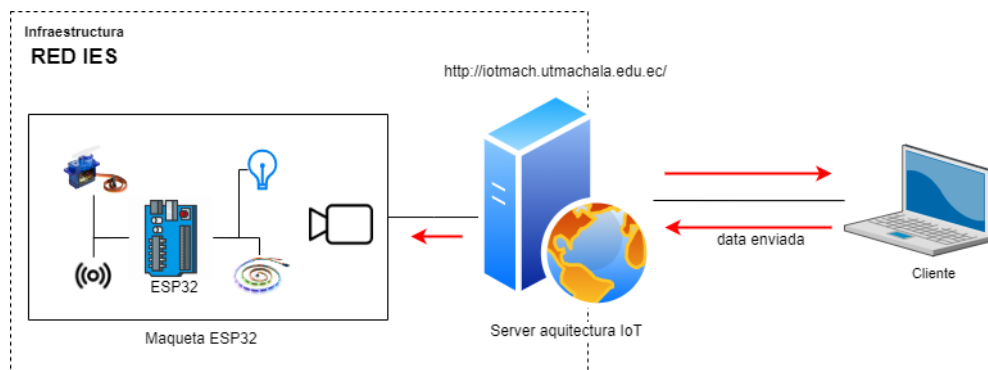


Ilustración 65: Escenario de pruebas de caso de estudio

Con ayuda de estudiantes de la carrera de Tecnologías de la Información en la asignatura “Internet de las Cosas” se comprobó el acceso al laboratorio y funcionamiento del escenario, como se puede observar en la ilustración 65.

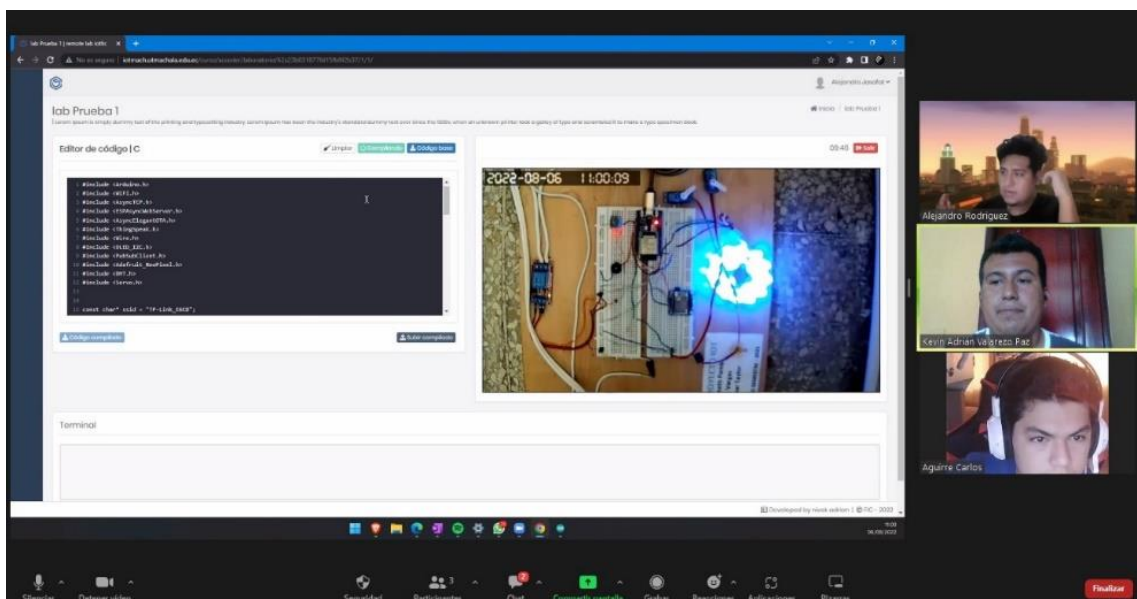


Ilustración 66: Estudiantes de la asignatura de IOT probando el sistema

El flujo de la prueba del escenario comienza con ingresar el código a compilar, luego se pulsa el botón compilar, que envía el código al microservicio para ser evaluado si tiene algún problema, de ser correcto el código se compila en un archivo binario que activa el botón de código compilado que permite la descarga de este archivo. Una vez descargado el archivo, se procede a dar clic en el botón de subir compilado que muestra una ventana emergente donde se carga el archivo compilado que tiene como función la de cambiar el firmware del ESP32 y encender algún periférico conectado.

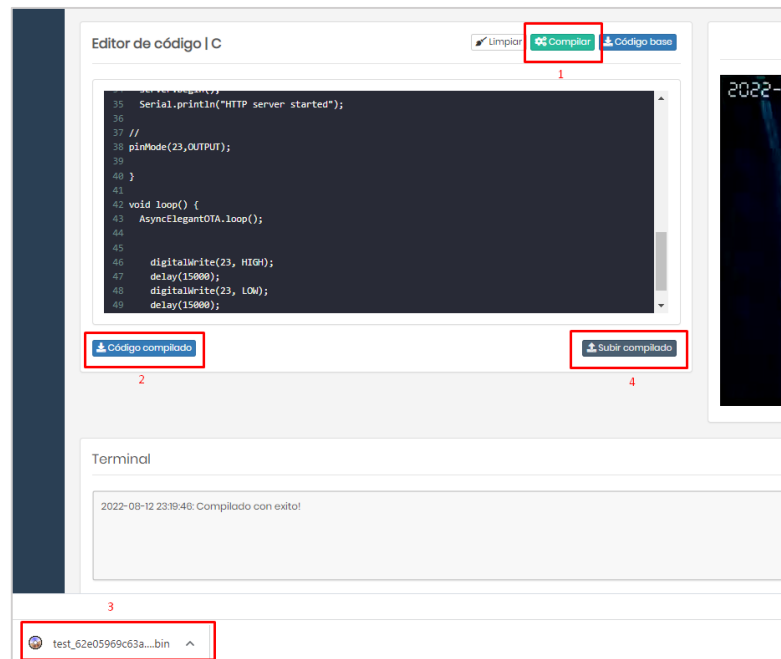


Ilustración 67: Secuencia de pasos del escenario de prácticas del caso de estudio

4.2.3. Evaluación de Calidad

Para la evaluación de la calidad del sistema se aplicó la norma ISO/IEC 9126, la cual según [104], esta norma permite establecer el nivel de calidad que el sistema basado en parámetros de funcionalidad, usabilidad, fiabilidad, eficiencia y portabilidad. Estos parámetros representan el cumplimiento de los requerimientos de usuario en la implementación del sistema.

Implementación Norma ISO/IEC 9126

La implementación de la norma ISO/IEC 9126 permite evaluar al sistema mediante la realización de un cuestionario que incluye criterios establecidos en la métrica interna y externa de la norma. Para evaluar los resultados se estableció la utilización de la escala de Likert, que determina al 1 como pésimo hasta 5 calificado como excelente según lo indica [105]. Estas escalas se utilizan para determinar la percepción de alguna variable cualitativa que por su naturaleza denota algún orden.

Una vez definida la escala de valoración, mediante el uso de la norma ISO/IEC 9126 con sus métricas, se procedió a la evaluación del sistema de gestión de laboratorios remotos por medio del uso de los criterios establecidos en la tabla 17. Las características evaluadas son la funcionalidad, eficiencia, usabilidad, portabilidad y mantenibilidad.

Características	Subcaracterísticas	Criterio	Valoración
Funcionalidad	Adecuación	El sistema cumple con la capacidad de realizar las tareas establecidas en los requerimientos	5
	Exactitud	El sistema devuelve los resultados esperados de manera correcta	5
	Interoperabilidad	El sistema tiene la capacidad de interactuar con otro sistema	4
	Conformidad	El sistema cumple con los estándares de funcionalidad	5
	Seguridad	El sistema impide el acceso no autorizado	5
Total Funcionalidad			24/5 = 4.8
Eficiencia	Comportamiento en el tiempo	El tiempo de respuesta del sistema es adecuado	5
Total Eficiencia			5/1 = 5
Usabilidad	Aprendizaje	El usuario aprende fácilmente el uso de la aplicación	5
	Comprensión	Es de fácil entendimiento y comprensión las tareas y funciones que realiza el sistema.	4
	Operatividad	El usuario interviene pocas veces para operar el sistema.	4
	Atractividad	Las interfaces del sistema son adaptables y amigables para el usuario.	5
Total Usabilidad			18/4 = 4.5
Portabilidad	Facilidad de Instalación	La instalación del software es fácil	5
	Reemplazabilidad	Es fácil reemplazar el software por otros con funcionalidades similares	3
	Conformidad	El software está desarrollado con algún lenguaje multiplataforma	5
	Coexistencia	El software tiene la capacidad de interactuar con otro sistema	4
	Adaptabilidad	El software es fácil de adaptar a varios entornos de trabajo	5
Total Portabilidad			22/3 = 4.4
Mantenibilidad	Capacidad de Análisis	Es fácil identificar fallas para ser corregidas	5
	Capacidad de Modificación	El software se puede modificar ante cualquier cambio del mismo	5
	Estabilidad	El software se mantiene funcional pese a cambios	5
	Facilidad de prueba	Es fácil realizar pruebas del sistema	5
Total Mantenibilidad			20/5 = 5

Tabla 17: Evaluación de calidad bajo la norma ISO/IEC 9126

La aplicación de la evaluación de calidad determina que el sistema evaluado cumple con las características de la norma ISO/IEC 9126, debido a que se obtuvo un alto porcentaje de aceptación en los criterios de evaluación establecidos, según se muestra en la tabla 17. Para ello se procedió a realizar el cálculo de la puntuación del prototipo del sistema con la utilización de la siguiente ecuación:

$$\text{Puntuación del Sistema de gestión de acceso} = \frac{F + E + U + P + M}{5}$$

$$\text{Puntuación del Sistema de gestión de acceso} = \frac{4.8 + 5 + 4.5 + 4.4 + 5}{5}$$

$$\text{Puntuación del Sistema de gestión de acceso} = 4.74$$

En donde,

F = puntaje promedio de Funcionalidad

E = puntaje promedio de Eficiencia

U = puntaje promedio de Usabilidad

P = puntaje promedio de Portabilidad

M = puntaje promedio de Mantenibilidad

Los resultados de la evaluación aplicando la norma ISO/IEC 9126, muestran que la puntuación obtenida en el sistema de gestión de acceso a los laboratorios remotos es de 4.74 sobre 5. Esta puntuación al ser interpretada con el uso de las escalas Likert, representa que el sistema es de buena calidad.

Herramienta de evaluación

La herramienta de evaluación automatizada que se seleccionó fue Lighthouse Developer Tool de Google, puesto que, es una herramienta de experiencia de página de código abierto que permite ayudar a los desarrolladores de sitios web mejoren la calidad de sus aplicaciones. Con esta herramienta se realizaron las auditorías de rendimiento, de accesibilidad, de buenas prácticas y del Search Engine Optimization (SEO), (**AnexoA**).



Ilustración 68: Resultados de auditoría realizada con Lighthouse

En la ilustración 67, se indica los valores que arrojó la auditoría realizada por esta herramienta al sistema de gestión de laboratorios remotos, en ella se muestra que el rendimiento y el SEO alcanzan los más altos valores de calidad. La evaluación del rendimiento evalúa cuales son los factores que afectan a la velocidad de carga de un sitio web. La herramienta considera al sistema como rápido ya que tiene buenos tiempos de respuesta con un puntaje de 98/100. El resultado de los niveles de accesibilidad que tiene el sistema, el cual según la auditoria cumple con los factores de accesibilidad con un puntaje de 91/100, a su vez que brinda oportunidades de mejorar en esta métrica.

En cambio, en la auditoría de buenas prácticas se analizan aspectos generales y otros relacionados con la confianza y seguridad, como el uso de HTTPS uno de los factores que provocaron un puntaje de 83/100 puesto que, al implementar el sistema en un ambiente de laboratorio, el dominio asignado no cuenta con certificados SSL ni un puerto para salida al internet disponible para su implementación. En cuanto al puntaje de posicionamiento, el sistema tuvo un excelente puntaje en el nivel de visibilidad de la aplicación en diferentes buscadores.

4.2.4. Latencia de comunicación

Para medir el rendimiento de los tiempos de transmisión en la comunicación entre la aplicación con los dispositivos electrónicos de los laboratorios remotos. Para ello, se realizaron capturas de tiempos en el tráfico de tramas de datos con el uso de un cliente simulador de envío y recepción de datos a través de MQTT y WebSockets.

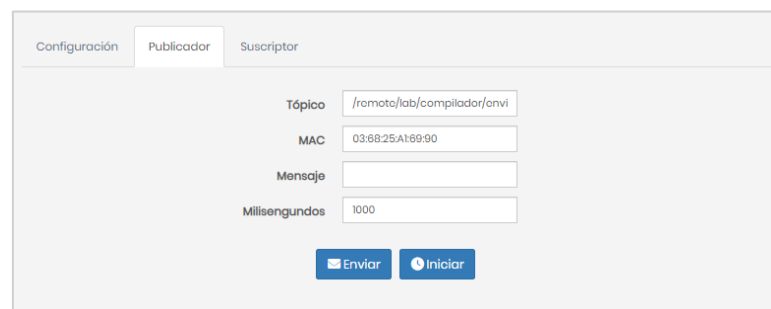


Ilustración 69: Simulador de pruebas de transmisión de datos

Envío de tramas de datos

Se realizaron las mediciones de tiempo correspondientes al envío de tramas de datos desde la aplicación hasta el dispositivo cliente de laboratorio remoto. La captura de datos se realizó en un escenario donde un cliente envía un dato hacia el servidor, y este le envía ese dato al simulador, como se muestra en la ilustración 69.

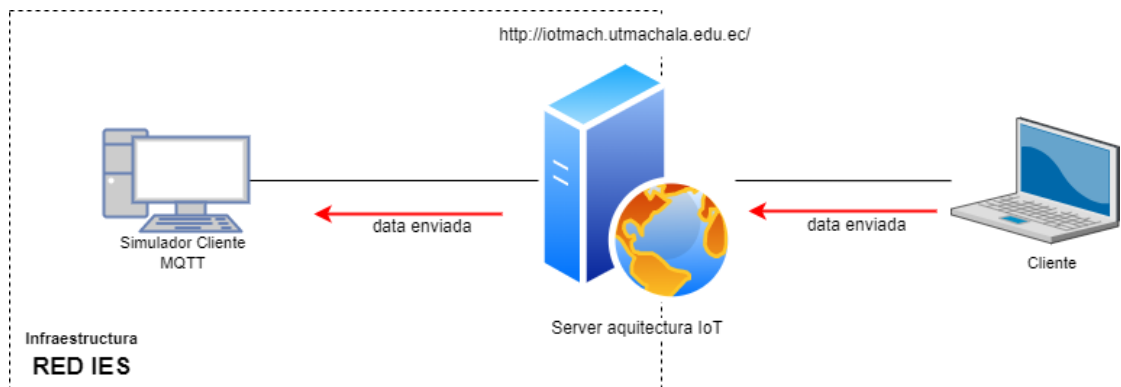


Ilustración 70: Proceso de envío de tramas de datos

Para realizar esta prueba de latencia en la transmisión de datos, se tomó la captura de 32 tramas enviadas, las mismas que dieron un promedio de 587.50 ms, como se observa en la tabla 18, donde se muestra una estadística descriptiva de las tramas de datos enviadas.

Parámetros	Tiempos (ms)
Media aritmética	587.50
Mediana	504
Menor valor	197
Mayor valor	1434
Primer cuartil	368
Tercer cuartil	799
Desviación estándar	270

Tabla 18: Estadística descriptiva del envío de tramas

Los datos analizados se pueden apreciar en la ilustración 70, en donde se verificó con la ayuda de un diagrama de cajas que, en el rango de datos no existe una variabilidad atípica.

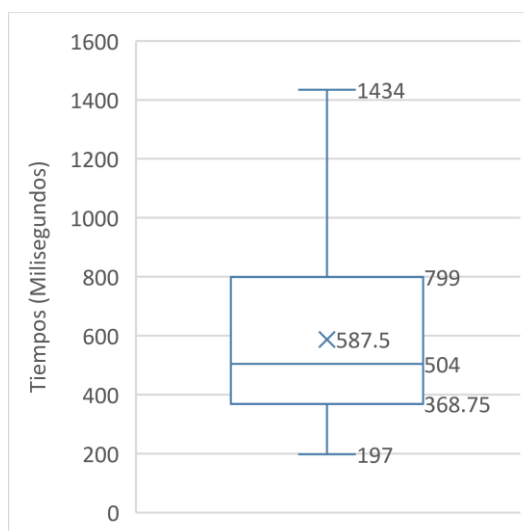


Ilustración 71: Diagrama de cajas de los datos de tramas enviadas

La frecuencia de los datos se puede apreciar con la ayuda de un histograma, como en la ilustración 71, se puede notar que la mayoría de datos se encuentra en un rango de 197ms y 444ms, este tiempo es menor a un segundo en las tramas de envío analizadas.

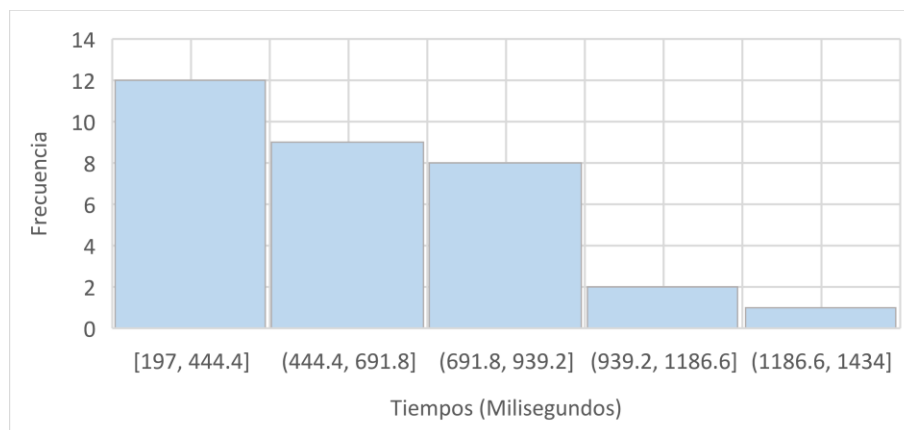


Ilustración 72: Histograma de frecuencias de datos de envío de tramas

Recepción de tramas de datos

En cuanto a la recepción de tramas de datos, se realizaron las mediciones de tiempo de la transmisión originada desde el dispositivo electrónico del laboratorio remoto hasta la aplicación web. La captura de datos se realizó en un escenario donde el simulador envía un dato hacia el servidor, y este le envía ese dato al cliente, como se muestra en la ilustración 72.

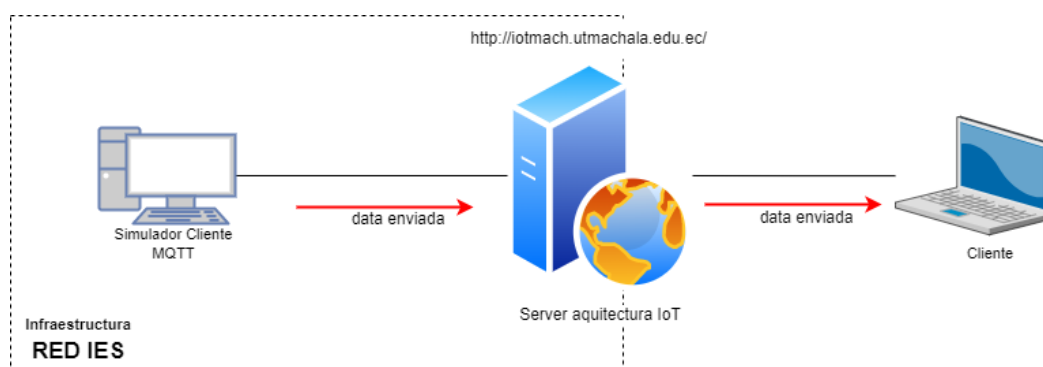


Ilustración 73: Proceso de envío de tramas de datos

Para medir el rendimiento de transmisión se tomó la captura de 39 tramas con un promedio de 130.77ms, como se observa en la tabla 19, donde se muestra una estadística descriptiva de las tramas de datos receptados.

Parámetros	Tiempos (ms)
Media aritmética	130.77
Mediana	105
Menor valor	30
Mayor valor	331
Primer cuartil	61.00
Tercer cuartil	191
Desviación estándar	84.38

Tabla 19: Estadística descriptiva del envío de tramas

En la ilustración 73, se pueden apreciar con la ayuda de un diagrama de cajas, los datos analizados dentro del rango de los datos no existen una variabilidad atípica.

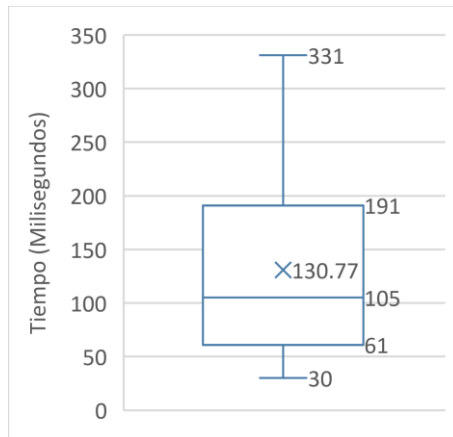


Ilustración 74: Diagrama de cajas de los datos de recepción de tramas

La frecuencia de los datos de recepción de tramas se puede apreciar con la ayuda de un histograma, como en la ilustración 74, en el cual se puede notar que la mayoría de datos se encuentra en un rango de 30ms y 105ms, lo que indica que fue un rango muy pequeño de tiempo para la transmisión de información al igual que las tramas enviadas.

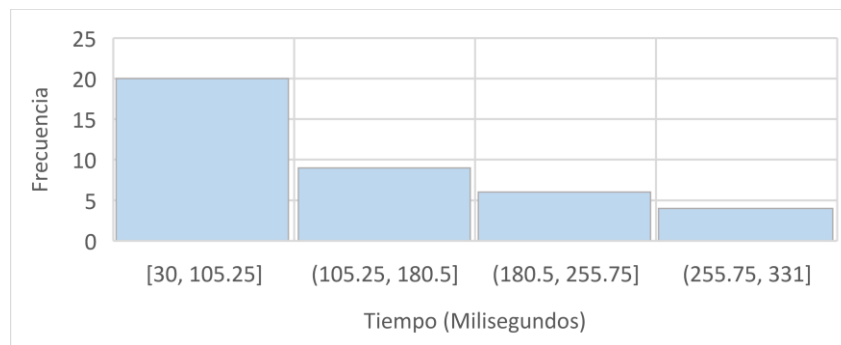


Ilustración 75: Histograma de frecuencias de datos de recepción de tramas

Tanto en el envío como la recepción de tramas de datos se ejecutan en tiempos menores a un segundo por lo que la comunicación entre el sistema y los dispositivos electrónicos es instantánea. La transmisión de los datos en una plataforma IoT es la parte fundamental en la comunicación entre los dispositivos y una aplicación.

La utilización de un sistema de comunicación permite el registro y descubrimiento de servicios, enrutamiento, equilibrio de carga y gestión de configuración distribuida [106]. Lo que llevó a la implementación de MQTT como protocolo de comunicación entre los microservicios y el hardware, y WebSockets para comunicar los microservicios con el MQTT. En el trabajo de [107] el uso de protocolos como MQTT reduce el tiempo de procesamiento de mensajes, a diferencia que en esta investigación se demostró en los tiempos de envío y recepción de tramas de datos, los tiempos de respuesta fueron menores a un segundo para cada proceso de envío/recepción de datos lo que indica que los tiempos de respuesta del sistema son rápidos.

4.2.5. Concurrencia de acceso a laboratorios

Dentro del sistema de gestión de acceso a los laboratorios remotos, los usuarios acceden a los recursos asignados a una asignatura. En la aplicación, se gestionan las restricciones de acceso a los estudiantes a través de los módulos Cursos, Laboratorio y Agendamiento. En el módulo Cursos, se listan los laboratorios que dispone la asignatura, el módulo Laboratorio gestiona el ingreso al recurso mediante una sala de espera para ver la disponibilidad del laboratorio remoto dónde se va a ingresar. Según se haya reservado un laboratorio, mediante el módulo de Agendamiento, los usuarios ingresan a los laboratorios desde uno hasta cuatro estudiantes simultáneamente para realizar las prácticas de la asignatura.

Fecha	Hora Inicio	Hora Finalización	Duración	Laboratorio	Usuario	Sesión
2022-08-13	21:02:40.489	21:32:06.499	0:29:26	ESP32 + per entrada/salida	María Cristina Cabrera Fajardo	174
2022-08-12	23:32:41.380	23:32:41.855	0:00:00	ESP32 + per entrada/salida	María Cristina Cabrera Fajardo	173
2022-08-12	23:08:21.480	23:27:55.793	0:19:34	ESP32 + per entrada/salida	María Cristina Cabrera Fajardo	171
2022-08-10	23:06:14.398	23:07:50.306	0:01:35	lab Prueba 1	Super Administrador	170
2022-08-10	23:05:30.961	23:07:50.306	0:02:19	lab Prueba 1	Chuchuca Cuenca Andres Vicente	170
2022-08-10	23:05:09.951	23:07:50.306	0:02:40	lab Prueba 1	Anna Valentina Alvarado Flores	170
2022-08-10	23:05:08.065	23:07:50.306	0:02:42	lab Prueba 1	Julissa Fernanda Torres Jaramillo	170
2022-08-10	23:05:05.653	23:07:50.306	0:02:44	lab Prueba 1	Helen Nahomy Alvarez Gaona	170
2022-08-10	23:05:05.158	23:07:50.306	0:02:45	lab Prueba 1	Sandra Yessenia Torres Troya	170
2022-08-10	23:05:03.520	23:07:50.306	0:02:46	lab Prueba 1	Thalia Ariana Trejo Perea	170

Ilustración 76: Historial de acceso a laboratorios proporcionado con el sistema

En cada sesión de prácticas de los laboratorios se genera un historial de acceso, en donde se puede observar fechas, duración de las sesiones y los usuarios que participan de las mismas, como se visualiza en la ilustración 76. Con estos datos se demuestra la accesibilidad y la optimización del uso de los laboratorios que se realiza a los usuarios.

Accesos de los Usuarios a los recursos

Los estudiantes tienen acceso a la cantidad de laboratorios remotos que se encuentren asignados al curso en el que estén matriculados. A su vez, accederán a los laboratorios remotos las veces en las que estos se encuentren disponibles para su uso. En estas pruebas se utilizó la base de datos de los estudiantes de la asignatura de Internet de las Cosas, en las cuales, todos los usuarios al menos una vez accedieron a un escenario de prácticas.

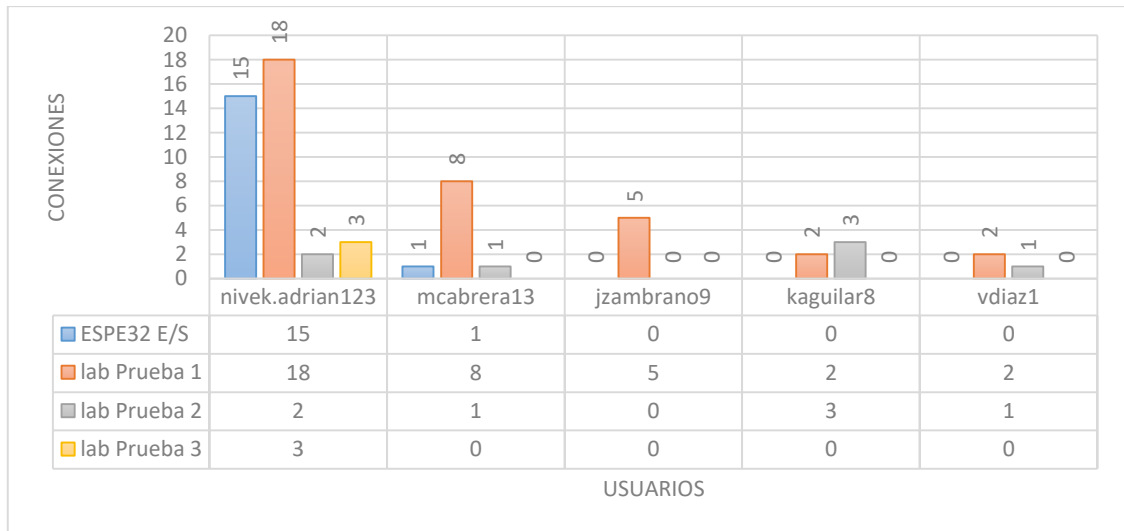


Ilustración 77: Usuarios con más conexiones en laboratorios

El sistema registra las veces que un usuario ha accedido a un escenario de práctica de laboratorio, esto se puede ver reflejado en la ilustración 77, en donde se observa que los usuarios con mayores conexiones hechas en los cuatro laboratorios registrados en el sistema, en esta ilustración indica la cantidad de veces que un usuario accede al recurso. De esta manera se determina que un usuario puede acceder las veces que un laboratorio esté disponible por un tiempo limitado, lo que permitió a un usuario realizar la práctica establecida en un laboratorio remoto las veces que se encuentre disponible el recurso.

Utilización de laboratorios

Como se observa en la ilustración 77, el historial de sesiones permite conocer cuál ha sido el laboratorio que los usuarios han ingresado. En esta ilustración indica que, el laboratorio “lab Prueba 1” ha sido utilizado por 52 estudiantes. Lo que demuestra la accesibilidad que tienen los estudiantes hacia los laboratorios disponibles en el sistema.

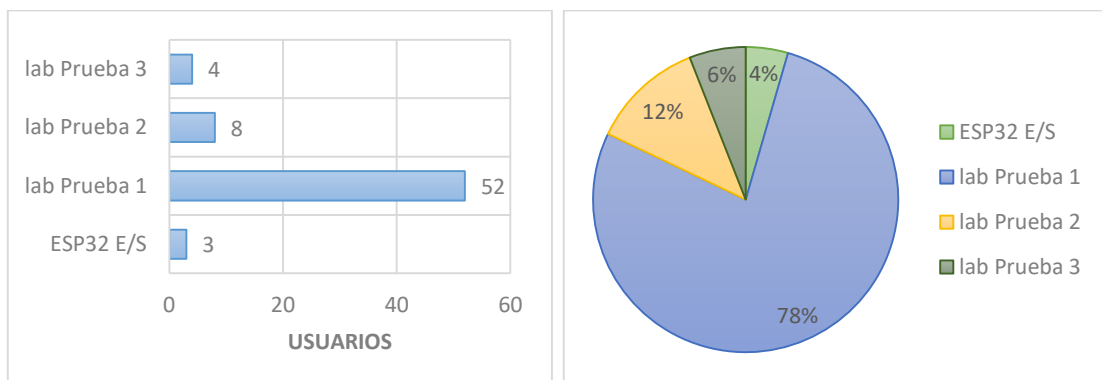


Ilustración 78: Utilización de laboratorios

Con estos datos se demuestra que un usuario puede acceder a los escenarios de prácticas de los laboratorios remotos que disponga un curso, mientras que, un laboratorio permite

la conexión de muchos usuarios para su utilización. Esto hace que se optimice la utilización de los laboratorios por parte de los estudiantes, lo que permite a los docentes tener herramientas adicionales para la realización de prácticas académicas en un ambiente controlado.

4.2.6. Rendimiento

En una sesión de prácticas de laboratorios remotos, pueden conectarse desde un usuario hasta cuatro al mismo tiempo, lo que permite utilizar un laboratorio por una mayor cantidad de usuarios. Como se observa en la tabla 20, el registro de sesiones de los laboratorios remotos indica que un laboratorio es albergado hasta por cuatro usuarios de manera simultánea en una sesión.

Sesión	Laboratorio	Conexiones
170	lab Prueba 1	38
168	lab Prueba 1	21
137	lab Prueba 2	4
162	lab Prueba 1	3
154	lab Prueba 1	3
135	lab Prueba 2	2
121	lab Prueba 1	2

Tabla 20: Registro de sesiones de los laboratorios remotos

Consumo de recursos

Para la prueba de consumo de recursos, se realizaron pruebas con una mayor cantidad de usuarios máximos permitidos en una sesión con la finalidad de observar el consumo de los recursos hardware del servidor. La prueba de rendimiento se realizó en relación de la cantidad de estudiantes que puede tener una asignatura. En una sesión de práctica de laboratorio con 21 estudiantes conectados de manera simultánea, en la que el consumo en memoria RAM fue de 73.5 MB en promedio, como se observa en la ilustración 79.

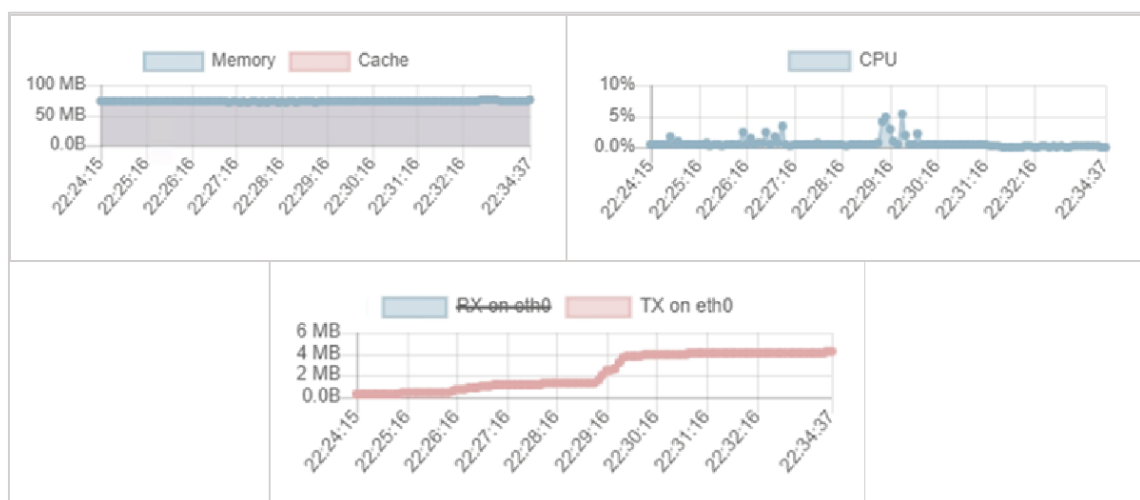


Ilustración 79: Consumo de recursos con 21 usuarios conectados simultáneamente

El uso del CPU con 21 usuarios conectados simultáneamente alcanzó un pico de 5% del procesador de la máquina virtual en que se alojó el microservicio de laboratorio. En cuanto al uso de la red, se observa el ancho de banda consumido por los paquetes transmitidos por la interfaz TX eth0 fue de 3.7MB sin tener datos de paquetes recibidos por la interfaz RX eth0.

Luego, en una sesión de práctica de laboratorio con 38 estudiantes conectados de manera simultánea, el consumo del microservicio en memoria RAM fue de 77.5 MB en promedio, como se observa en la ilustración 80.



Ilustración 80: Consumo de recursos con 38 usuarios conectados simultáneamente

Respecto al uso del CPU con 38 usuarios conectados simultáneamente alcanzó un pico de 12% del consumo del procesador de la máquina virtual del microservicio. En cuanto el uso de red, el ancho de banda consumido por los paquetes transmitidos por la interfaz TX eth0 fue de 5.9MB y los paquetes recibidos por la interfaz RX eth0 fue de 3.2MB. En los laboratorios remotos se puede utilizar varios tipos de tecnologías para comunicar una aplicación cliente con un laboratorio remoto.

Al analizar estas mediciones se puede entender que el rendimiento del sistema dependerá de las características del servidor que se utiliza para el despliegue del proyecto, puesto que, a pesar de que a medida que se fue aumentando la cantidad de usuarios, el aumento en el consumo de recursos fue bajo. Esto indica que el rendimiento del sistema se ve limitado a los recursos que se disponga para su ejecución. En el trabajo de [108] que utilizó Matlab como una arquitectura de cliente/servidor, a diferencia de este trabajo, esta investigación utiliza tecnologías de bajo consumo de recursos, tales como WebSockets y MQTT lo que se vio reflejado tanto en el rendimiento del sistema como en los tiempos de transmisión de información

CONCLUSIONES

La investigación concluyó exitosamente con todos los objetivos que se plantearon, en la que se comprobó la hipótesis de que un sistema para la gestión de laboratorios remotos incrementa el acceso a recursos para la realización de prácticas académicas de la asignatura de Internet de las Cosas.

El sistema permite la gestión del uso de las herramientas con las que cuenta la asignatura en un ambiente controlado, también permite limitar a los estudiantes el acceso a los laboratorios en horario sincrónico o asincrónico, lo que incrementa el acceso a los recursos asignatura de Internet de las Cosas. Se debe tener en cuenta que, el sistema puede ser empleado para otras asignaturas, de esta manera los docentes tendrán a su disposición más herramientas para la realización de actividades académicas en sus clases.

La investigación bibliográfica a través de la aplicación de un RSL, permitió conocer cuáles eran las tendencias de desarrollo de los laboratorios remotos y la arquitectura que se utilizaba, esta revisión indicó que el enfoque brindado a las investigaciones sobre laboratorios remotos era la creación y diseño de hardware, por lo que fue realizar una investigación con un enfoque en el desarrollo de software y la implementación de arquitecturas IoT ha sido necesaria.

Usar microservicios facilita la escalabilidad de la arquitectura, ya que permite integrar nuevos microservicios en el sistema con diferentes funcionalidades sin afectar la integridad del proyecto. Al implementar este tipo de arquitecturas, se logra una distribución de recursos que permite alcanzar el mejor rendimiento posible del uso del hardware utilizado como servidor.

El enfoque de DevOps como metodología ágil permitió que el desarrollo del sistema se realice de manera organizada y en lo posible automatizada debido a la integración y despliegue continuo, provee un entorno de trabajo que admite el uso de herramientas que facilitan la verificación de fallos continuamente mediante la realización de pruebas en cada una de sus fases, lo que permitió la detección de errores que perjudiquen el desarrollo del proyecto.

El sistema implementado fue puesto a prueba con cuatro prácticas de laboratorio y 38 usuarios remotos. En esta prueba se demostró que el sistema realiza un bajo consumo de recursos de ancho de banda, memoria RAM y CPU en el hardware del servidor, con una latencia menor a un segundo para las interacciones del sistema con dichos usuarios.

RECOMENDACIONES

Para implementar una arquitectura IoT se debe tener en cuenta los recursos con los que se cuenta, ya sea que se vaya a implementar en la nube o en un servidor propio, ya que organizar los servicios y herramientas de manera equilibrada, permitirá acoger la mayor cantidad de recursos posibles para que el rendimiento del sistema no se vea afectado.

Cuando se utiliza un enfoque metodológico DevOps para el desarrollo de un sistema, debería realizarse un estudio de las herramientas de automatización existentes, esto permitirá desarrollar las etapas del ciclo de vida que la metodología propone de una manera mejor controlada y organizada. Estas herramientas facilitan los procesos de integración y despliegue continuo de un proyecto.

La implementación de microservicios facilita la escalabilidad de un sistema, por lo tanto, es recomendable que los próximos laboratorios remotos que vayan a integrarse sean desarrollados junto a su escenario de prácticas basados en microservicios, lo que permite que tanto la configuración y adaptabilidad en el sistema no afecte a la integridad del funcionamiento del proyecto.

Trabajos futuros

El desarrollo de esta investigación abre un camino a la realización de varios trabajos futuros que por cuestiones de tiempo y el alcance de esta investigación no se han desarrollado aún para este sistema.

- En el desarrollo de este proyecto, en el frontend se utilizó el framework Bootstrap y JQuery, lo que vuelve limitado al sistema en niveles de mantenimiento y agilidad en el desarrollo de interfaces, lo que hace necesario la adopción de tecnologías como React.js, Vue.js o Angular.js que ofrecen una estructura y organización del código predeterminada, reutilización del código evitando la duplicidad de código y el uso de buenas prácticas de desarrollo con el uso de patrones de diseño.
- Implementar mecanismos de trabajo simultáneo en el editor de código, ayudaría una optimización de trabajo comunitario para gestionar lo que los usuarios escriben al mismo tiempo.
- El uso de sistemas de almacenamientos seguros es muy importante para arquitecturas IoT como lo proponen en el trabajo de [109] en el que aplican blockchain en una plataforma IoT. Por lo que implementar este tipo de tecnologías proveen alta seguridad a un sistema basado en IoT.

BIBLIOGRAFÍA

- [1] N. CEPAL and UNESCO, *La educación en tiempos de la pandemia de COVID-19*. REPOSITORIO DIGITAL - Comisión Económica para América Latina y el Caribe, 2020.
- [2] A. Fernández-Pacheco, S. Martin, and M. Castro, “Implementation of an arduino remote laboratory with raspberry pi,” *IEEE Glob. Eng. Educ. Conf. EDUCON*, vol. April-2019, pp. 1415–1418, 2019.
- [3] S. Martin, A. Gordillo, E. Sancristobal, M. Castro, and J. Quemada, “Analysis of management systems for virtual and remote labs,” in *2020 IEEE Global Engineering Education Conference (EDUCON)*, 2020, pp. 1632–1636.
- [4] M. MOUSSA, A. BENACHENHOU, A. ADDA-BENATTIA, and A. BOUMEHDI, “A New Design and Evaluation for Sharing Analog Electronic Labs,” in *2021 International e-Engineering Education Services Conference (e-Engineering)*, 2021, pp. 111–114.
- [5] T. Nenov, B. Evstatiev, and S. Kadirova, “Conceptual Model of a Remote Laboratory for Investigation of DC Motors,” in *2021 17th Conference on Electrical Machines, Drives and Power Systems (ELMA)*, 2021, no. July, pp. 1–4.
- [6] F. Guc, J. Viola, and Y. Chen, “Digital Twins Enabled Remote Laboratory Learning Experience for Mechatronics Education,” in *2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI)*, 2021, pp. 242–245.
- [7] N. Fujii and N. Koike, “IoT Remote Group Experiments in the Cyber Laboratory: A FPGA-based Remote Laboratory in the Hybrid Cloud,” in *2017 International Conference on Cyberworlds (CW)*, 2017, vol. 2017-Janua, pp. 162–165.
- [8] K. Tokarz *et al.*, “Internet of Things Network Infrastructure for The Educational Purpose,” in *2020 IEEE Frontiers in Education Conference (FIE)*, 2020, pp. 1–9.
- [9] M. A. Zaman, L. T. Neustock, and L. Hesselink, “iLabs as an online laboratory platform: A case study at Stanford University during the COVID-19 Pandemic,” in *2021 IEEE Global Engineering Education Conference (EDUCON)*, 2021, vol. 2021-April, no. April, pp. 1615–1623.
- [10] H. Guerra, A. M. Garcia, L. M. Gomes, and A. Cardoso, “An IoT remote lab for seismic monitoring in a programming course,” in *2017 4th Experiment@International Conference (exp.at'17)*, 2017, pp. 129–130.
- [11] P. Vitliemov, D. Bratanov, and M. Marinov, “An Approach to Use Virtual and Remote Labs in Mechatronics Education Based on Cloud Services,” in *2020 7th International Conference on Energy Efficiency and Agricultural Engineering (EE&AE)*, 2020, no. November, pp. 1–4.
- [12] S. Guan, W. Hu, and H. Zhou, “Real-Time Data Transmission Method Based on Websocket Protocol for Networked Control System Laboratory,” in *2019 Chinese Control Conference (CCC)*, 2019, pp. 5339–5344.
- [13] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, “Systematic literature reviews in software engineering – A systematic literature review,” *Inf. Softw. Technol.*, vol. 51, no. 1, pp. 7–15, Jan. 2009.

- [14] L. Rosado and J. Herreros, “Nuevas aportaciones didácticas de los laboratorios virtuales y remotos en la enseñanza de la Física,” *Recent Res. Dev. Learn. ...*, pp. 1–5, 2005.
- [15] R. M V, P. G K, and P. K R, “Design and Implementation of IoT Based Remote Laboratory for Sensor Experiments,” *Int. J. Interact. Mob. Technol.*, vol. 14, no. 09, p. 227, Jun. 2020.
- [16] K. D. Taylor, J. W. Honchell, and W. E. DeWitt, “Distance learning in courses with a laboratory,” in *Technology-Based Re-Engineering Engineering Education Proceedings of Frontiers in Education FIE’96 26th Annual Conference*, 1996, vol. 1, pp. 44–46.
- [17] P. Baizan *et al.*, “IoT Remote Laboratory Based on ARM Device Extension of VISIR Remote Laboratories to Include IoT Support,” in *Lecture Notes in Networks and Systems*, vol. 80, 2020, pp. 269–279.
- [18] J. A. Vento, “Application of LabVIEW in higher education laboratories,” in *Proceedings Frontiers in Education Conference*, 1988, pp. 444–447.
- [19] T. Kikuchi, T. Kenjo, and S. Fukuda, “Remote laboratory for a brushless DC motor,” *IEEE Trans. Educ.*, vol. 44, no. 2, p. 12 pp., May 2001.
- [20] M. Billaud *et al.*, “Real measures, virtual instruments,” in *Proceedings of the Fourth IEEE International Caracas Conference on Devices, Circuits and Systems (Cat. No.02TH8611)*, 2002, pp. I023-1-I023-4.
- [21] R. M. Fernandes, L. G. B. Rolim, and W. I. Suemitsu, “Design and implementation of a power-electronic remote-laboratory (ELEPOT-rLab),” in *2003 IEEE International Symposium on Industrial Electronics (Cat. No.03TH8692)*, 2003, vol. 1, pp. 307–311.
- [22] C. Hopp, S. Stoll, and U. Konigorski, “Remote control design and implementation using the Internet,” in *Proceedings of the 5th Biannual World Automation Congress*, 2002, pp. 481–486.
- [23] I. Gustavsson, “Remote laboratory experiments in electrical engineering education,” in *Proceedings of the Fourth IEEE International Caracas Conference on Devices, Circuits and Systems (Cat. No.02TH8611)*, 2002, pp. I025-1-I025-5.
- [24] I. Gustavsson, “Remote Laboratory for Electrical Experiments,” in *Lab on the Web*, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2005, pp. 175–219.
- [25] J. S. Pastor, I. Gonzalez, J. Lopez, F. Gomez-Arribas, and J. Martinez, “A remote laboratory for debugging FPGA-based microprocessor prototypes,” in *IEEE International Conference on Advanced Learning Technologies, 2004. Proceedings.*, 2004, pp. 86–90.
- [26] D. Lopez-de-Ipina, J. Garcia-Zubia, and P. Orduna, “Remote Control of Web 2.0-Enabled Laboratories from Mobile Devices,” in *2006 Second IEEE International Conference on e-Science and Grid Computing (e-Science’06)*, 2006, pp. 123–123.
- [27] J. Garcia-Zubia, D. López-de-Ipiña, and P. Orduña, “Mobile devices and remote labs in engineering education,” *Proc. - 8th IEEE Int. Conf. Adv. Learn. Technol. ICAIT 2008*, pp. 620–622, 2008.

- [28] O. GomisBellmunt, D. MontesinosMiracle, S. GalceranArellano, A. Sumper, and A. SudriaAndreu, "A Distance PLC Programming Course Employing a Remote Laboratory Based on a Flexible Manufacturing Cell," *IEEE Trans. Educ.*, vol. 49, no. 2, pp. 278–284, May 2006.
- [29] F. Davoli, G. Spanospano, S. Vignola, and S. Zappatore, "LABNET: Towards Remote Laboratories With Unified Access," *IEEE Trans. Instrum. Meas.*, vol. 55, no. 5, pp. 1551–1558, Oct. 2006.
- [30] F. Garcia-Loro, E. Sancristobal, G. Diaz, and M. Castro, "Remote Laboratories Integration into Electronics Engineer Curricula," in *2018 IEEE World Engineering Education Conference (EDUNINE)*, 2018, pp. 1–6.
- [31] S. Martin, A. Fernandez-Pacheco, J. A. Ruiperez-Valiente, G. Carro, and M. Castro, "Remote Experimentation Through Arduino-Based Remote Laboratories," *IEEE Rev. Iberoam. Tecnol. del Aprendiz.*, vol. 16, no. 2, pp. 180–186, May 2021.
- [32] M. Gericota, P. Ferreira, A. Fidalgo, G. Andrieu, and C. Dalmay, "e-Engineering: Remote Labs in an Electronics and Optics e-Learning for Embedded Systems Course," in *2019 5th Experiment International Conference (exp.at '19)*, 2019, pp. 405–409.
- [33] A. Hariton, C. Zet, E. Vremera, and C. Fosalau, "Remote Laboratory – Study of Digital Oscilloscope," in *2020 International Conference and Exposition on Electrical And Power Engineering (EPE)*, 2020, pp. 495–500.
- [34] T. Klinger, C. Kreiter, A. Pester, and C. Madritsch, "Low-cost Remote Laboratory Concept based on NI myDAQ and NI ELVIS for Electronic Engineering Education," in *2020 IEEE Global Engineering Education Conference (EDUCON)*, 2020, pp. 106–109.
- [35] J. Alvarez, G. Diaz, and M. Macias, "Programming logical controllers using remote labs and virtual reality," in *2019 IEEE International Conference on Engineering Veracruz (ICEV)*, 2019, pp. 1–4.
- [36] A. Liccardo *et al.*, "Augmented Reality Laboratory for Instrumentation and Measurements Online Course," in *2021 IEEE 6th International Forum on Research and Technology for Society and Industry (RTSI)*, 2021, pp. 86–90.
- [37] P. Vitliemov, D. Bratanov, and M. Marinov, "An Approach to Use Virtual and Remote Labs in Mechatronics Education Based on Cloud Services," in *2020 7th International Conference on Energy Efficiency and Agricultural Engineering (EE&AE)*, 2020, pp. 1–4.
- [38] G. S. Machado, Y. M. L. R. Silva, and V. F. de Lucena, "A Tool for the Automatic Selection of Mechatronics Remote Laboratories based on their Actual Effective Costs," in *2018 IEEE Frontiers in Education Conference (FIE)*, 2018, pp. 1–5.
- [39] N. M. Mamani, F. J. Garcia-Penalvo, M. A. Conde, and J. Goncalves, "A systematic mapping about simulators and remote laboratories using hardware in the loop and robotic: Developing STEM/STEAM skills in pre-university education," in *2021 International Symposium on Computers in Education (SIIE)*, 2021, pp. 1–6.

- [40] C. A. S. do Rosario and J. A. das Rosas, "Affordable Remote Platform for Robotics Learning in Engineering Courses," in *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*, 2019, pp. 6789–6794.
- [41] L. Favario, "Remote Programming Environments: the Robotic Laboratory Case," in *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, 2018, pp. 820–824.
- [42] Y. Gorbounov and H. Chen, "Low-Cost Remote Laboratory for Cyber-Physical Experiments," in *2021 29th National Conference with International Participation (TELECOM)*, 2021, pp. 149–152.
- [43] V. M. Cvjetković and U. Stanković, "Arduino Based Physics and Engineering Remote Laboratory," 2017, pp. 560–574.
- [44] A. Maiti, A. Raza, and B. H. Kang, "Teaching Embedded Systems and Internet-of-Things Supported by Multipurpose Multiobjective Remote Laboratories," *IEEE Trans. Learn. Technol.*, vol. 14, no. 4, pp. 526–539, Aug. 2021.
- [45] M. Mehrtash, K. Ghalkhani, and I. Singh, "IoT-based Experiential E-Learning Platform (EELP) for Online and Blended Courses," in *2021 International Symposium on Educational Technology (ISET)*, 2021, pp. 252–255.
- [46] W. C. Karunianto and A. H. Saputro, "Design and implementation remote laboratory based on Internet of Things: Study case in diffraction grating experiment," in *2017 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, 2017, pp. 143–146.
- [47] D. Pirrone, C. Fornaro, and D. Assante, "Open-source multi-purpose remote laboratory for IoT education," in *2021 IEEE Global Engineering Education Conference (EDUCON)*, 2021, pp. 1462–1468.
- [48] T. D. Drysdale and N. S. J. Braithwaite, "An internet of laboratory things," in *2017 4th Experiment@International Conference (exp.at'17)*, 2017, pp. 236–240.
- [49] P. A. Harsha Vardhini, S. S. Prasad, and S. N. Korra, "Medicine Allotment for COVID-19 Patients by Statistical Data Analysis," in *2021 International Conference on Emerging Smart Computing and Informatics (ESCI)*, 2021, pp. 665–669.
- [50] S. Nabi and V. P. Mishra, "Analysis and Impact of COVID-19 on Economy and Organization," in *2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, 2021, pp. 219–224.
- [51] E. Hitchings and M. Maclean, "Unprecedented times: some thoughts on the consequences of the COVID-19 pandemic from a family and social welfare law perspective," *J. Soc. Welf. Fam. Law*, vol. 42, no. 3, pp. 277–280, Jul. 2020.
- [52] T. Crick, C. Knight, R. Watermeyer, and J. Goodall, "The International Impact of COVID-19 and 'Emergency Remote Teaching' on Computer Science Education Practitioners," in *2021 IEEE Global Engineering Education Conference (EDUCON)*, 2021, vol. 2021-April, no. April, pp. 1048–1055.
- [53] P. C. Nicolete, F. Herpich, E. T. D. O. Junior, L. M. R. Tarouco, and J. B. Da Silva, "Analysis of student motivation in the use of a Physics Augmented Remote Lab during the Covid-19 pandemic," in *2021 IEEE Global Engineering Education Conference (EDUCON)*, 2021, vol. 2021-April, no. April, pp. 1040–

1047.

- [54] D. Khanna and A. Prasad, “Problems Faced by Students and Teachers During Online Education Due to COVID-19 and How to Resolve Them,” in *2020 6th International Conference on Education and Technology (ICET)*, 2020, pp. 32–35.
- [55] N. Sharma, M. Shamkuwar, and I. Singh, “The History, Present and Future with IoT,” in *Intelligent Systems Reference Library*, vol. 154, 2019, pp. 27–51.
- [56] P. Suresh, J. V. Daniel, V. Parthasarathy, and R. H. Aswathy, “A state of the art review on the Internet of Things (IoT) history, technology and fields of deployment,” in *2014 International Conference on Science Engineering and Management Research (ICSEMR)*, 2014, pp. 1–8.
- [57] M. Weyrich and C. Ebert, “Reference Architectures for the Internet of Things,” *IEEE Softw.*, vol. 33, no. 1, pp. 112–116, Jan. 2016.
- [58] M. Perales, L. Pedraza, and P. Moreno-Ger, “Work-In-Progress: Improving Online Higher Education with Virtual and Remote Labs,” in *2019 IEEE Global Engineering Education Conference (EDUCON)*, 2019, vol. April-2019, pp. 1136–1139.
- [59] F. Garcia-Loro, E. Sancristobal, G. Diaz, and M. Castro, “Remote laboratories integration into electronics engineer curricula,” *EDUNINE 2018 - 2nd IEEE World Eng. Educ. Conf. Role Prof. Assoc. Contemp. Eng. Careers, Proc.*, 2018.
- [60] H. Considine, A. Nafalski, and Z. Nedic, “Remote Laboratory Environments for Smart E-Learning,” in *Smart Innovation, Systems and Technologies*, vol. 75, 2018, pp. 82–91.
- [61] H. Baez and O. E. Mojica, “Use of remote laboratories in the teaching of technological degree,” in *2018 IEEE Global Engineering Education Conference (EDUCON)*, 2018, vol. 2018-April, pp. 1957–1961.
- [62] J. Broisin, R. Venant, and P. Vidal, “Lab4CE: a Remote Laboratory for Computer Education,” *Int. J. Artif. Intell. Educ.*, vol. 27, no. 1, pp. 154–180, Mar. 2017.
- [63] V. J. Bhute, P. Inguva, U. Shah, and C. Brechtelsbauer, “Transforming traditional teaching laboratories for effective remote delivery—A review,” *Educ. Chem. Eng.*, vol. 35, pp. 96–104, Apr. 2021.
- [64] K. C. Narasimhamurthy, T. Nabi, S. Sinha, and S. N. Bhat, “Analysis of LC-tuned Oscillator in Remote Laboratory,” *Procedia Comput. Sci.*, vol. 172, no. 2019, pp. 34–42, 2020.
- [65] F. Alkhabbas, R. Spalazzese, and P. Davidsson, “Architecting Emergent Configurations in the Internet of Things,” in *2017 IEEE International Conference on Software Architecture (ICSA)*, 2017, pp. 221–224.
- [66] D. Hernandez-Rojas, B. Mazon-Olivo, J. Novillo-Vicuña, C. Escudero-Cascon, A. Pan-Bermudez, and G. Belduma-Vacacela, “IoT Android Gateway for Monitoring and Control a WSN,” 2018, pp. 18–32.
- [67] D. Hernández-Rojas, B. Mazon-Olivo, and C. Escudero, *Análisis de Datos Agropecuarios - Capítulo 3 Internet de las Cosas (IoT)*. UTMACH, 2018.
- [68] J. Novillo-Vicuña, D. Hernández Rojas, B. Mazón Olivo, J. Molina Ríos, and O. Cárdenas Villavicencio, *Arduino y el Internet de las cosas*. Editorial Científica

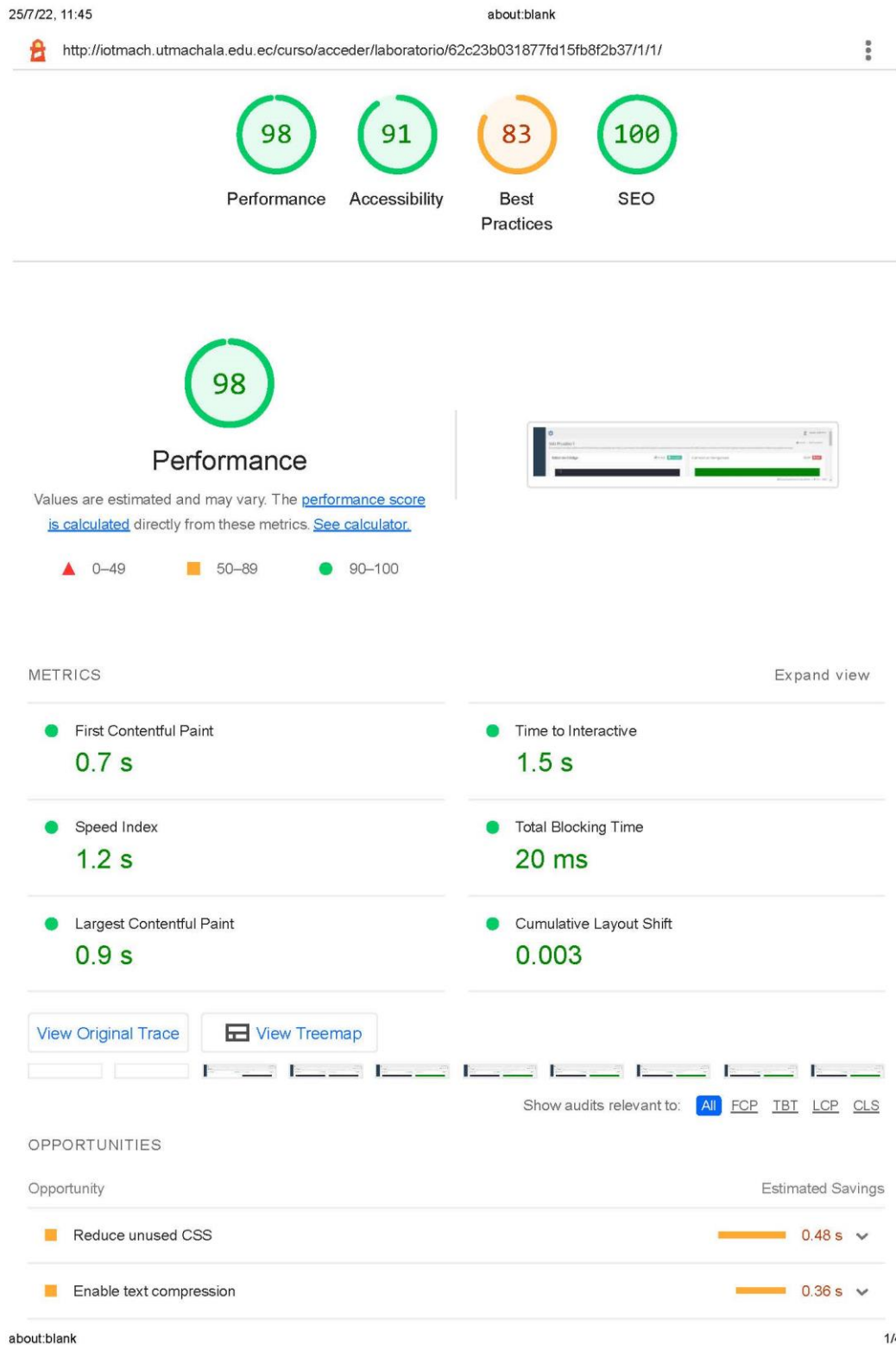
- 3Ciencias, 2018.
- [69] J. Berrú-Ayala, D. Hernandez-Rojas, P. Morocho-Díaz, J. Novillo-Vicuña, B. Mazon-Olivo, and A. Pan, “SCADA System Based on IoT for Intelligent Control of Banana Crop Irrigation,” 2020, pp. 243–256.
 - [70] C. Zhang, “Intelligent Internet of things service based on artificial intelligence technology,” in *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, 2021, no. Icbaie, pp. 731–734.
 - [71] D. Hernández-Rojas, T. Fernández-Caramés, P. Fraga-Lamas, and C. Escudero, “A Plug-and-Play Human-Centered Virtual TEDS Architecture for the Web of Things,” *Sensors*, vol. 18, no. 7, p. 2052, Jun. 2018.
 - [72] J. J. Cartuche Calva, D. L. Hernández Rojas, R. F. Morocho Román, and C. D. Radicelli García, “Seguridad IoT: Principales amenazas en una taxonomía de activos,” *HAMUT'AY*, vol. 7, no. 3, p. 51, Jan. 2021.
 - [73] K. Shafique, B. A. Khawaja, F. Sabir, S. Qazi, and M. Mustaqim, “Internet of things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT Scenarios,” *IEEE Access*, vol. 8, pp. 23022–23040, 2020.
 - [74] H. P. Sultana, “IoT Architecture,” 2017, pp. 1–18.
 - [75] H. Landaluce, L. Arjona, A. Perallos, F. Falcone, I. Angulo, and F. Muralter, “A Review of IoT Sensing Applications and Challenges Using RFID and Wireless Sensor Networks,” *Sensors*, vol. 20, no. 9, p. 2495, Apr. 2020.
 - [76] N. Niknejad, A. R. C. Hussin, and I. S. Amiri, “Literature Review of Service-Oriented Architecture (SOA) Adoption Researches and the Related Significant Factors,” 2019, pp. 9–41.
 - [77] S. A. Kumar and P. Ilango, “The Impact of Wireless Sensor Network in the Field of Precision Agriculture: A Review,” *Wirel. Pers. Commun.*, vol. 98, no. 1, pp. 685–698, Jan. 2018.
 - [78] A. Campoverde, D. Hernández, and B. Mazón, “Cloud computing con herramientas open-source para Internet de las cosas,” *Actas del III Congr. Ecuatoriano Tecnol. la Inf. y Comun. - TIC.EC 2015*, vol. 6, pp. 173–182, 2015.
 - [79] B. Mazon-Olivo, D. Hernández-Rojas, J. Maza-Salinas, and A. Pan, “Rules engine and complex event processor in the context of internet of things for precision agriculture,” *Comput. Electron. Agric.*, vol. 154, pp. 347–360, Nov. 2018.
 - [80] G. Kaur, P. Tomar, and P. Singh, “Design of Cloud-Based Green IoT Architecture for Smart Cities,” 2018, pp. 315–333.
 - [81] H. Atlam, R. Walters, and G. Wills, “Fog Computing and the Internet of Things: A Review,” *Big Data Cogn. Comput.*, vol. 2, no. 2, p. 10, Apr. 2018.
 - [82] D. Hernández-Rojas, T. Fernández-Caramés, P. Fraga-Lamas, and C. Escudero, “Design and Practical Evaluation of a Family of Lightweight Protocols for Heterogeneous Sensing through BLE Beacons in IoT Telemetry Applications,” *Sensors*, vol. 18, no. 2, p. 57, Dec. 2017.
 - [83] I. Machorro-Cano *et al.*, “An IoT-Based Architecture to Develop a Healthcare

- Smart Platform,” in *Communications in Computer and Information Science*, vol. 749, R. Valencia-García, K. Lagos-Ortiz, G. Alcaraz-Mármol, J. Del Cioppo, N. Vera-Lucio, and M. Bucaram-Leverone, Eds. Cham: Springer International Publishing, 2017, pp. 133–145.
- [84] H. Xu, W. Yu, D. Griffith, and N. Golmie, “A Survey on Industrial Internet of Things: A Cyber-Physical Systems Perspective,” *IEEE Access*, vol. 6, pp. 78238–78259, 2018.
- [85] J. Singh, Y. Bello, A. R. Hussein, A. Erbad, and A. Mohamed, “Hierarchical Security Paradigm for IoT Multiaccess Edge Computing,” *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5794–5805, Apr. 2021.
- [86] S. Kolhe, S. Nagpal, and J. Desai, “Security in Internet of Things,” in *Intelligent Systems Reference Library*, vol. 172, no. i, 2020, pp. 279–290.
- [87] P. P. Ray, “A survey on Internet of Things architectures,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 30, no. 3, pp. 291–319, Jul. 2018.
- [88] S. Trilles, A. González-Pérez, and J. Huerta, “An IoT Platform Based on Microservices and Serverless Paradigms for Smart Farming Purposes,” *Sensors*, vol. 20, no. 8, p. 2418, Apr. 2020.
- [89] O. Al-Debagy and P. Martinek, “A Comparative Review of Microservices and Monolithic Architectures,” in *2018 IEEE 18th International Symposium on Computational Intelligence and Informatics (CINTI)*, 2018, pp. 000149–000154.
- [90] S. Ali, M. G. Kibria, M. A. Jarwar, S. Kumar, and I. Chong, “Microservices model in WoO based IoT platform for depressive disorder assistance,” in *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, 2017, vol. 2017-Decem, pp. 864–866.
- [91] H. Vural, M. Koyuncu, and S. Guney, “A Systematic Literature Review on Microservices,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10409 LNCS, 2017, pp. 203–217.
- [92] D. Trihinas, A. Tryfonos, M. D. Dikaiakos, and G. Pallis, “DevOps as a Service: Pushing the Boundaries of Microservice Adoption,” *IEEE Internet Comput.*, vol. 22, no. 3, pp. 65–71, May 2018.
- [93] L. Chen, “Microservices: Architecting for Continuous Delivery and DevOps,” *Proc. - 2018 IEEE 15th Int. Conf. Softw. Archit. ICSA 2018*, pp. 39–46, 2018.
- [94] A. Alnafessah, A. U. Gias, R. Wang, L. Zhu, G. Casale, and A. Filieri, “Quality-Aware DevOps Research: Where Do We Stand?,” *IEEE Access*, vol. 9, pp. 44476–44489, 2021.
- [95] P. Agrawal and N. Rawat, “Devops, A New Approach To Cloud Development & Testing,” in *2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, 2019, pp. 1–4.
- [96] M. A. Lopez-Pena, J. Diaz, J. E. Perez, and H. Humanes, “DevOps for IoT Systems: Fast and Continuous Monitoring Feedback of System Availability,” *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10695–10707, Oct. 2020.
- [97] M. S. Khan, A. W. Khan, F. Khan, M. A. Khan, and T. K. Whangbo, “Critical Challenges to Adopt DevOps Culture in Software Organizations: A Systematic

- Review,” *IEEE Access*, vol. 10, pp. 14339–14349, 2022.
- [98] Y. Guo and L. Chen, “An Investigation on Online Learning for K12 in Rural Areas in China during COVID-19 Pandemic,” in *2020 Ninth International Conference of Educational Innovation through Technology (EITT)*, 2020, pp. 13–18.
- [99] L. M. Sánchez Ruiz, S. Moll-López, J. A. Moraño-Fernández, and N. Llobregat-Gómez, “B-Learning and Technology: Enablers for University Education Resilience. An Experience Case under COVID-19 in Spain,” *Sustainability*, vol. 13, no. 6, p. 3532, Mar. 2021.
- [100] I. Herrera, “¿Qué es la investigación explicativa?,” 2022. [Online]. Available: <https://www.questionpro.com/blog/es/investigacion-explicativa/>.
- [101] A. Aguilar, “¿Qué es la investigación descriptiva?,” 2022. [Online]. Available: <https://www.questionpro.com/blog/es/investigacion-descriptiva/>.
- [102] D. S. Foundation, “Django Documentation,” *User authentication in Django*, 2022. [Online]. Available: <https://docs.djangoproject.com/en/4.0/topics/auth/>.
- [103] L. S. Inc., “¿Cuáles son tus necesidades de creación de wireframes?,” 2022. [Online]. Available: <https://www.lucidchart.com/pages/es/que-es-un-wireframe-para-un-sitio-web>.
- [104] E. Sulistiyani and S. D. Putri S, “Evaluation of IBSI Education System Use ISO/IEC 9126 Quality Model: How is the Quality?,” in *2021 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE)*, 2021, pp. 68–73.
- [105] Á. G. Canto de Gante, “Escala de Likert: Una alternativa para elaborar e interpretar un instrumento de percepción social.,” *AmericanJournal*, vol. 12, no. 1, pp. 38–45, 2020.
- [106] X. Sun, Y. Liang, and H. Huang, “Design and Implementation of Internet of Things Platform based on Microservice and Lightweight Container,” in *2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, 2020, pp. 1353–1357.
- [107] M. MATIC, M. ANTIC, I. PAPP, and S. IVANOVIC, “Optimization of MQTT Communication Between Microservices in the IoT Cloud,” in *2021 IEEE International Conference on Consumer Electronics (ICCE)*, 2021, pp. 1–3.
- [108] P. Bistak, “3D Three-Tank Remote Laboratory Based on Matlab and Websockets,” in *2019 5th Experiment International Conference (exp.at'19)*, 2019, pp. 85–89.
- [109] J. M. Ayala Mendieta, “Control de iluminación IOT usando Blockchain,” Universidad Técnica de Machala, 2021.

ANEXOS

Anexo A: Reporte auditoria de Lighthouse



■ Eliminate render-blocking resources 0.33 s ▾

These suggestions can help your page load faster. They don't [directly affect](#) the Performance score.

DIAGNOSTICS

- ▲ Ensure text remains visible during webfont load ▾
- ▲ Does not use passive listeners to improve scrolling performance ▾
- ▲ Image elements do not have explicit `width` and `height` ▾
- ▲ Serve static assets with an efficient cache policy — 16 resources found ▾
- Avoid chaining critical requests — 21 chains found ▾
- User Timing marks and measures — 1 user timing ▾
- Keep request counts low and transfer sizes small — 33 requests • 797 KiB ▾
- Largest Contentful Paint element — 1 element found ▾
- Avoid large layout shifts — 5 elements found ▾
- Avoid long main-thread tasks — 5 long tasks found ▾

More information about the performance of your application. These numbers don't [directly affect](#) the Performance score.

PASSED AUDITS (27) Show



Accessibility

These checks highlight opportunities to [improve the accessibility of your web app](#). Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

NAMES AND LABELS

- ▲ Form elements do not have associated labels ▾

These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

CONTRAST

- ▲ Background and foreground colors do not have a sufficient contrast ratio. ▼

These are opportunities to improve the legibility of your content.

ADDITIONAL ITEMS TO MANUALLY CHECK (10) Show

These items address areas which an automated testing tool cannot cover. Learn more in our guide on [conducting an accessibility review](#).

PASSED AUDITS (21) Show

NOT APPLICABLE (21) Show



Best Practices

TRUST AND SAFETY

- ▲ Does not use HTTPS — 23 insecure requests found ▼
- ▲ Includes front-end JavaScript libraries with known security vulnerabilities — 5 vulnerabilities detected ▼
- Ensure CSP is effective against XSS attacks ▼

GENERAL

- Detected JavaScript libraries ▼

PASSED AUDITS (11) Show

NOT APPLICABLE (1) Show



SEO

These checks ensure that your page is following basic search engine optimization advice. There are many additional factors Lighthouse does not score here that may affect your search ranking, including performance on [Core Web Vitals](#). [Learn more.](#)

ADDITIONAL ITEMS TO MANUALLY CHECK (1)

Show

Run these additional validators on your site to check additional SEO best practices.

PASSED AUDITS (10)

Show

NOT APPLICABLE (4)

Show

Captured at Jul 25, 2022, 10:56 AM GMT-5

Emulated Desktop with Lighthouse 9.6.1

Single page load

Initial page load

Custom throttling

Using Chromium 103.0.0.0 with devtools

Generated by Lighthouse 9.6.1 | [File an issue](#)