



FACULTAD DE INGENIERÍAS Y CIENCIAS APLICADAS

Trabajo de fin de Carrera titulado:

**CONTROL DE USO DE LABORATORIOS ACTIVADA MEDIANTE IOT PARA
EL ÁREA DE MECÁNICA DE LA FACULTAD DE INGENIERÍAS Y CIENCIAS
APLICADAS DE LA UNIVERSIDAD INTERNACIONAL SEK**

Realizado por:

Kevin Gabriel Haro Garcés

Director del proyecto:

M.Sc. Diego Fernando Bustamante Villagómez

Como requisito para la obtención del título de:

MAGISTER EN DISEÑO INDUSTRIAL Y DE PROCESOS

QUITO, 18 DE OCTUBRE DEL 2022

Resumen

El presente proyecto de titulación se basa en el diseño de un sistema automático de reservas para el Laboratorio de Mecánica de la Facultad de Ingeniería y Ciencias Aplicadas de la Universidad Internacional SEK. El diseño busca presentar una propuesta a la ya existente forma de tomar las reservas para el laboratorio. Este sistema fue desarrollado con tecnologías y herramientas de código abierto como NodeRED, Xampp, y Telegram. El diseño busca implementar el Internet de las cosas y el Internet Industrial de las cosas, puesto que la plataforma es accesible desde cualquier lugar con retroalimentación inmediata con respecto al estado de la reserva. El prototipo del sistema fue diseñado para correr tanto en computadoras personales como también en la Computadora Raspberry Pi. Dado a la popularidad y accesibilidad de NodeRED, más estudios de investigación se podrían realizar con el fin de no solo automatizar la reserva de los laboratorios sino también un control en el uso de los materiales y herramientas dentro de los mismos.

Palabras Clave: NodeRED, Telegram, Internet de las Cosas, Internet Industrial de las Cosas, Reservas de laboratorio, Automático.

Abstract

This present dissertation project is based on the design of an automatic booking system for the Mechatronics Laboratory of the Faculty of Engineering and Applied Sciences of SEK International University. The design seeks to present a proposal to the already existing way of taking reservations for the laboratory. This system was developed with open-source technologies and tools such as NodeRED, Xampp, and Telegram. The design seeks to implement the Internet of Things and Industrial Internet of Things, as the platform is accessible from anywhere with immediate feedback regarding the status of the reservation. The prototype system was designed to run on both personal computers as well as the Raspberry Pi Computer. Given the popularity and accessibility of NodeRED, further research studies could be conducted in order to not only automate the booking of laboratories but also control the use of materials and tools within them.

Keywords: NodeRED, Telegram, Internet of Things, Industrial Internet of Things, Lab booking, Automated.

DECLARACIÓN JURAMENTADA

Yo, KEVIN GABRIEL HARO GARCÉS, ecuatoriano, con Cédula de ciudadanía N° 1724157084, declaro bajo juramento que el trabajo aquí desarrollado es de mi autoría, que no ha sido presentado anteriormente para ningún grado o calificación profesional, y se basa en las referencias bibliográficas descritas en este documento.

A través de esta declaración, cedo los derechos de propiedad intelectual a la

UNIVERSIDAD INTERNACIONAL SEK, según lo establecido en la Ley de Propiedad Intelectual, reglamento y normativa institucional vigente.



KEVIN GABRIEL HARO GARCÉS

C.I.: 1724157084

DECLARACIÓN DEL DIRECTOR DE TESIS

Declaro haber dirigido este trabajo a través de reuniones periódicas con el estudiante, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.



Diego Fernando Bustamante Villagómez

Master

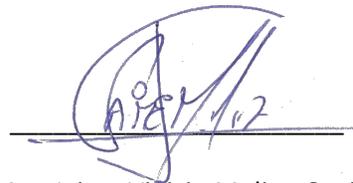
LOS PROFESORES INFORMANTES:

JOHANNA MEDRANO BARBOZA

JAIME VINICIO MOLINA OSEJOS

Después de revisar el trabajo presentado lo han calificado como apto para su defensa oral ante el tribunal examinador.

Ing. Johanna Medrano Barboza

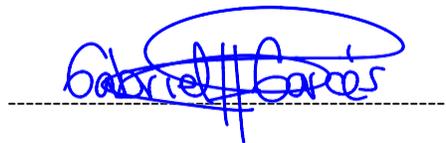


Msc. Jaime Vinicio Molina Osejos

Quito, 18 de Octubre de 2022

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.



KEVIN GABRIEL HARO GARCÉS

C.I.: 1724157084

Contenidos

Introducción	12
Antecedentes	12
Planteamiento del Problema	17
Justificación	19
Objetivo General:	20
Objetivos Específicos:	20
Hipótesis	20
Estado del arte	21
Internet de las Cosas (IoT)	21
Node-RED	24
Telegram	26
Metodología	28
Pertinencia	29
Preámbulos:	29
Diseño e Implementación:	38
Resultados	54
Conclusiones	58
Recomendaciones	59
Referencias Bibliográficas	61
Anexos	66
Pedir Nombre:	66
Get.name:	67
Pedir fecha:	67
Get.date	68
Pedir hora:	68
Get.hour:	69
Confirmar reserva:	69
Insertar reserva:	70
Validar reserva:	70
Fecha no disponible:	70

Insertar reserva:

71

Confirmacion:

71

Codigo en JSON

71

Índice de figuras

Figura 1 Proceso de Transformación de la Industria	13
Figura 2. Aplicaciones del Internet de las cosas	15
Figura 3. IoT Adaptado en Campus universitario	22
Figura 4. Ambiente de trabajo de Node-RED	25
Figura 5. Interacción entre Telegram y Node-RED	27
Figura 6. Telegram Web desde el Navegador	31
Figura 7. BotFather de Telegram	31
Figura 8. Inicialización de BotFather	32
Figura 9. Creación de un Nuevo Bot	32
Figura 10. Chat con el Nuevo Bot	32
Figura 11. Website de descarga de Node.js	33
Figura 12. Proceso de Instalación de Node.js	34
Figura 13. Run Dialog Box	34
Figura 14. Símbolo del Sistema	35
Figura 15. Node-RED en Computar Personal	35
Figura 16. Pagina web de XAMPP para descarga gratuita.	36
Figura 17. Configuración de XAMPP para proceso de instalación	37
Figura 18. Software XAMPP	38
Figura 19. Diagrama de Flujo del Programa de Reservas del Laboratorio de Mecatrónica.	38
Figura 20. Descarga de Nodos de Telegram en Node-RED	40
Figura 21. Nodos de Telegram	41
Figura 22. Flujo de Recepción de Mensajes de Telegram a Node-RED	42
Figura 23. Configuración del Nodo 'Receiver'	42
Figura 24. Configuración Avanzada del Nodo 'Receiver'	43
Figura 25. Mensajes Enviados desde Chat de Telegram con el Bot	44
Figura 26. Recepción de mensaje en Node-RED	44
Figura 27. Habilidad de Bot en Grupos de Telegram	45
Figura 28. Mensaje de Usuario a un Grupo de Telegram 'Laboratorio'	45
Figura 29. Recepcion de Mensaje en Node-RED	46
Figura 30. Recepción de Comando y Envío de Mensaje	46
Figura 31. Configuración de Nodo 'Command'	47
Figura 32. Configuración de nodo 'Function'	47
Figura 33. Respuesta de Bot a través de Node-RED al Grupo de Telegram 'Laboratorio'	48
Figura 34. Configuración de nodo MySQL	50
Figura 35. phpMyAdmin	50
Figura 36. Programación en Node-RED de Agendamiento de Laboratorio	51
Figura 37. Código QR para unirse a Grupo de Telegram	55
Figura 38. Inicialización de Reserva	55
Figura 39. Reserva de Laboratorio	55
Figura 40. Inicialización de Reserva con Nuevos Datos	56
Figura 41. Doble reserva y nuevo agendamiento	57
Figura 42. Base de Datos XAMMP	57

Índice de Tablas

Tabla 1. Comparación de Universidades con IoT y sin implementación de IoT	23
Tabla 2. Descripción de los Nodos en el Flujo de Recepción de Mensajes	43
Tabla 3. Descripción de Flujo Recibir – Enviar Mensaje	48
Tabla 4. Descripción de Flujo de Programación en Node-RED	52

Introducción

Antecedentes

En los últimos 200 años, el ser humano ha venido creando y expandiendo su tecnología de manera ininterrumpida. Estos avances tecnológicos han catapultado áreas como medicina, comunicaciones, ciencias y educación. A la vez, cada avance tecnológico ha permitido que los procesos industriales hayan tenido cambios en su manera de producir bienes o productos. Por esta razón, es que en la última década se ha venido hablando de la revolución Industrial 4.0 (Salah, 2021).

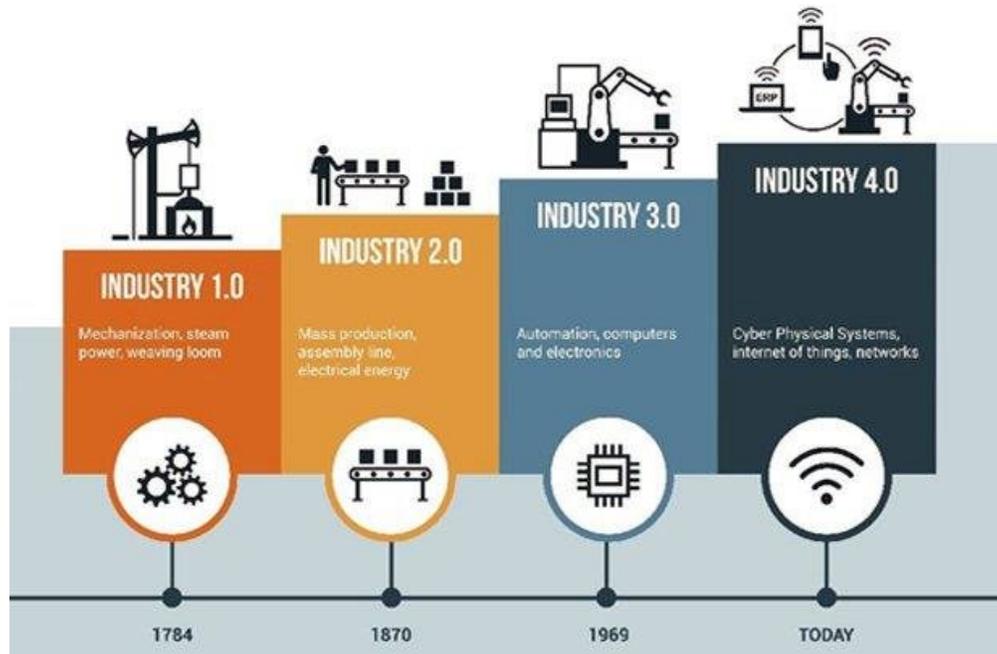
Para entender de qué se trata esta nueva terminología de Industria 4.0 es menester explorar los cambios que han venido ocurriendo en el mundo Industrial que se encuentra resumido en la [figura 1](#). La primera revolución industrial o Industria 1.0 data de finales del siglo 18. Alrededor del año 1765 la industria revolucionó y dejó su base agrónoma y su economía basada en el campo por una economía empoderada por la industria y la invención de la máquina de vapor. Durante esta época también se vivió la extracción de carbón en grandes cantidades, los procesos se volvieron mecanizados y se empezó con la manufactura de productos. A través de la invención de la máquina de vapor, el transporte fue dejando atrás a las bestias de carga por máquina a vapor y hubo una mayor producción e innovación en el ámbito textil (UpKeep, 2022).

La segunda revolución industrial ocurre 100 años después que su primera contraparte. En esta Industria 2.0 su principal característica es la del descubrimiento de la electricidad y el uso del gas natural. A la par del descubrimiento de la electricidad, vino la invención del motor a combustible. Estas nuevas tecnologías se encargaron de traer nuevos procesos a la industria y con ello, la manufactura de nuevos productos como el proceso de metales y químicos en el

ámbito industrial. El campo mecánico-automotriz tuvo un incremento exponencial gracias a la producción en masa y el modelo T (Asean Post & Nortajuddin, 2020).

Figura 1

Proceso de Transformación de la Industria



Nota. La imagen tiene la descripción de cada una de las evoluciones que han pasado a la industria. Desde la primera revolución industrial, hasta la actual. Tomado de *Character Education In The Era Of Industrial Revolution 4.0 And Its Relevance To The High School Learning Transformation Process*, por F. Herinyato, 2019, Research Gate.

Entre la segunda y tercera revolución industrial el mundo tuvo que esperar por cerca de 100 años. En 1969, el mundo empezó a implementar procesos electrónicos en la industria. A la par se implementan también nuevas fuentes de energía como lo es la proveniente de las centrales nucleares, principalmente en Europa y Estados Unidos. Esta Industria 3.0 trajo consigo la automatización de los procesos industriales a través de Controladores Lógico-programables o

Programmable Logic Controller (PLC por sus siglas en inglés) y microcontroladores. También se empieza a hablar de las Information Technologies (IT por sus siglas en inglés) (Nortajuddin, 2020).

Finalmente se presenta la industria 4.0. Esta se caracteriza por la acelerada innovación que ha ocurrido en las tecnologías digitales. A través del internet y las ventajas que este ofrece, ahora el mundo industrial puede estar conectado con sus líneas de producción estando dentro o fuera de las paredes físicas de la industria. Junto con el internet, se han introducido nuevos términos como el internet de las cosas (IoT), tecnología e información cargada en la nube, inteligencia artificial e incluso un metaverso que junta al mundo virtual con el mundo real. Las ventajas de esta cuarta revolución industrial se presentan en la forma de adquisición de datos en tiempo real, mantenimiento predictivo que impulsará a la toma de decisiones de manera inteligente para las empresas e industrias en todo el mundo (Heriyanto et al. 2019)

Vale la pena mencionar que esta nueva etiqueta de revolución Industrial 4.0 abarca todo lo relacionado al internet de las cosas o Internet of Things (IoT – por sus siglas en inglés). La tecnología IoT lleva cerca de una década estudiándose y expandiéndose en diferentes áreas cotidianas. IoT en su concepción fue visionada como una red mundial de trabajo entre máquinas capaces de comunicarse e interactuar con sus semejantes; sea que estas otras máquinas o dispositivos se encuentren cerca o a una gran distancia. Lo único necesario, como su nombre lo indica, es que exista un punto de internet (Lee & Lee, 2015).

De esta manera, el IoT no solo ha permitido mejorar las comunicaciones entre máquinas y sus pares sino también la comunicación existente entre seres humanos que se encuentran envueltos en un ámbito industrial. Además, ha potencializado la innovativa manera de comunicación humano a máquina y de máquina a máquina (M2M – machine-to-machine). Esta

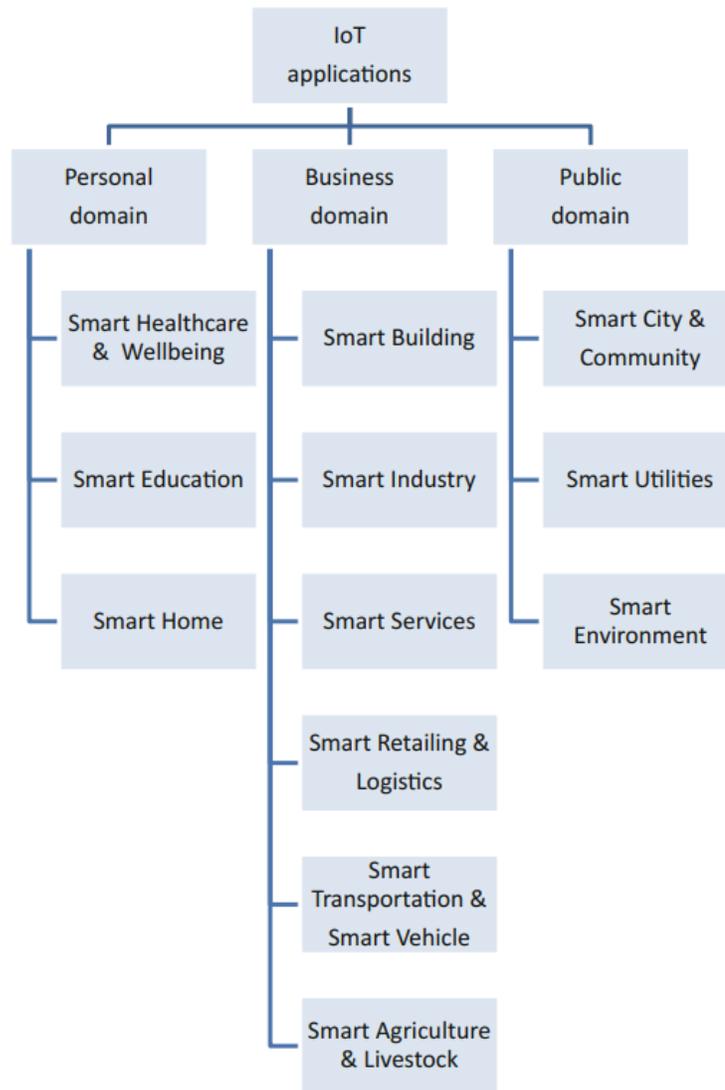
innovadora idea del IoT simplifica aspectos de la vida diaria pues no se necesita más que algunos sensores, actuadores o dispositivos electrónicos para poder tener acceso a este nuevo alcance de comunicaciones (Rajalakshmi, 2017).

IoT ha permitido la evolución de interacciones humanas con sus alrededores. El internet de las cosas permite, con poco esfuerzo, que el ser humano desarrolle avances tecnológicos en áreas de automatización, domótica, monitoreo de datos, educación, entre otras áreas que se pueden observar en la Figura 2. Sin embargo, IoT se presenta con algunos desafíos que deberá enfrentar en esta temprana etapa de su desarrollo y establecimiento en el mercado.

De acuerdo con Economides (2017), IoT presenta desafíos sociales en algunas áreas. Por mencionar un par de ejemplos, se tiene desafíos en educación y en el ámbito social. Estos desafíos abarcan temáticas con respecto a educar a especialistas y público general acerca de IoT y cómo usar el IoT para mejorar el acceso a la educación. Otro ámbito que presentará desafíos en lo social para IoT es el de acceso Universal, inclusión y no discriminación. En esta área, IoT deberá responder a desafíos como, que tanto afectará el IoT a la inclusión y como mejorar el acceso a la información y educación sin necesidad de discriminar. Esto se presenta como desafío puesto que el IoT está ligado intrínsecamente a la programación y lenguajes de programación, área que un gran porcentaje mundial aún desconoce (Soumyalatha, 2016).

Figura 2.

Aplicaciones del Internet de las cosas



Nota. Mapa conceptual de las diferentes aplicaciones de Internet de las Cosas tanto para dominio personal, de negocios y público. Tomado de *User Perceptions of Internet of Things (IoT) Systems*, por Economides, 2017, E-Business and Telecommunication.

Juntando entonces los conceptos de Industria 4.0 y de IoT, se produce un nuevo término conocido como Industrial Internet of Things (IIoT por sus siglas en inglés). Este concepto de IIoT nace de la idea de tener sensores, actuadores, máquinas y más elementos característicos de la industria, conectados al internet, sin necesidad de tener una computadora incrustada en estos

aparatos (Abuhasel & Khan, 2020). No solo esto, sino también que los dispositivos y sensores puedan monitorear, recolectar, intercambiar, analizar e instantáneamente actuar inteligentemente para el continuo proceso industrial, cambiando el ambiente de trabajo o el comportamiento de las máquina y dispositivos, con poca o nula participación de un ser humano. (Boyes et al., 2018)

La programación es de vital importancia cuando de IoT y IIoT se habla. Una buena interfaz permite una relación armoniosa entre humano y máquina. Por el otro lado, una interfaz errónea o comprometida por código incorrecto puede causar rechazo a nuevas tecnologías y la relación entre usuario y dispositivo se vuelve lejana, algo que evitaría a las industrias a crecer.

En la mayoría de los casos, el ser humano se abstiene de involucrarse en el área de lenguaje de programación por su complejidad y, hasta cierto punto, su modo abstracto de lectura de lenguajes de programación (James, 2021).

Node-RED es un lenguaje de programación gráfico, y aunque ciertamente requiere de conocimientos básicos de programación, es un gran paso para que el usuario pueda comprender que sucede detrás de su interfaz preferida de manera gráfica (Howe, 2021). Gracias a Node-RED, su versatilidad y adaptabilidad, se puede implementar IoT en el ámbito social y de educación.

Planteamiento del Problema

De acuerdo con cifras publicadas por la Senescyt, en un lapso de 4 años el acceso de estudiantes a la educación superior incrementó en 12.3 % (SENESCYT, 2020). Este incremento de alumnos crea una mayor demanda en todo aspecto de la vida universitaria. Las instalaciones de estos centros de educación superior son parte de esa demanda que se debe cumplir para dar

buen servicio y una educación adecuada, mucho más en estos tiempos que se está volviendo a clases presenciales luego de haber pasado meses de pandemia con una modalidad remota.

La Universidad Internacional SEK también se encuentra en el grupo de establecimientos que está volviendo a abrir sus puertas a los estudiantes. Este retorno a las aulas genera desafíos, pero de igual manera oportunidades, para investigar y posiblemente solucionar los contratiempos que se puedan generar debido a este crecimiento del número de estudiantes accediendo a la educación superior.

Si el patrón de acceso a educación superior sigue en aumento, como ha ocurrido en los últimos 4 años (SENESCYT, 2020) y asumiendo que no se ha invertido en la construcción de nuevos laboratorios, la universidad se encontrará con el dilema de tener un gran número de alumnos con restringido acceso a los laboratorios.

Este mal no es solo de la Universidad Internacional SEK, sino un mal que afectará a las Universidades a nivel nacional. De acuerdo con datos reportados por diario El Universo de Ecuador, el presupuesto para educación superior en el año 2022 se reducirá en \$77 millones (dólares americanos). Haciendo así virtualmente imposible invertir en construcción de nuevos laboratorios. Esta problemática entonces afecta a todas las entidades de educación superior, que deberán encontrar soluciones realistas y sustentables a la demanda, no solo de una plaza a la educación, sino también de laboratorios una vez que el alumno se haya integrado a un ente universitario.

La presente tesis de investigación tomó a consideración esta demanda de acceder a establecimientos de desarrollo de habilidades como lo son los laboratorios de las Universidades. Específicamente, esta investigación se centró en el laboratorio de Mecatrónica de la facultad de

Ingenierías y Ciencias Aplicadas de la Universidad Internacional SEK. El tema en cuestión fue tratado a través de la Industria 4.0 y el internet de las cosas (IoT). Mediante el uso del lenguaje Node-RED y la aplicación de teléfonos móviles Telegram, se investigó en la implementación de un servicio de mensajería automática para crear un agendamiento controlado de las instalaciones del laboratorio de mecatrónica.

Justificación

El IoT es la nueva tendencia mundial que promueve la interacción entre seres humanos, dispositivos, aplicaciones y maquinas. A través de IoT el ser humanos podría mejorar sus procesos mediante recomendaciones y datos obtenidos de manera remota usando estos dispositivos o máquinas conectadas al internet. Se reportó que para el año 2021 existirían alrededor de 30 billones de dispositivos conectados al internet (Economides, 2017).

El Impacto de IoT ya se siente en sectores de ciudades inteligentes, Hogares Inteligentes (Smart homes) y, entre otros más, Escuelas Inteligentes (Smart Schools). El IoT tiene el potencial de unir a la gente y ayudar a cumplir con sus objetivos con respecto a educación, salud, familia, etc.

La UISEK, comprometida con el avance del país para alcanzar la revolución Industrial 4.0, promueve el estudio e investigación del IoT y sus aplicaciones en el ámbito de educación. Es por eso por lo que el siguiente trabajo de investigación tiene como objetivo regularizar y automatizar el proceso de reserva de espacios de laboratorio usado por los estudiantes de la Facultad de Ingenierías de Ciencias Aplicadas de la Universidad Internacional SEK.

En las siguientes etapas se implementó un nuevo sistema de reservas de forma virtual. Los estudiantes podrán hacer las reservas de los laboratorios a través de la aplicación celular Telegram. Esta aplicación se conectó en conjunto con Node Red, que es la plataforma de programación. Esta plataforma se encargó de enlazar Telegram para notificar a los estudiantes que su reserva ha sido realizada exitosamente o no.

Objetivo General:

Automatizar y controlar el uso y agendamiento de laboratorios de Mecatrónica de la Facultad de ingenierías y Ciencias Aplicadas de la Universidad Internacional SEK a través de IoT como parte de la Industria 4.0 en la Educación.

Objetivos Específicos:

- Aplicar IoT en educación y sus beneficios en la automatización de procesos.
- Implementar un código de programación en una plataforma dedicada a gestión de IoT
- Prototipar el diseño de un dispositivo de gestión y almacenamiento de datos para uso en el agendamiento de laboratorios.

Hipótesis

Esta investigación propone indagar si el ambiente de programación grafica Node-Red y el servicio de mensajería instantánea Telegram son capaces de dar abasto al requerimiento de reservar el laboratorio de mecatrónica de la Universidad Internacional SEK sumándose al Internet de las cosas (IoT).

Estado del arte

Internet de las Cosas (IoT)

El Internet de las cosas IoT es una revolucionaria tecnología que permite la conexión entre dispositivos, máquinas, ambientes y el ser humano. Esta conexión se logra a través de sensores, actuadores y dispositivos que transmiten información importante a un lugar virtual como la nube. La información que se obtiene a través de IoT puede luego ser útil para mejorar procesos, adaptar ambientes más adecuados para el usuario, reducir gastos innecesarios y promover el uso de la tecnología en diferentes áreas sociales (Majeed & Ali, 2022).

Otra área donde se puede observar la actuación e incorporación de humanos a través de IoT es con el uso de Tablet, computadoras portátiles y celulares. Gartner (2017), quien es citado por Abed, predijo hace algunos años que 20.4 billones de dispositivos conectados a la red serían usados en el 2020 (Abed, 2019).

De esta manera, si las Universidades buscan estar actualizadas y usando tecnología de punta a nivel mundial, estos centros de educación superior deben adaptarse al IoT que sigue evolucionando, cada vez comunicándose con más dispositivos y de manera eficaz. De hecho, algunos de los beneficios que provee el IoT en instituciones de educación son en campos como la eficiencia energética, comunicaciones entre profesores/ tutores y alumnos, y la reducción en costos de operación. Por ejemplo, escuelas en el área de New Richmond en los Estados Unidos de Norteamérica ahorran anualmente \$128 000 (ciento veintiocho mil dólares americanos) utilizando un sistema web que se encarga de gestionar todas las herramientas mecánicas en dichas escuelas (Meola, 2022).

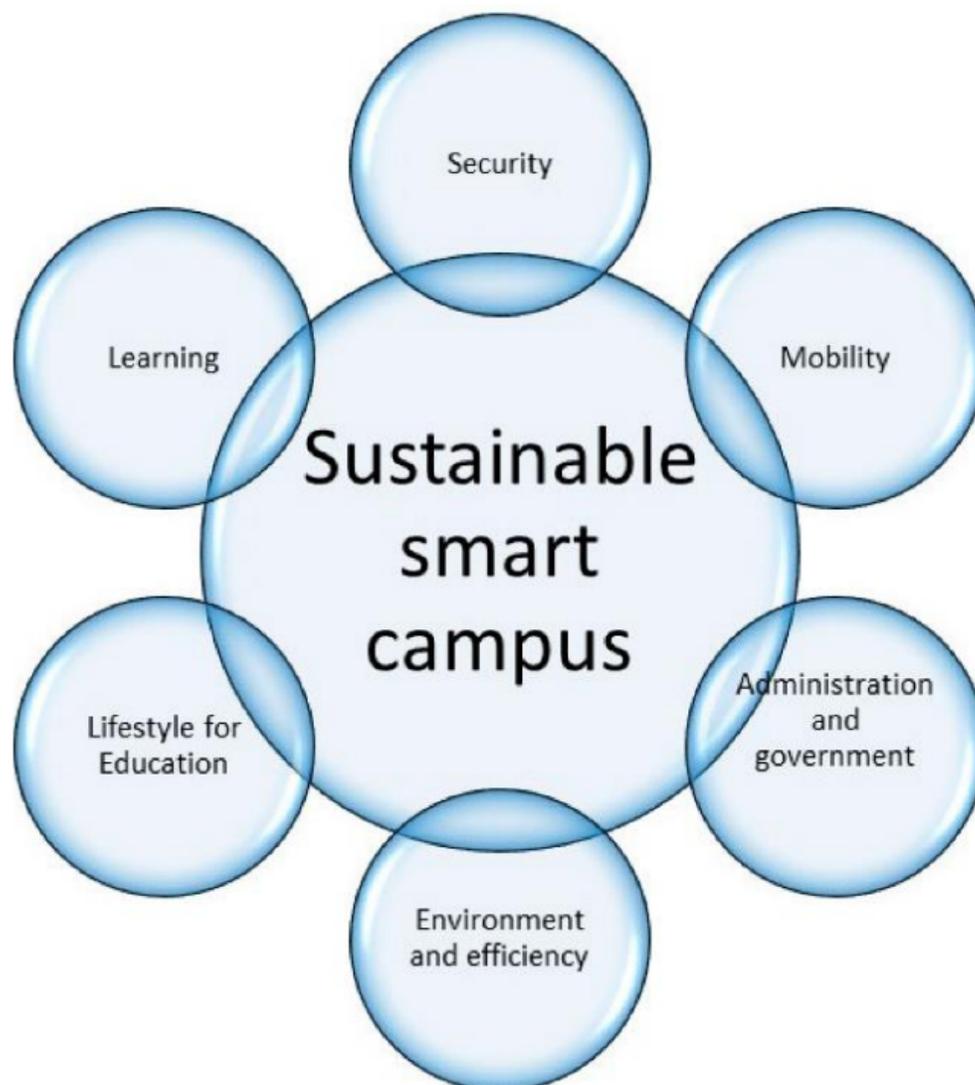
Por otro lado, Universidades en todo el mundo están usando dispositivos IoT como son cámaras de seguridad, dispositivos electrónicos controlados por luz o temperatura, dentro de las aulas en sus respectivos campus. Estas tecnologías conectadas al internet son útiles para optimizar actividades dentro del aula, monitorear la salud y seguridad de los alumnos, mejorar la colaboración y participación de estudiantes de manera remota. También esta implementación de IoT en el ámbito educativo permite, tanto a docentes como alumnos, automatizar el proceso educativo (Ban et.al, 2017).

La Universidad de Brescia desarrolló su campus con Edificios Inteligentes (Smart Buildings). Esto le ha permitido ahorrar costes y tiempos en cuanto al uso de sus instalaciones, automatizar el tiempo requerido de mantenimiento de maquinaria y laboratorios, proteger el medio ambiente y mejorar la gestión energética del Campus (Abuarqoub et al., 2017).

En el estudio llevado a cabo por Al-Emran et al. (2019), se presentan numerosos beneficios, que se pueden observar en la Figura 3 y Tabla 1, al implementar campus universitarios de tecnología IoT. Entre sus remarcaciones, es interesante hablar de las aplicaciones que tiene el IoT en controlar la asistencia a clases y laboratorios de los estudiantes. A adaptar el proceso educativo a las necesidades, en cuestión del tiempo disponible, que cada estudiante presente.

Figura 3.

IoT Adaptado en Campus universitario



Nota. Beneficios de poder aplicar el Internet de las Cosas dentro de los campus universitarios. Tomado de *Sustainable Smart Campus*, por K. Seymeliyski, 2020, Burgas University.

Tabla 1.

Comparación de Universidades con IoT y sin implementación de IoT

	Instituciones Educativas con IoT	Instituciones Educativas sin IoT
Asistencia	Automático utilizando diferentes sistemas de toma de asistencia	Manualmente, toma tiempo y esfuerzo

Seguimiento estudiantil	Facilidad de realizar seguimiento a estudiantes	Dificultad en realizar seguimiento a estudiantes
Campus	Inteligente, seguro y fácil de controlar	Inseguro y difícil de controlar
Energía	Facilidad de gestionar el consumo energético	Dificultad al gestionar el consumo energético
Parqueadero	Facilidad para encontrar parqueadero	Dificultad para encontrar lugares de aparcamiento
Edificios	Inteligente y sencillo de gestionar	Dificultad en la gestión del edificio
Aprendizaje	Aprendizaje personalizado y remoto	Aprendizaje presencial como única forma de adquirir información

Nota. Tomado de *IoT in Education: Its Impacts and Its Future in Saudi Universities and Educational Environments*, por S. Abed, et al., 2019, First International Conference on Sustainable Technologies for Computational Intelligence.

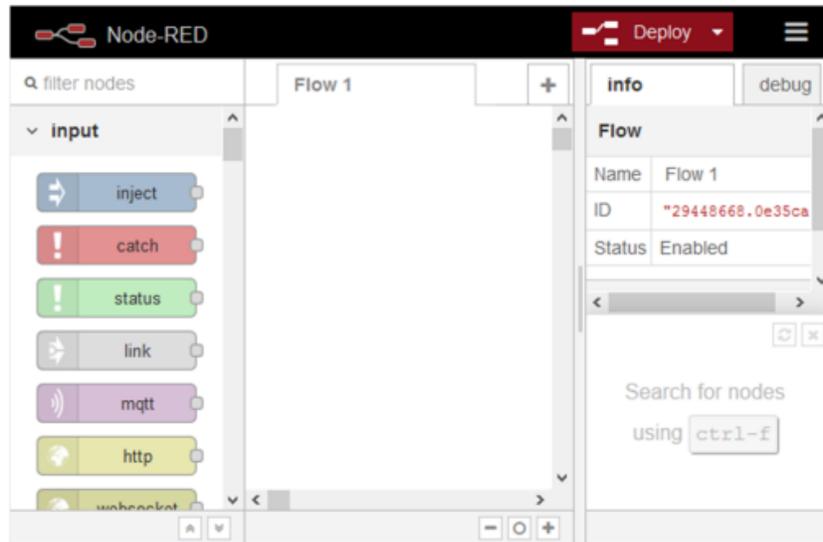
Node-RED

Como fue mencionado con anterioridad, se busca implementar IoT en la UISEK juntamente con la plataforma de programación Node-RED. Node-RED es una herramienta de desarrollo de software de fuente abierta (open source) basada en diagramas (Flow-based). A través de Node-RED se pueden integrar dispositivos de hardware IoT, interfaces de aplicaciones programables (API por sus siglas en inglés) y servicios online. Node-RED es desarrollado por IBM Emerging technology. Node-RED es una herramienta basada en el modelo libre de

JavaScript, construido en plataforma Node.js, la cual provee de un editor de flujo visual que se observa en la Figura 4 (Node-RED).

Figura 4.

Ambiente de trabajo de Node-RED



Nota. La figura muestra como luce el ambiente grafico de trabajo de Node-RED. En la izquierda se pueden ver algunos nodos disponibles. A la derecha esta la consola para el 'debug' de las funciones.

Tomado de *IoT Sensor Integrating to Node-RED platform*, por Lekić, M. & Gardašević, G, 2018, INFOTEH

Node-RED es una herramienta de programación para conectar dispositivos de hardware, APIs y servicios en línea de una manera innovadora. Node-RED provee un editor basado en un modelo de buscador que facilita la conexión entre nodos. El conjunto de nodos conectados los unos a los otros son conocido como flujo (Flow). La paleta de Node-RED ofrece una amplia opción de nodos que realizan múltiples y versátiles funciones. Adicionalmente a los nodos preestablecidos en la herramienta de programación, el bloque 'Function' permita al usuario crear funciones usando JavaScript. Node-RED es construido basándose en Node.js. Esto da grandes

ventajas como modelo de cero bloqueos, impulsado por eventos que permite correr Node-RED en hardware de bajo costo como la tarjeta Raspberry Pi así también como correrlo desde la nube (*Node-RED*, 2021).

Node-RED permite a sus usuarios desarrolladores de software enlazar entradas, salidas y procesos a través de nodos. Estas redes entrelazadas crean un flujo de información para el procesamiento de datos, control de cosas, y el enviar alarmas a través de diferentes plataformas. Node-RED trabaja con el principio siguiente: Permite la conexión de nodos personalizados a una o varias cosas con el fin de realizar varias funciones tales como enviar y recibir información vía correo electrónico, Twitter o mensajería electrónica como Telegram (Lekic & Gardasevic, 2018).

Telegram

Telegram es una gran aplicación para telefonía móvil. El fin de esta aplicación es el de mensajería instantánea con gran adaptabilidad (Telegram FAQ, n.d.). Esta adaptabilidad hace que Telegram sea una aplicación versátil que permite recibir notificaciones o mensajes de asistentes de hogares como Google home o Alexa de Amazon. No solamente esto, sino que Telegram puede conectarse fácilmente a Node-RED.

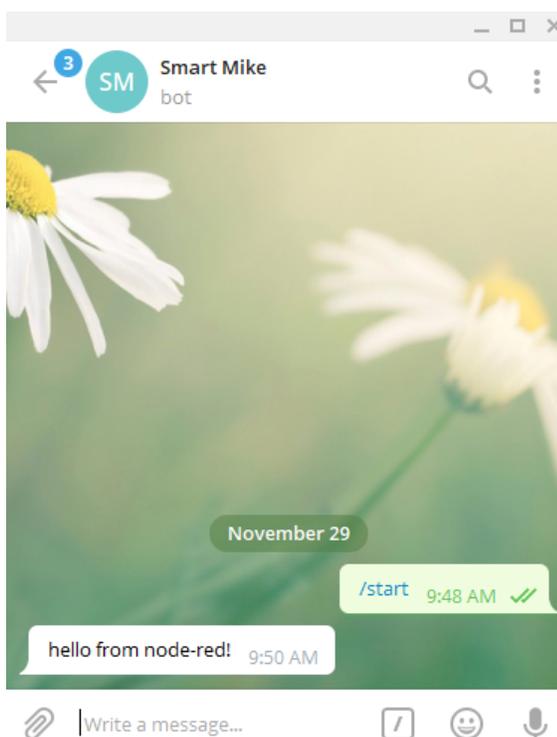
El modelo en gestión para este proyecto fue en su totalidad virtual. Este modelo comprendió el código escrito en el ambiente de programación online Node-RED. Node-RED da la facilidad de crear los Bots. Estos Bots, como se los conoce en el campo informático, se encargarán de la interacción entre usuario y sistema de recepción de mensajes.

Las ventajas que presenta esta simbiosis de Telegram con Node-RED es la facilidad de uso, puesto que para programar en Node-RED no se necesita una amplia capa de conocimiento en software o informática. El hecho es que Node-RED es un lenguaje visual el cual permite ver

de manera más gráfica las interacciones entre comandos de programación (Kravchenko, 2020). De esta manera, lo primero que se debe crear es la plataforma y el enlace entre Telegram y Node-RED como lo muestra la Figura 5.

Figura 5.

Interacción entre Telegram y Node-RED



Nota. La figura muestra el comando '/start' usando en un chat de Telegram con un bot de nombre "Smart Mike". A través de Telegram, el bot responde automáticamente al comando con el texto "hello from node-red!". Tomado de *How to Create a Telegram Reminder Bot Using Node-RED*, por A. Kravchenko, 2020, Eleventh International Conference on Contemporary Computing (IC3).

XAMPP

XAMPP es un servidor web de plataformas cruzadas que es gratuito y de código abierto (open-source). XAMPP es de plataforma cruzada porque trabaja con plataformas de bases de

datos y servidores como Apache, MySQL, PHP y Perl. Esta herramienta es de fácil instalación y puede trabajar tanto en el sistema operativo Windows, así como Linux. Al ser una herramienta de código abierto, desarrolladores pueden acceder a esta plataforma de manera sencilla y rápida trabajando de manera local. Este proyecto usa XAMPP solamente como base de datos. Para almacenar los nombres de las personas y las fechas en las cuales los usuarios desean hacer uso del laboratorio.

Metodología

El siguiente trabajo basó su modelo metodológico en una investigación experimental. De acuerdo con la bibliografía, la investigación experimental en su forma base, tuvo como medio de trabajo el manipular, bajo condiciones debidamente preestablecidas y controladas, una o más variables con la finalidad de definir los cambios producidos al variar dicha variable independiente (Ramos-Galarza, 2021).

El planteamiento de esta metodología sigue los siguientes lineamientos:

- **Pertinencia:** La problemática establecida anteriormente presenta la necesidad de la Universidad Internacional SEK a implementar tendencias ya aplicadas en campus internacionales e implementar tecnologías del IIoT para la automatización del agendamiento del laboratorio.
- **Preámbulos:** Esta es la etapa donde se deben establecer los limitantes y condiciones que delinear el programa. Estas condiciones deben ser las necesarias para alcanzar el objetivo del programa.
- **Diseño e implementación:** La siguiente etapa, nombrada diseño, es en la cual se construye un algoritmo y diagramas de flujo que sigan las especificaciones del paso previo.

- **Codificación:** Una de las etapas más importantes en el desarrollo del programa es la de codificación. En esta etapa se traducirá, lo anteriormente especificado en el algoritmo, en el respectivo lenguaje de programación.
- **Verificación:** Esta es la etapa de pruebas, en las cuales el programador evalúa la viabilidad y funcionalidad del programa. Se deben volver a tomar en cuenta las condiciones y los objetivos principales para llegar a la conclusión de si el programa satisface las demandas impuestas en los primeros puntos. De no ser así, el programa debe ser revisado tanto en su etapa de diseño como de codificación.

Pertinencia

El 2 de junio del 2021, el COE (Comité de Operaciones de Emergencia) Nacional aprobó el retorno a clases presenciales de 37 entes educativos de educación superior entre los cuales se encontraba la Universidad Internacional SEK. Esto después de más de un año de que la presencialidad había quedado suspendida debido a la emergencia sanitaria causado por la pandemia de COVID-19 (Quillupangui, 2021). Este retorno progresivo a clases generó nuevas oportunidades de desarrollo en todos los ámbitos educativos. Uno de estos es la manera en la cual se reservan los espacios de laboratorio de la Universidad. El problema entonces planteado es el de buscar una herramienta accesible a todos los estudiantes a través de la cual se consiga un fácil acceso a un sistema de reservas del laboratorio de la Facultad de Ingeniería y Ciencias Aplicadas (FICA).

Preámbulos:

Para llevar a cabo este programa, se propuso utilizar el servicio de mensajería instantánea Telegram, el ambiente de trabajo y programación grafica Node-Red y la base de datos XAMPP.

Telegram:

Telegram es una aplicación de mensajería instantánea que, hace algunos años atrás, contaba con más de 62 millones de usuarios de los cuales 15 millones usaban la aplicación a diario y más de un millón de nuevos usuarios se descargaban o empezaban a usar la aplicación (Muslih et al., 2018). Por su seguridad y versatilidad, Telegram es usada en trabajos investigativos enfocados en IoT (Dargahi Nobari et al., 2017). Adicionalmente, el código de Telegram es un código abierto, lo cual permite a desarrolladores de software de todo el mundo diseñar proyectos en esta aplicación. No solo eso, sino también que desarrolladores de software puedes crear sus propios Telegram Bots. Y es que Telegram cuenta con Telegram Bots, que son cuentas de la plataforma operadas por software y que a menudo, tienen características distintivas de la inteligencia artificial. Estos Telegram Bots son multifacéticos, pero una de sus características más útiles para la Industria 4.0 es la de pasar comandos a través del Internet de las cosas (*Telegram Bot Platform*, 2015).

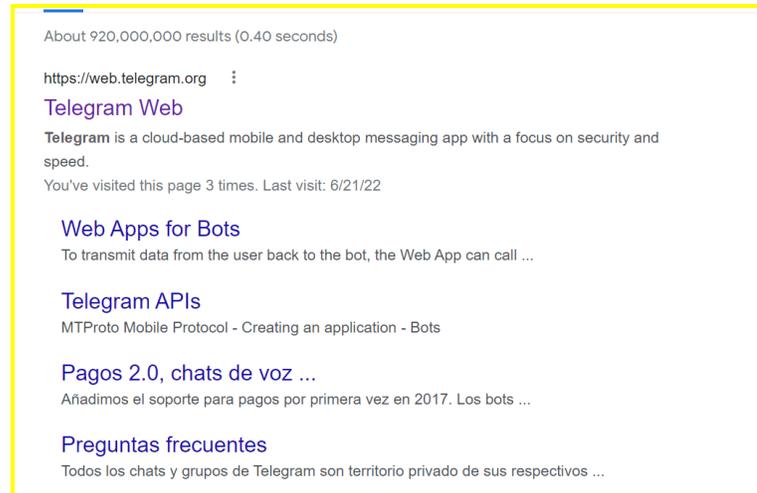
Para usar Telegram, lo primero que se debe hacer es descargar la aplicación en un teléfono móvil sea este usando el AppStore o Google Store. Se debe configurar Telegram en el teléfono móvil siguiendo los pasos. Una vez se tenga Telegram en el celular, lo siguiente es abrir la versión Web de Telegram en una computadora portátil como se muestra en la Figura 6.

Para poder usar Bots en Telegram, lo que se debe hacer es llamar al BotFather como muestra a continuación la Figura 7. Una vez que se acceda al BotFather, esta cuenta proporcionó con la ayuda para poder crear un nuevo Bot como se muestra en la Figura 8 y Figura 9. Para crear el nuevo Bot se digitó el comando /start. Seguidamente se escribió el comando /newbot. El botFather pidió darle un nombre al nuevo Bot y también tipear un nombre de usuario para el bot.

Este nombre de usuario debe terminar en ‘bot’. Este es el último paso, y con esto el bot quedó listo para ser usado como se observa en la Figura 10.

Figura 6.

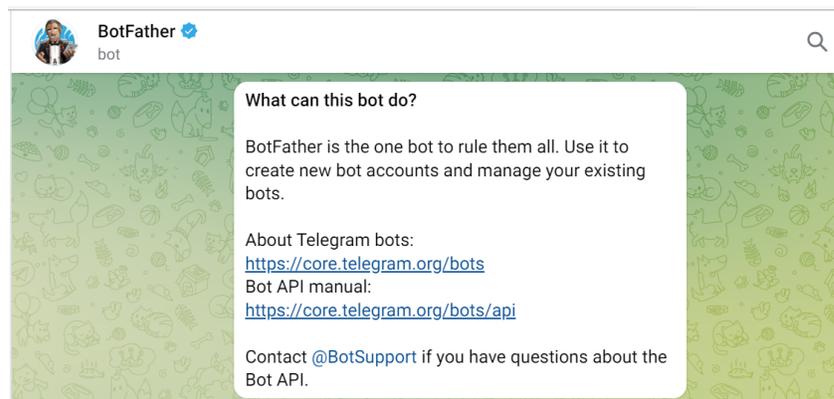
Telegram Web desde el Navegador



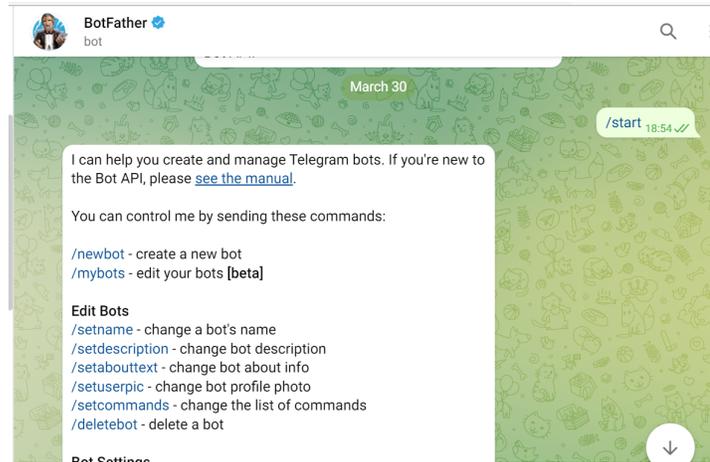
Nota. La figura es una captura de pantalla del navegador de Google.

Figura 7.

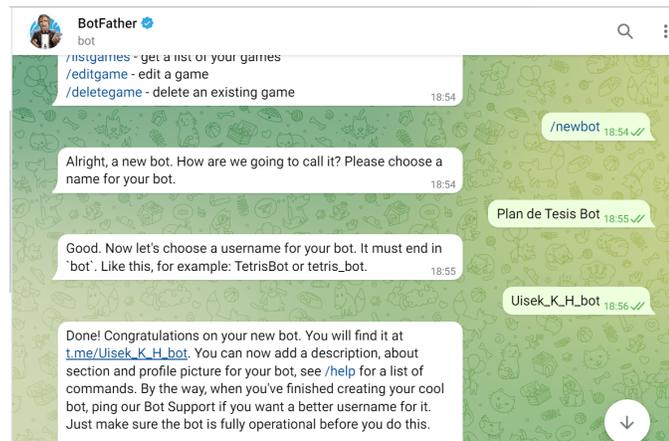
BotFather de Telegram



Nota. La figura muestra la ayuda proporcionada en Telegram por el 'Botfather'. Autoría propia.

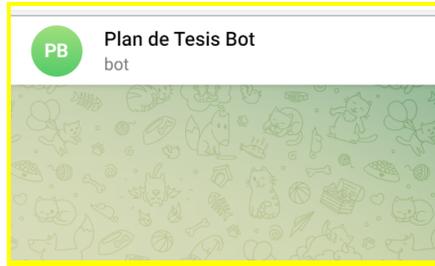
Figura 8.*Inicialización de BotFather*

Nota. Inicialización y configuración de BotFather. Autoría propia.

Figura 9.*Creación de un Nuevo Bot*

Nota. Autoría propia.

Figura 10.*Chat con el Nuevo Bot*



Nota. Autoría propia.

Node-RED

Node-RED puede correr de manera local en un ordenador personal con sistema operativo de Windows. Con el fin de que Node-RED pueda funcionar de manera local, es menester instalar de antemano Node.js para Windows. Node.js se puede instalar desde la página web como se puede apreciar en la Figura 11.

Figura 11.

Website de descarga de Node.js

The screenshot shows the Node.js website's download page. It features a navigation bar with links for HOME, ABOUT, DOWNLOADS, DOCS, GET INVOLVED, SECURITY, CERTIFICATION, and NEWS. Below the navigation bar, the 'Downloads' section is highlighted, indicating the 'Latest LTS Version: 16.15.1 (includes npm 8.11.0)'. A message encourages users to download the source code or a pre-built installer. The page is divided into two main sections: 'LTS Recommended For Most Users' and 'Current Latest Features'. Under 'LTS', there are three options: 'Windows Installer' (node-v16.15.1-x64.msi), 'macOS Installer' (node-v16.15.1.pkg), and 'Source Code' (node-v16.15.1.tar.gz). Under 'Current', there are two options: 'Windows Installer (.msi)' and 'Windows Binary (.zip)'. A table below these options shows the bitness for each: 32-bit and 64-bit.

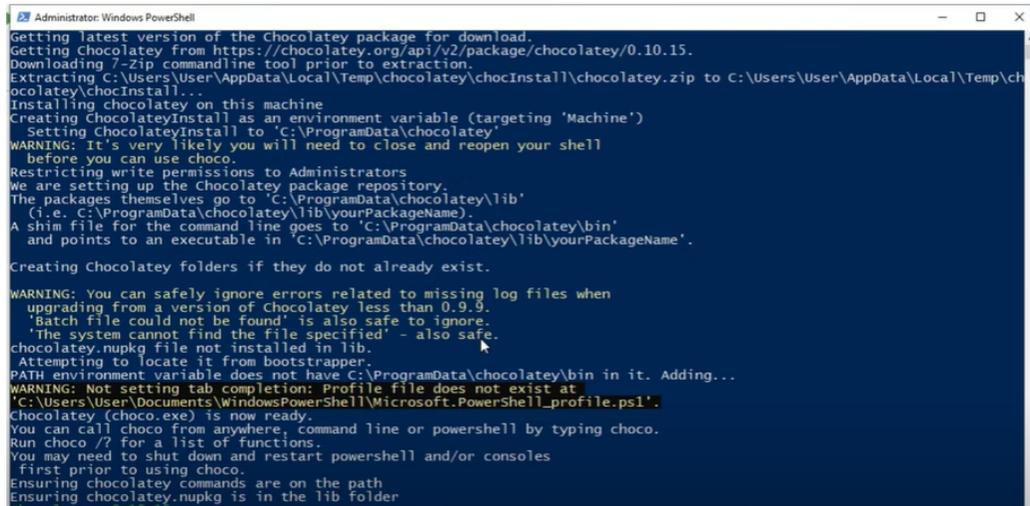
	32-bit	64-bit
Windows Installer (.msi)		
Windows Binary (.zip)		

Nota. Autoría propia.

Node.js empezara la descarga y para su instalación solo debemos seguir las indicaciones que aparecen en el software como se muestra en la Figura 12.

Figura 12.

Proceso de Instalación de Node.js

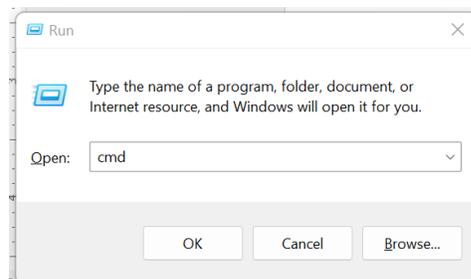


```
Administrator: Windows PowerShell
Getting latest version of the Chocolatey package for download.
Getting chocolatey from https://chocolatey.org/api/v2/package/chocolatey/0.10.15.
Downloading 7-Zip commandline tool prior to extraction.
Extracting C:\Users\User\AppData\Local\Temp\chocolatey\chocInstall\chocolatey.zip to C:\Users\User\AppData\Local\Temp\chocolatey\chocInstall...
Installing chocolatey on this machine
Creating ChocolateyInstall as an environment variable (targeting 'Machine')
Setting ChocolateyInstall to 'C:\ProgramData\chocolatey'
WARNING: It's very likely you will need to close and reopen your shell
before you can use choco.
Restricting write permissions to Administrators
We are setting up the Chocolatey package repository.
The packages themselves go to 'C:\ProgramData\chocolatey\lib'
(i.e. C:\ProgramData\chocolatey\lib\yourPackageName).
A shim file for the command line goes to 'C:\ProgramData\chocolatey\bin'
and points to an executable in 'C:\ProgramData\chocolatey\lib\yourPackageName'.
Creating Chocolatey folders if they do not already exist.
WARNING: You can safely ignore errors related to missing log files when
upgrading from a version of Chocolatey less than 0.9.9.
'Batch file could not be found' is also safe to ignore.
'The system cannot find the file specified' - also safe.
chocolatey.nupkg file not installed in lib.
Attempting to locate it from bootstrapper.
PATH environment variable does not have C:\ProgramData\chocolatey\bin in it. Adding...
WARNING: Not setting tab completion: Profile file does not exist at
'C:\Users\User\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1'.
Chocolatey (choco.exe) is now ready.
You can call choco from anywhere, command line or powershell by typing choco.
Run choco /? for a list of functions.
You may need to shut down and restart powershell and/or consoles
first prior to using choco.
Ensuring chocolatey commands are on the path
Ensuring chocolatey.nupkg is in the lib folder
```

Nota. Autoría propia.

Figura 13.

Run Dialog Box



Nota. Autoría propia.

Una vez que Node.js se haya instalado satisfactoriamente, se debe entrar al símbolo del sistema para poder instalar Node-RED como está indicado en la Figura 13. En el símbolo del sistema se tipea el comando “npm install -g --unsafe-perm node-red”. Este comando instala Node-RED de manera local. Finalmente, para correr Node-RED, se usa el comando Node-RED

en el símbolo del sistema como lo indica la Figura 14. Esto da como resultado una dirección IP que puede ser introducida en cualquier navegador y esta empezó a operar el ambiente de Node-RED como se muestra en la Figura 15 (Hacks 360, 2019).

Figura 14.

Símbolo del Sistema

```
node-red
Microsoft Windows [Version 10.0.22000.739]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Kevin>node-red
02 Jun 20:33:22 - [info]

Welcome to Node-RED
=====
02 Jun 20:33:22 - [info] Node-RED version: v2.2.2
02 Jun 20:33:22 - [info] Node.js version: v16.14.2
02 Jun 20:33:22 - [info] Windows_NT 10.0.22000 x64 LE
02 Jun 20:33:24 - [info] Loading palette nodes
Missing ENV var: CONFIG_PATH
02 Jun 20:33:26 - [info] Settings file : C:\Users\Kevin\.node-red\settings.js
02 Jun 20:33:26 - [info] Context store : 'default' [module=memory]
02 Jun 20:33:26 - [info] User directory : \Users\Kevin\.node-red
02 Jun 20:33:26 - [warn] Projects disabled : editorTheme.projects.enabled=false
02 Jun 20:33:26 - [info] Flows file : \Users\Kevin\.node-red\flows.json
02 Jun 20:33:26 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

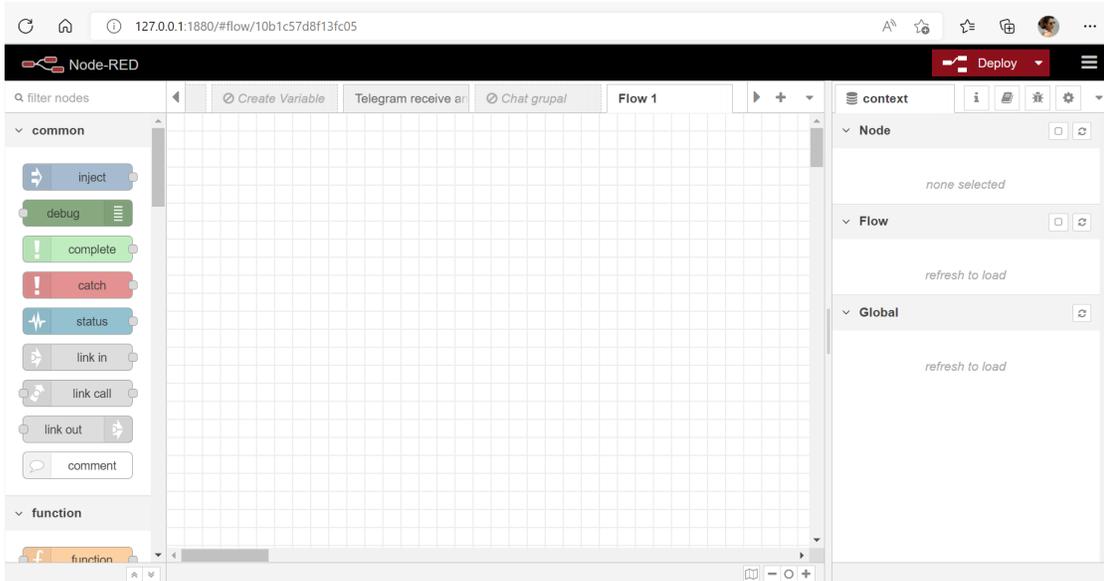
If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----
02 Jun 20:33:26 - [info] Server now running at http://127.0.0.1:1880/
02 Jun 20:33:26 - [info] Starting flows
02 Jun 20:33:26 - [info] Started flows
02 Jun 20:33:26 - [error] [MySQLdatabase:Laboratorio] Error: connect ECONNREFUSED 127.0.0.1:3306
```

Nota. Autoría propia.

Figura 15.

Node-RED en Computar Personal



Nota. Autoría propia.

XAMPP

XAMPP puede ser instalado en Windows de manera gratuita. Para la instalación debemos buscar en un navegador a XAMPP. Una vez dentro de la página Web, se presionó clic en el botón de descarga de XAMPP para Windows como se observa en la Figura 16.

Figura 16.

Página web de XAMPP para descarga gratuita.

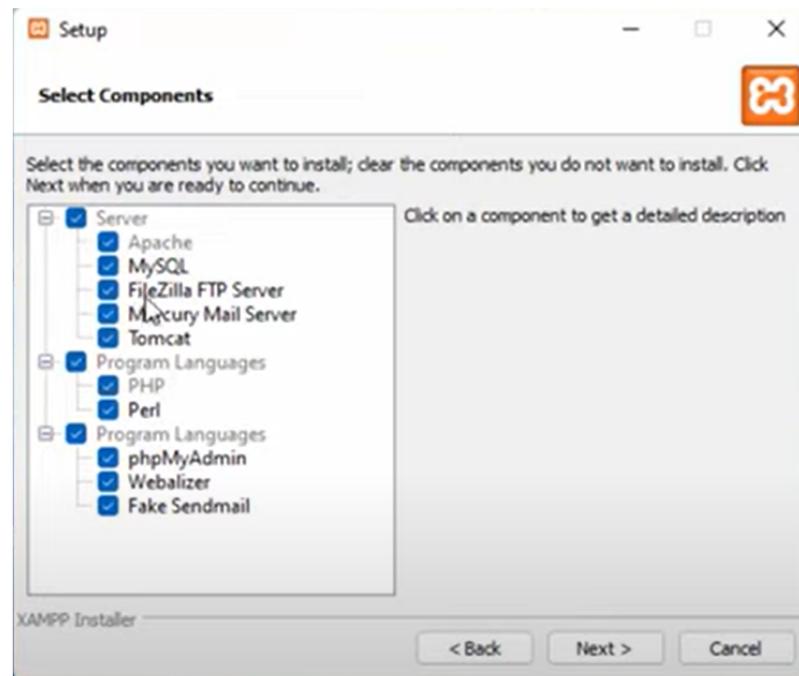


Nota. Autoría propia. Captura de pantalla del proceso de descarga de XAMPP

Una vez confirmada la descarga, instalar XAMPP es sencillo. Lo que se debe tener en cuenta son las instrucciones de XAMPP. Por ejemplo, uno de los primeros pasos es seleccionar los servidores que se desean tener en la base de datos. Para este proyecto se necesitó solamente el servidor deseado y compatible con NodeRed más otro servidor que viene instalado por default. El servidor que viene instalado por defecto es Apache, y el que se instala manualmente seleccionando en la barra de selección es MySQL como se indica en la Figura 17.

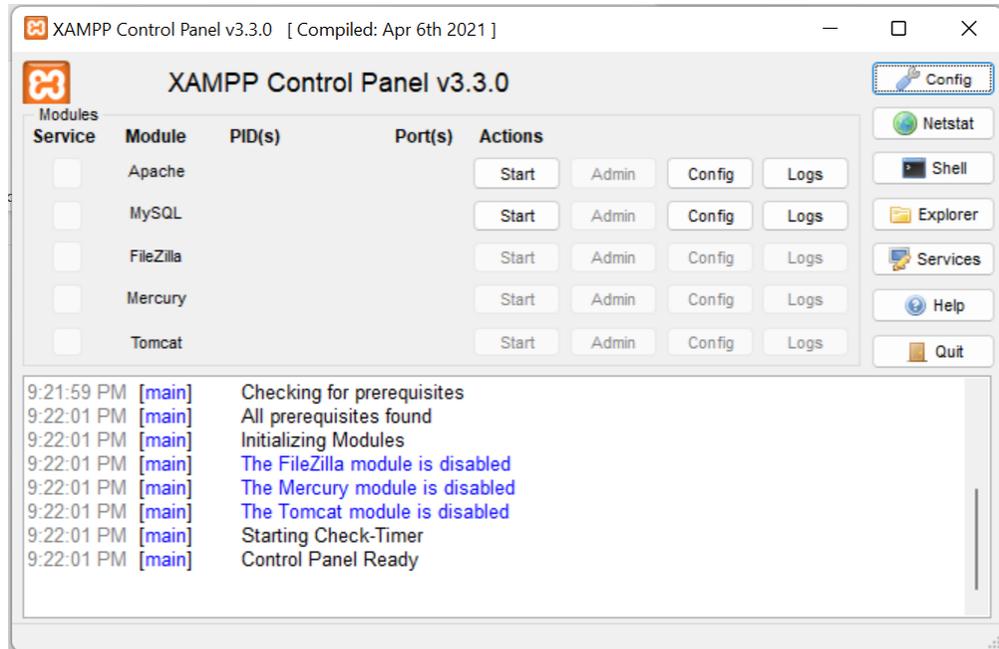
Figura 17.

Configuración de XAMPP para proceso de instalación



Nota. Autoría Propia.

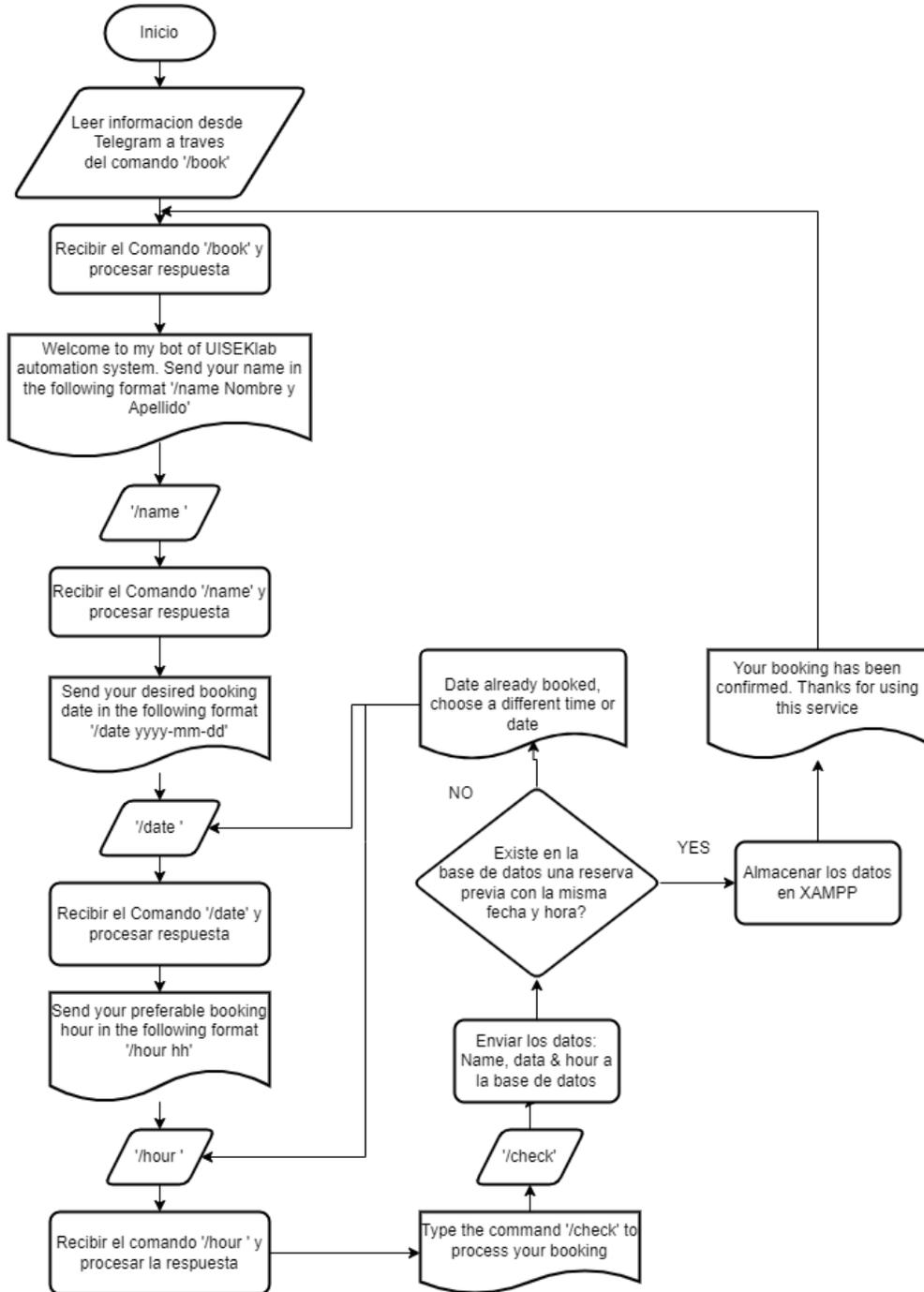
El proceso de instalación a partir de este punto es solamente leer que la información sea correcta y aceptar las configuraciones del software. Por ejemplo, XAMPP preguntó el idioma y el navegador de preferencia que se usó. Una vez que el proceso de finalización haya terminado, el ambiente de trabajo de XAMPP luce como se muestra en Figura

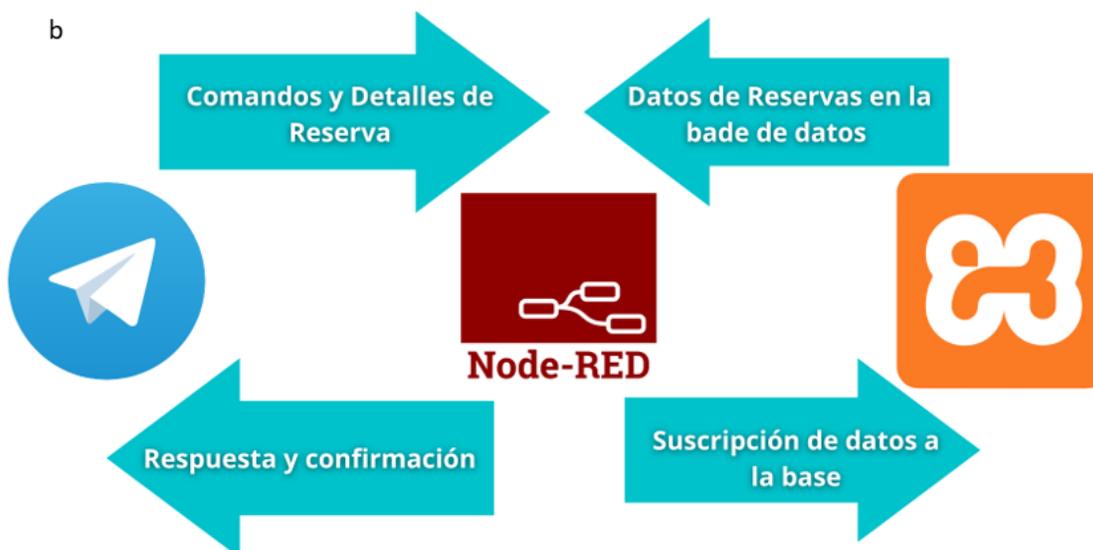
18.Figura 18.*Software XAMPP**Nota.* Autoría propia.**Diseño e Implementación:**

El diseño del programa, con sus especificaciones, se muestra en el siguiente algoritmo en la figura 19.

Figura 19. Diagrama de Flujo del Programa de Reservas del Laboratorio de Mecatrónica.

a





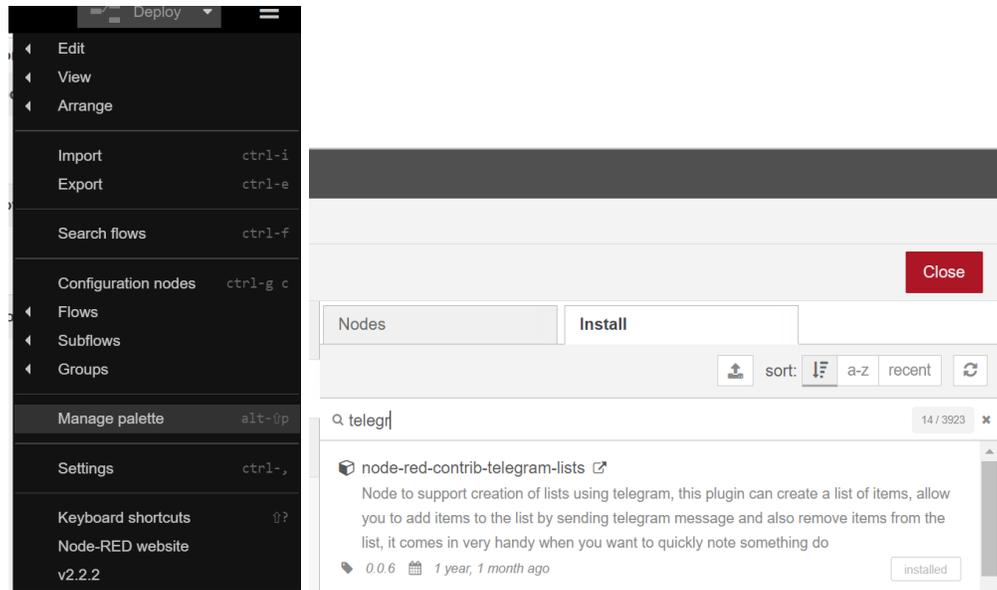
Nota. a) Diagrama de flujo de la programación. b) Diagrama de comunicación entre Telegram, Node-RED y XAMPP. Autoría propia.

Instalación de nodos de Telegram en Node-RED

El primer paso, para que la comunicación entre Telegram y Node-RED sea posible, fue instalar los nodos de Telegram en la plataforma de programación. En el botón de opciones, se buscó y se abrió la opción de ‘manage pallette’, ‘install’ y se escribió Telegram como se observa en la Figura 20. Seguido a eso se descargó los nodos de Telegram que se muestran en la Figura 21.

Figura 20.

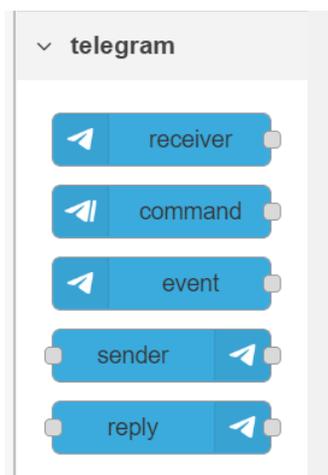
Descarga de Nodos de Telegram en Node-RED



Nota. Autoría propia.

Figura 21.

Nodos de Telegram



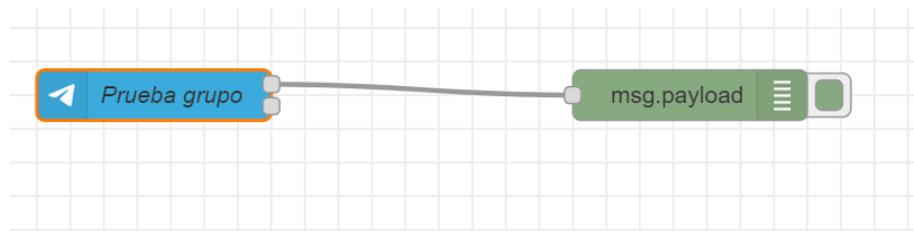
Nota. Autoría propia.

Comunicación básica entre Telegram App y Node-RED

El siguiente paso de configuración fue programar la recepción de mensajes desde Telegram hacia Node-Red como se observa en la Figura 22 y Figura 23. Para ello es necesario conocer el 'user_name' del bot y el token de seguridad que fue otorgado por el Bot_Father. La configuración y entrada de estos detalles se observa en la Figura 24.

Figura 22.

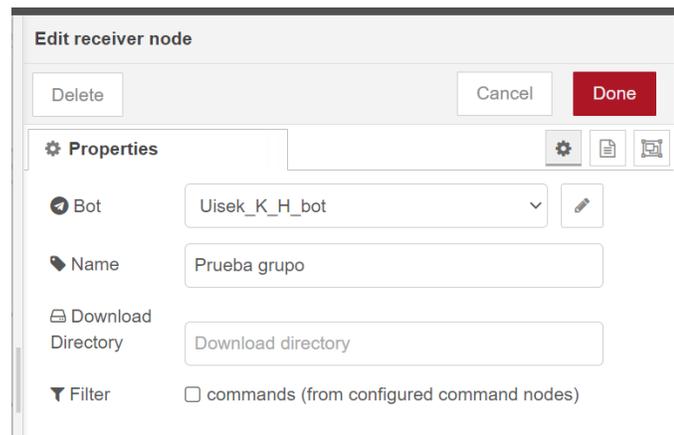
Flujo de Recepción de Mensajes de Telegram a Node-RED



Nota. Autoría propia.

Figura 23.

Configuración del Nodo 'Receiver'



Nota. Autoría propia.

Figura 24.*Configuración Avanzada del Nodo 'Receiver'*

Edit receiver node > Edit telegram bot node

Delete Cancel Update

Properties

Bot-Name: Uisek_K_H_bot

Token: 5168995401:AAEAe0BI_HH25KJ_o4tg7RqwoWO2oqW4Ew

Tip: If you don't have a token yet, you can create a new one here:
@BotFather.

Users: (Optional list of authorized user names e.g.: hugo,sepp,egon)

ChatIds: (Optional list of authorized chat-ids e.g.: -1234567,2345678,-3456789)

Server URL: (Optional URL for proxying and testing e.g.: https://api.telegram.org)

Update Mode: Polling

Nota. Autoría propia.

La descripción de los Nodos de la Figura 22 se muestra a continuación en la tabla 2.

Tabla 2.*Descripción de los Nodos en el Flujo de Recepción de Mensajes*

Nodo	Función
Prueba de grupo	Permite recibir mensajes de Telegram y guardarlos de forma de 'pay.load' en Node-RED
Msg.payload	Permite visualizar el valor en el pay.load en la consola de Node-RED para realizar un 'debug' de así necesitarlo.

Nota. Autoría propia.

A continuación, la Figura 25 y Figura 26 muestran la interacción entre Telegram y Node-RED para recibir mensajes.

Figura 25.

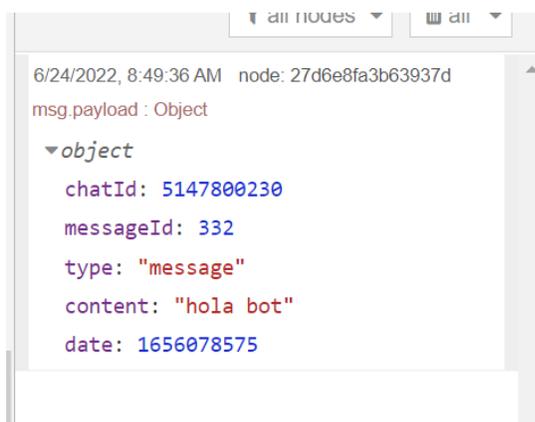
Mensajes Enviados desde Chat de Telegram con el Bot



Nota. Autoría propia.

Figura 26.

Recepción de mensaje en Node-RED



Nota. Autoría propia.

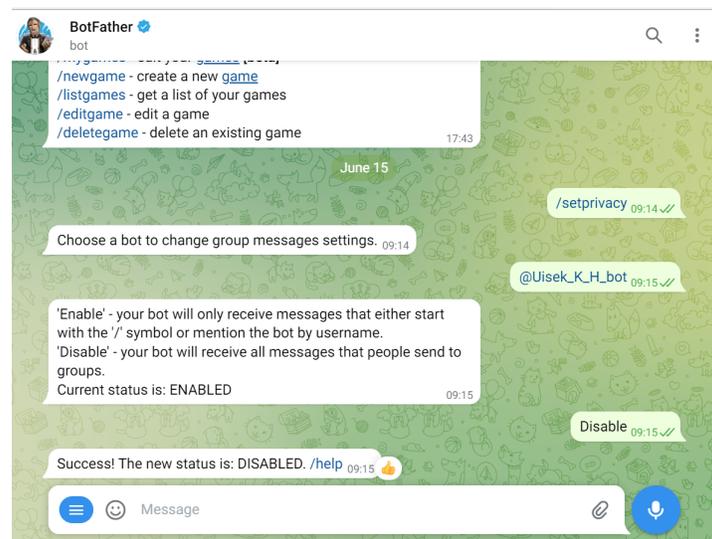
Configuración de bot desde BotFather

Una vez se recibió mensajes desde Telegram, se pudo crear un grupo en la plataforma de mensajería en donde se encuentre el bot. Es en este grupo donde se realizó el proceso para la

reserva del laboratorio. El bot por defecto se encontraba desactivado para recibir los mensajes del grupo. Lo que se hizo fue configurar esta opción desde el BotFather para que así el bot pueda recibir los mensajes que se envíen a grupos de Telegram donde el bot sea un integrante de la manera que lo muestra la Figura 27, Figura 28 y Figura 29.

Figura 27.

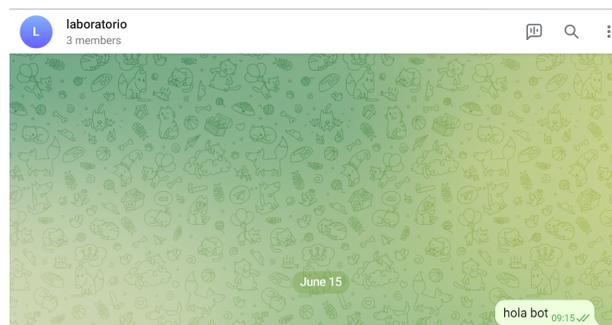
Habilitación de Bot en Grupos de Telegram



Nota. Autoría propia.

Figura 28.

Mensaje de Usuario a un Grupo de Telegram 'Laboratorio'



Nota. Autoría propia.

Figura 29.

Recepción de Mensaje en Node-RED

```
6/24/2022, 8:59:22 AM node: 27d6e8fa3b63937d
msg.payload : Object
▼ object
  chatId: -671403896
  messageId: 333
  type: "message"
  content: "hola bot"
  date: 1656079161
```

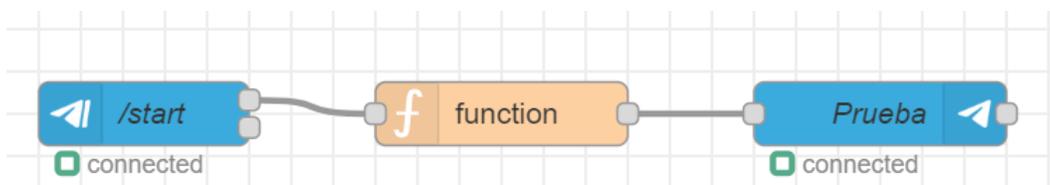
Nota. Autoría propia.

Configuración del Programa con un Grupo de Telegram

También se observó que el chatId de un grupo es diferente al chatId de una persona. El siguiente paso fue crear un flujo en Node-RED para enviar respuestas programadas a Telegram a través de comandos como se ve en la Figura 30. Estos comandos fueron procesados a través de un nodo, que activó una función para confirmar la lectura del comando enviado al grupo de Telegram llamado “Laboratorio”. La configuración del nodo ‘command’ se observa en la Figura 31. La configuración del nodo ‘function’ se observa en la Figura 32. La Figura 33 muestra la interacción entre Node-RED y Telegram y la Tabla 3 es una descripción de cada uno de los nodos en la Figura 30.

Figura 30.

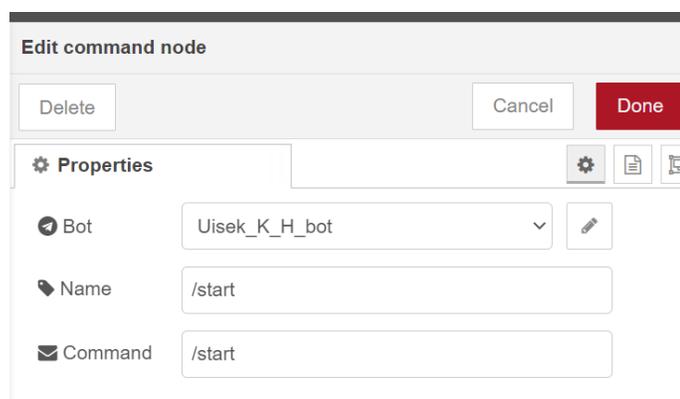
Recepción de Comando y Envío de Mensaje



Nota. Autoría propia.

Figura 31.

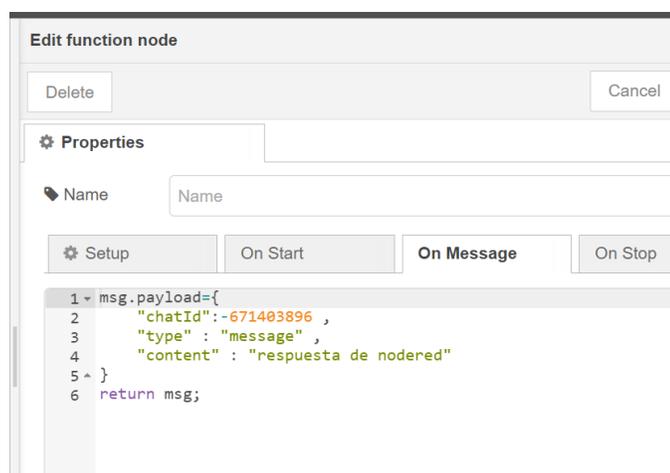
Configuración de Nodo 'Command'



Nota. Autoría propia.

Figura 32.

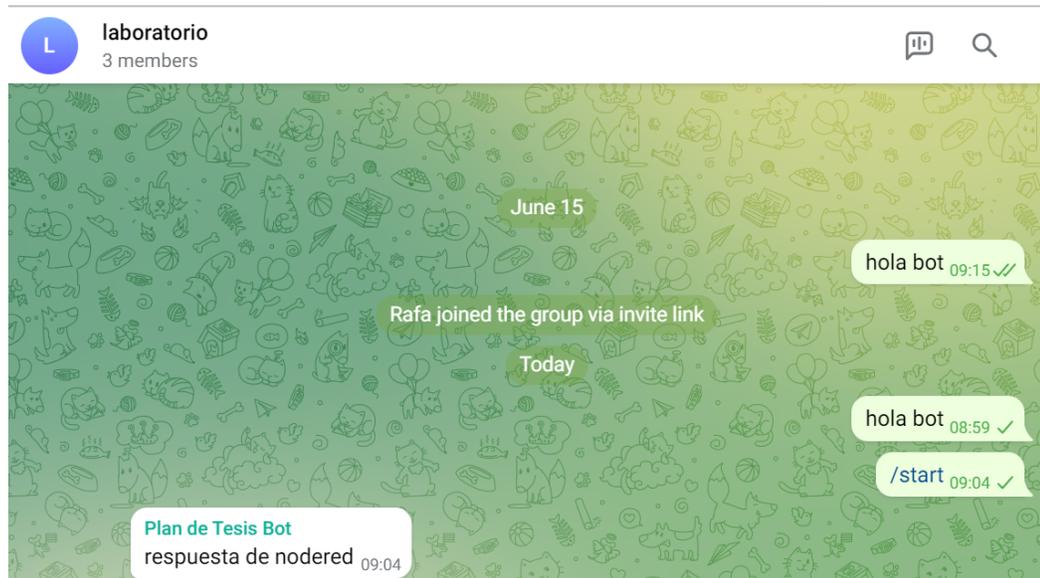
Configuración de nodo 'Function'



Nota. Autoría propia.

Figura 33.

Respuesta de Bot a través de Node-RED al Grupo de Telegram 'Laboratorio'



Nota. Autoría propia.

Tabla 3.

Descripción de Flujo Recibir – Enviar Mensaje

Nodo	Función
/start	Nodo 'command' permite recibir comandos de Telegram. La configuración de este nodo es igual a la del nodo 'receiver'
Function	Este nodo permite crear funciones usando lenguaje Node.js. En este flujo, el nodo 'function' crea un mensaje que lo guarda en payload del tipo 'message' con contenido: "Respuesta desde Node-RED y el chatId del grupo 'Laboratorio'"

Prueba	Nodo 'sender' permite enviar mensajes a Telegram. La configuración es la misma que el nodo 'receiver' y 'command'.
---------------	--

Nota. Autoría propia.

Configuración de la base de Datos y el Nodo MySQL

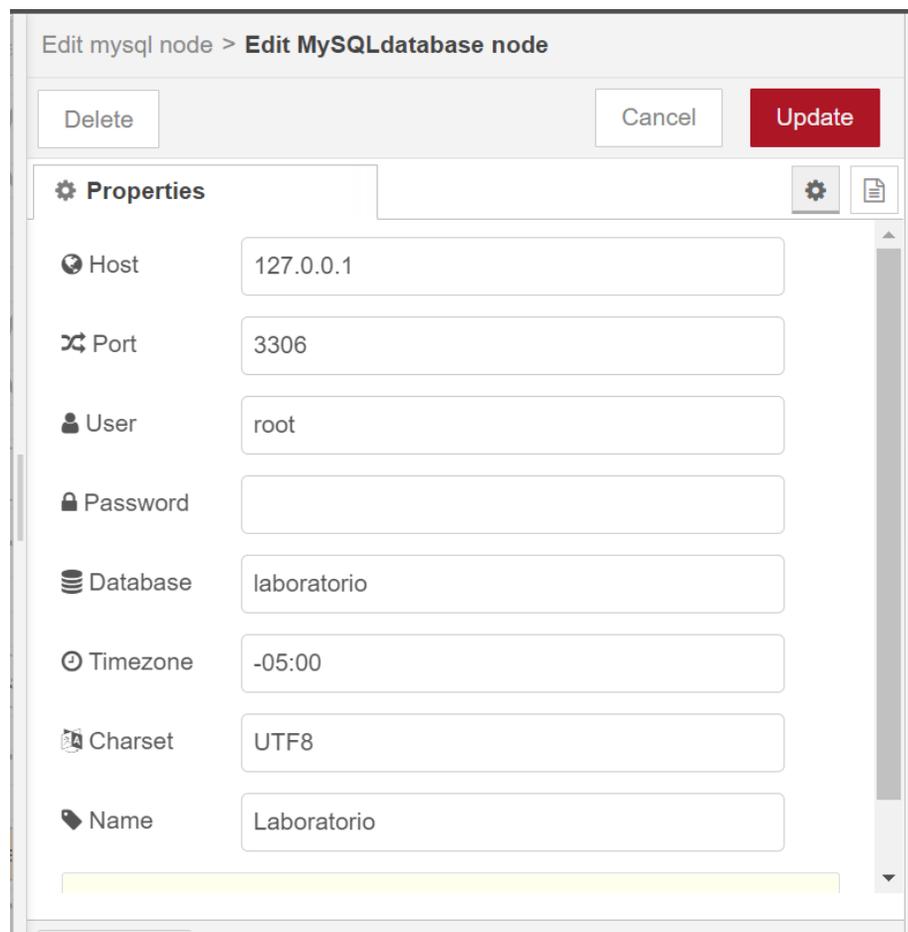
El nodo MySQL permite acceder a una base de datos MySQL. Este nodo es versátil y permite las operaciones de Insertar y Borrar datos desde Node-RED hasta la base de datos. El mensaje que se carga dentro de la base de datos debe contener la configuración msg.topic. A su vez, el resultado regresa de la base de datos a Node-RED como msg.payload. Cuando este resultado retorna es típicamente en la forma de una matriz de resultados. Algunos de los parámetros que pueden ser seteados en este nodo son la zona de tiempo, el tiempo de reconexión, entre otros.

Para este programa se configuraron los siguientes parámetros que se muestran en la figura 34. Host, Puerto (Port), User, Databse, Timezone. Para ingresar a la base de datos se presionó en el botón 'admin' junto a MySQL del software instalado XAMPP. Una vez que el software fue redirigido a localhost, que es la base de datos corriendo de manera local en el computador, se creó una nueva base de datos en la columna de opciones izquierda. Se asignó un nombre a la base de datos, y posteriormente se presionó el botón crear. En esta sección, el programa requirió que se le asigne también el número de columnas. Para este proyecto, las columnas asignadas fueron 3: "Nombre", "Fecha" y "Hora". Para poder crear esta tabla, también se le asignó un nombre a la misma. En el caso del proyecto el nombre de la tabla es "reservas". El siguiente paso fue asignar nombres y tipo de variables a las columnas de la base de datos. Algunos tipos de configuración que existen son número entero, VARCHAR, entre otros. Cuando la configuración de la tabla esté lista, se procedió a guardar la tabla para generarla.

Una vez la tabla fue creada, se procedió a poblar la misma con datos desde Node-RED. Estos datos que ingresaron a la base de datos son los nombres de los estudiantes que buscan realizar la reserva junto con la fecha y finalmente la hora en la que se desea realizar la reserva como se observa en la Figura 35.

Figura 34.

Configuración de nodo MySQL



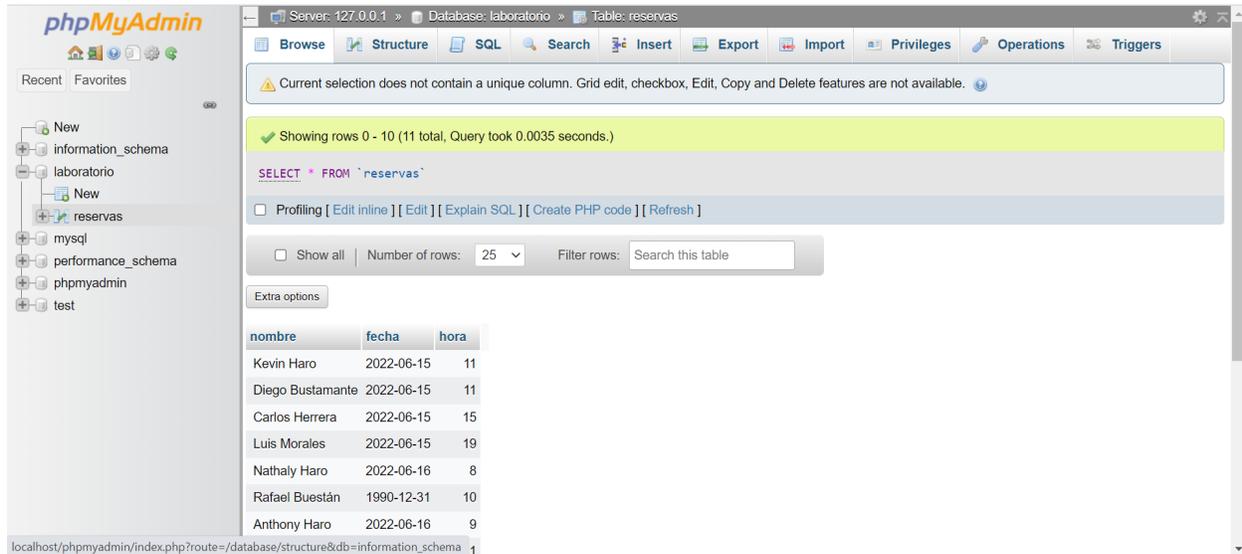
The image shows a screenshot of the Node-RED interface for configuring a MySQL database node. The window title is "Edit mysql node > Edit MySQLdatabase node". At the top, there are three buttons: "Delete", "Cancel", and "Update". Below the buttons is a "Properties" section with a gear icon and a document icon. The properties are listed as follows:

Property	Value
Host	127.0.0.1
Port	3306
User	root
Password	
Database	laboratorio
Timezone	-05:00
Charset	UTF8
Name	Laboratorio

Nota. Autoría Propia

Figura 35.

phpMyAdmin



Server: 127.0.0.1 » Database: laboratorio » Table: reservas

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 10 (11 total, Query took 0.0035 seconds.)

`SELECT * FROM `reservas``

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

nombre	fecha	hora
Kevin Haro	2022-06-15	11
Diego Bustamante	2022-06-15	11
Carlos Herrera	2022-06-15	15
Luis Morales	2022-06-15	19
Nathaly Haro	2022-06-16	8
Rafael Buestán	1990-12-31	10
Anthony Haro	2022-06-16	9

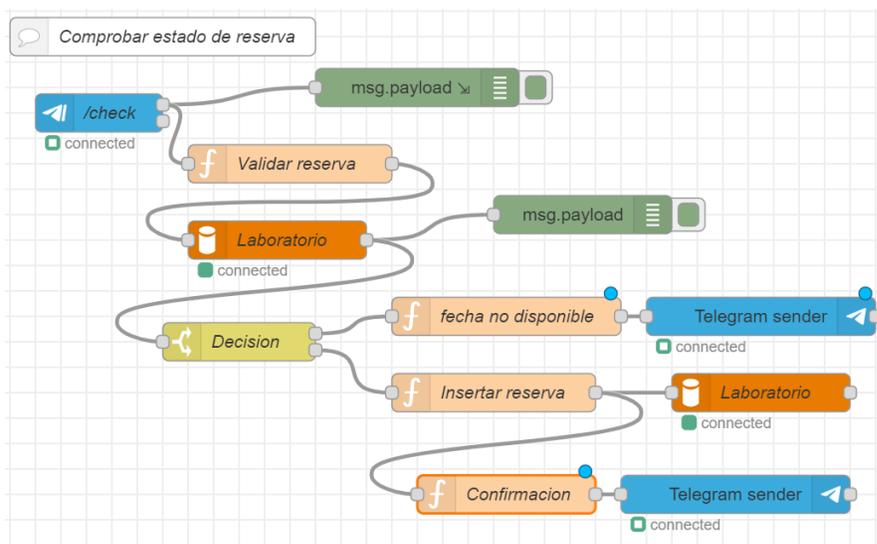
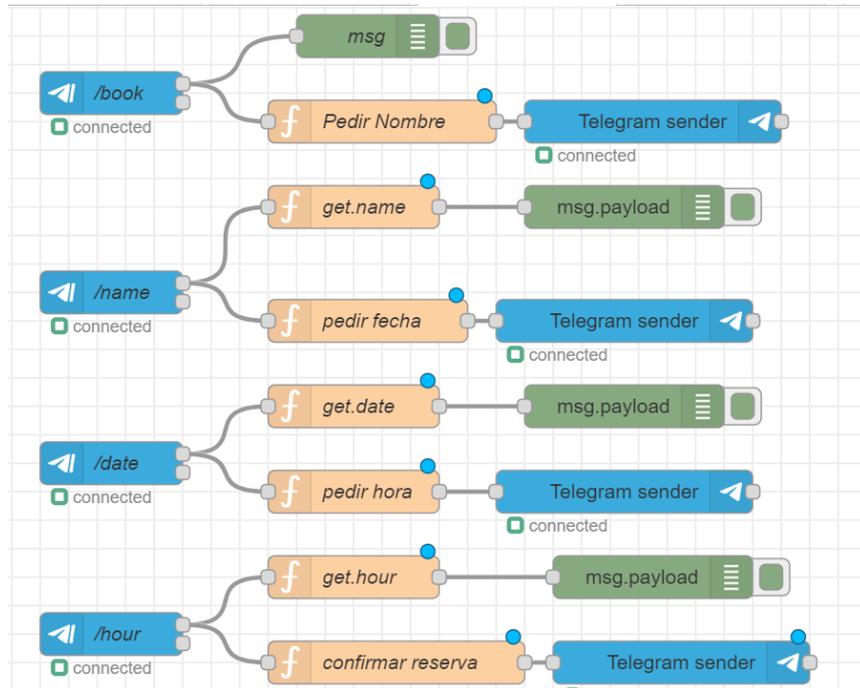
localhost/phpmyadmin/index.php?route=/database/structure&db=information_schema 1

Nota. Autoría Propia. Base de datos conteniendo las tres columnas creadas para almacenar datos de reserva.

Una vez que se configuró al bot y Node-RED para enviar y recibir mensajes al grupo “Laboratorio” se procedió a la configuración del programa en cuestión con los nodos que se muestra en la Figura 36. La finalidad del programa es reservar una hora en el Laboratorio de Mecatrónica con los siguientes datos: Nombre, fecha y hora. Si el espacio está disponible Node-RED guarda los datos en la base de datos de XAMPP y envía un mensaje de confirmación al grupo de Node-RED. Si el espacio ha sido reservado anteriormente, Node-RED no guarda el dato en XAMPP y pide al usuario cambiar la hora y/o la fecha deseada de reserva del laboratorio. La función y detalle de cada Nodo de la Figura 36 se muestran en la Tabla 4.

Figura 36.

Programación en Node-RED de Agendamiento de Laboratorio



Nota. Autoría propia.

Tabla 4.

Descripción de Flujo de Programación en Node-RED

Nodo	Función
------	---------

/book	Recibe solamente el comando '/book' desde Telegram
/name	Recibe el comando '/name' desde Telegram
/date	Recibe el comando '/date' desde Telegram
/hour	Recibe el comando '/hour' desde Telegram
/check	Recibe el comando '/hour' desde Telegram
Pedir nombre	Pay.load tipo 'message' con contenido que pide al usuario ingresar el comando '/name' más el nombre y apellido para la reserva
Get.name	Lee el payload junto al comando '/name' y lo guarda en una variable
Pedir fecha	Pay.load tipo 'message' con contenido que pide al usuario ingresar el comando '/date' más la fecha en formato yyyy-mm-dd para la reserva
Get.date	Lee el payload junto al comando '/date' y lo guarda en una variable
Pedir hora	Pay.load tipo 'message' con contenido que pide al usuario ingresar el comando '/hour' más la hora deseada para la reserva
Get.hour	Lee el payload junto al comando '/hour' y lo guarda en una variable
Confirmar reserva	Pay.load tipo 'message' con contenido que pide al usuario ingresar el comando '/check'
Validar reserva	Selecciona la hora y la fecha guardadas en las variables y las compara con las horas y fechas que estaban previamente en la base de datos de XAMPP
Decision	Si la fecha y hora guardadas en la variable son iguales a una fecha y hora en la base de datos asigna el valor de 1 y escoge la salida 1. Caso contrario asigna el valor de cero y escoge la salida 2.

Fecha no disponible	Payload tipo 'message' con contenido que avisa al usuario que la fecha y hora ya han sido elegidos previamente y pide al usuario ingresa otra fecha y/u hora.
Insertar reserva	Inserta las variables guardadas en nombre, fecha y hora en la base de datos de XAMPP
Confirmación	Payload tipo 'message' con contenido que avisa al usuario su reserva fue guardada exitosamente en la base de datos
Telegram sender	Envía el mensaje contenido en el payload del nodo anterior.
Msg.payload	Muestra lo que pasa en el nodo anterior en la consola de Node-RED
Laboratorio	Permite acceso básico a una base de datos MySQL

Nota. Autoría propia.

Resultados

En esta sección se muestra la interacción entre un usuario deseando reservar el laboratorio y la plataforma de reservas automáticas usando IoT, NodeRED y Telegram. Se presenta, de igual manera, dos posibles resultados. Cuando la reserva del usuario fue guardada con éxito y cuando la fecha deseada por el usuario para su reserva ya había sido tomada con antelación. En el segundo caso el bot sugerirá cambiar la fecha y/o la hora para intentar realizar otra reserva.

Para empezar, los estudiantes que buscan unirse al grupo de reservas tuvieron que escanear el código QR generado por el Telegram bot QRcoder que se muestra en la Figura 37. Una vez en el grupo, la descripción en Telegram indica el primer paso. Para empezar el proceso

de reserva se debe usar el comando ‘/book’ como se puede observar en la Figura 38 y continua en la Figura 39.

Figura 37.

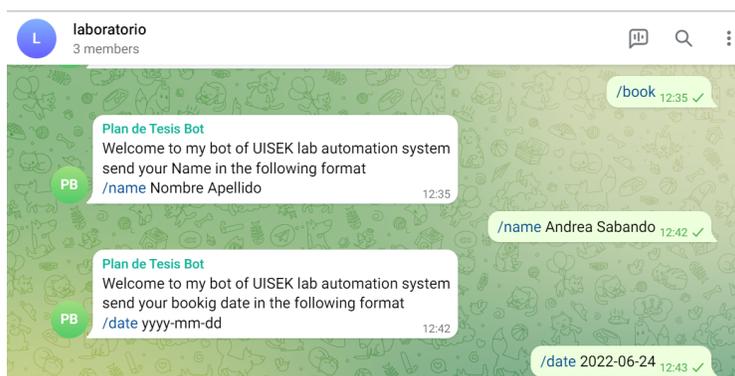
Código QR para unirse a Grupo de Telegram



Nota. Autoría propia.

Figura 38.

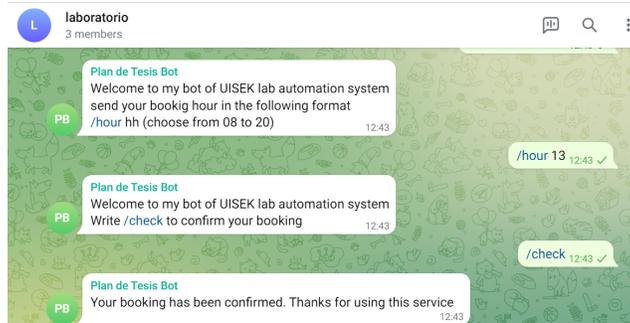
Inicialización de Reserva



Nota. Autoría propia.

Figura 39.

Reserva de Laboratorio



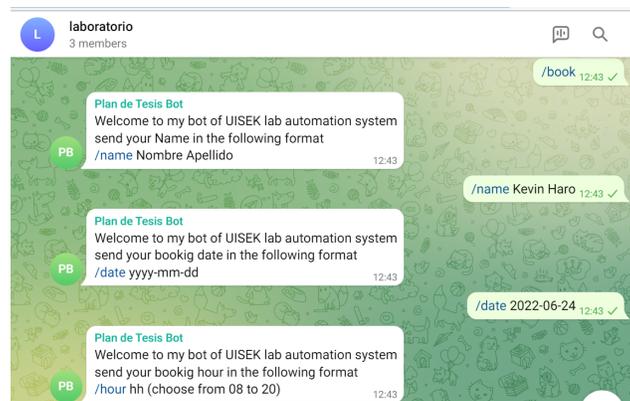
Nota. Autoría propia.

Se puede observar que, la estudiante con nombre Andrea Sabando ha podido reservar satisfactoriamente una sesión de laboratorio el viernes 24 de junio 2022 a las 14:00. La reserva fue realizada solo interactuando con Telegram, sin necesidad de manipular la programación en Node-RED o la base de datos en XAMPP. Tanto la programación como el seguimiento de resultados a través de la consola de Node-RED se encuentran en los anexos.

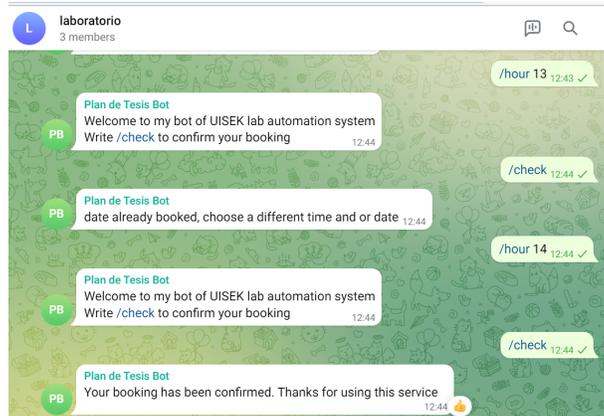
La siguiente prueba es realizar una reserva, con diferente nombre, pero con misma hora y fecha, mediante Telegram como se indica en la Figura 40 y Figura 41.

Figura 40.

Inicialización de Reserva con Nuevos Datos



Nota. Autoría propia.

Figura 41.*Doble reserva y nuevo agendamiento*

Nota. En esta figura se observa a un usuario que intento reservar el laboratorio en un horario que ya había sido tomado con antelación. El Bot indico a este usuario que debe seleccionar otro horario y/o fecha. Autoría propia.

Figura 42.*Base de Datos XAMMP*

The screenshot shows the phpMyAdmin interface. The left sidebar shows the database structure with 'laboratorio' selected. The main area displays the 'reservas' table with the following data:

nombre	fecha	hora
Kevin Haro	2022-06-15	11
Diego Bustamante	2022-06-15	11
Carlos Herrera	2022-06-15	15
Luis Morales	2022-06-15	19
Nathaly Haro	2022-06-16	8
Rafael Buestán	1990-12-31	10
Anthony Haro	2022-06-16	9
Pepito Perez	2022-06-16	11
Andrea Sabando	2022-06-24	13
Kevin Haro	2022-06-24	14

Nota. Autoría propia.

El programa se comporta de manera esperada. Cuando el estudiante Kevin Haro intenta reservar el laboratorio de mecatrónica a través del grupo de Telegram “Laboratorio” el día 24 de

junio 2022 a las 14:00, el alumno recibe el mensaje de que la hora fecha ya no están disponibles. El bot a su vez le recomienda escoger una nueva fecha o una nueva hora. Cuando el estudiante cambia la hora de 14:00 a 15:00, la reserva es procesada y guardada en la base de datos. Ambas reservas se pueden visualizar en la figura 42 junto con otras reservas anteriores.

Esto satisface la hipótesis planteada en el presente documento. NodeRed junto con Telegram son capaces de dar abasto a una plataforma de agendamiento y control de reservas del laboratorio de Mecatrónica de la Facultad de Ingenierías y Ciencias Aplicadas de la Universidad Internacional SEK. Conclusiones

Node-RED y Telegram permitieron desarrollar un programa autónomo y eficiente para automatizar el control de las reservas del laboratorio de Mecatrónica de la Facultad de Ingenierías y Ciencias Aplicadas de la Universidad Internacional SEK.

IoT como concepto fue aplicado durante el desarrollo de este proyecto de tesis, permitiendo que el proceso de reservas de laboratorio se automatice gracias al trabajo conjunto de NodeRED y Telegram. Estas dos aplicaciones permitieron realizar reservas a cualquier hora y obtener retroalimentación del estado de la reserva en tiempo real.

El código de programación con el que el programa planteado en este proyecto de tesis fue realizado en NodeRED cumpliendo así con el objetivo de trabajar con herramientas dedicadas al IoT y al IIoT.

El prototipo de sistema fue capaz de funcionar tanto en computadoras personales con un sistema operativo de Windows como en la computadora Raspberry Pi. El almacenamiento de los datos tomados para la reserva se pudo almacenar usando XAMPP.

La interacción entre usuario y Telegram es sencilla, fácil de seguir y entender, que permite obtener una respuesta con respecto al agendamiento del laboratorio en cuestión de segundos.

Recomendaciones

Este proyecto es innovador y el primero de su clase en la Universidad Internacional SEK y otras Universidades del Ecuador, con respecto a las herramientas usadas y el enfoque de IoT. Se encontró otro proyecto de tesis de la Universidad Central del Ecuador sin embargo, el estudio no usa las mismas herramientas que el presente proyecto; más bien se enfoca en diseño de páginas web. De esta manera se abren las puertas a que estudiantes en el futuro investiguen y amplíen el alcance de la combinación de plataformas como Node-RED y Telegram.

Siguiendo esta temática de automatización, se recomienda ampliar la aplicación de esta estrategia de agendamiento para temas de programación de mantenimientos preventivos y/o correctivos basándose en datos reales de uso. Por ejemplo, después de un cierto número de reservas del laboratorio, suponiendo que todos los estudiantes hayan asistido fielmente a la reserva de su lugar de trabajo, se puede agendar mantenimiento de las herramientas y maquinaria. De esta manera evitando que el personal de mantenimiento deba ir de vez en cuando a comprobar el estado de las herramientas y máquinas.

Otro uso que se puede dar a proyectos de esta índole es el de medición de fluidos dentro del laboratorio. Con la ayuda de sensores conectados a nodos y flujos en Node-RED se podría programar alarmas que se conecten directo a Telegram y alerte de valores escasos o peligrosamente altos de fluidos. Este mismo principio se puede aplicar a aceites y otros líquidos

que se usen en el laboratorio. Al medir la cantidad de líquido en el recipiente, se puede alertar al personal encargado de rellenar o comprar este elemento de manera remota y a tiempo real.

Adicionalmente, se sugiere extender este programa, no solamente al agendamiento del laboratorio a tiempo real y de manera remota por parte de los estudiantes, sino también proporcionar, a través de código, un control en tiempo real del docente a cargo del laboratorio. De esta manera el docente obtendría datos de las reservas para un día del laboratorio en tiempo real y con una pequeña interacción con la aplicación móvil de mensajería instantánea Telegram.

Referencias Bibliográficas

- Abed, S., Alyahya, N., & Altameem, A. (2019). IoT in Education: Its Impacts and Its Future in Saudi Universities and Educational Environments. *First International Conference on Sustainable Technologies for Computational Intelligence*, 47–62.
https://doi.org/10.1007/978-981-15-0029-9_5
- Abuarqoub, A., Abusaimh, H., Hammoudeh, M., Uliyan, D., Abu-Hashem, M. A., Murad, S., Al-Jarrah, M., & Al-Fayez, F. (2017). A Survey on Internet of Things Enabled Smart Campus Applications. *Proceedings of the International Conference on Future Networks and Distributed Systems*. <https://doi.org/10.1145/3102304.3109810>
- Abuhasel, K. A., & Khan, M. A. (2020). A Secure Industrial Internet of Things (IIoT) Framework for Resource Management in Smart Manufacturing. *IEEE Access*, 8, 117354–117364. <https://doi.org/10.1109/access.2020.3004711>
- Al-Emran, M., Malik, S. I., & Al-Kabi, M. N. (2019). A Survey of Internet of Things (IoT) in Education: Opportunities and Challenges. Toward Social Internet of Things (SIoT): Enabling Technologies, Architectures and Applications, 197–209.
https://doi.org/10.1007/978-3-030-24513-9_12
- Ban, Y., Okamura, K., & Kaneko, K. (2017). Effectiveness of Experiential Learning for Keeping Knowledge Retention in IoT Security Education. *2017 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*. <https://doi.org/10.1109/iiiai-aaai.2017.206>
- Boyes, H., Hallaq, B., Cunningham, J., & Watson, T. (2018). The industrial internet of things (IIoT): An analysis framework. *Computers in Industry*, 101, 1–12.
<https://doi.org/10.1016/j.compind.2018.04.015>

- Dargahi Nobari, A., Reshadatmand, N., & Neshati, M. (2017). Analysis of Telegram, An Instant Messaging Service. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. <https://doi.org/10.1145/3132847.3133132>
- Economides, A. A. (2017). User Perceptions of Internet of Things (IoT) Systems. *E-Business and Telecommunications*, 3–20. https://doi.org/10.1007/978-3-319-67876-4_1
- Grajales, T. (2000, March 27). *Tipos de Investigacion* [Slides]. Tgrajales. <https://cmapspublic2.ihmc.us/rid=1RM1F0L42-VZ46F4-319H/871.pdf>
- Hacks 360. (2019, November 26). *How to install Node-RED in windows 10* [Video]. YouTube. <https://www.youtube.com/watch?v=TbXU3PRDN1w>
- Herinyato, F. (2019). Character Education In The Era Of Industrial Revolution 4.0 And Its Relevance To The High School Learning Transformation Process. ResearchGate. https://www.researchgate.net/publication/338065420_Character_Education_In_The_Era_Of_Industrial_Revolution_40_And_Its_Relevance_To_The_High_School_Learning_Transformation_Process
- Howe, Andrew. “A Guide to Using Telegram with Node-Red and Home-Assistant | The Smarthome Book.” *The Smarthome Book | Simple Ideas to Assist with Your Smarthome Renovation*, 25 July 2021, www.thesmarthomebook.com/2020/10/13/a-guide-to-using-telegram-with-node-red-and-home-assistant.
- James, J. (2021, January 27). How computer programming became the worst choice of career. eFinancialCareers. <https://www.efinancialcareers.co.uk/news/finance/developer-jobs-terrible-career>

- Kravchenko, A. "How to Create a Telegram Reminder Bot Using Node-RED." Chatbots.Studio, 6 Aug. 2020, chatbots.studio/blog/creating-a-telegram-reminder-chatbot-using-node-red.
- Krishnamurthi, R. (2018). Teaching Methodology for IoT Workshop Course Using Node-RED. 2018 Eleventh International Conference on Contemporary Computing (IC3).
<https://doi.org/10.1109/ic3.2018.8530664>
- Lee, I., & Lee, K. (2015). The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Business Horizons*, 58(4), 431–440.
<https://doi.org/10.1016/j.bushor.2015.03.008>
- Lekić, M. & Gardašević, G. "IoT sensor integration to Node-RED platform," 2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH), 2018, pp. 1-5, doi: 10.1109/INFOTEH.2018.8345544.
- Majeed, A., & Ali, M. (2018). How Internet-of-Things (IoT) making the university campuses smart? QA higher education (QAHE) perspective. *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*.
<https://doi.org/10.1109/ccwc.2018.8301774>
- Meola, A. (2022, April 15). *Impact of IoT Technology on Education in 2022*. Insider Intelligence.
<https://www.insiderintelligence.com/insights/iot-technology-education/>
- Meola, A. (2022, February 3). Applications of IoT technology in the education sector for smarter schooling. Business Insider.
<https://www.businessinsider.com/iot-technology-education?international=true&r=US&IR=T#:~:text=The%20rise%20of%20mobile%20technology%20and%20the%20IoT%20allows%20schools,traditional%20stoic%20plans%20of%20yesteryear.>

- Muslih, M., Somantri, Supardi, D., Multipli, E., Nyaman, Y. M., Rismawan, A., & Gunawansyah. (2018). Developing Smart Workspace Based IOT with Artificial Intelligence Using Telegram Chatbot. *2018 International Conference on Computing, Engineering, and Design (ICCED)*. <https://doi.org/10.1109/icced.2018.00052>
- Node-RED. *Node-RED Programming Guide*. Node-RED by WordPress. <http://noderedguide.com/>
- Node-RED. (2021). OpenJS Foundation. Retrieved April 20, 2022, from <https://nodered.org/>
- Pérez, A. (2021, August 13). *Metodología de programación: definición, tipos y aplicación*. OBS Business School. Retrieved June 20, 2022, from <https://www.obsbusiness.school/blog/metodologia-de-programacion-definicion-tipos-y-aplicacion>
- Quillupangui, S. (2021, June 3). 37 universidades podrán retomar clases presenciales; asistencia no es obligatoria. *El Comercio*. Retrieved June 21, 2022, from <https://www.elcomercio.com/actualidad/clases-presenciales-universidades-pandemia-institutos.html>
- Rajalakshmi & H. Shahnasser, "Internet of Things using Node-Red and alexa," 2017 17th International Symposium on Communications and Information Technologies (ISCIT), 2017, pp. 1-4, doi: 10.1109/ISCIT.2017.8261194.
- Ramos-Galarza, C. (2021, February 24). Editorial: Diseños de investigación experimental. *CienciAmérica*, 10(1), 1–7. <https://doi.org/10.33210/ca.v10i1.356>
- Salah, B. (2021). Real-Time Implementation of a Fully Automated Industrial System Based on IR 4.0 Concept. *Actuators*, 10(12), 318. <https://doi.org/10.3390/act10120318>

- Sandoval, P. (2021, November 6). Baja de \$ 77 millones para universidades se explica por baja autogestión, retención del IVA y menores préstamos. *El Universo*.
<https://www.eluniverso.com/noticias/economia/cuales-son-las-universidades-que-recibiran-menos-recursos-en-2022-segun-la-proforma-presuestaria-nota/>
- “Send Messages Using Telegram and Node-RED – IoT with Us.” IoT with Us, 30 Nov. 2018, www.iotwithus.com/send-messages-using-telegram-and-node-red.
- SENESCYT & Rosales, B. (202–08-2). Analisis anual de los principales indicadores de educacion superior, ciencia, tecnologia e innovacion [Boletin Anual]. Boletin Anual, Quito, Ecuador.
https://www.educacionsuperior.gob.ec/wp-content/uploads/2020/09/Boletin_Anuual_Educacion_Superior_Ciencia_Tecnologia_Innovacion_Agosto2020.pdf
- Seymeliyski, K. (2020). *Sustainable Smart Campus* [Graph]. Burgas University.
https://unece.org/fileadmin/DAM/energy/se/pp/geee/WS_Nizhny_Novgorad_July_2019/12_Silvia_Letskovska_IoT.pdf
- Soumyalatha, S. G. H. (2016, May). *Study of IoT: understanding IoT architecture, applications, issues and challenges*. In 1st International Conference on Innovations in Computing & Net-working (ICICN16), CSE, RRCE. International Journal of Advanced Networking & Applications (Vol. 478).
- Telegram FAQ*. (n.d.). Telegram. Retrieved September 17, 2022, from <https://telegram.org/faq#q-what-is-telegram-what-do-i-do-here>
- Telegram Bot Platform*. (2015, June 24). Telegram. Retrieved April 5, 2022, from <https://telegram.org/blog/bot-revolution>

The ASEAN Post, & Nortajuddin, A. (2020, August 12). Is Cambodia Ready For Industry 4.0?

The ASEAN Post. Retrieved May 2, 2022, from

<https://theaseanpost.com/article/cambodia-ready-industry-40>

UpKeep. (2022). CMMS, EAM & IIoT Software by UpKeep Asset Operations Management |

Try Free. Retrieved May 2, 2022, from

<https://www.upkeep.com/learning/four-industrial-revolutions>

Anexos

Líneas de códigos de cada Bloque 'funcion' en Node-RED

Pedir Nombre:

```
var helpMessage = "Welcome to my bot of UISEK lab automation system\r\n";

helpMessage += "send your Name in the following format\r\n";

helpMessage += "/name Nombre Apellido"

helpMessage += "\r\n";

msg.payload.content = helpMessage;

return msg;
```

Get.name:

```
var name = msg.payload.content

flow.set("name", name);

msg.payload = flow.get("name")

return msg;
```

Pedir fecha:

```
var helpMessage = "Welcome to my bot of UISEK lab automation system\r\n";
```

```
helpMessage += "send your bookig date in the following format\r\n";
```

```
helpMessage += "/date yyyy-mm-dd";
```

```
helpMessage += "\r\n";
```

```
msg.payload.content = helpMessage;
```

```
return msg;
```

Get.date

```
var date = msg.payload.content
```

```
flow.set("fecha", date);
```

```
msg.payload = flow.get("fecha")
```

```
return msg;
```

Pedir hora:

```
var helpMessage = "Welcome to my bot of UISEK lab automation system\r\n";
```

```
helpMessage += "send your bookig hour in the following format\r\n";
```

```
helpMessage += "/hour hh (choose from 08 to 20)";
```

```
helpMessage += "\r\n";

msg.payload.content = helpMessage;

return msg;
```

Get.hour:

```
var hour = msg.payload.content

flow.set("hora", hour);

msg.payload = flow.get("hora")

return msg;
```

Confirmar reserva:

```
var helpMessage = "Welcome to my bot of UISEK lab automation system\r\n";

//helpMessage += "Thanks for using UISEK booking system";

helpMessage += "Write /check to confirm your booking";

helpMessage += "\r\n";

msg.payload.content = helpMessage;
```

```
return msg;
```

Insertar reserva:

```
msg.topic="INSERT INTO reservas (nombre,fecha,hora) VALUES ('"+flow.get("name")+","  
"+flow.get("fecha")+"," , '"+flow.get("hora")+");"
```

```
return msg;
```

Validar reserva:

```
msg.topic="SELECT count(1) numero FROM reservas WHERE FECHA =  
"+flow.get("fecha")+"" AND HORA = '"+flow.get("hora")+""";"
```

```
return msg;
```

Fecha no disponible:

```
msg.payload = {  
  
  "chatId":-671403896,  
  
  "type":"message",  
  
  "content":"date already booked, choose a different time and or date"  
  
}
```

```
return msg;
```

Insertar reserva:

```
msg.topic="INSERT INTO reservas (nombre,fecha,hora) VALUES ("'+flow.get("name")+'",  
"'"+flow.get("fecha")+"', "'"+flow.get("hora")+");"
```

```
return msg;
```

Confirmacion:

```
msg.payload = {
```

```
  "chatId":-671403896,
```

```
  "type":"message",
```

```
  "content":"Your booking has been confirmed. Thanks for using this service"
```

```
}
```

```
return msg;
```

Codigo en JSON

```
[
```

```
{
```

```
  "id": "cf9e73a0397e7b02",
```

```
  "type": "tab",
```

```
  "label": "Telegram receive and send commands ",
```

```
"disabled": false,
"info": "",
"env": []
},
{
  "id": "36b253873abe4e18",
  "type": "telegram command",
  "z": "cf9e73a0397e7b02",
  "name": "/book",
  "command": "/book",
  "description": "",
  "registercommand": false,
  "language": "",
  "scope": "default",
  "bot": "c469e0d20a762282",
  "strict": false,
  "hasresponse": true,
  "userregex": false,
  "removerexcommand": false,
  "outputs": 2,
  "x": 130,
  "y": 80,
  "wires": [
    [
      "9f251f44b2860b06",
      "8dd93615b9792005"
    ],
    []
  ]
}
```

```
},
{
  "id": "9f251f44b2860b06",
  "type": "debug",
  "z": "cf9e73a0397e7b02",
  "name": "msg",
  "active": true,
  "tosidebar": true,
  "console": false,
  "tostatus": false,
  "complete": "payload",
  "targetType": "msg",
  "statusVal": "",
  "statusType": "auto",
  "x": 310,
  "y": 40,
  "wires": []
},
{
  "id": "8dd93615b9792005",
  "type": "function",
  "z": "cf9e73a0397e7b02",
  "name": "Pedir Nombre",
  "func": "var helpMessage = \"Welcome to my bot of UISEK lab automation
system\\r\\n\";\\nhelpMessage += \"send your Name in the following format\\r\\n\";\\nhelpMessage +=
\\/name Nombre Apellido\\\" \\nhelpMessage += \"\\r\\n\";\\n\\n\\nmsg.payload.content =
helpMessage;\\nreturn msg;\",
  "outputs": 1,
  "noerr": 0,
  "initialize": "",
```

```
"finalize": "",
"libs": [],
"x": 320,
"y": 100,
"wires": [
  [
    "1c706b8d90a8b425"
  ]
]
},
{
  "id": "1c706b8d90a8b425",
  "type": "telegram sender",
  "z": "cf9e73a0397e7b02",
  "name": "",
  "bot": "c469e0d20a762282",
  "haserroroutput": false,
  "outputs": 1,
  "x": 510,
  "y": 100,
  "wires": [
    []
  ]
},
{
  "id": "ab1bbceaaa52378f",
  "type": "function",
  "z": "cf9e73a0397e7b02",
  "name": "get.name ",
```

```
"func": "var name = msg.payload.content\nflow.set(\"name\", name);\n\nmsg.payload =\nflow.get(\"name\")\nreturn msg;\n\n",
"outputs": 1,
"noerr": 0,
"initialize": "",
"finalize": "",
"libs": [],
"x": 300,
"y": 160,
"wires": [
  [
    "076a52c73c4fa05d"
  ]
]
},
{
  "id": "076a52c73c4fa05d",
  "type": "debug",
  "z": "cf9e73a0397e7b02",
  "name": "",
  "active": true,
  "tosidebar": true,
  "console": false,
  "tostatus": false,
  "complete": "false",
  "statusVal": "",
  "statusType": "auto",
  "x": 490,
  "y": 160,
```

```
"wires": []
},
{
  "id": "2d2f5aa460f9a154",
  "type": "function",
  "z": "cf9e73a0397e7b02",
  "name": "get.date",
  "func": "var date = msg.payload.content\nflow.set(\"fecha\", date);\n\nmsg.payload =
flow.get(\"fecha\")\nreturn msg;\n\n",
  "outputs": 1,
  "noerr": 0,
  "initialize": "",
  "finalize": "",
  "libs": [],
  "x": 300,
  "y": 300,
  "wires": [
    [
      "743edd4767051486"
    ]
  ]
},
{
  "id": "743edd4767051486",
  "type": "debug",
  "z": "cf9e73a0397e7b02",
  "name": "",
  "active": true,
  "tosidebar": true,
```

```
"console": false,
"toaststatus": false,
"complete": "false",
"statusVal": "",
"statusType": "auto",
"x": 490,
"y": 300,
"wires": []
},
{
  "id": "f6ec0375b3b545a6",
  "type": "function",
  "z": "cf9e73a0397e7b02",
  "name": "get.hour",
  "func": "var hour = msg.payload.content\nflow.set(\"hora\", hour);\n\nmsg.payload =
flow.get(\"hora\")\nreturn msg;\n\n",
  "outputs": 1,
  "noerr": 0,
  "initialize": "",
  "finalize": "",
  "libs": [],
  "x": 300,
  "y": 420,
  "wires": [
    [
      "e30445a302760315"
    ]
  ]
},
```

```
{
  "id": "e30445a302760315",
  "type": "debug",
  "z": "cf9e73a0397e7b02",
  "name": "",
  "active": true,
  "tosidebar": true,
  "console": false,
  "tostatus": false,
  "complete": "false",
  "statusVal": "",
  "statusType": "auto",
  "x": 510,
  "y": 420,
  "wires": []
},
{
  "id": "673452194fc493a8",
  "type": "telegram command",
  "z": "cf9e73a0397e7b02",
  "name": "/name",
  "command": "/name",
  "description": "",
  "registercommand": false,
  "language": "",
  "scope": "default",
  "bot": "c469e0d20a762282",
  "strict": false,
  "hasresponse": true,
```

```
"userregex": false,
"removeregexcommand": false,
"outputs": 2,
"x": 130,
"y": 220,
"wires": [
  [
    "ab1bbceea52378f",
    "8b1088deeba0f08d"
  ],
  []
]
},
{
  "id": "ee82b995cd6ebafb",
  "type": "telegram command",
  "z": "cf9e73a0397e7b02",
  "name": "/hour",
  "command": "/hour",
  "description": "",
  "registercommand": false,
  "language": "",
  "scope": "default",
  "bot": "c469e0d20a762282",
  "strict": false,
  "hasresponse": true,
  "userregex": false,
  "removeregexcommand": false,
  "outputs": 2,
```

```
"x": 130,
"y": 460,
"wires": [
  [
    "f6ec0375b3b545a6",
    "1b599dbbcd5a2288"
  ],
  []
]
},
{
  "id": "a45c4cd4f5cb1304",
  "type": "telegram command",
  "z": "cf9e73a0397e7b02",
  "name": "/date",
  "command": "/date",
  "description": "",
  "registercommand": false,
  "language": "",
  "scope": "default",
  "bot": "c469e0d20a762282",
  "strict": false,
  "hasresponse": true,
  "userregex": false,
  "removeregexcommand": false,
  "outputs": 2,
  "x": 130,
  "y": 340,
  "wires": [
```

```
[
  "2d2f5aa460f9a154",
  "6501542c1e248149"
],
[]
]
},
{
  "id": "74a47f17412ec7e1",
  "type": "function",
  "z": "cf9e73a0397e7b02",
  "name": "Insertar reserva",
  "func": "msg.topic=\"INSERT INTO reservas (nombre,fecha,hora) VALUES
(\""+flow.get(\"name\")+\"\", \""+flow.get(\"fecha\")+\"\", \""+flow.get(\"hora\")+\"\")\";\n\nreturn
msg;\n\n//payload.content\n//payload.content\n//2022\n//payload.type payload.content
\""+msg.payload.content[*]\"\",
  "outputs": 1,
  "noerr": 0,
  "initialize": "",
  "finalize": "",
  "libs": [],
  "x": 340,
  "y": 560,
  "wires": [
    [
      "1db9d7d04255a96f"
    ]
  ]
},
{
```

```
"id": "1db9d7d04255a96f",
"type": "mysql",
"z": "cf9e73a0397e7b02",
"mydb": "2db0982a091d39c8",
"name": "Laboratorio",
"x": 530,
"y": 540,
"wires": [
  [
    "75215badc320f24c"
  ]
]
},
{
  "id": "0132ac93dac120b1",
  "type": "inject",
  "z": "cf9e73a0397e7b02",
  "name": "",
  "props": [
    {
      "p": "payload"
    },
    {
      "p": "topic",
      "vt": "str"
    }
  ]
},
"repeat": "",
"crontab": ""
```

```
"once": false,
"onceDelay": 0.1,
"topic": "",
"payload": "",
"payloadType": "date",
"x": 140,
"y": 560,
"wires": [
  [
    "74a47f17412ec7e1"
  ]
]
},
{
  "id": "8b1088deeba0f08d",
  "type": "function",
  "z": "cf9e73a0397e7b02",
  "name": "pedir fecha",
  "func": "var helpMessage = \"Welcome to my bot of UISEK lab automation
system\\r\\n\\n\";\nhelpMessage += \"send your bookig date in the following
format\\r\\n\\n\";\nhelpMessage += \"/date yyyy-mm-dd\";\nhelpMessage +=
\"\\r\\n\\n\";\n\nmsg.payload.content = helpMessage;\nreturn msg;",
  "outputs": 1,
  "noerr": 0,
  "initialize": "",
  "finalize": "",
  "libs": [],
  "x": 310,
  "y": 240,
  "wires": [
```

```
[
  "857bc1f17c8d58b2"
]
]
},
{
  "id": "857bc1f17c8d58b2",
  "type": "telegram sender",
  "z": "cf9e73a0397e7b02",
  "name": "",
  "bot": "c469e0d20a762282",
  "haserroroutput": false,
  "outputs": 1,
  "x": 490,
  "y": 240,
  "wires": [
    []
  ]
},
{
  "id": "6501542c1e248149",
  "type": "function",
  "z": "cf9e73a0397e7b02",
  "name": "pedir hora",
  "func": "var helpMessage = `Welcome to my bot of UISEK lab automation
system\\r\\n`;\\nhelpMessage += `send your bookig hour in the following
format\\r\\n`;\\nhelpMessage += `/hour hh (choose from 08 to 20)`;\\nhelpMessage +=
`\\r\\n`;\\n\\nmsg.payload.content = helpMessage;\\nreturn msg;",
  "outputs": 1,
  "noerr": 0,
```

```
"initialize": "",
"finalize": "",
"libs": [],
"x": 300,
"y": 360,
"wires": [
  [
    "00bdc5c6ef673189"
  ]
]
},
{
  "id": "00bdc5c6ef673189",
  "type": "telegram sender",
  "z": "cf9e73a0397e7b02",
  "name": "",
  "bot": "c469e0d20a762282",
  "haserroroutput": false,
  "outputs": 1,
  "x": 490,
  "y": 360,
  "wires": [
    []
  ]
},
{
  "id": "1b599dbbcd5a2288",
  "type": "function",
  "z": "cf9e73a0397e7b02",
```

```
"name": "confirmar reserva",

"func": "var helpMessage = \"Welcome to my bot of UISEK lab automation
system\\r\\n\";\\n//helpMessage += \"Thanks for using UISEK booking system\";\\nhelpMessage +=
\\\"Write /check to confirm your booking\";\\nhelpMessage += \"\\r\\n\";\\n\\n\\nmsg.payload.content =
helpMessage;\\nreturn msg;\",

"outputs": 1,

"noerr": 0,

"initialize": "",

"finalize": "",

"libs": [],

"x": 330,

"y": 480,

"wires": [

  [

    "6ddd9c67ba5add8d"

  ]

]

},

{

  "id": "6ddd9c67ba5add8d",

  "type": "telegram sender",

  "z": "cf9e73a0397e7b02",

  "name": "",

  "bot": "c469e0d20a762282",

  "haserroroutput": false,

  "outputs": 1,

  "x": 530,

  "y": 480,

  "wires": [

    []

  ]

}
```

```
]
},
{
  "id": "c9d32cc3d4db583c",
  "type": "function",
  "z": "cf9e73a0397e7b02",
  "name": "fecha no disponible",
  "func": "msg.payload = {\n  \"chatId\":-671403896,\n  \"type\": \"message\", \n  \"content\": \"date already booked, choose a different time and or date\"\n}\nreturn msg; ",
  "outputs": 1,
  "noerr": 0,
  "initialize": "",
  "finalize": "",
  "libs": [],
  "x": 410,
  "y": 860,
  "wires": [
    [
      "cda5c88eb2dfdbec"
    ]
  ]
},
{
  "id": "cda5c88eb2dfdbec",
  "type": "telegram sender",
  "z": "cf9e73a0397e7b02",
  "name": "",
  "bot": "c469e0d20a762282",
  "haserroroutput": false,
```

```
"outputs": 1,
"x": 610,
"y": 860,
"wires": [
  []
]
},
{
  "id": "982e6b290f1eddb3",
  "type": "debug",
  "z": "cf9e73a0397e7b02",
  "name": "",
  "active": true,
  "tosidebar": true,
  "console": true,
  "tostatus": false,
  "complete": "payload",
  "targetType": "msg",
  "statusVal": "",
  "statusType": "auto",
  "x": 340,
  "y": 680,
  "wires": []
},
{
  "id": "c88e560248695978",
  "type": "mysql",
  "z": "cf9e73a0397e7b02",
  "mydb": "2db0982a091d39c8",
```

```
"name": "Laboratorio",
"x": 230,
"y": 800,
"wires": [
  [
    "79af6fa1f1f5aec6",
    "bc493d3c6f74212d"
  ]
]
},
{
  "id": "af25d91b7a1c03ab",
  "type": "function",
  "z": "cf9e73a0397e7b02",
  "name": "Validar reserva",
  "func": "msg.topic=\"SELECT count(1) numero FROM reservas WHERE FECHA =
'+flow.get(\"fecha\")+'\" AND HORA = '"+flow.get(\"hora\")+'\";\\n//return
msg;\\n//msg.topic=\"SELECT count(1) numero FROM reservas WHERE FECHA = '2022-06-12' AND HORA
= '11'\"; \\nreturn msg;\"",
  "outputs": 1,
  "noerr": 0,
  "initialize": "",
  "finalize": "",
  "libs": [],
  "x": 240,
  "y": 740,
  "wires": [
    [
      "c88e560248695978"
    ]
  ]
}
```

```
]
},
{
  "id": "03b34c045cbd8ede",
  "type": "function",
  "z": "cf9e73a0397e7b02",
  "name": "Insertar reserva",
  "func": "msg.topic=\"INSERT INTO reservas (nombre,fecha,hora) VALUES
(\\'+flow.get(\"name\\")+\\', '\\'+flow.get(\"fecha\\")+\\', '\\'+flow.get(\"hora\\")+\\');\\\"\\nreturn msg;",
  "outputs": 1,
  "noerr": 0,
  "initialize": "",
  "finalize": "",
  "libs": [],
  "x": 400,
  "y": 920,
  "wires": [
    [
      "4d3a5f9dda4ce33f",
      "13d5aa4631113bc5"
    ]
  ]
}
{
  "id": "79af6fa1f1f5aec6",
  "type": "debug",
  "z": "cf9e73a0397e7b02",
  "name": "",
  "active": true,
```

```
"tosidebar": true,
"console": false,
"toastatus": false,
"complete": "payload",
"targetType": "msg",
"statusVal": "",
"statusType": "auto",
"x": 470,
"y": 780,
"wires": []
},
{
  "id": "bc493d3c6f74212d",
  "type": "switch",
  "z": "cf9e73a0397e7b02",
  "name": "Decision",
  "property": "payload[0].numero",
  "propertyType": "msg",
  "rules": [
    {
      "t": "gte",
      "v": "1",
      "vt": "num"
    },
    {
      "t": "eq",
      "v": "0",
      "vt": "num"
    }
  ]
}
```

```
],  
  "checkall": "true",  
  "repair": false,  
  "outputs": 2,  
  "x": 200,  
  "y": 880,  
  "wires": [  
    [  
      "c9d32cc3d4db583c"  
    ],  
    [  
      "03b34c045cbd8ede"  
    ]  
  ]  
},  
{  
  "id": "4d3a5f9dda4ce33f",  
  "type": "mysql",  
  "z": "cf9e73a0397e7b02",  
  "mydb": "2db0982a091d39c8",  
  "name": "Laboratorio",  
  "x": 610,  
  "y": 920,  
  "wires": [  
    []  
  ]  
},  
{  
  "id": "0d3511efe76e3ff3",
```

```
"type": "comment",
"z": "cf9e73a0397e7b02",
"name": "Insertar reserva Manual ",
"info": "",
"x": 120,
"y": 520,
"wires": []
```

```
},
```

```
{
```

```
"id": "8f8af5992c7cb714",
"type": "telegram command",
"z": "cf9e73a0397e7b02",
"name": "/check",
"command": "/check",
"description": "",
"registercommand": false,
"language": "",
"scope": "default",
"bot": "c469e0d20a762282",
"strict": false,
"hasresponse": true,
"userregex": false,
"removeregexcommand": false,
"outputs": 2,
"x": 90,
"y": 700,
"wires": [
  [
    "982e6b290f1eddb3",
```

```
    "af25d91b7a1c03ab"
  ],
  []
]
},
{
  "id": "75215badc320f24c",
  "type": "debug",
  "z": "cf9e73a0397e7b02",
  "name": "",
  "active": true,
  "tosidebar": true,
  "console": false,
  "tostatus": false,
  "complete": "false",
  "statusVal": "",
  "statusType": "auto",
  "x": 570,
  "y": 600,
  "wires": []
},
{
  "id": "13d5aa4631113bc5",
  "type": "function",
  "z": "cf9e73a0397e7b02",
  "name": "Confirmacion ",
  "func": "msg.payload = {\n  \"chatId\":-671403896,\n  \"type\": \"message\", \n\n  \"content\": \"Your booking has been confirmed. Thanks for using this service\\n\\n\"}\nreturn msg; ",
  "outputs": 1,
```

```
"noerr": 0,
"initialize": "",
"finalize": "",
"libs": [],
"x": 410,
"y": 1000,
"wires": [
  [
    "df8020103a5870ab"
  ]
],
},
{
  "id": "df8020103a5870ab",
  "type": "telegram sender",
  "z": "cf9e73a0397e7b02",
  "name": "",
  "bot": "c469e0d20a762282",
  "haserroroutput": false,
  "outputs": 1,
  "x": 590,
  "y": 1000,
  "wires": [
    []
  ]
},
{
  "id": "e16dde103cf101d6",
  "type": "comment",
```

```
"z": "cf9e73a0397e7b02",
"name": "Comprobar estado de reserva ",
"info": "",
"x": 140,
"y": 640,
"wires": []
},
{
  "id": "c469e0d20a762282",
  "type": "telegram bot",
  "botname": "Uisek_K_H_bot",
  "usernames": "",
  "chatids": "",
  "baseapiurl": "",
  "updatemode": "polling",
  "pollinterval": "300",
  "usesocks": false,
  "sockshost": "",
  "socksport": "6667",
  "socksusername": "anonymous",
  "sockspassword": "",
  "bothost": "",
  "botpath": "",
  "localbotport": "8443",
  "publicbotport": "8443",
  "privatekey": "",
  "certificate": "",
  "useselfsignedcertificate": false,
  "sslterminated": false,
```

```
"verboseLogging": false
},
{
  "id": "2db0982a091d39c8",
  "type": "MySQLdatabase",
  "name": "Laboratorio",
  "host": "127.0.0.1",
  "port": "3306",
  "db": "laboratorio",
  "tz": "-05:00",
  "charset": "UTF8"
}
]
```