



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA
CARRERA DE INGENIERÍA DE SISTEMAS

**IMPLEMENTACIÓN DE UN SERVIDOR DE ADMINISTRACIÓN Y GESTIÓN DE
UNA RED LORAWAN E INTEGRACIÓN CON AMAZON WEB SERVICE**

Trabajo de titulación previo a la obtención del
título de Ingeniero de Sistemas

AUTORES: BRYAN PATRICIO CARRASCO ILLESCAS
LUIS FERNANDO DELEG AGUILAR
TUTOR: ING. ERWIN JAIRO SACOTO CABRERA, PhD.

Cuenca - Ecuador
2022

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Bryan Patricio Carrasco Illescas con documento de identificación N° 0105911853 y Luis Fernando Deleg Aguilar con documento de identificación N° 0105135446; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Cuenca, 27 de julio del 2022

Atentamente,

Bryan Patricio Carrasco Illescas

0105911853

Luis Fernando Deleg Aguilar

0105135446

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, Bryan Patricio Carrasco Illescas con documento de identificación N° 0105911853 y Luis Fernando Deleg Aguilar con documento de identificación N° 0105135446, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto técnico: “Implementación de un servidor de administración y gestión de una red LoRaWAN e integración con Amazon Web Service”, el cual ha sido desarrollado para optar por el título de: Ingeniero de Sistemas, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, 27 de julio del 2022

Atentamente,

Bryan Patricio Carrasco Illescas

0105911853

Luis Fernando Deleg Aguilar

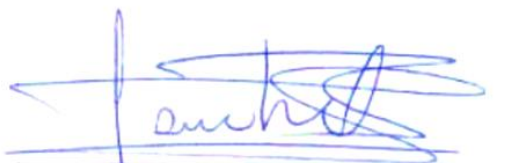
0105135446

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Erwin Jairo Sacoto Cabrera con documento de identificación N° 0301185229, docente de la Universidad Politécnica Salesiana declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: IMPLEMENTACIÓN DE UN SERVIDOR DE ADMINISTRACIÓN Y GESTIÓN DE UNA RED LORAWAN E INTEGRACIÓN CON AMAZON WEB SERVICE, realizado por Bryan Patricio Carrasco Illescas con documento de identificación N° 0105911853 y por Luis Fernando Deleg Aguilar con documento de identificación N° 0105135446, obteniendo como resultado final el trabajo de titulación bajo la opción de Proyecto técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 27 de julio del 2022

Atentamente,



Ing. Erwin Jairo Sacoto Cabrera, PhD.

0301185229

DEDICATORIA

Dedico esta tesis a mis padres Rocío y Luis por el apoyo incondicional que brindaron durante todo este lapso de tiempo.

A mis hermanos Adriana y Raúl, y a mi sobrino Jhosue que estuvieron en todo momento apoyándome y regalándome su compañía para seguir adelante.

A todos los que me apoyaron para escribir y concluir esta tesis, estando en buenos y malos momentos.

Para ellos es esta dedicatoria de tesis, pues es a ellos a quienes se las debo por su apoyo incondicional.

Y de manera especial al PhD. Erwin Sacoto director de Tesis por su guía durante todo el desarrollo del trabajo de titulación, por su paciencia y predisposición para ayudar en todos los inconvenientes y dudas.

Luis Fernando Deleg Aguilar

Por este trabajo principalmente quiero agradecer principalmente a Dios y sobre todo de manera especial a mi papá Mauro y mis abuelitos Beatriz y Bolívar por el cariño y apoyo brindado en todo momento.

A mi prima Lizbeth, a mis tíos y tías que estuvieron en todo momento brindándome su apoyo incondicional para seguir adelante.

A mis amigos y compañeros por cada experiencia compartida a lo largo de toda la carrera universitaria.

Y especialmente al PhD. Erwin Sacoto director de Tesis por su asesoría durante todo el desarrollo del trabajo de titulación, por su paciencia y predisposición para ayudar en todos los inconvenientes y dudas.

Bryan Patricio Carrasco Illescas

Resumen

En la ciudad de Cuenca se realizan las mediciones de agua urbana de manera manual, es decir varias personas recorren la ciudad realizando el registro de consumo, lo descrito no permite a las empresas tener un registro actualizado para la facturación del servicio, así como la generación de facturas en base al consumo mensual de agua urbana, es por ello que gracias al Internet de las Cosas (IoT) se puede automatizar estas mediciones y tener un proceso mucho más rápido y preciso.

La forma para llegar a solventar esta problemática se la realiza con medidores de agua inteligente que se conectan a una red LoRaWAN para que se puedan obtener los datos que se registran en los medidores inteligentes.

Por esta razón se propuso implementar un servidor de administración y gestión de una red LoRaWAN e integración con Amazon Web Service, de tal manera que se puedan calcular y presentar los datos del consumo de agua en un dashboard que se realiza en el servicio de QuickSight de Amazon el cual nos indicará el total de consumo de agua en metros cúbicos que se utilizan en el día a día.

Palabras clave: LoRaWAN, ChirpStack, Medidores Inteligentes, Amazon Web Service, PostgreSQL, Servidor, Dashboard, Integración, Gateway, QuickSight.

Abstract

In the city of Cuenca, urban water measurements are performed manually, i.e. several people go around the city to record consumption, which does not allow companies to have an updated record for billing the service, as well as the generation of invoices based on the monthly consumption of urban water, which is why thanks to the Internet of Things (IoT) these measurements can be automated and have a much faster and more accurate process.

The way to solve this problem is with smart water meters that are connected to a LoRaWAN network so that the data recorded in the smart meters can be obtained.

For this reason it was proposed to implement a server for administration and management of a LoRaWAN network and integration with Amazon Web Service, so that water consumption data can be calculated and presented in a dashboard that is done in Amazon's QuickSight service, which will indicate the total water consumption in cubic meters that are used on a daily basis.

Keywords: LoRaWAN, ChirpStack, Smart Meters, Amazon Web Service, PostgreSQL, Server, Dashboard, Integration, Gateway, QuickSight

ÍNDICE

I INTRODUCCIÓN	16
II PROBLEMA	17
2.1 Antecedentes	17
2.2 Importancia.....	18
2.3 Alcance.....	19
III OBJETIVOS	20
3.1 General	20
3.2 Específicos	20
IV REVISIÓN DE LA LITERATURA O FUNDAMENTOS TEÓRICOS	21
4.1 IoT.....	21
4.2 Redes LPWAN.....	22
4.2.1 SigFoX	22
4.2.2 Arquitectura SigFoX	23
4.2.3 NB-IoT	23
4.2.4 Arquitectura NB-IoT	24
4.3 LoRaWAN	25
4.3.1 Tipos de Dispositivos LoRaWAN	26
4.3.2 Parámetros de Comunicación	27

4.3.3	Estructura del Paquete LoRa.....	28
4.3.4	Arquitectura de red LoRaWAN.....	29
4.4	Servidor de Aplicaciones.....	30
4.4.1	ChirpStack.....	30
4.4.2	ChirpStack Gateway Bridge, Network Server y Application Server	31
4.5	Amazon Web Service (AWS).....	32
4.5.1	Servicios AWT IoT.....	32
4.5.2	Servicios de Análisis	32
4.6	Medidores Inteligentes	33
4.7	Comunicación entre cliente y servidor	34
4.8	Herramientas que posee Amazon Web Service.	34
4.8.1	AWS Cloud9.....	35
4.8.2	AWS CodeCommit	36
4.8.3	AWS CodeBuild	37
4.8.4	AWS CodeDeploy	37
4.8.5	AWS CodePipeline	38
4.8.6	AWS CodeStar.....	39
4.9	Seguridad y autenticación dentro del servidor LORA.....	40
4.9.1	Activation By Personalization (ABP).....	40
4.9.2	Over-The-Air Activation (OTAA).....	40

V MARCO METODOLÓGICO 42

5.1	Implementación de un servidor de ChirpStack para la gestión de los dispositivos IoT.....	42
5.2	Instalación de Chirpstack Gateway Bridge.....	47
5.3	Instalación de Chirpstack Network Server	48
5.4	Instalación de Chirpstack Application Server	50
5.5	Implementación de la autenticación en los dispositivos de la red LoRaWAN.	57

5.6	Implementación de ChirpStack en Amazon Web Services.....	61
5.6.1	Atributos del mensaje	63
5.7	Implementar Amazon Web Service para obtener la data de ChirpStack	63
5.7.1	Creación de cuenta en Amazon Web Service.	63
5.7.2	Ingreso a Amazon Web Services (AWS).....	69
5.7.3	Configuración de Simple Notification Service (SNS).....	71
5.7.4	Configuración de Simple Queue Service (SQS)	73
5.7.5	Conectar "Simple Notification Service (SNS)" a "Simple Queue Service (SQS)"	78
5.8	Obtener una mejor visualización de todos los datos que se tienen de ChirpStack.	81
5.9	Analizar las herramientas de Amazon Web Service para almacenamiento, análisis y visualización de datos.....	82
5.10	Analizar y visualizar los datos de manera eficiente utilizando la herramienta QuickSight.....	87
5.10.1	Configuración en la consola del servidor ChirpStack	87
5.10.2	Comprobación de la conexión remota desde pgAdmin	89
5.11	Configuración de pgAgent para la automatización de trabajos.....	90
5.11.1	Instalación y configuración del servicio pgAgent.....	91
5.11.2	Creación de Jobs en pgAgent para automatizar la base de datos 'chirp-stack_as_events'	93
5.12	Configuración de QuickSight para la visualización de los datos obtenidos en ChirpStack.....	121

VI RESULTADOS 135

6.1	Ejecución de los planes de pruebas.	135
6.1.1	Prueba de funcionalidad de la plataforma ChirpStack.....	139
6.1.2	Prueba de funcionalidad SQS(AWS).....	140
6.1.3	Prueba de funcionalidad QuickSight	141

VII CRONOGRAMA	144
VIII PRESUPUESTO	147
IX CONCLUSIONES	149
X RECOMENDACIONES	151

Índice de tablas

4.1	Tipos de LoRa	27
4.2	Servicios de ChirpStack	31
7.1	Cronograma de actividades	146
8.1	Presupuesto del proyecto	148

Índice de figuras

4.1	Arquitectura SigFoX Martres (2022).....	23
4.2	Arquitectura NB-IoT Saiz Miranda (2019)	24
4.3	Protocolo de Comunicación LoRaWAN Lavric and Petrariu (2018).....	26
4.4	Estructura del Paquete LoRa Guambuquete Gómez and Soledispa Reyes (2021)	28
4.5	Arquitectura de red LoRaWAN Guambuquete Gómez and Soledispa Reyes (2021)	29
4.6	Arquitectura de Servicios de ChirpStack Chirpstack.io (2022a)	31
4.7	Herramientas de AWS. Devalal and Karthikeyan (2018)	35
4.8	AWS CodeCommit. Pothecary (2021)	36
4.9	Funcionamiento de CodePipeline. Inc. (2022c).....	39
4.10	Activación ABP y OTAA Natalia (2019).	41
5.1	Instalación del protocolo MQTT y la base de datos en postgresql.	43
5.2	Configuración en la base de datos de PostgreSQL.	43
5.3	Creación del rol "chirpstack_as".	44
5.4	Creación del rol "chirpstack_ns".	44
5.5	Creación de la base de datos "chirpstack_as".	44
5.6	Creación de la base de datos "chirpstack_ns".	44
5.7	Ingreso a la base de datos "chirpstack_as".	45
5.8	Creación de la extensión "pg_trgm".	45
5.9	Creación de la extensión hstore.	45
5.10	Finalización a la conexión de base de datos.	45

5.11	Configuración del repositorio de chirpstack.	46
5.12	Configuración de la llave del servidor de chirpstack.	46
5.13	Visualización de los diferentes paquetes de chirpstack.	46
5.14	Actualización de la lista de paquetes disponibles.	47
5.15	Instalación del paquete "chirpstack-gateway-bridge".	47
5.16	Inicio del servicio de "chirpstack-gateway-bridge".	48
5.17	Habilitación del servicio de "chirpstack-gateway-bridge".	48
5.18	Instalación del paquete "chirpstack-network-server".	48
5.19	Inicio del servicio de chirpstack-network-server.	49
5.20	Habilitación del servicio de chirpstack-network-server.	49
5.21	Integración con la base de datos creada.	50
5.22	Nombre de la banda del servidor de red.	50
5.23	Instalación del paquete chirpstack-application-server.	51
5.24	Integración con la base de datos "chirpstack_as".	51
5.25	Generación aleatoria del token.	52
5.26	Ingreso del token generado aleatoriamente.	52
5.27	Inicio del servicio de chirpstack-application-server.	52
5.28	Habilitación del servicio de chirpstack-application-server.	53
5.29	Visualización de logs del servidor chirpstack.	53
5.30	Autenticación en la página de ChirpStack.	54
5.31	Ingreso al panel principal de ChirpStack.	54
5.32	Creación de rol 'chirpstack_as_events'.	55
5.33	Creación de la base de datos 'chirpstack_as_events'.	55
5.34	Conexión a la nueva base de datos 'chirpstack_as_events'.	55
5.35	Creación de la extensión 'hstore'.	55
5.36	Verificación de conexión a la base de datos 'chirpstack_as_events'.	56
5.37	Configuración del archivo 'chirpstack-application-server.toml'.	56

5.38	Estado del servicio chirpstack-application-server	57
5.39	Inicio de sesión Gateway MikroTik.....	58
5.40	Contraseña del Wi-Fi del Gateway MikroTik.....	58
5.41	Apartado de LoRa.	59
5.42	Agregar el servidor de ChirpStack al Gateway MikroTik.	59
5.43	Datos del Servidor de ChirpStack.	60
5.44	Apagar el servidor	60
5.45	Agregar el Servidor de red.....	61
5.46	Plataforma de ChirpStack.....	62
5.47	Pestaña de integraciones de ChirpStack.	62
5.48	Registro en Amazon Web Services (AWS).	64
5.49	Verificación de correo electrónico.	64
5.50	Código de verificación de correo electrónico.	65
5.51	Creación de la Contraseña para la cuenta de "Amazon Web Services (AWS)".....	65
5.52	Confirmación de la contraseña para el registro en AWS.	66
5.53	Información de contacto para el registro en AWS.....	66
5.54	Información de facturación en AWS.	67
5.55	Confirmación de identidad para el registro en AWS.	67
5.56	Ingreso de código enviado al celular ingresado.....	68
5.57	Selección del plan de soporte.	68
5.58	Visualización del registro completado exitosamente.	69
5.59	Inicio de sesión en "Amazon Web Services (AWS)".	69
5.60	Comprobación de seguridad.	70
5.61	Verificación de seguridad.	70
5.62	Buscar el servicio Simple Notification Service (SNS).	71
5.63	Crear un tema en Simple Notification Service (SNS).....	71
5.64	Tipos de temas de Amazon SNS.	72

5.65	Confirmación del tema creado.....	73
5.66	Buscar el servicio Simple Queue Service (SQS).....	73
5.67	Creación de una nueva cola SQS.....	74
5.68	Configuración de la cola SQS.....	74
5.69	Selección de la configuración de la cola SQS.....	75
5.70	Políticas de acceso de la cola SQS.	75
5.71	Cola SQS creada correctamente.	76
5.72	Ingreso a la cola SQS creada.	76
5.73	Suscripción al tema de Amazon SNS.	77
5.74	Selección del tema de Amazon SNS disponible para la cola.....	77
5.75	Relación entre el ARN de suscripción y el tema de Amazon SNS.	78
5.76	Panel de Amazon SNS.....	78
5.77	Visualización de temas creados en Amazon SNS.....	79
5.78	ID de suscripción en Amazon SNS.	79
5.79	Detalles de la suscripción de Amazon SNS.....	80
5.80	Habilitación de la entrega de mensajes sin procesar.....	80
5.81	Cambios guardados correctamente de la suscripción de Amazon SNS.....	81
5.82	Logo de "Amazon QuickSight".....	81
5.83	Menú de ChirpStack.....	82
5.84	Aplicaciones creadas en la plataforma ChirpStack.....	83
5.85	Lista de integraciones de ChirpStack.	83
5.86	Integración con "Amazon Web Services SNS".....	84
5.87	Colas Amazon SQS.....	84
5.88	Visualización del ARN creado.	85
5.89	Enviar y Recibir mensajes de la cola SQS.....	85
5.90	Sondeo de Mensajes en Amazon SQS.....	86
5.91	Cuerpo del Mensaje Sondeado en Amazon SQS.....	86

5.92	Ruta de la carpeta de postgresql en la consola del servidor	87
5.93	Configuración del archivo 'postgresql.conf'	88
5.94	Configuración del archivo 'pg_hba.conf'	88
5.95	Estado del servicio PostgreSQL.	89
5.96	Configuración de la Conexión Remota en pgAdmin 4.	89
5.97	Visualización de la conexión remota en pgAdmin 4.	90
5.98	Instalación del paquete pgAgent en la consola del servidor	91
5.99	Creación de la extensión pgAgent.	91
5.100	Creación del lenguaje de procedimiento pl/pgsql.	92
5.101	Ejecución del pgAgent.	92
5.102	Visualización del pgAgent instalado en pgAdmin 4.	92
5.103	Lista de Jobs creados en pgAgent.	93
5.104	Configuración de la primera pestaña 'General'.	93
5.105	Configuración de la segunda pestaña 'Steps'.	94
5.106	Digitación del código a ejecutar	94
5.107	Configuración de la tercera pestaña 'Schedules'.	95
5.108	Configuración del horario en el que se repetirá el trabajo programado.	96
5.109	Visualización de las propiedades de 'ActualizarConsumo'.	96
5.110	Configuración de la primera pestaña 'General'.	97
5.111	Configuración de la segunda pestaña 'Steps'.	97
5.112	Digitación del código a ejecutar	98
5.113	Configuración de la tercera pestaña 'Schedules'.	99
5.114	Configuración del horario en el que se repetirá el trabajo programado.	99
5.115	Visualización de las propiedades de 'ActualizarSeparado'	100
5.116	Configuración de la primera pestaña 'General'.	100
5.117	Configuración de la segunda pestaña 'Steps'.	101
5.118	Digitación del código a ejecutar	101

5.119 Configuración de la tercera pestaña 'Schedules'.	102
5.120 Configuración del horario en el que se repetirá el trabajo programado.	103
5.121 Visualización de las propiedades de 'ActualizarNumerico'.	103
5.122 Configuración de la primera pestaña 'General'.	104
5.123 Configuración de la segunda pestaña 'Steps'.	104
5.124 Digitación del código a ejecutar	105
5.125 Configuración de la tercera pestaña 'Schedules'.	106
5.126 Configuración del horario en el que se repetirá el trabajo programado.	106
5.127 Visualización de las propiedades de 'ActualizarPorDia'.	107
5.128 Configuración de la primera pestaña 'General'.	107
5.129 Configuración de la segunda pestaña 'Steps'.	108
5.130 Digitación del código a ejecutar	108
5.131 Configuración de la tercera pestaña 'Schedules'.	109
5.132 Configuración del horario en el que se repetirá el trabajo programado.	110
5.133 Visualización de las propiedades de 'ActualizarResta'.	110
5.134 Configuración de la primera pestaña 'General'.	111
5.135 Configuración de la segunda pestaña 'Steps'.	111
5.136 Digitación del código a ejecutar	112
5.137 Configuración de la tercera pestaña 'Schedules'.	113
5.138 Configuración del horario en el que se repetirá el trabajo programado.	113
5.139 Visualización de las propiedades de 'ActualizarResultado'.	114
5.140 Configuración de la primera pestaña 'General'.	114
5.141 Configuración de la segunda pestaña 'Steps'.	115
5.142 Digitación del código a ejecutar	115
5.143 Configuración de la tercera pestaña 'Schedules'.	116
5.144 Configuración del horario en el que se repetirá el trabajo programado.	117
5.145 Visualización de las propiedades de 'ActualizarConsumoPorDia'.	117

5.146	Configuración de la primera pestaña 'General'.	118
5.147	Configuración de la segunda pestaña 'Steps'.	118
5.148	Digitación del código a ejecutar	119
5.149	Configuración de la tercera pestaña 'Schedules'.	120
5.150	Configuración del horario en el que se repetirá el trabajo programado.	120
5.151	Visualización de las propiedades de 'ActualizarConsumoPorMes'.	121
5.152	Menú de Amazon Web Services.	122
5.153	Menú de QuickSight.	122
5.154	Registro de correo electrónico.	123
5.155	Registro de contraseña para acceder a AWS.	123
5.156	Inscripción exitosa en Amazon QuickSight.	124
5.157	Ventana principal de Amazon QuickSight.	124
5.158	Selección de conjunto de datos.	125
5.159	Selección de PostgreSQL como nuevo origen de datos.	125
5.160	Configuración de PostgreSQL como nuevo origen de datos.	126
5.161	Creación exitosa del origen de datos.	127
5.162	Selección de una consulta SQL personalizada.	127
5.163	Escritura de la consulta SQL personalizada.	128
5.164	Consulta SQL Personalizada.	128
5.165	Resultado de la consulta SQL Personalizada.	129
5.166	Lista de campos a utilizar para elaborar el Dashboard.	129
5.167	Elección del tipo Gráfico circular para visualizar el número de registros por 'device_name'.	130
5.168	Elección del tipo tabla dinámica para visualizar la cantidad de consumo al día.	130
5.169	Elección del tipo Gráfico de líneas para visualizar la cantidad de consumo en metros cúbicos en base a la fecha de consumo.	131
5.170	Elección del tipo tabla dinámica para visualizar la cantidad de consumo por mes.	131

5.171 Elección del tipo Gráfico de líneas para visualizar la cantidad de consumo en metros cúbicos en base al mes de consumo.....	132
5.172 Elección del tipo Gráfico circular para visualizar el número de registros por 'id'.	132
5.173 Elección del tipo Histograma para visualizar la distribución de frecuencia por 'frequency'.....	133
5.174 Publicación del panel.....	133
5.175 Visualización del panel publicado.	134
6.1 Elaboración de la parte frontal del prototipo	136
6.2 Funcionamiento del Medidor Inteligente.....	137
6.3 Instalación de la tubería para el flujo de agua.....	137
6.4 Colocación de la bomba de agua	138
6.5 Elaboración de la parte posterior del prototipo.....	138
6.6 Implementación de un Gateway LoRa.....	139
6.7 Visualización del estado de la conexión de los medidores inteligentes en la plataforma ChirpStack	140
6.8 Visualización del estado de la última conexión del medidor inteligente.	140
6.9 Elaboración de la parte posterior del prototipo.....	141
6.10 Tabla de consumo diario de agua durante la monitorización	141
6.11 Ilustración del consumo diario de agua durante la monitorización.....	142
6.12 Tabla de consumo de agua por mes durante la monitorización.....	142
6.13 Ilustración del consumo de agua por mes durante la monitorización	143
6.14 Ilustración de la frecuencia utilizada durante la monitorización	143

Capítulo I

INTRODUCCIÓN

En Muñoz (2019); Sacoto Cabrera et al. (2020); Vimos et al. (2016), los autores plantean que el Internet de las Cosas (IoT) es una red de objetos físicos conectados a través de internet, los cuales logran interactuar vía sistemas embebidos, redes de comunicación, y aplicaciones conectadas a la nube.

Como menciona el autor en Muñoz (2019), el Internet de las Cosas se está aplicando dentro del ámbito del medio ambiente, algunos ejemplos de esto tenemos la contaminación del aire, la contaminación del agua ya sea ríos, lagos, mares y también en sensores que sirven para detectar incendios en los bosques.

Teniendo en cuenta a Sacoto Cabrera et al. (2020) menciona que otro sector en el cual es aplicado el IoT, son las ciudades, en los cuales se mejora los servicios disponibles. A las ciudades que utilizan el IoT, en su día a día se las denomina ciudades inteligentes “Smart Cities”, algunos ejemplos son sistemas de detección de parqueo libre, sensores para el control de edificios, puentes, construcciones patrimoniales.

Capítulo II

PROBLEMA

2.1 Antecedentes

Hoy en día ETAPA-EP, la empresa Municipal de la ciudad de Cuenca se encarga del mantenimiento, monitoreo y la operación de la red de agua potable, la misma que busca constantemente optimizar y brindar mayor calidad en el servicio.

ETAPA-EP tiene previsto garantizar un suministro constante de agua potable a sus usuarios, ya que cuenta con 33 centros de almacenamiento ubicados en diferentes zonas de la ciudad, con una capacidad total de almacenamiento de ciento veinte mil metros cuadrados para la ciudad. En las zonas rurales, cada Fábrica cuenta con las reservas necesarias y suficientes para atender a la población. La red de distribución de agua limpia alcanza el 96 por ciento en las zonas urbanas y el 88 por ciento en las zonas rurales.

La empresa ETAPA-EP, con el fin de verificar el proceso de lectura de consumo, ha dividido la cobertura de los sistemas que administra, en sectores, que a su vez se dividen en líneas o rutas de lectura, las cuales tienen una sección de expansión que comprende un promedio de 300 conexiones donde se esperan turnos para una determinada fecha de ejecución.

La empresa posee una relación directa y de largo plazo con los bienes de los usuarios, ya que existen medidores de flujo, los mismos que serán leídos, es decir saber la cuantía exacta de metros cúbicos de agua consumida durante un establecido ciclo de tiempo, para esto es necesario consultar y anotar en informes o en dispositivos móviles la cantidad que aparece en las lecturas del contador. Para la comprobar el consumo cada lector es guiado a través de una ruta anticipadamente establecida, trazada y adecuada por el subsistema catastral, cuyo informe contendrá los aforos a leer, así como datos de ubicación del inmueble, nombre del propietario, número de cuenta y otros datos relacionados con el medidor. El consumo de los usuarios se establece por la diferencia entre la estimación actual registrada por el medidor y el obtenido en el período anterior, o por una estimación del consumo.

Hoy en día la tecnología de IoT está cada vez más avanzada es por ello que se puede realizar diversas funcionalidades, como el uso de sensores para la medición del agua ya que esto nos servirá para la obtención del consumo de agua que se tiene en la ciudad, en base a esto se implementará el servidor LoRa, el cual nos recolecta toda la información obtenida mediante los sensores del consumo de agua sin embargo debemos realizar una integración más, ya que el servidor LoRa recolecta los datos pero estos no se pueden observar una manera clara, es por ello que se va a implementar la integración con Amazon Web Service (AWS).

Facilitar a los usuarios de las redes LoRaWAN a través de la implementación de un servidor LoRa y su conectividad con Amazon Web Service para la adquisición de información de los diferentes tipos de sensores.

2.2 Importancia

El siguiente proyecto pretende brindar una solución para un mejor manejo de los datos obtenidos por los sensores que se encuentra en el servidor LORA integrando la plataforma de ChirpStack con la nube de Amazon Web Service, de tal manera que al momento de consultar la información podamos verla de una manera más ordenada y simplificada ya que en ChirpStack solamente tenemos una data en donde no se puede tener la información clara de cada sensor, a diferencia de AWS que es en donde se puede realizar una integración de toda la data obtenida de ChirpStack para así poder tener un mejor manejo de la data, es decir que se pueda obtener una información más clara y detallada de cada sensor que se tenga en la plataforma de ChirpStack.

El alcance de este proyecto es poder brindar una ayuda en la ciudad de Cuenca para que se puedan tener datos del consumo diario del agua y así poder evitar el mal uso de la misma.

2.3 Alcance

Lo que se busca es tener una mejor visualización de los datos que se registran en el servidor ChirpStack mediante la integración con Amazon Web Service.

Los beneficiarios del proyecto será ETAPA ya que ellos podrán tener una mejor información acerca del consumo diario del agua, así también como poder actuar ante un consumo excesivo o fugas de agua dentro de la ciudad de Cuenca.

Capítulo III

OBJETIVOS

3.1 General

Implementación de un servidor de administración y gestión de una red LoRaWAN e integración con Amazon Web Service.

3.2 Específicos

- OE1. Investigar los fundamentos de LoRaWAN orientado a IoT. Fundamentos de servidores de gestión y administración de la red que comunica los módulos y dispositivos IoT.
- OE2. Implementar un servidor LoRaWAN.
- OE3. Integrar la data de ChirpStack a Amazon Web Service para una óptima visualización de los datos registrados en el Servidor.
- OE4. Analizar las herramientas de Amazon Web Service para almacenamiento, análisis y visualización de datos.
- OE5. Diseñar e implementar un plan de prueba, para posteriormente poder verificar el acoplamiento del servidor con el sistema y su correcto funcionamiento.

Capítulo IV

REVISIÓN DE LA LITERATURA O FUNDAMENTOS TEÓRICOS

4.1 IoT

Según Parra-Valencia et al. (2017); Vimos and Cabrera (2018) IoT se refiere a la conexión de objetos tecnológicos o que sean electrónicos a Internet, este concepto se deriva del avance de la tecnología y a la necesidad de compartir y controlar las cosas que nos rodean. Algunas de las áreas en las que IoT tiene una fuerte presencia son: adopción generalizada de redes basadas en Protocolo de Internet (IP), ahorros en potencia informática, miniaturización, avances en análisis de datos y aparición de computación en la nube, atención médica, herramientas de aprendizaje, seguridad, optimización de procesos, agricultura entre otros.

Como afirman los autores en Arasteh et al. (2016); Sacoto-Cabrera et al. (2018), en ciertos lugares se requieren de muchos más servicios e infraestructuras para poder seguir desarrollándonos como ciudad, debido a esto dentro de la ciudad existe un gran aumento en el uso de sensores, teléfonos inteligentes y actuadores, los cuales están teniendo una gran acogida para IoT debido a la facilidad que estos tienen para poder conectarse entre sí a través de internet. IoT

se puede conectar entre sí a través de una red global, estos se consideran como objetos reales que poseen una baja capacidad de procesamiento y almacenamiento, ayudando de esta manera a tener un mejor rendimiento, confiabilidad y seguridad tanto para la infraestructura como para la ciudad en donde se la aplique.

4.2 Redes LPWAN

Las redes de baja potencia y área ampliada (LPWAN), según Carrasco Galdame (2020); Aranda et al. (2021) estas son soluciones tecnológicas para la conectividad de dispositivos dirigidos al internet de las cosas (IoT).

Las redes LPWAN pueden alcanzar hasta más de 10 kilómetros, dependiendo de la zona en donde se implemente, estas distancias se logran gracias a las modulaciones para transmitir información y las frecuencias de radio enlace que utilizan.

Actualmente existen tres grandes tecnologías LPWAN las cuales son:

4.2.1 SigFoX

Como menciona Carrasco Galdame (2020) este tipo de tecnología busca crear una red de telecomunicaciones para el IoT, de tal forma que cualquier dispositivo pueda estar conectado a través de Internet mediante su red.

Teniendo en cuenta a Pardal Garcés (2017) menciona que SigFoX es eficaz para las comunicaciones desde los puntos finales a las estaciones base, con una velocidad de 100 bps, a diferencia de que no es eficaz en transmisiones desde la base hacia los puntos finales, la descarga de información es más lenta.

Su alcance varía según el entorno en donde se implemente por ejemplo si se implementa en un entorno urbano su alcance puede llegar hasta los 3-10 km, para un entorno rural se puede llegar hasta 30-50 km.

4.2.2 Arquitectura SigFoX

De acuerdo con Martres (2022) SigfoX cuenta con su propia red, la cual tiene su propia estación base y antenas. La comunicación entre los objetos y la estación base se da de forma permanente ya que están a la escucha e interpretan las señales que reciben.

Para que un dispositivo funcione con esta red es necesario de un módulo de comunicación que sea compatible con SigFoX.

En la Fig.4.1, se observa cómo funciona la arquitectura de SigFoX para el envío de datos.

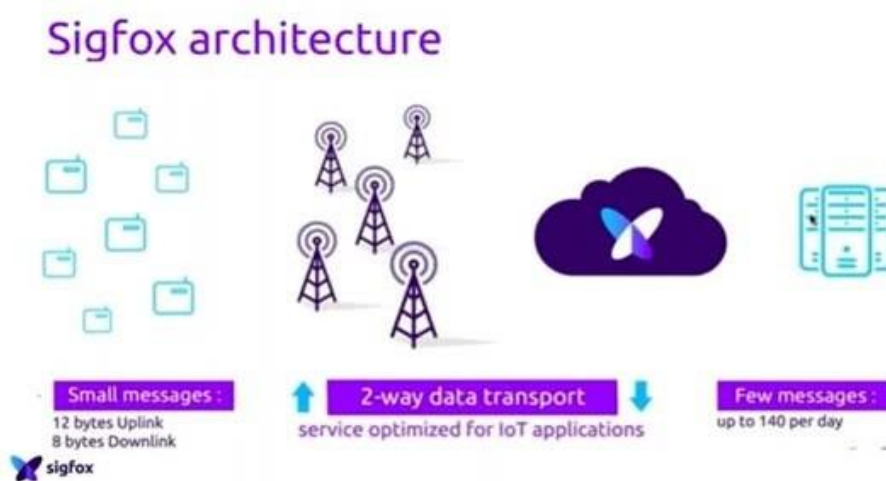


Figura 4.1: Arquitectura SigFoX Martres (2022)

4.2.3 NB-IoT

Según Sánchez Rosado (2019) Narrow-Band IoT, define como una tecnología móvil sobre espectro licenciado, se difunde en la banda ISM con frecuencias que están entre los 790Mhz y 862Mhz, este tipo de tecnología es muy similar a LTE, con diferencias en que NB-IoT se enfocó en mantener lo más simple posible sus comunicaciones, con el objetivo de reducir los costes de producción y consumo de batería.

De acuerdo con Machado González (2019) NB-IoT permite el acceso a servicios de red

utilizando una capa física que está optimizada para un bajo coste y consumo, en cuanto a su ancho de banda del canal completo es de 180 KHz, con una separación entre las sub-portadoras de 15kHz.

4.2.4 Arquitectura NB-IoT

En Saiz Miranda (2019), se indica que NB-IoT es una red similar a la LTE, con la diferencia de que está optimizada para una mayor cantidad de dispositivos, permitiendo transmisiones cortas de datos.

En la Fig.4.2, se observa como los dispositivos se enlazan con el eNodeB a través de la capa de acceso, luego este utiliza el protocolo Non-Access Stratum (NAS) para trasladarse al EPC para tener una movilidad con MME y gestionar las sesiones. Por último, para llegar a los servidores y aplicaciones se debe pasar a un plano de usuario a través de IP o Non-IP, esta última es más adecuada para reducir las cabeceras en el equipo de usuario para que la transmisión sea más segura.

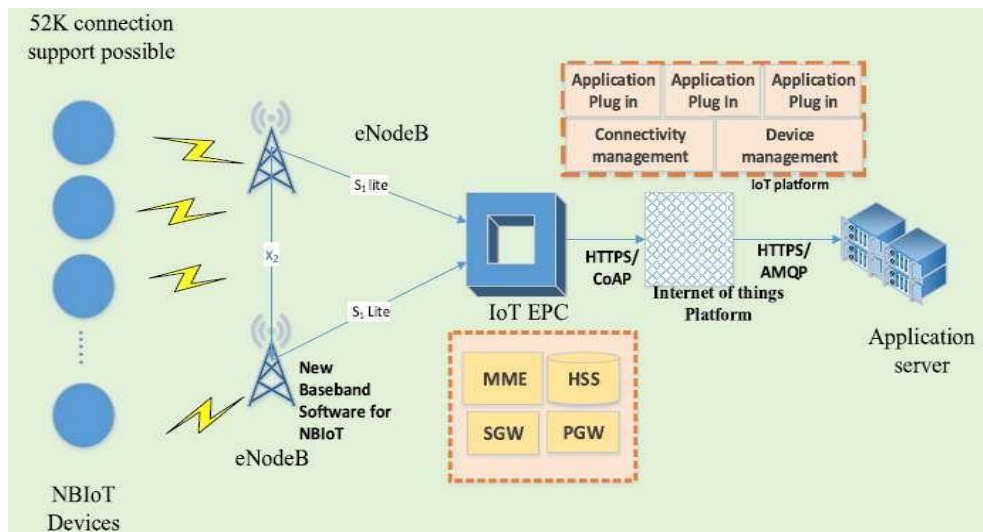


Figura 4.2: Arquitectura NB-IoT Saiz Miranda (2019)

4.3 LoraWAN

De acuerdo con Khanderao and Kemkar (2021), LoRa es una tecnología de modulación de espectro ensanchado desarrollada como soporte de conectividad para redes LPWAN. La arquitectura LoRa se divide en dos capas: capa física LoRa y la capa MAC LoRaWAN.

Como menciona Lluva Plaza et al. (2021), LoRa es una tecnología de capa física que se utiliza para poder establecer comunicaciones con los dispositivos IoT. LoRa utiliza los parámetros de factor de dispersión (SF) para poder ajustar el alcance de la transmisión y la tasa de datos. La tasa de transmisión de datos está comprendida entre 300 bps y 50 Kbps, en donde se pueden transmitir como máximo 243 bytes de carga útil.

Desde el punto de vista de Pérez García (2017), LoRa es una técnica de modulación basada en la técnica de espectro ensanchado y es una variante de Chirp Spread Spectrum (CSS), también menciona que LoRa mejora la sensibilidad del receptor y utiliza el ancho de banda completo del canal para transmitir la señal. LoRa puede demodular señales por debajo del nivel de ruido de 19,5 dB, mientras que otros sistemas de cambio de frecuencia (FSK) necesitan una intensidad de señal de 8 a 10 dB por encima del nivel de ruido para demodular correctamente.

De acuerdo con lo descrito en Liya and Aswathy (2020), LoRaWAN es un protocolo de comunicación, mientras que LoRa se define como la capa física que garantiza una comunicación de largo alcance.

De acuerdo con Lavric and Petrariu (2018), en la Fig.4.3 presenta la pila de protocolos de comunicación LoRaWAN. Se puede observar que las especificaciones LoRaWAN definidas por LoRa Alliance se definen sobre la modulación LoRa y la capa LoRa MAC (Medium Access Control).

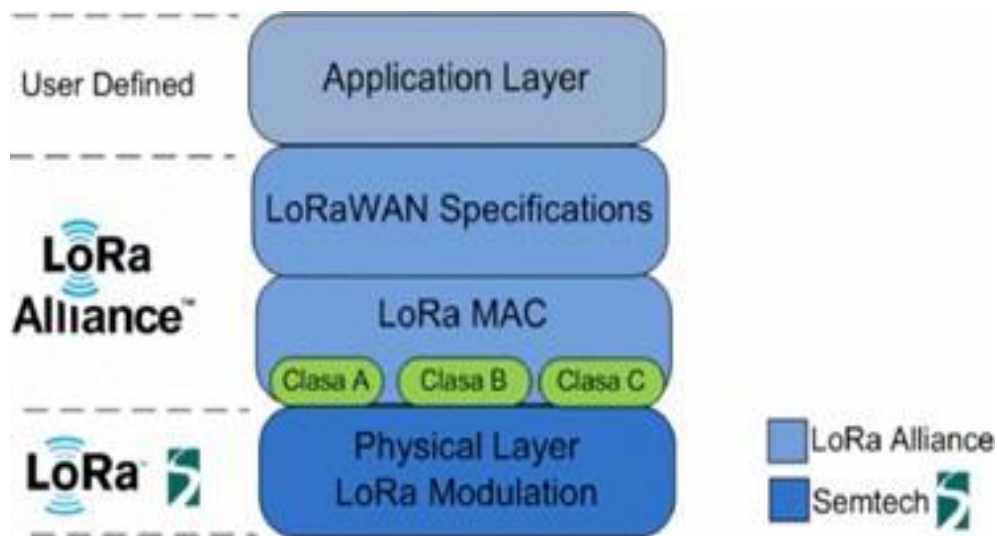


Figura 4.3: Protocolo de Comunicación LoRaWAN Lavric and Petrariu (2018)

Como principal ventaja de LoRaWAN es que usa la frecuencia ISM (Industrial, Scientific and Medical) sin licencia. La única restricción la impone el parámetro del ciclo de trabajo que se ajusta en función de las distintas zonas geográficas. Los dispositivos Lora funcionan con baterías.

LoRa Alliance determina tres clases de dispositivos: clase A, clase B y clase C. Para lograr la eficiencia energética, un nodo LoRa, además de la regulación del ciclo de trabajo, tiene la capacidad de recibir un paquete del módulo Gateway solo en un intervalo de tiempo restringido.

4.3.1 Tipos de Dispositivos LoRaWAN

Como se describe en Liya and Aswathy (2020), en la tabla 4.1 IEEE 802.15.4 es el protocolo de seguridad de LoRaWAN y cuenta con tres clases diferentes de dispositivos LoRaWAN los cuales son Clase A, Clase B, Clase C.

- Los dispositivos de clase A cuentan con mayor eficiencia energética, estos dispositivos necesitan mayor tiempo de latencia, por lo general es utilizado en varias redes de sensores de baja potencia.

- Los dispositivos de clase B se encuentran asociados con dispositivos de baja potencia ya que son aquellos que cuentan con un tiempo ajustado para percibir instrucciones e información desde la red. Estos dispositivos son perfectos tanto para sensores como actuadores.
- Los dispositivos de clase C cuenta con un tiempo de latencia bajo, pero tiene ranuras de recepción máximas, estos son bidireccionales, ya que necesitan una fuente de alimentación externa.

Tabla 4.1: Tipos de LoRa

Clase A	Clase B	Clase C
Alimentación por batería	Latencia Baja.	Latencia Baja.
Los dispositivos finales inician la comunicación.	Ranuras de ping.	El servidor puede iniciar la comunicación.
En una ventana de respuesta predeterminada, el servidor puede comunicarse con el dispositivo final.	En un intervalo de tiempo fijo, el servidor inicia la comunicación.	Los dispositivos finales reciben constantemente.
Bidireccional.	Bidireccional con ranuras de recepción.	Bidireccional.
Pequeña carga útil.	Pequeña carga útil.	Pequeña carga útil.
Unicast y Multicast	Unicast y Multicast	Unicast y Multicast

4.3.2 Parámetros de Comunicación

Como describe Guambuguete Gómez and Soledispa Reyes (2021), un transceptor LoRa cuenta con cinco parámetros para su configuración:

- **Potencia de transmisión (TP):** este parámetro se puede configurar entre -4dBm y +20 dBm.
- **Frecuencia portadora (CF):** esta frecuencia se puede sintonizar entre 137 MHz y 1020 MHz.
- **Factor de dispersión (SF):** este parámetro se suele establecer entre SF7 y SF12.

- **Ancho de banda (BW):** al contar con más ancho de banda puede causar más velocidad de datos, pero también causa más tiempo de aire y menos sensibilidad. LoRaWAN se puede utilizar entre uno de estos tres anchos de banda: 125 KHz, 250 KHz y 500 KHz.
- **Tasa de código (CR):** esta tasa es la misma que la tasa de la corrección de errores hacia adelante (FEC), que utiliza el módem LoRa y protege los datos contra el ruido y la interferencia. Este parámetro se puede establecer en uno de los siguientes: 4/5, 4/6, 4/7, 4/8.

4.3.3 Estructura del Paquete LoRa

Como se expresa en Guambugete Gómez and Soledispa Reyes (2021), en la Fig.4.4 muestra la estructura del paquete de datos LoRa. La capa física incluye lo que sería un preámbulo, un encabezado y un CRC de 16 bits para el encabezado, todos seguidos de una carga útil y un CRC de carga útil de 16 bits.

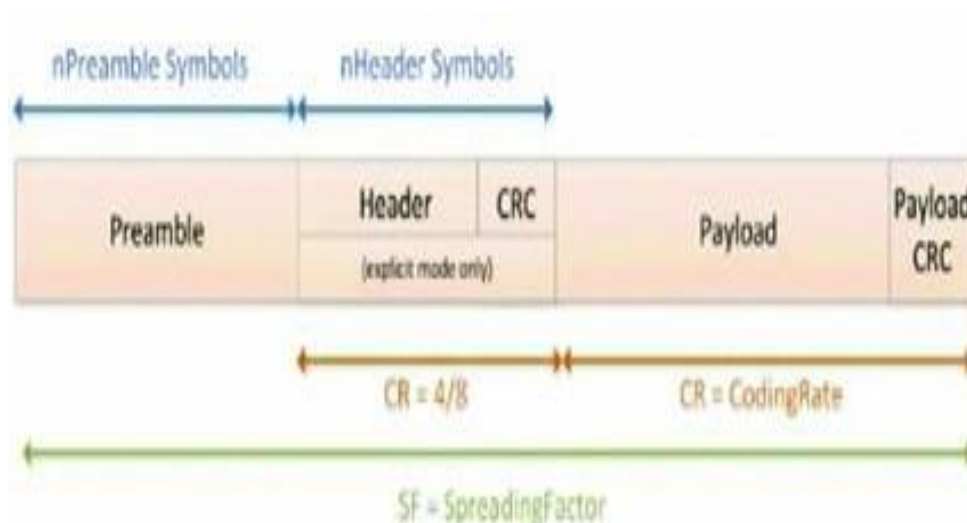


Figura 4.4: Estructura del Paquete LoRa Guambugete Gómez and Soledispa Reyes (2021)

4.3.4 Arquitectura de red LoRaWAN

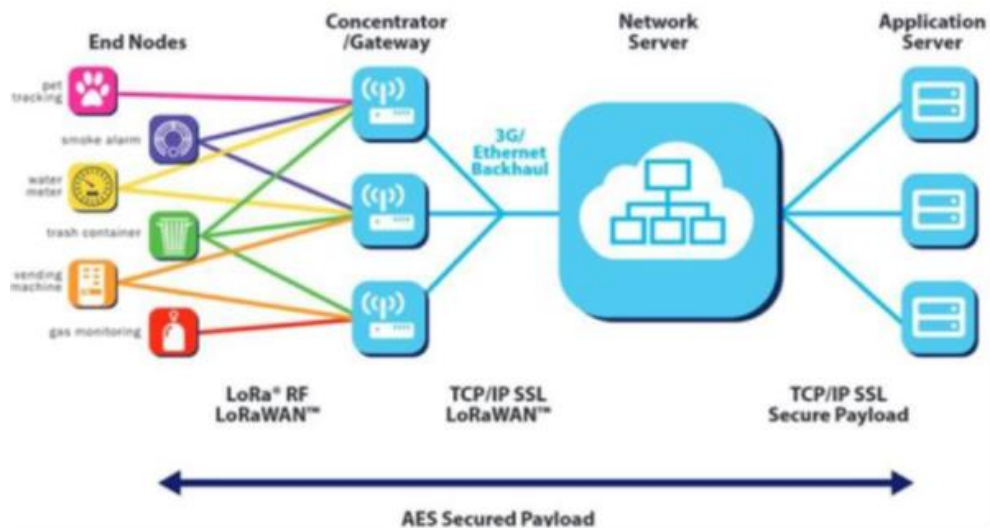


Figura 4.5: Arquitectura de red LoRaWAN Guambugete Gómez and Soledispa Reyes (2021)

Como menciona AlfaIOT (2022); Sacoto-Cabrera et al. (2017, 2022), en la Fig.4.5 una red LoRaWAN se puede tener diferentes elementos los cuales son: Nodos finales, Gateway, Servidores de Red y Servidores de Aplicación.

- **Nodos Finales:** Estos son mecanismos de hardware que tiene la capacidad de detección, los nodos remiten y admiten datos hacia o desde los Gateway LoRaWAN. Según Castan et al. (2021); Sanchis-Cano et al. (2017), Cuando un dispositivo final envía un mensaje al Gateway esto se lo denomina como Uplink, en cambio cuando recibe un mensaje del Gateway se lo denomina Downlink.
- **Gateway:** Es un dispositivo de hardware que recopila todos los mensajes LoRaWAN de Nodos terminales, estos mensajes se convierten en bits que se pueden emitir a través de los nodos red IP.
- **Servidores de Red:** Todos los mensajes del Gateway se envían al servidor de red, en este servidor se tiene los procesos para el tratamiento de los datos.

El servidor de Red es el responsable de:

- Enrutamiento y reenvío de mensajes a la aplicación adecuada.
 - Busca y selecciona la mejor puerta de enlace para el mensaje descendente.
 - Eliminación de mensajes duplicados en el caso de que se reciban por múltiples Gateway.
 - Descifrar y encriptar los mensajes que se envían y reciben.
- Servidor de Aplicaciones: Según Farinango Tuquerres (2018), en los servidores de aplicación se tienen procesos como la recolección de datos de los dispositivos finales de la red, estos servidores por lo general se ejecutan en la nube ya sea de forma pública o privada. El servidor de aplicación es el responsable de gestionar, manipular e interpretar los datos de las aplicaciones de los dispositivos finales.

4.4 Servidor de Aplicaciones

4.4.1 ChirpStack

ChirpStack es un servidor LoRaWAN, como se indica en Chirpstack.io (2022a) hace posible la conexión y envío de datos desde los dispositivos LoRa al Orion Context Broker (OCB) del sistema. La arquitectura de los servidores de ChirpStack se conectan con diferentes componentes del servidor red.

En la Fig.4.6, se observa la arquitectura de los servicios que posee ChirpStack.

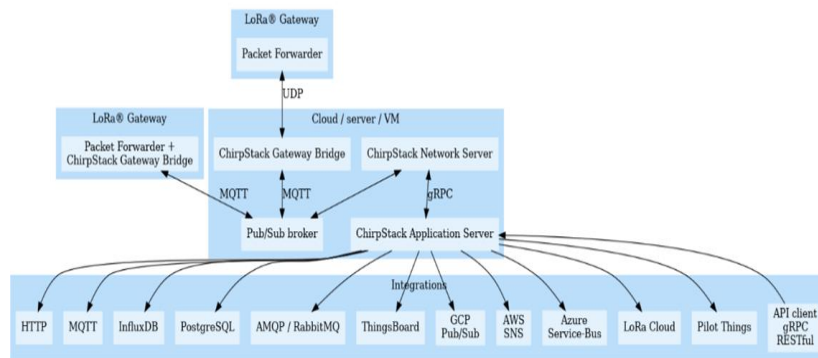


Figura 4.6: Arquitectura de Servicios de ChirpStack Chirpstack.io (2022a)

4.4.2 ChirpStack Gateway Bridge, Network Server y Application Server

Como describe en Chirpstack.io (2022a), en la tabla 4.2 ChirpStack cuenta con tres servicios para su aplicación y uso, cada uno tiene una tarea y protocolo determinado, a continuación, se muestran breves descripciones de cada uno de estos servicios.

Tabla 4.2: Servicios de ChirpStack

Servicio	Descripción
ChirpStack Gateway Bridge	Este servicio se encuentra entre el reenvío de paquetes y el agente MQTT Chirpstack.io (2022b). Aquí es donde se lleva a cabo la conversión de formato por lotes. El formato de datos utilizado por el componente Forwarder.
ChirpStack Network Server	Se encarga de gestionar el estado de la red, y también ha activado el conocimiento Dispositivos en la red Chirpstack.io (2022c). Cuando el servidor de aplicaciones necesita enviar datos en el dispositivo, el servidor web los mantiene en espera hasta que pueda enviárselos a uno de ellos.
ChirpStack Application Server	Este es un servidor de aplicaciones LoRaWAN que proporciona interfaces web y API para administrar usuarios, organizaciones, aplicaciones, puertas de enlace y dispositivos Lahtela y Kaplan (1966). La aplicación final recibe datos del dispositivo mediante una de las integraciones configuradas, cuyo objetivo principal puede ser el análisis, alertas, visualización de datos, etc.

4.5 Amazon Web Service (AWS)

Como se afirma en Devalal and Karthikeyan (2018); Rodriguez Alvarado and Sacoto-Cabrera (2021), AWS es una plataforma en la nube que cuenta con muchos servicios, estos ofrecen tecnologías de infraestructura como almacenamiento, bases de datos y cómputo, incluyendo inteligencia artificial, lago de datos y análisis e internet de las cosas. El diseño de AWS es de los más seguros y flexibles que están disponibles en la actualidad.

4.5.1 Servicios AWT IoT

Amazon Web Service ofrece diversos tipos de servicios como:

- **FreeRTOS:** implementa un sistema operativo para microcontroladores que favorece la administración de los dispositivos de extremos pequeños y limitada potencia.
- **AWS IoT Greengrass:** cuenta con un servicio de nube y tiempo de ejecución de código abierto lo cual permite crear, implementar y administrar aplicaciones.
- **AWS IoT ExpressLink:** módulos que permiten transformar cualquier dispositivo integrado en un dispositivo conectado a IoT.

4.5.2 Servicios de Análisis

- **AWS IoT SiteWise:** es un servicio gestionado que reduce la recopilación, la organización y el análisis de datos de equipos industriales.
- **AWS IoT Analytics:** es un servicio en donde se puede implementar herramientas para el análisis sofisticado de cantidades masivas de datos de IoT.

- **AWS IoT Events:** es un servicio que facilita detectar y responder a eventos de sensores y aplicaciones compatible con IoT.

4.6 Medidores Inteligentes

El crecimiento en la población y la reducción de recursos naturales han generado la necesidad de tener un mejor control sobre el consumo de agua potable es por ello por lo que se han logrado desarrollar dispositivos y sistemas los cuales permiten contabilizar y gestionar el uso de este recurso.

Como señala Hernández et al. (2021), los medidores de agua son instrumentos precisos, registra el caudal volumétrico de agua (m³/s) que fluye desde la red de distribución hasta el sistema de abastecimiento de agua dentro de la casa. Las herramientas que se utilizan actualmente son mecánicas. Genera mensajes estáticos que impiden el control remoto del dispositivo para obtener la lectura de datos es por ello por lo que con IoT se busca implementar un sensor de agua el cual facilite la lectura de datos.

Los medidores inteligentes del agua ofrecen una mejora para una óptima administración del agua, la evaluación inteligente permite captar en tiempo real el consumo de agua además de que permite leer esta información ya sea de forma remota o local. Para la realización de este proyecto se tiene como medidor inteligente un sensor de agua LoRaWAN de clase A.

Como menciona el autor en Álvarez et al. (2021), un dispositivo LoRaWAN clase A aumenta la eficacia en cuanto al uso de energía, para la transmisión, cada nodo decide cuándo transmitir, dichos nodos intentan transmitir en cada ventana y únicamente un nodo tendrá éxito, cuando se haya transmitido se obtendrán dos ventanas de recepción en donde las cuales se podrán escuchar por mensajes de bajada desde algún Gateway.

En cuanto a los Gateways el autor menciona en Lavric and Petrariu (2018) que estos no envían una repetición de mensaje hasta que el nodo la requiera, haya realizado una nueva transmisión y tenga ventanas de recepción.

4.7 Comunicación entre cliente y servidor

Como menciona el autor Rodríguez Moreno et al. (2017); Sacoto-Cabrera et al. (2018), los gateways se conectan mediante direcciones IP al servidor de red, a diferencia de los dispositivos que se conectan de manera inalámbrica mediante saltos a través de uno o varios gateways.

Como se indica en Maurin et al. (2018), los clientes son aquellos dispositivos que se utilizan para la conexión de objetos como pueden ser sensores o medidores entre otros, estos elementos son los encargados de transmitir la información recolectada a los Gateways.

También afirma que los servidores son equipos que pueden estar tanto físicamente como en la nube, estos se encargan de la recepción y procesamiento de la información que proviene de los dispositivos finales, además de que el servidor es el encargado de la configuración de red y también de la gestión de los dispositivos que estén conectados.

4.8 Herramientas que posee Amazon Web Service.

Como se puede observar en la Fig.4.7 visualizamos las diferentes herramientas que posee Amazon Web Services (AWS).

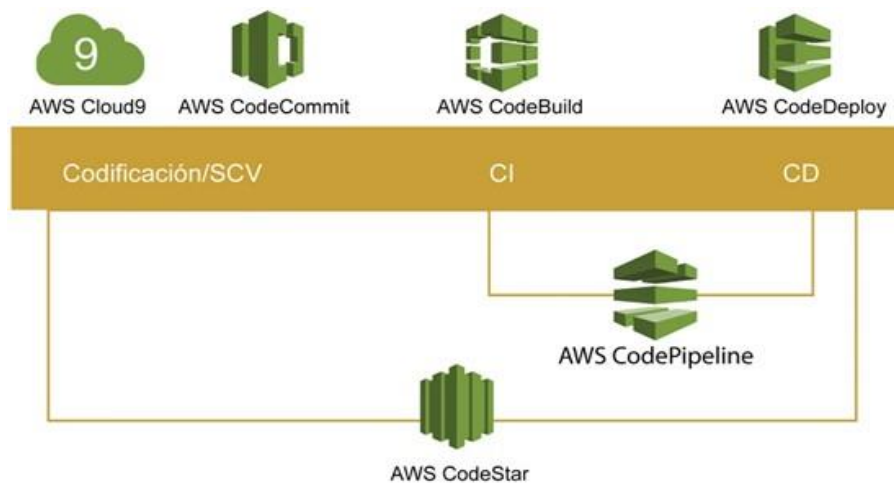


Figura 4.7: Herramientas de AWS. Devalal and Karthikeyan (2018)

4.8.1 AWS Cloud9

Cloud9 se desarrolló en 2010 y fue adquirido por AWS en 2016. Esta es la misma empresa que ofrece un IDE de desarrollo integrado que pueden utilizar varios desarrolladores al mismo tiempo para colaborar en el mismo proyecto.

Teniendo en cuenta al autor Potheary (2021), AWS Cloud9 proporciona un entorno completo desde su navegador. Esto ofrece muchos beneficios porque no tiene que instalar nada en su computadora local y los desarrolladores pueden leer y escribir código usando cualquier dispositivo.

De acuerdo con Inc. (2022d), AWS cuenta con varios beneficios los cuales son:

- Codificación de un solo navegador: Brinda flexibilidad de ejecución para su entorno de desarrollo en instancias de Amazon EC2, es decir que puede usar un solo navegador para compilar, ejecutar y depurar programas sin instalar un IDE nativo.
- Codificación en tiempo real: Facilita la modificación de código al permitir que los miembros de su equipo de desarrollo vean los cambios en tiempo real.

- Cree aplicaciones sin servidor de forma sencilla: Permite crear, ejecutar y depurar aplicaciones sin la necesidad de un servidor y rellenar previamente su entorno de desarrollo, contiene todos los SDK, bibliotecas y complementos necesarios para el desarrollo sin servidor.

4.8.2 AWS CodeCommit

Como señala Potheary (2021), como se puede observar en la Fig.4.8 AWS CodeCommit implementa Git de forma escalable, persistente y detallada para garantizar que su proyecto sea seguro y accesible. AWS CodeCommit utiliza un repositorio cifrado con control de acceso proporcionado por Amazon IAM o usuarios federados. AWS CodeCommit es un servicio completamente administrado. Por lo tanto, los recursos se escalan automáticamente para que pueda continuar con el desarrollo del código en lugar de controlar la capacidad de su infraestructura.

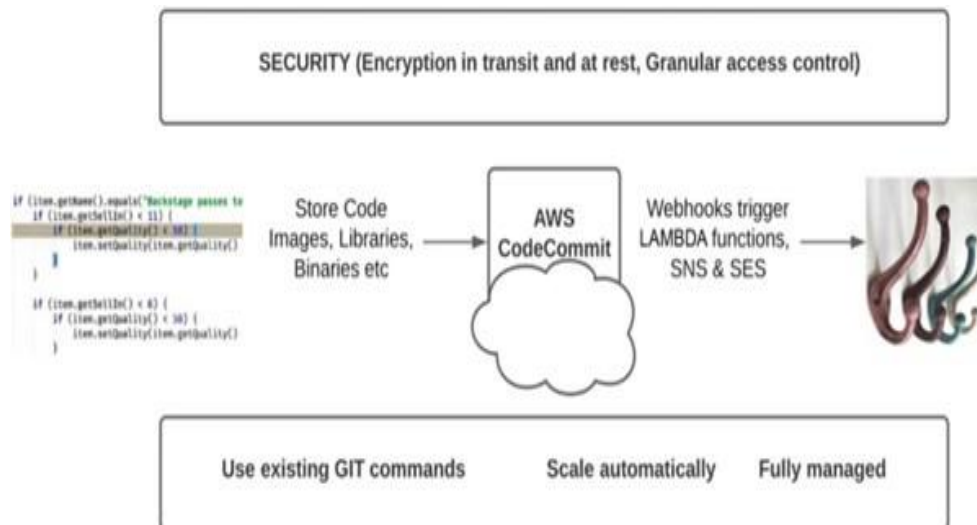


Figura 4.8: AWS CodeCommit. Potheary (2021)

Como señala Inc. (2022b), AWS CodeCommit nos brinda los siguientes beneficios:

- Administrador completo: Elimina la necesidad de alojar, mantener, proteger y ampliar su propio servidor de control de código fuente.
- Seguridad: cifra automáticamente los archivos en tránsito y en reposo.

- Alta disponibilidad: Posee una arquitectura altamente escalable, redundante y duradera.
- Colaboración en el código: Ayuda a los miembros del equipo a poder modificar el código.

4.8.3 AWS CodeBuild

Como señala Olaoye (2022), AWS CodeBuild es un servidor de compilación que se administra en la nube proporcionando un entorno informático para compilar código y ejecutar artefactos y pruebas de productos. CodeBuild obtiene el código fuente de su proveedor y realiza las acciones definidas en el archivo de especificación YAML, generalmente llamado Buildspec. Para los proveedores de fuentes, CodeBuild actualmente es compatible con Amazon S3, AWS CodeCommit, GitHub, Bitbucket y GitHub Enterprise.

Teniendo en cuenta a Inc. (2022a), AWS CodeBuild es un servicio de integración que compila, prueba y crea paquetes de software, con los siguientes beneficios:

- Servicio de compilación totalmente administrado: suprime la necesidad de configurar, parchear, actualizar y administrar tanto el software como el servidor.
- Con AWS CodeBuild, se le cobrará por el servicio según la cantidad de minutos que tarde en completar la compilación.
- Se puede agregar herramientas de compilación personalizados para que funciones con AWS CodeBuild.
- Seguridad: Las herramientas de compilación se cifran con una clave personalizada por AWS Key Management Service (KMS).

4.8.4 AWS CodeDeploy

Como el autor menciona en Dalbhanjan (2015), AWS CodeDeploy es un servicio que coordina la implementación de aplicaciones, esta herramienta se la recomienda cuando alguna organización o persona implementa el código en la infraestructura de administración.

Con AWS CodeDeploy, puede implementar su aplicación de la misma manera en diferentes entornos, incluidos los de prueba, producción y desarrollo.

Según Inc. (2022f), se pueden implementar varios contenidos de aplicaciones los cuales son:

- Código
- Funciones de AWS Lambda sin servidor
- Archivos web y de configuración
- Ejecutables
- Paquetes
- Guiones
- Archivos multimedia

4.8.5 AWS CodePipeline

Como menciona Inc. (2022c), CodePipeline es una utilidad de entrega constante que permite motorizar su proceso de dispersión para actualizaciones de aplicaciones e infraestructura rápidas y confiables.

Cada cambio de código automatiza las fases de implementación, prueba y construcción del proceso de lanzamiento.

Desde el punto de vista de Inc. (2022c), como se puede observar en la Fig.4.9 CodePipeline es un servicio que automatiza el lanzamiento y la entrega continua para lanzar al instante nuevas funciones a los usuarios. Se puede ampliar o integrar con herramientas de terceros.

Funcionamiento

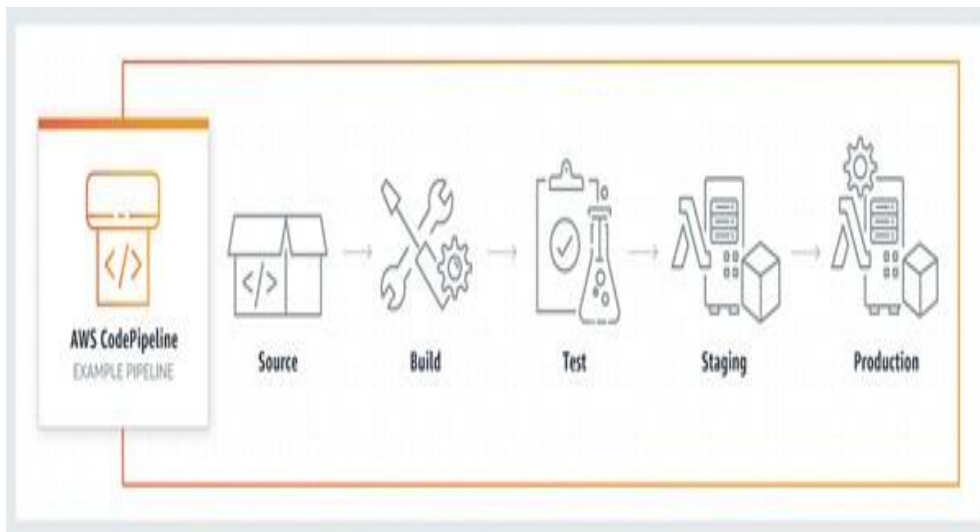


Figura 4.9: Funcionamiento de CodePipeline. Inc. (2022c)

Beneficios

AWS CodePipeline tiene varios beneficios los cuales son:

- Implementación rápida: automatice el proceso de lanzamiento de software para ofrecer rápidamente nuevas funciones a sus usuarios.
- Inicio ágil: no se requiere configuración del servidor. El proceso de lanzamiento de software se modela instantáneamente.
- Flujo de trabajo configurable: la interfaz de la consola le permite modelar diferentes fases del proceso de lanzamiento del software.
- Fácil integración: Ampliable a las necesidades de cada usuario.

4.8.6 AWS CodeStar

Como menciona el autor en Inc. (2022e), con AWS CodeStar, es muy fácil configurar todo el proceso de desarrollo y obtener resultados rápidos.

Como opina el autor en Inc. (2022e), CodeStar le permite implementar, crear y desarrollar rápidamente sus aplicaciones en AWS.

4.9 Seguridad y autenticación dentro del servidor LORA.

Como indica Natalia (2019), LoRaWAN emplea dos capas de seguridad: una para la capa de aplicación y otra para la capa de red. En la capa de red, se certifica la autenticidad de los nodos de la red, es decir los cambios a nivel de aplicación aseguran que los operadores de red no estén disponibles datos de la aplicación del usuario final.

Para estar dentro de una red LoRaWAN, los dispositivos finales deben estar personalizados y activados, sin tomar en cuenta el tipo de activación los dispositivos almacenan la siguiente información: "Application Session Key (AppSKey)", "Device Address (DevAddr)", "Network Session Key (NwkSKey)" y "Application Identifier (AppEUI)". Para la activación de los dispositivos en una red LoRaWAN se pueden activar de dos maneras diferentes:

4.9.1 Activation By Personalization (ABP)

Para este tipo de activación se debe personalizar los siguientes ítems:

- DevAddr: Dirección del dispositivo.
- AppEUI: Identificador de la Aplicación.
- NwkSKey: Clave de sesión de red.
- AppSKey: Clave de sesión de la aplicación.

4.9.2 Over-The-Air Activation (OTAA)

Para este tipo de activación el dispositivo final como describe en la Fig.4.10 genera una nueva sesión con el servidor de red, este genera un par de claves de sesión NwkSKey y AppSKey, el

dispositivo final encamina las claves de sesión durante el procedimiento de unión mediante las tramas Join-request y Join-accept usando la siguiente información:

- AppEUI: identificador de aplicación de destino.
- Device Identifier (DevEUI): identificador único de dispositivo.
- Application Key (AppKey): clave Advanced Encryption Standard (AES)-128 bits.

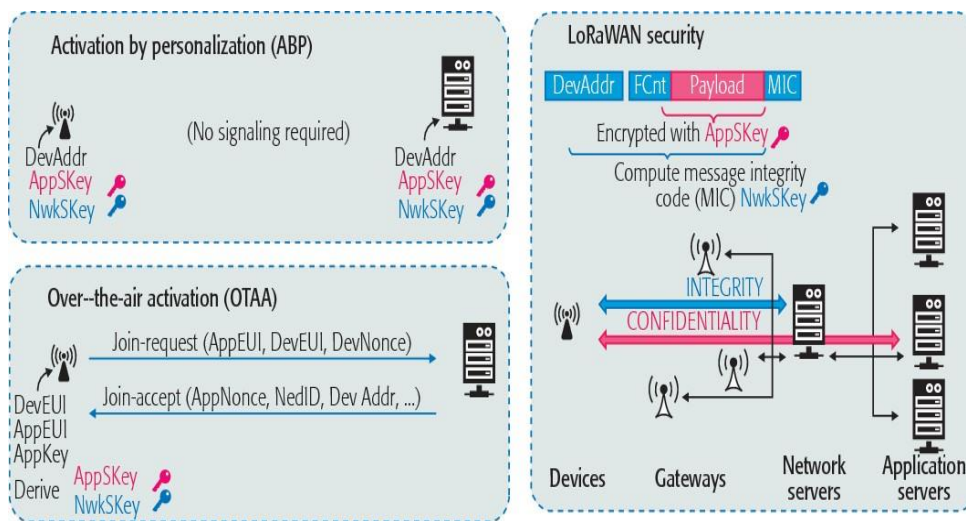


Figura 4.10: Activación ABP y OTAA Natalia (2019).

Capítulo V

MARCO METODOLÓGICO

5.1 Implementación de un servidor de ChirpStack para la gestión de los dispositivos IoT.

ChirpStack es un servidor LoRaWAN que permite realizar la conexión con dispositivos LoRa y conectar dispositivos IoT, el objetivo de esto es el de realizar la instalación para poder conectar los dispositivos y obtener los datos que estos nos envíen.

Para la instalación de este servidor se requiere por lo menos se cumpla con los requerimientos mínimos los cuáles son:

- Sistema Operativo: Ubuntu Server 18.04
- RAM: 2GB
- Disco Duro: 20GB

Para el proceso de instalación, es necesario instalar ciertas librerías y componentes iniciales para poder instalar satisfactoriamente el Chirpstack.

- MQTT broker: Un protocolo de publicación/suscripción que permite a los usuarios publicar información sobre temas a los que otros pueden suscribirse. Una implementación popular del protocolo MQTT es Mosquitto.
- Redis: Una base de datos en memoria utilizada para almacenar datos relativamente transitorios.
- PostgreSQL: La base de datos de almacenamiento a largo plazo utilizada por los paquetes de código abierto.

Para realizar la instalación de chirpstack se ejecuta la siguiente línea de código como se puede observar en la Fig.5.1.

```
root@ubuntu:/home/tesiscarrascodereg# sudo apt install mosquitto mosquitto-clients redis-server redis-tools postgresql
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
 fonts-liberation2 fonts-opensymbol gir1.2-gst-plugins-base-1.0
 gir1.2-gstreamer-1.0 gir1.2-gudev-1.0 gir1.2-udisks-2.0
 grilo-plugins-0.3-base gstreamer1.0-gtk3 libboost-date-time1.65.1
 libboost-filesystem1.65.1 libboost-iostreams1.65.1
 libboost-locale1.65.1 libcdr-0.1-1 libclucene-contribs1v5
 libclucene-core1v5 libcnis-0.5-5v5 libcoland2 libdazzle-1.0-0
 libdumbnet1 libe-book-0.1-1 libdataserverui-1.2-2 libeet0
 libepubgen-0.1-1 libetonyek-0.1-1 libevent-2.1-6 libext2-14
 libfreerdp-client2-2 libfreerdp2-2 libgpic2 libgee-0.8-2 libgextv2-2
 libgom-1.0-0 libgpgmepp0 libgpgod-common libgpod4 liblangtag-common
 liblangtag1 liblirc-client0 liblua5.3-0 libmedaart-2.0-0
 libmspub-0.1-1 libodfgen-0.1-1 libqwt6 libraw16 librevenge-0.0-0
 libsgutil2-2 libssh-4 libsuitesparseconfigs libvncclient1 libwinpr2-2
 libxapian3 libx11sec1-nss lp-solve media-player-info python3-nako
 python3-markupsafe syslinux syslinux-common syslinux-legacy
 usb-creator-common
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
 libc-ares2 libev4 libjemalloc1 libmosquitto1 libpq5 libuv1
 libwebsockets8 postgresql-10 postgresql-client-10
 postgresql-client-common postgresql-common sysstat
Suggested packages:
 postgresql-doc locales-all postgresql-doc-10 libjson-perl ruby-redis
 lisp
The following NEW packages will be installed:
 libc-ares2 libev4 libjemalloc1 libmosquitto1 libpq5 libuv1
 libwebsockets8 mosquitto mosquitto-clients postgresql postgresql-10
 postgresql-client-10 postgresql-client-common postgresql-common
 redis-server redis-tools sysstat
0 upgraded, 17 newly installed, 0 to remove and 3 not upgraded.
Need to get 6,324 kB of archives.
After this operation, 25.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libc-ares2 amd64 1.14.0-1ubuntu0.1 [37.5 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 libmosquitto1 amd64 1.4.15-2ubuntu0.18.04.3 [32.9 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libpq5 amd64 10.19-0ubuntu0.18.04.1 [108 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu bionic/main amd64 libuv1 amd64 1.18.0-3 [64.4 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu bionic/universe amd64 libev4 amd64 1:4.22-1 [26.3 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu bionic/universe amd64 libwebsockets8 amd64 2.0.3-3build1 [71.8 kB]
```

Figura 5.1: Instalación del protocolo MQTT y la base de datos en postgresql.

Para realizar la configuración de la base de datos se ingresa a postgresql como se puede observar en la Fig.5.2.

```
root@ubuntu:/home/tesiscarrascodereg# sudo -u postgres psql
psql (10.19 (Ubuntu 10.19-0ubuntu0.18.04.1))
Type "help" for help.

postgres=#
```

Figura 5.2: Configuración en la base de datos de PostgreSQL.

A continuación, se crean los roles con la contraseña requerida por el sistema, tal como se puede observar en la Fig.5.3.

```
postgres=# create role chirpstack_as with login password 'dbcuenca';  
CREATE ROLE  
postgres=#
```

Figura 5.3: Creación del rol "chirpstack_as".

De acuerdo con la Fig.5.4 se ejecuta la siguiente línea de código para crear un rol:

```
postgres=# create role chirpstack_ns with login password 'dbcuenca';  
CREATE ROLE  
postgres=#
```

Figura 5.4: Creación del rol "chirpstack_ns".

Como se puede observar en la Fig.5.5 para la creación de la base de datos en el servidor junto con los roles creados anteriormente se ejecuta la siguiente línea de código:

```
postgres=# create database chirpstack_as with owner chirpstack_as;  
CREATE DATABASE  
postgres=#
```

Figura 5.5: Creación de la base de datos "chirpstack_as".

Como se puede observar en la Fig.5.6 se ejecuta la siguiente línea de código para crear una base de datos:

```
postgres=# create database chirpstack_ns with owner chirpstack_ns;  
CREATE DATABASE  
postgres=#
```

Figura 5.6: Creación de la base de datos "chirpstack_ns".

De acuerdo con la Fig.5.7 se dirige a la base de datos chirpstack_as ejecutando la siguiente línea de código:

```
postgres=# \c chirpstack_as
You are now connected to database "chirpstack_as" as user "postgres".
chirpstack_as=#
```

Figura 5.7: Ingreso a la base de datos "chirpstack_as".

Dentro de la base de datos se crea una extensión la cual ayuda a facilitar la función de búsqueda.

Como se puede observar en la Fig.5.8 se crea la extensión 'pg_trgm' con la siguiente línea de código:

```
chirpstack_as=# create extension pg_trgm;
CREATE EXTENSION
chirpstack_as=#
```

Figura 5.8: Creación de la extensión "pg_trgm".

Como se puede observar en la Fig.5.9 se crea la extensión 'hstore' con la siguiente línea de código 'create extension hstore';

```
chirpstack_as=# create extension hstore;
CREATE EXTENSION
chirpstack_as=#
```

Figura 5.9: Creación de la extensión hstore.

Una vez terminada la configuración de la base de datos como se puede observar en la Fig.5.10 se debe salir de PostgreSQL ejecutando el siguiente comando:

```
chirpstack_as=# \q
root@ubuntu:/home/tesiscodeleg#
```

Figura 5.10: Finalización a la conexión de base de datos.

Como se puede observar en la Fig.5.11 se procede a configurar los repositorios de Chirp-Stack para obtener todos los componentes que se necesitan para realizar la instalación correctamente.

```
root@ubuntu:/home/tesiscarrascodelg# sudo apt install apt-transport-https dirnmgr
Reading package lists... Done
Building dependency tree
Reading state information... Done
dirnmgr is already the newest version (2.2.4-1ubuntu1.4).
dirnmgr set to manually installed.
The following packages were automatically installed and are no longer required:
 fonts-liberation2 fonts-opensymbol gir1.2-gst-plugins-base-1.0 gir1.2-gstreamer-1.0
 gir1.2-gudev-1.0 gir1.2-udisks-2.0 grilo-plugins-0.3-base gstreamer1.0-gtk3
 libboost-date-time1.65.1 libboost-filesystem1.65.1 libboost-iostreams1.65.1
 libboost-locale1.65.1 libbdr-0.1-1 libclucene-contribs1v5 libclucene-core1v5 libcnis-0.5-5v5
 libcolamd2 libdazze1.0-0 libdumbnet1 libe-book-0.1-1 libedataserverui-1.2-2 libeot0
 libepubgen-0.1-1 libetonyek-0.1-1 libevent-2.1-6 libexiv2-14 libfreerdp-client2-2
 libfreerdp2-2 libgic2 libgee-0.8-2 libgexiv2-2 libgom-1.0-0 libgpgmepp6 libgpod-common
 libgpod4 liblangtag-common liblangtag1 liblirc-client0 liblua5.3-0 libmediaart-2.0-0
 libmsspub-0.1-1 libodfgen-0.1-1 libqwing2v5 libraw16 librevenge-0.0-0 libsgutils2-2 libssh-4
 libsuitesparseconfig5 libvncclient1 libwinpr2-2 libxapian30 libxmisc1-nss lp-solve
 media-player-info python3-mako python3-markupsafe syslinux syslinux-common syslinux-legacy
 usb-creator-common
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
 apt-transport-https
0 upgraded, 1 newly installed, 0 to remove and 3 not upgraded.
Need to get 4,348 B of archives.
After this operation, 154 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 apt-transport-https all 1.6.14 [4,348 B]
Fetched 4,348 B in 3s (1,255 B/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 115707 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_1.6.14_all.deb ...
Unpacking apt-transport-https (1.6.14) ...
Setting up apt-transport-https (1.6.14) ...
root@ubuntu:/home/tesiscarrascodelg#
```

Figura 5.11: Configuración del repositorio de chirpstack.

Para configurar la llave del servidor se ejecuta la siguiente línea de código como se puede observar en la Fig.5.12.

```
root@ubuntu:/home/tesiscarrascodelg# sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys ICE2AFD36DBCCA00
Executing: /tmp/apt-key-gpghome.uv42d5bxbA/gpg.1.sh --keyserver keyserver.ubuntu.com --recv-keys ICE2AFD36DBCCA00
gpg: key ICE2AFD36DBCCA00: public key "Orne Brocaar <info@brocaar.com>" imported
gpg: Total number processed: 1
gpg:      imported: 1
root@ubuntu:/home/tesiscarrascodelg#
```

Figura 5.12: Configuración de la llave del servidor de chirpstack.

Para visualizar los diferentes paquetes de chirpstack se ejecuta la siguiente línea de código como se puede observar en la Fig.5.13.

```
root@ubuntu:/home/tesiscarrascodelg# sudo echo "deb https://artifacts.chirpstack.io/packages/3.x
/deb stable main" | sudo tee /etc/apt/sources.list.d/chirpstack.list
deb https://artifacts.chirpstack.io/packages/3.x/deb stable main
root@ubuntu:/home/tesiscarrascodelg#
```

Figura 5.13: Visualización de los diferentes paquetes de chirpstack.

Para actualizar la lista de paquetes disponibles, de acuerdo con la Fig.5.14 se ejecuta la siguiente línea de código:

```

root@ubuntu:/home/tesiscarrascodelog# sudo apt update
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Hit:2 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Get:3 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:4 https://artifacts.chirpstack.io/packages/3.x/deb stable InRelease [14.0 kB]
Get:5 https://artifacts.chirpstack.io/packages/3.x/deb stable/main i386 Packages [3,324 B]
Get:6 https://artifacts.chirpstack.io/packages/3.x/deb stable/main amd64 Packages [3,329 B]
Get:7 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Fetched 273 kB in 8s (35.9 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
3 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ubuntu:/home/tesiscarrascodelog#

```

Figura 5.14: Actualización de la lista de paquetes disponibles.

5.2 Instalación de Chirpstack Gateway Bridge

Como se puede observar en la Fig.5.15 para instalar el paquete "chirpstack-gateway-bridge" se ejecuta la siguiente línea de código.

```

root@ubuntu:/home/tesiscarrascodelog# sudo apt install chirpstack-gateway-bridge
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
 fonts-liberation2 fonts-opensymbol gir1.2-gst-plugins-base-1.0 gir1.2-gstreamer-1.0
 gir1.2-gudev-1.0 gir1.2-udisks-2.0 grilo-plugins-0.3-base gstreamer1.0-gtk3
 libboost-date-time1.65.1 libboost-fsystem1.65.1 libboost-iostreams1.65.1
 libboost-locale1.65.1 libcdr-0.1-1 libclucene-contribs1v5 libclucene-core1v5 libcms-0.5-5v5
 libcolamd2 libdazzle-1.0-0 libdumbnet1 libe-book-0.1-1 libedataseverul-1.2-2 libeot0
 libepubgen-0.1-1 libetonyek-0.1-1 libevent-2.1-6 libexiv2-14 libfreerdp-client2-2
 libfreerdp2-2 libgic2 libgee-0.8-2 libgexiv2-2 libgom-1.0-0 libgpgmepp6 libgpod-common
 libgpod4 liblangtag-common liblangtag1 liblirc-client0 liblua5.3-0 libmediaart-2.0-0
 libmspub-0.1-1 libodfgen-0.1-1 libqwing2v5 libraw16 librevenge-0.0-0 libsgutils2-2 libssh-4
 libsuitesparseconfigs libvncclient1 libwinpr2-2 libxapian3 libxnlseci-nss lp-solve
 media-player-info python3-nako python3-narkupsafe syslinux syslinux-common syslinux-legacy
 usb-creator-common
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
 chirpstack-gateway-bridge
0 upgraded, 1 newly installed, 0 to remove and 3 not upgraded.
Need to get 4,550 kB of archives.
After this operation, 12.3 MB of additional disk space will be used.
Get:1 https://artifacts.chirpstack.io/packages/3.x/deb stable/main amd64 chirpstack-gateway-bridge amd64 3.13.3 [4,550 kB]
Fetched 4,550 kB in 12s (393 kB/s)
Selecting previously unselected package chirpstack-gateway-bridge.
(Reading database ... 115711 files and directories currently installed.)
Preparing to unpack .../chirpstack-gateway-bridge_3.13.3_amd64.deb ...
Unpacking chirpstack-gateway-bridge (3.13.3) ...
Setting up chirpstack-gateway-bridge (3.13.3) ...

```

Figura 5.15: Instalación del paquete "chirpstack-gateway-bridge".

Como se puede observar en la Fig.5.16 para iniciar el servicio de "chirpstack-gateway-bridge" se ejecuta la siguiente línea de código:


```
root@ubuntu:/home/tesiscarrascodeleg# sudo systemctl start chirpstack-gateway-bridge
root@ubuntu:/home/tesiscarrascodeleg#
```

Figura 5.16: Inicio del servicio de "chirpstack-gateway-bridge".

Para habilitar el servicio de "chirpstack-gateway-bridge" se ejecuta la siguiente línea de código como se puede observar en la Fig.5.17.

```
root@ubuntu:/home/tesiscarrascodeleg# sudo systemctl enable chirpstack-gateway-bridge
root@ubuntu:/home/tesiscarrascodeleg#
```

Figura 5.17: Habilitación del servicio de "chirpstack-gateway-bridge".

5.3 Instalación de Chirpstack Network Server

Para instalar el paquete "chirpstack-network-server" se ejecuta la siguiente línea de código como se puede observar en la Fig.5.18.

```
root@ubuntu:/home/tesiscarrascodeleg# sudo apt install chirpstack-network-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
 fonts-liberation2 fonts-opensymbol gir1.2-gst-plugins-base-1.0 gir1.2-gstreamer-1.0
 gir1.2-gudev-1.0 gir1.2-udisks-2.0 grilo-plugins-0.3-base gstreamer1.0-gtk3
 libboost-date-time1.65.1 libboost-filesystem1.65.1 libboost-iostreams1.65.1
 libboost-locale1.65.1 libcdr-0.1-1 libclucene-contribs1v5 libclucene-core1v5 libcmis-0.5-5v5
 libcolamd2 libdazzle-1.0-0 libdumbnet1 libe-book-0.1-1 libdataserverui-1.2-2 libeot0
 libepubgen-0.1-1 libetonyek-0.1-1 libevent-2.1-6 libexiv2-14 libfreerdp-client2-2
 libfreerdp2-2 libgc1c2 libgee-0.8-2 libgexiv2-2 libgon-1.0-0 libgpgmepp0 libgpod-common
 libgpod4 liblangtag-common liblangtag1 liblirc-client0 liblua5.3-0 libmediaart-2.0-0
 libmsspub-0.1-1 libodfgen-0.1-1 libqwing2v5 libraw16 librevenge-0.0-0 libsgutils2-2 libssh-4
 libsuitesparseconfigs libvncclient1 libwinpr2-2 libxapian3 libxmlsec1-nss lp-solve
 media-player-info python3-mako python3-markupsafe syslinux syslinux-common syslinux-legacy
 usb-creator-common
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
 chirpstack-network-server
0 upgraded, 1 newly installed, 0 to remove and 3 not upgraded.
Need to get 8,109 kB of archives.
After this operation, 24.2 MB of additional disk space will be used.
Get:1 https://artifacts.chirpstack.io/packages/3.x/deb stable/main amd64 chirpstack-network-serve
r amd64 3.16.1 [8,109 kB]
Fetched 8,109 kB in 1min 22s (98.8 kB/s)
Selecting previously unselected package chirpstack-network-server.
(Reading database ... 115719 files and directories currently installed.)
Preparing to unpack .../chirpstack-network-server_3.16.1_amd64.deb ...
Unpacking chirpstack-network-server (3.16.1) ...
Setting up chirpstack-network-server (3.16.1) ...
```

Figura 5.18: Instalación del paquete "chirpstack-network-server".

Para iniciar el servicio "chirpstack-network-server" se ejecuta la siguiente línea de código

cómo se puede observar en la Fig.5.19.

```
root@ubuntu:/home/testscarrascodeleg# sudo systemctl start chirpstack-network-server
root@ubuntu:/home/testscarrascodeleg#
```

Figura 5.19: Inicio del servicio de chirpstack-network-server.

Para habilitar el servicio de 'chirpstack-network-server' se ejecuta la siguiente línea de código como se puede observar en la Fig.5.20.

```
root@ubuntu:/home/testscarrascodeleg# sudo systemctl enable chirpstack-network-server
root@ubuntu:/home/testscarrascodeleg#
```

Figura 5.20: Habilitación del servicio de chirpstack-network-server.

Antes de inicializar el servicio 'chirpstack-network-server' se debe realizar cambios en el archivo chirpstack-network-server.toml, tal como se puede observar en la Fig.5.21.

postgreSQL

- dsn="postgres://chirpstack_ns:dbcuenca@localhost/chirpstack_ns?sslmode=disable"

network_server.band

- name= 'US_902_928'

- nano /etc/chirpstack-network-server/chirpstack-network-server.toml

```
GNU nano 2.9.3 /etc/chirpstack-network-server/chirpstack-network-server.toml Modified
# * sslmode - Whether or not to use SSL (default is require, this is not the default for libpq)
# * fallback_application_name - An application_name to fall back to if one isn't provided.
# * connect_timeout - Maximum wait for connection, in seconds. Zero or not specified means wait $
# * sslcert - Cert file location. The file must contain PEM encoded data.
# * sslkey - Key file location. The file must contain PEM encoded data.
# * sslrootcert - The location of the root certificate file. The file must contain PEM encoded d$
#
# Valid values for sslmode are:
#
# * disable - No SSL
# * require - Always SSL (skip verification)
# * verify-ca - Always SSL (verify that the certificate presented by the server was signed by a $
# * verify-full - Always SSL (verify that the certification presented by the server was signed b$
dsn="postgres://chirpstack_ns:dbcuenca@localhost/chirpstack_ns_ns?sslmode=disable"
```

Figura 5.21: Integración con la base de datos creada.

Como se puede observar en la Fig.5.22 para configurar el nombre de la banda del servidor de red a utilizar se agregará la siguiente línea de código:

- name='US_902_928'

```
# Network-server settings.
[network_server]
# Network identifier (NetID, 3 bytes) encoded as HEX (e.g. 010203)
net_id="000000"

# LoRaWAN regional band configuration.
#
# Note that you might want to consult the LoRaWAN Regional Parameters
# specification for valid values that apply to your region.
# See: https://www.lora-alliance.org/lorawan-for-developers
[network_server.band]
name="US_902_928"
```

Figura 5.22: Nombre de la banda del servidor de red.

5.4 Instalación de Chirpstack Application Server

La instalación de este servicio es muy importante debido a que en ella se mostrará la interfaz de la aplicación. Como se puede observar en la Fig.5.23:

```
root@ubuntu:/home/tesiscarrascodelog# sudo apt install chirpstack-application-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
 fonts-liberation2 fonts-opensymbol gir1.2-gst-plugins-base-1.0 gir1.2-gstreamer-1.0
 gir1.2-gudev-1.0 gir1.2-udisks-2.0 grilo-plugins-0.3-base gstreamer1.0-gtk3
 libboost-date-time1.65.1 libboost-filesystem1.65.1 libboost-iostreams1.65.1
 libboost-locale1.65.1 libcdr-0.1-1 libclucene-contribs1v5 libclucene-core1v5 libcmis-0.5-5v5
 libcolamd2 libdazzle-1.0-0 libdumbnet1 libe-book-0.1-1 libedataserverut-1.2-2 libeot0
 libepubgen-0.1-1 libetonyek-0.1-1 libevent-2.1-6 libexiv2-14 libfreerdp-client2-2
 libfreerdp2-2 libgc1c2 libgee-0.8-2 libgexiv2-2 libgom-1.0-0 libgpgmepp6 libgpod-common
 libgpod4 liblangtag-common liblangtag1 liblirc-client0 liblua5.3-0 libmediaart-2.0-0
 libmspub-0.1-1 libodfgen-0.1-1 libqqwing2v5 libraw16 libvenge-0.0-0 libsgutils2-2 libssh-4
 libsuitesparseconfig5 libvncclient1 libwinpr2-2 libxapian3 libxmlsec1-nss lp-solve
 media-player-info python3-nako python3-markupsafe syslinux syslinux-common syslinux-legacy
 usb-creator-common
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
 chirpstack-application-server
0 upgraded, 1 newly installed, 0 to remove and 3 not upgraded.
Need to get 14.0 MB of archives.
After this operation, 43.8 MB of additional disk space will be used.
Get:1 https://artifacts.chirpstack.io/packages/3.x/deb stable/main amd64 chirpstack-application-s
erver amd64 3.17.6 [14.0 MB]
Fetched 14.0 MB in 1min 6s (213 kB/s)
Selecting previously unselected package chirpstack-application-server.
(Reading database ... 115730 files and directories currently installed.)
Preparing to unpack .../chirpstack-application-server_3.17.6_amd64.deb ...
Unpacking chirpstack-application-server (3.17.6) ...
Setting up chirpstack-application-server (3.17.6) ...
```

Figura 5.23: Instalación del paquete chirpstack-application-server.

Para realizar la configuración del archivo "chirpstack-application-server.toml" se procede a ubicarse en la ruta donde este se encuentre.

Posteriormente se agrega la siguiente línea de código como se puede observar en la Fig.5.24.

```
GNU nano 2.9.3 /etc/chirpstack-application-server/chirpstack-application-server.toml
# * connect_timeout - Maximum wait for connection, in seconds. Zero or not specified means wait $
# * sslcert - Cert file location. The file must contain PEM encoded data.
# * sslkey - Key file location. The file must contain PEM encoded data.
# * sslrootcert - The location of the root certificate file. The file must contain PEM encoded d$
#
# Valid values for sslmode are:
#
# * disable - No SSL
# * require - Always SSL (skip verification)
# * verify-ca - Always SSL (verify that the certificate presented by the server was signed by a $
# * verify-full - Always SSL (verify that the certification presented by the server was signed b$
dsn="postgres://chirpstack_as:dbcuenca@localhost/chirpstack_as?sslmode=disable"
```

Figura 5.24: Integración con la base de datos "chirpstack_as".

Como se puede observar en la Fig.5.25 se procede a ejecutar una nueva terminal la cuál mostrará una clave cifrada.


```
root@ubuntu:/home/tesiscarrascodeleg# openssl rand -base64 32
XrJI0660XJgWdKmurNqITsPdJGIG4ZXw2caQohnF9P8=
root@ubuntu:/home/tesiscarrascodeleg#
```

Figura 5.25: Generación aleatoria del token.

Como se puede observar en la Fig.5.26 se procede a configurar el parámetro 'jwt_secret=""' en el archivo 'chirpstack-application-server.toml', en el cual para configurar la clave cifrada obtenida anteriormente se agrega la siguiente línea de código.

```
# Settings for the "external api"
#
# This is the API and web-interface exposed to the end-user.
[application_server.external_api]
# ip:port to bind the (user facing) http server to (web-interface and REST / gRPC api)
bind="0.0.0.0:8080"

# http server TLS certificate (optional)
tls_cert=""

# http server TLS key (optional)
tls_key=""

# JWT secret used for api authentication / authorization
# You could generate this by executing 'openssl rand -base64 32' for example
jwt_secret="XrJI0660XJgWdKmurNqITsPdJGIG4ZXw2caQohnF9P8="
```

Figura 5.26: Ingreso del token generado aleatoriamente.

Para iniciar el servicio 'chirpstack-application-server' se ejecuta la siguiente línea de código como puede observar en la Fig.5.27.

- sudo systemctl start chirpstack-application-server

```
root@ubuntu:/home/tesiscarrascodeleg# sudo systemctl start chirpstack-application-server
root@ubuntu:/home/tesiscarrascodeleg#
```

Figura 5.27: Inicio del servicio de chirpstack-application-server.

Para habilitar el servicio 'chirpstack-application-server' se ejecuta la siguiente línea de código como se puede observar en la Fig.5.28.

- `sudo systemctl enable chirpstack-application-server`

```
root@ubuntu:/home/tesiscarrascodeleg# sudo systemctl enable chirpstack-application-server
root@ubuntu:/home/tesiscarrascodeleg#
```

Figura 5.28: Habilitación del servicio de chirpstack-application-server.

Por último, como se puede observar en la Fig.5.29 se ejecuta el siguiente comando que permite obtener los logs del servicio de ChirpStack.

```
root@ubuntu:/home/tesiscarrascodeleg# sudo journalctl -f -n 100 -u chirpstack-application-server
-- Logs begin at Tue 2022-04-26 05:36:18 PDT. --
Apr 26 13:53:55 ubuntu systemd[1]: Started ChirpStack Application Server.
Apr 26 13:53:55 ubuntu chirpstack-application-server[88129]: time="2022-04-26T13:53:55.655162176-07:00" level=info msg="starting ChirpStack Application Server" docs="https://www.chirpstack.io/" version=3.17.6
Apr 26 13:53:55 ubuntu chirpstack-application-server[88129]: time="2022-04-26T13:53:55.655253185-07:00" level=info msg="storage: setup metrics"
Apr 26 13:53:55 ubuntu chirpstack-application-server[88129]: time="2022-04-26T13:53:55.655264025-07:00" level=info msg="storage: setting up Redis client"
Apr 26 13:53:55 ubuntu chirpstack-application-server[88129]: time="2022-04-26T13:53:55.65529307-07:00" level=info msg="storage: PostgreSQL data migrations applied" from_version=0 to_version=60
Apr 26 13:53:55 ubuntu chirpstack-application-server[88129]: time="2022-04-26T13:53:55.660195506-07:00" level=info msg="storage: applying PostgreSQL data migrations"
Apr 26 13:53:56 ubuntu chirpstack-application-server[88129]: time="2022-04-26T13:53:56.697754716-07:00" level=info msg="integration: configuring global integrations"
Apr 26 13:53:56 ubuntu chirpstack-application-server[88129]: time="2022-04-26T13:53:56.705932419-07:00" level=info msg="integration/mqtt: TLS config is empty"
Apr 26 13:53:56 ubuntu chirpstack-application-server[88129]: time="2022-04-26T13:53:56.705972589-07:00" level=info msg="integration/mqtt: connecting to mqtt broker" server="tcp://localhost:1883"
Apr 26 13:53:56 ubuntu chirpstack-application-server[88129]: time="2022-04-26T13:53:56.706414342-07:00" level=info msg="api/as: starting application-server api" bind="0.0.0.0:8001" ca_cert= tls_cert= tls_key=
Apr 26 13:53:56 ubuntu chirpstack-application-server[88129]: time="2022-04-26T13:53:56.706669192-07:00" level=info msg="api/external: starting api server" bind="0.0.0.0:8080" tls_cert= tls_key=
Apr 26 13:53:56 ubuntu chirpstack-application-server[88129]: time="2022-04-26T13:53:56.706789152-07:00" level=info msg="integration/mqtt: connected to mqtt broker"
Apr 26 13:53:56 ubuntu chirpstack-application-server[88129]: time="2022-04-26T13:53:56.706804008-07:00" level=info msg="integration/mqtt: subscribing to tx topic" qos=0 topic=application/+device/+command/down
Apr 26 13:53:56 ubuntu chirpstack-application-server[88129]: time="2022-04-26T13:53:56.807961459-07:00" level=info msg="api/external: registering rest api handler and documentation endpoint" path=/api
Apr 26 13:53:56 ubuntu chirpstack-application-server[88129]: time="2022-04-26T13:53:56.808032629-07:00" level=info msg="api/js: starting join-server api" bind="0.0.0.0:8003" ca_cert= tls_cert= tls_key=
```

Figura 5.29: Visualización de logs del servidor chirpstack.

Como parte final para acceder a la dirección IP del servidor chirpstack, se ingresa la siguiente dirección IP '34.148.16.7' más el puerto 8080 como se puede observar en la Fig.5.30.

- En la página web se ingresa a la siguiente URL: 'http://34.148.16.7:8080'.
- Para realizar la autenticación en la plataforma chirpstack se debe colocar el usuario y contraseña.

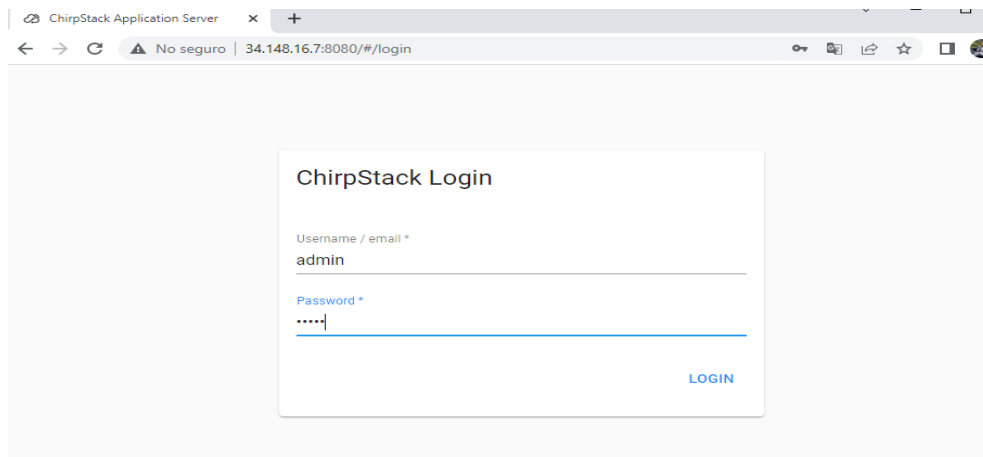


Figura 5.30: Autenticación en la página de ChirpStack.

En la Fig.5.31 se observa el panel principal de ChirpStack, donde se visualiza los dispositivos y gateways activos.

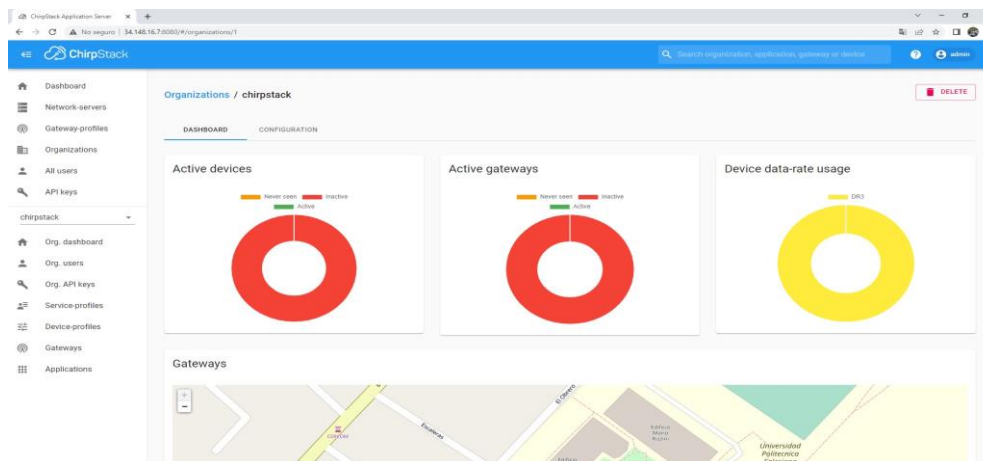


Figura 5.31: Ingreso al panel principal de ChirpStack.

Para finalizar este proceso se debe integrar con una base de datos nueva para poder obtener los datos requeridos, para ello se accede a la consola del servidor y se ejecuta los siguientes pasos detallados a continuación:

Como se puede observar en la Fig.5.32 para acceder como superusuario a postgresql se coloca el comando: 'sudo -u postgresql psql', una vez accedido para crear el rol se ejecuta la siguiente línea de código:

```
bryanca272@chirpstack:~$ sudo -u postgres psql
psql (10.21 (Ubuntu 10.21-0ubuntu0.18.04.1))
Type "help" for help.

postgres=# create role chirpstack_as_events with login password 'dbcuencia';
```

Figura 5.32: Creación de rol 'chirpstack_as_events'.

Para crear la base de datos 'chirpstack_as_events' se ejecuta la siguiente línea de código como se puede observar en la Fig.5.33.

```
postgres=# create database chirpstack_as_events with owner chirpstack_as_events;
CREATE DATABASE
postgres=#
```

Figura 5.33: Creación de la base de datos 'chirpstack_as_events'.

Para acceder a la base de datos creada anteriormente se ejecuta la siguiente línea de código como se puede observar en la Fig.5.34.

- \c chirpstack_as_events

```
postgres=# \c chirpstack_as_events
You are now connected to database "chirpstack_as_events" as user "postgres".
chirpstack_as_events=#
```

Figura 5.34: Conexión a la nueva base de datos 'chirpstack_as_events'.

Para crear la extensión 'hstore' se ejecuta la siguiente línea de código como se puede observar en la Fig.5.35.

```
chirpstack_as_events=# create extension hstore;
CREATE EXTENSION
chirpstack_as_events=#
```

Figura 5.35: Creación de la extensión 'hstore'.

Como se puede observar en la Fig.5.36 se verifica que exista conexión con la base creada.

```
bryanca272@chirpstack:~$ psql -h localhost -U chirpstack_as_events -W chirpstack_as_events
Password for user chirpstack_as_events:
psql (10.21 (Ubuntu 10.21-0ubuntu0.18.04.1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off
)
Type "help" for help.

chirpstack_as_events=> |
```

Figura 5.36: Verificación de conexión a la base de datos 'chirpstack_as_events'.

Para la configuración del archivo "chirpstack-application-server.toml" como se puede observar en la Fig.5.37 en primer lugar se dirige a la ruta '/etc/chirpstack-application-server' y posteriormente se edita el parámetro 'enabled=["mqtt"]' y se lo reemplaza con: ["mqtt", "postgresql"].

A continuación, al inferior de la línea de código '[application_server.integration.postgresql]' se agrega lo siguiente.

```
[application_server.integration]
# Payload marshaler.
#
# This defines how the MQTT payloads are encoded. Valid options are:
# * protobuf: Protobuf encoding
# * json:      JSON encoding (easier for debugging, but less compact than 'protobuf')
# * json_v3:  v3 JSON (will be removed in the next major release)
marshaler="json_v3"

# Enabled integrations.
enabled=["mqtt", "postgresql"]

[application_server.integration.postgresql]
dsn="postgres://chirpstack_as_events:bcuencia@localhost/chirpstack_as_events?sslmode=disable"
```

Figura 5.37: Configuración del archivo 'chirpstack-application-server.toml'.

Para reiniciar el servicio 'chirpstack-application-server' se ejecuta la siguiente línea de código como se puede observar en la Fig.5.38.

```
root@chirpstack:~# systemctl restart chirpstack-application-server
root@chirpstack:~# systemctl status chirpstack-application-server
● chirpstack-application-server.service - ChirpStack Application Server
   Loaded: loaded (/lib/systemd/system/chirpstack-application-server.service; enabled; vendor
   Active: active (running) since Thu 2022-06-02 12:08:08 -05; 4s ago
     Docs: https://www.chirpstack.io/
    Main PID: 12463 (chirpstack-appl)
      Tasks: 7 (limit: 4656)
   CGroup: /system.slice/chirpstack-application-server.service
           └─12463 /usr/bin/chirpstack-application-server

Jun 02 12:08:09 chirpstack chirpstack-application-server[12463]: time="2022-06-02T12:08:09.055
Jun 02 12:08:09 chirpstack chirpstack-application-server[12463]: time="2022-06-02T12:08:09.055
Jun 02 12:08:09 chirpstack chirpstack-application-server[12463]: time="2022-06-02T12:08:09.056
Jun 02 12:08:09 chirpstack chirpstack-application-server[12463]: time="2022-06-02T12:08:09.056
Jun 02 12:08:09 chirpstack chirpstack-application-server[12463]: time="2022-06-02T12:08:09.056
Jun 02 12:08:09 chirpstack chirpstack-application-server[12463]: time="2022-06-02T12:08:09.058
Jun 02 12:08:09 chirpstack chirpstack-application-server[12463]: time="2022-06-02T12:08:09.062
Jun 02 12:08:09 chirpstack chirpstack-application-server[12463]: time="2022-06-02T12:08:09.062
Jun 02 12:08:09 chirpstack chirpstack-application-server[12463]: time="2022-06-02T12:08:09.164
Jun 02 12:08:09 chirpstack chirpstack-application-server[12463]: time="2022-06-02T12:08:09.164
lines 1-19/19 (END)
```

Figura 5.38: Estado del servicio chirpstack-application-server.

Por último, se verifica el estado del servicio ejecutando la siguiente línea de código:

- `systemctl status chirpstack-application-server`

5.5 Implementación de la autenticación en los dispositivos de la red LoRaWAN.

1. Para poder autenticar tanto el servidor como el dispositivo final, dentro del Gateway se debe realizar la siguiente configuración:
 - (a) Para acceder a la interfaz del Gateway se coloca la dirección IP '192.168.88.1', posteriormente se ingresa las credenciales para una autenticación correcta como se puede observar en la Fig.5.39.

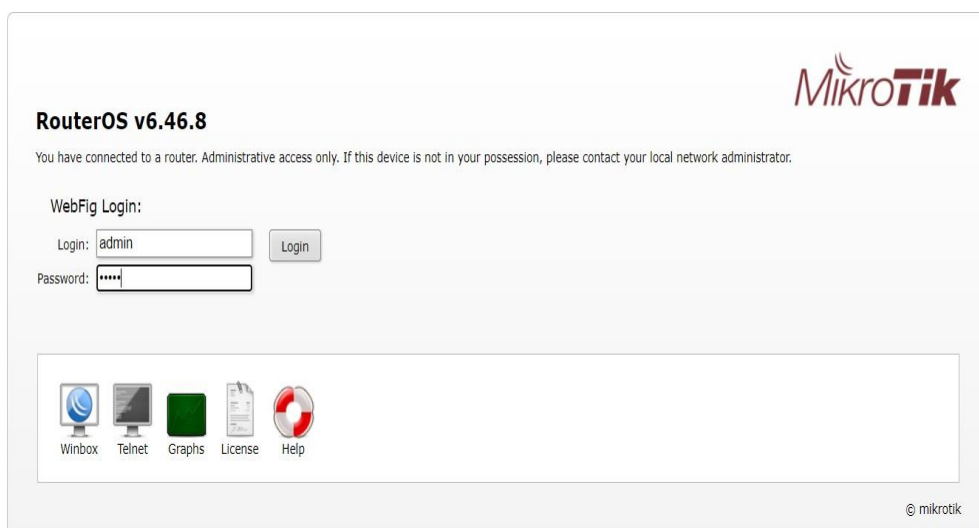


Figura 5.39: Inicio de sesión Gateway MikroTik.

- (b) Una vez que se accede al panel del gateway MikroTik se coloca una contraseña de seguridad para lograr autenticarse a la red Wi-Fi del gateway como se puede observar en la Fig.5.40.

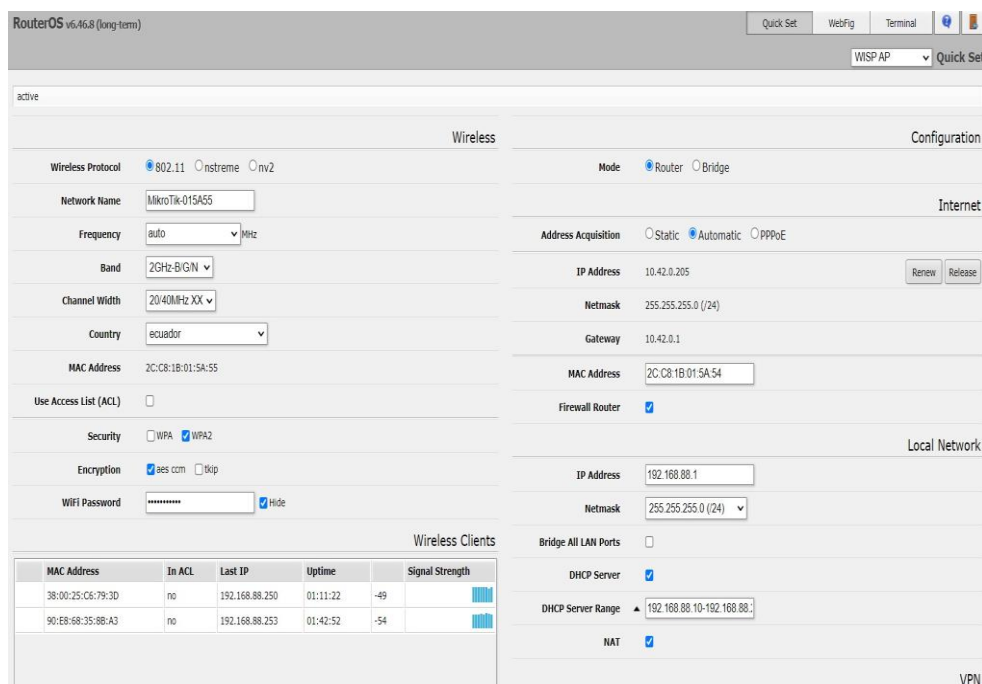


Figura 5.40: Contraseña del Wi-Fi del Gateway MikroTik.

- (c) Luego de configurar la contraseña en el menú del Gateway, se procede a seleccionar

el apartado de LoRa como se puede observar en la Fig.5.41.

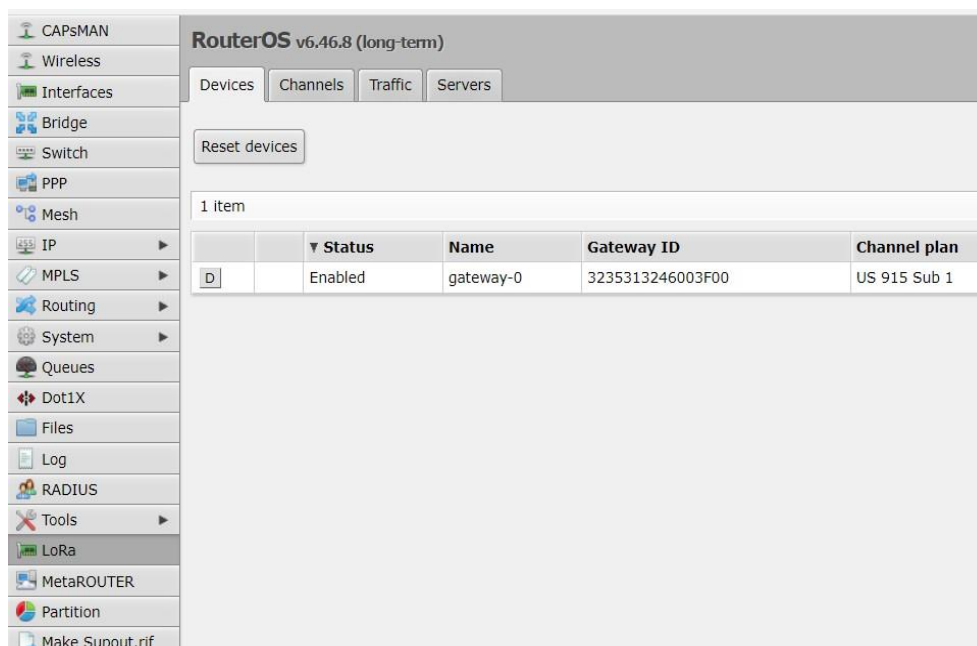


Figura 5.41: Apartado de LoRa.

- (d) Se agrega el servidor de ChirpStack en la pestaña 'Servers' dentro del apartado de LoRa como se puede observar en la Fig.5.42.

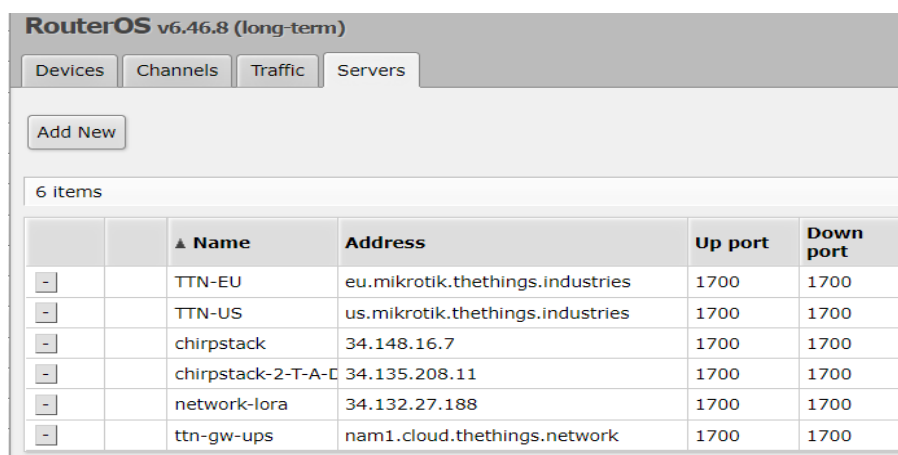


Figura 5.42: Agregar el servidor de ChirpStack al Gateway MikroTik.

- (e) Como indica la Fig.5.43 se completa los campos ingresando el nombre del servidor de ChirpStack, la IP remota del mismo, el número de puerto y posteriormente se guarda los cambios.

RouterOS v6.46.8 (long-term)

OK Cancel Apply

Name chirpstack

Address 34.148.16.7

Up port 1700

Down port 1700

Figura 5.43: Datos del Servidor de ChirpStack.

- (f) Como se puede observar en la Fig.5.44 se regresa a la pestaña 'Devices' y posteriormente se apaga el servidor por un momento, a continuación, se ingresa al mismo para realizar los cambios.

RouterOS v6.46.8 (long-term)

Devices Channels Traffic Servers

Reset devices

1 item

	▼ Status	Name	Gateway ID	Channel plan
D	Enabled	gateway-0	3235313246003F00	US 915 Sub 1

Figura 5.44: Apagar el servidor.

- (g) Como se puede observar en la Fig.5.45 se agrega el servidor configurado anteriormente, se debe tener en cuenta que se puede tener registrado y conectado varios servidores al mismo tiempo dentro del Gateway, posteriormente se guarda los cambios y se activa el gateway.

RouterOS v6.46.8 (long-term)

OK Cancel Apply

Enabled

Status Enabled

Name gateway-0

Hardware ID 3235313246003F00

Gateway ID 3235313246003F00

Network Servers

- ▼ chirpstack ▼ ▲
- ▼ network-lora ▼ ▲
- ▼ chirpstack-2-T-A-D ▼ ▲
- ▼ ttn-gw-ups ▼ ▲

Channel plan US 915 Sub 1 ▼

Antenna Gain 0 dB

Forward Valid Error Disabled

Network Public Private

LBT

Src. Address ▼

Figura 5.45: Agregar el Servidor de red.

5.6 Implementación de ChirpStack en Amazon Web Services.

En la Fig.5.46 dentro de la plataforma de Chirpstack tenemos varias pestañas, pero debemos centrarnos en la pestaña de Application server.

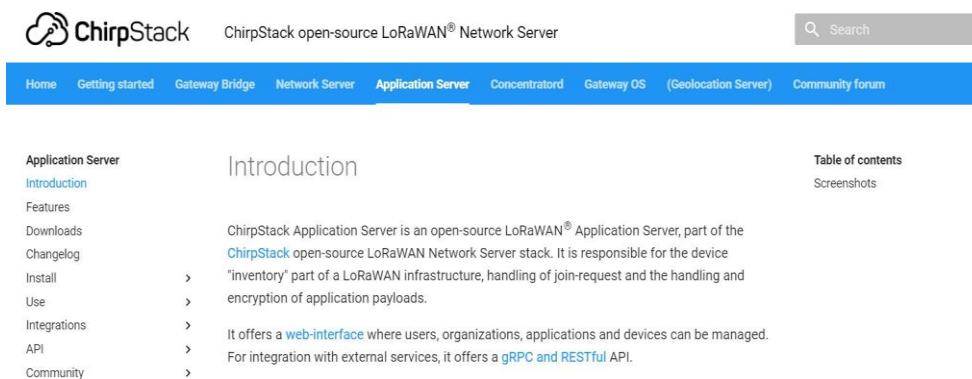


Figura 5.46: Plataforma de ChirpStack.

Como se puede observar en la Fig.5.47 se puede visualizar dentro del menú lateral la pestaña 'Integrations' dentro de esta se encuentra diferentes formas de integración de ChirpStack con otras plataformas.

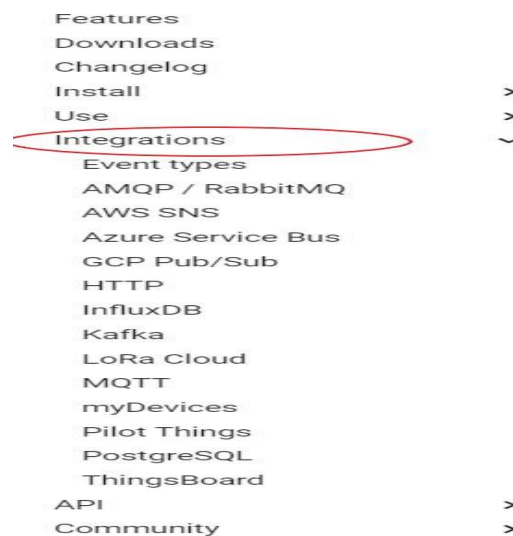


Figura 5.47: Pestaña de integraciones de ChirpStack.

Dentro de la pestaña 'Integrations' se accede a Amazon SNS donde se observa la información sobre la integración con ChirpStack, la cual describe que Amazon SNS publica todos los eventos en un tema de SNS donde se puede suscribir servicios y aplicaciones para su procesamiento.

5.6.1 Atributos del mensaje

En la página de ChirpStack indica que los siguientes atributos se agregan en cada mensaje publicado:

- **event:** define el tipo de evento.
- **dev_eui:** define el dispositivo EUI.
- **application-id:** define el ID de la aplicación del servidor de aplicaciones ChirpStack.

Con esta información que se encuentra dentro de la misma página de Chirpstack se visualiza que la integración con Amazon Web Service SNS es posible, por lo cual se puede proceder a realizar la integración.

5.7 Implementar Amazon Web Service para obtener la data de ChirpStack

Pasos a seguir para la implementación de Amazon Web Services.

5.7.1 Creación de cuenta en Amazon Web Service.

Como punto principal se dirige a la página web '<https://aws.amazon.com/es/?nc2=hlg>'.

Como se puede observar en la Fig.5.48 se visualiza la página de registro en la plataforma Amazon Web Services (AWS).

Registrarse en AWS

Dirección de correo electrónico del usuario raíz
Se utiliza para la recuperación de cuentas y algunas funciones administrativas

tesis_carrascodeleg@hotmail.com

Nombre de la cuenta de AWS
Elija un nombre para la cuenta. Podrá cambiarlo en la configuración de la cuenta después de registrarse.

Tesis_LoraWAN

Verificar la dirección de correo electrónico

O

Iniciar sesión en una cuenta de AWS existente

Figura 5.48: Registro en Amazon Web Services (AWS).

Para la confirmación del registro en AWS se envía un código de verificación a la dirección de correo electrónico ingresado como se puede observar en la Fig.5.49.

Registrarse en AWS

Confirme que es usted

Garantizar que esté seguro, es lo que hacemos.

Hemos enviado un correo electrónico con un código de verificación a **tesis_carrascodeleg@hotmail.com**. [¿No es usted?](#)

Introdúzcalo a continuación para confirmar su correo electrónico.

Código de verificación

Verificar

Volver a enviar el código

¿No ha recibido el código?

- Los códigos pueden tardar hasta 5 minutos en llegar.
- Revise su carpeta de spam.

Figura 5.49: Verificación de correo electrónico.

Para la confirmación del registro en AWS se ingresa el código de verificación que se envió a la dirección de correo electrónico ingresado como se puede observar en la Fig.5.50.

Registrarse en AWS

Confirme que es usted

Garantizar que esté seguro, es lo que hacemos.

Hemos enviado un correo electrónico con un código de verificación a **tesis_carrascodeleg@hotmail.com**. (¿No es usted?)

Introdúzcalo a continuación para confirmar su correo electrónico.

Código de verificación

Verificar

Volver a enviar el código

¿No ha recibido el código?

- Los códigos pueden tardar hasta 5 minutos en llegar.
- Revise su carpeta de spam.

Figura 5.50: Código de verificación de correo electrónico.

Una vez verificado el código ingresado se procede a crear la contraseña para la cuenta de AWS como se puede observar en la Fig.5.51.

Registrarse en AWS

Cree la contraseña

✓ Hemos verificado su identidad. ✕
La dirección de correo electrónico se ha verificado correctamente.

La contraseña proporciona acceso de inicio de sesión a AWS, por lo que es importante que este proceso se realice de forma correcta.

Contraseña de usuario raíz:

Confirmar la contraseña del usuario raíz:

Continuar (paso 1 de 5)

Iniciar sesión en una cuenta de AWS existente

Figura 5.51: Creación de la Contraseña para la cuenta de "Amazon Web Services (AWS)".

Para completar el primer paso se procede a ingresar una contraseña para la cuenta de "Amazon Web Services (AWS)" como se puede observar en la Fig.5.52.

Registrarse en AWS

Cree la contraseña

✔ Hemos verificado su identidad. La dirección de correo electrónico se ha verificado correctamente. ✕

La contraseña proporciona acceso de inicio de sesión a AWS, por lo que es importante que este proceso se realice de forma correcta.

Contraseña de usuario raíz:

Confirmar la contraseña del usuario raíz:

Continuar (paso 1 de 5)

O

Iniciar sesión en una cuenta de AWS existente

Figura 5.52: Confirmación de la contraseña para el registro en AWS.

Para completar el segundo paso se agrega la información de contacto como: Nombres completos, Nombre de la Organización, Número de teléfono, País o región, Dirección y Código postal como se puede observar en la Fig.5.53.

Registrarse en AWS

Información de contacto

¿Cómo tiene previsto utilizar AWS?

Empresarial: para su trabajo, escuela u organización

Personal: para sus propios proyectos

¿A quién debemos contactar para consultar sobre esta cuenta?

Nombre completo

Nombre de la organización

Número de teléfono

Introduzca el código de país y el número de teléfono.

País o región

Dirección

Ciudad

Estado, provincia o región

Código postal

He leído y acepto los términos del Contrato de usuario de AWS [↗](#)

Continuar (paso 2 de 5)

Figura 5.53: Información de contacto para el registro en AWS.

Para completar el tercer paso se ingresa la información para la facturación como: Número de tarjeta de crédito o débito, Fecha de vencimiento, Nombre de titular de la tarjeta y la dirección de facturación como se puede observar en la Fig.5.54.

Registrarse en AWS

Información de facturación

Número de tarjeta de crédito o débito



AWS acepta todas las tarjetas de crédito y débito principales. Para obtener más información sobre las opciones de pago, consulte nuestras preguntas frecuentes.

Fecha de vencimiento

Julio

Nombre del titular de la tarjeta

Bryan Carrasco

Dirección de facturación

Utilizar mi dirección de contacto

Av. Héroes de Verdeloma y la Merced
Cuenca Azuay 010107
EC

Utilizar una nueva dirección

Verificar y continuar (paso 3 de 5)

Es posible que se le redirija al sitio web de su banco para autorizar el cargo de verificación.

Figura 5.54: Información de facturación en AWS.

Para completar el cuarto paso se confirma la identidad con un código enviado al número de teléfono ingresado como se puede observar en la Fig.5.55.

Registrarse en AWS

Confirme su identidad

Para poder utilizar la cuenta de AWS, debe verificar su número de teléfono. Cuando continúe, el sistema automatizado de AWS se comunicará con usted para proporcionarle un código de verificación.

Código de país o región

Ecuador (+593)

Número de teléfono móvil

0987198940

Comprobación de seguridad



Escriba los caracteres como se indica arriba

2w2xnm

Enviar SMS (paso 4 de 5)

Figura 5.55: Confirmación de identidad para el registro en AWS.

Se ingresa el código enviado al número de celular ingresado como se puede observar en la Fig.5.56.

Registrarse en AWS

Confirme su identidad

Verificar código

6109

Continuar (paso 4 de 5)

¿Tiene algún problema? A veces, se necesitan hasta 10 minutos para recibir el código de verificación. Si ha transcurrido más tiempo del mencionado, vuelva a la página anterior e inténtelo de nuevo.

Figura 5.56: Ingreso de código enviado al celular ingresado.

Para completar en último paso se selecciona el plan de soporte a utilizar y finalizamos el registro como se puede observar en la Fig.5.57.

Registrarse en AWS

Seleccionar un plan de soporte

Elija un plan de soporte para su cuenta personal o empresarial. Compare planes y ejemplos de precio. Puede cambiar su plan en cualquier momento desde la consola de administración de AWS.

<input checked="" type="radio"/> Soporte de nivel Básic: gratis <ul style="list-style-type: none">Recomendado para los usuarios nuevos que recién comienzan a utilizar AWS.Acceso de autoservicio las 24 horas del día, los 7 días de la semana a los recursos de AWS.Solo para problemas de facturación y cuentas.Acceso a Personal Health Dashboard y Trusted Advisor. 	<input type="radio"/> Soporte Developer: a partir de 29 USD al mes <ul style="list-style-type: none">Recomendado para desarrolladores que experimentan con AWS.Acceso por correo electrónico a AWS Support durante el horario laboral.Tiempos de respuesta de 12 horas (horario laboral). 	<input type="radio"/> Soporte Business: a partir de 100 USD al mes <ul style="list-style-type: none">Recomendado para ejecutar cargas de trabajo de producción en AWS.Soporte técnico las 24 horas, los 7 días de la semana por correo electrónico, teléfono y chat.Tiempos de respuesta de 1 hora.Conjunto completo de recomendaciones de prácticas de Trusted Advisor. 
--	--	--

¿Necesita soporte de nivel Enterprise?
A partir de los 15 000 USD por mes, tendrá tiempos de respuesta de 15 minutos y una experiencia de conseje con un director técnico de cuenta asignado. Más información

Finalizar registro

Figura 5.57: Selección del plan de soporte.

Como se puede observar en la Fig.5.58 el registro en Amazon Web Services se completó correctamente.



Figura 5.58: Visualización del registro completado exitosamente.

5.7.2 Ingreso a Amazon Web Services (AWS)

Para ingresar a la consola de administración de AWS, se coloca la dirección de e-mail del usuario raíz anteriormente creado como se observa en la Fig.5.59.



Figura 5.59: Inicio de sesión en "Amazon Web Services (AWS)".

Para realizar la comprobación de seguridad se ingresa los caracteres que se muestra en la Fig.5.60.



Figura 5.60: Comprobación de seguridad.

Para acceder a la cuenta de "Amazon Web Services (AWS)" se ingresa la contraseña creada anteriormente como se puede observar en la Fig.5.61.



Figura 5.61: Verificación de seguridad.

Una vez ingresado a la consola de administración de AWS se procede a configurar el "Simple Notification Service (SNS)" y "Simple Queue Service (SQS)" en AWS.

5.7.3 Configuración de Simple Notification Service (SNS)

Para acceder al servicio SNS a implementar en la barra de búsqueda se escribe 'Simple Notification Service (SNS)' como se puede observar en la Fig.5.62.

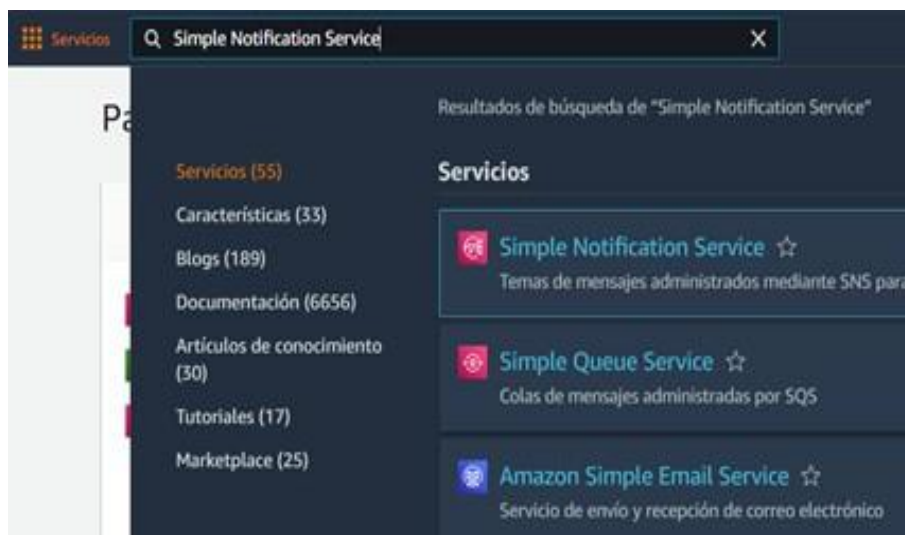


Figura 5.62: Buscar el servicio Simple Notification Service (SNS).

Como se puede observar en la Fig.5.63 se procede a crear un tema en Simple Notification Service (SNS) con el nombre que se pueda identificar el tema que se va a realizar.



Figura 5.63: Crear un tema en Simple Notification Service (SNS).

Como se puede observar en la Fig.5.64 luego de tener creado el tema se procede a seleccionar uno de los dos tipos los cuales son FIFO y Estándar.

- **FIFO:** Se usa para casos que requieren ordenar y duplicar mensajes.
- **Estándar:** Se usa para casos que requieren mayores tasas de procesamiento de publicación y entrega de mensajes.

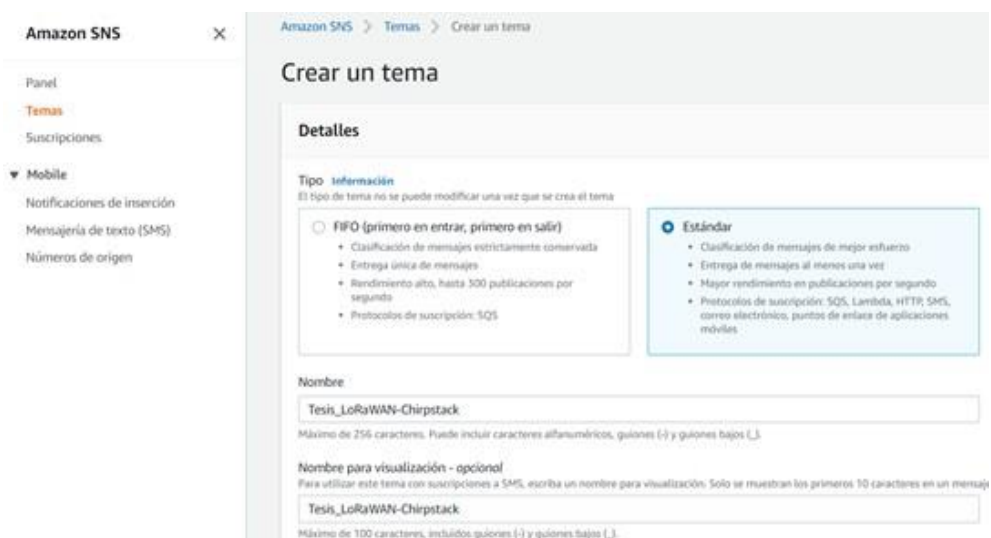


Figura 5.64: Tipos de temas de Amazon SNS.

Como se puede observar en la Fig.5.65 una vez creado el tema aparece un mensaje que indica que el tema está creado correctamente.

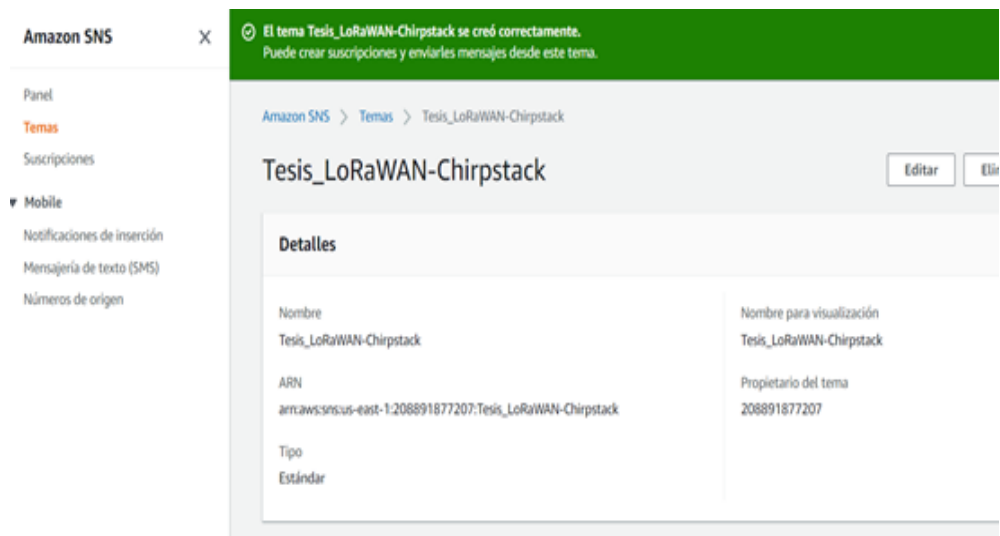


Figura 5.65: Confirmación del tema creado.

5.7.4 Configuración de Simple Queue Service (SQS)

Como se puede observar en la Fig.5.66 se busca el servicio 'Simple Queue Service (SQS)' a implementar.



Figura 5.66: Buscar el servicio Simple Queue Service (SQS).

Como se puede observar en la Fig.5.67 se crea una nueva cola Simple Queue Service (SQS).



Figura 5.67: Creación de una nueva cola SQS.

Como se puede observar en la Fig.5.68 se selecciona el tipo de cola que se desea aplicar en este caso se elige 'Estándar', posteriormente se procede a darle un nombre el cual es "Tesis_LoRaWAN-Chirpstack-Queue".

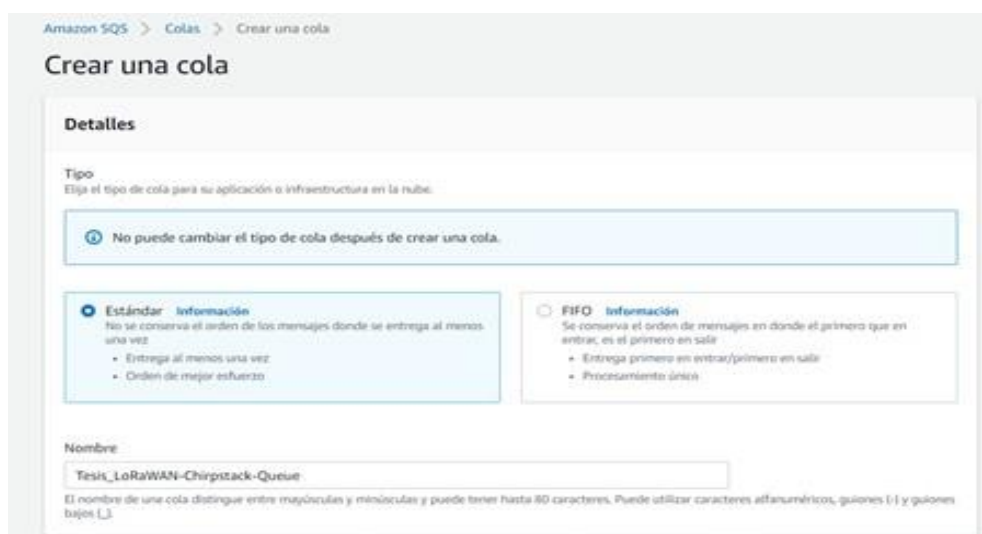


Figura 5.68: Configuración de la cola SQS.

Como se puede observar en la Fig.5.69 en el apartado de configuración se selecciona la configuración que está por defecto.

Configuración
Establezca el tamaño máximo del mensaje, la visibilidad para otros consumidores y la retención de mensajes. [Información](#)

Tiempo de espera de visibilidad [Información](#)
30 Segundos
Debe estar entre 0 segundos y 12 horas.

Periodo de retención del mensaje [Información](#)
4 Días
Debe estar entre 1 minuto y 14 días.

Retraso de entrega [Información](#)
0 Segundos
Debe estar entre 0 segundos y 15 minutos.

Tamaño máximo del mensaje [Información](#)
256 KB
Debe estar entre 1 KB y 256 KB.

Tiempo de espera de recepción del mensaje [Información](#)
0 Segundos
Debe estar entre 0 y 20 segundos.

Figura 5.69: Selección de la configuración de la cola SQS.

Como se puede observar en la Fig.5.70 en el apartado de políticas de acceso se selecciona la configuración que está por defecto.

Política de acceso
Defina quién puede tener acceso a la cola. [Información](#)

Elegir método

Básico
Utilice criterios sencillos para definir una política de acceso básica.

Avanzado
Utilice un objeto JSON para definir una política de acceso avanzada.

Defina quién puede enviar mensajes a la cola

Solo el propietario de la cola
Solo el propietario de la cola puede enviar mensajes a la cola.

Solo las cuentas de AWS, los usuarios y las funciones de IAM especificados
Solo los ID de cuenta de AWS, los usuarios y las funciones de IAM especificados pueden enviar mensajes a la cola.

Defina quién puede recibir mensajes de la cola

Solo el propietario de la cola
Solo el propietario de la cola puede recibir mensajes de la cola.

Solo las cuentas de AWS, los usuarios y las funciones de IAM especificados
Solo los ID de cuenta de AWS, los usuarios y las funciones de IAM especificados pueden recibir mensajes de la cola.

JSON (solo lectura)

```
{
  "Version": "2008-10-17",
  "Id": "__default_policy_ID",
  "Statement": [
    {
      "Sid": "_owner_statement",
      "Effect": "Allow",
      "Principal": {
        "AWS": "208891877207"
      },
      "Action": [
        "SQS:*"
      ],
      "Resource": "arn:aws:sqs:us-east-1:208891877207:Tesis_LoRaWAN-Chirpstack-Queue"
    }
  ]
}
```

Figura 5.70: Políticas de acceso de la cola SQS.

Como se puede observar en la Fig.5.71 una vez concluida la configuración de SQS se debe guardar y verificar que la cola se creó correctamente.

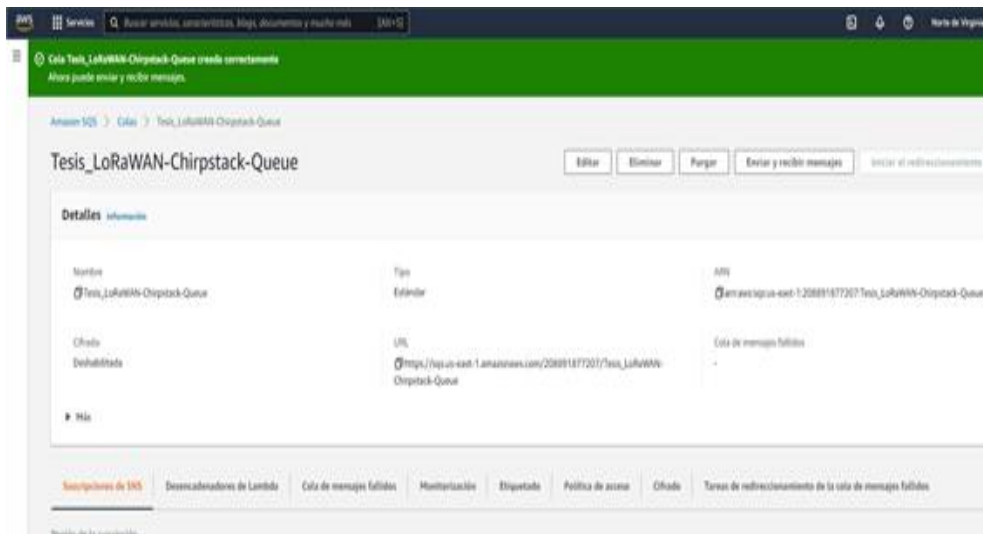


Figura 5.71: Cola SQS creada correctamente.

Como se puede observar en la Fig.5.72 se ingresa a la cola SQS recientemente creada.

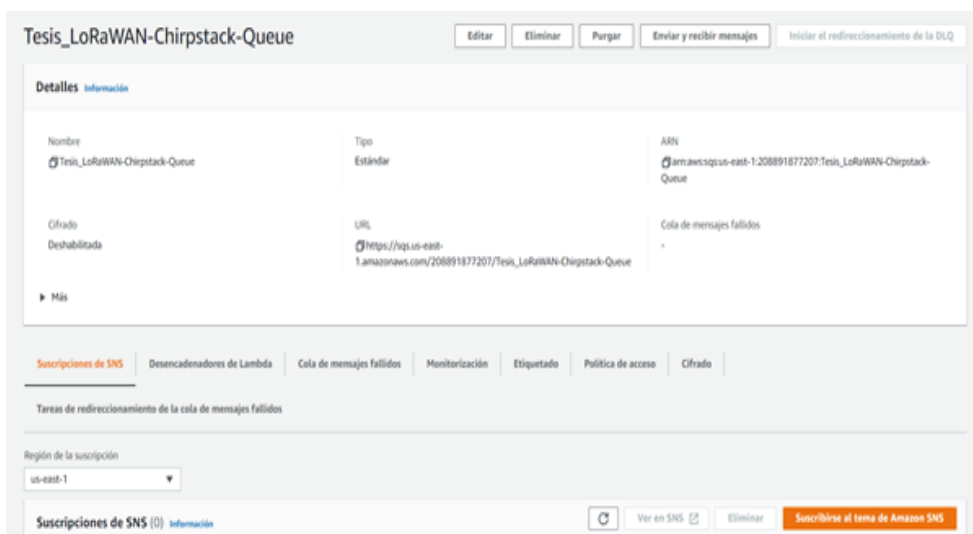


Figura 5.72: Ingreso a la cola SQS creada.

Como se puede observar en la Fig.5.73 se da clic en la opción 'suscribirse al tema de Amazon SNS' en la parte inferior derecha.

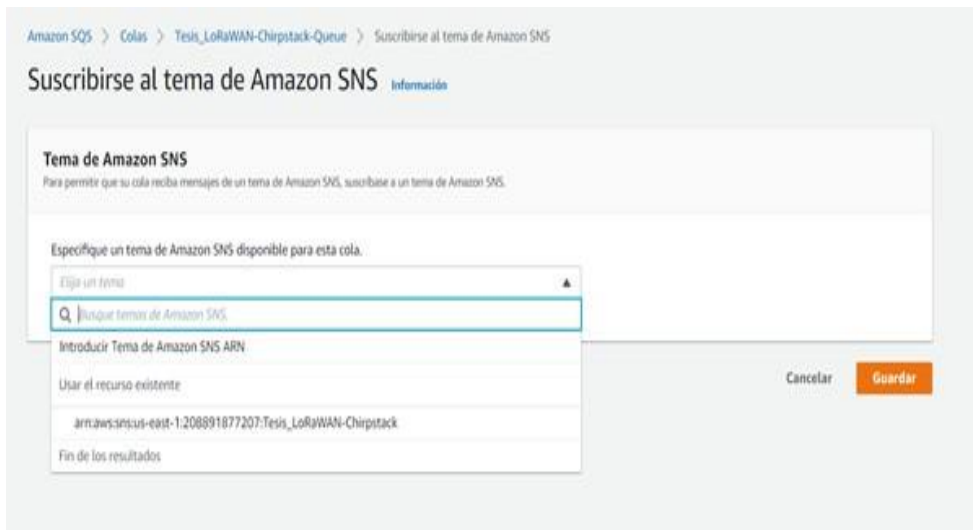


Figura 5.73: Suscripción al tema de Amazon SNS.

Como se puede observar en la Fig.5.74 se selecciona el ARN del tema de "Amazon SNS" que se creó anteriormente y se procede a guardar.



Figura 5.74: Selección del tema de Amazon SNS disponible para la cola.

Para visualizar la relación que existe entre el ARN de suscripción y el ARN del tema se procede a guardar el tema de Amazon SNS como se puede observar en la Fig.5.75.

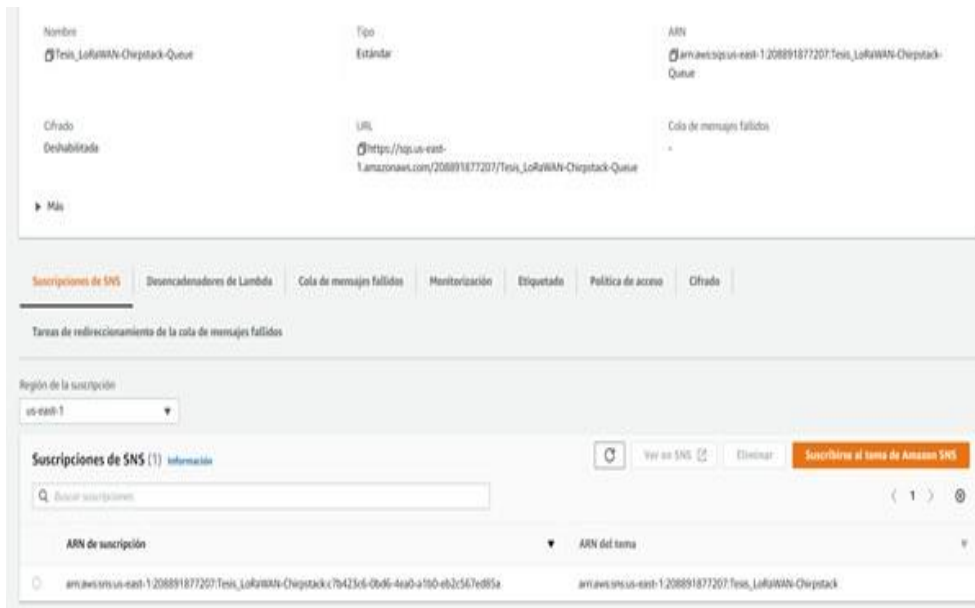


Figura 5.75: Relación entre el ARN de suscripción y el tema de Amazon SNS.

5.7.5 Conectar "Simple Notification Service (SNS)" a "Simple Queue Service (SQS)"

En el panel de Amazon SNS para ingresar a los temas creados anteriormente se da clic en la sección 'Temas' como se puede observar en la Fig.5.76.

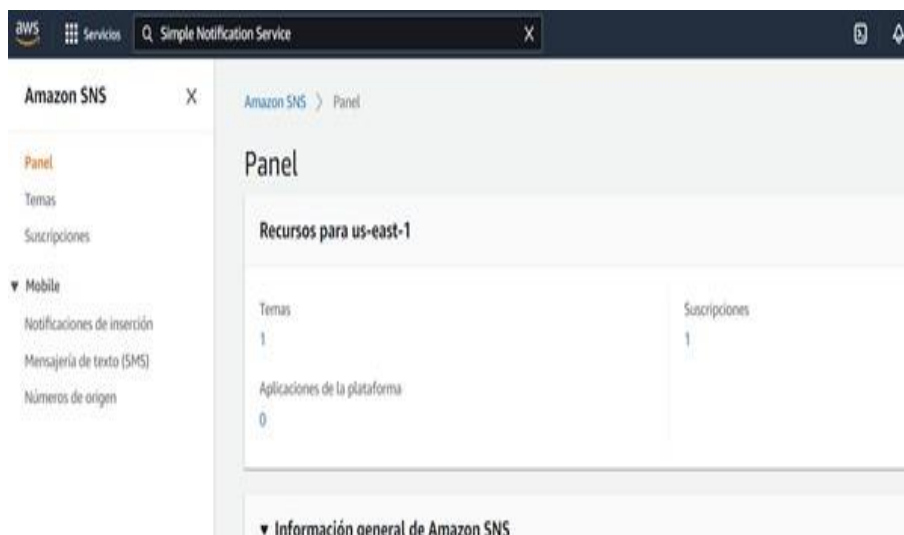


Figura 5.76: Panel de Amazon SNS.

Como se puede observar en la Fig.5.77 se visualiza los temas creados en este caso solamente aparece uno el cual ya se creó anteriormente.



Figura 5.77: Visualización de temas creados en Amazon SNS.

Como se puede observar en la Fig.5.78 al final se cuenta con: un ID de suscripción, el estado en el que se encuentra y además el protocolo que se está usando.

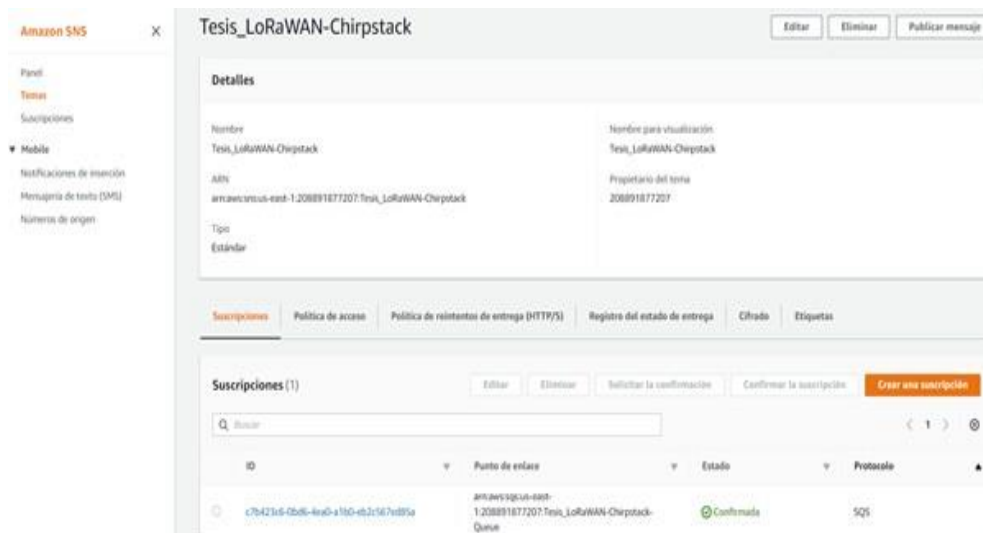


Figura 5.78: ID de suscripción en Amazon SNS.

Como se puede observar en la Fig.5.79 se da clic en el ID y se visualiza más información de la suscripción, adicionalmente se procede a editar haciendo clic en el botón 'editar' que se

encuentra en la parte inferior derecha.

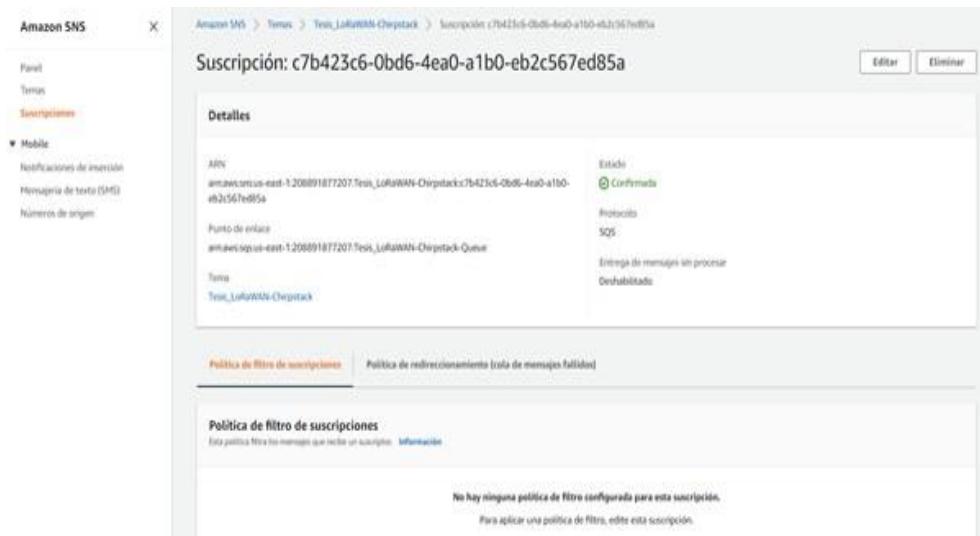


Figura 5.79: Detalles de la suscripción de Amazon SNS.

Como se puede observar en la Fig.5.80 se marca la casilla 'Habilitar la entrega de mensajes sin procesar', con el fin de que se elimine cualquier cosa específica de AWS del mensaje, con esto el mensaje tendrá la forma en la que se envía originalmente y se procede a guardar los cambios realizados.

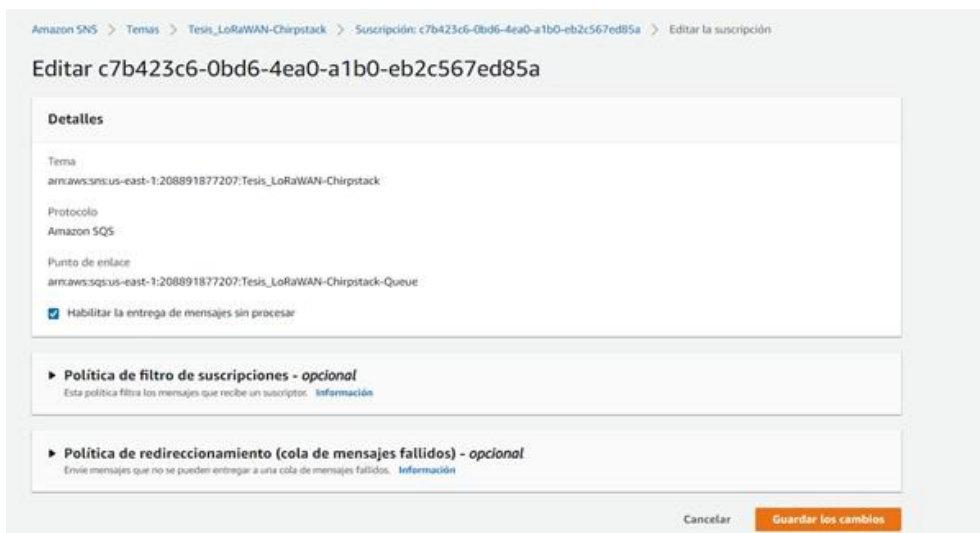


Figura 5.80: Habilitación de la entrega de mensajes sin procesar.

Como se puede observar en la Fig.5.81 una vez guardado con éxito la suscripción se visual-

iza la confirmación de que los cambios se guardaron correctamente.

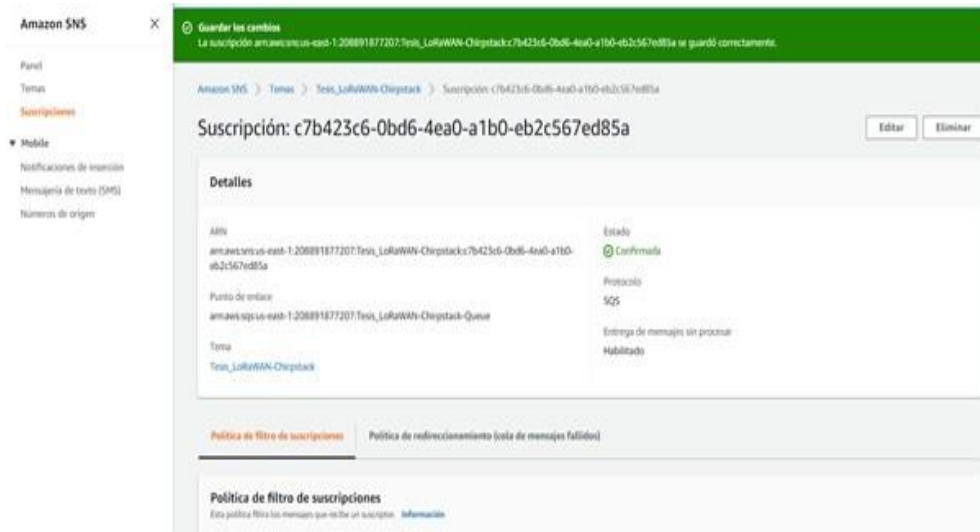


Figura 5.81: Cambios guardados correctamente de la suscripción de Amazon SNS.

5.8 Obtener una mejor visualización de todos los datos que se tienen de ChirpStack.

En este punto se investigará la forma de poder implementar un Dashboard el cual nos permitirá tener una mejor visualización de los datos obtenidos.

Dentro de la plataforma de "Amazon Web Services" se cuenta con el servicio "QuickSight" como podemos observar en la Fig.5.82, donde se realizará el Dashboard, este servicio ofrece varias alternativas de poder obtener y visualizar los datos, dentro de esta función se encuentra la de poder conectarse a una base de datos.



Figura 5.82: Logo de "Amazon QuickSight".

QuickSight permite que una organización pueda comprender datos a través de paneles interactivos o también con la búsqueda de patrones de manera automática.

5.9 Analizar las herramientas de Amazon Web Service para almacenamiento, análisis y visualización de datos.

Para poder integrar la plataforma de ChirpStack con AWS debemos seguir los siguientes pasos que se explican a continuación.

Como se puede observar en la Fig.5.83 dentro de la plataforma de ChirpStack seleccionamos la pestaña 'Aplicaciones'.

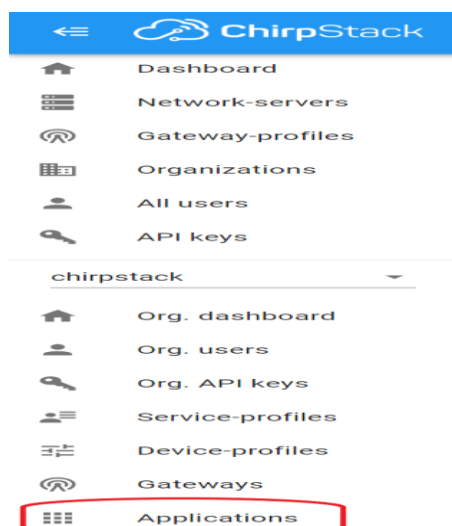


Figura 5.83: Menú de ChirpStack.

Como se puede observar en la Fig.5.84 en la sección de 'aplicaciones' únicamente se visualiza las aplicaciones creadas en este caso contamos con solamente una.

Applications

ID	Name	Service-profile	Description
10	MEDIDORES	LORA-SERVER	MEDIDORES


Rows per page: 10 ▾ 1-1 of 1

Figura 5.84: Aplicaciones creadas en la plataforma ChirpStack.


Al acceder a la aplicación creada como se puede observar en la Fig.5.85 seleccionamos la pestaña 'INTEGRATIONS'.

Applications / MEDIDORES


DEVICES MULTICAST GROUPS APPLICATION CONFIGURATION **INTEGRATIONS**



AWS SNS
The AWS SNS integration forwards events to an AWS SNS topic.
[EDIT](#) [REMOVE](#)




MQTT
The MQTT integration forwards events to a MQTT broker.
[GET CERTIFICATE](#)




Azure Service-Bus
The Azure Service-Bus integration forwards events to an Azure Service-Bus topic or queue.
[ADD](#)

HTTP : //

HTTP
The HTTP integration forwards events to a user-configurable endpoint as POST requests.
[ADD](#)



InfluxDB
The InfluxDB integration writes events into an InfluxDB time-series database.
[ADD](#)



myDevices
The myDevices integration forwards events to the myDevices platform.
[ADD](#)

Figura 5.85: Lista de integraciones de ChirpStack.

Como se puede observar en la Fig.5.86 se selecciona el primer ítem (AWS SNS), el cual permite realizar la integración con Amazon y se procede a llenar los datos correspondientes.

DEVICES MULTICAST GROUPS APPLICATION CONFIGURATION **INTEGRATIONS**

Update AWS SNS integration

Payload marshaler *
JSON

This defines how the payload will be encoded.

AWS region *
us-east-1

AWS Access Key ID *
AKIATBIXNC5LZRCUBV6Z

AWS Secret Access Key *

AWS SNS topic ARN *
arn:aws:sns:us-east-1:208891877207:Tesis_LoRaWAN-Chirpstack

[UPDATE INTEGRATION](#)

Figura 5.86: Integración con "Amazon Web Services SNS".

Como se puede observar en la Fig.5.87 al final de completar los campos se pide un topic ARN, para obtener ese topic se debe dirigir a la página de Amazon SQS en el apartado de colas.

Amazon SQS > Colas

Colas (1) 🔄 Editar Eliminar Enviar y recibir mensajes Acciones ▾ Crear una cola

🔍 *Buscar colas por prefijo* < 1 > 🔊

Nombre	Tipo	Creado	Mensajes disponibles	Mensajes en tránsito	Cifrado	Desduplicación basada en el contenido
Tesis_LoRaWAN-Chirpstack-Queue	Estándar	22/4/2022, 11:46:07 GMT-5	12	0	Deshabilitada	-

Figura 5.87: Colas Amazon SQS

Dentro de la única cola que se creó anteriormente, como se puede observar en la Fig.5.88 se visualiza la información del ARN la cual se copia y pega para poder completar la integración.

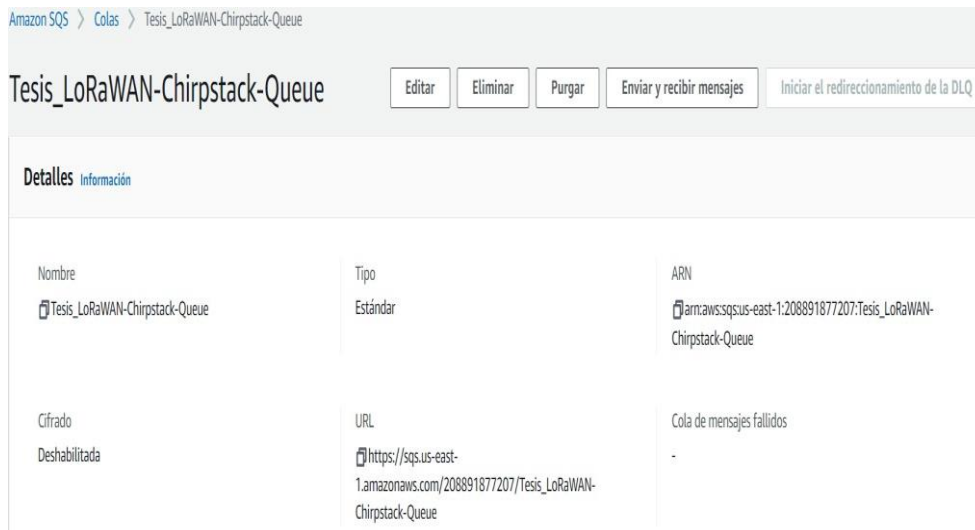


Figura 5.88: Visualización del ARN creado.

Una vez que se realiza la integración se selecciona la pestaña 'Enviar y recibir mensajes' que se encuentra en la página de Amazon SQS como se puede observar en la Fig.5.89.

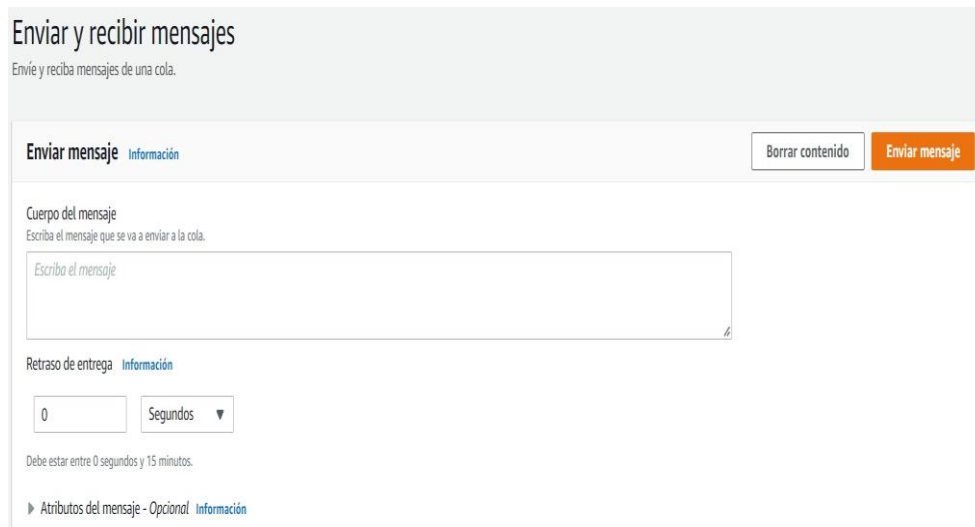


Figura 5.89: Enviar y Recibir mensajes de la cola SQS.

Como se puede observar en la Fig.5.90 en la parte baja se encuentra la pestaña 'Sondeo de mensajes' al hacer clic empezará a escanear los datos obtenidos en la plataforma de ChirpStack.

Recepción de mensajes Información

Editar la configuración de sondeo Detener el sondeo Sondeo de mensajes

Mensajes disponibles: 12 Duración del sondeo: 30 Recuento máximo del mensaje: 50 Progreso del sondeo: 24% (4 recepciones/segundo)

Mensajes (12) Ver detalles Eliminar

Q Buscar mensajes < 1 > ⌂

ID	Enviado	Tamaño	Recuento de recepción
2b91fc52-b15c-4217-a52b-a300e00fbf45	10/6/2022, 12:01:41 GMT-5	999 bytes	1
c4fd5dc0-5fd9-44f0-a6cd-7aee7bee1a5c	10/6/2022, 12:01:31 GMT-5	1003 bytes	1

Figura 5.90: Sondeo de Mensajes en Amazon SQS.

Una vez terminado el sondeo se elige cualquier mensaje y se podrá visualizar el contenido de dicho mensaje como se puede observar en la Fig.5.91

Mensaje: 2b91fc52-b15c-4217-a52b-a300e00fbf45 ✕

Detalles **Cuerpo** Atributos

```
{
  "applicationID": "9",
  "applicationName": "LORA-TESES",
  "deviceName": "T11637100373",
  "devEUI": "VQ+gtQIAAAA=",
  "rxInfo": {
    "gatewayID": "MjUxMkYAPwA=",
    "time": "2022-06-10T17:01:41.536261Z",
    "timeSinceGPSEPOCH": null,
    "rssi": -113,
    "loRaSNR": 3,
    "channel": 0,
    "rfChain": 0,
    "board": 0,
    "antenna": 0,
    "location": {
      "latitude": -2.88656238102356,
      "longitude": -78.99088917580572,
      "altitude": 2500,
      "source": "UNKNOWN",
      "accuracy": 0,
      "fineTimestampType": "NONE",
      "context": "ZJL9JA==",
      "uplinkID": "arW+AZQoS1CX4XF3nf3NrA==",
      "crcStatus": "CRC_OK"
    },
    "txInfo": {
      "frequency": 902300000,
      "modulation": "LORA",
      "loRaModulationInfo": {
        "bandwidth": 125,
        "spreadingFactor": 7,
        "codeRate": "4/5",
        "polarizationInversion": false
      }
    },
    "adr": true,
    "dr": 3,
    "fCnt": 2,
    "fPort": 0,
    "data": null,
    "objectJSON": {
      "Data": "",
      "tags": {}
    },
    "confirmedUplink": false,
    "devAddr": "AVttag==",
    "publishedAt": "2022-06-10T17:01:41.794699904Z",
    "deviceProfileID": "6fb3f8ae-4ba4-4cb9-b2cb-72cd5250fbe1",
    "deviceProfileName": "LORA-DEVICE"
  }
}
```

Figura 5.91: Cuerpo del Mensaje Sondeado en Amazon SQS.

5.10 Analizar y visualizar los datos de manera eficiente utilizando la herramienta QuickSight.

Se utilizará el servicio 'QuickSight' para tener una visualización de los datos a través de la base de datos de Chirpstack, para lo cual primeramente se realiza ciertas modificaciones a la base de datos de PostgreSQL para que se pueda conectar de manera remota a la base de datos que contiene la información necesaria.

5.10.1 Configuración en la consola del servidor ChirpStack

Como se puede observar en la Fig.5.92 se dirige a la carpeta 'main' de postgresql con la siguiente línea de código: 'cd/etc/postgresql/10/main'.

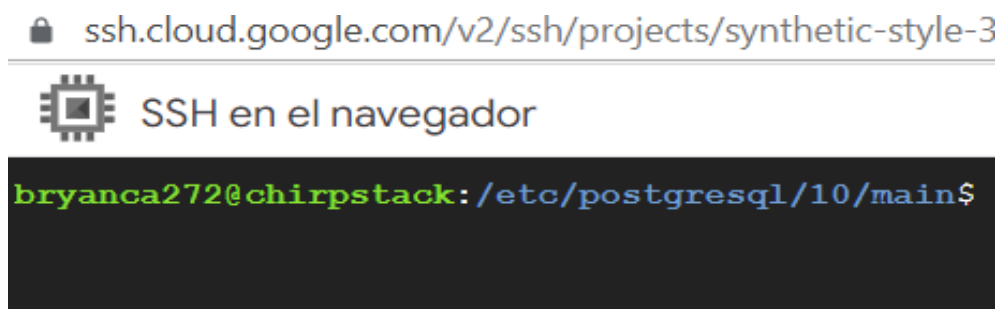


Figura 5.92: Ruta de la carpeta de postgresql en la consola del servidor.

Como se puede observar en la Fig.5.93 se edita el archivo 'postgresql.conf', posteriormente se agrega la siguiente línea: listen_addresses = '*'

```
#-----  
# CONNECTIONS AND AUTHENTICATION  
#-----  
#  
# - Connection Settings -  
listen_addresses = '*'  
  
#listen_addresses = 'localhost'          # what IP address(es) to listen on;  
                                         # comma-separated list of addresses;
```

Figura 5.93: Configuración del archivo 'postgresql.conf'.

Como se puede observar en la Fig.5.94 se reinicia el servicio de 'postgresql' con la siguiente línea de código: 'service postgresql restart'. Posteriormente se configura el archivo 'pg_hba.conf' que se encuentra dentro de la misma carpeta donde se realizó la configuración anterior, luego se ingresa al archivo con la siguiente línea de código: 'sudo nano pg_hba.conf', en el apartado 'IPv4 local connections' se debe agregar lo siguiente: 'host-all-all-md5'.

```
# Database administrative login by Unix domain socket  
local all postgres peer  
  
# TYPE DATABASE USER ADDRESS METHOD  
  
# "local" is for Unix domain socket connections only  
local all all peer  
# IPv4 local connections:  
host all all 127.0.0.1/32 md5  
  
host all all all md5  
# IPv6 local connections:  
host all all ::1/128 md5  
# Allow replication connections from localhost, by a user with the  
# replication privilege.  
local replication all peer  
host replication all 127.0.0.1/32 md5  
host replication all ::1/128 md5
```

Figura 5.94: Configuración del archivo 'pg_hba.conf'.

Como se puede observar en la Fig.5.95 se reinicia el servicio de 'postgresql' con la siguiente línea de código: 'service postgresql restart'. Por último, se verifica que el estado del servicio postgresql esté funcionando correctamente con la siguiente línea de código: 'service postgresql status'.

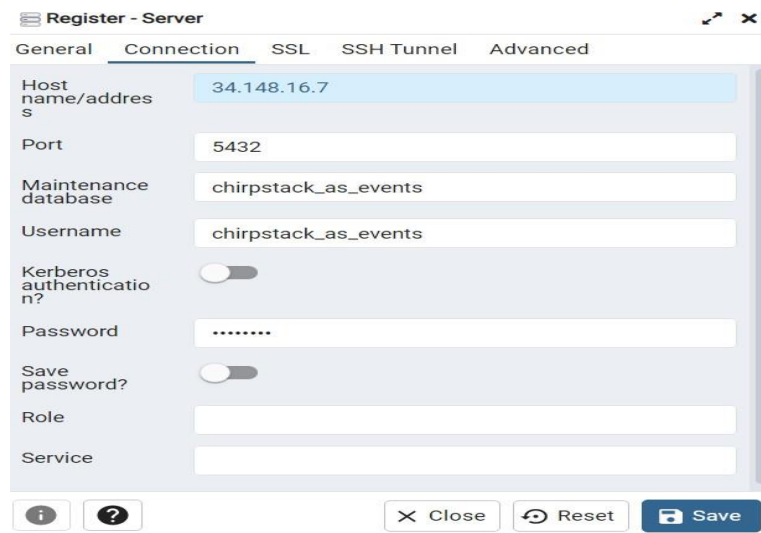
```
bryanca272@chirpstack:~$ service postgresql status
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
   Active: active (exited) since Wed 2022-06-22 22:27:52 -05; 1 day 1h ago
   Process: 13321 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
   Main PID: 13321 (code=exited, status=0/SUCCESS)

Jun 22 22:27:52 chirpstack systemd[1]: Starting PostgreSQL RDBMS...
Jun 22 22:27:52 chirpstack systemd[1]: Started PostgreSQL RDBMS.
bryanca272@chirpstack:~$
```

Figura 5.95: Estado del servicio PostgreSQL.

5.10.2 Comprobación de la conexión remota desde pgAdmin

Como se puede observar en la Fig.5.96 primeramente se debe registrar el servidor de la siguiente manera, teniendo en cuenta que se debe conocer la IP remota, el puerto, nombre de la base de datos, usuario y contraseña, esos datos ya se los configuró anteriormente cuando se creó la base de datos.



The screenshot shows the 'Register - Server' dialog box in pgAdmin 4, with the 'Connection' tab selected. The fields are filled with the following values:

- Host name/address: 34.148.16.7
- Port: 5432
- Maintenance database: chirpstack_as_events
- Username: chirpstack_as_events
- Kerberos authentication:
- Password:
- Save password?:
- Role: (empty)
- Service: (empty)

At the bottom, there are buttons for 'Close', 'Reset', and 'Save'.

Figura 5.96: Configuración de la Conexión Remota en pgAdmin 4.

Como se puede observar en la Fig.5.97 se puede visualizar que la conexión remota fue exitosa.



Figura 5.97: Visualización de la conexión remota en pgAdmin 4.

5.11 Configuración de pgAgent para la automatización de trabajos

Al trabajar con una base de datos se decodifica los valores que se obtiene en la base de datos 'chirpstack_as_events' específicamente de la tabla device_up donde se encuentra la data codificada.

Para poder decodificar los valores del consumo se realiza varios queries y además guardar los valores de los query en tablas hasta poder obtener el valor del consumo específico por día, uno de los problemas con los que se puede encontrar es que para actualizar una tabla se debe ejecutar uno por uno los query para cada tabla, es por ello que se encontró la solución de automatizar estos query para que de tal forma las tablas que hacen todo el proceso de decodificación se actualicen.

pgAgent es un agente de programación de trabajos disponible para PostgreSQL que permite la ejecución de procedimientos almacenados, declaraciones SQL y scripts de shell. Su configuración se almacena en la base de datos de postgres en el clúster.

5.11.1 Instalación y configuración del servicio pgAgent

Como se puede observar en la Fig.5.98 se instala el paquete de pgAgent con la siguiente línea de comando:

```
root@chirpstack:/home/bryanca272# sudo apt-get install pgagent
Reading package lists... Done
Building dependency tree
Reading state information... Done
pgagent is already the newest version (3.4.1-5build1).
The following packages were automatically installed and are no longer required:
 libjemalloc1 libnuma1 linux-gcp-5.4-headers-5.4.0-1073 linux-gcp-5.4-headers-5.4.0-1075
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@chirpstack:/home/bryanca272#
```

Figura 5.98: Instalación del paquete pgAgent en la consola del servidor.

Una vez que se accede a la base de datos instalada se crea la extensión "pgagent", con la siguiente línea de comando: 'CREATE EXTENSION pgagent;' como se puede observar en la Fig.5.99.

```
root@chirpstack:/home/bryanca272# su postgres
postgres@chirpstack:/home/bryanca272$ psql
psql (10.21 (Ubuntu 10.21-0ubuntu0.18.04.1))
Type "help" for help.

postgres=# CREATE EXTENSION pgagent; █
```

Figura 5.99: Creación de la extensión pgAgent.

Para crear el lenguaje de procedimiento pl/pgsql se ejecuta la siguiente línea de comando: 'CREATE LANGUAGE plpgsql;' como se puede observar en la Fig.5.100.


```
root@chirpstack:/home/bryanca272# su postgres
postgres@chirpstack:/home/bryanca272$ psql
psql (10.21 (Ubuntu 10.21-0ubuntu0.18.04.1))
Type "help" for help.

postgres=# CREATE LANGUAGE plpgsql;
```

Figura 5.100: Creación del lenguaje de procedimiento pl/pgsql.

Como se puede observar en la Fig.5.101 la siguiente línea de comando ejecutará pgAgent contra un servidor que escucha en el host local, usando una base de datos llamada 'chirpstack_as_events', posteriormente se autenticará con el usuario 'chirpstack_as_events'.

```
postgres@chirpstack:/home/bryanca272$ pgagent hostaddr=localhost dbname=chirpstack_as_events user=chirpstack_as_events
```

Figura 5.101: Ejecución del pgAgent.

Como se puede observar en la Fig.5.102 se encuentra instalado correctamente el pgAgent.

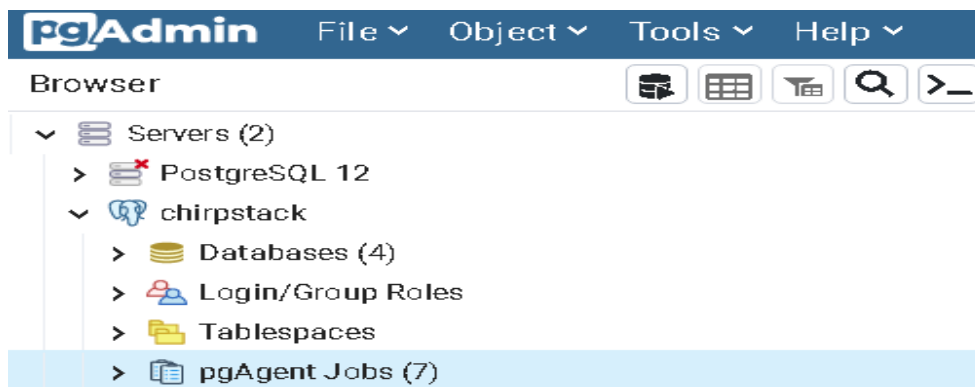


Figura 5.102: Visualización del pgAgent instalado en pgAdmin 4.

5.11.2 Creación de Jobs en pgAgent para automatizar la base de datos 'chirpstack_as_events'

Se visualiza la lista de Jobs creados correctamente en pgAgent que permitirá automatizar la base de datos 'chirpstack_as_events' como se puede observar en la Fig.5.103.

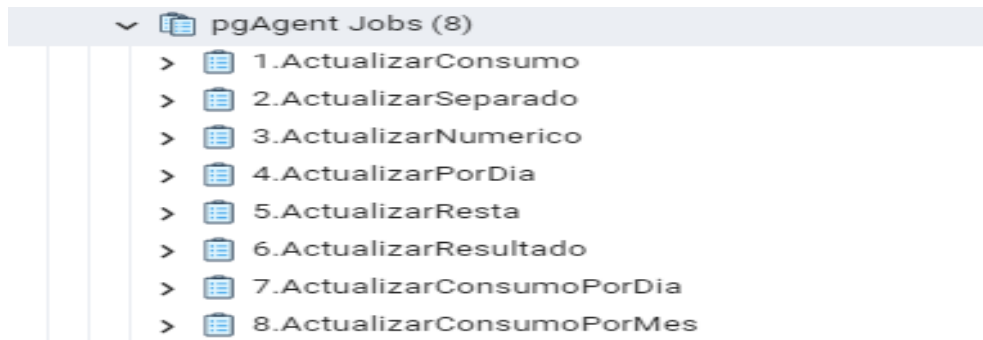


Figura 5.103: Lista de Jobs creados en pgAgent.

El primer pgAgent Job creado es 'ActualizarConsumo', posteriormente se selecciona la pestaña 'General' y se coloca un nombre como se puede observar en la Fig.5.104.

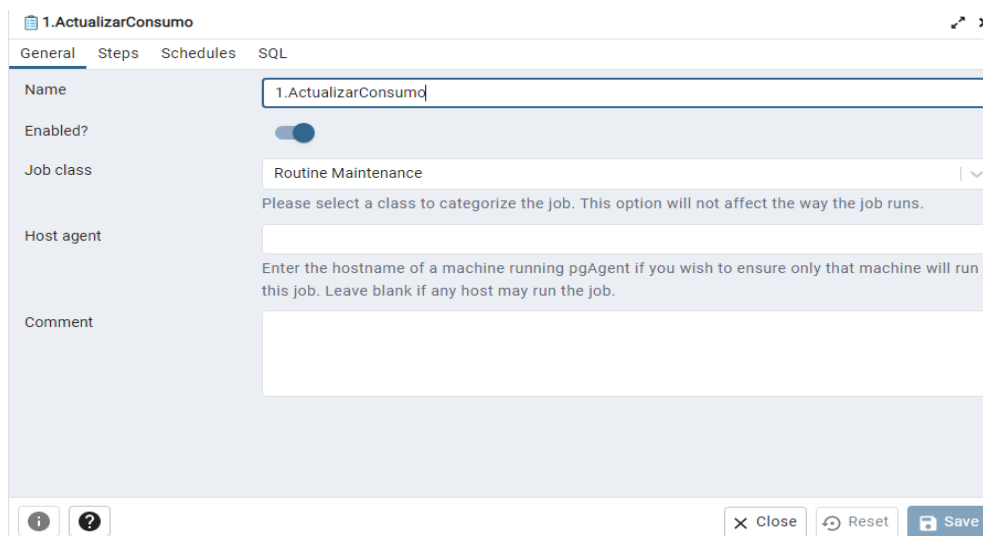


Figura 5.104: Configuración de la primera pestaña 'General'.

En la segunda pestaña 'Steps' se selecciona la subpestaña 'General' y se procede a realizar los siguientes pasos como se puede observar en la Fig.5.105.

- Colocar un nombre.
- Seleccionar la opción de Activado
- Elegir el tipo que se utilizará ya sea SQL o Batch.
- Elegir el tipo de conexión.
- Seleccionar la base de datos en la que correrá el trabajo programado.

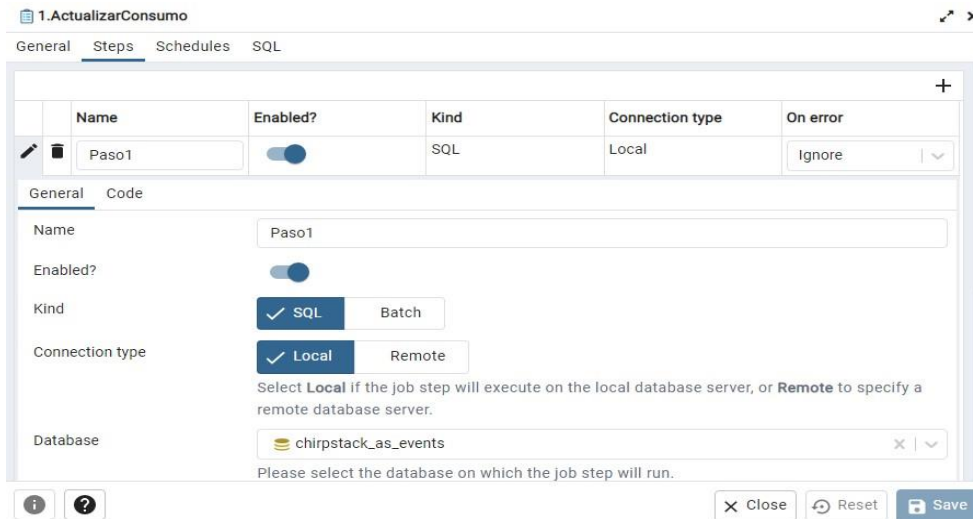


Figura 5.105: Configuración de la segunda pestaña 'Steps'.

En la segunda pestaña 'Steps' se selecciona la subpestaña 'Code' y posteriormente se coloca el código a ejecutar como se puede observar en la Fig.5.106.

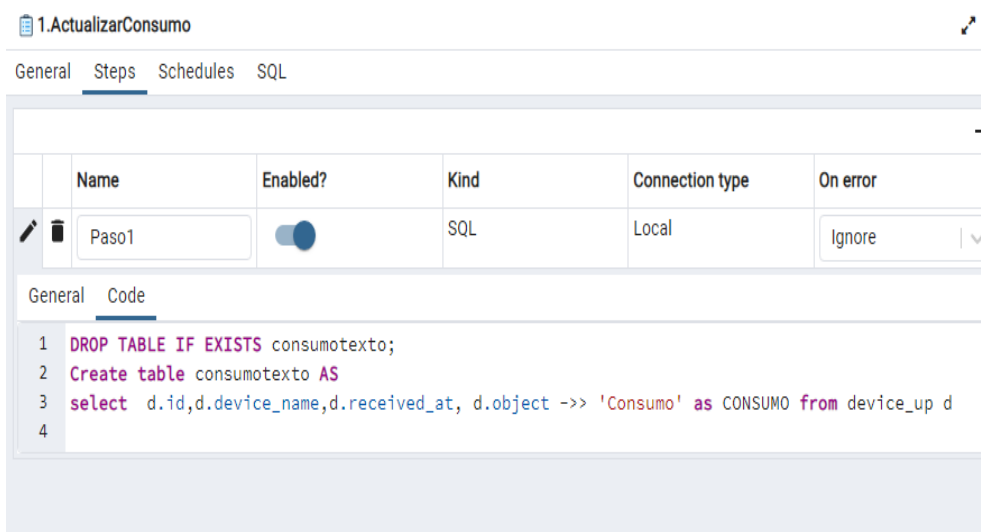


Figura 5.106: Digitación del código a ejecutar.

En la tercera pestaña 'Schedules' se selecciona la subpestaña 'General' y se procede a realizar los siguientes pasos como se puede observar en la Fig.5.107.

- Colocar un nombre.
- Seleccionar la opción de Activado.
- Seleccionar la hora de inicio del trabajo programado.
- Seleccionar la hora de fin del trabajo programado.

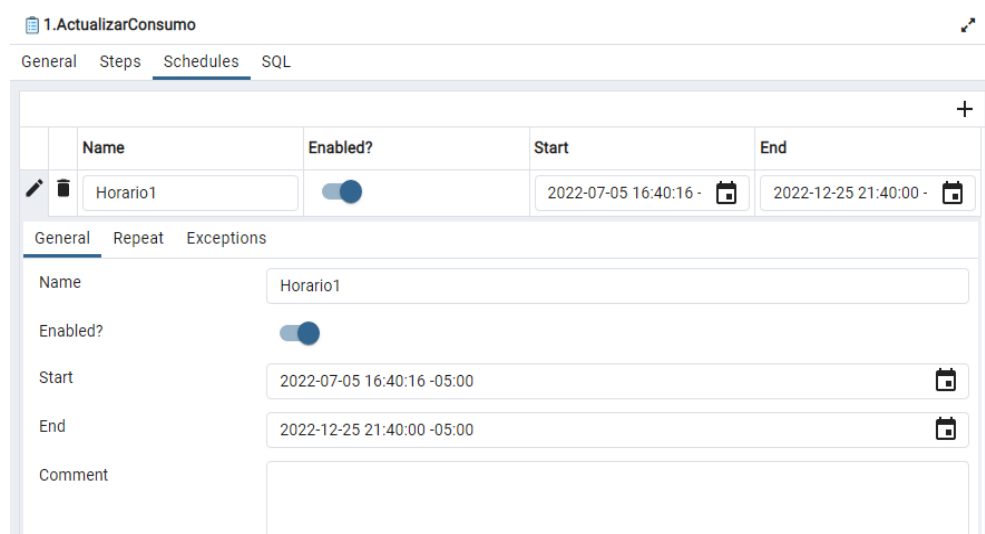


Figura 5.107: Configuración de la tercera pestaña 'Schedules'.

En la tercera pestaña 'Schedules' se selecciona la subpestaña 'Repeat' y posteriormente se procede a realizar los siguientes pasos como se puede observar en la Fig.5.108.

- Seleccionar los días de la semana en la que se repetirá el trabajo programado.
- Seleccionar los días del mes en el que se repetirá el trabajo programado.
- Seleccionar los meses en el que se repetirá el trabajo programado.
- Seleccionar la hora en el que se repetirá el trabajo programado.
- Seleccionar los minutos en los que se repetirá el trabajo programado.

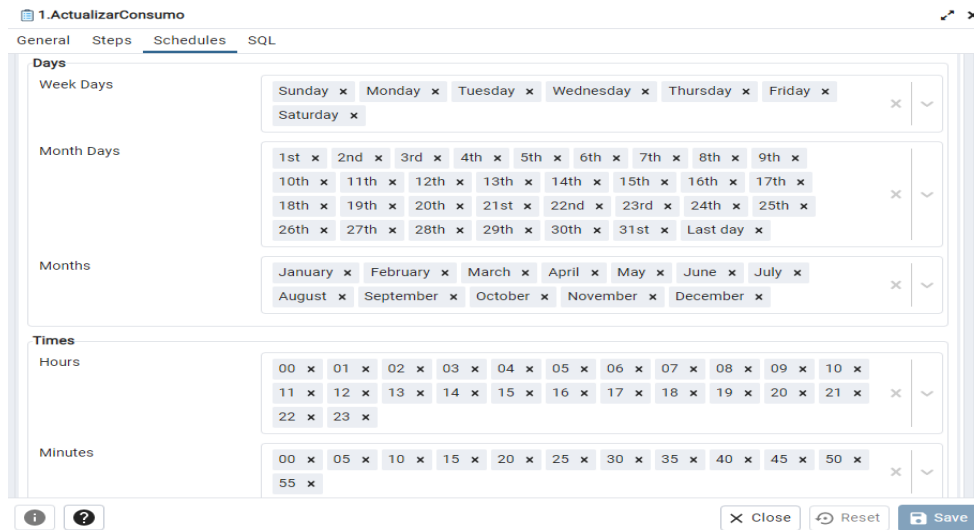


Figura 5.108: Configuración del horario en el que se repetirá el trabajo programado.

Como se puede observar en la Fig.5.109 se visualiza las diferentes propiedades del trabajo programado 'ActualizarConsumo' creado anteriormente.

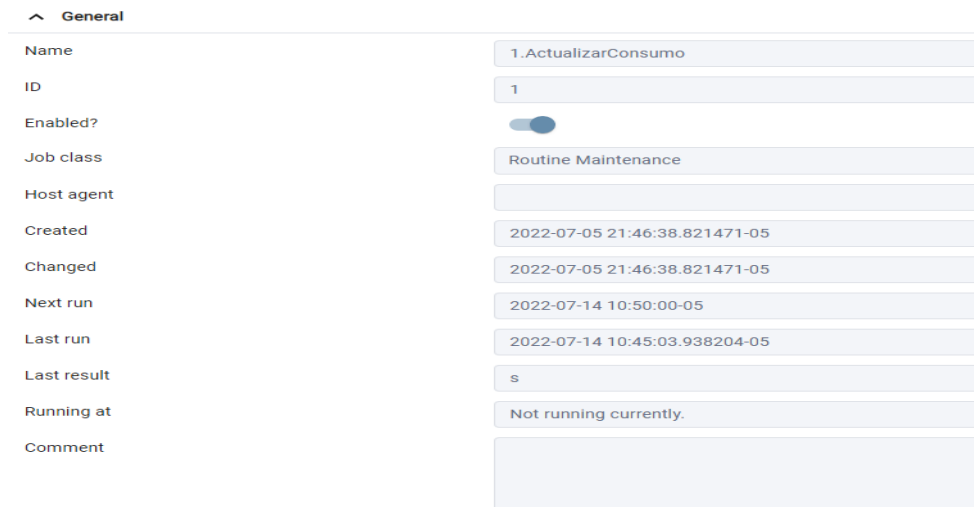


Figura 5.109: Visualización de las propiedades de 'ActualizarConsumo'.

El segundo pgAgent Job creado es 'ActualizarSeparado', posteriormente se selecciona la pestaña 'General' y se coloca un nombre como se puede observar en la Fig.5.110.

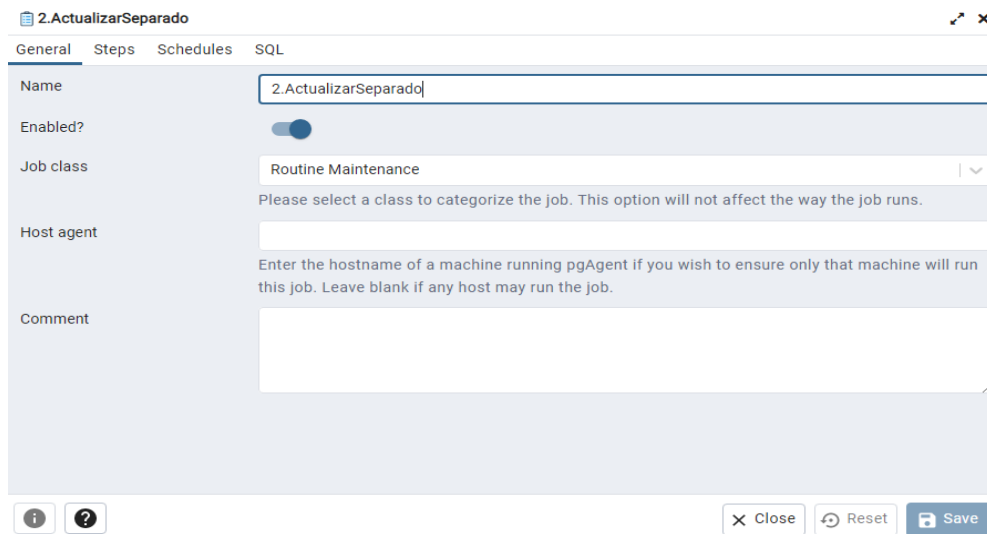


Figura 5.110: Configuración de la primera pestaña 'General'.

En la segunda pestaña 'Steps' se selecciona la subpestaña 'General' y se procede a realizar los siguientes pasos como se puede observar en la Fig.5.111.

- Colocar un nombre.
- Seleccionar la opción de Activado.
- Elegir el tipo que se utilizará ya sea SQL o Batch.
- Elegir el tipo de conexión.
- Seleccionar la base de datos en la que correrá el trabajo programado.

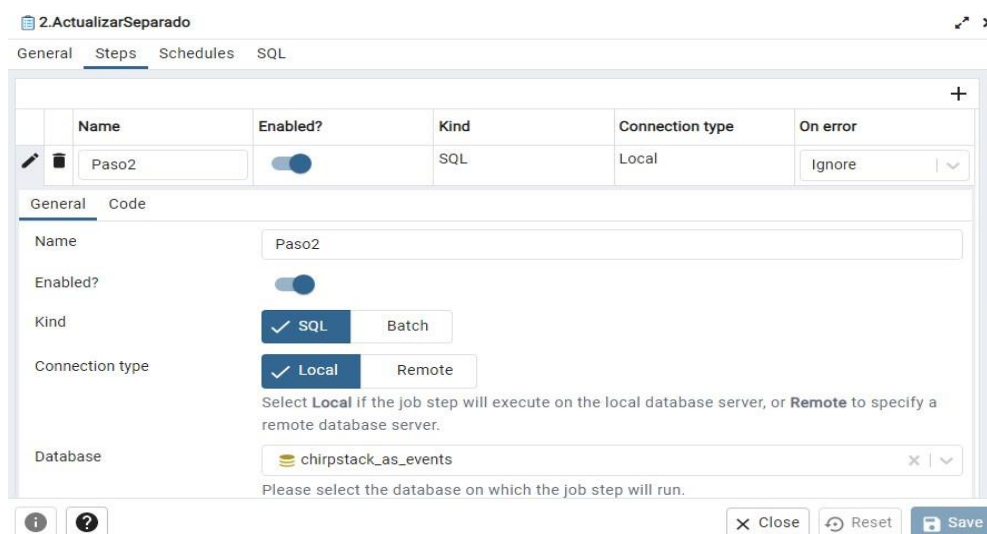


Figura 5.111: Configuración de la segunda pestaña 'Steps'.

En la segunda pestaña 'Steps' se selecciona la subpestaña 'Code' y posteriormente se coloca el código a ejecutar como se puede observar en la Fig.5.112.



Figura 5.112: Digitación del código a ejecutar.

Como se puede observar en la Fig.5.113 se visualiza las diferentes propiedades del trabajo programado 'ActualizarSeparado' creado anteriormente.

- Colocar un nombre.
- Seleccionar la opción de Activado.
- Seleccionar la hora de inicio del trabajo programado.
- Seleccionar la hora de fin del trabajo programado.

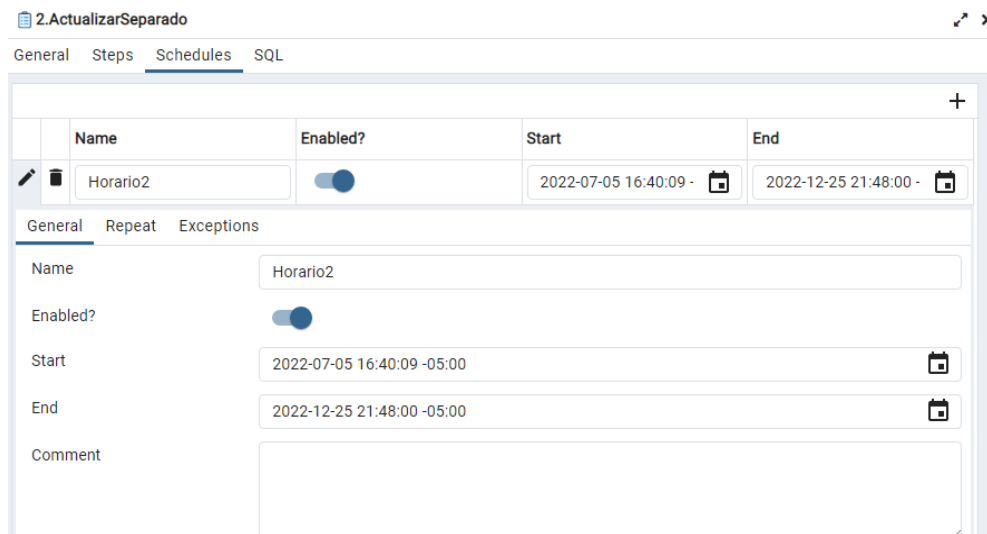


Figura 5.113: Configuración de la tercera pestaña 'Schedules'.

En la tercera pestaña 'Schedules' se selecciona la subpestaña 'Repeat' y posteriormente se procede a realizar los siguientes pasos como se puede observar en la Fig.5.114.

- Seleccionar los días de la semana en la que se repetirá el trabajo programado.
- Seleccionar los días del mes en el que se repetirá el trabajo programado.
- Seleccionar los meses en el que se repetirá el trabajo programado.
- Seleccionar la hora en el que se repetirá el trabajo programado.
- Seleccionar los minutos en los que se repetirá el trabajo programado.

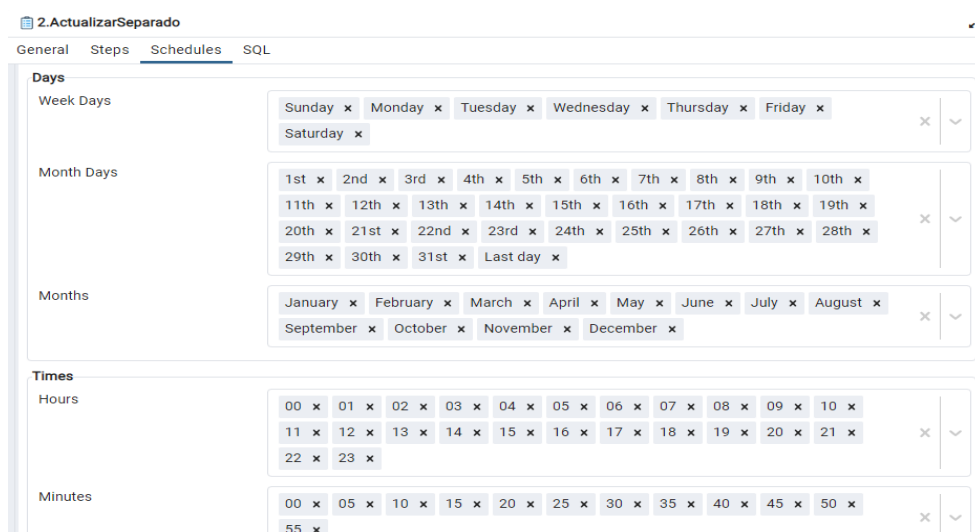


Figura 5.114: Configuración del horario en el que se repetirá el trabajo programado.

Como se puede observar en la Fig.5.115 se visualiza las diferentes propiedades del trabajo programado 'ActualizarSeparado' creado anteriormente.

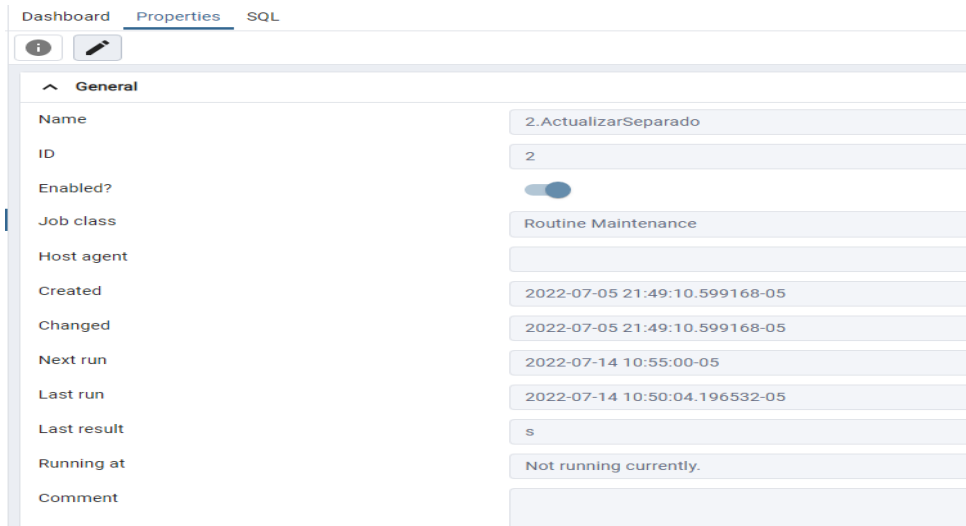


Figura 5.115: Visualización de las propiedades de 'ActualizarSeparado'.

El tercer pgAgent Job creado es 'ActualizarNumerico', posteriormente se selecciona la pestaña 'General' y se coloca un nombre como se puede observar en la Fig.5.116.

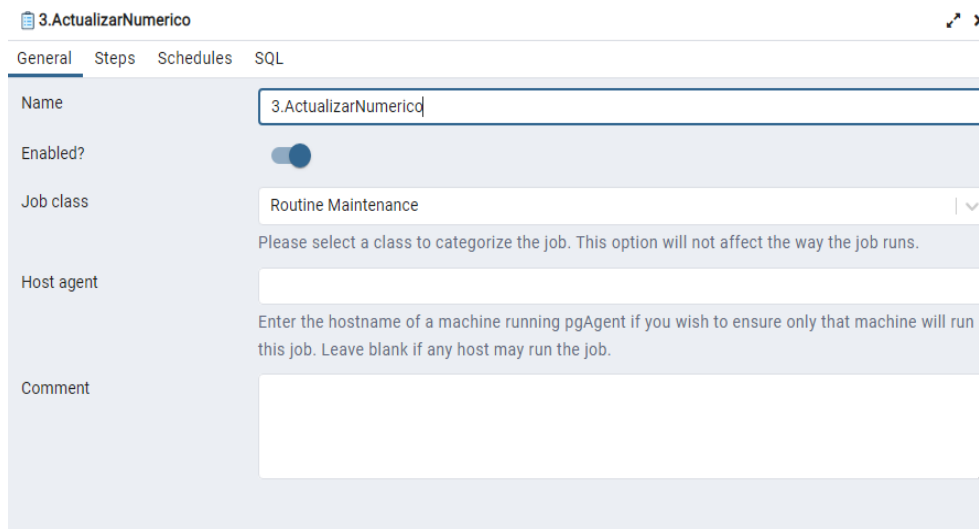


Figura 5.116: Configuración de la primera pestaña 'General'.

En la segunda pestaña 'Steps' se selecciona la subpestaña 'General' y se procede a realizar los siguientes pasos como se puede observar en la Fig.5.117.

- Colocar un nombre.
- Seleccionar la opción de Activado.
- Elegir el tipo que se utilizará ya sea SQL o Batch.
- Elegir el tipo de conexión.
- Seleccionar la base de datos en la que correrá el trabajo programado.

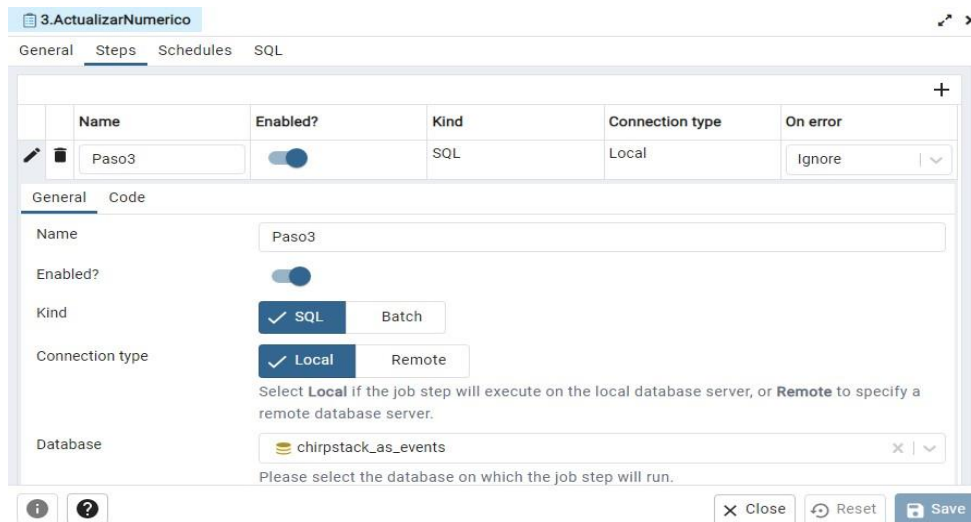


Figura 5.117: Configuración de la segunda pestaña 'Steps'.

En la segunda pestaña 'Steps' se selecciona la subpestaña 'Code' y posteriormente se coloca el código a ejecutar como se puede observar en la Fig.5.118.



Figura 5.118: Digitación del código a ejecutar.

En la tercera pestaña 'Schedules' se selecciona la subpestaña 'General' y se procede a realizar los siguientes pasos como se puede observar en la Fig.5.119.

- Colocar un nombre.
- Seleccionar la opción de Activado.
- Seleccionar la hora de inicio del trabajo programado.
- Seleccionar la hora de fin del trabajo programado.

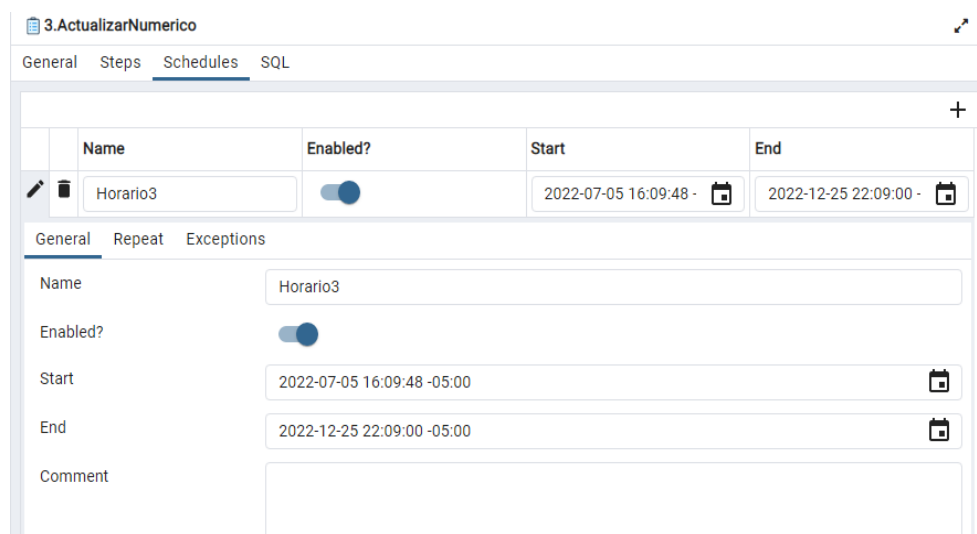


Figura 5.119: Configuración de la tercera pestaña 'Schedules'.

En la tercera pestaña 'Schedules' se selecciona la subpestaña 'Repeat' y posteriormente se procede a realizar los siguientes pasos como se puede observar en la Fig.5.120.

- Seleccionar los días de la semana en la que se repetirá el trabajo programado.
- Seleccionar los días del mes en el que se repetirá el trabajo programado.
- Seleccionar los meses en el que se repetirá el trabajo programado.
- Seleccionar la hora en el que se repetirá el trabajo programado.
- Seleccionar los minutos en los que se repetirá el trabajo programado.

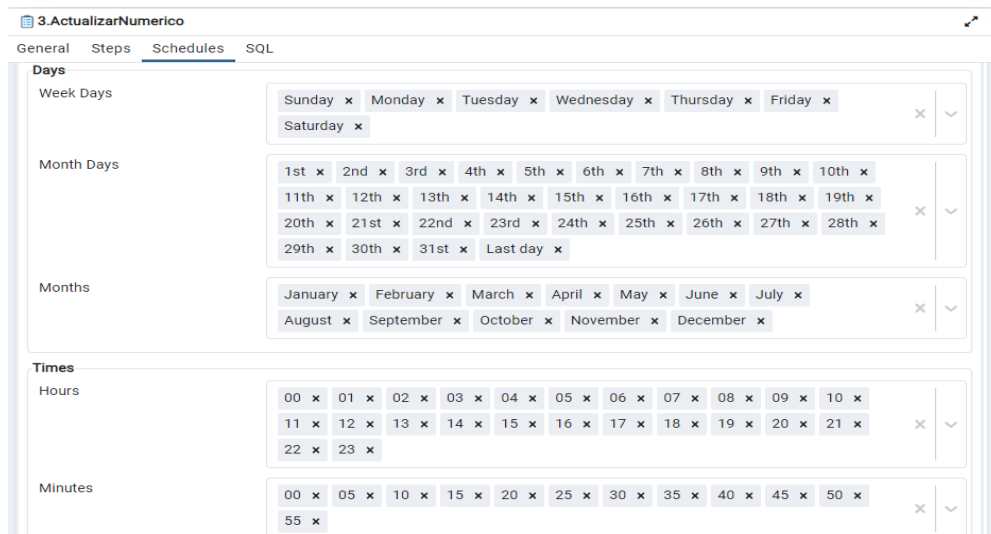


Figura 5.120: Configuración del horario en el que se repetirá el trabajo programado.

Como se puede observar en la Fig.5.121 se visualiza las diferentes propiedades del trabajo programado 'ActualizarNumerico' creado anteriormente.

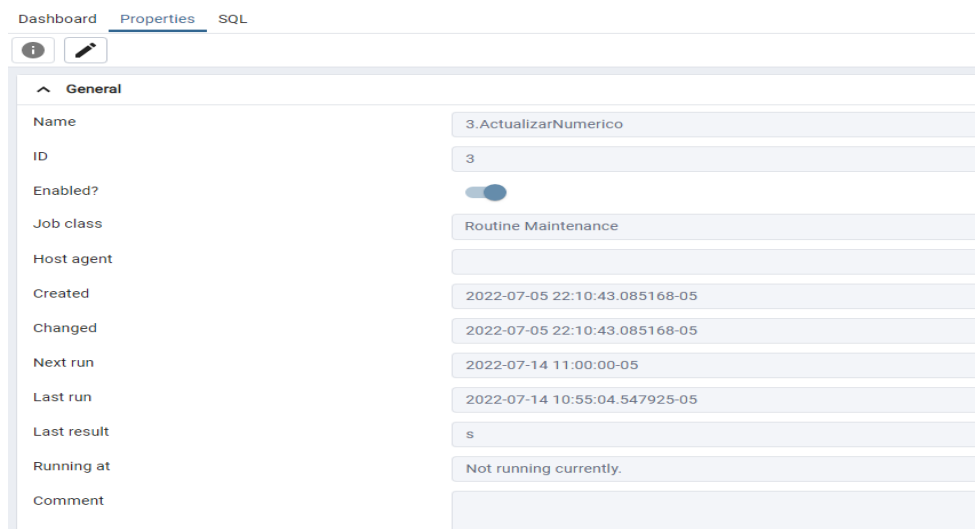


Figura 5.121: Visualización de las propiedades de 'ActualizarNumerico'.

El cuarto pgAgent Job creado es 'ActualizarPorDia', posteriormente se selecciona la pestaña 'General' y se coloca un nombre como se puede observar en la Fig.5.122.

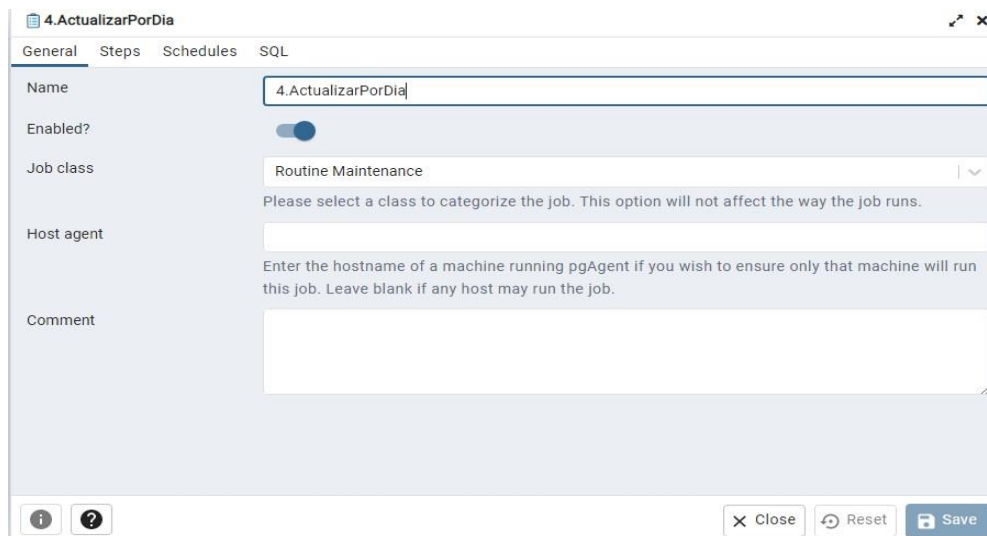


Figura 5.122: Configuración de la primera pestaña 'General'.

En la segunda pestaña 'Steps' se selecciona la subpestaña 'General' y se procede a realizar los siguientes pasos como se puede observar en la Fig.5.123.

- Colocar un nombre.
- Seleccionar la opción de Activado.
- Elegir el tipo que se utilizará ya sea SQL o Batch.
- Elegir el tipo de conexión.
- Seleccionar la base de datos en la que correrá el trabajo programado.

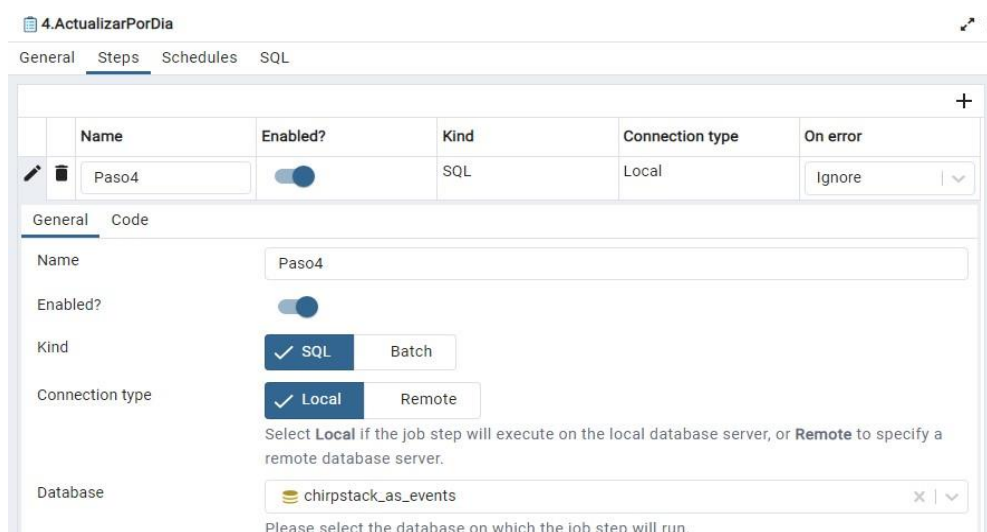


Figura 5.123: Configuración de la segunda pestaña 'Steps'.

En la segunda pestaña 'Steps' se selecciona la subpestaña 'Code' y posteriormente se coloca el código a ejecutar como se puede observar en la Fig.5.124.

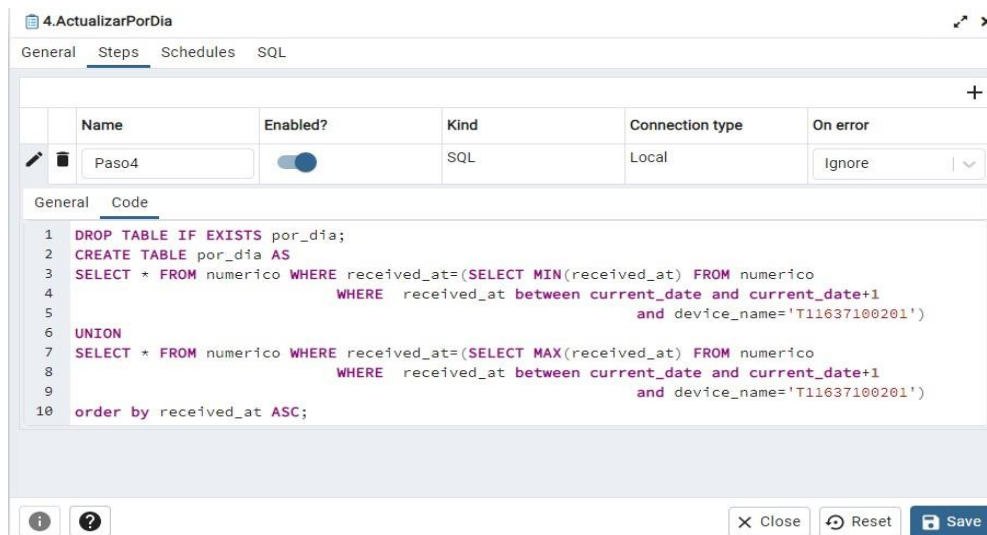


Figura 5.124: Digitación del código a ejecutar.

En la tercera pestaña 'Schedules' se selecciona la subpestaña 'General' y se procede a realizar los siguientes pasos como se puede observar en la Fig.5.125.

- Colocar un nombre.
- Seleccionar la opción de Activado.
- Seleccionar la hora de inicio del trabajo programado.
- Seleccionar la hora de fin del trabajo programado.

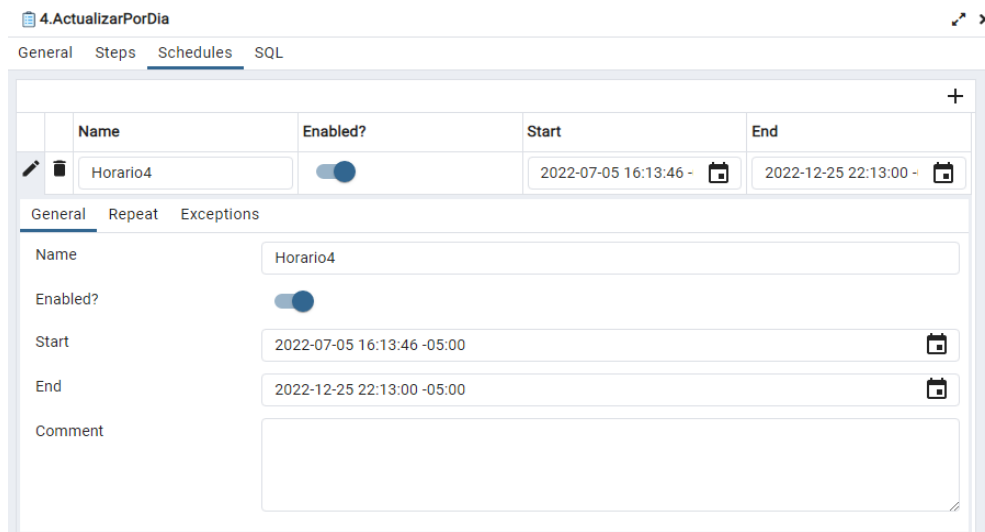


Figura 5.125: Configuración de la tercera pestaña 'Schedules'.

En la tercera pestaña 'Schedules' se selecciona la subpestaña 'Repeat' y posteriormente se procede a realizar los siguientes pasos como se puede observar en la Fig.5.126.

- Seleccionar los días de la semana en la que se repetirá el trabajo programado.
- Seleccionar los días del mes en el que se repetirá el trabajo programado.
- Seleccionar los meses en el que se repetirá el trabajo programado.
- Seleccionar la hora en el que se repetirá el trabajo programado.
- Seleccionar los minutos en los que se repetirá el trabajo programado.

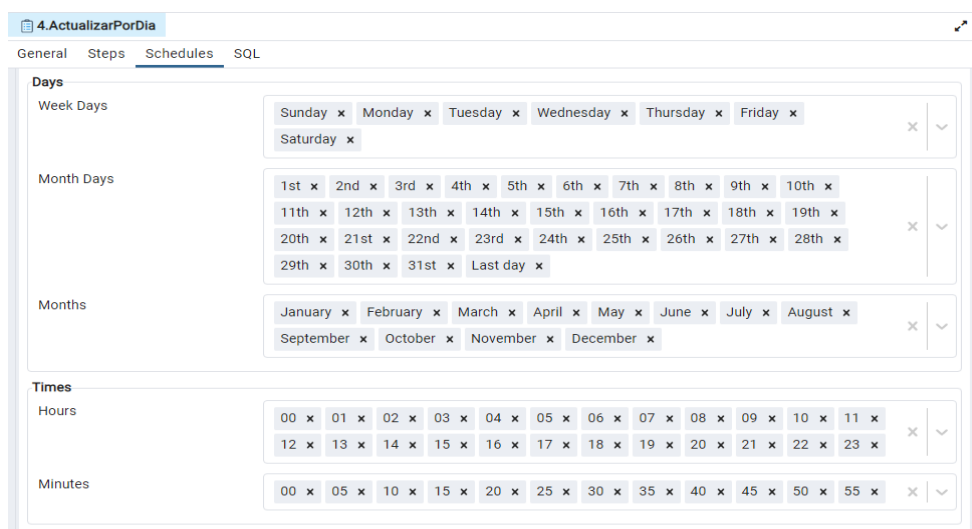


Figura 5.126: Configuración del horario en el que se repetirá el trabajo programado.

Como se puede observar en la Fig.5.127 se visualiza las diferentes propiedades del trabajo programado 'ActualizarPorDia' creado anteriormente.

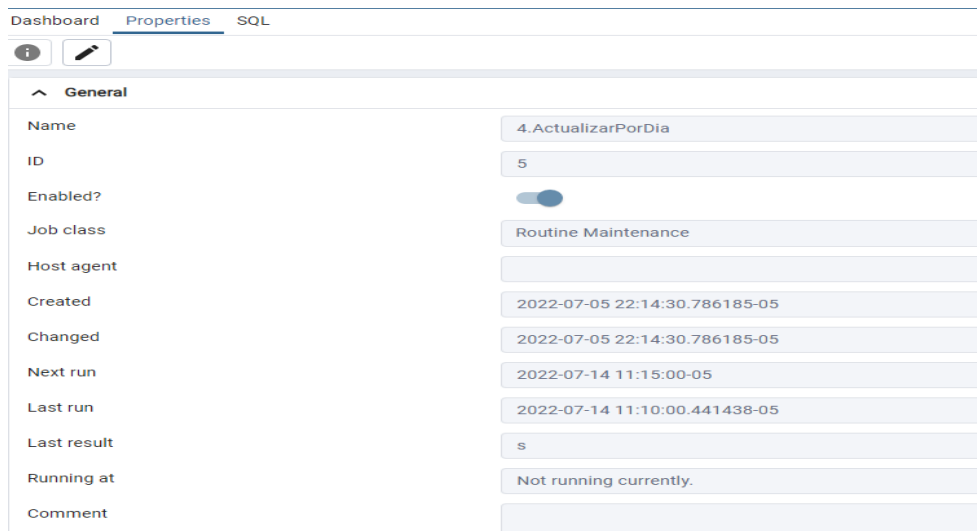


Figura 5.127: Visualización de las propiedades de 'ActualizarPorDia'.

El quinto pgAgent Job creado es 'ActualizarResta', posteriormente se selecciona la pestaña 'General' y se coloca un nombre como se puede observar en la Fig.5.128.

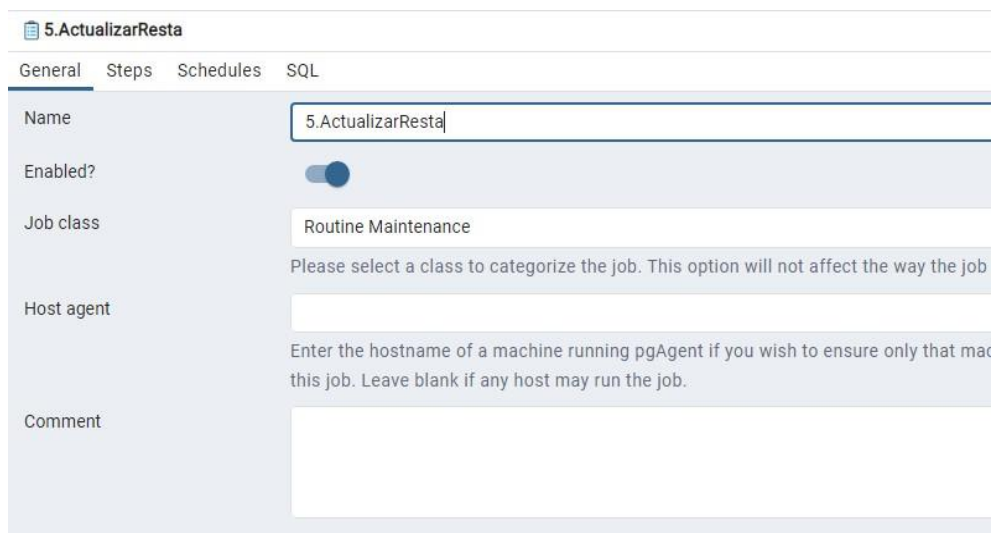


Figura 5.128: Configuración de la primera pestaña 'General'.

En la segunda pestaña 'Steps' se selecciona la subpestaña 'General' y se procede a realizar los siguientes pasos como se puede observar en la Fig.5.129.

- Colocar un nombre.
- Seleccionar la opción de Activado.
- Elegir el tipo que se utilizará ya sea SQL o Batch.
- Elegir el tipo de conexión.
- Seleccionar la base de datos en la que correrá el trabajo programado.

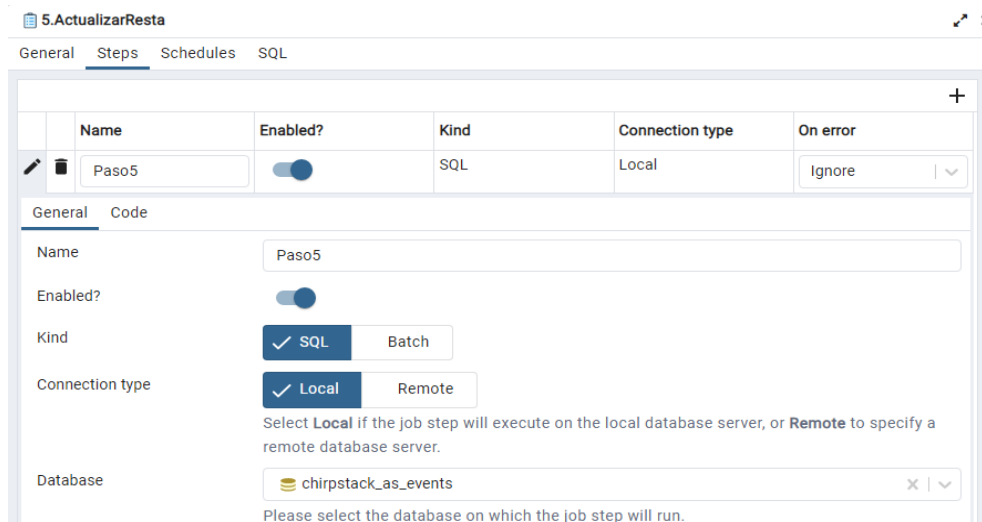


Figura 5.129: Configuración de la segunda pestaña 'Steps'.

En la segunda pestaña 'Steps' se selecciona la subpestaña 'Code' y posteriormente se coloca el código a ejecutar como se puede observar en la Fig.5.130.

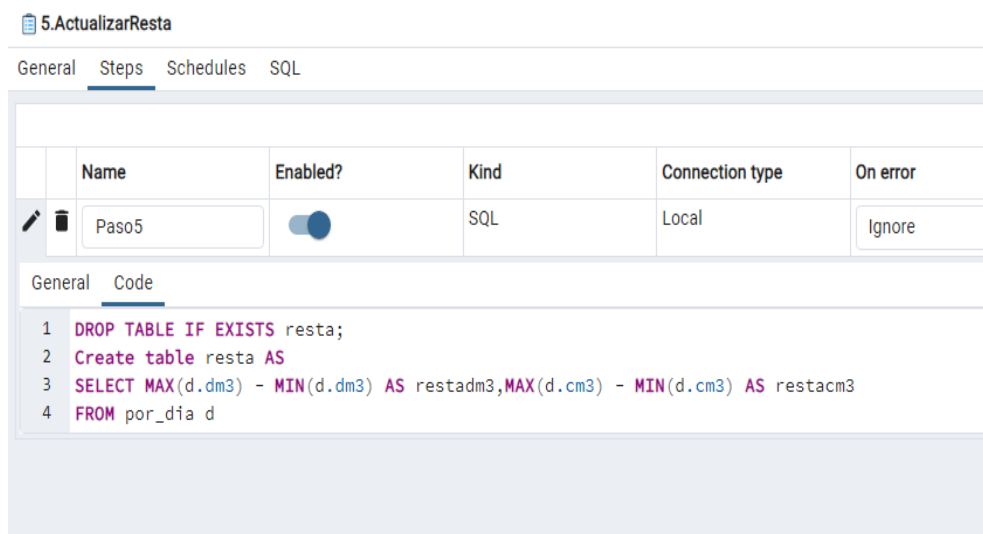


Figura 5.130: Digitación del código a ejecutar.

En la tercera pestaña 'Schedules' se selecciona la subpestaña 'General' y se procede a realizar los siguientes pasos como se puede observar en la Fig.5.131.

- Colocar un nombre.
- Seleccionar la opción de Activado.
- Seleccionar la hora de inicio del trabajo programado.
- Seleccionar la hora de fin del trabajo programado.

The screenshot shows the '5.ActualizarResta' application interface. At the top, there are tabs for 'General', 'Steps', 'Schedules', and 'SQL'. The 'Schedules' tab is selected. Below the tabs, there is a table with columns: Name, Enabled?, Start, and End. The table contains one row with the following data:

Name	Enabled?	Start	End
Horario5	<input checked="" type="checkbox"/>	2022-07-05 16:16:31	2022-12-25 22:16:00

Below the table, there are sub-tabs for 'General', 'Repeat', and 'Exceptions'. The 'General' sub-tab is active, showing the following configuration details:

- Name: Horario5
- Enabled?:
- Start: 2022-07-05 16:16:31 -05:00
- End: 2022-12-25 22:16:00 -05:00
- Comment: (empty text area)

Figura 5.131: Configuración de la tercera pestaña 'Schedules'.

En la tercera pestaña 'Schedules' se selecciona la subpestaña 'Repeat' y posteriormente se procede a realizar los siguientes pasos como se puede observar en la Fig.5.132.

- Seleccionar los días de la semana en la que se repetirá el trabajo programado.
- Seleccionar los días del mes en el que se repetirá el trabajo programado.
- Seleccionar los meses en el que se repetirá el trabajo programado.
- Seleccionar la hora en el que se repetirá el trabajo programado.
- Seleccionar los minutos en los que se repetirá el trabajo programado.

The screenshot shows the configuration interface for a scheduled job named '5.ActualizarResta'. The 'Schedules' tab is active. Under the 'Days' section, 'Week Days' includes Sunday through Friday, and 'Month Days' includes the 1st through 31st and 'Last day'. Under 'Months', all twelve months are selected. The 'Times' section shows 'Hours' from 00 to 23 and 'Minutes' from 00 to 55.

Figura 5.132: Configuración del horario en el que se repetirá el trabajo programado.

Como se puede observar en la Fig.5.133 se visualiza las diferentes propiedades del trabajo programado 'ActualizarResta' creado anteriormente.

The screenshot shows the 'Properties' page for the job '5.ActualizarResta'. The 'General' tab is active. The properties listed are:

Name	5.ActualizarResta
ID	6
Enabled?	<input checked="" type="checkbox"/>
Job class	Routine Maintenance
Host agent	
Created	2022-07-05 22:17:05.813407-05
Changed	2022-07-05 22:17:05.813407-05
Next run	2022-07-14 11:20:00-05
Last run	2022-07-14 11:15:00.715556-05
Last result	s
Running at	Not running currently.
Comment	

Figura 5.133: Visualización de las propiedades de 'ActualizarResta'.

El sexto pgAgent Job creado es 'ActualizarResultado', posteriormente se selecciona la pestaña 'General' y se coloca un nombre como se puede observar en la Fig.5.134.

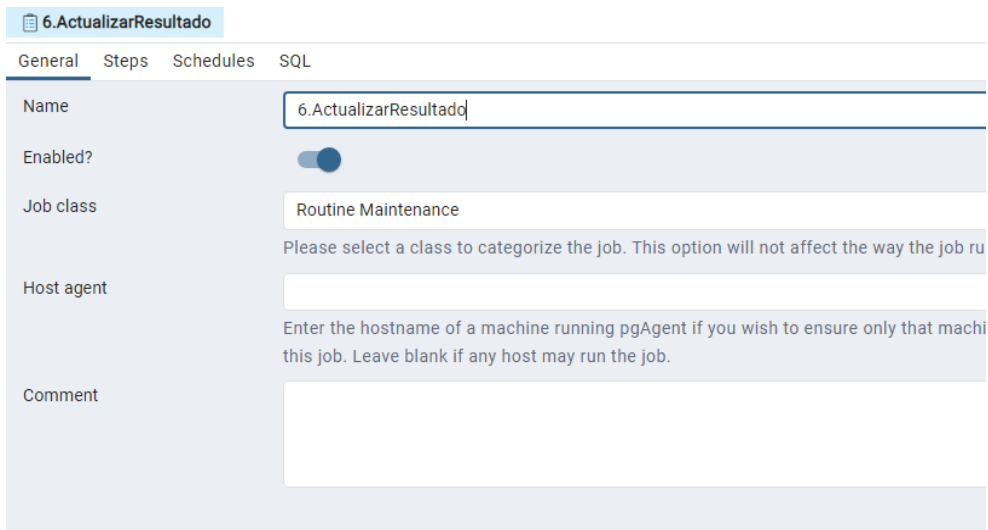


Figura 5.134: Configuración de la primera pestaña 'General'.

En la segunda pestaña 'Steps' se selecciona la subpestaña 'General' y se procede a realizar los siguientes pasos como se puede observar en la Fig.5.135.

- Colocar un nombre.
- Seleccionar la opción de Activado.
- Elegir el tipo que se utilizará ya sea SQL o Batch.
- Elegir el tipo de conexión.
- Seleccionar la base de datos en la que correrá el trabajo programado.

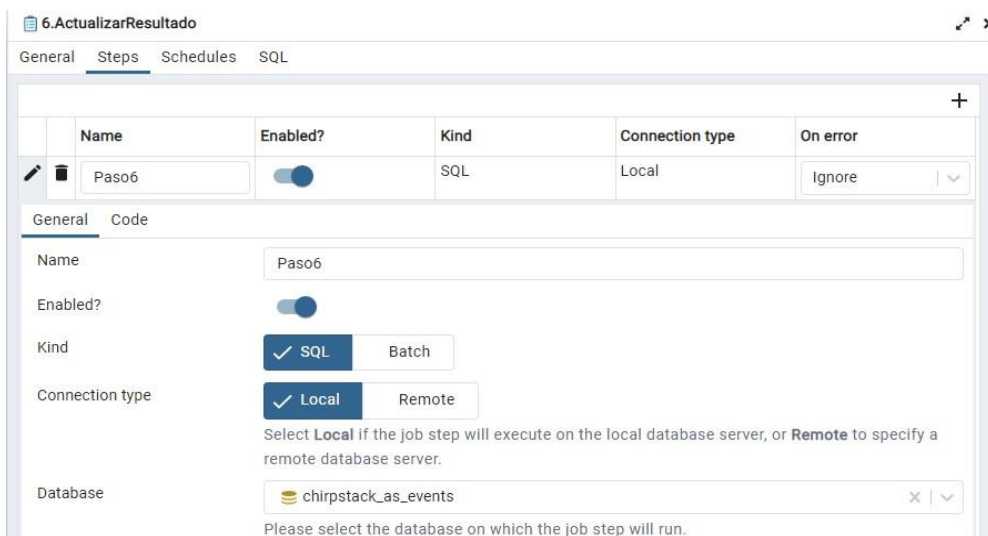


Figura 5.135: Configuración de la segunda pestaña 'Steps'.

En la segunda pestaña 'Steps' se selecciona la subpestaña 'Code' y posteriormente se coloca el código a ejecutar como se puede observar en la Fig.5.136.

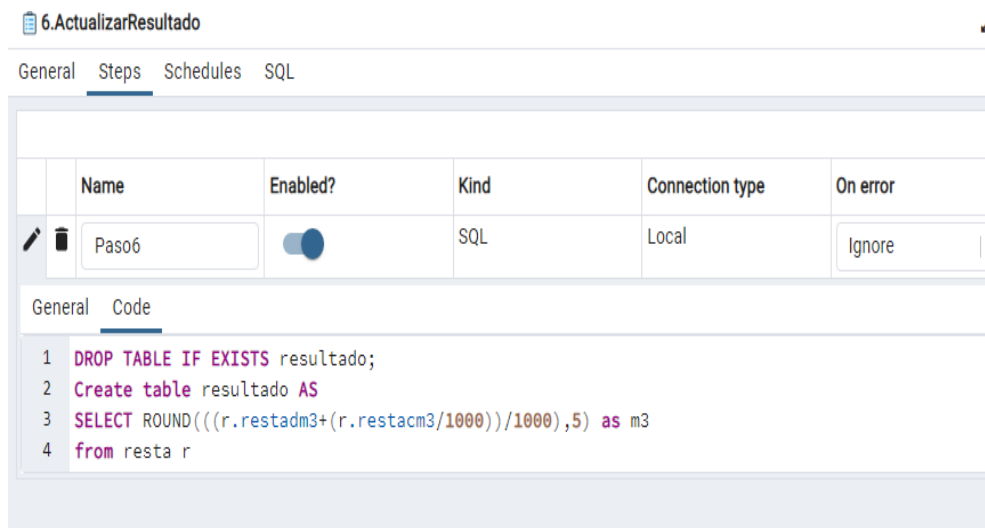


Figura 5.136: Digitación del código a ejecutar.

En la tercera pestaña 'Schedules' se selecciona la subpestaña 'General' y se procede a realizar los siguientes pasos como se puede observar en la Fig.5.137.

- Colocar un nombre.
- Seleccionar la opción de Activado.
- Seleccionar la hora de inicio del trabajo programado.
- Seleccionar la hora de fin del trabajo programado.

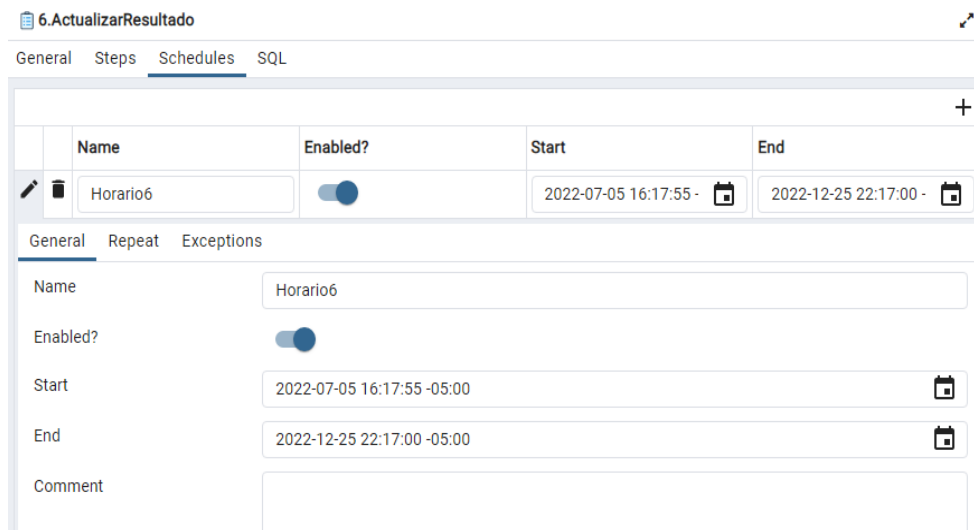


Figura 5.137: Configuración de la tercera pestaña 'Schedules'.

En la tercera pestaña 'Schedules' se selecciona la subpestaña 'Repeat' y posteriormente se procede a realizar los siguientes pasos como se puede observar en la Fig.5.138.

- Seleccionar los días de la semana en la que se repetirá el trabajo programado.
- Seleccionar los días del mes en el que se repetirá el trabajo programado.
- Seleccionar los meses en el que se repetirá el trabajo programado.
- Seleccionar la hora en el que se repetirá el trabajo programado.
- Seleccionar los minutos en los que se repetirá el trabajo programado.

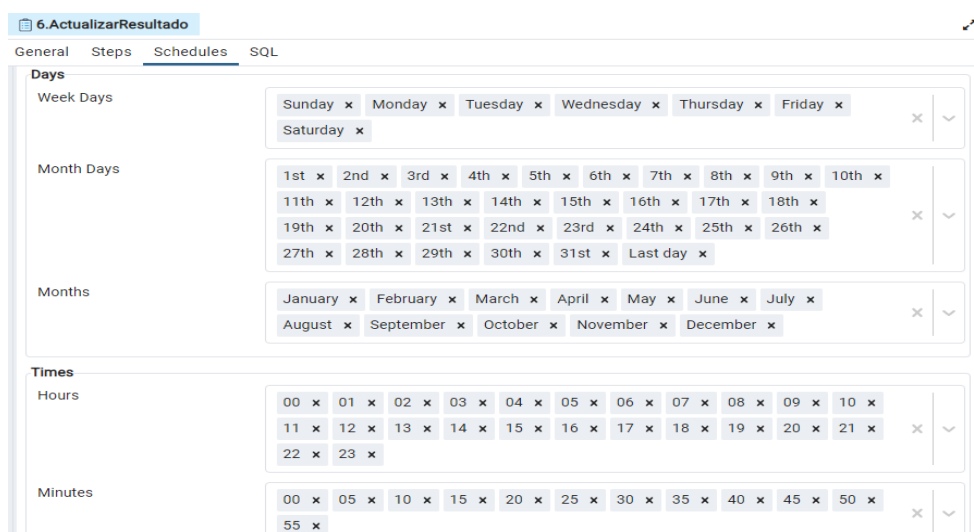


Figura 5.138: Configuración del horario en el que se repetirá el trabajo programado.

Como se puede observar en la Fig.5.139 se visualiza las diferentes propiedades del trabajo programado 'ActualizarResultado' creado anteriormente.

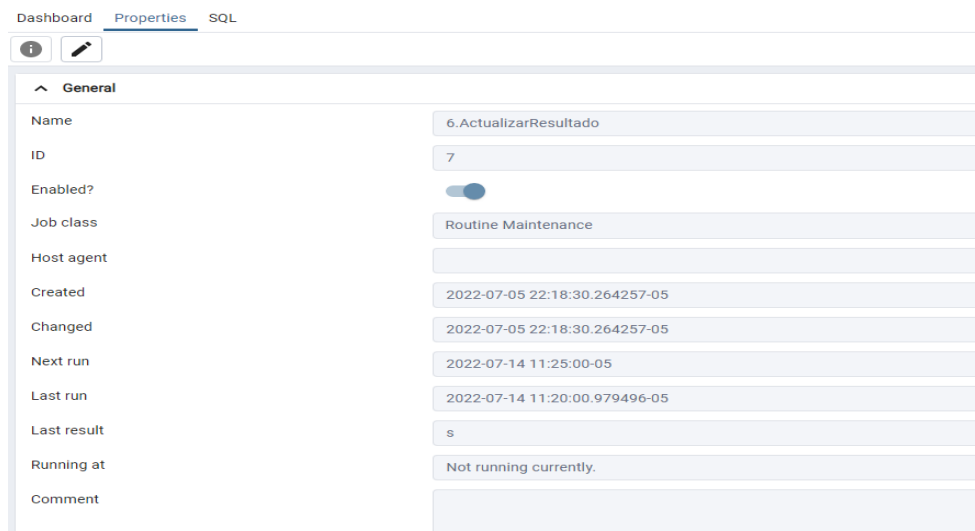


Figura 5.139: Visualización de las propiedades de 'ActualizarResultado'.

El séptimo pgAgent Job creado es 'ActualizarConsumoPorDia', posteriormente se selecciona la pestaña 'General' y se coloca un nombre como se puede observar en la Fig.5.140.

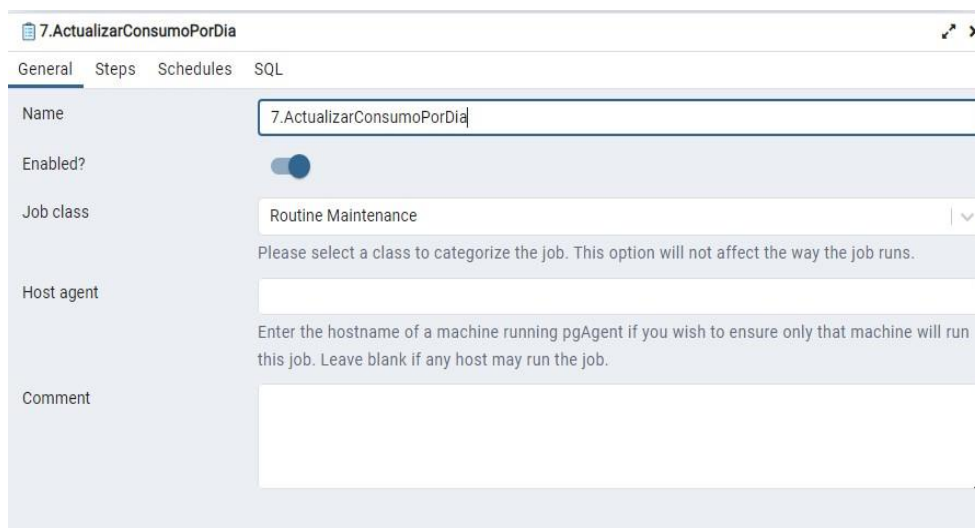


Figura 5.140: Configuración de la primera pestaña 'General'.

En la segunda pestaña 'Steps' se selecciona la subpestaña 'General' y se procede a realizar los siguientes pasos como se puede observar en la Fig.5.141.

- Colocar un nombre.
- Seleccionar la opción de Activado.
- Elegir el tipo que se utilizará ya sea SQL o Batch.
- Elegir el tipo de conexión.
- Seleccionar la base de datos en la que correrá el trabajo programado.

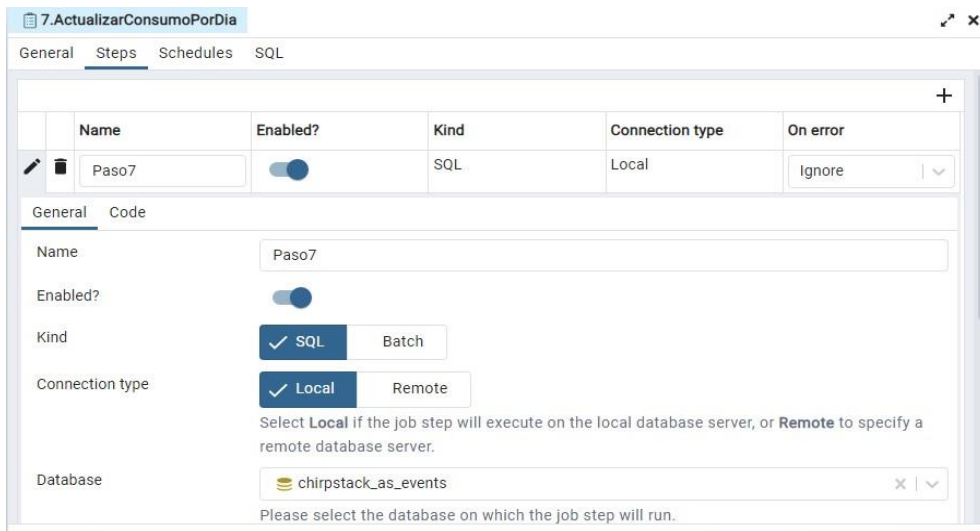


Figura 5.141: Configuración de la segunda pestaña 'Steps'.

En la segunda pestaña 'Steps' se selecciona la subpestaña 'Code' y posteriormente se coloca el código a ejecutar como se puede observar en la Fig.5.142.

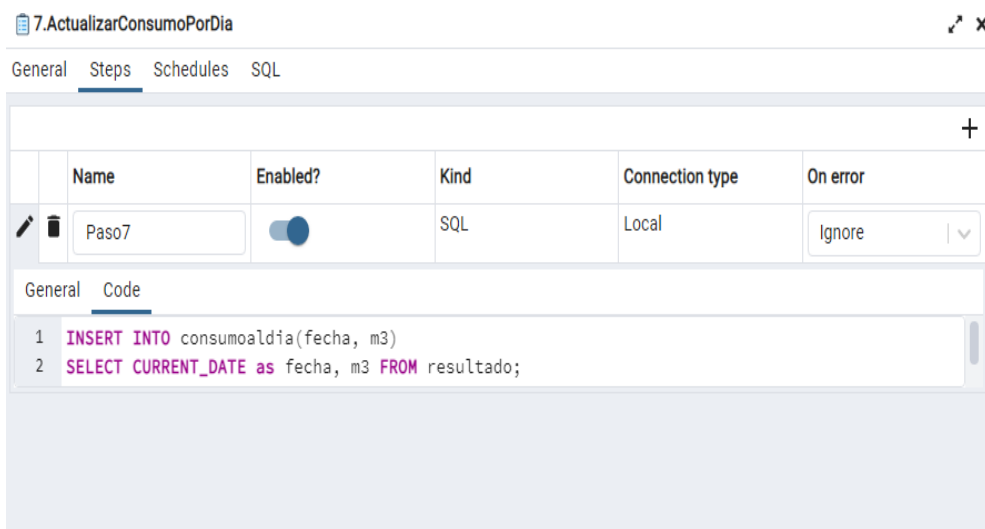


Figura 5.142: Digitación del código a ejecutar.

En la tercera pestaña 'Schedules' se selecciona la subpestaña 'General' y se procede a realizar los siguientes pasos como se puede observar en la Fig.5.143.

- Colocar un nombre.
- Seleccionar la opción de Activado.
- Seleccionar la hora de inicio del trabajo programado.
- Seleccionar la hora de fin del trabajo programado.

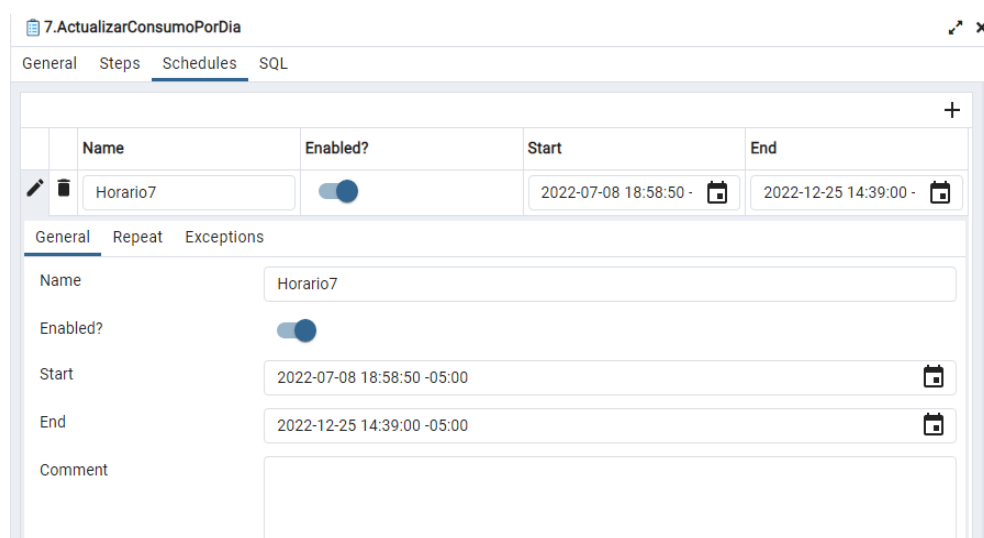


Figura 5.143: Configuración de la tercera pestaña 'Schedules'.

En la tercera pestaña 'Schedules' se selecciona la subpestaña 'Repeat' y posteriormente se procede a realizar los siguientes pasos como se puede observar en la Fig.5.144.

- Seleccionar los días de la semana en la que se repetirá el trabajo programado.
- Seleccionar los días del mes en el que se repetirá el trabajo programado.
- Seleccionar los meses en el que se repetirá el trabajo programado.
- Seleccionar la hora en el que se repetirá el trabajo programado.
- Seleccionar los minutos en los que se repetirá el trabajo programado.

The screenshot shows the configuration for a scheduled job named '7.ActualizarConsumoPorDia'. The 'Schedules' tab is active, displaying the following settings:

- Week Days:** Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday (all selected).
- Month Days:** 1st through 31st, and Last day (all selected).
- Months:** January through December (all selected).
- Times:** Hours: 23, Minutes: 58.

Figura 5.144: Configuración del horario en el que se repetirá el trabajo programado.

Como se puede observar en la Fig.5.145 se visualiza las diferentes propiedades del trabajo programado 'ActualizarConsumoPorDia' creado anteriormente.

The screenshot shows the 'Properties' tab for the job '7.ActualizarConsumoPorDia'. The 'General' section is expanded, showing the following properties:

Name	7.ActualizarConsumoPorDia
ID	8
Enabled?	<input checked="" type="checkbox"/>
Job class	Routine Maintenance
Host agent	
Created	2022-07-08 14:41:03.673402-05
Changed	2022-07-08 14:41:03.673402-05
Next run	2022-07-14 23:58:00-05
Last run	2022-07-13 23:58:04.957703-05
Last result	s
Running at	Not running currently.
Comment	

Figura 5.145: Visualización de las propiedades de 'ActualizarConsumoPorDia'.

El octavo pgAgent Job creado es 'ActualizarConsumoPorMes', posteriormente se selecciona la pestaña 'General' y se coloca un nombre como se puede observar en la Fig.5.146.

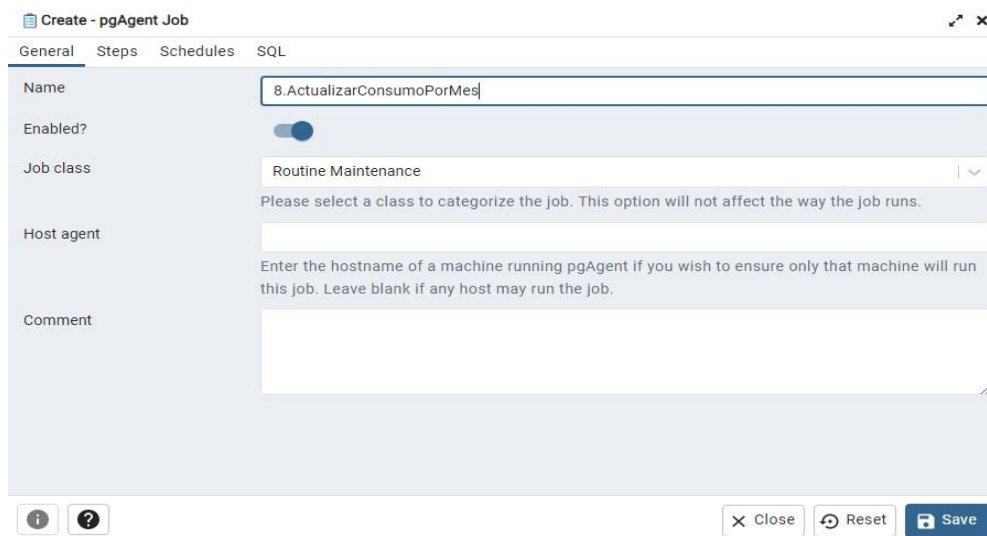


Figura 5.146: Configuración de la primera pestaña 'General'.

En la segunda pestaña 'Steps' se selecciona la subpestaña 'General' y se procede a realizar los siguientes pasos como se puede observar en la Fig.5.147.

- Colocar un nombre.
- Seleccionar la opción de Activado.
- Elegir el tipo que se utilizará ya sea SQL o Batch.
- Elegir el tipo de conexión.
- Seleccionar la base de datos en la que correrá el trabajo programado.

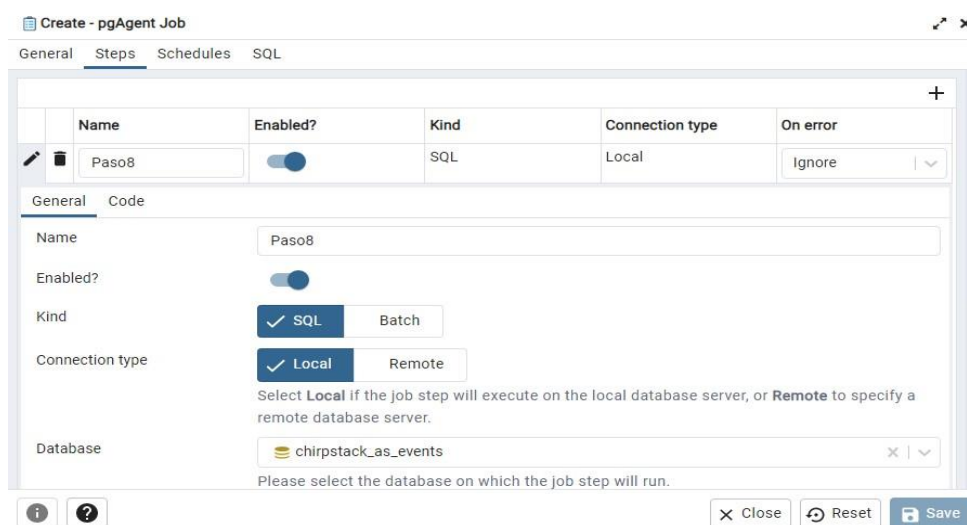


Figura 5.147: Configuración de la segunda pestaña 'Steps'.

En la segunda pestaña 'Steps' se selecciona la subpestaña 'Code' y posteriormente se coloca el código a ejecutar como se puede observar en la Fig.5.148.

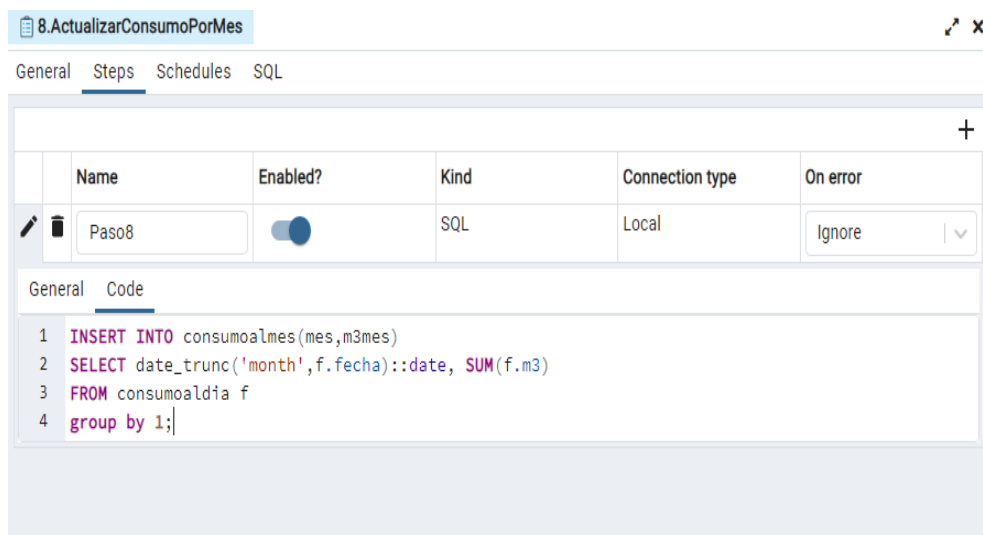


Figura 5.148: Digitación del código a ejecutar.

En la tercera pestaña 'Schedules' se selecciona la subpestaña 'General' y se procede a realizar los siguientes pasos como se puede observar en la Fig.5.149.

- Colocar un nombre.
- Seleccionar la opción de Activado.
- Seleccionar la hora de inicio del trabajo programado.
- Seleccionar la hora de fin del trabajo programado.

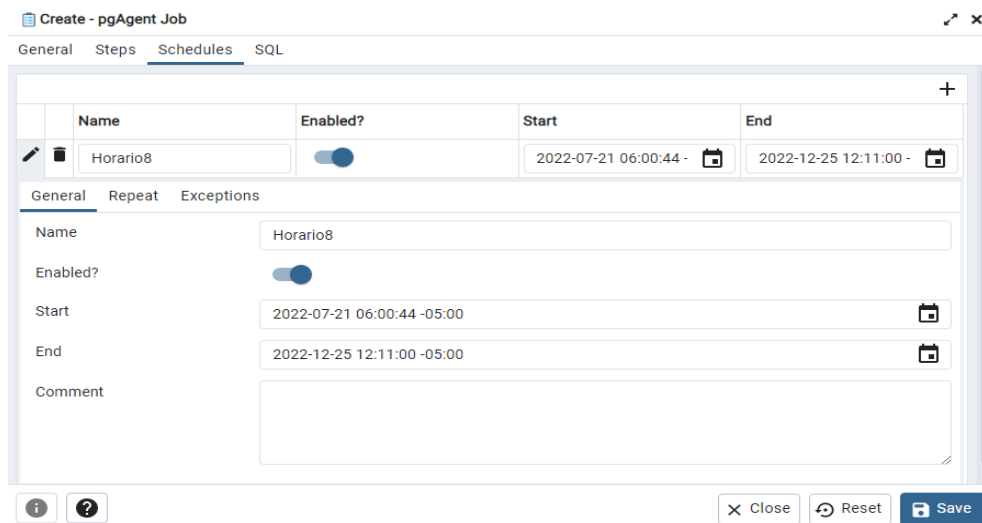


Figura 5.149: Configuración de la tercera pestaña 'Schedules'.

En la tercera pestaña 'Schedules' se selecciona la subpestaña 'Repeat' y posteriormente se procede a realizar los siguientes pasos como se puede observar en la Fig.5.150.

- Seleccionar los días de la semana en la que se repetirá el trabajo programado.
- Seleccionar los días del mes en el que se repetirá el trabajo programado.
- Seleccionar los meses en el que se repetirá el trabajo programado.
- Seleccionar la hora en el que se repetirá el trabajo programado.
- Seleccionar los minutos en los que se repetirá el trabajo programado.

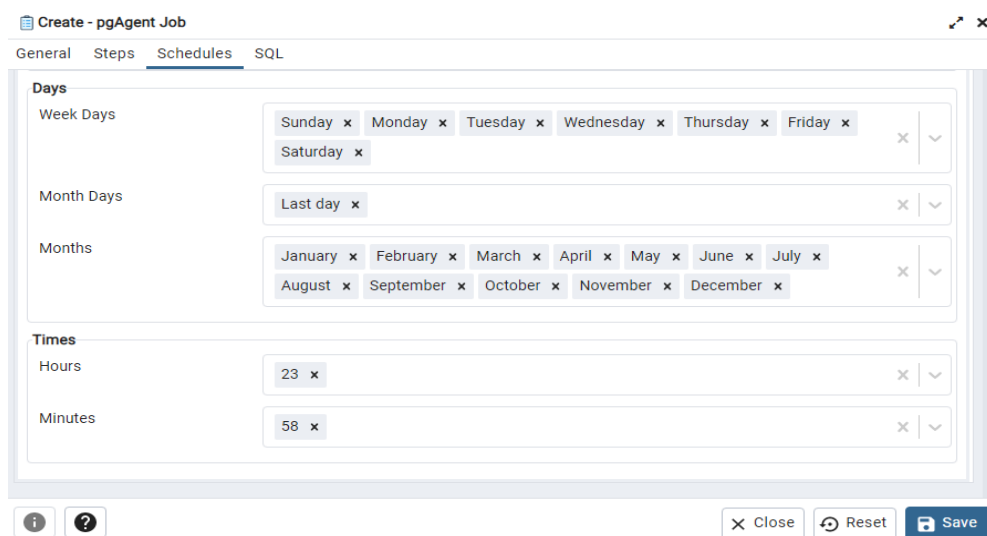


Figura 5.150: Configuración del horario en el que se repetirá el trabajo programado.

Como se puede observar en la Fig.5.151 se visualiza las diferentes propiedades del trabajo programado 'ActualizarConsumoPorMes' creado anteriormente.

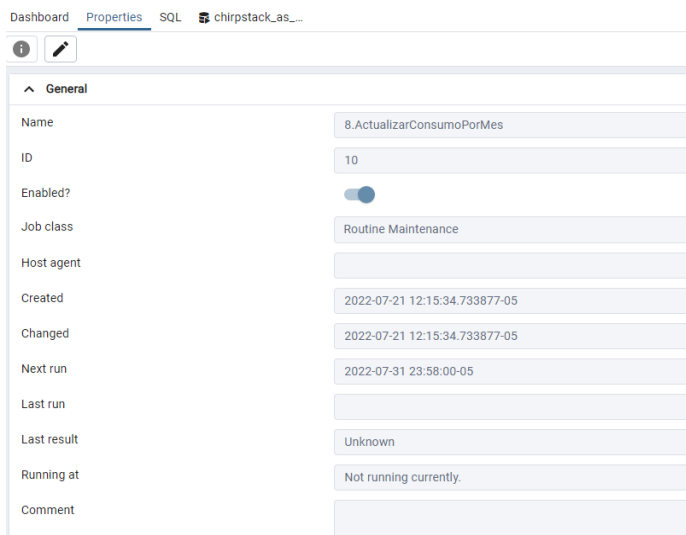


Figura 5.151: Visualización de las propiedades de 'ActualizarConsumoPorMes'.

5.12 Configuración de QuickSight para la visualización de los datos obtenidos en ChirpStack

Para visualizar de manera eficiente los datos obtenidos de Chirpstack hacia Amazon Web Services, se utilizará la herramienta 'QuickSight'.

Para realizar la configuración de forma correcta en el servicio 'QuickSight' de Amazon se debe seguir los siguientes pasos que se explican a continuación.

1. En la plataforma de Amazon Web Services seleccionamos la pestaña 'Servicios' y se coloca el nombre 'QuickSight' como se puede observar en la Fig.5.152.



Figura 5.152: Menú de Amazon Web Services.

2. Una vez seleccionado el servicio 'QuickSight' se redireccionará hacia la ventana principal, donde se procede a registrarse como se puede observar en la Fig.5.153.



Figura 5.153: Menú de QuickSight.

3. Para realizar la inscripción en 'QuickSight' se utiliza las credenciales de Amazon Web Services donde se procede a colocar el correo electrónico y contraseña como se puede observar en la Fig.5.154.



Sign in

Root user

Account owner that performs tasks requiring unrestricted access. [Learn more](#)

IAM user

User within an account that performs daily tasks. [Learn more](#)

Root user email address

tesis_carrascodeleg@hotmail.com

Next

By continuing, you agree to the [AWS Customer Agreement](#) or other agreement for AWS services, and the [Privacy Notice](#). This site uses essential cookies. See our [Cookie Notice](#) for more information.

[New to AWS?](#)

Figura 5.154: Registro de correo electrónico.

4. Como se puede observar en la Fig.5.155 se coloca la contraseña para acceder a la cuenta de Amazon Web Services.



Root user sign in

Email: tesis_carrascodeleg@hotmail.com

Password

[Forgot password?](#)

.....|

Sign in

[Sign in to a different account](#)

[Create a new AWS account](#)

Figura 5.155: Registro de contraseña para acceder a AWS.

5. Una vez realizada la autenticación se mostrará un mensaje indicando que se ha ejecutado con éxito la inscripción en Amazon QuickSight como se puede observar en la Fig.5.156.

¡Felicitaciones! Se ha inscrito en Amazon QuickSight.

Obtenga acceso a QuickSight con la siguiente información

Nombre de la cuenta: tesis-lora

Los permisos relacionados con los recursos no se han actualizado. Edite los permisos relacionados con los recursos de AWS siguiendo las instrucciones que están disponibles en [Administración de permisos de Amazon QuickSight para los recursos de AWS](#).

Vaya a Amazon QuickSight

Figura 5.156: Inscripción exitosa en Amazon QuickSight.

6. Luego de realizar la inscripción se procede a dirigirse a la página principal de Amazon QuickSight como se puede observar en la Fig.5.157.

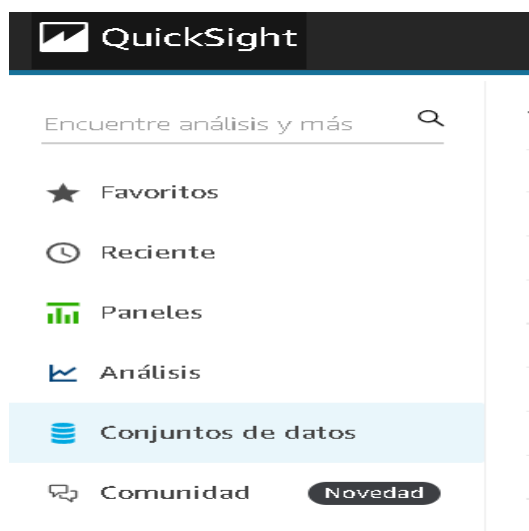


Figura 5.157: Ventana principal de Amazon QuickSight.

7. En la ventana principal de QuickSight se visualiza el menú al lado superior izquierdo, posteriormente se selecciona la opción de 'conjunto de datos' donde se configurará la extracción de datos de la base de datos en PostgreSQL como se puede observar en la Fig.5.158.

Conjuntos de datos Nuevo conjunto de datos

Nombre	Propietario	Última modificación	
Nueva consulta SQL personalizada	Me	hace 15 días	...
Datos.xlsx	Me	hace 20 días	...
device_up	Me	hace 21 días	...
device	Me	hace 21 días	...
application	Me	hace 21 días	...
Business Review	Me	hace 22 días	...
Business Review	Me	hace 22 días	...
People Overview	Me	hace 22 días	...
Web and Social Media Analytics	Me	hace 22 días	...

Figura 5.158: Selección de conjunto de datos.

8. Como se puede observar en la Fig.5.159 se selecciona la opción 'Nuevo conjunto de datos', en la cual se configurará el origen de los datos extraídos. Posteriormente en la ventana de 'Conjunto de Datos' se seleccionará la opción de PostgreSQL.

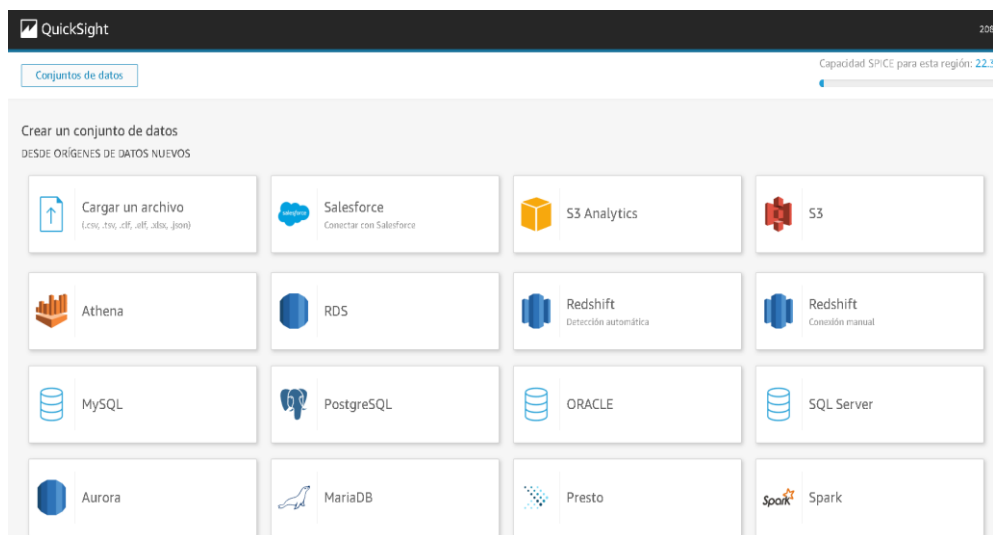


Figura 5.159: Selección de PostgreSQL como nuevo origen de datos.

9. Una vez seleccionado a PostgreSQL como nuevo origen de datos, se desplegará una ventana donde se ingresará varios parámetros como se puede observar en la Fig.5.160.
- Nombre del origen de datos.

- Tipo de conexión.
- IP del servidor de base de datos.
- Puerto.
- Nombre de la base de datos.
- Nombre de usuario.
- Contraseña.

Nuevo origen de datos PostgreSQL

Nombre del origen de datos
chirpstack_events

Tipo de conexión
Red pública

Servidor de bases de datos
34.148.16.7

Puerto
5432

Nombre de la base de datos
chirpstack_as_events

Nombre de usuario
chirpstack_as_events

Contraseña

Validar conexión Habilitar SSL

Figura 5.160: Configuración de PostgreSQL como nuevo origen de datos.

10. Luego de validar la conexión de la base de datos se procede a crear el origen de datos, posteriormente se visualizará una ventana donde se seleccionará la opción 'Crear conjunto de datos' como se puede observar en la Fig.5.161.

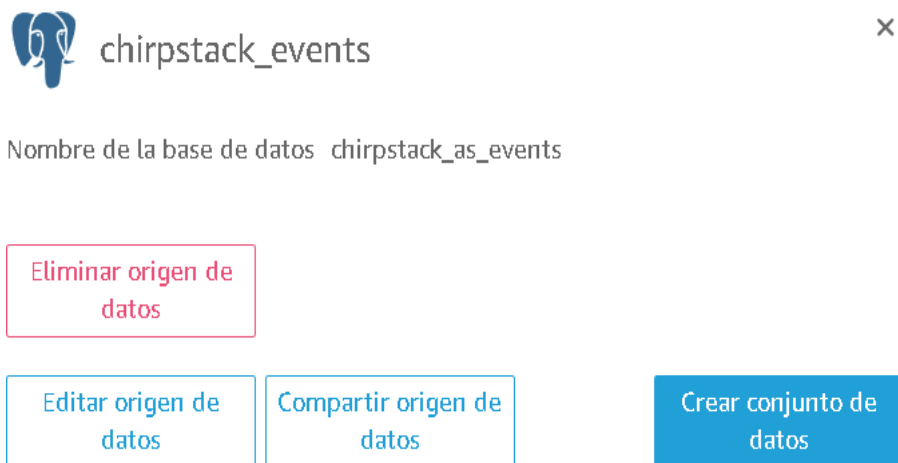


Figura 5.161: Creación exitosa del origen de datos.

11. Una vez creado el conjunto de datos se visualiza una ventana donde se procede a seleccionar el esquema y tablas a utilizar o se selecciona la opción de utilizar una consulta SQL personalizada como se puede observar en la Fig.5.162.

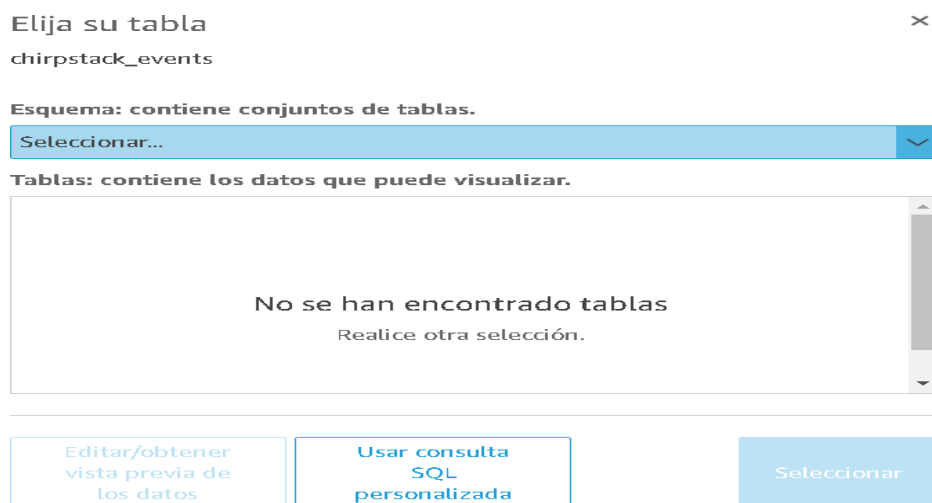


Figura 5.162: Selección de una consulta SQL personalizada.

12. Se procede a digitar la consulta SQL personalizada para una mayor facilidad se recomienda utilizar la opción de editar y obtener vista previa de los datos como se puede observar en la Fig.5.163.

Origen de datos chirpstack_events PostgreSQL Cerrar Aplicar

Conjunto de datos Nueva consulta SQL personalizada x

id	application_name	device_name	frequency	fecha	m3	mes	m3mes
797cac75-29a4-4cff-8593-...	MEDIDORES	T11637100201	903500000	2022-07-06T00:00:00.000Z	0.0023	2022-07-01T00:00:00.000Z	2.8355
797cac75-29a4-4cff-8593-...	MEDIDORES	T11637100201	903500000	2022-07-07T00:00:00.000Z	0.0201	2022-07-01T00:00:00.000Z	2.8355
797cac75-29a4-4cff-8593-...	MEDIDORES	T11637100201	903500000	2022-07-08T00:00:00.000Z	0.0035	2022-07-01T00:00:00.000Z	2.8355
797cac75-29a4-4cff-8593-...	MEDIDORES	T11637100201	903500000	2022-07-09T00:00:00.000Z	0.0	2022-07-01T00:00:00.000Z	2.8355
797cac75-29a4-4cff-8593-...	MEDIDORES	T11637100201	903500000	2022-07-10T00:00:00.000Z	0.0	2022-07-01T00:00:00.000Z	2.8355
797cac75-29a4-4cff-8593-...	MEDIDORES	T11637100201	903500000	2022-07-11T00:00:00.000Z	0.0241	2022-07-01T00:00:00.000Z	2.8355
797cac75-29a4-4cff-8593-...	MEDIDORES	T11637100201	903500000	2022-07-12T00:00:00.000Z	0.0	2022-07-01T00:00:00.000Z	2.8355
797cac75-29a4-4cff-8593-...	MEDIDORES	T11637100201	903500000	2022-07-13T00:00:00.000Z	0.4891	2022-07-01T00:00:00.000Z	2.8355
797cac75-29a4-4cff-8593-...	MEDIDORES	T11637100201	903500000	2022-07-14T00:00:00.000Z	0.5413	2022-07-01T00:00:00.000Z	2.8355
797cac75-29a4-4cff-8593-...	MEDIDORES	T11637100201	903500000	2022-07-15T00:00:00.000Z	0.3671	2022-07-01T00:00:00.000Z	2.8355

Figura 5.165: Resultado de la consulta SQL Personalizada.

- Una vez que la consulta SQL funcione correctamente, en la parte superior derecha se da clic sobre la opción de 'publicar y visualizar', posteriormente se mostrará la siguiente pantalla donde se seleccionará el tipo de elemento visual que se utilizará para cada lista de campos como se puede observar en la Fig.5.166.

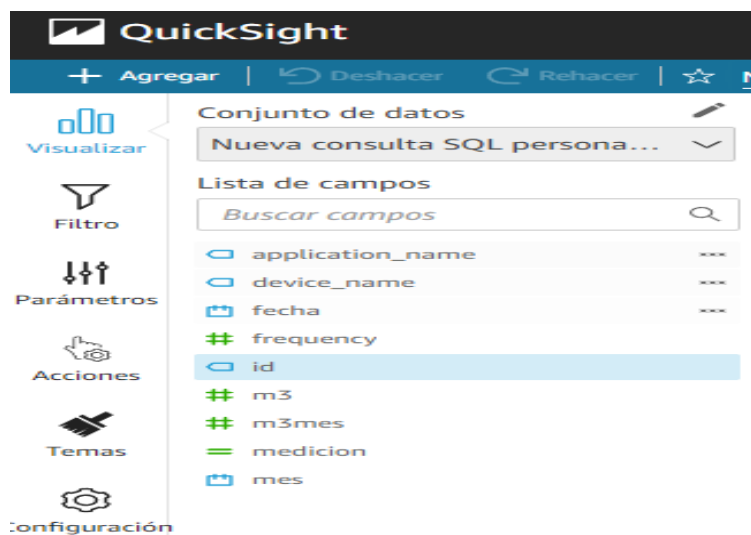


Figura 5.166: Lista de campos a utilizar para elaborar el Dashboard.

- Se selecciona el tipo de elemento visual para la lista de campo a mostrar, en este caso se utilizará el campo 'device_name', y el tipo de elemento visual que se implementará será

el gráfico circular como se puede observar en la Fig.5.167.

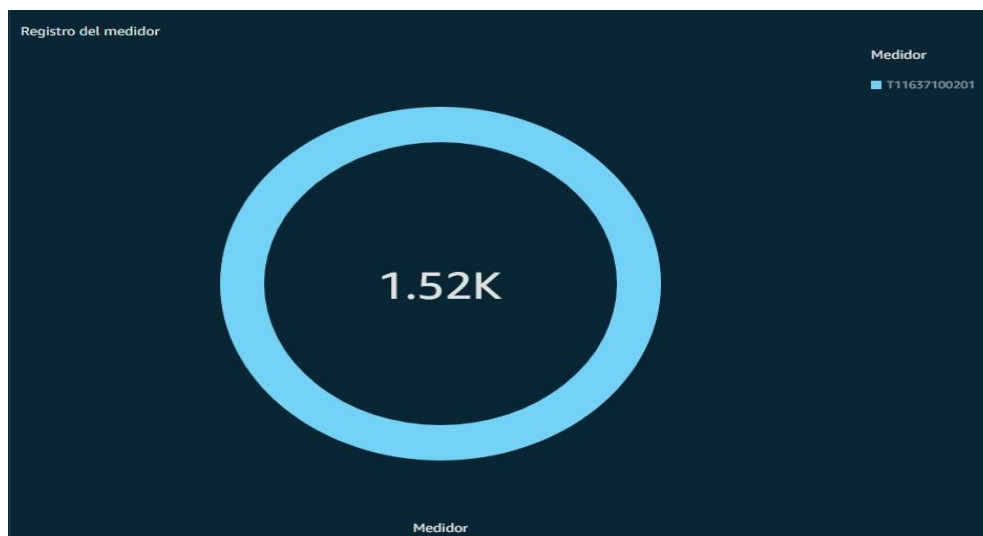


Figura 5.167: Elección del tipo Gráfico circular para visualizar el número de registros por 'device_name'.

17. Se selecciona el tipo de elemento visual para la lista de campo a mostrar, en este caso se utilizará los campos 'fecha' y 'm3', el tipo de elemento visual que se implementará será la tabla dinámica como se puede observar en la Fig.5.168.

Total de consumo al día	
fecha	m3
jul. 20, 2022	0.17562
jul. 19, 2022	0.39561
jul. 18, 2022	0.29730
jul. 17, 2022	0.31634
jul. 16, 2022	0.20316

Figura 5.168: Elección del tipo tabla dinámica para visualizar la cantidad de consumo al día.

18. Se selecciona el tipo de elemento visual para la lista de campo a mostrar, en este caso se

utilizará los campos: 'fecha' y 'm3', el tipo de elemento visual que se implementará será el gráfico de líneas como se puede observar en la Fig.5.169.

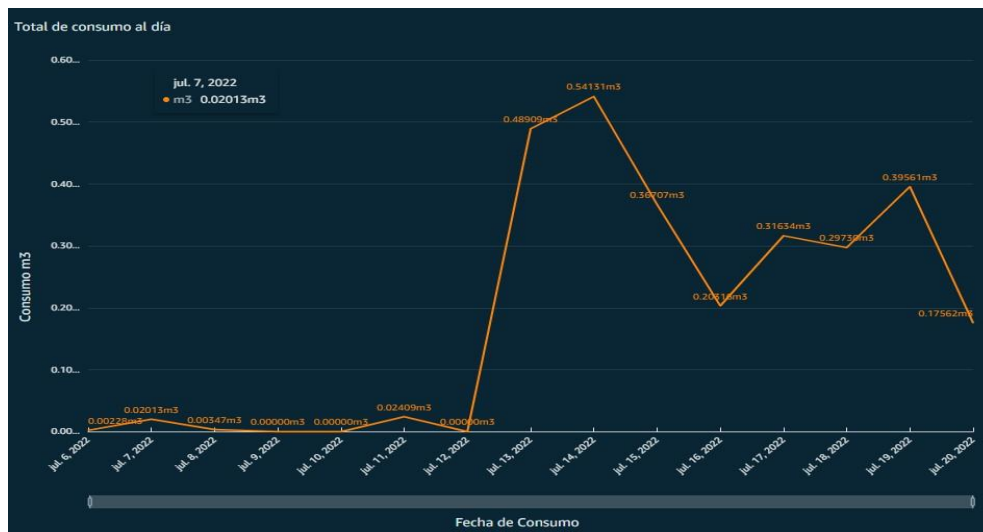


Figura 5.169: Elección del tipo Gráfico de líneas para visualizar la cantidad de consumo en metros cúbicos en base a la fecha de consumo.

- Se selecciona el tipo de elemento visual para la lista de campo a mostrar, en este caso se utilizará los campos 'fecha' y 'm3', el tipo de elemento visual que se implementará será la tabla dinámica como se puede observar en la Fig.5.170.

mes	m3
Jul. 2022	2.83547

Figura 5.170: Elección del tipo tabla dinámica para visualizar la cantidad de consumo por mes.

- Se selecciona el tipo de elemento visual para la lista de campo a mostrar, en este caso se utilizará los campos: 'fecha' y 'm3', el tipo de elemento visual que se implementará será el gráfico de líneas como se puede observar en la Fig.5.171.



Figura 5.171: Elección del tipo Gráfico de líneas para visualizar la cantidad de consumo en metros cúbicos en base al mes de consumo.

21. Se selecciona el tipo de elemento visual para la lista de campo a mostrar, en este caso se utilizará el campo 'id', y el tipo de elemento visual que se implementará será el gráfico circular como se puede observar en la Fig.5.172.



Figura 5.172: Elección del tipo Gráfico circular para visualizar el número de registros por 'id'.

22. Se selecciona el tipo de elemento visual para la lista de campo a mostrar, en este caso se utilizará el campo 'frequency', y el tipo de elemento visual que se implementará será un

Histograma como se puede observar en la Fig.5.173.

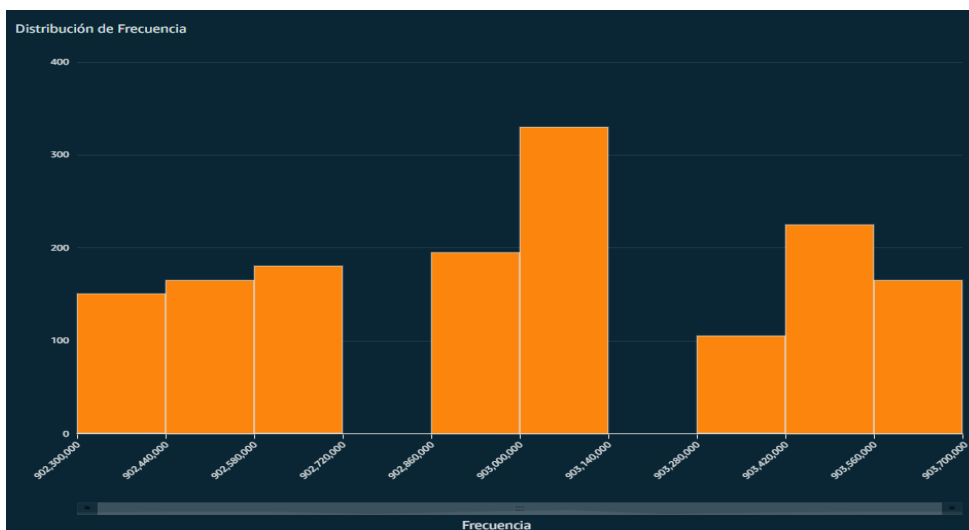


Figura 5.173: Elección del tipo Histograma para visualizar la distribución de frecuencia por 'frequency'.

23. Para publicar el Dashboard una vez creado se procede a dirigirse a la parte superior derecha y se selecciona la pestaña 'Compartir' y se elige la opción 'Publicar panel' como se puede observar en la Fig.5.174.



Figura 5.174: Publicación del panel.

24. Luego de publicar el panel se visualiza de mejor manera el Dashboard creado como se puede observar en la Fig.5.175.

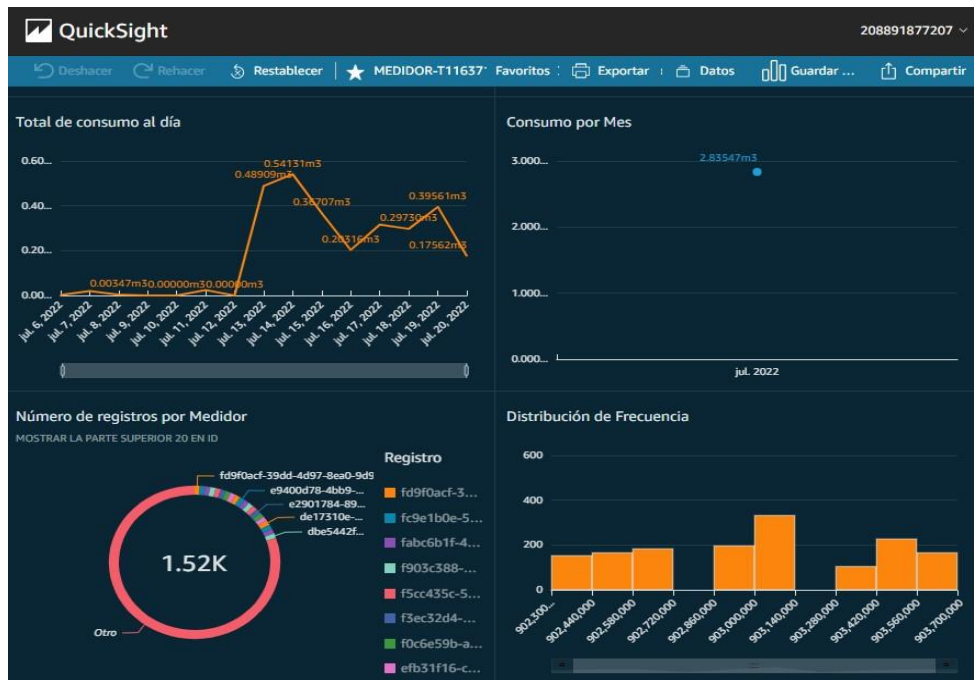


Figura 5.175: Visualización del panel publicado.

Capítulo VI

RESULTADOS

En este capítulo se puede visualizar el resultado del diseño e implementación que se ha realizado para la integración del servidor de red ChirpStack con Amazon Web Services.

Conjuntamente se presenta una explicación de los componentes que conforman el prototipo final el cual permitió observar el consumo de agua en los medidores inteligentes en base al flujo de agua.

6.1 Ejecución de los planes de pruebas.

Para poder realizar la ejecución de pruebas se puede observar en la Fig.6.1 que se procedió a elaborar un prototipo funcional.

Donde se implementó un Gateway LoRa conjuntamente con los dispositivos LoRaWAN (medidores de agua inteligente), los cuales permitieron la comunicación con el servidor de red ChirpStack.

Para la instalación de la parte frontal del prototipo se utilizaron los siguientes materiales:

- Monitor.
- Raspberry pi.
- Ratón.
- Medidores inteligentes de agua.
- Lavamanos de dos llaves sencillas.



Figura 6.1: Elaboración de la parte frontal del prototipo

Los medidores que se implementaron en el proyecto pertenecen a la empresa TARPUQ, los cuales permiten medir y enviar datos de consumo siendo que los 5 primeros dígitos están en decímetros cúbicos y los últimos 3 dígitos están midiendo en centímetros cúbicos como se puede observar en la Fig.6.2, posteriormente para obtener el consumo en metros cúbicos se debe realizar los cálculos correspondientes.



Figura 6.2: Funcionamiento del Medidor Inteligente

Para la instalación de la parte posterior del prototipo se utilizaron varios tubos PVC rosca- bles, uniones PVC, codos rosca- bles. Una vez determinada la distribución de los elementos se procedió a perforar la caja para colocar la tubería plástica que permite el flujo de agua hacia los medidores inteligentes como se puede observar en la Fig.6.3.



Figura 6.3: Instalación de la tubería para el flujo de agua

Para permitir el flujo de agua hacia los medidores se instaló una bomba de agua, la cual permite reutilizar el agua que se encuentra en el recipiente para así evitar el gasto indebido de

agua como se puede observar en la Fig.6.4.

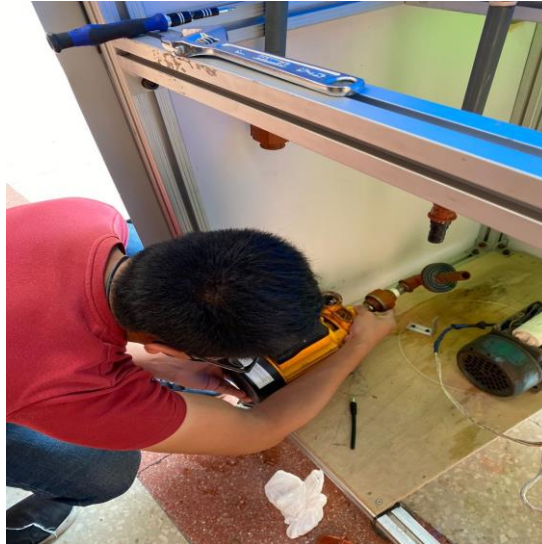


Figura 6.4: Colocación de la bomba de agua

Para permitir el paso del flujo de agua hacia los medidores se cuenta con una implementación de varios tubos PVC, un recipiente de agua, y un motor los cuales permitirán reutilizar el agua que se encuentra en el recipiente para dar paso al flujo de agua hacia los medidores inteligentes como se puede observar en la Fig.6.5.



Figura 6.5: Elaboración de la parte posterior del prototipo

Para la comunicación entre los medidores inteligentes y el servidor de red ChirpStack se implementó un Gateway LoRa, el cual permite realizar la conexión entre la plataforma y los dispositivos como se puede observar en la Fig.6.6.



Figura 6.6: Implementación de un Gateway LoRa

6.1.1 Prueba de funcionalidad de la plataforma ChirpStack

La plataforma ChirpStack maneja diferentes dispositivos LoRaWAN como se puede observar en la Fig.6.7 y Fig.6.8, ya que estos envían datos hacia el servidor de red Chirpstack, a través de un Gateway. Estos dispositivos permitirán medir el consumo de agua.

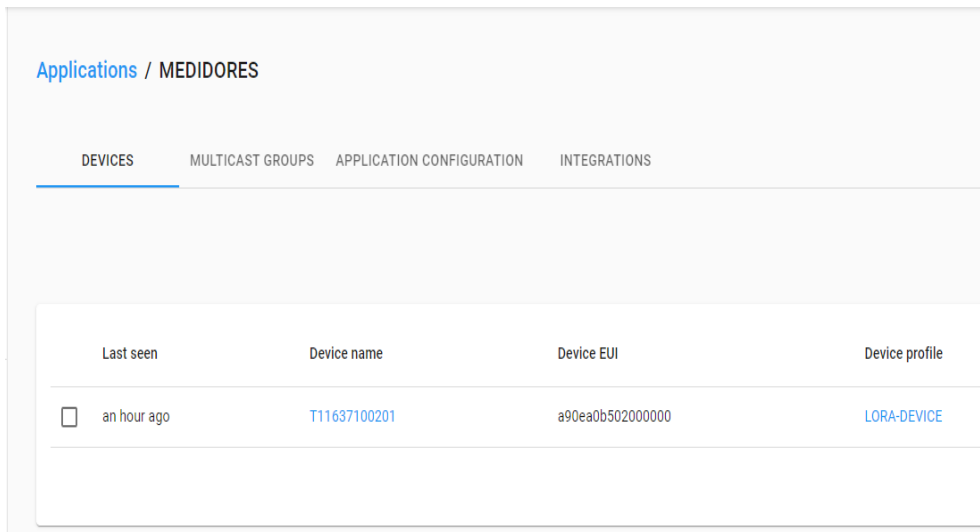


Figura 6.7: Visualización del estado de la conexión de los medidores inteligentes en la plataforma ChirpStack

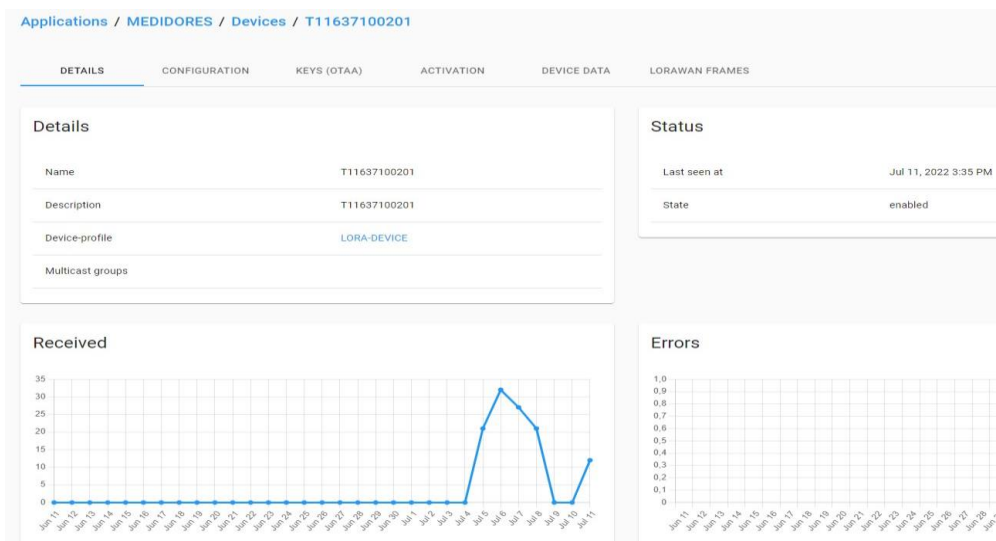


Figura 6.8: Visualización del estado de la última conexión del medidor inteligente.

6.1.2 Prueba de funcionalidad SQS(AWS)

En la plataforma de Amazon Web Services se pudo visualizar la recepción de mensajes en el servicio Simple Queue Service (SQS) de los datos obtenidos de los medidores inteligentes una vez realizado la integración con la plataforma ChirpStack como se puede observar en la Fig.6.9.

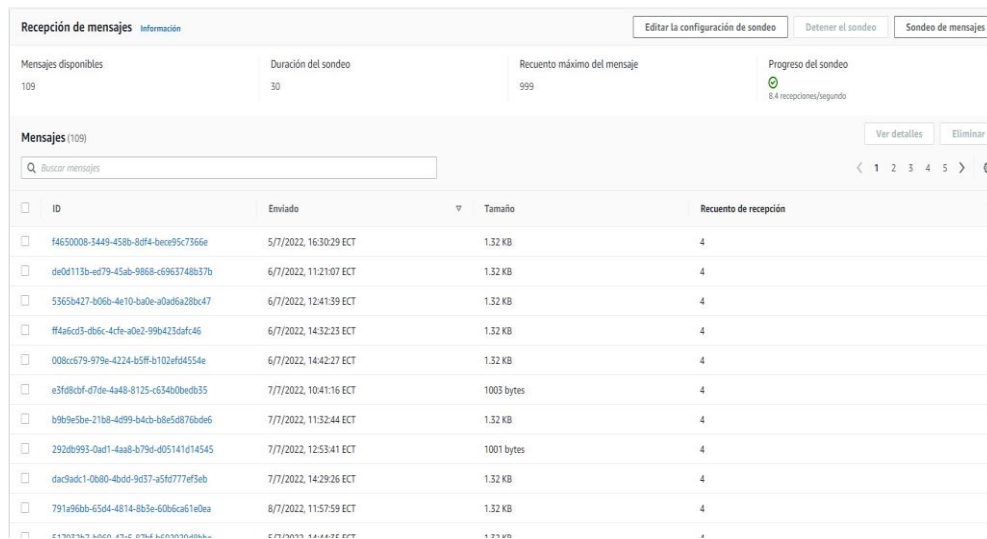


Figura 6.9: Elaboración de la parte posterior del prototipo.

6.1.3 Prueba de funcionalidad QuickSight

A continuación, se puede visualizar la información sobre el consumo diario de agua durante la monitorización del prototipo como se puede observar en la Fig.6.10.

Total de consumo al día	
fecha	m3
jul. 13, 2022	0.48909
jul. 12, 2022	0.00000
jul. 11, 2022	0.02409
jul. 10, 2022	0.00000
jul. 9, 2022	0.00000
jul. 8, 2022	0.00347
jul. 7, 2022	0.02013
jul. 6, 2022	0.00228

Figura 6.10: Tabla de consumo diario de agua durante la monitorización

El consumo de agua durante la monitorización se puede visualizar en base a la fecha de consumo y la cantidad de consumo en metros cúbicos como se puede observar en la Fig.6.11.

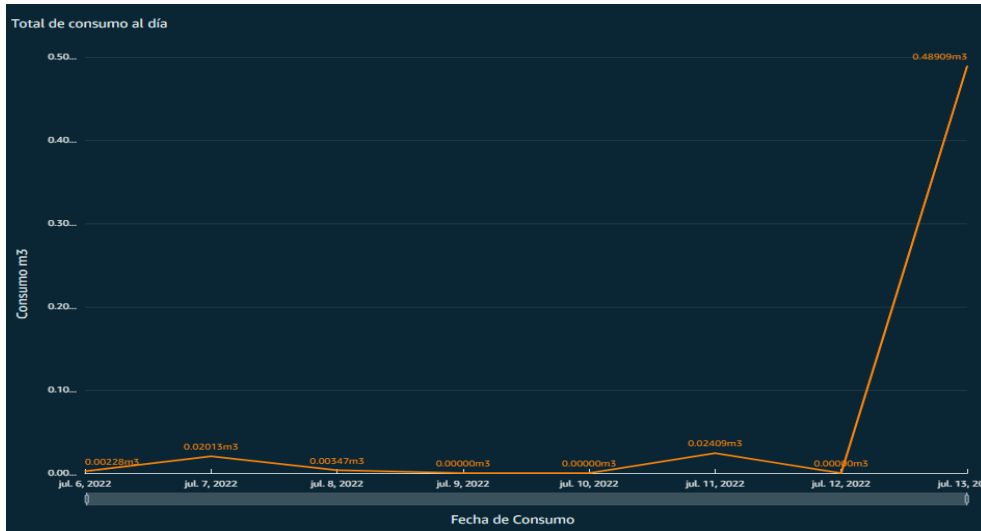


Figura 6.11: Ilustración del consumo diario de agua durante la monitorización

A continuación, se puede visualizar la información sobre el consumo por mes de agua durante la monitorización del prototipo como se puede observar en la Fig.6.12.

mes	m³
jul. 2022	5.88384

Figura 6.12: Tabla de consumo de agua por mes durante la monitorización

El consumo de agua durante la monitorización se puede visualizar en base a la fecha de consumo y la cantidad de consumo en metros cúbicos como se puede observar en la Fig.6.13.

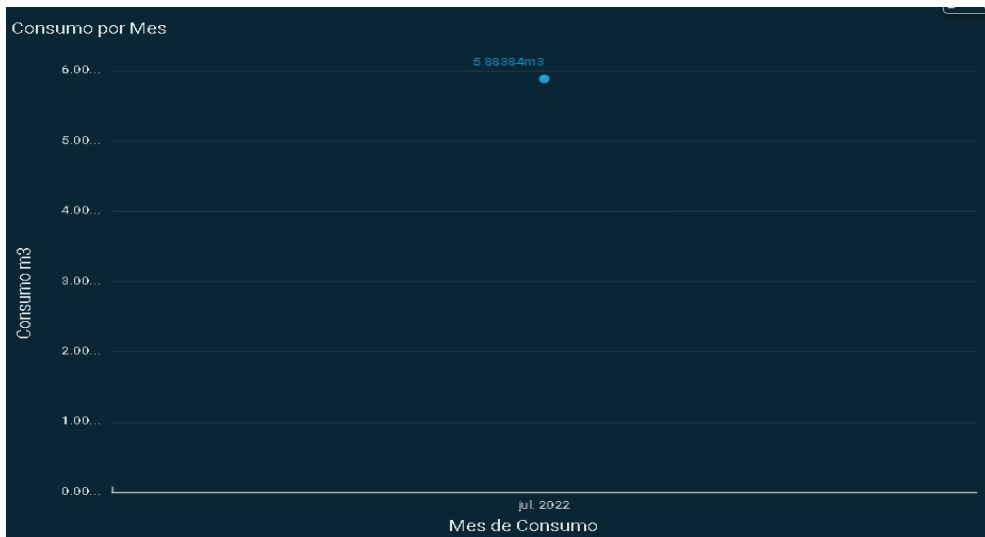


Figura 6.13: Ilustración del consumo de agua por mes durante la monitorización

La distribución de frecuencia utilizada durante la monitorización se puede visualizar en un histograma como se puede observar en la Fig.6.14.

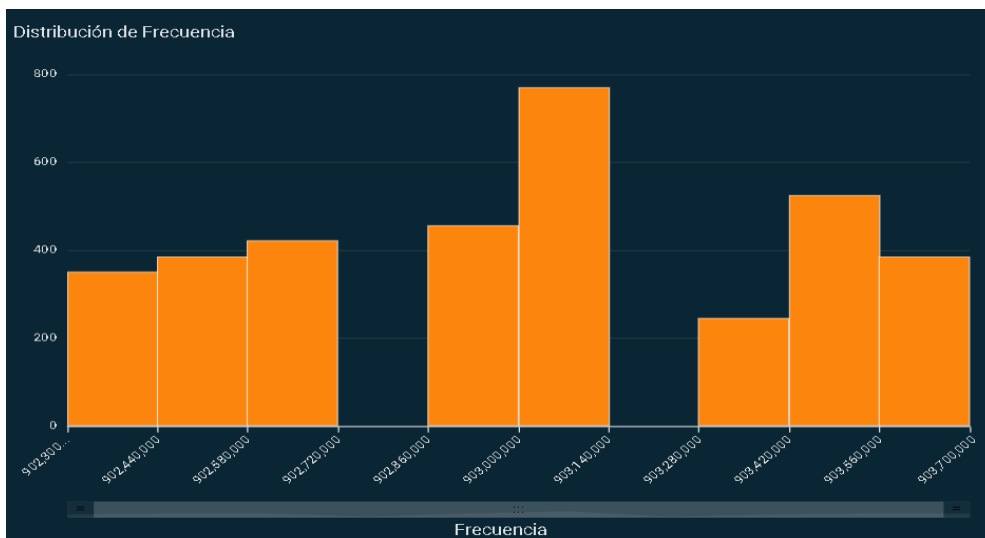


Figura 6.14: Ilustración de la frecuencia utilizada durante la monitorización

Capítulo VII

CRONOGRAMA

Como se observa en la tabla 7.1 tenemos el cronograma de actividades que se llevó en el proyecto.

Nombre de la tarea	Duración	Comienzo	Fin
Proyecto	800 horas	11/04/2022	27/07/2022
OE. 1	186 horas	11/04/2022	04/05/2022
ACT 1. Estudio de los fundamentos de LoRa WAN orientado a IoT.	46 horas	11/04/2022	14/04/2022
ACT 2. Estudio de los fundamentos de administración y gestión de una red LoRa WAN.	24 horas	18/04/2022	20/04/2022
ACT 3. Estudio de la implementación de un servidor LORA.	40 horas	21/04/2022	27/04/2022
ACT 4. Estudio de los medidores inteligentes.	32 horas	28/04/2022	29/04/2022
ACT 5. Estudio de los métodos de comunicación entre cliente y servidor.	44 horas	02/05/2022	04/05/2022
OE. 2	138 horas	05/05/2022	23/05/2022
ACT 1. Implementación de un servidor de gestión de tipo GNU Linux en los servidores de la Universidad Politécnica Salesiana.	52 horas	05/05/2022	11/05/2022
ACT 2. Creación de las políticas de seguridad y autenticación dentro del servidor LORA.	60 horas	12/05/2022	18/05/2022
ACT 3. Implementación de la autenticación en los dispositivos de la red LoRaWAN.	26 horas	19/05/2022	23/05/2022
OE. 3	188 horas	24/05/2022	20/06/2022
ACT 1. Creación de cuenta en Amazon Web Service.	64 horas	24/05/2022	31/05/2022
ACT 2. Investigar sobre la implementación de ChirpStack en Amazon Web Services.	48 horas	01/06/2022	06/06/2022
ACT 3. Implementar Amazon Web Service para obtener la data de ChirpStack.	40 horas	07/06/2022	13/06/2022
ACT 4. Obtener una mejor visualización de todos los datos que se tienen de ChirpStack.	36 horas	14/06/2022	20/06/2022

OE. 4	168 horas	21/06/2022	12/07/2022
ACT 1. Investigar sobre las diferentes herramientas que posee Amazon Web Service.	32 horas	21/06/2022	24/06/2022
ACT 2. Integrar los datos obtenidos de la plataforma de ChirpStack hacia AWS.	72 horas	27/06/2022	04/07/2022
ACT 3. Analizar y visualizar los datos de manera eficiente.	64 horas	05/07/2022	12/07/2022
OE. 5	120 horas	13/07/2022	27/07/2022
ACT 1. Diseño del plan de pruebas de integración.	30 horas	13/07/2022	15/07/2022
ACT 2. Diseño del plan de pruebas funcionales, no funcionales y de aceptación.	30 horas	18/07/2022	20/07/2022
ACT 3. Ejecución de los planes de pruebas.	60 horas	21/07/2022	27/07/2022

Tabla 7.1: Cronograma de actividades

Capítulo VIII

PRESUPUESTO

Como se observa en la tabla 8.1 tenemos el presupuesto que se llevó en el proyecto.

DENOMINACION	CANTIDAD	COSTO UNITARIO	COSTO TOTAL
	unidades	dólares	dólares
1. Bienes			
Copias	200	0,05	10,00
Impresiones	500	0,10	50,00
2. Tecnológico			
Laptop	2	1000,00	2000,00
wAP LR9 kit	1	200,00	200,00
Raspberry Pi	1	190,00	190,00
Monitor AOC E2070S	1	100,00	100,00
Mouse Inalámbrico	1	10,00	10,00
Teclado GENIUS USB	1	15,00	15,00
3. Servicios			
GoogleCloud	1	25	25
Amazon Web Service	1	5	5
QuickSight	1	15	15
5. Otros			
Tubos PVC 3/4 de 6 metros	1	13	13,00
Codo Roscable 3/4	6	0,80	4,80
Tee Roscable 3/4	5	0,80	4,00
Unión Roscable 3/4	6	0,40	2,40
Mangueras flexibles para agua	4	10,00	40,00
Fregadero	1	35,00	35,00
Llave de agua	2	15,00	30,00
Imprevistos	1	100,00	100,00
Total	736	\$ 1735,15	\$ 2849,2

Tabla 8.1: Presupuesto del proyecto

Capítulo IX

CONCLUSIONES

En cuanto a la implementación y servicio de alojamiento del servidor de ChirpStack, se optó para que su alojamiento sea en la plataforma de Google Cloud debido a su fácil implementación, de tal forma que el acceso al mismo se lo pueda realizar de manera remota, teniendo que previamente habilitar los puertos de red correspondientes, para establecer la comunicación.

Para realizar la simulación del consumo de agua, se realizó un prototipo dentro de un ambiente cerrado, cuyo funcionamiento se limita al consumo de agua de dos grifos, en donde cada uno de estos está dividido por tuberías de PVC, dándonos así una fácil y mejor manera de controlar el consumo de agua.

Para realizar la conexión entre los dos medidores inteligentes implementados en el prototipo y la plataforma de ChirpStack al momento de configurar la implementación del medidor en la plataforma de ChirpStack se debe adicionar la clave OTTA con la que cuenta cada medidor, esta clave es única para cada dispositivo.

Si se desea implementar los medidores inteligentes registrados anteriormente en otra plataforma

de ChirpStack para registrarlos se debe eliminar el medidor inteligente que ya se encuentra registrado en la primera plataforma de ChirpStack y posteriormente registrarlo en la nueva plataforma de ChirpStack e ingresar la misma clave OTAA.

En relación a la forma de obtener los datos del medidor, se debe tener en cuenta que estos se han calculado a través de una fórmula matemática, la misma que nos da como resultado en centímetros cúbicos, valores que han sido corroborados y verificados y consecuentemente, existe una alta precisión en los mismos.

Amazon Web Service posee el servicio QuickSight, el mismo que brinda un Dashboard en el que se puede observar gráficas lineales que contienen todos los registros del consumo que se guardan dentro de la base de datos de PostgreSQL, obteniendo una alta calidad en la presentación de los mismos

Por lo tanto, podemos indicar que la integración de ChirpStack con Amazon Web Service, nos ha generado una herramienta eficiente para la presentación de este proyecto, por lo que la experiencia de integración entre ambas plataformas tecnológicas ha sido exitosa.

Capítulo X

RECOMENDACIONES

Debido a la seguridad con la que cuenta la Universidad Politécnica Salesiana en los puertos de red internos, se sugiere utilizar Internet con todos los accesos, como por ejemplo los datos móviles, evitando así la pérdida de información que envían los medidores inteligentes hacia el Gateway LoRa.

El Gateway wAP LR9 kit tiene aproximadamente un alcance teórico de 50 metros ya que se está utilizando la antena interna que posee el Gateway, si se quiere extender el rango de alcance se debe optar por instalar una antena adicional que se acople al Gateway.

Se debe tener en cuenta que la base de datos tiene que estar sincronizada con la zona horaria de 'América/Guayaquil', para así evitar problemas.

Para desplegar este proyecto se debe tener en cuenta que debemos automatizar la base de datos, de tal forma que se ejecuten los Query respectivos que ayudan a calcular el consumo diario.

Se recomienda utilizar pgAgent Jobs para automatizar los procesos ya que este es una herramienta propia de PostgreSQL y no se tendrá problema alguno al momento de realizar la ejecución de las tareas.

BIBLIOGRAFÍA

AlfaIOT (2022). Alfaiot - disfruta de la tecnología.

Álvarez, J. P. M., Abdalá, V. I. R., Barboza, F. M. R. M., and Soria, F. R. C. (2021). Estudio descriptivo de lorawan y aplicaciones específicas.

Aranda, J., Sacoto Cabrera, E. J., Haro Mendoza, D., and Astudillo Salinas, F. (2021). 5g networks: A review from the perspectives of architecture, business models, cybersecurity, and research developments.

Arasteh, H., Hosseinnezhad, V., Loia, title=Iot-based smart cities: A survey, V., Tommasetti, A., Troisi, O., Shafie-khah, M., and Siano, P. (2016).

Carrasco Galdame, E. E. (2020). Metodología para selección de tecnologías lpwan para diversas aplicaciones de internet de las cosas.

Castan, S., Antonio, C., et al. (2021). Integridad y confidencialidad de datos en redes iot lorawan.

Chirpstack.io (2022a). Introduction - chirpstack open-source lorawan® network server.

Chirpstack.io (2022b). Introduction - chirpstack open-source lorawan® network server.

Dalbhanjan, P. (2015). Overview of deployment options on aws.

Devalal, S. and Karthikeyan, A. (2018). Lora technology-an overview. In *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*.

Farinango Tuquerres, J. C. (2018). Sistema de medidor inteligente de agua potable domiciliario para la junta administradora de agua “el abra”. Master’s thesis.

Guambuguete Gómez, M. M. and Soledispa Reyes, J. R. (2021). *Diseño e implementación de una red con sensores inalámbricos (WSN) utilizando protocolos open source basado en tecnología lorawan, para medir la contaminación del aire y los niveles de ruido en las extensiones de la Universidad de Guayaquil*. PhD thesis.

Hernández, D. L., Arias, J., Villanueva, D., Gutiérrez, S., and Zavaleta, I. (2021). Iot system prototype based on lora and the orion context broker data model of fiware.

Inc., A. W. S. (2022a). AWS CodeBuild Servicio de compilación completamente administrado.

Inc., A. W. S. (2022b). AWS CodeCommit | Servicio administrado de control de código fuente.

Inc., A. W. S. (2022c). AWS CodePipeline | Integración y entrega continuas.

Inc., A. W. S. (2022d). AWS,Cloud9,Amazon Web Services.

Inc., A. W. S. (2022e). AWS, CodeStar, Desarrolle, compile e implemente rápidamente aplicaciones en AWS.

Inc., A. W. S. (2022f). What is codedeploy?

Khandaray, R. B. and Kemkar, O. (2021). Analysis of lora framework in iot technology.

Lavric, A. and Petrariu, A. I. (2018). Lorawan communication protocol: The new era of iot.

Liya, M. and Aswathy, M. (2020). Lora technology for internet of things (iot): a brief survey.

Lluva Plaza, S. et al. (2021). Desarrollo de un sistema remoto de adquisición de datos basado en lorawan para aplicaciones iot.

Machado González, M. L. (2019). Estudio de nb-iot y comparativa con otras tecnologías lpwan.

Martres, I. P. (2022). Particularidades de la red sigfox.

Maurin, T., Ducreux, L.-F., Caraiman, G., and Sissoko, P. (2018). Iot security assessment through the interfaces p-scan test bench platform.

Muñoz, O. Q. (2019). *Internet de las Cosas (IoT)*.

Natalia, C. R. (2019). Mejora de la seguridad en redes lorawan. Master's thesis.

Olaoye, A. (2022). Devops on amazon web services (aws).

Pardal Garcés, P. (2017). Redes de área extensa para aplicaciones de iot: modelado de comunicaciones sigfox.

Parra-Valencia, J. A., Guerrero, C. D., and Rico-Bautista, D. (2017). Iot: Una aproximación desde ciudad inteligente a universidad inteligente.

Pérez García, R. (2017). Avaluació de lora/lorawan per a escenaris de smart city. Master's thesis.

Pothecary, R. (2021). Developing on aws.

Rodriguez Alvarado, D. P. and Sacoto-Cabrera, E. J. (2021). Implementation and analysis of the results of the application of the methodology for hybrid multi-cloud replication systems. In *The International Conference on Advances in Emerging Trends and Technologies*.

Rodríguez Moreno, E. S., López Ordoñez, V. F., et al. (2017). Diseño e implementación de un sistema inteligente para un edificio mediante iot utilizando el protocolo de comunicación lorawan.

Sacoto-Cabrera, E., Rodriguez-Bustamante, J., Gallegos-Segovia, P., Arevalo-Quishpi, G., and León-Paredes, G. (2017). Internet of things: Informatic system for metering with communications mqtt over gprs for smart meters.

Sacoto-Cabrera, E. J., León-Paredes, G., and Verdugo-Romero, W. (2022). Lorawan: application of nonlinear optimization to base stations location.

- Sacoto Cabrera, E. J., Palaguachi, S., León-Paredes, G. A., Gallegos-Segovia, P. L., and Bravo-Quezada, O. G. (2020). Industrial communication based on mqtt and modbus communication applied in a meteorological network.
- Sacoto-Cabrera, E. J., Sanchis-Cano, A., Guijarro, L., Vidal, J. R., and Pla, V. (2018). Strategic interaction between operators in the context of spectrum sharing for 5g networks.
- Saiz Miranda, J. (2019). Estudio en detalle de nb-iot: comparación con otras tecnologías lpwan considerando diferentes patrones de tráfico.
- Sánchez Rosado, D. (2019). Nb-iot tecnologías celulares narrow-band: análisis práctico de las soluciones de telefónica y vodafone.
- Sanchis-Cano, A., Romero, J., Sacoto-Cabrera, E. J., and Guijarro, L. (2017). Economic feasibility of wireless sensor network-based service provision in a duopoly setting with a monopolist operator.
- Vimos, V. and Cabrera, E. J. S. (2018). Results of the implementation of a sensor network based on arduino devices and multiplatform applications using the standard opc ua.
- Vimos, V., Sacoto, E., and Morales, D. (2016). Conceptual architecture definition: Implementation of a network sensor using arduino devices and multiplatform applications through opc ua.