

| Curso “Especialización en IoT”

Módulo 6: Plataformas Clase 1

Docentes:
Sebastián Valerio Guerrero
Jonattan Silva Castillo



Programa Módulo 6

- **Clase 1: Servicios de mensajería**
 - PubSub, Apache Kafka, RabbitMQ, MQTT Server (mosquitto)
- **Clase 2: Bases de datos**
 - Relacional (SQL), NoSQL, MongoDB, InfluxDB
- **Clase 3: Servicios web**
 - Flask (Python), Express (NodeJS)
- **Clase 4: Servicios de Visualización**
 - Grafana



Programa Módulo 6

- **Clase 5: Taller 1**
 - Freeboard + dweet
- **Clase 6: Taller 2**
 - Ubidots vía HTTP Wemos
- **Clase 7: Taller 3**
 - Ubidots vía servicio Python
- **Clase 8: Taller 4**
 - Wemos + MQTT + InfluxDB + Grafana
- **Clase 9: Arquitecturas de referencia de plataformas**
 - AWS, Azure



PONTIFICIA UNIVERSIDAD
CATOLICA
DE VALPARAISO

Rúbrica Evaluación



Rúbrica de evaluación

- **5%** Autoevaluación
- **15%** Coevaluación
- **30%** Bitácora o Portafolio
- **20%** Ejercicios y guías de estudios
- **30%** Proyecto Aplicado



PONTIFICIA UNIVERSIDAD
CATOLICA
DE VALPARAISO

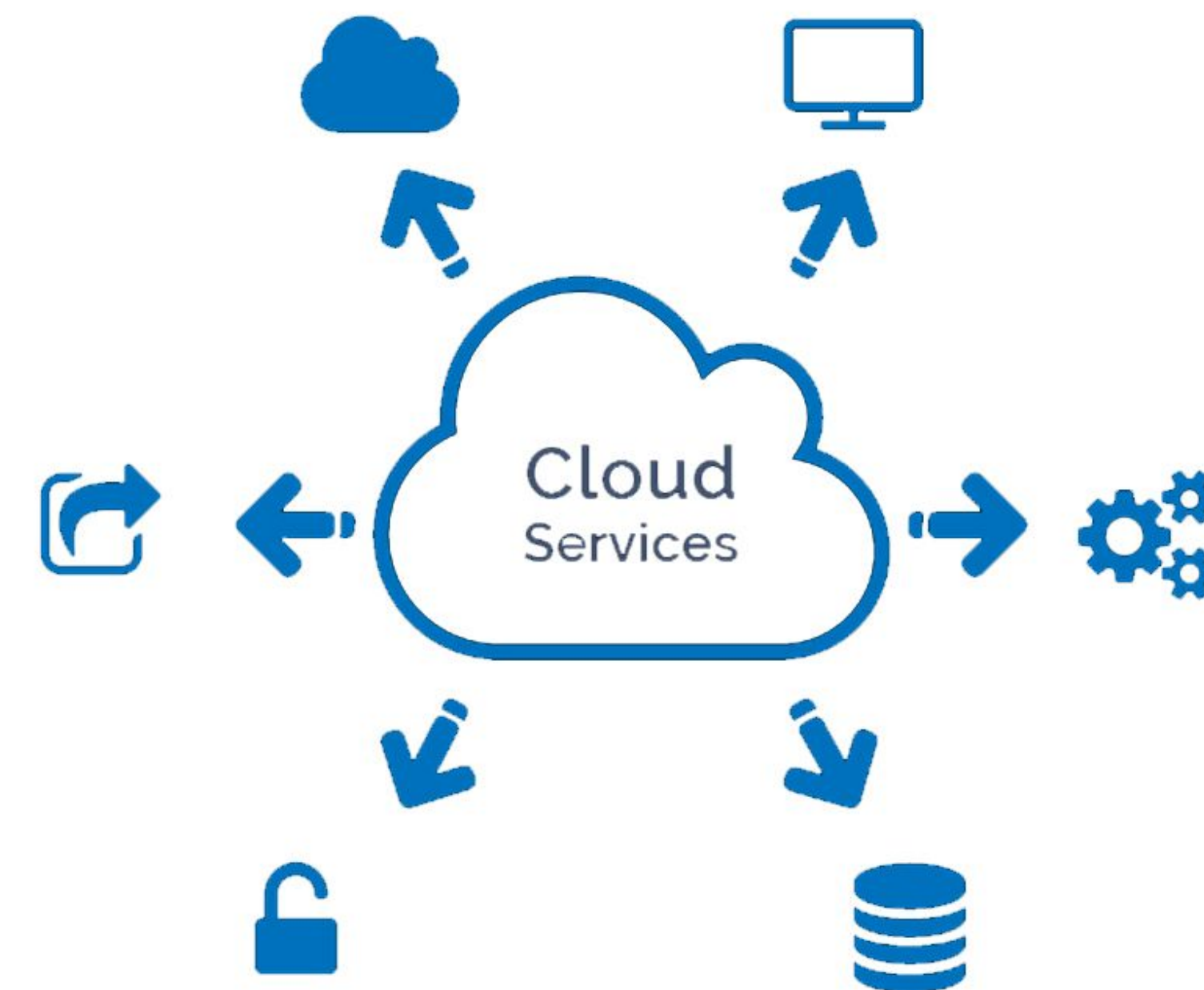
Servicios de mensajería



Qué es un **Servicio**?

Según RedHat:

“Los servicios en la nube son infraestructuras, plataformas o software alojados por proveedores externos y puestos a disposición de los usuarios a través de Internet.”

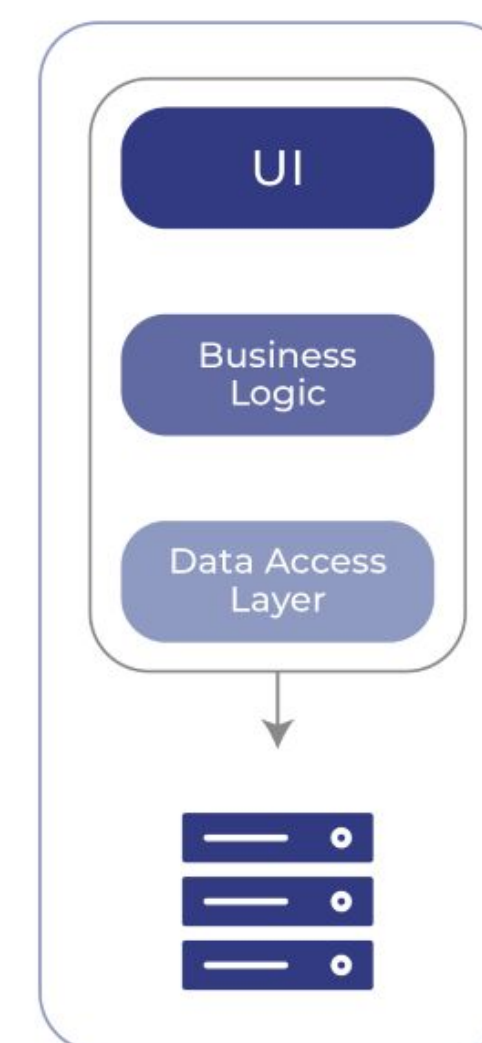


Qué es un **Microservicio**?

Según IBM:

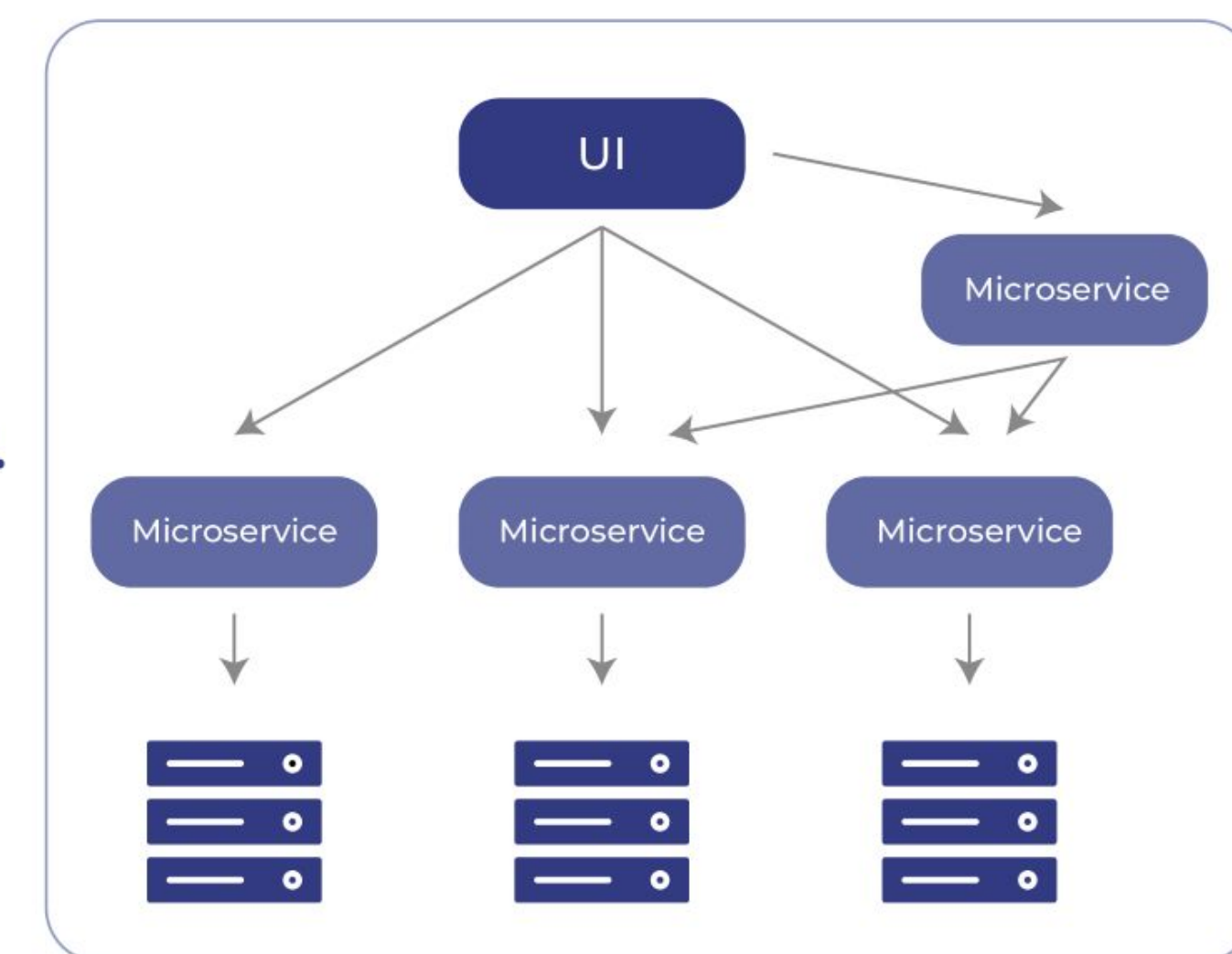
“Los microservicios (o arquitectura de microservicios) son un enfoque arquitectónico nativo de la nube en el que una sola aplicación se compone de muchos componentes o servicios más pequeños acoplados libremente y que se pueden implementar de forma independiente.”

Monolithic



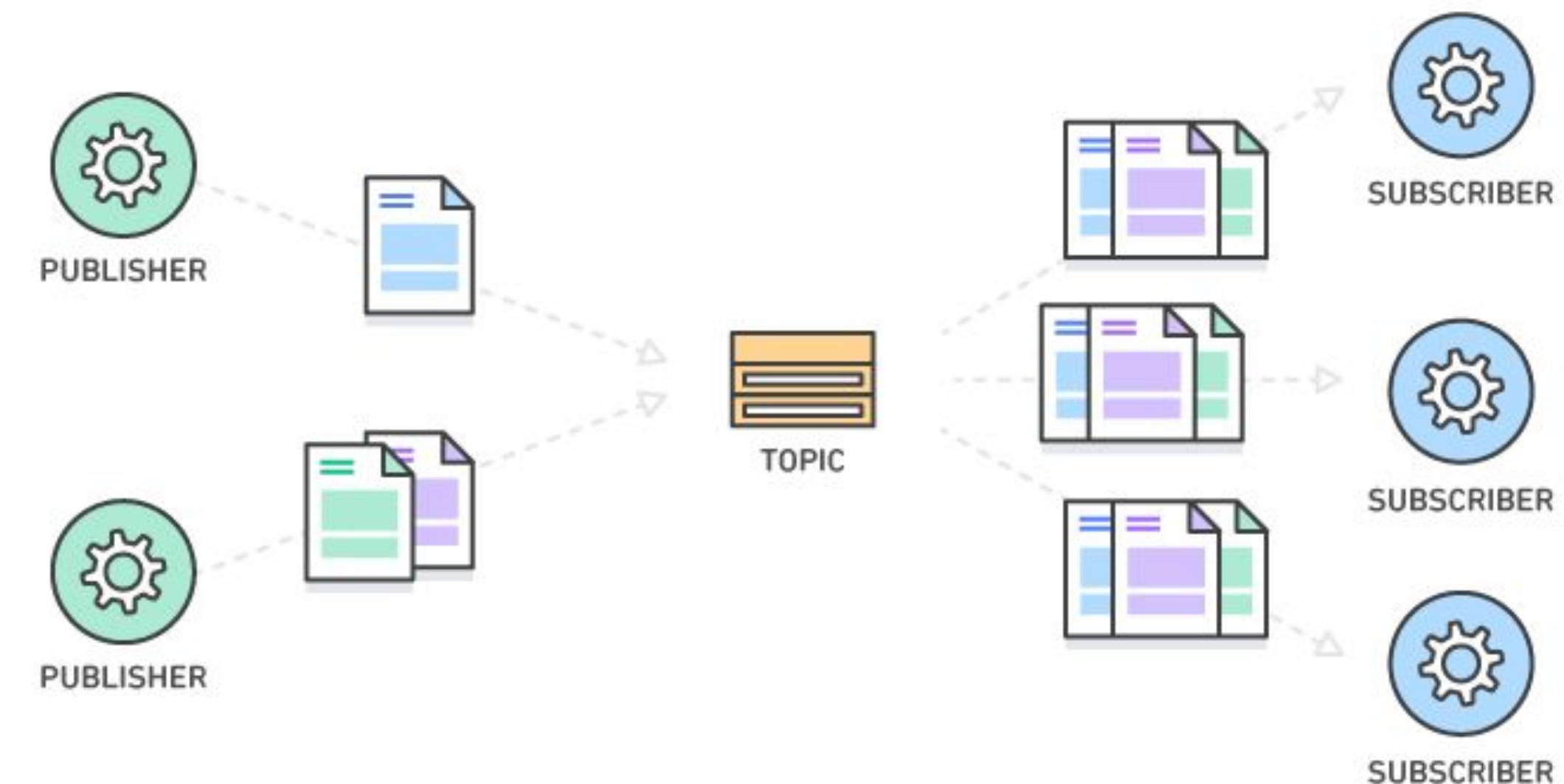
VS.

Microservices



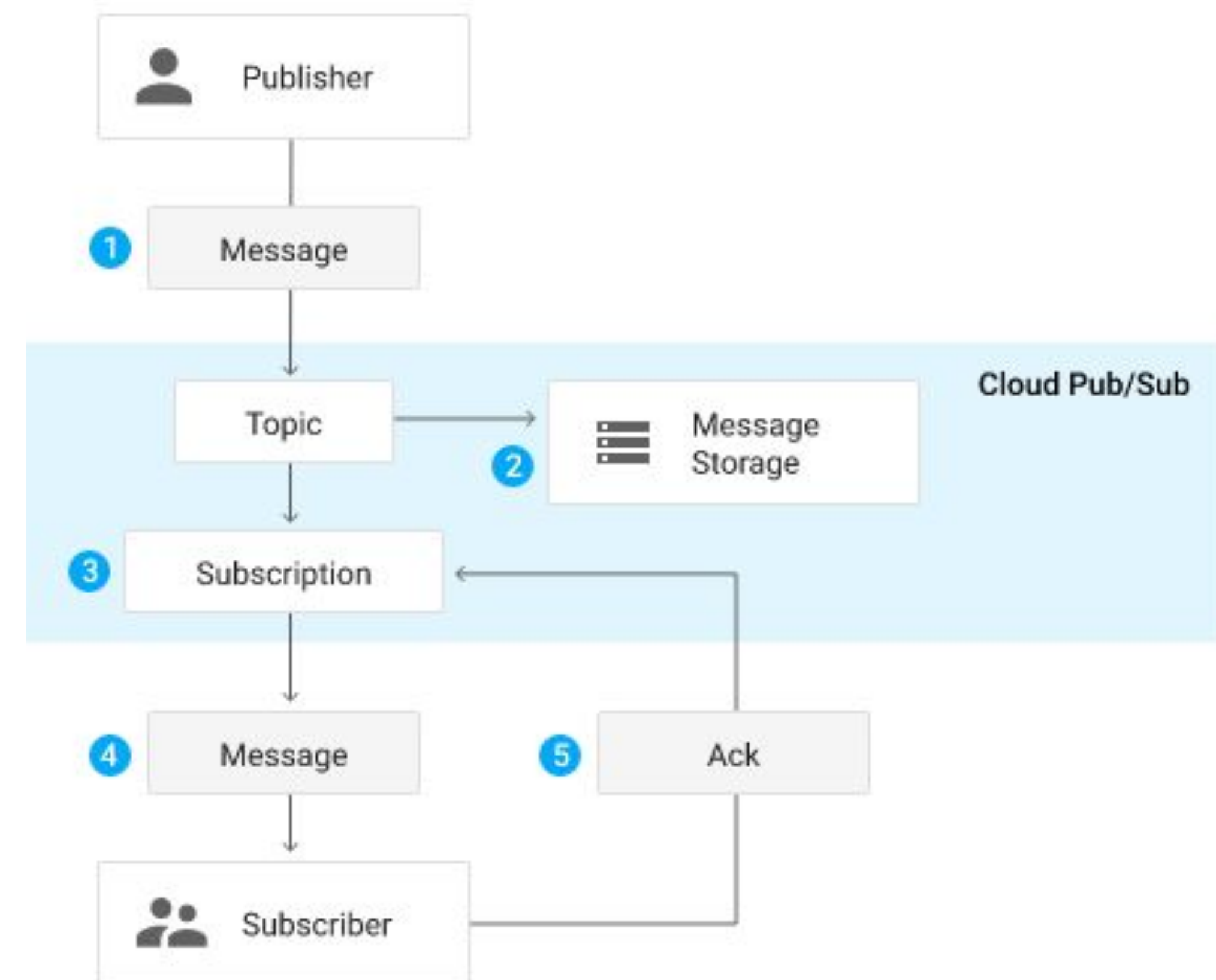
Pub/Sub (GCP/AWS)

- Servicio de comunicación entre servicios.
- Sistemas de productores y consumidores de eventos, llamados **publicadores(Pub)** y **suscriptores(Sub)**.
- Publicadores se comunican con los suscriptores de forma **asíncrona** mediante la transmisión de eventos.



Conceptos Pub/Sub

- **Tema:** recurso con nombre al que los publicadores envían mensajes.
- **Suscripción:** recurso con nombre que representa la transmisión de mensajes de un solo tema específico para entregar a la aplicación de suscripción.
- **Mensaje:** es la combinación de datos y atributos (opcional) que un publicador envía a un tema y, finalmente, se entrega a los suscriptores.
- **Publicador:** una aplicación que crea y envía mensajes a uno o varios temas.
- **Suscriptor:** una aplicación con una suscripción a uno o varios temas para recibir mensajes de ella.



Beneficios de Pub/Sub

Elimina el polling

- Entrega instantánea basada en notificaciones push, elimina la necesidad de que los consumidores de mensajes verifiquen o **sondeen** periódicamente para obtener nueva información y actualizaciones.



Simplifica la comunicación

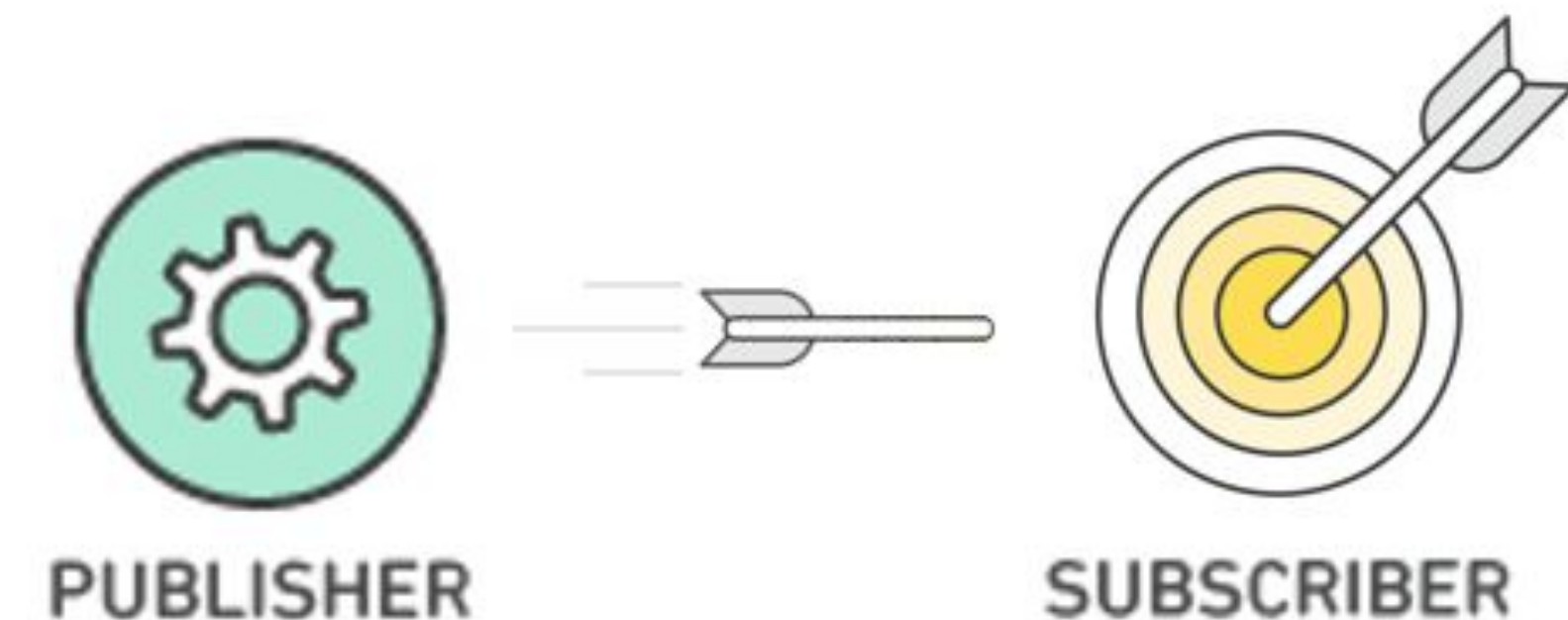
- El modelo Publish Subscribe reduce la complejidad al eliminar todas las conexiones punto a punto con una sola conexión a un tema de mensaje, que administrará las suscripciones para decidir qué mensajes deben entregarse a qué puntos finales.



Beneficios de Pub/Sub

Orientación dinámica

- Descubrimiento de servicios más fácil, más natural y menos propenso a errores. En vez de mantener una lista de pares a los que una aplicación puede enviar mensajes, un editor simplemente publicará mensajes en un tema.
- Cualquier parte interesada se suscribirá al tema y comenzará a recibir estos mensajes.
- Los suscriptores pueden cambiar, actualizar, multiplicar o desaparecer y el sistema se ajusta dinámicamente





PONTIFICIA UNIVERSIDAD
CATOLICA
DE VALPARAISO

Apache Kafka

Es un almacén de datos distribuido y optimizado para la ingesta y procesamiento de datos en modo ***streaming*** en tiempo real.

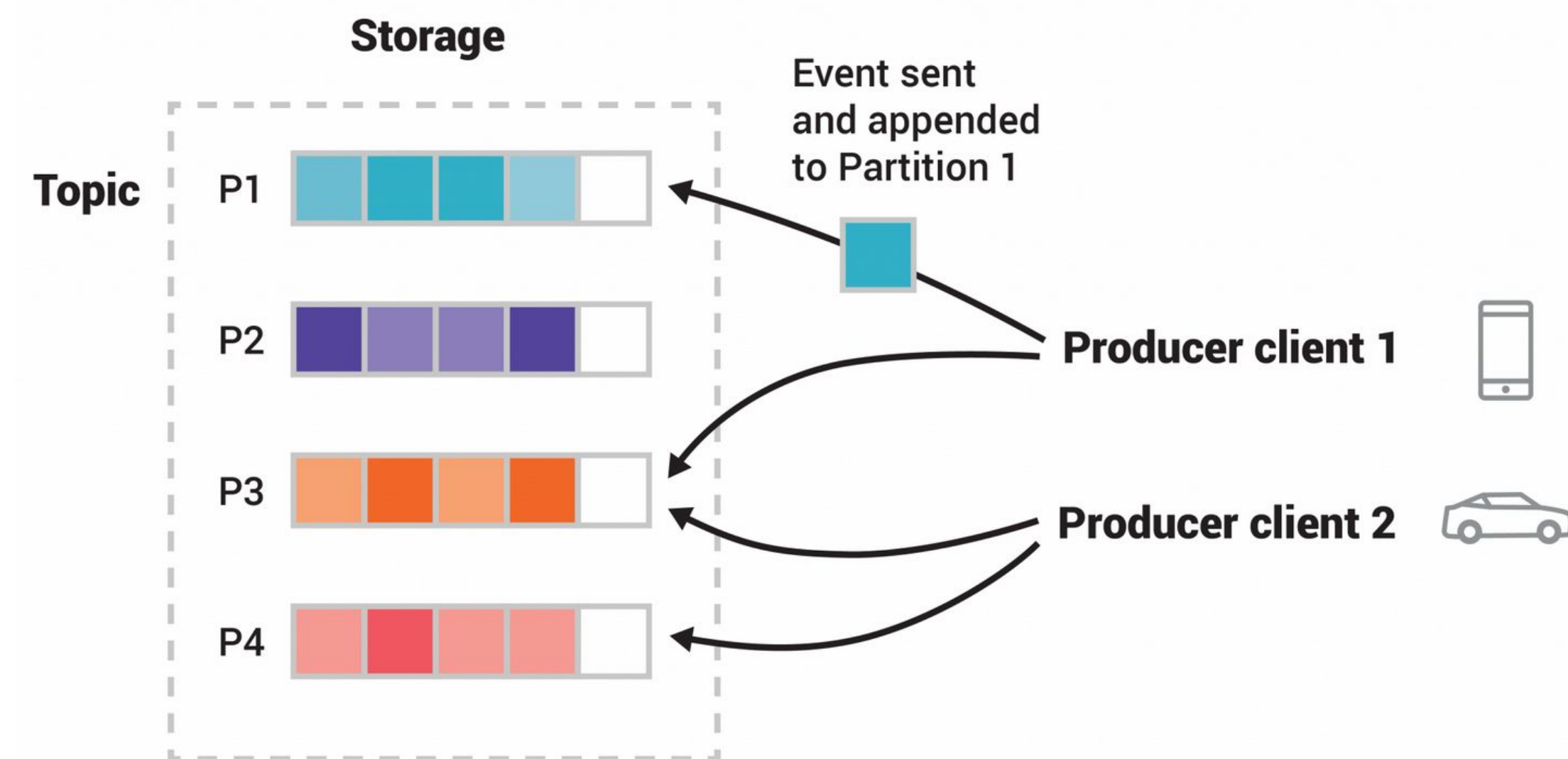
Streaming data son datos generados continuamente por miles de fuentes de datos, que normalmente envían los registros de datos simultáneamente.

Una plataforma de transmisión debe manejar esta afluencia constante de datos y procesar los datos de forma secuencial e incremental.

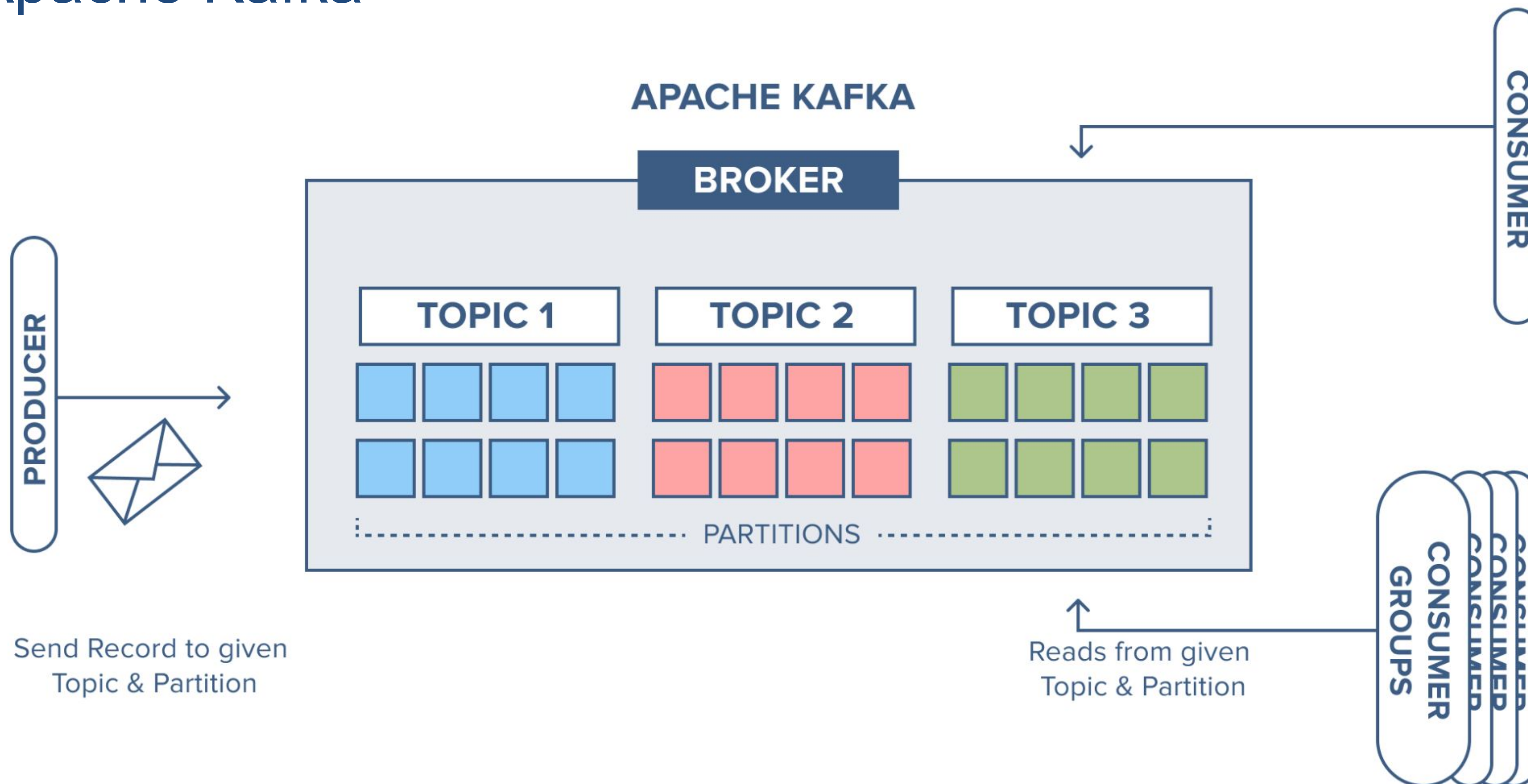


Conceptos Apache Kafka

- **Evento:** en evento registra el hecho de que "algo sucedió" en el mundo. Cuando lee o escribe datos en Kafka, lo hace en forma de eventos. Tiene una clave, un valor, una marca de tiempo y encabezados de metadatos opcionales.
- **Productores:** son aquellas aplicaciones cliente que publican (escriben) eventos en Kafka.
- **Consumidores:** son los que se suscriben (leen y procesan) estos eventos.
- **Temas:** los eventos se organizan y almacenan de forma duradera en temas. Un tema es similar a una carpeta en un sistema de archivos, y los eventos son los archivos en esa carpeta.
- **Partición:** los temas están particionados, lo que significa que un tema se distribuye en varios *buckets* ubicados en diferentes *brokers* de Kafka.



Apache Kafka



Eclipse Mosquitto

Es un intermediario(**broker**) de mensajes de código abierto.

Implementa el protocolo MQTT en las versiones:

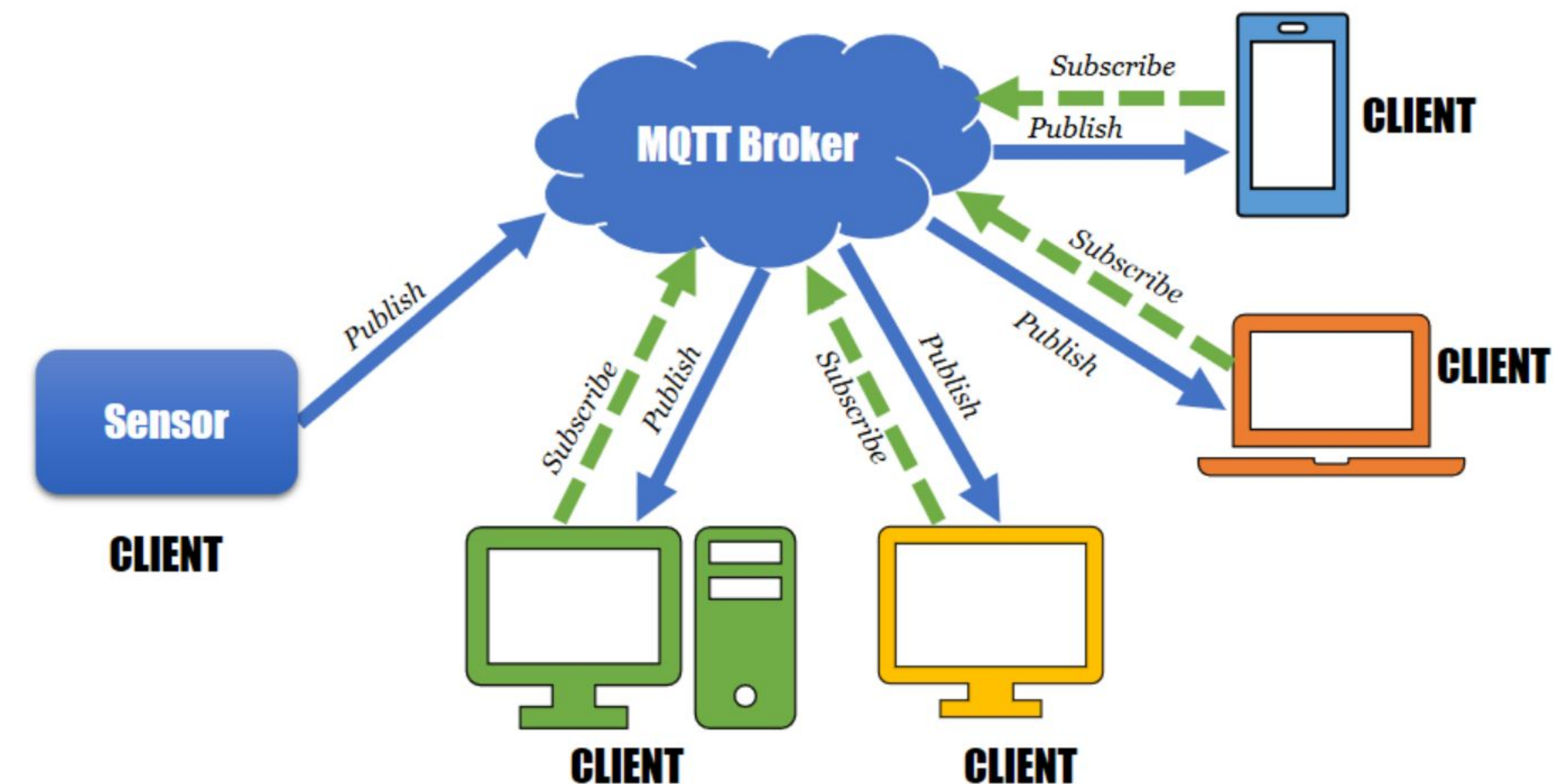
- 5.0
- 3.1.1
- 3.1

Mosquitto es **liviano** y es adecuado para su uso en todos los dispositivos, desde computadores de placa única(**SBC**) de bajo consumo hasta servidores completos.



Conceptos Eclipse Mosquitto

- **Publicar/suscribir:** en un sistema de publicación y suscripción, un dispositivo puede publicar un mensaje sobre un tema o puede suscribirse a un tema en particular para recibir mensajes.
- **Mensajes:** son la información que desea intercambiar entre sus dispositivos. Esto puede ser un comando o datos.
- **Temas:** son la forma en que se registra el interés por los mensajes entrantes o dónde se desea publicar el mensaje.
- **Intermediario:** es el principal responsable de recibir todos los mensajes, filtrarlos, decidir quién está interesado en ellos y luego publicar el mensaje para todos los clientes suscritos.





PONTIFICIA UNIVERSIDAD
CATOLICA
DE VALPARAISO

Rabbit MQ

Investigar a nivel de grupo

The RabbitMQ logo, consisting of an orange icon of a rabbit's head and the text "RabbitMQ" in orange, with a trademark symbol (TM) at the end.

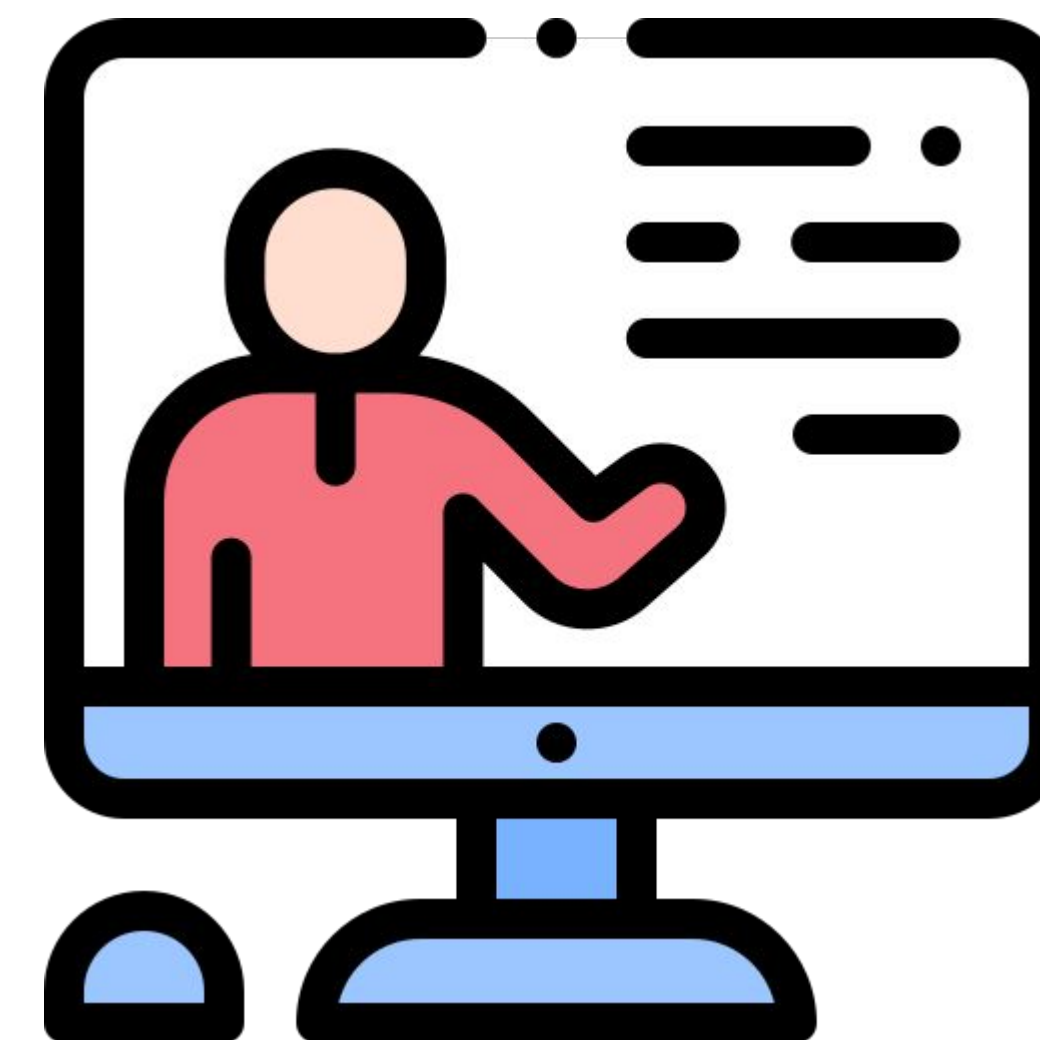
RabbitMQTM



Ejemplo 1

Utilizando Python y la librería **paho-mqtt**:

- Vamos a crear una aplicación que publique y suscriba mensajes en un servicio Mosquitto.
- Vamos a utilizar estos parámetros:
 - **broker** = "test.mosquitto.org"
 - **port** = 1883
 - **topic** = "pucv/iot/docente"





PONTIFICIA UNIVERSIDAD
CATOLICA
DE VALPARAISO





PONTIFICIA UNIVERSIDAD
CATOLICA
DE VALPARAISO

Trabajo Grupal



Grupos de trabajo P3

Grupo	Nombre	Apellido Paterno	Apellido Materno	Sala
1	CHANTAL ALEJANDRA	SALCEDO	BURGOS	1
	SERGIO ISMAEL	MORAN	VALDES	
	BRIAN LANTANO	GARCIA	SIMONTTI	
1	VALENTIN ANTONIO	SEPULVEDA	PIZARRO	2
	MIGUEL ANTONIO	SOTO	VEGA	
	GABRIEL ALEJANDRO BENITO	SAAVEDRA	RAMIREZ	
	ANDRES ESTEBAN	ORELLANA	RIQUELME	
2	ARLETTE IVONNE	ARAYA	ARAYA	3
	HECTOR EDUARDO	MEZA	SALGADO	
	ALVARO RODRIGO	HENRIQUEZ	PACHECO	
2	VICTOR	REYES	DURAN	4
	JULIO CESAR	ESPARZA	HENRIQUEZ	
	RENATO HUMBERTO	VARGAS	DIAZ	
	ALEX EDUARDO	VEGA	PROVOSTE	
3	GUSTAVO ADOLFO	CORDOVA	CATALAN	5
	CHRISTIAN DANIEL	SILVA	BARRERA	
	ITALO IVAN	IBACACHE	VARGAS	
3	ADRIAN MICHEL	TAPIA	COLL	6
	VICTOR ANDRES	ESPINOZA	MONARES	
	JOSE LUIS	MENDEZ	VASQUEZ	

Grupo	Nombre	Apellido Paterno	Apellido Materno	Sala
4	FELIPE EDUARDO ALEXIS	ARCOS	ZAMORA	7
	DAVID ALEXANDRO	GIRALT	CHACANA	
	RICARDO ANTONIO	PEZO	FRIAS	
4	ROGELIO EDGARDO	MORENO	ARANEDA	8
	MARIO ERNESTO	VILLANUEVA	GUTIERREZ	
	RUBEN ANTONIO	CONTRERAS	OTAIZA	
5	JAVIER ANTONIO	SILVA	RUIZ	9
	CARLOS IGNACIO	VALDERRAMA	GUERRA	
	CARLOS ALBERTO	ASTUDILLO	VASQUEZ	
5	JOSUE ALEXANDER	ISTURIZ	PEREZ	10
	ADOLFO ALBERTO	BRAVO	SILVA	
	KENY OSCAR	CORTES	GONZALEZ	
6	CLAUDIO ANDRES	JERALDO	PASTEN	11
	JOSE DANIEL	CASTRO	CERDA	
	HUMBERTO ALEJANDRO	CHAVEZ	PINO	
6	PATRICIO ERNESTO	MORENO	TORRES	12
	ARIEL FRANCISCO	MARTINEZ	GAVILAN	
	LUIS HUGO	BONILLA	YANEZ	

Ejercicio 1

En Salas de Grupo:

- En el **dispositivo** Wemos deben publicar en un mensaje **MQTT** la temperatura ambiente con el sensor DS18B20 cada 1 segundo.
- En **Python** deben crear una aplicación que muestre en consola, a modo de resumen, la cantidad de mensajes recibidos, la temperatura promedio, mínima y máxima, actualizada al último mensaje recibido (tiempo real).
- Vamos a utilizar estos parámetros:
 - **broker** = "test.mosquitto.org"
 - **port** = 1883
 - **topic** = "pucv/iot/m6/pX/gY" *#reemplazar X e Y por paralelo y sala correspondiente*



Feedback



PONTIFICIA UNIVERSIDAD
CATOLICA
DE VALPARAISO





PONTIFICIA UNIVERSIDAD
CATOLICA
DE VALPARAISO

Cierre

En esta clase vimos:

1. Distintos servicios de mensajería utilizados en IoT.
2. Aprendimos, ejercitando en grupo, cómo utilizar el servicio de Mosquitto.



Bibliografía

- Mosquitto documentación oficial:
 - <https://mosquitto.org/documentation/>
- RabbitMQ
 - <https://www.rabbitmq.com/documentation.html>