TECNICATURA SUPERIOR EN

Telecomunicaciones

Arquitectura y Conectividad

MÓDULO III: Arquitectura de Red IoT

ISPC
INSTITUTO SUPERIOR POLITÉCNICO CÓRDOBA

Dirección General de EDUCACIÓN TÉCNICA Y FORMACIÓN PROFESIONAL
Ministerio de EDUCACIÓN
CÓRDOBA entre todos

Arquitectura de Red IoT

¿Qué es una Arquitectura de red IoT?

El diseño de la arquitectura de red para una solución de IoT puede variar dependiendo de los requisitos y las necesidades específicas de cada caso. A continuación, se presenta una estructura básica para el diseño de la arquitectura de red en una solución de IoT:

1) **Dispositivos IoT:** En el nivel más bajo de la arquitectura se encuentran los dispositivos IoT, que pueden ser sensores, actuadores u otros dispositivos conectados. Estos dispositivos recopilan datos o envían comandos a otros dispositivos o sistemas.

2) **Capa de conectividad:** Esta capa se encarga de establecer y gestionar la conectividad entre los dispositivos IoT y la infraestructura de red. Aquí se seleccionan las tecnologías de conectividad adecuadas, como Wi-Fi, Bluetooth, Zigbee, LoRaWAN o 5G, según los requisitos de la solución. También se definen los protocolos de comunicación y los estándares a utilizar.

3) **Puertas de enlace (Gateways):** Las puertas de enlace actúan como intermediarios entre los dispositivos IoT y la infraestructura de red. Estas puertas de enlace pueden realizar funciones como la traducción de protocolos, la filtración de datos, el enrutamiento y la seguridad. Además, pueden almacenar y procesar datos localmente antes de enviarlos a la nube o a otros sistemas.

Arquitectura de Red IoT

4) **Nube (Cloud):** En esta capa, los datos recopilados por los dispositivos IoT se almacenan y se procesan en plataformas de nube. La nube ofrece una infraestructura escalable y flexible para gestionar grandes volúmenes de datos y ejecutar aplicaciones analíticas o de aprendizaje automático. Aquí se pueden realizar análisis, visualizaciones, aplicaciones de inteligencia artificial y almacenamiento de datos a largo plazo.

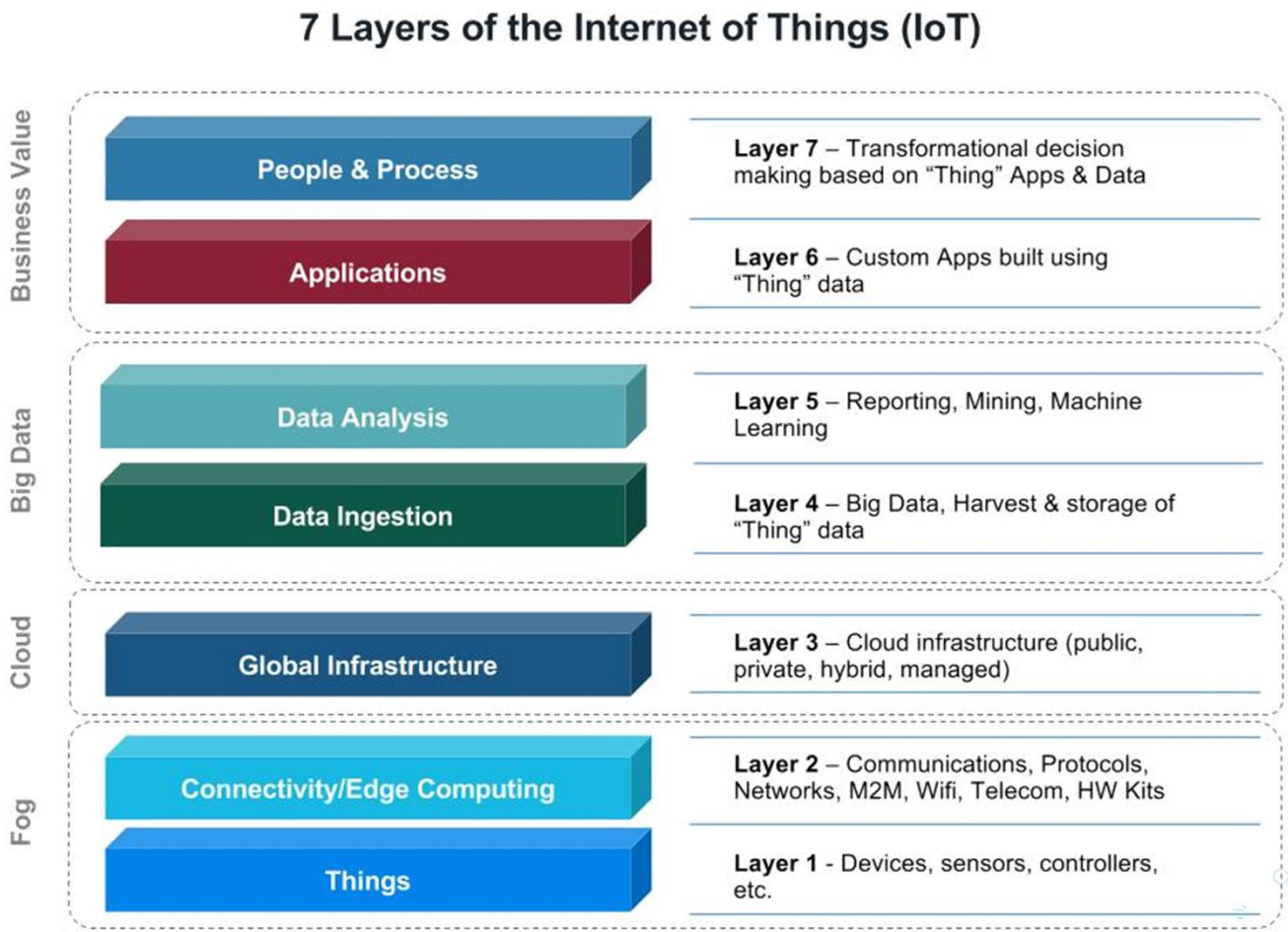
5) **Plataforma de gestión IoT:** Esta capa proporciona una plataforma centralizada para gestionar los dispositivos IoT, la conectividad y los datos. Aquí se pueden realizar tareas como el registro y autenticación de dispositivos, la gestión del ciclo de vida de los dispositivos, el monitoreo y control remoto, la gestión de datos y el análisis de datos en tiempo real.

6) **Aplicaciones y servicios:** En la capa superior se encuentran las aplicaciones y servicios que utilizan los datos recopilados por los dispositivos IoT y procesados en la nube. Estas aplicaciones pueden incluir paneles de control, visualizaciones de datos, sistemas de alerta, aplicaciones móviles, integraciones con otros sistemas empresariales, entre otros.

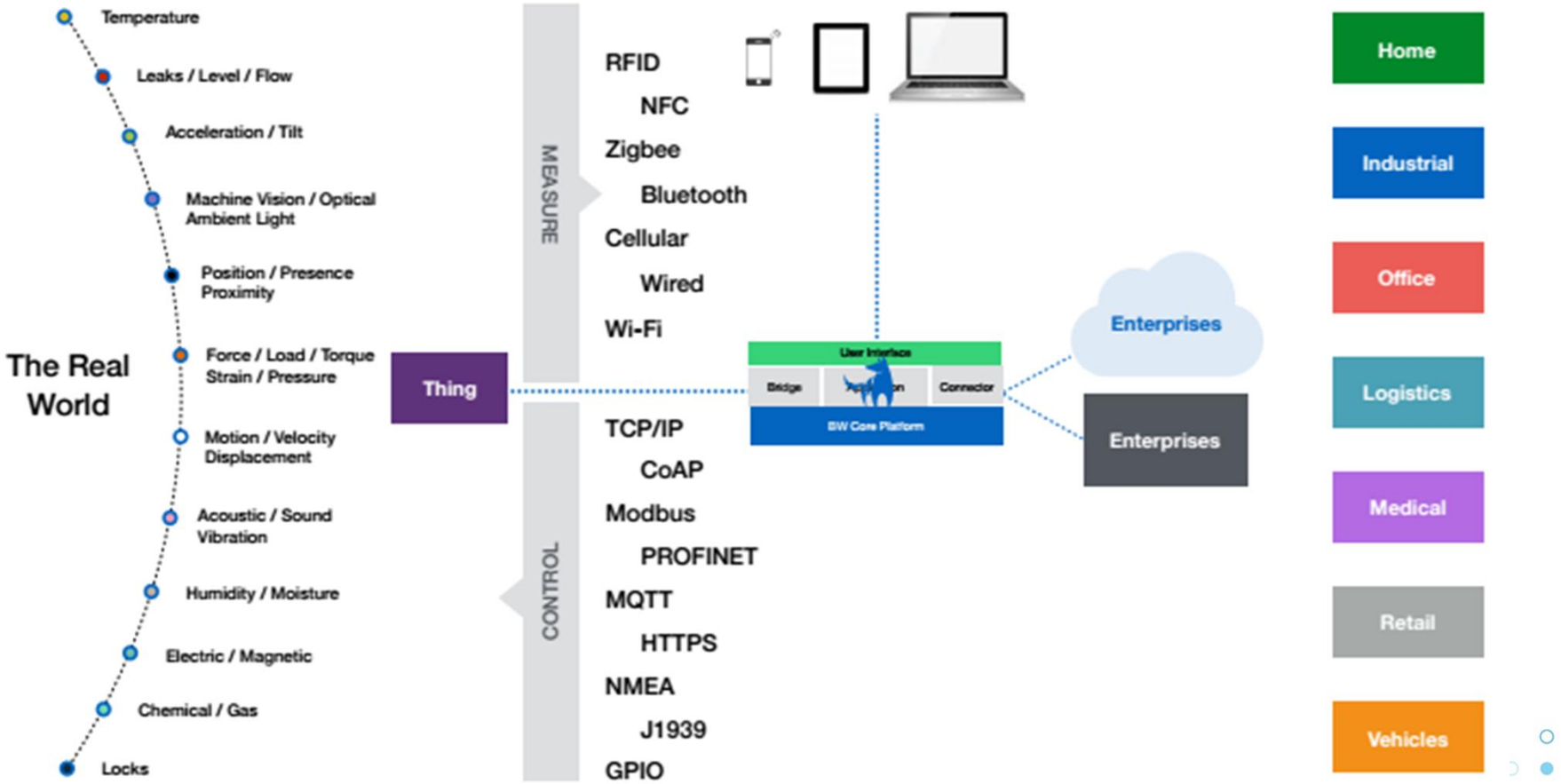
7) **Seguridad:** A lo largo de todas las capas de la arquitectura, es crucial considerar la seguridad de extremo a extremo. Esto implica implementar medidas de seguridad como autenticación, autorización, cifrado de datos, detección y prevención de intrusiones, gestión de claves y certificados, y seguimiento de eventos de seguridad.

Arquitectura de Red IoT

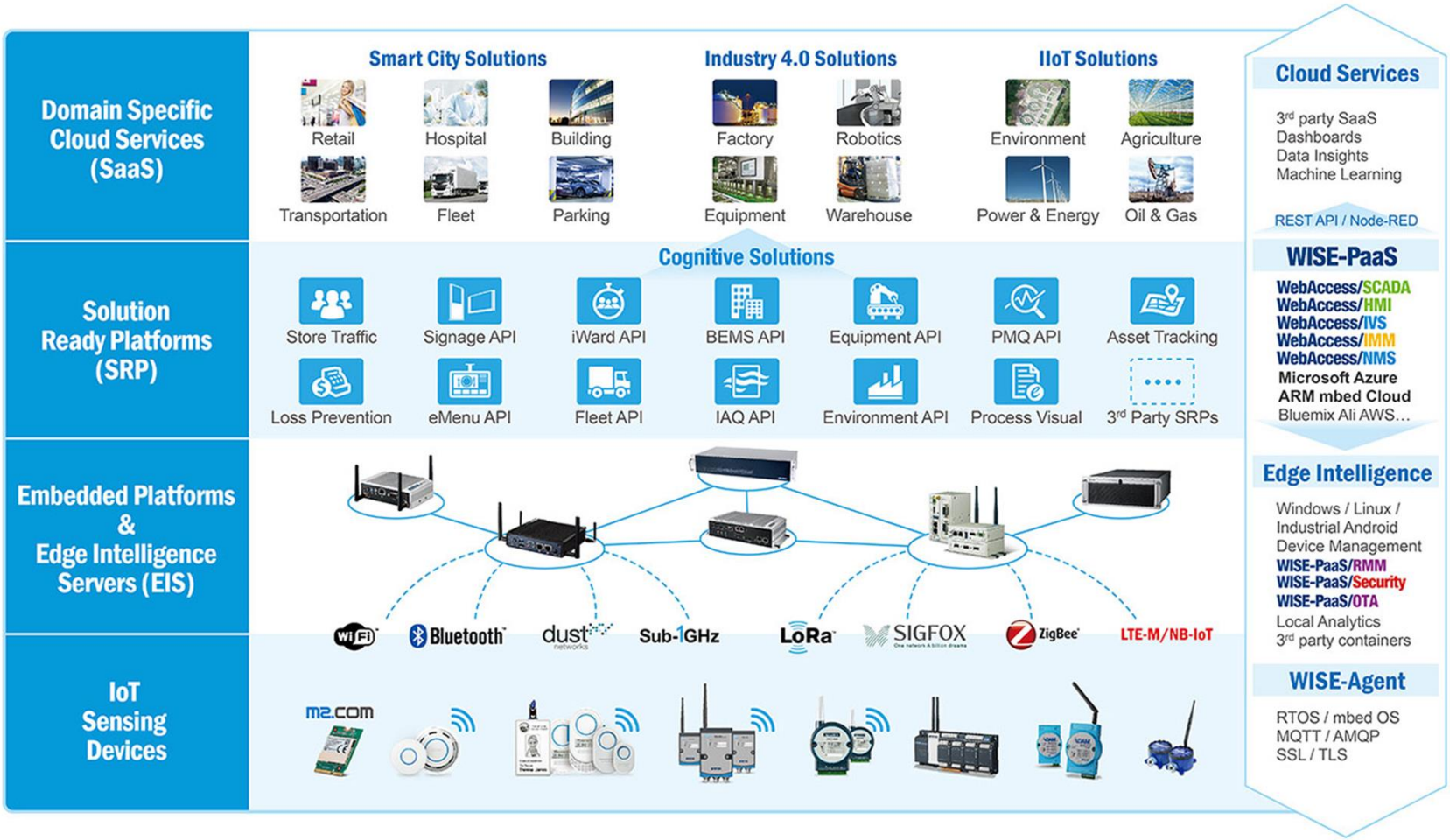
Es importante tener en cuenta que esta arquitectura puede adaptarse y personalizarse según los requisitos específicos de cada solución de IoT. También es fundamental considerar aspectos como la escalabilidad, la disponibilidad, la redundancia y los requisitos de latencia en el diseño de la arquitectura de red de IoT.



Arquitectura de Red IoT



Advantech IoT Solution Architecture



Protocolo COAP

¿Qué es el Protocolo COAP?

El protocolo COAP (Constrained Application Protocol) es un protocolo de aplicación diseñado para redes de dispositivos con recursos limitados, como sensores y actuadores en el Internet de las cosas (IoT, por sus siglas en inglés). Fue desarrollado para proporcionar una comunicación eficiente y confiable en entornos de baja potencia y baja capacidad de red.

COAP se basa en el protocolo de transferencia de hipertexto (HTTP) y utiliza el modelo de solicitud-respuesta similar al HTTP. Sin embargo, COAP está optimizado para redes con restricciones de ancho de banda y energía, y utiliza un encabezado más compacto y un conjunto de métodos más liviano en comparación con HTTP.

Algunas características clave de COAP son:

- **Eficiencia:** COAP está diseñado para funcionar en redes con ancho de banda limitado y recursos de energía, lo que lo hace adecuado para dispositivos IoT que operan con batería o en redes de baja capacidad.
- **Baja sobrecarga:** Utiliza encabezados compactos y opciones de mensajes para reducir la sobrecarga de datos en las comunicaciones.
- **Soporte para multicast:** COAP permite enviar mensajes a múltiples dispositivos en un solo envío, lo que lo hace adecuado para aplicaciones de IoT donde es necesario enviar información a varios destinos.
- **Seguridad:** COAP ofrece opciones de seguridad para garantizar la confidencialidad e integridad de los datos transmitidos. Puede utilizar el protocolo Datagram Transport Layer Security (DTLS) para la protección de datos.

Protocolo COAP

Un ejemplo sencillo de cómo se podría utilizar el protocolo COAP en una comunicación entre un cliente y un servidor:

- Configuración:
  - Dirección IP del cliente: 192.168.1.100
  - Dirección IP del servidor: 192.168.1.200
  - Puerto del servidor COAP: 5683
- Cliente envía una solicitud GET al servidor:
  - Dirección de solicitud: coap://192.168.1.200:5683/sensor/temperature
  - Tipo de método: GET
- El cliente envía una solicitud GET para obtener la temperatura del sensor al servidor COAP en la dirección IP 192.168.1.200 y puerto 5683. La ruta de la solicitud es "/sensor/temperature".
- Servidor responde con la temperatura:
  - Código de respuesta: 2.05 Content
  - Tipo de contenido: application/json
  - Cuerpo de la respuesta: {"temperature": 25.5}
- El servidor responde con un código de respuesta 2.05 Content, lo que indica que la solicitud se procesó correctamente. El tipo de contenido de la respuesta es application/json, y el cuerpo de la respuesta contiene el valor de la temperatura, que es 25.5 grados Celsius.

Protocolo AMQP

¿Qué es el Protocolo AMQP?

El protocolo AMQP (Advanced Message Queuing Protocol) es un protocolo de red estándar y abierto diseñado para la mensajería avanzada y el enrutamiento de mensajes entre aplicaciones. Fue desarrollado para mejorar la interoperabilidad y la eficiencia en los sistemas de mensajería empresarial.

AMQP proporciona un conjunto de reglas y formatos para el intercambio de mensajes entre diferentes componentes de software a través de una red. Permite la comunicación confiable, segura y eficiente entre aplicaciones distribuidas y servicios.

Algunas características clave del protocolo AMQP son:

1) **Mensajería orientada a colas:** AMQP se basa en el concepto de colas de mensajes, donde los productores (emisores) envían mensajes a una cola y los consumidores (receptores) reciben y procesan los mensajes de la cola. Esto permite una comunicación asíncrona y desacoplada entre aplicaciones.

2) **Enrutamiento flexible:** AMQP admite un enrutamiento de mensajes sofisticado, lo que significa que los mensajes pueden dirigirse a destinos específicos basados en reglas y criterios definidos. Esto permite una distribución de mensajes eficiente y adaptada a escenarios complejos.

Protocolo AMQP

3) **Confianza y entrega garantizada:** El protocolo AMQP proporciona mecanismos para garantizar la entrega confiable de mensajes incluso en entornos de red no confiables. Los mensajes pueden ser confirmados y se pueden implementar estrategias de recuperación en caso de fallas.

4) **Modelos de calidad de servicio (QoS):** AMQP ofrece diferentes niveles de calidad de servicio para adaptarse a las necesidades de las aplicaciones. Puede manejar garantías de entrega, confirmaciones, control de flujo y otras políticas para equilibrar la carga y la eficiencia en la comunicación.

5) **Interoperabilidad:** El protocolo AMQP es estándar y abierto, lo que significa que es independiente de cualquier proveedor o plataforma específica. Esto permite que las aplicaciones implementadas en diferentes lenguajes de programación y en diferentes sistemas operativos se comuniquen entre sí.

En resumen, el protocolo AMQP es un estándar para la mensajería empresarial y el enrutamiento de mensajes entre aplicaciones distribuidas. Proporciona una comunicación confiable y eficiente, permitiendo la entrega garantizada de mensajes, enrutamiento flexible y interoperabilidad entre sistemas heterogéneos.

Protocolo AMQP

Un ejemplo básico de cómo se podría utilizar el protocolo AMQP para enviar y recibir mensajes entre un productor y un consumidor:

a) Configuración:
   - Cola de mensajes: "mi_tren"
   - Dirección del servidor AMQP: amqp://mi_servidor_amqp
   - Credenciales de autenticación: usuario:contraseña

b) Productor envía un mensaje a la cola:
   - Mensaje: "Hola, esto es un mensaje de ejemplo“

El productor establece una conexión con el servidor AMQP utilizando la dirección "amqp://mi_servidor_amqp" y las credenciales de autenticación proporcionadas. Luego, envía el mensaje "Hola, esto es un mensaje de ejemplo" a la cola "mi_tren".

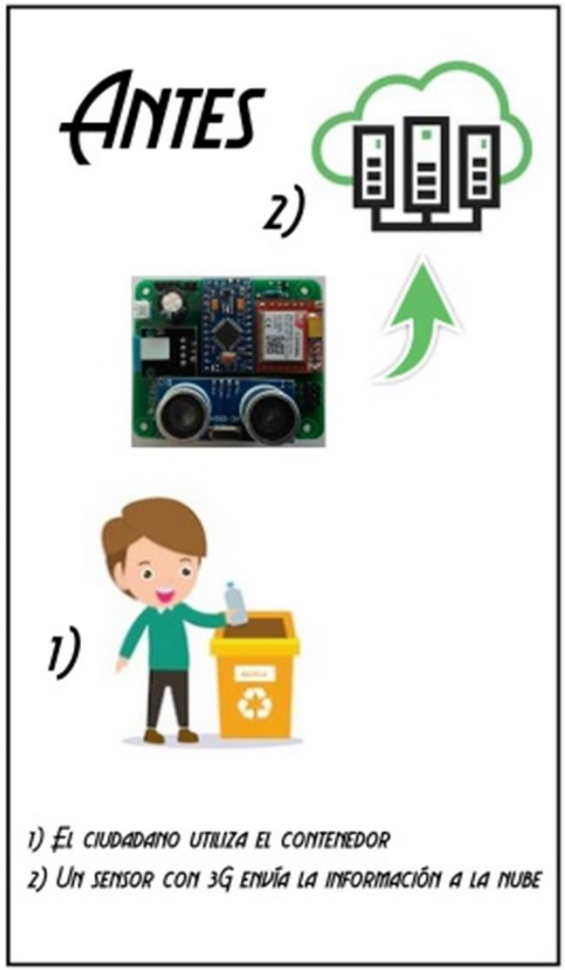
c) Consumidor recibe y procesa el mensaje de la cola:
   - Suscripción a la cola: "mi_tren"

El consumidor también establece una conexión con el servidor AMQP utilizando la misma dirección y credenciales. Luego, se suscribe a la cola "mi_tren" para recibir los mensajes. Cuando un mensaje llega a la cola, el consumidor lo recibe y lo procesa según sus necesidades.

Protocolos Modernos

Sin información relativa al ciudadano

Costo de comunicaciones por cuenta del gestor.



Incluye identificación del ciudadano

Permite implementar mecanismos de recompensa.

Costo de comunicaciones por cuenta del ciudadano.

¡Muchas gracias!

ISPC
INSTITUTO SUPERIOR POLITÉCNICO CÓRDOBA

Dirección General de
EDUCACIÓN TÉCNICA Y
FORMACIÓN PROFESIONAL

Ministerio de
EDUCACIÓN

CÓRDOBA
entre todos