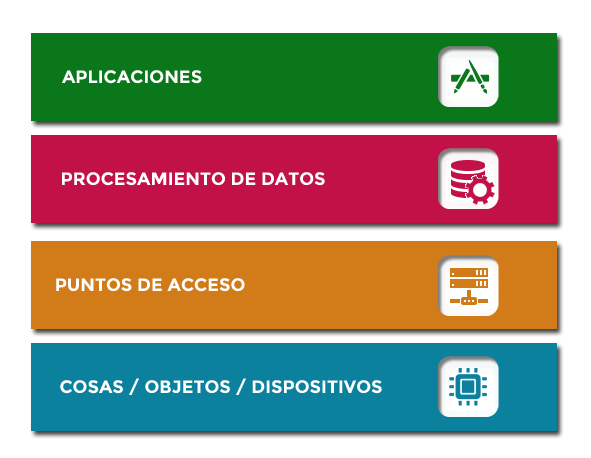
**Arquitectura de Sistemas IoT:**



CONECTIVIDAD Y COMUNICACIÓN

HTTP puede ofrecer una conectividad unificada y uniforme incluso en controladores de 8-bits para crear solicitudes de GET y POST. Pero la sobrecarga de este y otros protocolos tradicionales pueden ser un problema por dos razones principales. **“Los protocolos tradicionales, como HTTP, pueden suponer un problema para los dispositivos IoT”**. HTTP tiene altos requerimientos de energía. Para solventar esta problemática necesitaremos un protocolo binario pequeño y simple con habilidad para superar firewalls.

GESTIÓN DE DISPOSITIVOS

Cada vez son más los dispositivos que controlamos de manera activa (como ordenadores o móviles) y lo más probable, y adecuado, es que los instrumentos IoT también sigan esta tendencia. En la siguiente tenemos algunos ejemplos de qué nos permitiría un dispositivo gestionado remotamente

* La posibilidad de desconectar un dispositivo robado
* La habilidad de actualizar el software de un dispositivo
* La actualización de credenciales de seguridad
* Autorizar o denegar algunas capacidades del hardware remotamente
* Localizar dispositivos perdidos
* Limpiar información confidencial de un dispositivo robado
* Reconfigurar parámetros de Wi-Fi, GPRS u otras redes remotamente.

RECOLECCIÓN, ANÁLISIS Y ACTUACIÓN DE LOS DATOS

Ciertos dispositivos tienen algún tipo de interfaz de usuario (UI), pero en general los dispositivos IoT están centrados en ofrecer sensores, actuadores o una combinación de ambos. El requerimiento principal del sistema es que podamos recolectar datos e información desde una multitud de dispositivos y poder almacenarla, analizarla y actuar sobre esta información.

**“El requerimiento principal del sistema es que podamos recolectar y gestionar datos”**

ESCALABILIDAD

Un requerimiento importante para una arquitectura es que pueda ser escalada y que pueda ir de soportar una integración pequeña a soportar un gran número de dispositivos. Es esencial contar con una escalabilidad elástica y la habilidad de desarrollar en una infraestructura Cloud.

SEGURIDAD

En muchas ocasiones los dispositivos IoT recolectan un gran número de información personal. Por su función de llevar el mundo real a internet (y viceversa), esto ubica dentro de tres categorías de riesgo:

* Riesgos inherentes de cualquier sistema de internet pero que los diseñadores IoT o de producto no tengan consciencia de ellos.
* Riesgos específicos de los dispositivos IoT
* Seguridad para cerciorarnos de que no se causan daños por, por ejemplo, por el mal uso de los actuadores.

Nuestros requerimientos de seguridad deben de soportar;

* Encriptación en aquellos dispositivos que son lo suficiente potentes.
* Un modelo de identidades basado en tokens moderno y no una contraseña/usuario
* Una gestión de tokens y accesos lo más remota y fácil posible; y
* Control de accesos basados en políticas y con gestión de usuarios para sistemas basados en XACML.

**Elementos de sistemas IoT:**

Nuestra arquitectura de IoT contiene los siguientes componentes:

* **Cosas** equipadas con sensores para recoger datos y **actuadores** para realizar comandos recibidos desde la nube.
* **Gateways** para filtrar, preprocesar y mover datos a la nube y viceversa, – recibir comandos desde la nube.
* **Pasarelas en nube (Cloud Gateways)** para garantizar la transición de datos entre las pasarelas sobre el terreno y los servidores centrales de IoT.
* **Procesadores de datos en tiempo real** para distribuir los datos procedentes de los sensores entre los componentes de las soluciones de IoT pertinentes.
* **Bases de Datos** para almacenar todos los datos de valor definido e indefinido.
* **Big Data Warehouse** para la recogida de datos valiosos.
* **Aplicaciones de control** para enviar comandos a los actuadores.
* **Machine Learning** para generar los modelos que luego son utilizados por las aplicaciones de control.
* **Aplicaciones de usuario** que permiten a los usuarios monitorizar el control de sus cosas conectadas.
* **Análisis de datos** para el procesamiento manual de datos.

