



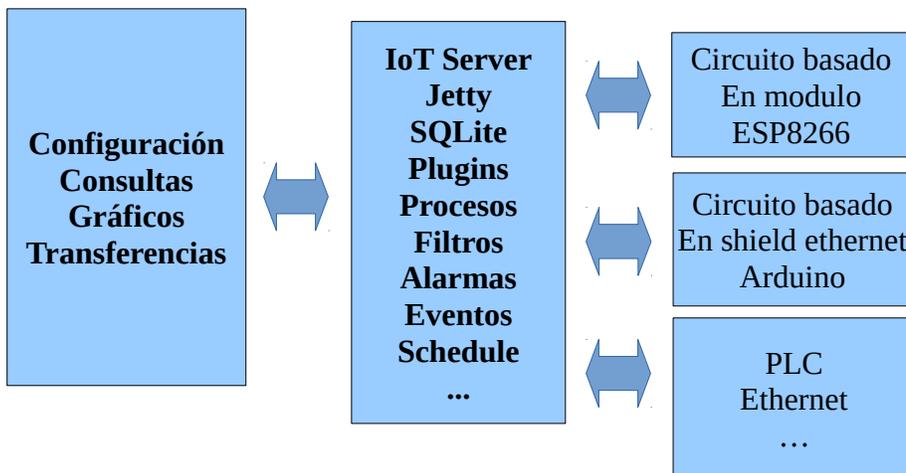
## Práctica Profesional Trabajo Práctico: Desarrollo de Servidor IoT version 1.0, Mayo 2016

**Objetivo:** desarrollar una aplicación web en java EE que permita una simple forma de administrar, consultar y almacenar datos provenientes de circuitos IoT..

**Problema:** En el contexto de IoT (internet de las cosas), los circuitos tienen capacidad de transmitir valores de sensores a través de una red LAN/WAN para que un servicio web o servidor local permita el almacenamiento de dichos valores. El volumen de datos puede ser importante, imaginemos por ejemplo valores de presión o temperatura, transferidos en intervalos de tiempo muy cortos.

**Solución:** Se propone el desarrollo de una aplicación java SE que embeba en su interior un contenedor de servlets y servidor web Jetty. Como soporte de datos se propone utilizar una base de datos SQLite para soportar el volumen de datos que deberá soportar el servidor. La interfaz gráfica, consultas, etc se realizarán utilizando JQuery, Jtable, JQueryUI, etc.

### Diseño:



### Tipos de circuitos

Se identifican 3 tipos de circuitos, desde el punto de vista del servidor:

Circuitos de Entrada (I): circuitos que envían información desde el campo hacia el servidor.

Circuitos de Salida (O): el servidor envía mensaje al circuito.

Circuitos de Entrada y Salida (I/O): diálogo entre servidor y circuito, enviando y recibiendo mensajes.

## **Identificación de los circuitos**

Cada circuito se identificará con un TAG (id) único dentro del sistema, código alfanumérico que identificará al circuito (el mismo podrá incluir el código nemotécnico que desee el usuario). Todo circuito pertenece a un Área (sector del edificio o planta). Toda Área pertenece a un sitio. Básicamente, la BD almacenará TAG,FECHA-HORA,VALOR.

## **Configuración de circuitos**

- Configuración de urls para identificar circuitos y servlets para su procesamiento
- Activar / desactivar circuitos, áreas y sitios.
- Configuración de tags, tags calculados, etc.
- tag history: frecuencia, autoborrado de datos, etc.

## **Eventos**

- Inicio servidor
- Fin servidor
- Sitio: agregar área, quitar área, modificar área
- Área: agregar tag, borrar tag, modificar tag
- Arribo de datos
- Envío de datos
- Timer
- Disparo de Alarma: dato filtrado y válido pero fuera de rango permitido
- Filtro: arribo de dato inválido

## **Filtros**

- Validación de datos
- Ignorar datos no válidos
- Disparo de evento por valores no válidos

## **Alarma**

- Valores con rango mínimo, máximo
- Ausencia de valor por x tiempo
- Envío de alarma por e-mail, url, ejecución de código, etc

## **Schedule**

- Planificación de tareas periódicas dentro del servidor al estilo del cron de linux

## **Importación/Exportación de datos, transferencias de datos**

-Exportacion/importación masiva de datos planificada o a pedido, formato de datos html, xml, json, etc. configuración de tags a enviar y rango de valores, envio a disco, a servidor ftp, por correo electronico, a url/servicio web, etc. Permite replicar datos a otros servidores, etc.

Atte. Guillermo Cherencio  
Practica Profesional  
2016