

VII CONGRESO INTERNACIONAL DE COMPUTACIÓN Y TELECOMUNICACIONES

Lima - Perú

Modelo middleware de datos y operaciones para el diseño de objetos inteligentes aplicados a Internet de las cosas

**Autores: Antonio Román Villarroel
Luis Joyanes Aguilar
Alicia Ramos Sanchez
Juan Camargo**



Universidad
Inca Garcilaso de la Vega
Nuevos Tiempos. Nuevas Ideas

FACULTAD DE INGENIERIA DE SISTEMAS, CÓMPUTO Y TELECOMUNICACIONES



Planteamiento

- Desde que el termino fue presentado en 2009 Internet de las cosas (IoT - Internet of Things) ha estado avanzando y evolucionando en forma decidida para darle al mundo que nos rodea una representación virtual en la red [Ashton09] [Schoenberger02] .
- IoT como inicio ha tomado prestadas las funcionalidades y modos de operaciones e integración a la red de otras tecnologías ya conocidas, como lo son WSN y M2M [Kortuem10].



Planteamiento

- Por un lado, hay alternativas hardware simplificadas para IoT, muchas de ellas basadas en plataformas de desarrollo accesibles (Arduino, Raspberry Pi, Beaglebone, etc), así como desarrollos hechos a la medida [Kranz10].
- Por el lado del software, también ha proliferado múltiples opciones de soporte a estas plataformas, siendo algunas de pago y otras de libre uso. La suma de ambas busca acercar la tecnología al diseño de dispositivos enfocados a IoT [Floerkemeier08].



Justificación

- Un problema es la inevitable fragmentación tecnológica. Cada plataforma está basada en algún hardware específico y este cuenta con su propio conjunto de herramientas de desarrollo software [Fantacci14] [Moore15].
- La tendencia es hacia la integración, ya que la fragmentación tecnológica divide los esfuerzos para lograr un despliegue definitivo de IoT, lo cual nos lleva a observar que opciones hay para estandarizar un poco este entorno [Vermesan13] .



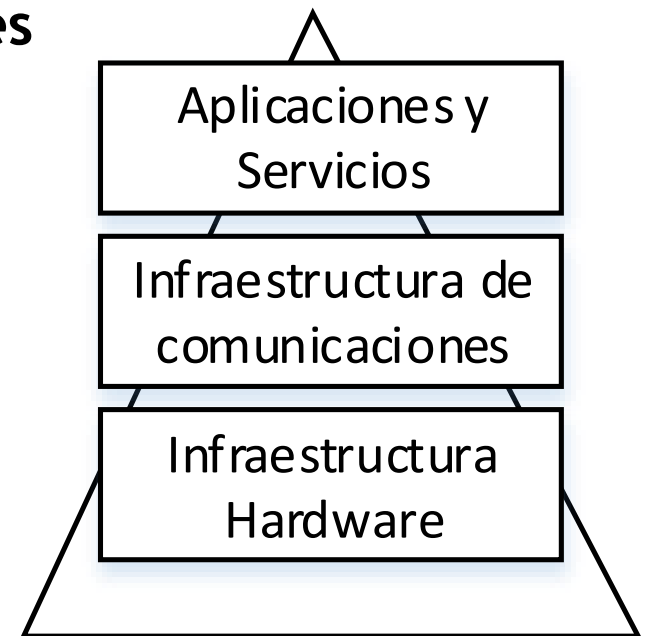
Objetos Inteligentes

- Un objeto inteligente orientado a IoT, debe cumplir tres requisitos básicos en su arquitectura [Atzori10]:

i) Infraestructura hardware

ii) Infraestructura de comunicaciones

iii) Aplicaciones y servicios





El problema...

- Primer problema: La infraestructura hardware es variable, y los otros dos aspectos (comunicaciones y servicios) son dependientes del hardware a nivel de codificación de rutinas software.
- Segundo problema: la forma en que se debe tener acceso a los periféricos hardware de la infraestructura, así como de su control y administración.
- Una posible solución al problema es la aplicación de una capa intermedia de software, un middleware [Aberer06] [Bandyopadhyay11] [Yingyou11].



La propuesta...

- Es cierto que existe una amplia variedad de middlewares, basados en algún sistema operativo y con diversos protocolos de comunicaciones [Teixeira11].
- Para un objeto inteligente de IoT se busca algo mas liviano que aproveche hardware mas económico.
- El middleware propuesto es una capa de software ligero lo suficientemente completa para ocultar los detalles del hardware y crear una capa de abstracción [Handziski05] que normalice en lo posible el acceso a los periféricos hardware, buscando la máxima portabilidad.



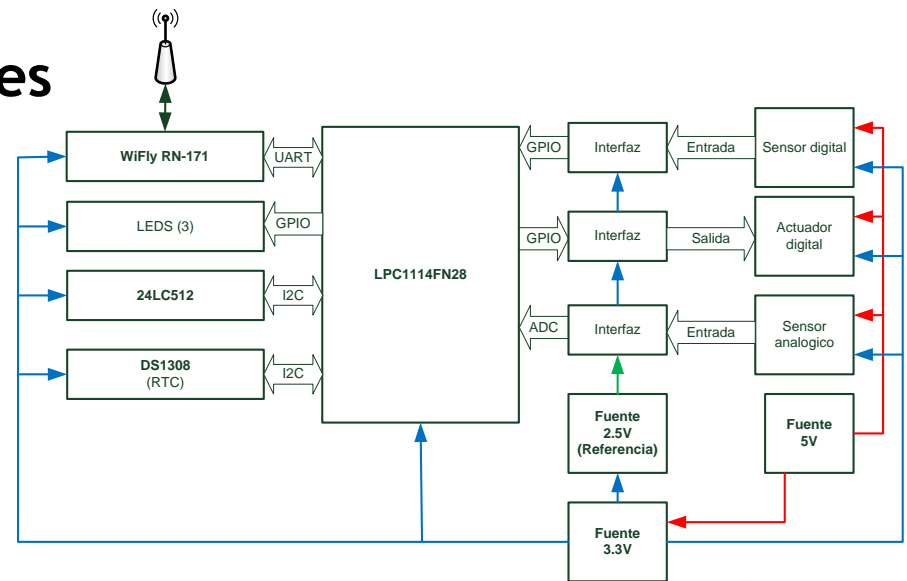
La propuesta...

- Este middleware se compone de dos capas.
- Una capa de abstracción para el acceso físico al hardware (HAL - Hardware Abstraction Layer), que es dependiente de la arquitectura seleccionada y debe ser adaptada y/o reescrita cada vez que se cambie esta.
- Una capa de interfaz de acceso (IAL - Interface Abstraction Layer) que haga las funciones de interface con las capas superiores. Esta última capa cumple funciones de traducción y oculta los detalles del hardware subyacente [Fortino14] [Kortuem10] [Sanchez12].



Modelo de operaciones

- A grandes rasgos, un OI se compone de:
 - a) sensores
 - b) actuadores
 - c) unidad de control
 - d) interfaz de comunicaciones





Modelo de operaciones

- En lo más básico, un OI debe estar preparado para:
 - i) adquirir datos (sensores)
 - ii) generar respuesta (actuadores)
 - iii) comunicar sus actividades.
- La combinación sensor/actuador en una plataforma hardware con enfoque a IoT es la primera línea de enlace o comunicación entre el mundo real y el mundo virtual. Los sensores adquieren datos sobre las variables físicas del entorno [Eisenhauer10].



Modelo de datos

- El interfaz de datos es una IAL, dependiente de la HAL del modelo de datos anteriormente definido, que ofrece un conjunto de funciones normalizadas para poder leer y escribir datos desde y hacia los sensores y actuadores respectivamente, entre otras funciones.





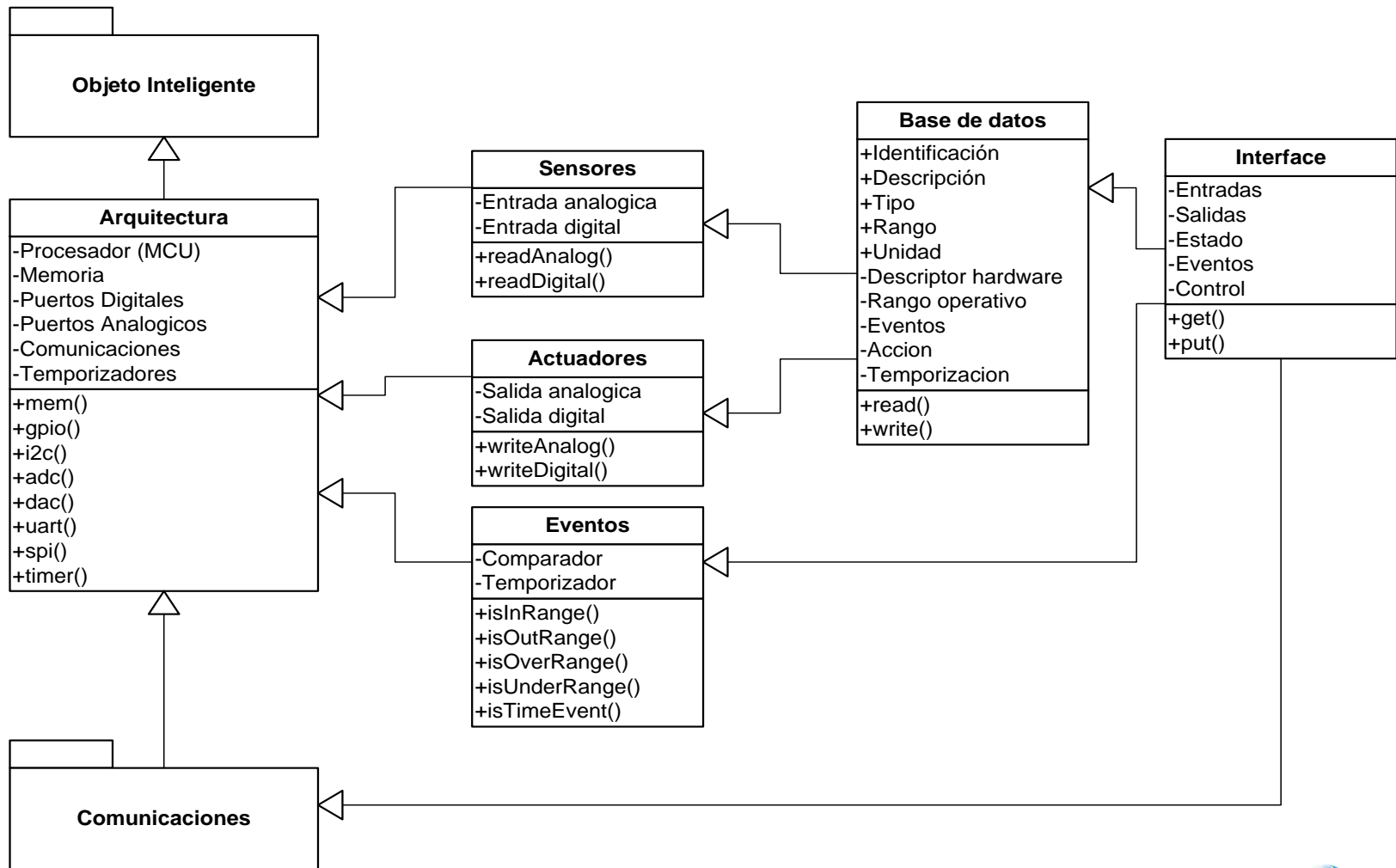
Modelo de datos

- El modelo para sensores y actuadores permite consultas y actualizaciones. Los atributos del objeto se manejan usando grupos de metadatos, que son tres: **operaciones**, **control** y **ubicación**. Estos datos son de tipo abstractos y son independientes de la arquitectura.





Arquitectura





Experimento

- Para probar la viabilidad de este modelo, se escogieron **dos plataformas hardware** con el mínimo de periféricos necesarios, de arquitectura ARM.
- Se seleccionaron **dos procesadores, NXP LPC11C14, y Freescale MKL25Z128VLK4**. Para ambos casos se usó como soporte base las librerías CMSIS (Cortex Microcontroller Software Interface Standard) proporcionadas por ARM y que cada fabricante adapta a sus respectivos diseños.

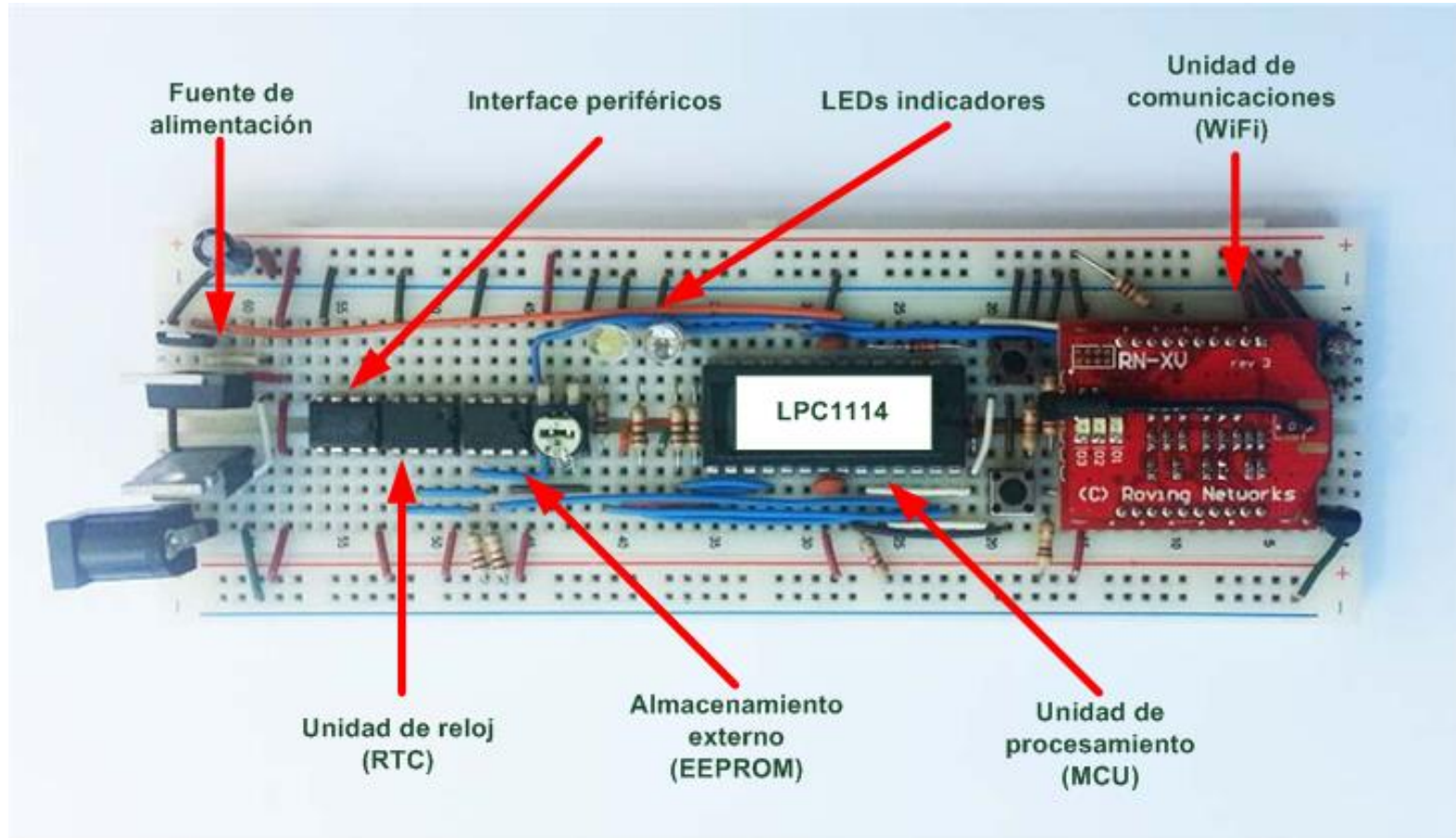


Experimento

- Se uso el mismo conjunto de perifericos hardware para cada plataforma, haciendo **la programación de todo el software en lenguaje C.**
- Finalmente, se implementó un servidor de websockets sobre un sistema **Raspberry Pi con Raspbian Linux**, mediante un script **Python** usando la librería Tornado. Para efectos de visualización e interacción, se escribió un pequeño sistema web usando **JavaScript** y **jQuery**, que actuaba como terminal de datos, funcionando sobre navegador web.



Experimento





Resultados

- La HAL requirió ser reescrita para cada arquitectura de tal forma que no se tuviera que tocar las capas por encima de la IAL. Para ambos casos, se implementó una FSM ajustada a los recursos de la arquitectura con menos prestaciones, y contiene las rutinas de administración y control de las funciones del sistema y se soporta completamente sobre la IAL. **A pesar de los contratiempos, la aplicación práctica del modelo es completamente viable.**



Resultados

The screenshot displays two windows. The top window is a web browser titled 'Terminal Websockets' at the URL 'http://192.168.10.200:80'. It shows a 'Conexion establecida.' message and a series of JSON messages received at specific times: 07:34:13, 07:34:33, 07:34:33, 07:34:49, and 07:34:50. The bottom window is a terminal titled '192.168.10.200 - PuTTY' showing the execution of 'ls -la' and 'python ws-server.py'. The terminal output includes file permissions, a connection log entry, and the same JSON messages seen in the browser.

```
root@raspberrypi:~/WS-Terminal# ls -la
total 16
drwxr-xr-x 2 root root 4096 sep 12 07:23 .
drwx----- 8 root root 4096 sep 12 07:22 ..
-rw-r--r-- 1 root root 2660 sep 12 07:23 index.html
-rw-r--r-- 1 root root 963 sep 12 07:23 ws-server.py
root@raspberrypi:~/WS-Terminal# python ws-server.py
[I 140912 07:28:14 web:1635] 304 GET / (192.168.10.195) 61.43ms
Nueva conexion abierta
Nueva conexion abierta
Mensaje entrante: {"status":"started","id":"444502B","device":{"name":"ST01","time":"12:00:06","time-zone":"GMT-0"}}
Mensaje entrante: {"get":"id"}
Mensaje entrante: {"status":"ok","id":"444502B","device":{"name":"ST01"}}
Mensaje entrante: {"get":"sysdata"}
Mensaje entrante: {"status":"ok","id":"444502B","device":{"name":"ST01","sysdata":{"alive":{"on":"1"},"sleep":{"on":"5"},"awake":{"on":"5"},"events":"on","actions":"on","status":"full","net-status":"on","verbose":"on","time-mode":"time"}}
```



Conclusiones

- Es posible crear un sistema middleware, que basado en el modelo de datos y en el modelo de operaciones propuestas, permita unificar los criterios básicos de control y acceso a las funciones y operaciones de una plataforma hardware que actúe como objeto inteligente (OI) para aplicaciones en Internet de las cosas (IoT).



Conclusiones

- A pesar de haber escogido dos procesadores de fabricantes distintos cuya base arquitectónica es la misma (ARM), hay diferencias muy difíciles de salvar para la creación de la HAL dado que cada fabricante introduce características propias en los periféricos.



Conclusiones

- Hubo que tomar algunas decisiones de compromiso, como solo usar las funciones básicas y descartar las funciones especiales. A pesar de estos detalles en el desarrollo de la capa HAL, las capas superiores dependientes de la IAL funcionaron perfectamente bien, con pocos cambios, y para los efectos del terminal de pruebas ambos prototipos funcionaban exactamente igual.



Trabajo futuro

- Como trabajo a futuro, queda pendiente la incorporación de un modelo de comunicaciones adaptable al medio físico usado para el enlace. Así mismo, es necesario incorporar mecanismos de control de acceso y posiblemente, encriptación de canal de datos, esto ya con miras la seguridad y uso controlado de los OI.



¡Gracias!