# CONTROLADOR CENTRAL PARA UN SISTEMA DOMÓTICO UTILIZANDO EL PROTOCOLO INALÁMBRICO ZIGBEE

# Bienvenidos!

## Trabajo de Fin de Grado

"Controlador central para un sistema domótico utilizando el protocolo inalámbrico ZigBee"


Por Enrique Hernández Bello

# Índice

# INTRODUCCIÓN

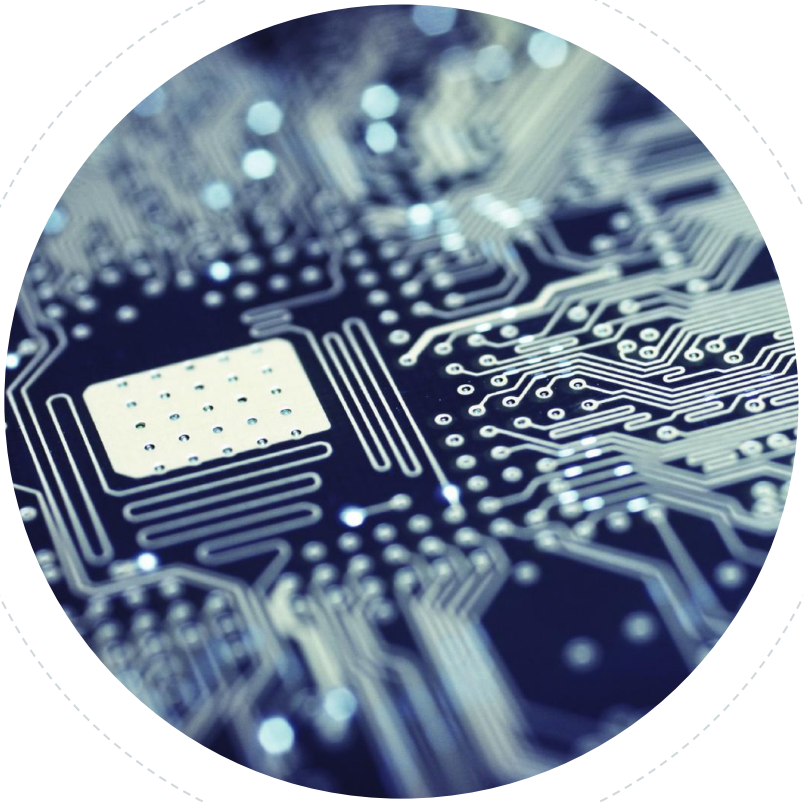
# 1.
# Antecedentes

- Internet of things
- Placas de desarrollo
- Protocolo ZigBee

## Definición

Se refiere a la interconexión digital de objetos cotidianos con **Internet**.

# Placas de desarrollo



- Bajo coste
- Filosofía libre
- Solución alternativa

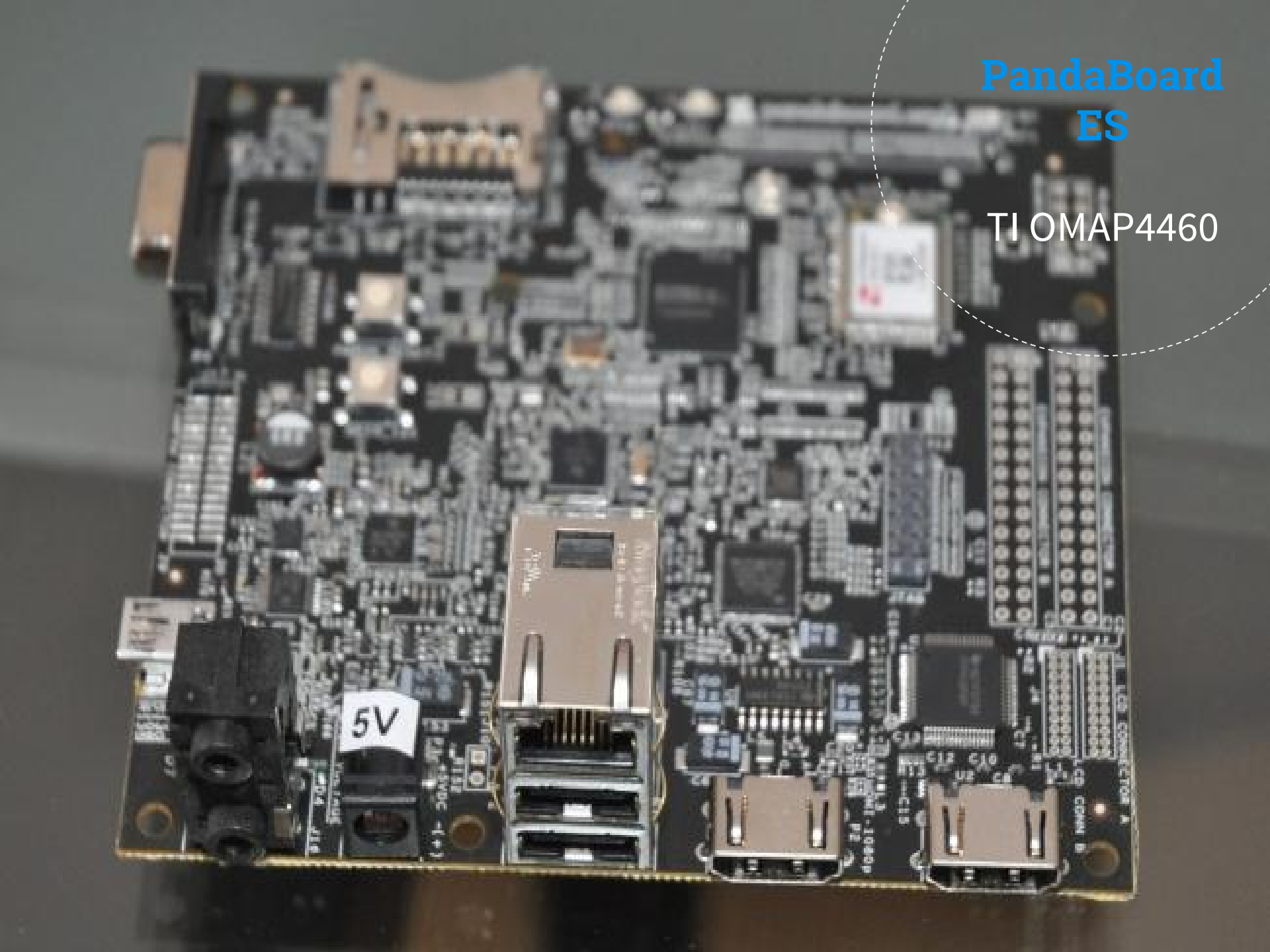
## Características

- Zigbee Alliance
- Basado en IEEE 802.15.2
- Banda ISM 2.4Ghz
- Bajo coste
- Topologías: árbol, estrella y malla

# Protocolo ZigBee

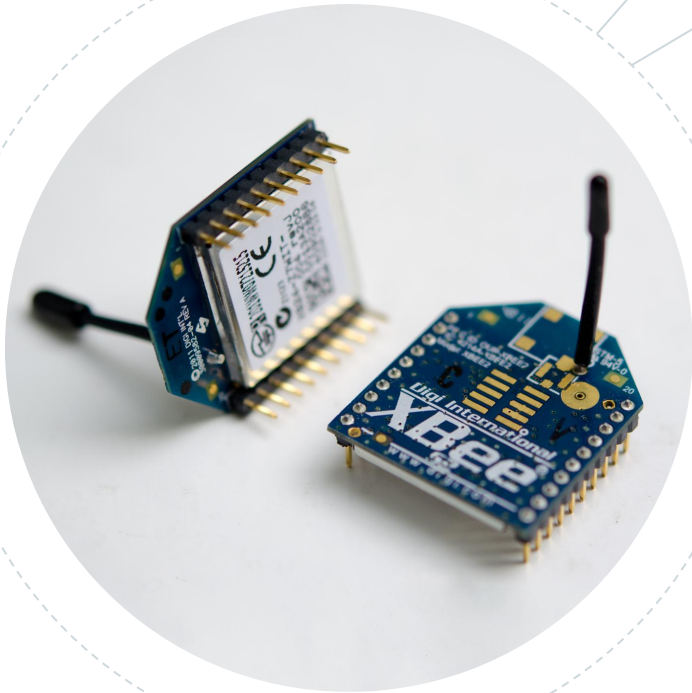
# 2.
# Elementos de partida

- PandaBoard ES
- Módulos XBee


PandaBoard
ES
TI OMAP4460
5V

## Módulos XBee

- Nodo **coordinador** y **nodo final**
- No se contemplan nodos tipo *router*
- Sin microcontrolador intermedio
- Preconfigurados
- Nodo final:
  - Botón
  - Led
  - Potenciómetro



# 3.
# Objetivos

- Descripción global
- Tabla resumen

# Descripción Global

# Resumen de objetivos

## Sistema Operativo

Instalación y configuración de GNU/Linux

## Conexión XBee

Conexión del emisor/receptor ZigBee y realización del driver que lo controle.

## Demonio

Desarrollo del programa de control haciendo uso del protocolo diseñado

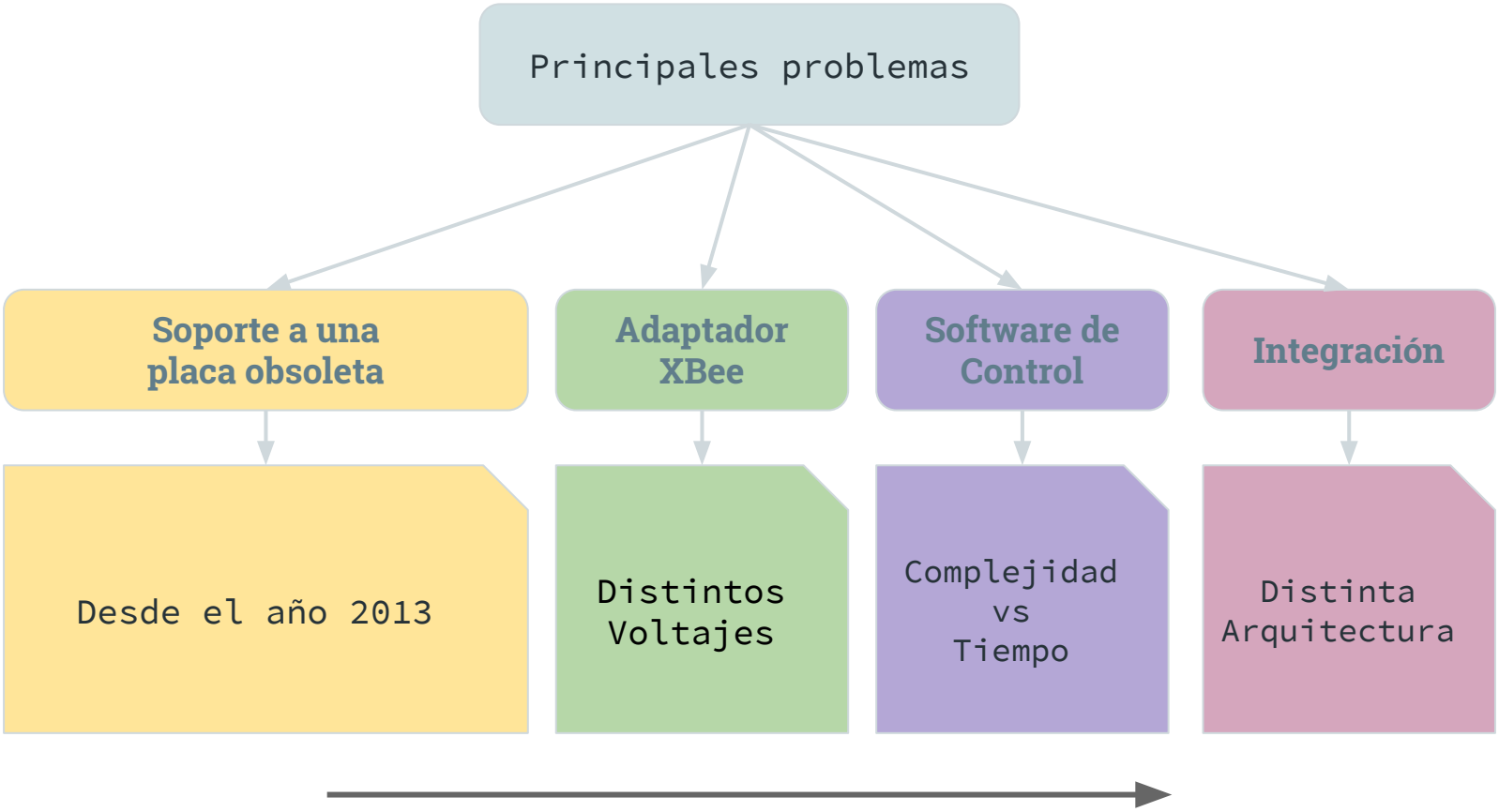
## Interfaz de usuario

Elección y desarrollo de al menos una interfaz de usuario propuesta p.ej.: Servidor Web, Control Remoto, Pantalla TFT o Salida HDMI

# ANÁLISIS

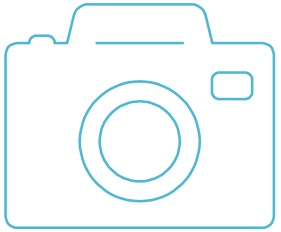

ANÁLISIS
Fases del proyecto
Observaciones
Decisiones tomadas
Presupuesto
INTRODUCCIÓN
Antecedentes
Elementos de partida
Objetivos
DESARROLLO
Sistema Operativo
Adaptador XBee
Interfaz de Usuario
Integración
CONCLUSIONES
Líneas a seguir
Repercusión en la comunidad
Conclusions

# Fases del proyecto

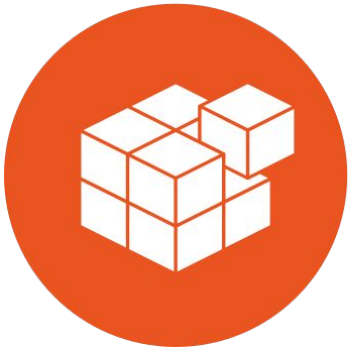


## Observaciones

- Funcionalidades genéricas
- Mayor acabado posible
- Perdurable en el tiempo
- Construcción y mantenimiento simple



# Decisiones tomadas



**Distribución**

Ubuntu Core

Contenedores

Snapcraft


**Adaptador**

DiY ready

Componentes THT

KiCad EDA

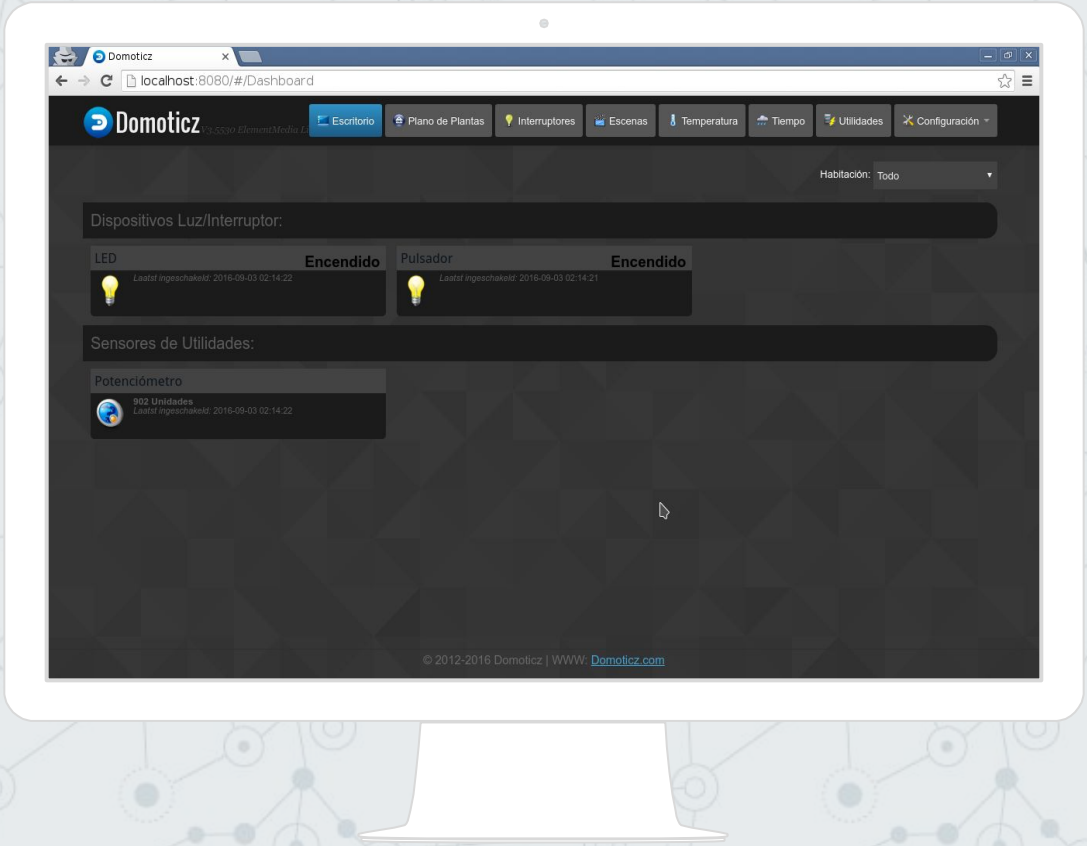
**Interfaz de usuario**

Interfaz web existente

Añadir soporte XBee

Se descarta OpenHAB

Se elige Domoticz

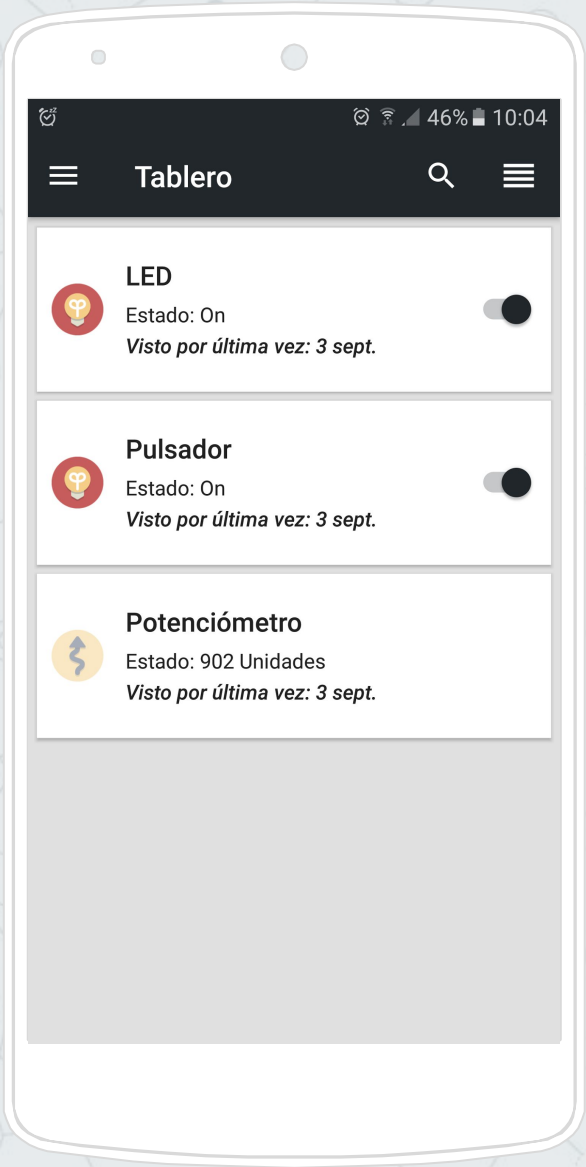



# Tablero de instrumentos

Pestaña de la interfaz web donde se recogen los dispositivos seleccionados como favoritos

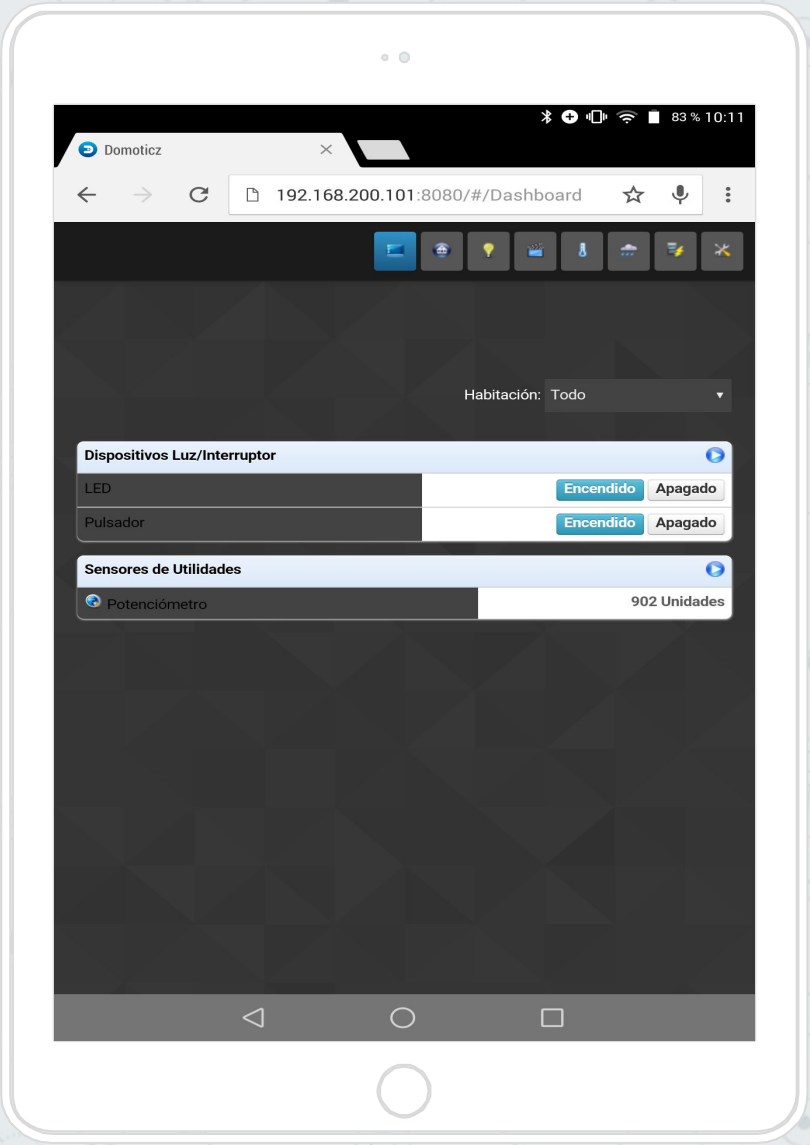
# Domoticz Lite

Tablero de instrumentos en la aplicación para dispositivos Android



# Navegador en Tablet

Estilo móvil de la interfaz web en una Tablet



# Presupuesto

| | Horas | Precio/Hora | Subtotal |
|---|---|---|---|
| Desarrollo | **420** | **10€** | **4200€** |
| Puesta en marcha | **20** | **10€** | **200€** |
| Hardware | | | **254,90€** |
| | | TOTAL | **4654,90€** |

# DESARROLLO

**INTRODUCCIÓN**

Antecedentes

Elementos de partida

Objetivos

**ANÁLISIS**

Fases del proyecto

Observaciones

Decisiones tomadas

Presupuesto

**DESARROLLO**

Sistema Operativo

Adaptador XBee

Interfaz de Usuario

Integración

**CONCLUSIONES**

Líneas a seguir

Repercusión en la comunidad

Conclusions

# 1.
# Sistema Operativo

- Kernel snap
- Gadget snap
- Imagen distribuíble
- Problemas

# Kernel snap

```yaml
name: linux-panda
version: 4.4.19
summary: A PandaBoard kernel built from source
description: This is the reference kernel for PandaBoard and PandaBoard ES
type: kernel
grade: stable
confinement: devmode
architectures: [armhf]

parts:
    kernel:
        plugin: kernel
        source: https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux-stable.git
        source-tag: v4.4.19
        source-type: git
        kdefconfig: [omap2plus_defconfig]
        kconfigs:
            - CONFIG_LOCALVERSION="-panda"
            - CONFIG_DEBUG_INFO=y
            - CONFIG_SQUASHFS=y
            - CONFIG_SQUASHFS_XZ=y
            - CONFIG_DEVPTS_MULTIPLE_INSTANCES=y
            - CONFIG_FB_PRE_INIT_FB=y
            - CONFIG_FB_OMAP2=y
            - CONFIG_RTC_DRV_TWL4030=y
            - CONFIG_USB=y
            - CONFIG_USB_KBD=y
            - CONFIG_USB_MOUSE=y
            - CONFIG_USB_OHCI_HCD=y
            - CONFIG_USB_EHCI_HCD=y
            - CONFIG_USB_STORAGE=y
            - CONFIG_OMAP2_DSS=y
            - CONFIG_DISPLAY_CONNECTOR_DVI=y
            - CONFIG_DISPLAY_CONNECTOR_HDMI=y
            - CONFIG_DISPLAY_PANEL_DPI=y
            - CONFIG_SECURITY_APPARMOR=y
            - CONFIG_SECURITY_APPARMOR_BOOTPARAM_VALUE=1
            - CONFIG_DEFAULT_SECURITY_APPARMOR=y
            - CONFIG_AUDIT=y
            - CONFIG_TI_ST= m
            - CONFIG_BT_WILINK=m
            - CONFIG_RFKILL=y
            - CONFIG_RFKILL_INPUT=y
        kernel-image-target: zImage
        kernel-device-trees:
            - omap4-panda-a4
            - omap4-panda
            - omap4-panda-es
        build-packages: [crossbuild-essential-armhf, bc, kmod, cpio]
    wl1xxx-firmware:
        plugin: dump
        source: https://git.kernel.org/pub/scm/linux/kernel/git/firmware/linux-firmware.git
        source-type: git
        organize:
            ti-connectivity: lib/firmware/ti-connectivity
```

snapcraft.yaml

# Gadget snap

```yaml
device-tree-origin: kernel
volumes:
  panda:
    schema: mbr
    bootloader: u-boot
    structure:
      - type: mbr
        size: 131072
        offset-write: 131072
        content:
          - image: MLO
      - type: mbr
        size: 393216
        offset-write: 393216
        content:
          - image: u-boot.img
      - type: 0C
        filesystem: vfat
        filesystem-label: system-boot
        size: 128M
        content:
          - source: uEnv.txt
            target: .
          - source: MLO
            target: .
          - source: u-boot.img
            target: .
```

meta/gadget.yaml

```yaml
name: panda
vendor: Enrique Hernández Bello <ehbello@gmail.com>
icon: meta/panda.png
license: meta/readme.md
version: 16.04-0.3
type: gadget
architectures: [armhf]
```

meta/snap.yaml

# Tabla de soporte

| Característica | Funciona |
|---|---|
| Sensores de temperatura | **Sí** |
| Ethernet | **Sí** |
| Wireless | **Sí** |
| Bluetooth | **Sin testear** |
| HDMI/DVI | **Sí** |
| Aceleración 3D | **No** |
| Sonido | **Sin testear** |
| USBs | **Sí** |

# Imagen distribuíble

```
type: model
authority-id: ngd1JO1HoTUHRzR5N2t34YEgbbQi2gjE
series: 16
brand-id: ngd1JO1HoTUHRzR5N2t34YEgbbQi2gjE
model: panda
architecture: armhf
gadget: panda
kernel: linux-panda
timestamp: 2016-09-16T19:01:43+00:00
sign-key-sha3-384: reXoK9znpgFWgOzdEPE_ePlW8jt0v8QHq3UtSsrWAXyoyG8xT5PNhQyu2KaF5Oqn

openpgp 2cln
```

## ASSERTION FILE

Archivo con contenido similar al superior.

- Define los parámetros que debe cumplir una máquina específica.

**<machine>.assertion**

## ubuntu-image

Herramienta para construir imágenes arrancables de Ubuntu.

- Soporta *assets*.
- Sustituye a sus antecesoras **ubuntu-device-flash** y **linaro-media-create**.
- Aún está en contínuo desarrollo.

# Un comando

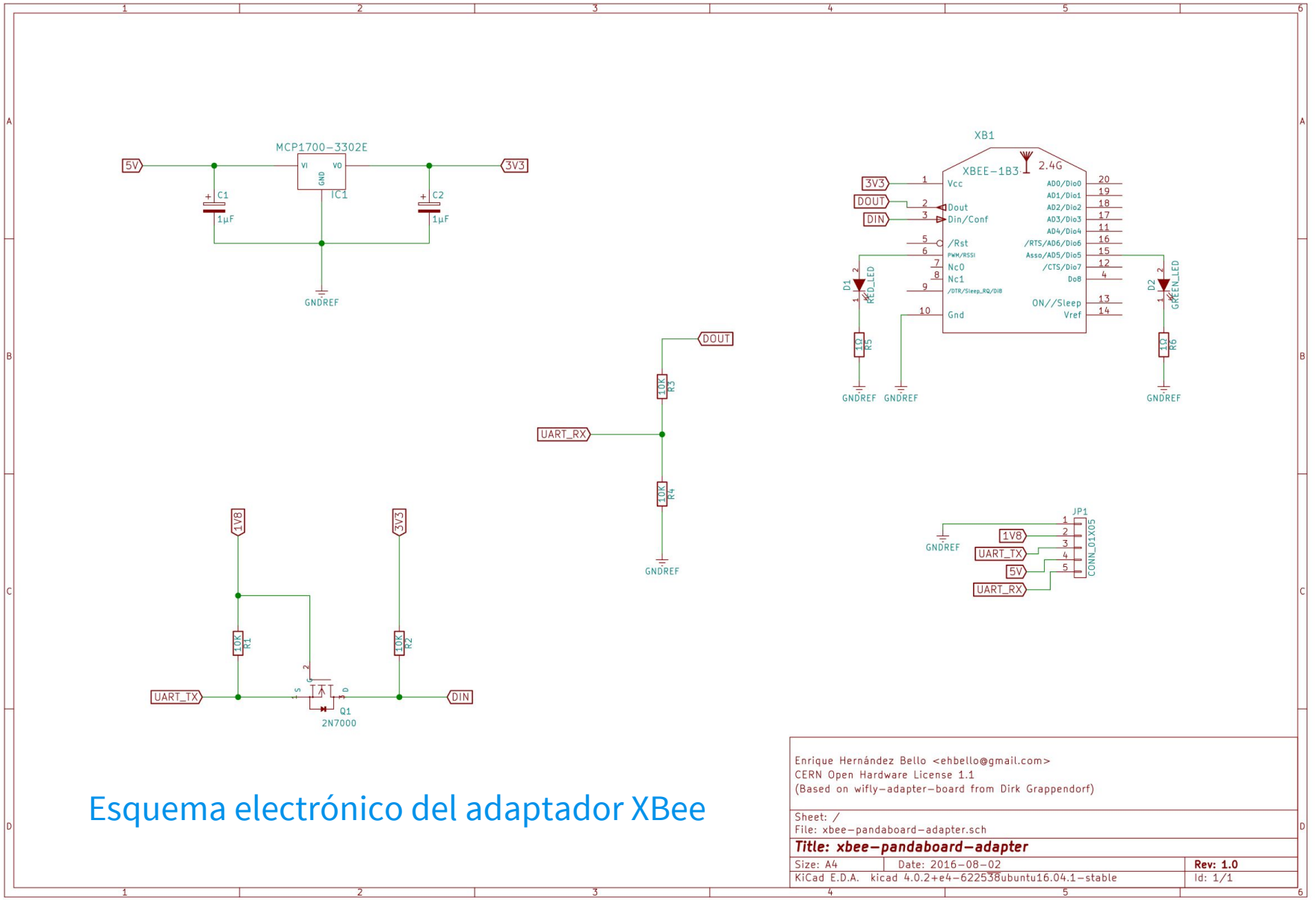
$ ubuntu-image panda.assertion -c edge -o panda.img [--extra-snaps <snap>]

## Problemas

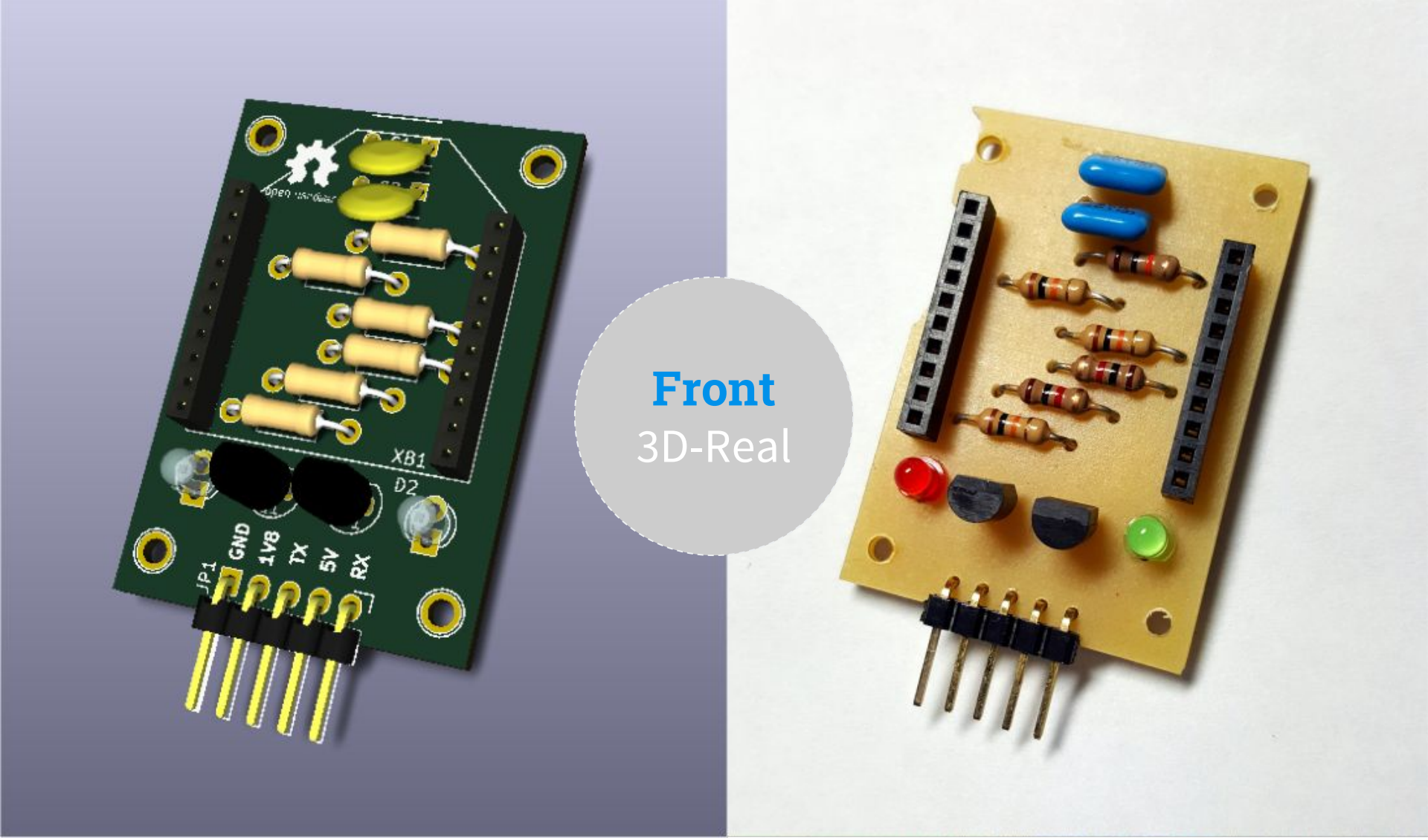
- Falta de documentación
- Snappy, Snapcraft 1.x y Ubuntu Core 15.04 obsoletos
- Constante evolución de Ubuntu Core 16
- Largos tiempos de compilación

# 2.
# Adaptador XBee

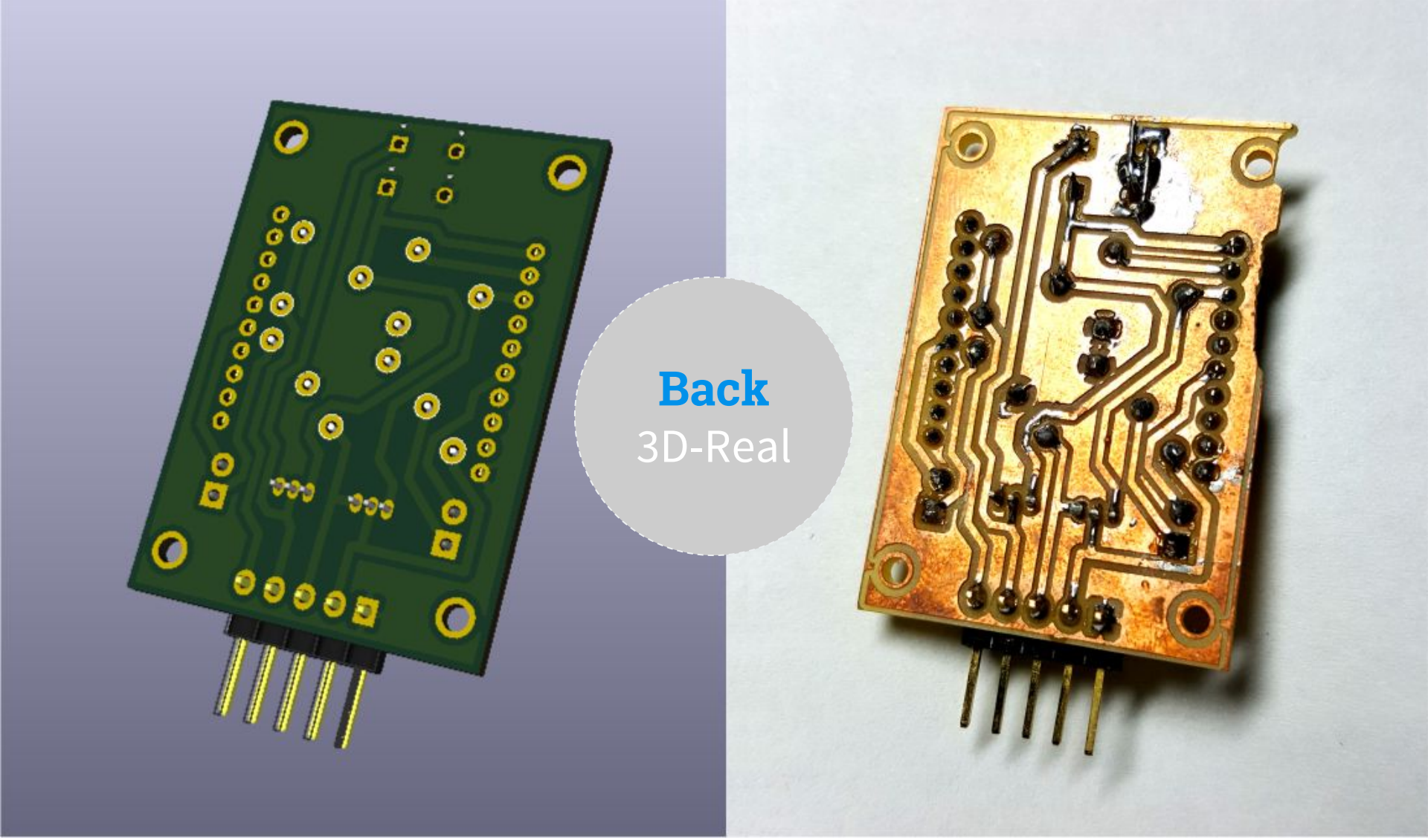
- Esquema electrónico
- Prototipo




# Esquema electrónico del adaptador XBee




Adaptador XBee resultante




Adaptador XBee resultante

# 3.
# Interfaz de usuario

- Nueva estrategia
- MQTT
- xbee2mqtt
- DomoticZ
- Traducción MQTT
- Salida HDMI

*“Más vale una retirada a tiempo, que una batalla perdida”*

## Cambio de estrategia

# Diagrama de funcionamiento MQTT

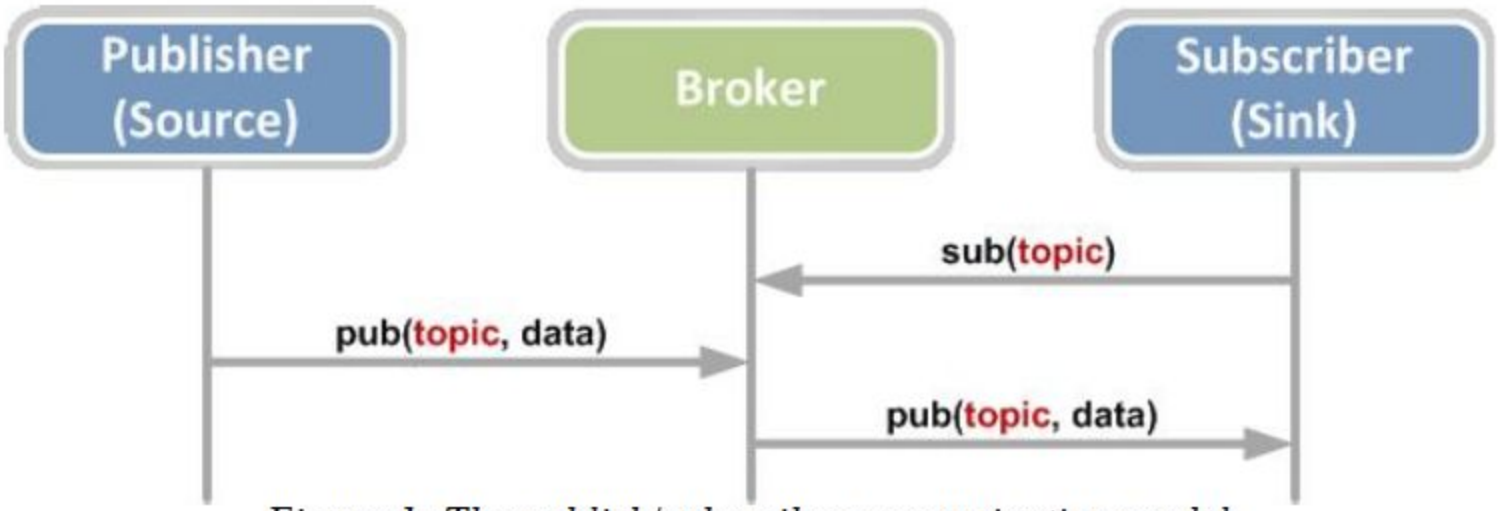
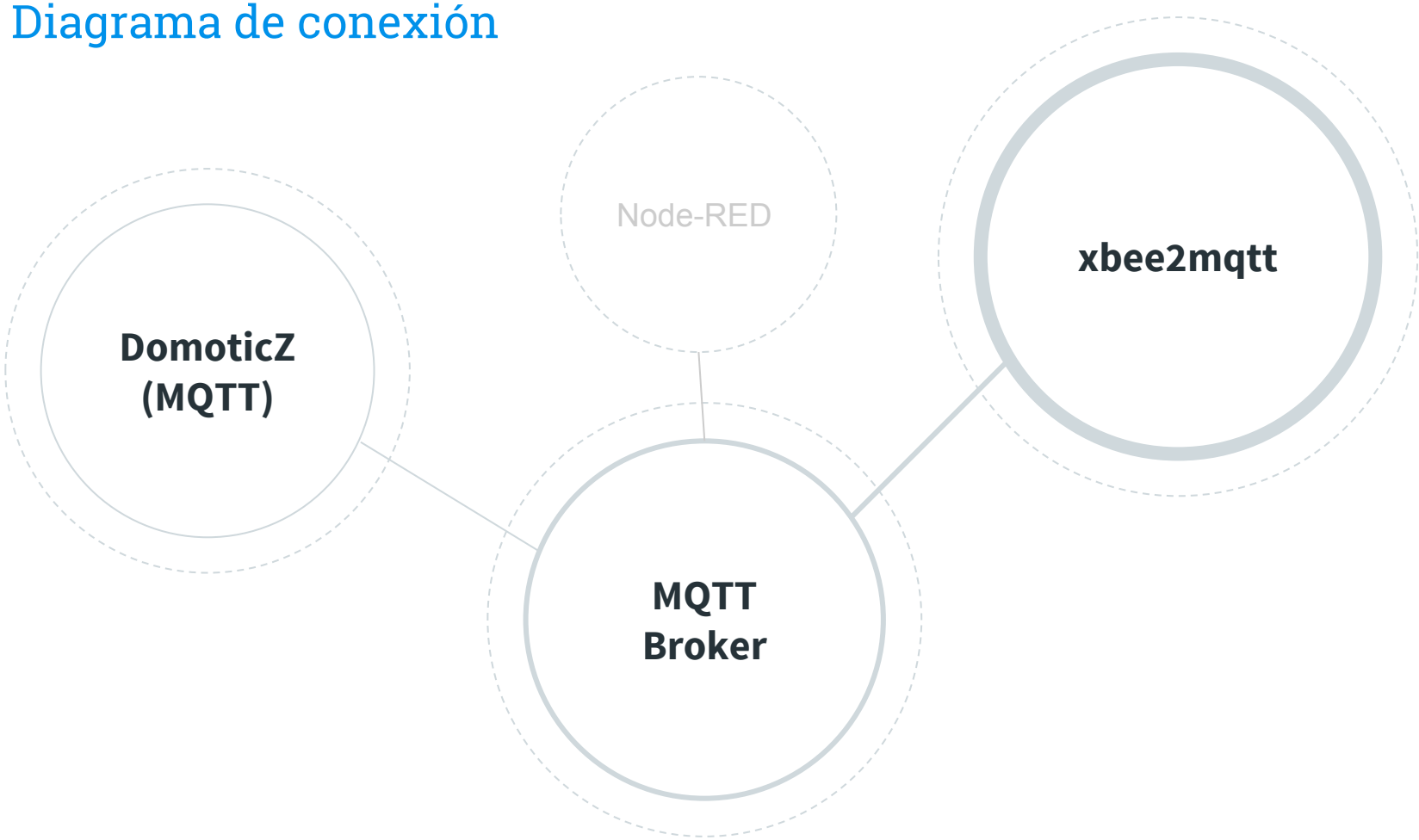



*Figure 1: The publish/subscribe communication model*

# Diagrama de conexión

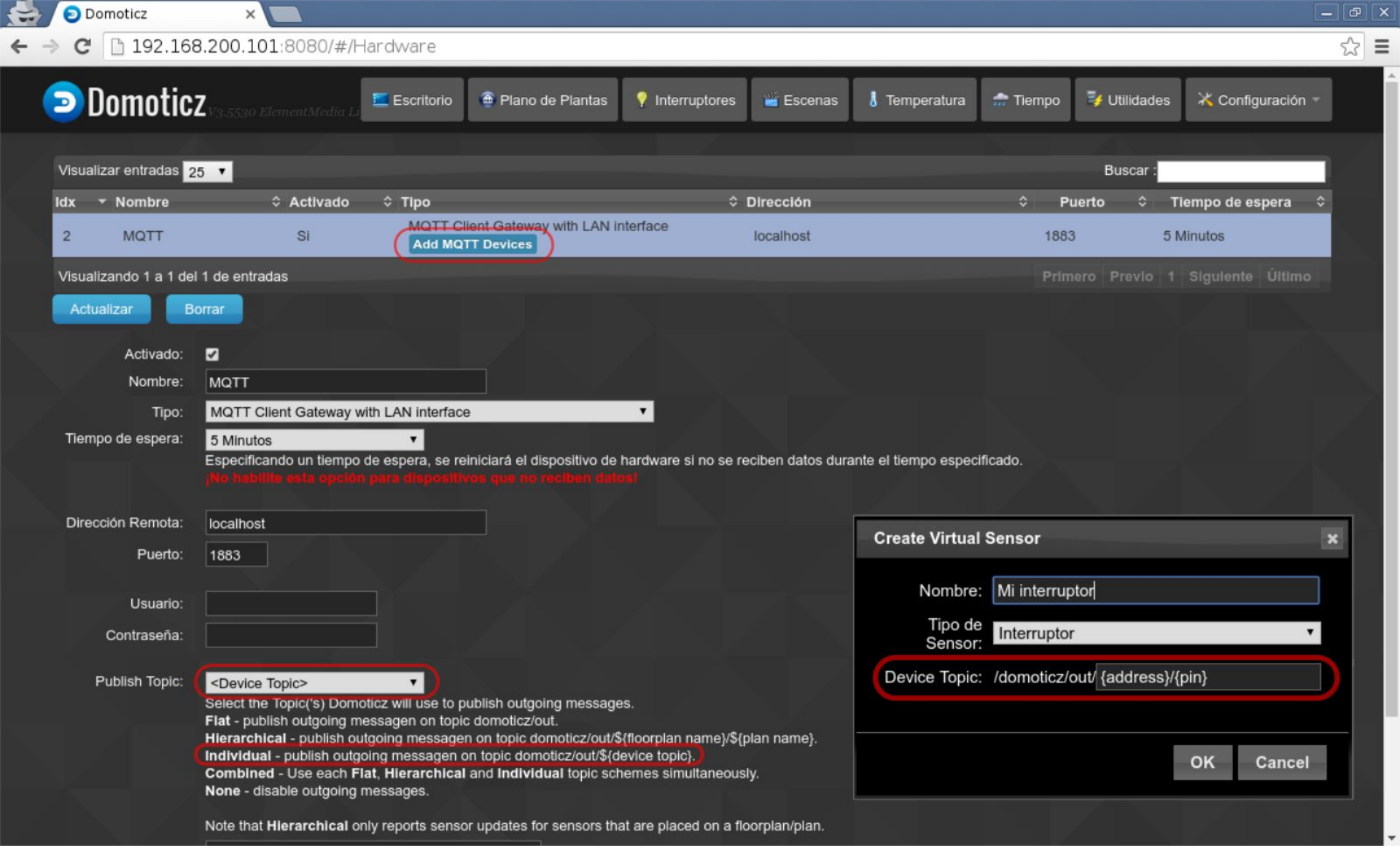


# xbee2mqtt

Pasarela XBee a protocolo MQTT escrita en Python

- Puesta en funcionamiento
- Ampliación del protocolo
  - Node Discovery
  - IO Sampling
  - Pin Modes
  - Etc.
- Subscripciones dinámicas

*Xose Pérez <xose.perez@gmail.com>*




# Configuración de hardware MQTT en **DomoticZ**

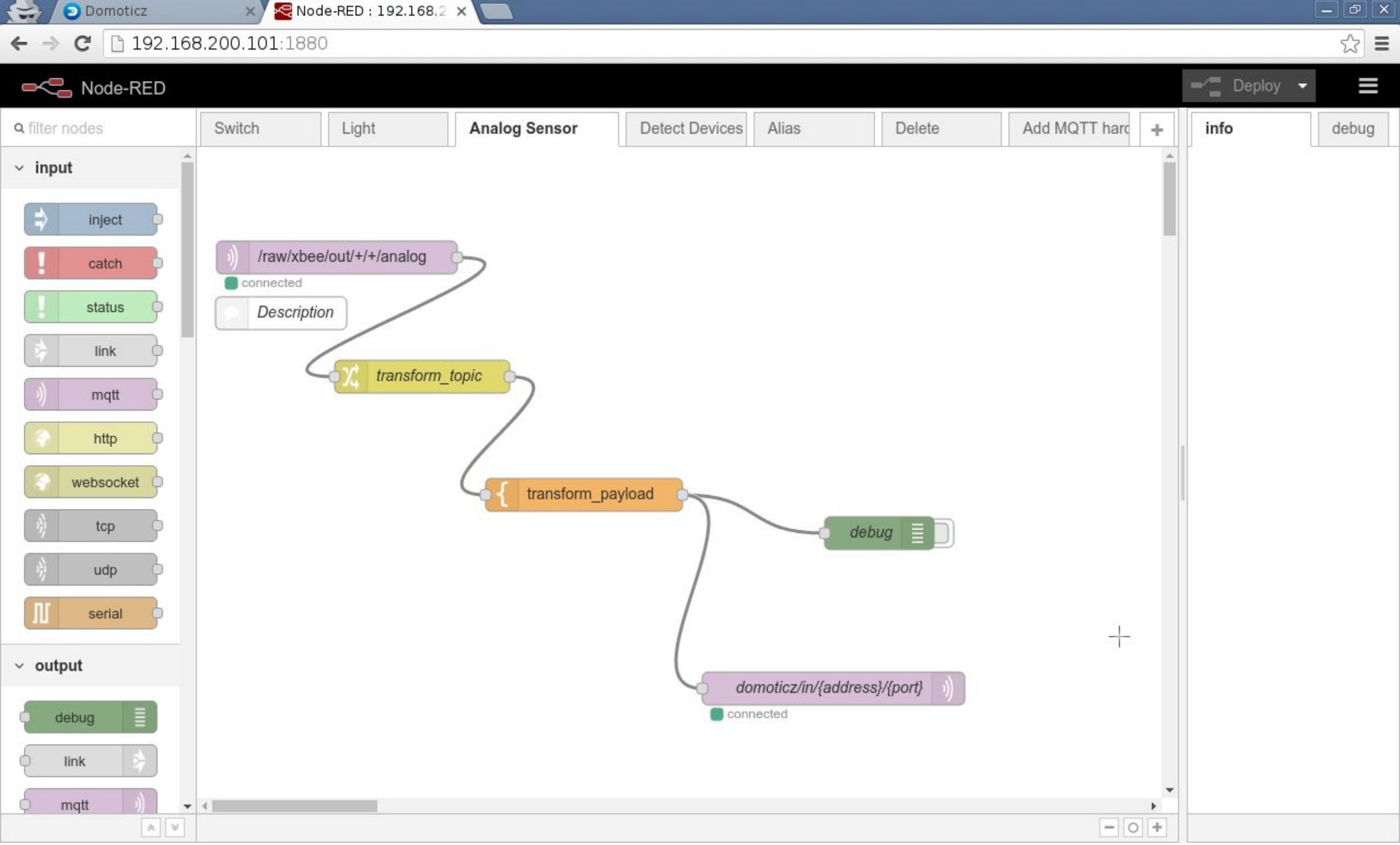
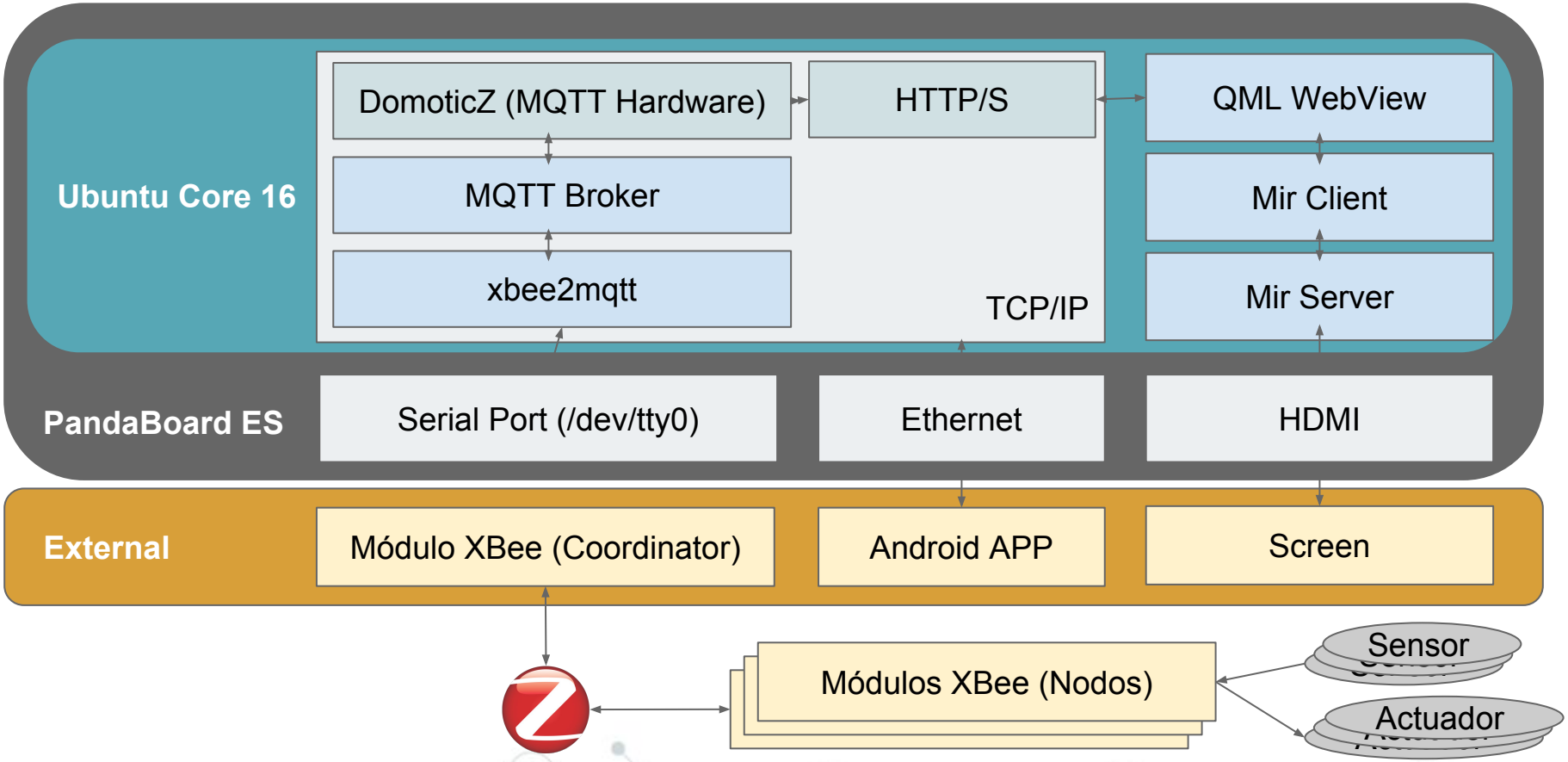



Diagrama de un flujo en **Node-RED**

# 4.
# Integración

- Visión global
- Qt WebView con QMLScenes

# Visión global del sistema



# Qt WebView con QMLScenes

**Lenguaje QML**

**QtQuick + WebView**

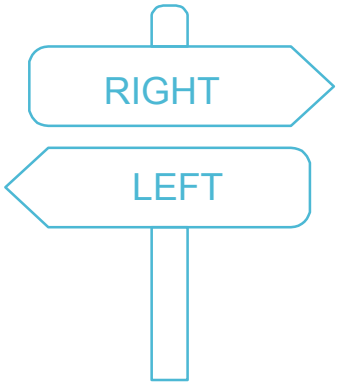
**QMLScenes**

Mir Server

# CONCLUSIONES

## Líneas a seguir

- Asegurar snaps (confinement, interfaces…)
- Configuración de red XBee
- Modos de sueño
- Ampliar o sustituir flujos de red Node-RED



*[...] the pandaboard is a **dead beef** nowadays.*

*Oliver Grawert <ogra@ubuntu.com>*

***15-09-2016***

# Repercusión en la comunidad

**+ 13min**

*“Thanks! [...] I’ll give it a twirl.”*

Andreas Hasenack
<andreas@canonical.com>

**+ 15min**

*“wow, this is impressive! I even see assertions and all… you should blog about it so we can share it on social media!”*

Oliver Grawert
<ogra@ubuntu.com>

**+4h**

*“+1, definitely!”*

Jammie Bennett
<jammie.bennet@canonical.com>

# Conclusions

This project attempted to provide a functional and highly customizable solution.

It has been a great challenge focused on the integration of elements of different characteristics to achieve the goals.

Although the project still has continuity and some aspects can be changed, I believe that these requirements have been surpassed.

On the other hand, this project has taken advantage of free software so it also has been designed to return the favor to the community.

I hope that, individually or completely, this work will be useful to someone.

# Gracias!

## ¿Preguntas?

Enrique Hernández Bello

alu0100355833@ull.edu.es

Repositorio GitHUB:

https://github.com/ULL-InformaticaIndustrial-Empotrados/TFG_ZigBee_Embedded_Controller.git

## Créditos

Un agradecimiento especial a todas las personas que realizan y publican recursos impresionantes de forma libre y gratuita.

- Plantilla de presentación por SlidesCarnival