

# Framework y Sistema Tea



Tiny Environmental Analyzer

# Enlaces

- **Blog**

- <http://proyectotea.wordpress.com>

- **Forja**

- <https://forja.rediris.es/projects/cusl4-tea/>

- **Archivos**

- [http://forja.rediris.es/frs/?group\\_id=692](http://forja.rediris.es/frs/?group_id=692)

# Contenido

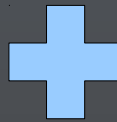
- **Resumen del proyecto, motivación**
- **Aspectos de interés**
- **Marco de desarrollo**
- **Ejemplo: Tea**
- **Hitos y logros**

# Acerca del proyecto

Marco de desarrollo



Entorno hardware/software



Herramientas



Metodología

# Objetivos

Orientado a

**SISTEMAS EMPOTRADOS**

Énfasis en

**Comportamiento dinámico**

**Flexibilidad**

**Extensibilidad**

# Motivación

## SISTEMAS EMPOTRADOS

Usados típicamente para:

- Monitorización
- Control de dispositivos

Los usamos a diario aunque no nos demos cuenta:

- Vehículos
- Reproductor MP3
- Electrodomésticos
- etc.

# Motivación

## SISTEMAS EMPOTRADOS

### Nueva generación de microcontroladores:

- Potencia para cualquier tipo de aplicación
- Soporte para sistemas operativos potentes
- Más posibilidades para el desarrollo
- Abanico más amplio de aplicaciones

# Motivación

- **Facilitar el desarrollo sobre sistemas empotrados.**
- **Demostrar el uso de tecnologías y herramientas libres para implementar un sistema completo.**
- **Capacidades avanzadas en aplicaciones para sistemas empotrados.**



# Aspectos de interés

- **Sistemas empotrados**
  - Arquitectura de computadores
  - Programación a bajo nivel
- **Sistemas operativos**
  - Control de dispositivos
  - Sistema operativo a medida
- **Desarrollo de aplicaciones**
  - Desarrollo orientado a objetos
  - Ingeniería del Software
- **Conectividad**
  - Web Services
  - Despliegue remoto de servicios

# Marco de desarrollo

## Cadena de herramientas

Nivel de aplicación



Plataforma de desarrollo

Sistema Operativo

Toolchain para desarrollo cruzado

Nivel hardware

# Toolchain (desarrollo cruzado)

- Nos permite generar código ejecutable en una plataforma de destino distinta.
- Necesaria porque en un sistema empotrado no tendremos un entorno de desarrollo.
- Necesaria para construir el sistema operativo para la plataforma destino.

# Toolchain (desarrollo cruzado)

- Opciones de configuración y construcción
  - Binarios
  - Biblioteca estándar
  - Otras opciones
- Problemas y soluciones
  - Crosstool-ng
- Alternativas

# Sistema Operativo

- Introduce un nivel de abstracción importante
  - Dispositivos
  - Memoria
  - Procesos
  - Red
  - Etc.
- Nos permite programar sobre la capa de servicios que proporciona
  - Archivos
  - Sockets
  - etc.

# GNU/Linux

## ¿Por qué Linux?

- **Software libre**
  - **Adaptable a muchas arquitecturas**
  - **Podemos modificarlo según nuestras necesidades**
- **Potente, completo y configurable**
- **Enorme cantidad de software disponible**
- **No nos cuesta nada**

# GNU/Linux

¿Cómo construirlo?

Kernel

+

Sistema de archivos raíz

+

Otros componentes

Cargador de arranque

Software adicional

Configuración

# Construcción del kernel

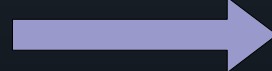
Código fuente

+

Parches

+

Configuración



Toolchain

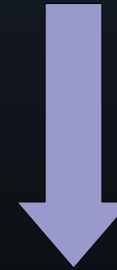
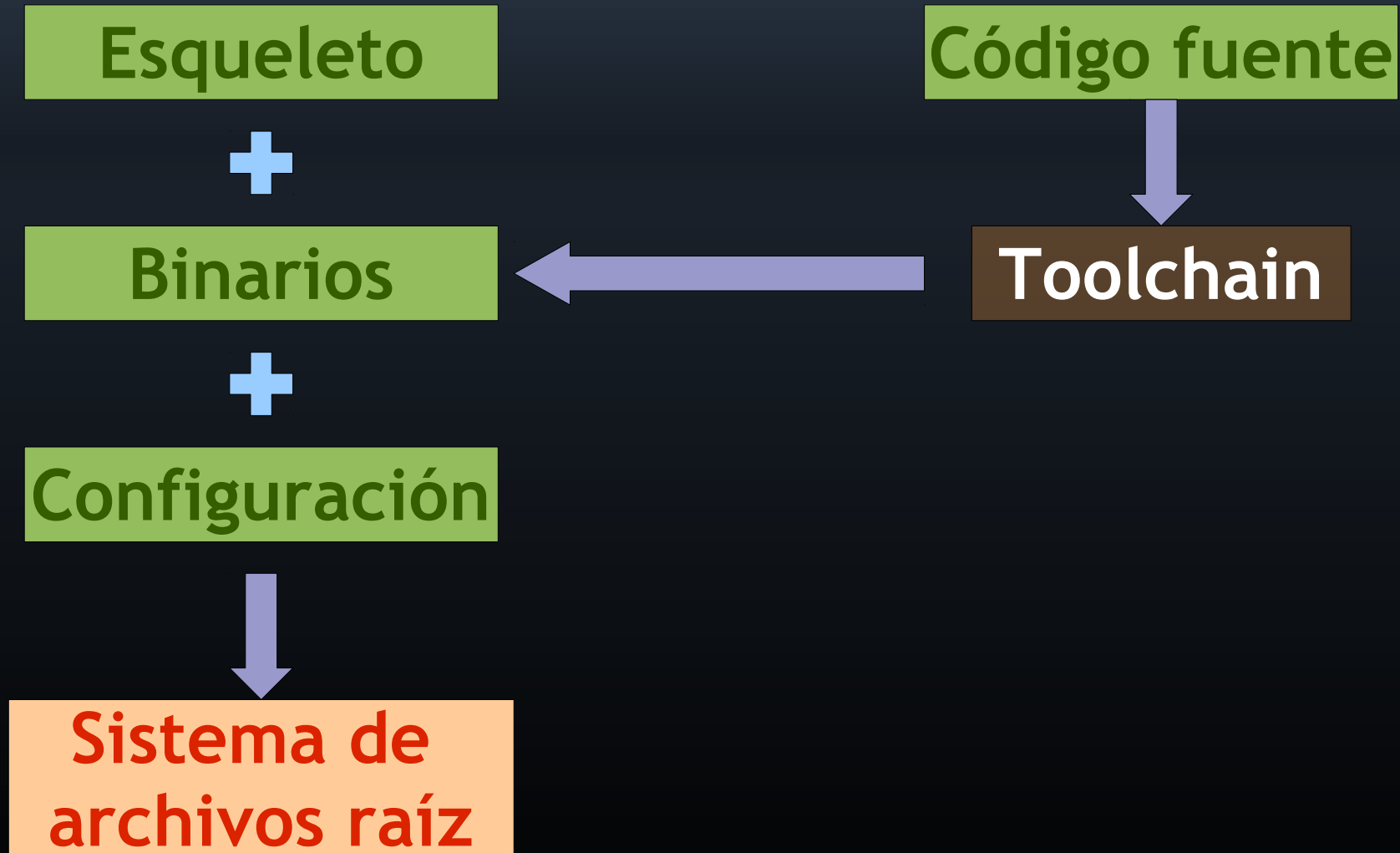


Imagen del kernel



# Sistema de archivos raíz



# Plataforma de desarrollo

## Objetivos

- Alto nivel
- Gran cantidad de bibliotecas
- Facilidad de desarrollo
- Dinámico
- Flexible

# Java

- **Cumple los requisitos anteriores**
- **Elimina la necesidad de compilación cruzada**
- **Excelente biblioteca estándar**
- **Fomenta buenas prácticas de IS**
- **Software libre**
  - **Alternativas en máquinas virtuales**
- **Útil para el desarrollo en servidor y cliente**

# OSGi

- **Añade a Java capacidades modulares**
  - Abstracción “módulo” en tiempo de ejecución
  - Completo sistema de módulos
- **Mejora y facilita el diseño y distribución de aplicaciones**
- **Permite añadir o quitar módulos a una aplicación “en caliente”**
  - Despliegue remoto de servicios

# Otras opciones

- Sustituir Java por C + bibliotecas necesarias
- Lenguajes de script ligeros: Lua
- Cualquier otra tecnología equivalente

# Ejemplo: Tea

- **Mini-estación ambiental**
- **Implementada como un sistema empotrado**
- **Autónoma**
- **Configurable remotamente**
- **Extensible**
- **Ofrece servicios a través de Internet**

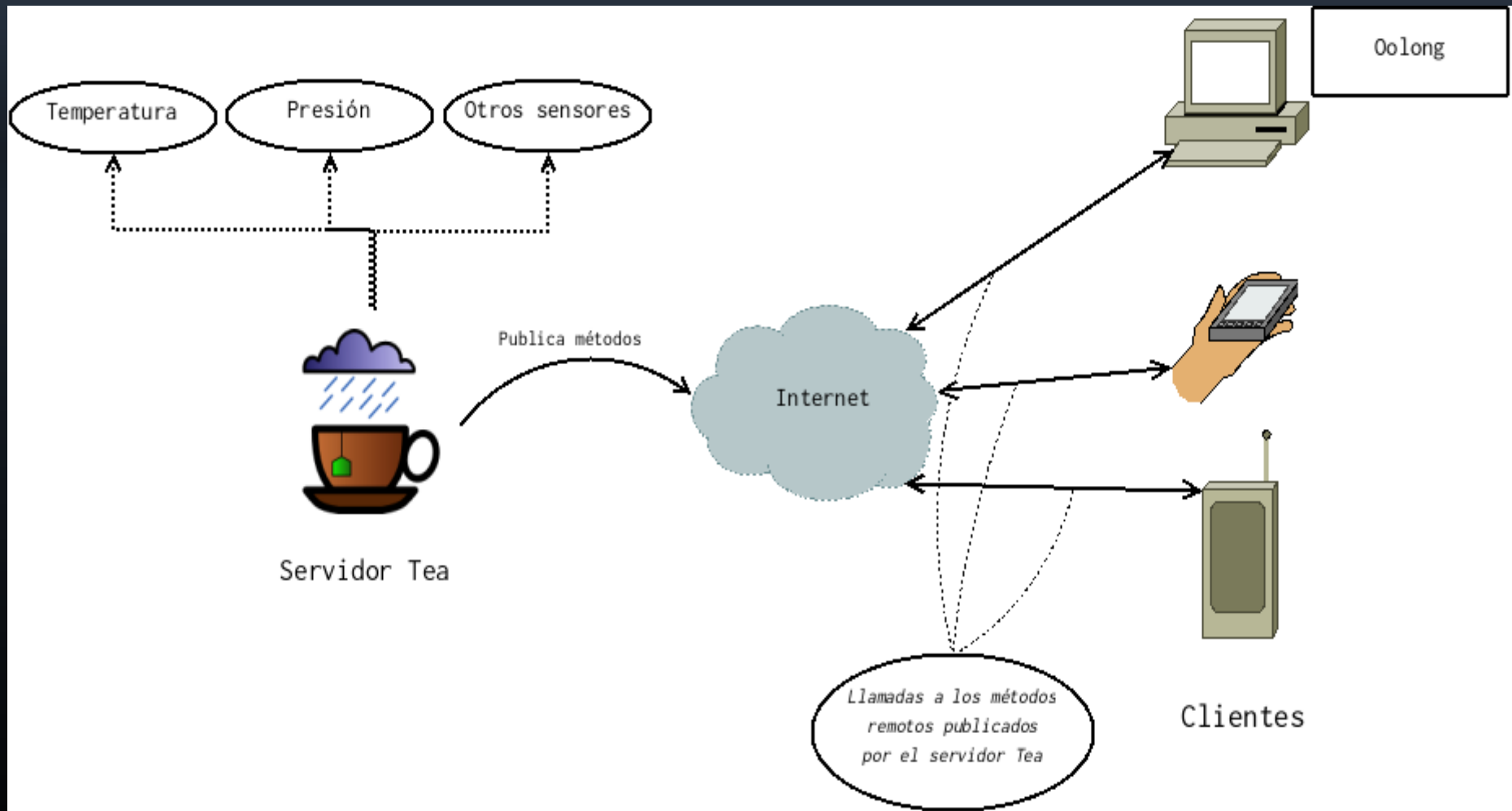
# Tea: Objetivos

Didáctico

Sencillo

Funcional

# ¿Cómo funciona?





# Tea: principales características

- **Bajo consumo**
- **Conexión en caliente de sensores**
- **Despliegue remoto de servicios**
- **Funcionamiento modular mediante plug-ins**

# Tea: Servicios

- **Datos meteorológicos en tiempo real**
  - Temperatura
  - Humedad
  - Presión
  - Etc.
- **Estadísticas**
- **Historial de medidas**
- **Servicios extensibles remotamente**

# Tea: Utilidad real

**Ejemplo: Estudio meteorológico de una zona de interés geográfico de difícil acceso**

- 1 Colocar la estación en la zona de interés**
- 2 Alimentación mediante baterías o energía solar**
- 3 Conexión a Internet**
- 4 La estación puede ser controlada, configurada y consultada desde cualquier lugar del mundo**

# Clientes y conectividad

Comunicación mediante  
Web Services



Accesibilidad desde  
Multitud de dispositivos

Único requisito para una aplicación cliente:

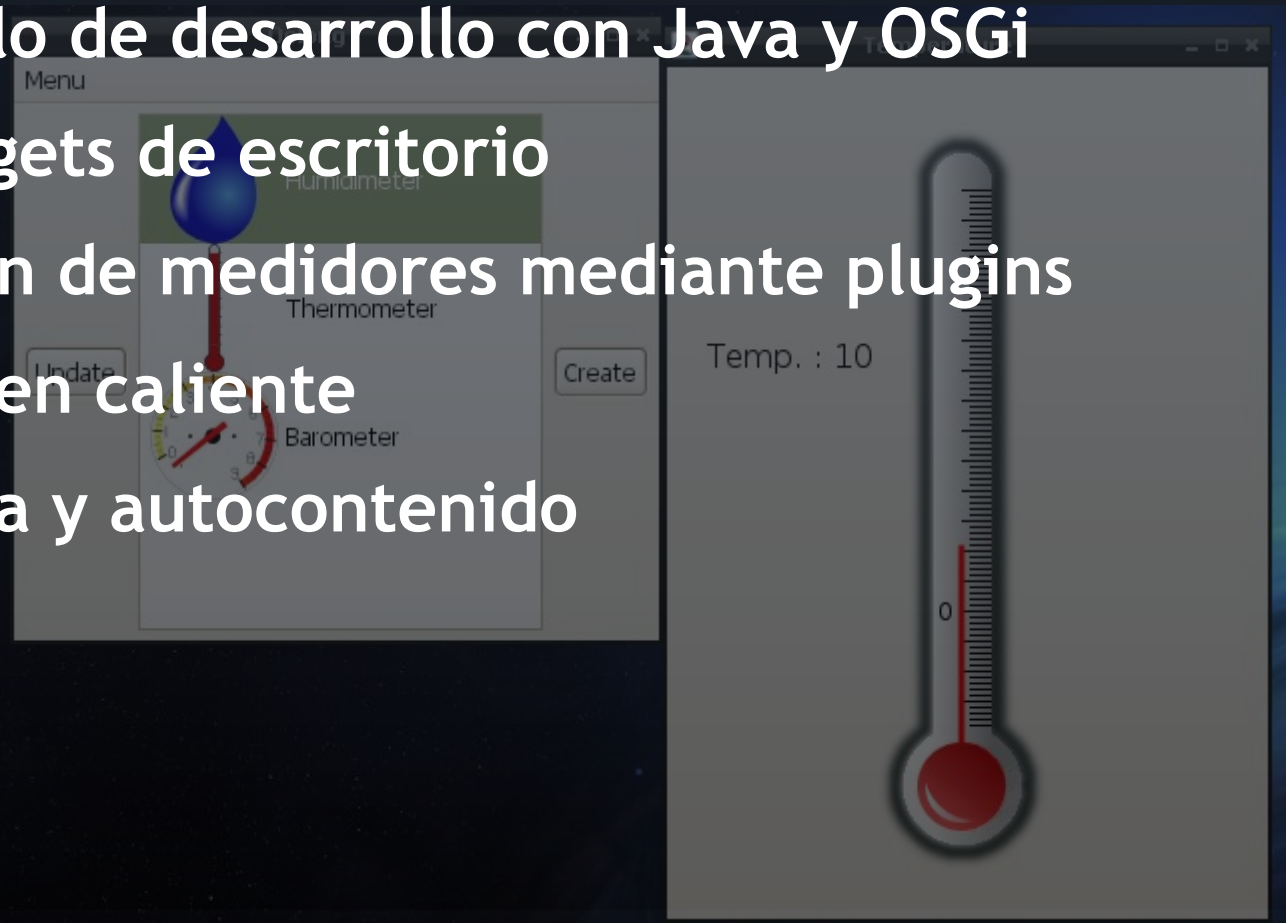
- Implementación de SOAP

Ejemplos de clientes para PC:

- Cliente gráfico basado en widgets: **Oolong**
- Cliente para línea de órdenes
- Cliente con interfaz web

# Oolong

- Cliente gráfico programado en Java con OSGi y Swing
- Ejemplo sencillo de desarrollo con Java y OSGi
- Modelo de widgets de escritorio
- Implementación de medidores mediante plugins
- Extensibilidad en caliente
- Multiplataforma y autocontenido



# Tea: Implementación

Hardware

BeagleBoard

+

Phidgets

# Tea: Implementación

## Software

- Servidor Tea en Java
- Servidor ssh para administración remota
- Interfaz web
  - Consulta
  - Administración

# TWI: Tea Web Interface

## Estación **Tea** Granada



[Configurar estación](#) [Administrar módulos](#)

**Uptime:** 2 days, 10:18

**Estado:** Off

**Módulos instalados:** Ninguno

**Sensores activos:** Ninguno

**Estadísticas:** [Ver...](#)



# Tea: Implementación

