

MarcoPolo

Diego Martín

13 de junio de 2015

1. Introducción

2. Objetivos del sistema

Se definen a continuación los objetivos del sistema a alto nivel:

3. Requisitos no funcionales

3.1. Independencia

La implementación del protocolo debe ser independiente de las aplicaciones que lo utilizarán, a fin de poder aumentar la adaptabilidad del protocolo a nuevas situaciones. Dicho objetivo se consigue delegando una gran parte de la funcionalidad a aplicaciones que se apoyan sobre esta implementación, en vez de albergar dicha funcionalidad en la misma.

3.2. *Zeroconf*

El protocolo funciona en una red sin requerir ningún tipo de configuración por parte del usuario, y en la mayoría de casos sin gran esfuerzo por parte del administrador.

3.3. Segmentación

Varias instancias del protocolo pueden ejecutarse en una misma red de forma independiente, permitiendo la creación de varias “mallas” de equipos. Dicha segmentación no debe alterar en absoluto el esquema de la red preexistente.

3.4. Conectable

Las diferentes aplicaciones presentes en los diferentes nodos deben poder aprovechar la funcionalidad del protocolo mediante una serie de elementos conectores (*bindings*).

3.5. Seguridad

En aquellos casos en los que la información enviada a través de los *bindings* o compartida por los nodos sea confidencial, el protocolo debe implementar las medidas oportunas para la protección de la misma.

3.6. Independencia de plataforma e implementación

Toda la comunicación entre elementos del protocolo se realizará a través de tecnologías que no dependan de una implementación concreta, tales como un lenguaje de programación o un sistema operativo dado. El único elemento que puede presentar tal dependencia es el conector final con otro código fuente, así como cualquier otro punto final (comandos, ficheros de configuración, etc).

3.7. Independencia del espacio de direcciones, nombres o cualquier otro elemento de red

El protocolo debe funcionar en cualquier espacio de direcciones dado, sin considerar en cualquier caso la dependencia con protocolos como **DHCP** o **DNS**.

3.8. Simplicidad

Los comandos del protocolo deben ser simples y, en caso de que sea posible, deben ser similares a otros ya conocidos por los usuarios del sistema, a fin de que estos ya estén familiarizados con los mismos.

3.9. Descentralización

El protocolo no debe en ningún momento establecer un potencial “cuello de botella”.

3.10. Configurable

3.11. Visibilidad

3.12. Optimización de la red

3.13. Sin conexión

3.14. Extensible

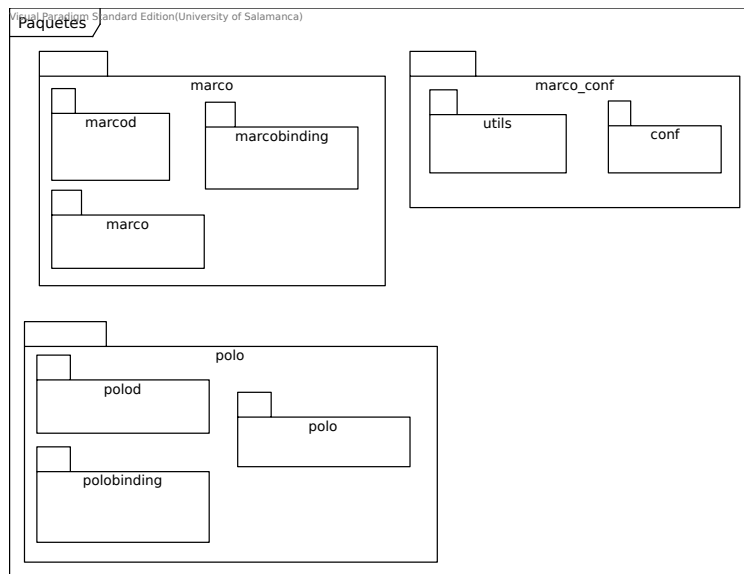
3.15. Extensible a diferente hardware

4. Protocolo

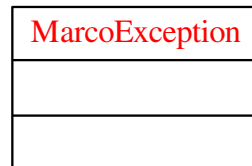
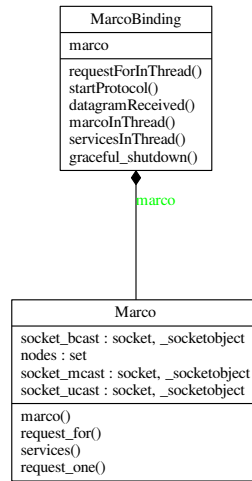
4.1. Marco

5. Propuesta de implementación

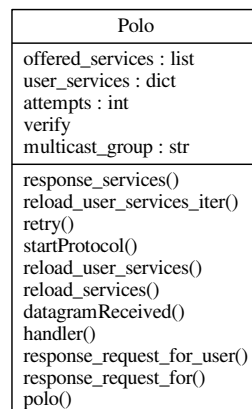
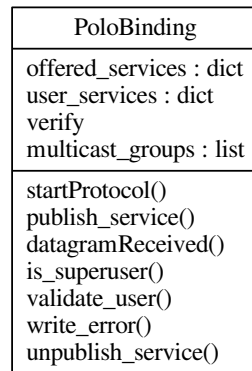
5.1. Vista estática



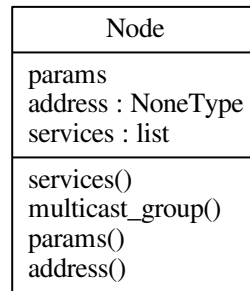
5.1.1. Paquete Marco



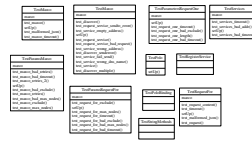
5.1.2. Paquete Polo



5.1.3. Paquete utils

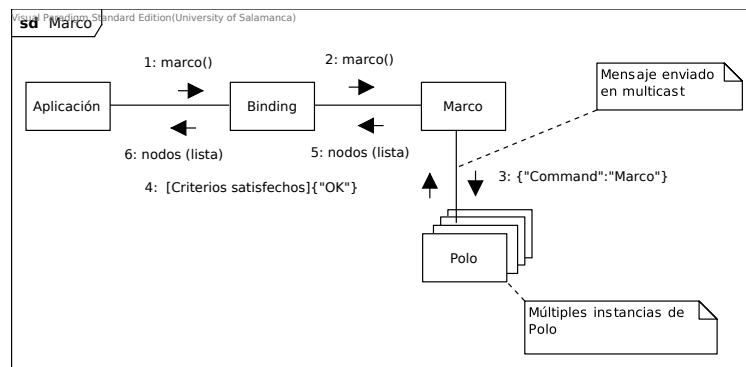


5.1.4. Paquete tests

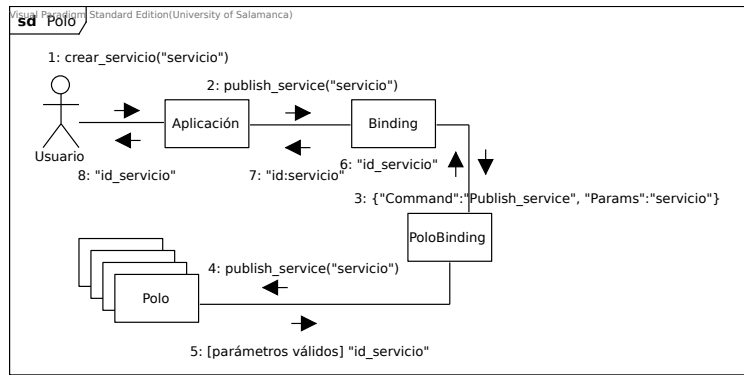


5.2. Vista de interacción

5.2.1. Diagramas de comunicación



Comunicación en Marco



Comunicación en Polo

5.3. Vista de actividad

