

# For-All - Uma Plataforma para Sistemas Pervasivos Orientados a Serviço

Elenilson Vieira da S. Filho<sup>1</sup>, Ângelo L. Vidal de Negreiros<sup>1</sup>, Alisson V. Brito<sup>2</sup>

<sup>1</sup>Departamento de Informática – Universidade Federal da Paraíba (UFPB) – Campus I  
Cidade Universitária, João pessoa, PB - Brasil

<sup>2</sup> Departamento de Ciências Exatas. Grupo de Computação Aplicada (*Applied*) –  
Universidade Federal da Paraíba (UFPB) – Campus IV  
Rua Das Mangueiras S/N, Rio Tinto, PB - Brasil

elenilson@elenilsonvieira.com, angelolvnegreiros@gmail.com,  
alissonbrito@cdae.ufpb.br

**Abstract.** *The objective of this work is to offer services focused on user, everytime and independent on place and technology through a platform called For-All for development, management and monitoring of intelligent buildings, enabling the creation of pervasive systems, placing intelligent sensors in a service oriented and based on context platform.*

**Resumo.** *O objetivo deste trabalho é oferecer serviços focados no usuário, o todo tempo e independente do local e tecnologia através de uma plataforma chamada For-All, para o desenvolvimento, gerência e monitoramento de ambientes inteligentes que proporcionarão a criação de sistemas pervasivos, unindo a utilização de sensores inteligentes em uma plataforma orientada a serviços e baseada no contexto.*

## 1. Introdução

Atualmente há uma crescente necessidade de informação em todos os lugares de forma rápida e simples. Essa necessidade vai desde uma simples informação da falta ou não do abastecimento de água em uma cidade até a temperatura de uma sala com importantes servidores. O acesso a essas informações é feito através de serviços disponibilizados que precisam ser de fácil acesso para que, desse modo, uma resposta possa ser gerada rapidamente.

A intenção está na oferta e gerência de tais serviços computacionais a qualquer tempo e lugar, com todas as funcionalidades desejadas e do modo desejado. Com a *Internet of Things*, é possível fazer com que qualquer dispositivo do mundo real tenha uma existência própria no mundo virtual [MELOAN, 2003], onde objetos do mundo físico possuem um identificador único e uma descrição bem simples, o que facilita a interação e o acesso a eles [SIORPAES, 2006]. Esses objetos são capazes de detectar mudanças na qualidade física das coisas usando tecnologias sensoriais. Dessa forma será possível criar um ambiente próprio, chamado de ambiente inteligente, moldável às situações do ambiente em que o usuário se encontra.

Com esses serviços disponíveis, aplicações poderão ser desenvolvidas focadas no monitoramento de ambientes como: casas, carros, fábricas ou ainda, por exemplo, podem ser desenvolvidos sistemas de segurança para monitoramento de máquinas ou entrada de funcionários.

As melhores tecnologias são aquelas que desaparecem, se misturam com o meio a ponto de não serem percebidas [WEISER, 1991]. Dessa forma tem-se a ubiquidade do computador, onde não é mais o usuário que entra no mundo virtual, mas o computador que entra no mundo do usuário. A computação pervasiva surge como um novo paradigma, permitindo aos usuários acesso aos seus ambientes computacionais de todo o lugar, o tempo todo [SAHA e MUKHERJEE, 2003]. A computação pervasiva é hoje uma realidade [YAMIN, 2004].

A plataforma proposta por este trabalho possibilita a manipulação e a configuração de dispositivos que estão distribuídos em um ambiente inteligente, disponibilizando serviços computacionais, como base para a computação pervasiva e baseada no contexto. Tais serviços devem ser oferecidos e utilizados por qualquer tipo de dispositivo independente de tecnologia e protocolo. O projeto For-All visa exatamente à universalização dos dispositivos e a transparência da troca dos serviços. O projeto Ninja [Gribble, 2001] possui objetivos semelhantes em termos de arquitetura, onde procuram habilitar uma inovação ampla em serviços distribuídos na internet. Sua arquitetura consiste de quatro elementos básicos: as bases, que são poderosos terminais com ambientes aglomerados usando um software como plataforma que simplificam a construção de serviços; unidades, que são os dispositivos pelos quais os usuários acessam os serviços; active proxies, são elementos de transformação que são usados pelas unidades ou adaptação de serviços específicos; e o path, que é uma abstração. O For-All tem como principais vantagens uma maior simplicidade e clareza, além da grande abstração que é provida sobre os protocolos de comunicação.

A seguir a seção 2 apresenta a plataforma For-All e suas características, e por fim, a seção 3 mostra as considerações finais sobre este projeto e trabalhos futuros.

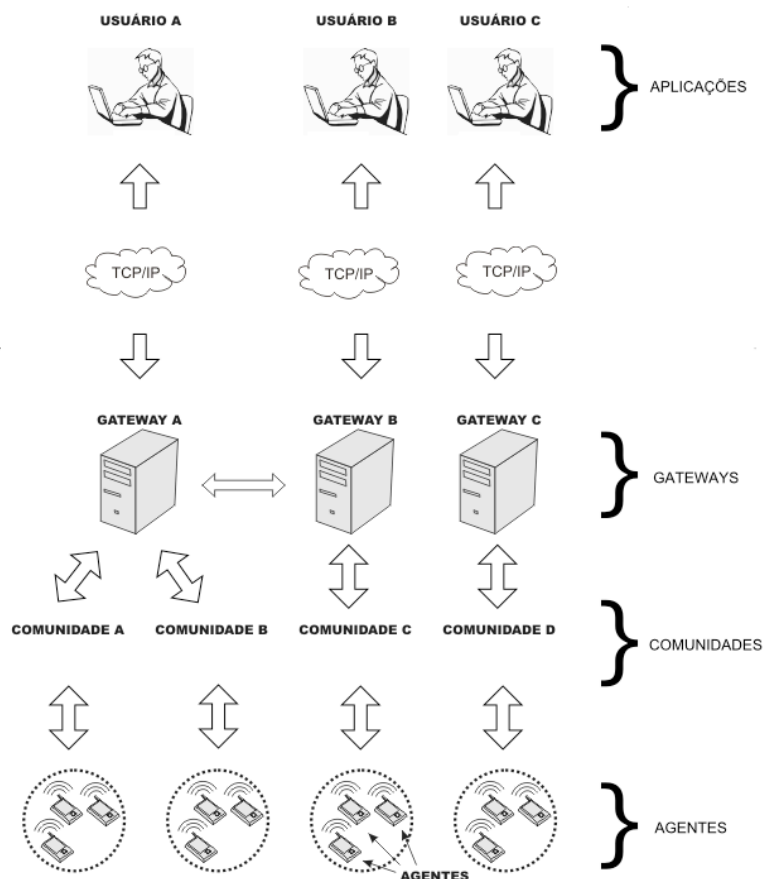
## **2. O projeto For-All**

### **2.1. Uma Arquitetura Orientada a Serviços**

Uma Arquitetura Orientada a Serviços é a evolução da computação distribuída com base em pedido/resposta para uma concepção síncrona e assíncrona de aplicações sendo um estilo de arquitetura de software cujo princípio fundamental preconiza que as funcionalidades implementadas pelas aplicações devem ser disponibilizadas na forma de serviços [JOSUTTIS, 2007].

Atualmente há diversos dispositivos fornecendo diversas operações sem qualquer padronização, tornando difícil o gerenciamento de suas utilizações. O Projeto For-All utiliza os conceitos de orientação a serviços para criar uma arquitetura de rede para o desenvolvimento, troca e manipulação de serviços baseados nesses dispositivos. A arquitetura da plataforma é composta de quatro camadas intercomunicantes, como pode ser visto na Figura 1. As camadas inferiores estão relacionadas a implementação e disponibilização dos serviços descritos no SOA, onde é possível, o sensoriamento e gerenciamento desses sensores. A camada superior é responsável pela comunicação com aplicações destinadas a disponibilização dos serviços. Entre essas camadas há outra

que faz a ligação entre a camada dos sensores e a camada de disponibilização dos serviços, abstraindo todo o protocolo de comunicação utilizado e facilitando o desenvolvimento de outras camadas que se façam necessárias.



**Figura 1. Esquema arquitetural disposto em camadas**

A camada de sensoriamento é a mais baixa do sistema e utiliza entidades, descritas como Agentes, equipados com sensores para coletar dados do ambiente. A segunda camada é a das Comunidades, que são entidades responsáveis por agrupar agentes em uma entidade maior (a identificação do agente é inserida em uma tabela na comunidade), possibilitando o acesso a vários sensores de maneira centralizada.

Acima das comunidades, existe a camada dos Gateways, responsável por ser o elo entre as comunidades e as aplicações finais. Sua principal característica é ser acessível através do protocolo TCP/IP, a fim de receber requisições tanto da camada de aplicação quanto da camada das comunidades. Seria inconveniente ter uma comunidade (capaz de se comunicar com agentes, seja qual for o protocolo utilizado) e ainda ser acessível por TCP/IP, pois, se o protocolo de comunicação utilizado entre os agentes e as comunidades for sem fio, essas comunidades deverão ficar em lugares estratégico, podendo ainda ter um número elevado para poder cobrir uma grande área. O gateway surge como solução e registra comunidades (a comunidade estabelece uma conexão e sua identificação é inserida em uma tabela no gateway com campo identificador sendo o par IP/porta), conhecendo seus agentes e serviços disponíveis, possibilitando a

requisição por aplicações finais. Esta entidade é capaz de se comunicar com outros gateways, podendo, assim, conhecer todas as comunidades, agentes e serviços disponíveis destes outros.

A mais alta camada da arquitetura é a de Aplicação. É nessa camada que se encontrarão as aplicações requisitantes de serviços ao gateway e é nela que será feita a interação com o usuário final.

## 2.2. Disponibilização de serviços

Agentes fornecem serviços para o ambiente externo, caracterizando o For-All como uma arquitetura orientada a serviços. Eles fornecem as descrições dos serviços oferecidos pela rede e oferecem acesso a esses serviços através das comunidades em que se encontram registrados. Ao mesmo tempo, comunidades atuam como solicitantes de serviços para agentes, requisitando seus serviços especializados a fim de atender às necessidades das aplicações.

Comunidades registram-se no gateway a fim de disponibilizar o repositório de agentes existente. No relacionamento gateway-comunidade, o gateway atua como solicitante de serviços, repassando as requisições recebidas das aplicações, ou ainda, se a solicitação do serviço não for para nenhuma de suas comunidades, o gateway encaminha a requisição para o gateway correto.

## 2.3. Etapas de uma requisição de serviços

Aplicações conectam-se a um gateway, obtêm quais comunidades e serviços estão disponíveis e submetem requisições de serviços. A plataforma funciona segundo quatro etapas principais (vide Figura 2), descritas a seguir:

Etapa 1: Requisições são enviadas ao gateway que identifica a qual gateway a comunidade que detém o serviço está registrada, podendo está em si próprio.

Etapa 2: Se a solicitação for enviada para outro gateway, o processo é repetido, caso contrário, a requisição é enviada para a comunidade correspondente.

Etapa 3: Ao receber a requisição, a comunidade identifica em qual agente o serviço solicitado se encontra e a repassa para o agente.

Etapa 4: Um agente recebe a requisição da comunidade, identifica o serviço requerido, processa-o como uma thread separada e envia a resposta para a comunidade, que a repassa para o gateway e, por consequência, para a aplicação que originou a requisição na Etapa 1.

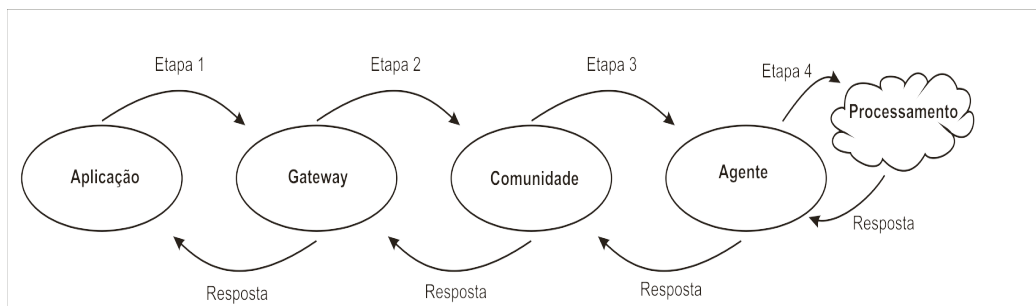


Figura 2. Etapas da comunicação

## 2.4. Service Discovery

O Service Discovery (SD) é um serviço padrão que cada entidade (aplicação, gateway, comunidade e agente) tem sua própria implementação, porém com a mesma finalidade: encontrar serviços. Nas aplicações o SD requisita ao Gateway (GW) os serviços registrados. No GW a requisição é feita à comunidade e nesta a requisição é feita procurando agentes em sua localização, onde esses agentes podem ser registrados e ter seus serviços disponibilizados.

## 2.5. Serviços

Dois serviços estão pré-configurados no framework, são eles: monitoramento e alarme. O Serviço de Monitoramento visa utilizar os sensores do agente para coletar dados do ambiente, podendo estes dados serem manipulados. O Serviço de Alarme utiliza um monitor para coletar os dados e verifica se estes estão dentro de um limiar pré-configurável. Se os dados coletados não satisfazem a condição de aceitação, uma ação, configurada pelo desenvolvedor do serviço, será tomada. Outros serviços podem ser configurados e dependem apenas das características físicas (sensores) dos agentes.

## 3. Considerações Finais e Trabalhos Futuros

A plataforma For-All possibilitará o desenvolvimento de aplicações orientadas a serviços, pervasivas e baseadas no contexto com baixo custo e eficiência. Atua na abstração de problemas e vem como solução para a implantação de ambientes inteligentes, provendo uma computação pervasiva de maneira simples, ou seja, a pervasividade não está na plataforma, mas sim nas aplicações que a utilizam como base.

Como trabalho futuro, será desenvolvido uma camada que ficará acima da camada de aplicação proporcionando a adaptação ao contexto. Aplicações baseadas no contexto levam em consideração na sua tomada de decisão e em seus processamentos não apenas as entradas explícitas, mas também entradas implícitas, provenientes do contexto físico e computacional do ambiente e de seus usuários [CALVI, 2005]. Com isso novas aplicações, baseadas em novos serviços e adaptadas ao contexto poderão se integrar à plataforma. Para validar completamente a arquitetura do For-All, será feito a prototipação de uma casa inteligente que terá agentes responsáveis por, por exemplo: monitoramento da luminosidade de ambientes, movimentação das portas (segurança) e monitoramento do fornecimento de energia elétrica e nível da caixa de água (comodidade).

## Referências Bibliográficas

- Calvi, C., Pessoa, R. and Filho, J. (2005). "Um Interpretador de Contexto para Plataformas de Serviços Context-Aware." In XXV Congresso da Sociedade Brasileira de Computação.
- Gribble, S. D. et. Al (2001). "The Ninja Architecture for Robust Internet-Scale Systems and Services". University of California at Berkeley ca 94720-1776, USA. Disponível em <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.17.6826>. Acesso em 10 de maio 2010.
- Josuttis, M., "SOA in Practice. The Art of Distributed System Design". 1. ed. Califórnia: editora O'Reilly, 2007.

- Meloan S. (2003). Toward a Global “Internet of Things”. In: Sun Developer Network. <http://java.sun.com/developer/technicalArticles/Ecommerce/rfid/>
- Saha, D and Mukherjee, A. “Pervasive computing: a paradigm for the 21st century”. *Ieee computer*, v.36, n.3, 2003, p.25–31.
- Siorpaes, S., Broll, G., Paolucci, M., Rukzio, E., Hamard, J., Wagner, M. and Schmidt, A. (2006). "Mobile Interaction with the Internet of Things." In Poster at 4th International Conference on Pervasive Computing (Pervasive 2006), Dublin, Ireland, 2006.
- Weiser, M. (1991). “The Computer For TheTwenty-First Century”. In *Scientific American*, pages. 94-104, september, 1991.
- Yamin, A., Augustin, I., Barbosa, J., Silva, L., Real, R., Cavalheiro, G. and Geyer, C. (2003) “Towards merging context-aware, mobile and grid computing.” In *International journal of high performance computing applications*, Londres. Anais. Pages 191–203.