

# IPSS: Uma Plataforma de *Middleware* para Distribuição de Informações em *Smart Spaces*

Marcos Alves Vieira<sup>1,2</sup>, Ernesto Fonseca Veiga<sup>1</sup>,  
Leandro Alexandre Freitas<sup>1</sup>, Sergio T. Carvalho<sup>1</sup>

<sup>1</sup>Instituto de Informática – Universidade Federal de Goiás (UFG)  
Goiânia – GO – Brasil

<sup>2</sup>Instituto Federal de Educação, Ciência e Tecnologia Goiano (IF Goiano)  
Iporá – GO – Brasil

{marcosalves, ernestofonseca, leandroalexandre, sergio}@inf.ufg.br

**Abstract.** *This paper proposes a middleware focused on the distribution of information in ubiquitous environments. The architecture of the Information Provider for Smart Spaces (IPSS) provides services that can be used by applications that need to provide and obtain information. The solution for the provision of information in smart spaces assumes that each individual has interests in particular topics. Therefore, we built a model of information distribution based on the interests of each user. An application was developed as proof of concept, using the IPSS. In this application, the focus is the provision of information in an academic environment.*

**Resumo.** *Este artigo propõe um middleware focado na distribuição de informações em ambientes ubíquos. A arquitetura do Information Provider for Smart Spaces (IPSS) fornece serviços que podem ser usados por aplicativos que precisam fornecer e obter informações. A solução para a disponibilização de informações em espaços inteligentes assume que cada indivíduo tem interesses em tópicos específicos. Dessa forma, foi construído um modelo de distribuição de informações a partir dos interesses de cada usuário. Um aplicativo foi desenvolvido como prova de conceito, usando o IPSS. Nesta aplicação, o foco é o fornecimento de informações em um ambiente acadêmico.*

## 1. Introdução

Nos dias de hoje, as pessoas são cada vez mais expostas a informações indesejadas pelas mais diversas formas de mídia, tais como propagandas de produtos, reportagens jornalísticas e outros tipos de divulgação em geral [Hlavac and Schaefer 2013]. Grande parte das informações recebidas não são úteis ou de interesse dessas pessoas, tomando seu tempo e não trazendo nenhum tipo de benefício. Por outro lado, cada indivíduo, de forma particular, possui interesse em determinadas informações que são relevantes para o seu dia a dia, sejam estas relativas à sua vida pessoal ou profissional.

Objetivando que as informações de interesse cheguem ao usuário de forma transparente e tirando proveito da crescente popularização dos dispositivos computacionais móveis, tais como *smartphones* e *tablets*, este artigo apresenta a proposta do *middleware Information Provider for Smart Spaces (IPSS)*, que se vale de características da

computação ubíqua, tais como mobilidade e sensibilidade ao contexto, para fornecer serviços de distribuição de informações. Os serviços do IPSS podem ser utilizados por aplicações que necessitem de primitivas para disponibilizar e obter informações, para os mais diversos propósitos, auxiliando dessa forma no aspecto de comunicação em *smart spaces* (espaços inteligentes).

O IPSS foi implementado na linguagem Java, utilizando o *middleware* de comunicação *Java Message Service* (JMS) e o *broker* de mensagens *Apache ActiveMQ*. Como prova de conceito, uma aplicação para disponibilização de informações em ambientes acadêmicos também foi desenvolvida a partir da *Application Programming Interface* (API) do IPSS.

O restante deste trabalho está estruturado como se segue. A Seção 2 apresenta os conceitos de computação ubíqua, *smart spaces* e *middleware* para computação ubíqua, nos quais este trabalho está fundamentado. A Seção 3 descreve a arquitetura do IPSS, bem como os serviços oferecidos. A Seção 4 apresenta, como prova de conceito, uma aplicação que utiliza a API do IPSS para disponibilização de informações em um *smart space* que representa uma instituição de ensino. A Seção 5 trata os trabalhos relacionados. Por fim, na Seção 6, são apresentadas as conclusões do trabalho realizado, além das perspectivas de trabalhos futuros.

## 2. Fundamentos

Esta seção apresenta brevemente alguns dos fundamentos teóricos nos quais um *middleware* para computação ubíqua se baseia.

### 2.1. Computação Ubíqua

O termo e a ideia de Computação Ubíqua tomaram forma a partir da visão de Mark Weiser [Weiser 1991], o qual vislumbrou a possibilidade de tornar a utilização da computação invisível ao usuário, fundindo-a com elementos do dia a dia, ou seja, fazendo com que o usuário não precise perceber a tecnologia para aproveitar seus benefícios. Segundo este conceito, a computação estaria permeada nos objetos do ambiente físico do usuário, não requerendo dispositivos computacionais tradicionais para a interação, tais como teclado e *mouse*. Segundo Weiser, na computação ubíqua o foco do usuário sai do dispositivo computacional que ele manipula e passa para a tarefa ou a ação a ser realizada.

### 2.2. Smart Spaces

Um *Smart Space* é uma área física delimitada, como uma sala, prédio ou bloco, permeada com serviços computacionais que orquestram a infraestrutura existente no ambiente [Coulouris et al. 2011, Román et al. 2002, Singh et al. 2006]. Por exemplo, foi utilizado para este trabalho o ambiente de instituições de ensino como *smart space*, onde as informações devem chegar de forma transparente aos dispositivos dos usuários, que podem ser dos tipos aluno e professor.

Para gerenciar, organizar e configurar os serviços que permeiam os *smart spaces*, são utilizadas as plataformas de *middleware* [Román et al. 2002, Sousa and Garlan 2002, Xu et al. 2003]. O foco principal dessas plataformas é fornecer serviços que integrem o usuário ao ambiente, especificando e distribuindo serviços pela infraestrutura do espaço inteligente com o objetivo de realizar a imersão do usuário no ambiente físico [Junior et al. 2013].

### 2.3. *Middleware* para Computação Ubíqua

O termo *middleware* se refere ao *software* e às ferramentas que objetivam esconder a complexidade e heterogeneidade do *hardware* e plataformas de rede subjacentes, além de facilitar o gerenciamento de recursos do sistema e aumentar a previsibilidade das execuções da aplicação [Raychoudhury et al. 2012].

O objetivo de um *middleware* para computação ubíqua é combinar as necessidades da aplicação com os recursos disponíveis, garantindo assim, qualidade e eficiência na utilização destes recursos.

A utilização de uma plataforma de *middleware* para computação ubíqua deve apresentar meios de permitir interconectar, gerenciar e orquestrar todos os dispositivos, tecnologias e funcionalidades presentes no *smart space*, além de garantir que problemas que porventura venham a ocorrer sejam tratáveis.

## 3. IPSS: Information Provider for Smart Spaces

O IPSS é um *middleware* que se vale das características da computação ubíqua, tais como mobilidade e sensibilidade ao contexto, para fornecer serviços de distribuição de informações em *smart spaces*. As próximas subseções detalham a sua arquitetura.

### 3.1. Distribuição de Informações em *Smart Spaces*

A plataforma IPSS visa implementar duas primitivas de distribuição: *disponibilizar e obter informações*. Para que a comunicação ocorra e, por conseguinte, a distribuição de informações, deve haver um emissor e, pelo menos, um receptor definidos. No escopo deste trabalho, o emissor é caracterizado como *publicador* e os receptores como *assinantes*. O papel do publicador é desempenhar a primitiva de *disponibilizar informações*, enquanto o papel do assinante se refere à primitiva *obter informações*. Toda informação é enviada por um publicador para um determinado tópico criado previamente. Apenas os assinantes que optaram por obter informações deste tópico, e o fizeram antes do envio da informação por parte do publicador, a receberão. Um publicador pode disponibilizar informações para um ou mais tópicos, enquanto que um assinante pode obter informações de um ou mais tópicos.

Para o IPSS, **informação** é qualquer tipo de mensagem que se queira transmitir, sendo esta manipulável apenas pela aplicação que utiliza o *middleware*. Em outras palavras, a informação transmitida é opaca à plataforma de *middleware* e sua função se restringe a fazer com que esta mensagem transite do publicador para o(s) assinante(s).

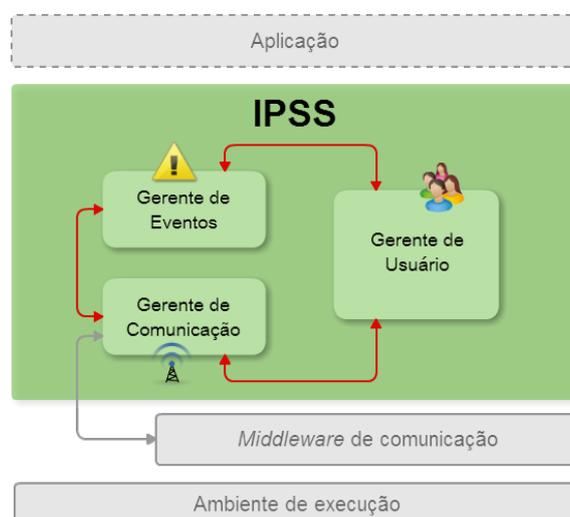
Dentre os tipos de mensagens que podem ser transmitidas pelo IPSS, encontram-se: mensagens de texto plano, arquivos XML, *slides* de apresentações (e.g. PPT, PPTX, ODP), arquivos PDF, dentre outros, além de qualquer outro objeto serializável.

### 3.2. Arquitetura

A arquitetura definida para o IPSS é baseada no modelo de notificação de eventos *publish/subscribe*, onde os dispositivos dos usuários podem tanto se inscrever como publicadores, disponibilizando informações em um *broker de mensagens*, como também se inscrever como assinantes, obtendo assim as informações publicadas.

Na arquitetura do IPSS, os serviços são oferecidos por gerentes, que colaboram entre si, implementando funcionalidades específicas no âmbito da distribuição de informações. Essa separação visa tornar a arquitetura proposta modular e flexível, através do desacoplamento dos serviços, de maneira que possam ser oferecidos por gerentes independentes. O conjunto dos serviços e, por consequência, dos gerentes, compõe a plataforma IPSS.

A visão geral da arquitetura do IPSS é mostrada na Figura 1. Ao centro, o *middleware* IPSS, composto pelos gerentes de usuário, de eventos e de comunicação. Na camada imediatamente abaixo do IPSS, encontra-se o *middleware* de comunicação, que fornece suporte ao modelo *publish/subscribe*. Abaixo do *middleware* de comunicação encontra-se o ambiente de execução, e, acima do IPSS, as aplicações que utilizam sua API.



**Figura 1. Visão geral da arquitetura do IPSS.**

Os gerentes situam-se entre a aplicação e o ambiente de execução da plataforma do dispositivo, bem como sobre um *middleware* de comunicação, que faz a interligação entre a plataforma IPSS e o *broker*. O *middleware* de comunicação tem como objetivo fornecer a infraestrutura para o envio de informações, seja dos dispositivos dos usuários para o *broker* ou do *broker* para os dispositivos. Por meio dos serviços oferecidos pelo conjunto de gerentes e do *middleware* de comunicação, o IPSS oferece uma interface de alto nível para o envio e recebimento de informações em *smart spaces*.

Foram definidos três gerentes para o IPSS: gerente de comunicação, gerente de usuário e gerente de eventos. Os serviços oferecidos por estes gerentes são especificados a seguir:

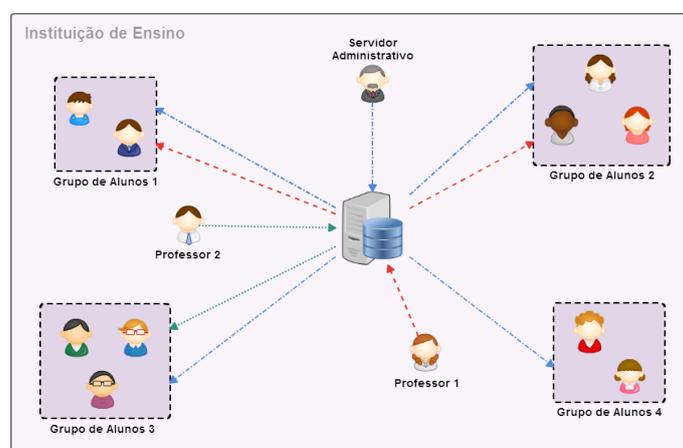
- **Gerente de Comunicação:** Permite que os usuários enviem e recebam informações dos tópicos nos quais são, respectivamente, publicadores e assinantes. É responsável pela conexão entre o IPSS e o *broker* de mensagens, sendo que essa conexão é realizada através de um *middleware* de comunicação externo ao IPSS. O *middleware* de comunicação é o responsável por fornecer a base do modelo de eventos, permitindo dessa forma a implementação de publicadores e assinantes e a entrega confiável de mensagens.

- Gerente de Usuário:** Fornece operações para a criação e exclusão de usuários que podem ser do tipo *publicador* ou *assinante*. Para os usuários publicadores, são associados os tópicos do *broker* de mensagens, nos quais estes poderão publicar. Para usuários do tipo assinante, são definidos os tópicos de interesse, cujos eventos serão monitorados por uma interface associada ao assinante. Cabe à aplicação que utilizar a API do IPSS, definir se o usuário poderá assumir a função de publicador ou assinante, ou eventualmente as duas.
- Gerente de Eventos:** Fornece uma interface para monitoramento de eventos no *broker* de mensagens, notificando as aplicações quando há novas publicações nos tópicos do *broker* assinados por um usuário. O gerente de eventos atua de forma específica sobre os usuários do tipo assinante, monitorando seus tópicos de interesse e notificando quando há novos eventos (publicação de informações) nestes tópicos de forma *assíncrona*, ou seja, caso o usuário não esteja conectado no momento da publicação ele será notificado na primeira oportunidade em que este se conectar.

#### 4. Prova de Conceito

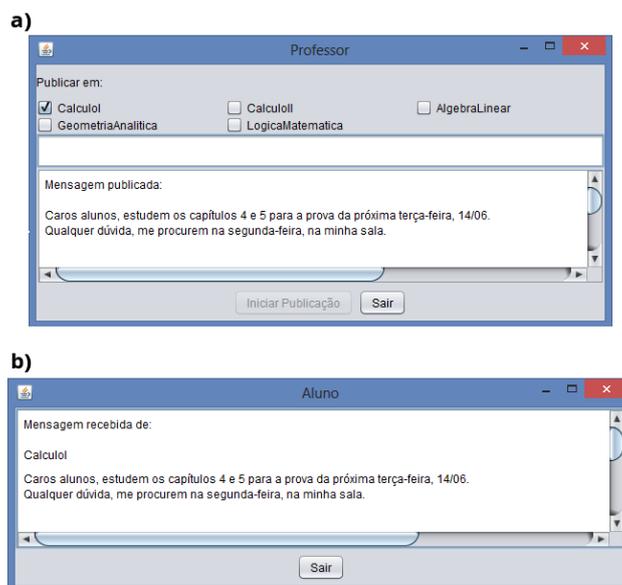
A plataforma de *middleware* IPSS foi implementada utilizando a linguagem de programação Java e o *middleware* de comunicação *Java Message Service* (JMS). Como *broker* de mensagens, oferecendo a base para o envio e recebimento de informações, foi utilizado o *Apache ActiveMQ*.

Como prova de conceito, foi desenvolvida uma aplicação para distribuição de informações em ambientes acadêmicos, utilizando a linguagem Java e a API do IPSS. Ela se baseia no cenário apresentado pela Figura 2, em que um *smart space* representa uma instituição de ensino na qual existem três usuários publicadores, sendo dois professores e um servidor administrativo. O Professor 1 publica uma mensagem em um tópico do *broker* de mensagens que é assinado por alunos dos Grupos 1 e 2. Já o Professor 2, publica em um tópico assinado apenas por alunos do Grupo 3. O Grupo de Alunos 4 recebe somente as mensagens do Servidor Administrativo, que publica em um tópico de informações diversas da instituição, assinado por todos os alunos.



**Figura 2. Cenário de aplicação do IPSS utilizando a aplicação para distribuição de informações em instituições de ensino.**

A Figura 3(a) mostra a interface gráfica da aplicação do usuário publicador, neste caso um professor, que permite a disponibilização de informações. A Figura 3(b) apresenta a interface gráfica da aplicação do aluno, no momento em que esta recebe uma notificação de publicação no tópico Cálculo I. Nesta aplicação, o professor pode escolher um ou mais tópicos para publicar as informações, que serão recebidas pelos alunos que assinam este tópico. Neste caso em particular, a aplicação está sendo utilizada por um professor de matemática que ministra diversas disciplinas. Essa figura mostra uma publicação realizada no tópico referente à disciplina de Cálculo I.



**Figura 3. Telas da aplicação no dispositivo do aluno (a) e do professor (b).**

A aplicação desenvolvida trabalha de maneira assíncrona, ou seja, os usuários assinantes recebem as informações publicadas nos seus tópicos de interesse no momento em que se conectam ao *broker*, não necessitando estar conectados no momento da publicação.

## 5. Trabalhos Relacionados

O foco do *middleware* IPSS é o compartilhamento de informações em ambientes ubíquos. Nesta seção são discutidos alguns trabalhos relacionados ao *middleware* IPSS.

Gaia [Román et al. 2002] é um *middleware* que visa orquestrar os serviços de um *smart space* através do gerenciamento e coordenação dos recursos, entidades de *software* e dispositivos heterogêneos que o compõem. Este é um artigo clássico da área, pois concretiza a visão de ubiquidade proposta por Mark Weiser [Weiser 1991].

Em [Mareli et al. 2013], os autores propõem um *framework* que oferece abstrações para facilitar o desenvolvimento, teste e execução de aplicações ubíquas em *smart spaces*, além de lidar com problemas recorrentes da área, tais como: *i)* heterogeneidade dos dispositivos; *ii)* variedade de informações de contexto e serviços; e *iii)* disponibilidade de recursos. A fim de testar a proposta, foi utilizada uma plataforma chamada *SmartAndroid*.

O *Ubiquitous Computing for Learning Environments* (UCLE) [Junior et al. 2012] é um *middleware* de movimentação de conteúdo que se baseia em três primitivas prin-

cipais: mover, clonar e espelhar. Estas primitivas tornam transparentes as tarefas de compartilhamento de conteúdo em aplicações colaborativas, ao abstrair problemas como modelos diferentes de comunicação, concorrência e representação externa do conteúdo.

O *Multimodal Interagent Communication Architecture* (MICA) [Kadous and Sammut 2004] é um *middleware* para computação ubíqua que tem por objetivo o compartilhamento de informações, a aprendizagem das preferências e a interação entre os usuários através de diversos dispositivos e modalidades. O MICA trabalha com o conceito de quadro-negro, que é uma estrutura global e centralizada, por onde todas as interações entre os usuários passam.

O Gaia e o *framework* apresentado por Mareli *et al.* introduzem um mecanismo mais abrangente para o gerenciamento dos dispositivos que compõem o *smart space* e as interações entre estes dispositivos. Contudo, o segundo trabalho possui um viés de automação e monitoramento de ambientes, que se assemelha ao IPSS no sentido de que as ações são disparadas somente se as regras definidas pelos usuários forem satisfeitas. Em outras palavras, os usuários só serão afetados por ações que lhe forem pertinentes, assim como ocorre com os usuários do IPSS que somente recebem informações em tópicos assinados previamente por eles.

O UCLE e o MICA apresentam *middlewares* para compartilhamento de conteúdo em *smart spaces*. O UCLE define três primitivas para movimentação de conteúdo e apresenta uma arquitetura que as tornam possível. O MICA possibilita que seus agentes recebam notificações sobre objetos de seu interesse colocados no quadro-negro. Esta característica se assemelha ao modelo proposto pelo IPSS.

A principal vantagem que o IPSS traz é a transparência para o desenvolvedor, no que diz respeito especificamente à construção de aplicações baseadas nas primitivas *obter* e *disponibilizar* informações. Isso possibilita a programação em alto nível para a criação de usuários publicadores e assinantes, e para a utilização dos serviços de disponibilizar e obter informações em ambientes ubíquos. A API do IPSS oculta do desenvolvedor a complexidade e os detalhes da implementação da plataforma adjacente, tais como as tecnologias utilizadas para interconectar os publicadores e os assinantes, o *middleware* de comunicação e o sistema de *broker*, permitindo que seu foco seja orientado para a lógica de negócio da aplicação.

## 6. Conclusão

O *Information Provider for Smart Spaces* (IPSS) é uma plataforma de *middleware* para distribuição de informações em *smart spaces*. A arquitetura apresentada neste artigo é composta por três gerentes que, de forma conjunta, oferecem os serviços do IPSS: *Gerente de Usuário*, *Gerente de Comunicação* e *Gerente de Eventos*.

Para validar a plataforma de *middleware* proposta, foi desenvolvida uma aplicação voltada para a distribuição de informações em instituições de ensino. Esta aplicação utiliza os serviços e primitivas do IPSS por meio da sua API, para definir usuários publicadores, como professores e servidores administrativos, e usuários assinantes, como alunos.

O próximo passo neste trabalho refere-se ao desenvolvimento de um *Gerente de Contexto*, responsável pela captura, processamento e disponibilização de informações de contexto provenientes do ambiente físico. Dentre as informações que poderiam ser

disponibilizadas, encontram-se: *i*) a localização do usuário; *ii*) os recursos dos dispositivos; e *iii*) os dados de sensores e atuadores que constituem o *smart space*. Essas informações de contexto poderiam ser utilizadas para desenvolver aplicações que se adaptem às características particulares de cada usuário ou dispositivo.

Outro trabalho futuro relevante seria a avaliação do IPSS em diferentes cenários, tanto em pequena quanto em larga escala. A partir desta avaliação, métricas poderiam ser definidas para a comparação do modelo proposto com outros modelos disponíveis, como, por exemplo, comparar o tempo, a facilidade de desenvolvimento e a utilização de uma aplicação concebida com base nos serviços disponibilizados pelo IPSS, com a mesma aplicação construída utilizando diretamente os serviços do JMS.

## Referências

- Coulouris, G. F., Dollimore, J., Kindberg, T., and Blair, G. (2011). *Distributed Systems: Concepts and Design*. International computer science series. Addison-Wesley.
- Hlavac, R. and Schaefer, M. (2013). The social media revolution as theorized by “the cluetrain manifesto”. *Journal of Integrated Marketing Communications*.
- Junior, M. P. R., Freitas, L. A., Massarani, M. A. L., Rocha, R. C., and Costa, F. M. (2012). Ucle: Um middleware de computação ubíqua para compartilhamento de conteúdo em salas de aula inteligentes. *Simpósio Brasileiro de Computação Ubíqua e Pervasiva – SBCUP*.
- Junior, M. P. R., Massarani, M. A. L., Freitas, L. A., da Rocha, R. C. A., and Costa, F. M. (2013). C3s: a content sharing middleware for smart spaces. *Workshop on Pervasive Collaboration and Social Networking*.
- Kadous, M. and Sammut, C. (2004). Mica: pervasive middleware for learning, sharing and talking. In *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, pages 176–180.
- Mareli, D., Erthal, M., Barreto, D., and Loques, O. (2013). Um framework de desenvolvimento de aplicações ubíquas em ambientes inteligentes. *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos – SBRC*, pages 643–656.
- Raychoudhury, V., Cao, J., Kumar, M., and Zhang, D. (2012). Middleware for pervasive computing: A survey. *Pervasive and Mobile Computing*.
- Román, M., Hess, C., Cerqueira, R., Campbell, R. H., and Nahrstedt, K. (2002). Gaia: A middleware infrastructure to enable active spaces. *IEEE Pervasive Computing*, 1:74–83.
- Singh, R., Bhargava, P., and Kain, S. (2006). State of the art smart spaces: application models and software infrastructure. *Ubiquity*, 2006(September):7.
- Sousa, J. P. and Garlan, D. (2002). Aura: an architectural framework for user mobility in ubiquitous computing environments.
- Weiser, M. (1991). The computer for the 21st century. *Scientific american*, 265(3):94–104.
- Xu, B., Lian, W., and Gao, Q. (2003). Migration of active objects in proactive. *Information and Software Technology*, 45(9):611–618.