

Engenharia de requisitos aplicada ao contexto de softwares ubíquos

Júlia Carneiro de Almeida

Orientador - Alex Sandro Gomes



Agenda

- **Principais referências;**
- **Principais autores;**
- **Introdução;**
- **Computação ubíqua - Sobre o paradigma:**
 - Definição;
 - Objetivos;
 - Características;
 - Desafios;
 - Aplicações ubíquas;
- **Computação ubíqua e Engenharia de requisitos:**
 - Revisão sistemática;
 - Análise das Abordagens;
- **Conclusão.**





Principais Referências

Artigos essenciais para o realização da pesquisa

Principais Referências

- [1] SPÍNOLA, R.O. (2010). **“Apoio à Especificação e Verificação de Requisitos Funcionais de Ubiquidade em Projetos de Software”**. Tese de Doutorado. COPPE/UFRJ. Rio de Janeiro.
- [2] SPÍNOLA, R. O.; TRAVASSOS, G. H.; MASSOLLAR, J. **“Checklist to Characterize Ubiquitous Software Projects”**. SBES 2007 - XXI Simpósio Brasileiro de Engenharia de Software, 2007.
- [3] PINTO, F. C. do R. **“Uma Abordagem Para Apoiar Especificação De Requisitos Para Projetos De Software Ubíquo”**. Dissertação de Mestrado. COPPE/UFRJ. Rio de Janeiro, 2009.
- [4] MOTA, L. da S. **“Uma Abordagem Para Especificação De Requisitos Funcionais De Ubiquidade Em Projetos De Software”**. 2013. 149 p. Dissertação (Mestrado) - UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação - Rio de Janeiro, 2013
- [5] WEISER, M. **“The computer for the 21st century”**. ACM SIGMOBILE mobile computing and communications review, vol 3(3). ACM Press, New York, p. 3–11, 1999.





Principais Autores

Apresentação dos autores dos artigos utilizados

Rodrigo Oliveira Spínola



É **Bacharel em Ciência da Computação** pela Universidade Salvador (2001), **Doutor e Mestre em Engenharia de Sistemas e Computação** pela COPPE/UFRJ (2010 e 2004 respectivamente).

Atualmente, é **professor titular** do **Programa de Pós-Graduação em Sistemas e Computação da Universidade Salvador**.

Sua principal área de pesquisa é Dívida Técnica, fundamentando as atividades realizadas nos conceitos da engenharia de software experimental.

Além disso, possui experiência na área de engenharia de software, tal que participou de diversos projetos de consultoria nas áreas de gerência de projetos, processos, requisitos e testes de software.

Fonte: Lattes [\(Link\)](#)



Guilherme Horta Travassos



Engenheiro Eletricista (Universidade Federal de Juiz de Fora, 1985), **Mestre** (1990) e **Doutor** (1994) em **Engenharia de Sistemas e Computação** pela COPPE/Universidade Federal do Rio de Janeiro. **Pós-Doutorado** na University of Maryland-College Park juntamente com o SEL/NASA (USA - 98/00).

Sua **linha de pesquisa** tem como focos principais experimentação aplicada na **Engenharia de Software e linha de produto de software**.

Atualmente, é **professor titular** (PESC/COPPE/UFRJ).

Atua nos seguintes temas: qualidade de software (VV&T), ciência em larga escala, engenharia de aplicações WEB e **sistemas ubíquos**, simulação em engenharia de software, ambientes e ferramentas para apoiar o desenvolvimento e experimentação em software.

Fonte: Lattes ([Link](#))



Jobson Luiz Massollar da Silva



Bacharel em Informática pela **Universidade Federal Fluminense** (1990), **Mestre em Engenharia de Sistemas e Computação** pela **COPPE/UFRJ** (1993) e **Doutor em Engenharia de Sistemas e Computação** pela **COPPE/UFRJ** (2011) com ênfase em Engenharia de Software.

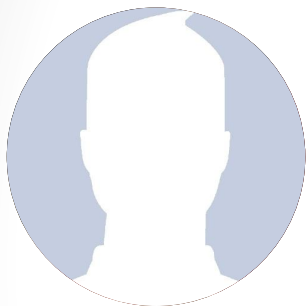
Na área de Engenharia de Software tem atuado nos seguintes temas: Engenharia de Aplicações Web, Engenharia de Software Experimental, Desenvolvimento Orientado por Modelos, Verificação e Validação de Sistemas e **Sistemas Sensíveis ao Contexto**.

Desde 2011 é professor da Universidade Veiga de Almeida, onde leciona nos cursos de graduação em Ciência da Computação e Engenharia de Computação.

Fonte: Lattes [\(Link\)](#)



Leonardo da Silva Mota



Graduado em **Sistemas de Informação**. Aluno de especialização em Desenvolvimento de Aplicações para Web. **Mestre** em **Engenharia de Sistemas e Computação** na **COPPE/UFRJ**.

Tem experiência na área de Ciência da Computação, com ênfase em Engenharia de Software, linguagens de programação e programador certificação Java (SCJP).

Fonte: Lattes [\(Link\)](#)



Mark Weiser



Conhecido como o **pai da Computação Ubíqua**, criou o conceito de **Computação Ubíqua** na publicação do artigo “A computação do século 21” no final dos anos 80.

Weiser estudou **Informática e Ciências da Comunicação na Universidade de Michigan** e um **Ph.D.** em 1979. Lecionou na Universidade de Maryland, College Park e foi cientista chefe(1988) diretor de tecnologia (1996) da Xerox PARC.

Weiser **defendeu o oposto da Realidade Virtual**, em que as pessoas são colocadas dentro do mundo gerado pelo computador.

Fonte: [Wikipédia](#)





Introdução

Serão apresentados a motivação e problemas identificados.

Introdução

Em consequência aos avanços tecnológicos, o **número de dispositivos aumentou**. Em consequência desses avanços, **surgiu a computação ubíqua** e com ela a **necessidade de técnicas, métodos e suporte** da engenharia de software e engenharia de requisitos para a realização das atividades de **desenvolvimento e concepção** de um software ubíquo.

O presente trabalho objetivou-se em **identificar** através da literatura abordagens que **destacam preocupações e soluções** para o **processo de especificação de requisitos de softwares ubíquos**.





O que é a computação ubíqua?

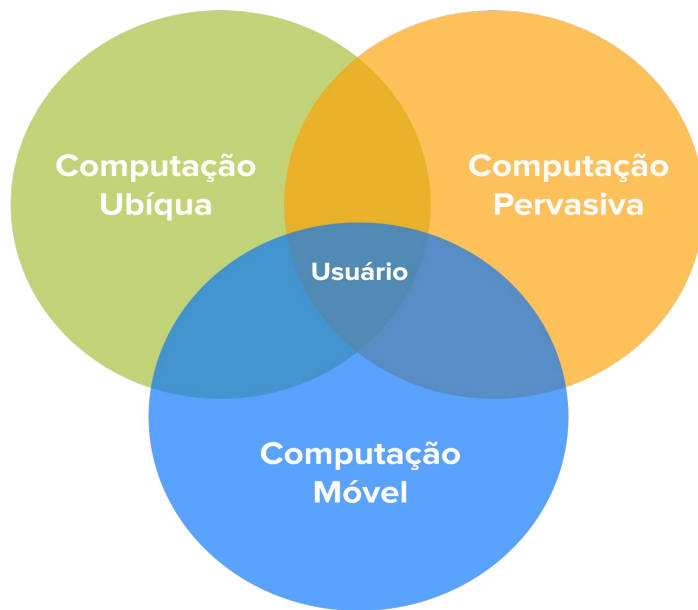
Sobre o paradigma

Definição

A Computação Ubíqua está presente em todos os **serviços ou facilidades computacionais** que são disponibilizados sem que o computador seja uma ferramenta **visível** ou **imprescindível** para acesso desses serviços. Ou seja, esses serviços ou facilidades podem **se materializar em qualquer momento ou lugar**, de forma **transparente**, através do **uso de dispositivos de uso comum** no dia-a-dia (SPÍNOLA et. al., 2006).



Definição



Definição

De acordo com Mark Weiser (1991), diferentes modos de pensar sobre computadores no mundo estão sendo apresentados, tornando **um ambiente conectado** tão **natural** que permite que **os próprios computadores não** sejam **percebidos** como algo **novo** e de **grande impacto** no ambiente.



Características da computação ubíqua

As características das aplicações ubíquas podem **influenciar** no **processo de desenvolvimento**, aumentando a **complexidade** do desenvolvimento do software (SPÍNOLA et al., 2007a):

- Onipresença de serviço;
- Invisibilidade;
- Sensibilidade ao contexto;
- Comportamento adaptável;
- Captura de experiências;
- Descoberta de serviços;
- Composição de funcionalidades;
- Interoperabilidade Espontânea;
- Heterogeneidade de dispositivos;
- Tolerância a falhas;



Desafios da computação ubíqua

Spínola (2010), ainda destaca relacionados aos **processos** que envolve todo o **ciclo de vida** do software, seja eles na:

- **Definição do processo;**
- **Planejamento do projeto;**
- **Especificação dos requisitos;**
- **Escolha das tecnologias** que devem ser utilizadas;
- Realização dos **testes**.



Desafios da computação ubíqua

Como pode ser observado os **desafios** normalmente são diretamente ligados a alguma **característica** dos softwares ubíquos:

- Heterogeneidade;
- Escalabilidade;
- Confiabilidade;
- Privacidade;
- Interoperabilidade espontânea;
- Conhecimento e gerenciamento do contexto; (COSTA, YAMIN e GEYER, 2008);
- Interoperabilidade;
- Modularização;
- Desenvolvimento Globalizado;
- Dependabilidade;
- Segurança; (ROCHA, et al, 2011);
- Integração;
- Adaptação;
- Descoberta;
- Estrutura de programação;
- Robustez (KINDBERG e FOX, 2002).



Desafios da computação ubíqua

Descoberta (KINDBERG E FOX, 2002):

Está relacionado as ações que ocorrem a partir da **identificação** de um **dispositivo**, surgindo discussões sobre:

- Quando um dispositivo entra em um ambiente, como ocorre a descoberta mútua entre ele, outros serviços e dispositivos disponíveis?
- Quais interações são apropriadas para aquele novo dispositivo?



Desafios da computação ubíqua

Dependabilidade (ROCHA et al, 2011):

Define a capacidade dos sistemas computacionais de **prestar um serviço confiável**, podendo ser decomposto nos atributos de **disponibilidade, confiabilidade, segurança, confidencialidade, integridade e manutenibilidade**.



Aplicações ubíquas

- Plataformas ubíquas podem ser encontradas e aplicadas a diversos contextos:
 - Assistência domiciliar à **saúde** (CARVALHO, COPETTI E LOQUES FILHO, 2011);
 - Área educacional através de **plataformas de u-learning** (MONTEIRO, 2015);
 - Sistema de **aprendizagem** ubíquo para **museus**, o BCAULS (CHIU, TSENG, HSU, 2017).





Computação ubíqua e Engenharia de requisitos

Computação ubíqua e Engenharia de requisitos

Diante desse novo paradigma a **engenharia de software** necessitou **evoluir**, para que seus processos pudessem **apoiar** o desenvolvimento de diferentes categorias de software, dentre eles os softwares ubíquos.

Além disso, desperta-se a **preocupação sobre os requisitos**, já que eles possuem um papel central no processo de **desenvolvimento**, pois geram a **especificação das estimativas, modelagem, projeto, implementação e testes**.



Computação ubíqua e Engenharia de requisitos

De acordo com Spínola (2010), é de suma importância **entender** as **características de ubiquidade**, pois elas podem **influenciar** no **processo de desenvolvimento** do software. Entretanto, precisa-se de um processo bem definido para apoiar as atividades do desenvolvimento.



Revisão sistemática (SPÍNOLA, 2010)

Objetivo: identificar abordagens que **apoiem a definição** e a **verificação de requisitos de ubiquidade** em projetos de software.



Fontes de busca e critérios (SPÍNOLA, 2010)

- **Fontes de busca:**

- Biblioteca digital IEEEExplorer;
- Biblioteca digital da ACM Portal;
- Proceedings do UBICOMP;
- International Requirements Engineering Conference;
- Possíveis referências identificadas nos artigos selecionados para

revisão.

- **CrITÉrios de Inclusão e Exclusão de artigos:**

- Artigos disponíveis na web;
- Artigos em inglês;
- Os artigos devem contemplar abordagens para definição ou verificação de requisitos de ubiquidade em projetos de software ubíquo com embasamento científico ou que foram experimentalmente caracterizadas.



Resultados obtidos



Poucas investigações
que apoiem o
desenvolvimento.



Aumenta as
dificuldades e os
riscos



Estratégias para especificação e documentação de requisitos para *softwares* ubíquos

Através da revisão da literatura realizada por Spínola (2010), verificou-se a existência de **9 abordagens** e recomendações para o processo de identificação e especificação de requisitos:

Contexto	Trabalhos Identificados	Abordagem Proposta	Características de Ubiquidade
Definição de requisitos do usuário com base em uma ontologia	(BO et al., 2007)	Identificação	Sensibilidade ao contexto
Metodologia para definir requisitos de sistemas adaptáveis e sensíveis ao contexto	(CHIU et al., 2007)	Identificação e especificação	Sensibilidade ao contexto
Exemplos de requisitos de ubiquidade do projeto ENQUETE-BAISE	(CHENG et al., 2007)	Especificação	Não especificado
Exemplos de requisitos de ubiquidade de um sistema multimodal multimídia	(CHERIF et al., 2007)	Especificação	Não especificado
Mecanismo para definição de requisitos de serviços	(XIANG et al., 2007)	Identificação	Onipresença de Serviços
Metodologia para definir	(HONG et al.,	Identificação e	Sensibilidade ao

requisitos de sistemas adaptáveis e sensíveis ao contexto	2005)	especificação	contexto
Modelagem de requisitos com a linguagem KAOS	(GOLDSBY e CHENG, 2007) (CHENG et al., 2005)	Identificação e especificação	Sensibilidade ao contexto e Comportamento adaptável
Casos de uso executáveis	(JORGENSEN e BOSSEN, 2003)	Especificação	Não especificado
Processo sistemático para análise de requisitos de segurança em sistemas embarcados	(MARKOSE et al., 2008)	Identificação e especificação	Não especificado
Processo para definição de requisitos com base nas intenções dos usuários	(OYAMA et al., 2008)(MING et al., 2008)	Identificação	Não especificado

Casos de Uso Executáveis (JORGENSEN e BOSSEN, 2003)

Compreende em uma **abordagem** para **definição de requisitos** considerando o contexto de **desenvolvimento** de um **sistema pervasivo de acompanhamento médico** (PHCS – *Pervasive Health Care System*).

O fluxo é composto por de três camadas:

- **Primeira camada: conhecimento** sobre o **domínio** do problema (definir através dos **casos de uso**);
- **Segunda camada:** as interações entre homem-máquina devem ser **modeladas** (diagrama de estados, diagrama de atividades, entre outros);
- **Terceira camada:** são criados os **protótipos** para validar e verificar os requisitos especificados.



Modelagem de Requisitos com a Linguagem KAOS (GOLDSBY e CHENG, 2007)

É uma abordagem aplicada a **sistemas dinamicamente adaptáveis (DAS)**, a qual é **apoiada** pela linguagem de especificação **KAOS**.

Tal abordagem propõem o uso de **quatro níveis** de atividades:

- **Nível 1:** identificar os requisitos de maneira **tradicional**;
- **Nível 2:** identificar as **variáveis do ambiente**, para realizar as adaptações;
- **Nível 3:** definir a partir do domínio, **qual técnica de adaptação** utilizar;
- **Nível 4:** pesquisar sobre a técnica para **criar novos mecanismos** de adaptação.

A abordagem **considera** as **características** sensibilidade ao contexto e comportamento adaptável e fornece através dos níveis dos requisitos, conhecimento sobre as informações que precisam ser capturadas.



Metodologia para Definir Requisitos de Sistemas Adaptáveis e Sensíveis ao Contexto (CHIU et al., 2007)

Apresentam uma abordagem para **auxiliar** na **identificação de requisitos de software ubíquo** cujas **características** principais são:

- Estar **disponível** na web;
- Ser **sensível ao contexto**.

Para isto, eles objetivam-se em garantir a **usabilidade e experiência do usuário**.

Dessa forma, ele propõe um passo a passo para auxiliar na **identificação de requisitos e projeto do software**, levando em consideração, **o contexto, características e prototipagem**.



Definição de Requisitos do Usuário com base em uma Ontologia (BO et al., 2007)

Compreende em uma abordagem para **identificação de requisitos** baseada em **ontologia**. Esta é a base semântica para a estruturação dos requisitos de maneira que permita-se **identificar e definir o contexto** no qual cada requisito estará associado.

A abordagem é composta por **duas etapas**:

- Construir a **ontologia** de requisitos;
- Realizar as seguintes atividades para **construir uma árvore de requisitos**:
 - **Decompor** a ontologia de requisitos em sub-requisitos;
 - **Repetir a atividade** anterior até que eles não possam mais ser decompostos;
 - **Construir uma árvore de requisitos** para conectar os requisitos definidos na ontologia com todos os seus sub-requisitos.



Mecanismo para Definição de Requisitos de Serviços (XIANG et al., 2007)

Os autores sem sua abordagem apresentam o **SREM** (*Service Requirement Elicitation Mechanism*). Este é formado por três componentes principais:

- **SRMO** (*Service Requirement Modeling Ontology*): define a **ontologia** para a modelagem de requisitos;
- **SREP** (*Service Requirement Elicitation Process*): direciona o **processo** para **elicitar detalhes de requisitos e informações estruturais**, onde é proposto um conjunto de questionamentos;
- **SRRP** (*Service Requirement Reconciliation Process*): utilizado para **integrar as instâncias** dos requisitos identificados a partir dos diferentes usuários.

Esse conhecimento é utilizado para construir um **repositório de conhecimento de requisitos**.



Exemplos de Requisitos de Ubiquidade do Projeto ENQUETE-BAISE (CHENG et al., 2007)

Compreende em um **servidor de questionários** para funcionar em ambientes computacionais ubíquos.

Abordagem **não** apresentam uma **técnica** para **definição** ou **verificação de requisitos**, apenas listam alguns requisitos definidos para o projeto.

Se resume nas atividades descritas abaixo:

1. **Listar as características genéricas** do software;
2. Definir **o conjunto de requisitos funcionais** (os chamados de requisitos **concretos**, pelos autores) associados a estas características.



Exemplos de Requisitos de Ubiquidade de um Sistema Multimodal Multimedia (CHERIF et al., 2007)

Os autores apresentam o desenvolvimento de um **sistema multimodal multimedia tolerante a falhas e ubíquo**. Tais autores **não** apresentam uma **técnica para definição ou verificação de requisitos**.

A abordagem apenas possui a seguinte atividade:

- Listar os **possíveis requisitos** para o sistema;

Não há um **detalhamento adicional** e **nem uma preocupação** diferenciada pelo fato de se tratar de um sistema ubíquo.



Processo Sistemático para Análise de Requisitos de Segurança em Sistemas Embarcados (MARKOSE et al., 2008)

Os autores propõem uma **análise** sobre os **requisitos de segurança em sistemas embarcados**.

Dessa forma, realiza-se as seguintes atividades:

1. Desenvolver um **modelo de objetos** de contexto para mostrar as relações do sistema com componentes externos.
2. **Representar os subcomponentes** de cada objeto;
3. **Especificar através de casos de uso** as principais funcionalidades de cada objeto;
4. **Identificar e tratar os casos de uso** que podem trazer **problemas de segurança**.



Processo para Definição de Requisitos com base nas Intenções dos Usuários (OYAMA et al., 2008)(MING et al., 2008)

Compreende em um processo para **definição de requisitos** com base nas **intenções dos usuários**. O processo objetiva-se em descobrir os **desejos do usuário** e **sugerir alterações no sistema**.

Dessa forma, são propostas as seguintes atividades:

1. Observar padrões de **comportamento dos usuários**, por exemplo, a frequência de uso do sistema;
2. Observar **relatos do usuário**, por exemplo, defeitos encontrados e comentários sobre o sistema;
3. Associar os **padrões de comportamento e relatos do usuário** aos requisitos do sistema e alterá-los de acordo com a necessidade.

Por mais que a abordagem **não trate** de comportamento ubíquo, ela **auxilia** na **percepção do comportamento humano** sobre a aplicação.



Análise das Abordagens Identificadas na Literatura

- **Adoção de projetos interativo** no desenvolvimento de um software;
- **Uso de ontologias** para apoiar as atividades de identificação dos requisitos;
- Utilização de **abordagens adaptativas**;
- As características de ubiquidade consideradas foram **comportamento adaptável e sensibilidade ao contexto**;



Análise das Abordagens Identificadas na Literatura

- A **ausência** de abordagens para lidar com a **verificação dos requisitos de ubiquidade**;
- **Não** há a definição de um **processo**, **nem** um **detalhamento** sobre **técnicas** que podem ser utilizadas para **apoiar** a execução de suas **atividades**.





Conclusões

Serão apresentadas as considerações finais.

Conclusões

Antes de **iniciar** o desenvolvimento de um software ubíquo, deve-se estar **ciente do contexto**, como também **entender** suas **características** e os **desafios** que estão atrelados a elas e que podem ser enfrentados.

Ficou claro que parte das abordagens as **características** **geram** tanto **impacto** na **especificação** do software ubíquo;

Entretanto verifica-se uma **lacuna** para estudos futuros, relacionando a **computação ubíqua** e **engenharia de requisitos**.



Dúvidas?

Engenharia de requisitos aplicada ao contexto de softwares ubíquos

Júlia Carneiro de Almeida - jca3@cin.ufpe.br





Referências

Referências

SPÍNOLA, R. O., SILVA, J. L. M, TRAVASSOS, G. H. "Towards a Conceptual Framework to Classify Ubiquitous Software Projects". Proceedings of the 8th International Conference on Software Engineering and Knowledge Engineering (SEKE), San Francisco, USA, 2006.

CARVALHO, S. T. DE; COPETTI, A.; LOQUES FILHO, O. G.. Sistema de computação ubíqua na assistência domiciliar à saúde. J. Health Inform, 2011.

MONTEIRO, B. de S. Ambiente de aprendizado ubíquo youubi: design e avaliação. Tese de Doutorado - UFPE 2015.

CHIU, C.; TSENG, J. C. R.; HSU, T. Blended context-aware ubiquitous learning in museums: environment, navigation support and system development. Personal and Ubiquitous Computing (2017). pp 355–369, 2017.

HONG, D.; CHIU, D. K. W; SHEN, V. Y.; Requirements Elicitation for the Design of Context-aware Applications in a Ubiquitous Environment. ICEC '05 Proceedings of the 7th international conference on Electronic commerce, p. 590-596, Xi'an, China, 2005.



Referências

SPÍNOLA, R.O. (2010). “Apoio à Especificação e Verificação de Requisitos Funcionais de Ubiquidade em Projetos de Software”. Tese de Doutorado. COPPE/UFRJ. Rio de Janeiro.

MOTA, L. da S. Uma Abordagem Para Especificação De Requisitos Funcionais De Ubiquidade Em Projetos De Software. 2013. 149 p. Dissertação (Mestrado) - UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação - Rio de Janeiro, 2013.

KIM, C.; MIRUSMONOV, M. Factors Influencing the Usage of Ubiquitous-Based Mobile Devices. Computing Technology and Information Management (ICCM), 8th International Conference, Seoul, South Korea, 2012.

PINTO, F. C. do R. “Uma Abordagem Para Apoiar Especificação De Requisitos Para Projetos De Software Ubíquo”. Dissertação de Mestrado. COPPE/UFRJ. Rio de Janeiro, 2009.

C. A. da COSTA, A. C. YAMIN, AND C. F. R. GEYER, “Toward a general software infrastructure for ubiquitous computing,”IEEE Pervasive Computing, vol. 7, no. 1, pp. 64–73, 2008.

KINDBERG, T.; FOX, A. System Software for Ubiquitous Computing, IEEE Pervasive Computing archive, v. 1, p. 70-81, 2002.

