

# Uma Contribuição à Reabilitação Cardíaca Explorando a Identificação de Situações na IoT

Douglas Scheunemann<sup>1</sup>, Adenauer Yamin<sup>1</sup>, João Lopes<sup>2</sup>, Cláudio Geyer<sup>2</sup>

<sup>1</sup>Universidade Católica de Pelotas (UCPEL)  
Pelotas – RS – Brasil

<sup>2</sup>Universidade Federal do Rio Grande do Sul (UFRGS)  
Porto Alegre – RS – Brasil

scheunemann.d.a@gmail.com, adenauer.yamin@ucpel.edu.br

{jlblopes, geyer}@inf.ufrgs.br

**Abstract.** *The Internet of Things (IoT) has influenced the development of computational systems, enabling a more proactive interaction with users, expanding features as mobility and availability. In this scenario increases the demand for applications that can recognize the user's context and can provide situation-based services. The identification of situations is a research challenge for applications in the IoT, given the complexity of the relationships that must be established and processed. This paper presents an approach for the identification of situations, which is integrated with the EXEHDA middleware. The proposed approach provides the collaboration between the middleware and the applications. A case study in the health area was developed to evaluate the architecture. This case study was focused on cardiac rehabilitation.*

**Resumo.** *A Internet das Coisas (IoT) vem influenciando a maneira como os sistemas computacionais são desenvolvidos, possibilitando uma interação mais proativa com os usuários, expandindo características de mobilidade e disponibilidade. Nesse cenário, cresce a demanda por aplicações que possam reconhecer o contexto do usuário e fornecer serviços baseados em sua situação. A identificação de situações representa um desafio de pesquisa para aplicações em IoT, dada a complexidade das relações que precisam ser estabelecidas e processadas. Neste trabalho é apresentada uma abordagem de processamento para identificação de situações integrada ao middleware EXEHDA. A abordagem de processamento proposta propicia a colaboração entre o middleware e as aplicações na identificação de situações. Como forma de avaliação da arquitetura foi executado um estudo de caso na área de saúde com foco em reabilitação cardíaca.*

## 1. Introdução

A reabilitação cardíaca pode envolver diversas terapias, incluindo administração de medicamentos, aconselhamento nutricional e também a prescrição de atividades físicas. A reabilitação cardíaca através de atividades físicas é considerada uma terapia central. Estudos indicam que a reabilitação baseada em exercícios físicos foi associada a uma redução de 20 a 30% nas taxas de mortalidade, quando comparada com cuidados sem exercício [Rabelo et al. 2006].

Um aspecto que deve ser considerado na terapia através de exercícios físicos diz respeito a respostas desproporcionais na frequência cardíaca. Isto pode indicar uma situação de risco para o paciente. Desta forma, o reconhecimento da atividade física e sua correlação com a frequência cardíaca pode permitir uma recuperação mais segura [Negrão and Barreto 2010].

Nesse sentido, o emprego de sistemas computacionais para detecção autônoma de atividades e a correlação destas com parâmetros fisiológicos pode minimizar a necessidade de intervenção do próprio paciente ou do terapeuta na identificação de situações de risco.

Avanços em tecnologias de sensores, atuadores e protocolos de rede sem fio tem permitido o desenvolvimento de soluções de computação embarcada para diferentes finalidades. Dispositivos como utensílios domésticos, carros, equipamentos médicos e *smartphones* podem ser conectados e interagir a qualquer momento, em qualquer lugar com qualquer pessoa ou outro dispositivo através de uma rede unificada, denominada de Internet das Coisas (IoT) [Razzaque et al. 2015].

O crescimento do número de dispositivos conectados, propiciado pelos avanços em IoT, fez com que a quantidade de dados coletados e disponibilizados para as aplicações crescesse significativamente. Porém, para muitas aplicações os contextos de interesse podem ser obtidos através da fusão de dados de sensores, estabelecendo uma visão mais abstrata do contexto atual, permitindo a inferência da situação do usuário e do próprio sistema [Perera et al. 2013].

Nesse cenário, um dos principais desafios de pesquisa em IoT é a ciência de situação. Uma situação corresponde a uma visão compreensível e em alto nível de abstração do contexto de interesse da aplicação. Esta visão é resultante de dados de contexto coletados de sensores distribuídos pelo ambiente [Bellavista et al. 2012, Knappmeyer et al. 2013].

A principal contribuição deste trabalho é uma abordagem de processamento, integrada ao middleware EXEHDA (*Execution Environment for Highly Distributed Applications*) [Lopes et al. 2014a], chamada EXEHDA-CS (EXEHDA - Collaborative Situations), a qual provê ciência de situação para aplicações em IoT, permitindo a colaboração entre o middleware e as aplicações na composição e parametrização dos mecanismos de processamento de contexto utilizados na identificação de situações.

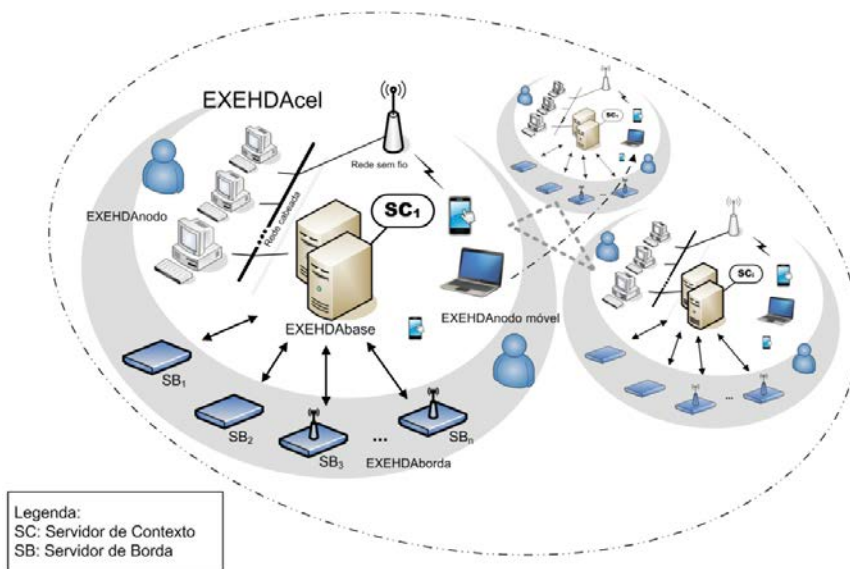
A arquitetura proposta foi avaliada através de um estudo na área de saúde com foco em reabilitação cardíaca. Este estudo de caso permitiu que funcionalidades relativas a parametrização de mecanismos de processamento e colaboração entre a aplicação e middleware fossem avaliadas.

O texto deste artigo está estruturado da seguinte forma: na seção 2 são abordadas características do middleware EXEHDA e da abordagem de processamento proposta. Na seção 3 é apresentado o estudo de caso que foi elaborado para avaliar a abordagem proposta. Na seção 4 são apresentados os trabalhos relacionados. Na seção 5 são apresentadas as considerações finais sobre o trabalho.

## **2. EXEHDA-CS: Concepção da Abordagem Proposta**

O middleware EXEHDA possui dois tipos de servidores: (i) Servidor de Borda, responsável por interagir com ambiente através de sensores e atuadores; e (ii) Servidor de Con-

texto, responsável por prover funcionalidades para ciência de situação. Estes servidores são alocados em células do ambiente gerenciado pelo EXEHDA, onde cada célula possui um Servidor de Contexto e pode possuir vários Servidores de Borda. Na Figura 1 pode ser vista a disposição destes servidores no ambiente.



**Figura 1. Ambiente gerenciado pelo EXEHDA.**

O foco de deste trabalho é a criação de mecanismos para configuração e acesso ao Servidor de Contexto, permitindo que a identificação de situações possa ser feita de maneira colaborativa com as aplicações. Na Subseção 2.1 é apresentada uma visão geral sobre o Servidor de Contexto do EXEHDA e na Subseção 2.2 é apresentada a abordagem de processamento proposta e as APIs criadas para possibilitar a configuração e o uso do middleware pelas aplicações.

## 2.1. Servidor de Contexto

O Servidor de Contexto é formado por cinco módulos: Aquisição, Atuação, Notificação, Comunicação e Processamento [Lopes et al. 2014b], os quais podem ser visualizados na Figura 2 e são descritos a seguir.

**Módulo de Aquisição:** é responsável por prover suporte à captura das informações contextuais, coletadas pelos Servidores de Borda considerando sensores lógicos, implementados através de interfaces de software, e sensores físicos providos por interfaces de hardware.

**Módulo de Atuação:** controla a ativação, desativação e configuração dos atuadores, como consequência de uma notificação de outros módulos do Servidor de Contexto. Esse módulo recebe o identificador do atuador envolvido e os parâmetros operacionais a serem utilizados e interopera com os Servidores de Borda para disparo dos atuadores pertinentes.

**Módulo de Processamento:** tem como principal função realizar as tarefas pertinentes ao tratamento das informações contextuais, bem como dos eventos para identificar situações de interesse das aplicações. Para execução destas tarefas, os componentes de

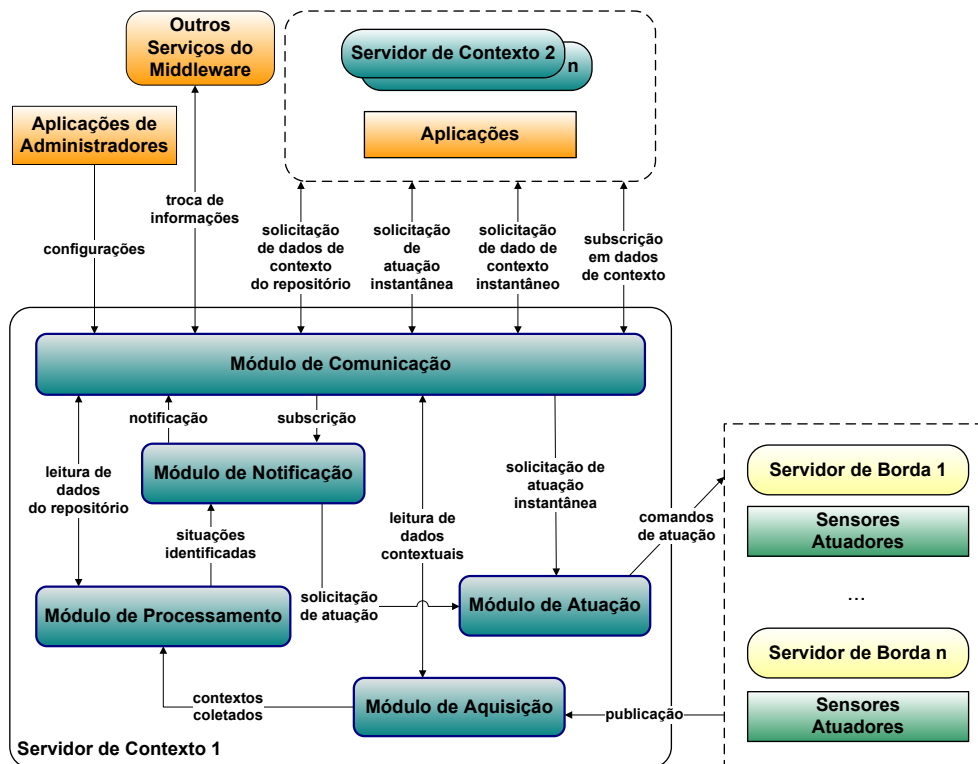


Figura 2. Arquitetura do servidor de contexto.

software que constituem esse módulo interagem com os modelos de representação do contexto.

**Módulo de Notificação:** é responsável por notificar o resultado do processamento das informações de contexto e da consequente identificação de situações realizada pelo Módulo de Processamento. Esse módulo opera recebendo subscrições dos serviços e/ou aplicações que desejem notificações a respeito dos estados contextuais e das situações identificadas, interoperando através do Módulo de Comunicação.

**Módulo de Comunicação:** é utilizado por Servidores de Contexto remotos e/ou aplicações para solicitação de situações constituídas por contextos de interesse correlacionados, dados contextuais e/ou o disparo de atuadores. Na arquitetura proposta neste trabalho será explorada uma API (*Application Programming Interface*) do tipo REST (*Representational State Transfer*) criada para suportar as etapas de configuração e acesso através do módulo de comunicação.

## 2.2. Abordagem Colaborativa para Identificação Situações

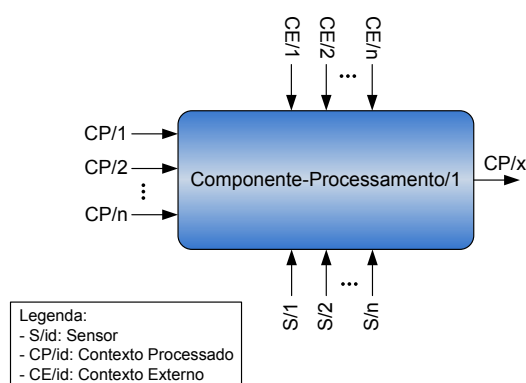
A abordagem de processamento colaborativa proposta neste trabalho contempla dois fluxos de dados contextuais: (i) o middleware executa a coleta de dados contextuais do meio de forma autônoma; (ii) as aplicações podem disponibilizar dados de contexto e/ou critérios que serão considerados quando do processamento combinado dos dados contextuais coletados.

A estrutura de processamento foi modelada para permitir acesso através de APIs do tipo REST [Fielding 2000]. Na Figura 3 pode ser vista a representação de um componente de processamento elementar da arquitetura <Componente-Processamento/id>.

Para compor o fluxo de processamento contextual são criadas instâncias dos componentes de processamento, alocadas em URIs (Uniform Resource Identifier) do tipo **<Instancia-Processamento/id>**.

A execução das instâncias de processamento é disparada por eventos de modificação nos dados de entrada. Cada instância deve possuir como entrada ao menos uma URI para sensor, contexto processado, ou contexto externo. A saída do processamento é armazenada em uma URI do tipo **<Contexto-Processado/id>**. Em cada instância de processamento estão envolvidos quatro tipos de URIs:

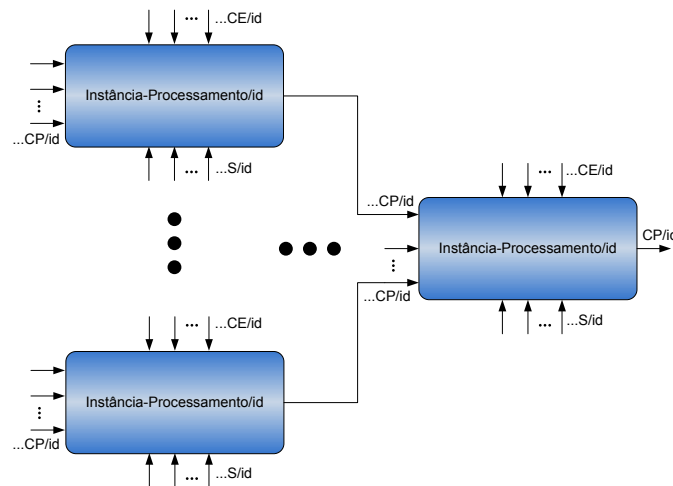
- **Sensores** <Sensor/id>: sensor disponibilizado por um Servidor de Borda.
- **Contextos Processados** <Contexto-Processado/id>: saída de uma instância de um componente de processamento, representa uma informação de contexto processado que poderá ser utilizada na identificação de situações ou composição de outros contextos processados.
- **Contextos Externos** <Contexto-Externo/id>: dados de contexto obtidos de aplicações externas, recebidos através do Módulo de Comunicação.
- **Componente de Processamento** <Componente-Processamento/id>: rotina de software armazenada no repositório do middleware.



**Figura 3. Componente de processamento elementar.**

Como pode ser visto na Figura 3, os componentes de processamento de contexto podem receber como dados de entrada variáveis de contexto processadas por outros módulos, permitindo a criação de arranjos de processamento conforme mostra a Figura 4.

A descrição da API REST criada pode ser vista na Tabela 1. Os recursos quando acessados sem um ID, possuem somente o comando GET, para o qual é retornada a lista de recursos disponíveis. Quando é especificado um ID, estão disponíveis os comandos GET, POST, PUT e DELETE, os quais possuem o comportamento padrão do protocolo HTTP. Os sensores possuem somente o comando GET porque o seu gerenciamento é feito através de uma API do servidor de borda.



**Figura 4. Modelo de processamento de contexto combinando múltiplas instâncias de componentes de processamento.**

**Tabela 1. Recursos da API REST da arquitetura proposta.**

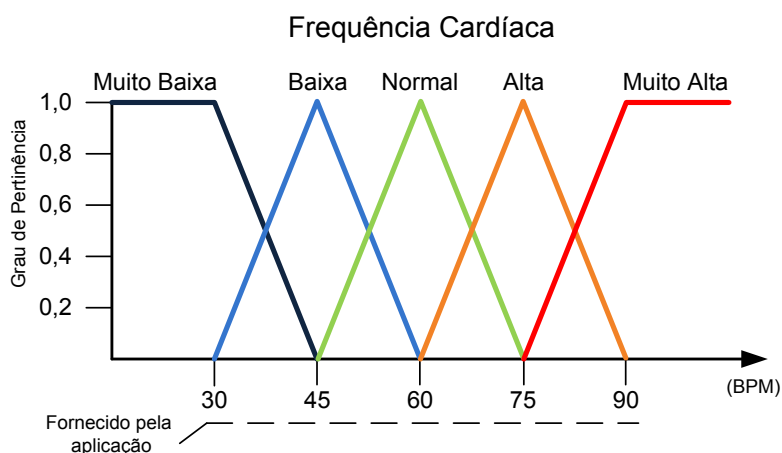
URI	Comandos	Descrição
Sensor	GET	Lê o conjunto de Sensores Disponíveis.
Sensor/id	GET	Lê os dados de um sensor.
Contexto-Processado	GET	Lê o conjunto de Contextos Processados cadastrados.
Contexto-Processado/id	POST, GET, PUT e DELETE	Operações sobre cadastros de Contextos Processados, incluindo leitura do valor atual.
Contexto-Externo	GET	Lê o conjunto de Contextos Externos cadastrados.
Contexto-Externo/id	POST, GET, PUT e DELETE	Operações sobre cadastros de Contextos Externos, incluindo leitura do valor atual.
Componente-Processamento	GET	Lê o conjunto de Componentes de Processamento cadastrados.
Componente-Processamento/id	POST, GET, PUT e DELETE	Operações sobre Componentes de Processamento.
Instancia-Processamento	GET	Lê o conjunto de instâncias de Componentes de Processamento criadas para a aplicação.
Instancia-Processamento/id	POST, GET, PUT e DELETE	Operações sobre uma dada instância de um componente de processamento.

### 3. EXEHDA-CS: Avaliação da Arquitetura

Para realizar a avaliação das funcionalidades da arquitetura foi prototipada uma aplicação para monitoramento de pacientes em reabilitação cardíaca após um acidente vascular. Estudos na área de cardiologia indicam que exercícios físicos podem reduzir o tempo de recuperação destes pacientes. Porém, respostas desproporcionais no batimento cardíaco podem indicar situações de risco para o paciente. Desta forma, o reconhecimento da atividade física e sua correlação com a frequência cardíaca pode permitir uma recuperação mais segura [Negrão and Barreto 2010]. O risco inerente à situação do paciente foi classificado no domínio linguístico: baixo, moderado e alto.

A aplicação proposta explora a capacidade de colaboração entre middleware e aplicações para a composição de situações. O middleware executa a identificação da atividade física do paciente e coleta a informação de frequência cardíaca do mesmo. A aplicação fornece ao middleware os parâmetros de normalidade do paciente, utilizados na etapa de conversão dos dados de contexto para o domínio fuzzy. Na Figura 5 pode ser visto um exemplo de funções de pertinência utilizadas para a conversão da frequência de batimento cardíaco para o domínio fuzzy.

A lógica fuzzy é adequada para a criação de regras para a classificação de situações, tendo em vista os seus mecanismos para tratamento de dados imprecisos e construção de algoritmos através de modelos interpretáveis, facilitando a criação de regras por especialistas da área de aplicação [Sobrevilla and Montseny 2003].



**Figura 5. Representação da frequência cardíaca através de lógica fuzzy, com padrão de normalidade fornecido pela aplicação.**

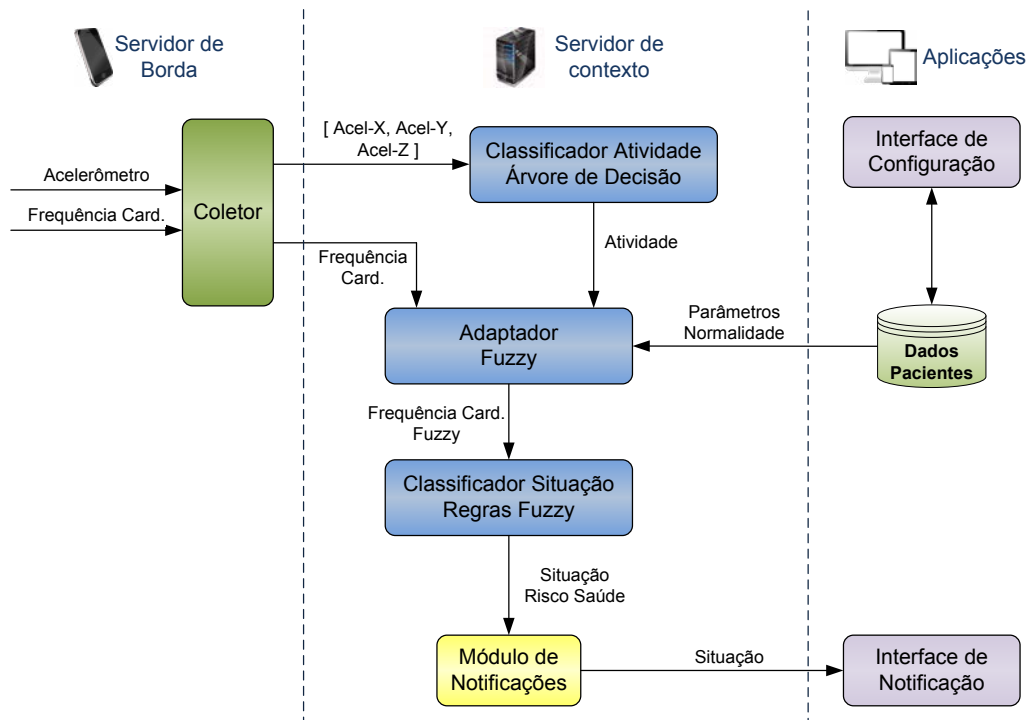
O processamento de contexto para identificação de situações, mostrado na Figura 6, baseou-se no uso da técnica de aprendizado Árvore de Decisão para identificação da atividade física realizada pelo paciente. As regras para inferência do risco de saúde foram criadas utilizando lógica fuzzy, considerando a atividade física e a frequência cardíaca do paciente. Os recursos alocados através da API REST são apresentados na Tabela 3.

O treinamento e o teste do componente para classificação de atividades foi feito utilizando a base de dados disponibilizada no trabalho de Kwapisz et al. [Kwapisz et al. 2011], no qual foram capturados sinais de acelerômetros de *smartphones* de 29 voluntários durante a execução das seguintes atividades físicas: caminhando,



**Tabela 2. Matriz de confusão para classificação de atividades utilizando Árvore de Decisão.**

	Repouso	Movimento lento	Movimento rápido
Repouso	<b>97,8%</b>	0,4%	0,1%
Movimento lento	2,2%	<b>97,8%</b>	3,2%
Movimento rápido	0,0%	1,8%	<b>96,7%</b>
<b>Total correto</b>	<b>97,5%</b>		

**Figura 6. Visão geral do estudo de caso proposto.**

correndo, subindo escada, descendo escada, sentado e de pé. Com a finalidade de adequar as atividades para a aplicação proposta as mesmas foram agrupadas como:

- **Repouso:** sentado e de pé;
- **Movimento lento:** caminhando, descendo escada e subindo escada;
- **Movimento rápido:** correndo.

Após o treinamento foi gerada uma Árvore de Decisão com 62 ramos e 123 elementos, sendo utilizadas 43 características extraídas do sinal do acelerômetro para composição das decisões. Dentre as características utilizadas, pode citar-se: frequência fundamental, módulo da aceleração resultante, aceleração média e desvio padrão em cada um dos três eixos do acelerômetro. O sinal do acelerômetro foi obtido com uma taxa de 20 amostras por segundo e o processamento das características foi executado em janelas temporais de 10 segundos. O resultado da classificação de atividades para a Árvore de Decisão é apresentado na Tabela 2. O percentual total de classificação correta obtido foi de 97,5%.

A frequência cardíaca apropriada para cada atividade física é obtida através de



**Tabela 3. Recursos alocados para o estudo de caso.**

URI	Descrição
Contexto-Processado/Atividade	Atividade inferida no classificador baseado em Árvore de Decisão.
Contexto-Processado/Freq-Cardiaca-Fuzzy	Frequência cardíaca em domínio fuzzy.
Contexto-Processado/Situacao	Saída que indica o risco de saúde atual do paciente.
Contexto-Externo/Parametros-Normalidade	Informações de normalidade do paciente fornecidas pela aplicação.
Instancia-Processamento/Adaptador-Fuzzy	Instância de componente de processamento para conversão da frequência cardíaca para domínio fuzzy.
Instancia-Processamento/Classificador-Atividade	Instância de componente de processamento para classificação de atividade utilizando Árvore de Decisão.
Instancia-Processamento/Classificador-Situacao	Classificador de situação utilizando regras em lógica fuzzy.

uma interface de **Contexto Externo** para a aplicação em que é feito o gerenciamento dos pacientes. O domínio linguístico para frequência cardíaca é dado por: muito baixa, baixa, normal, alta e muito alta. No bloco **Adaptador Fuzzy** os valores numéricos obtidos do sensor de batimento cardíaco são transformados para o domínio fuzzy aplicando funções de pertinência triangulares, parametrizadas com os padrões de normalidade do paciente.

Na instância de processamento em que é feita a classificação da situação de risco de saúde do paciente é aplicado um conjunto de regras utilizando lógica fuzzy. Estas regras podem ser vistas na Figura 7.

```

GET freq_cardiaca
IF freq_cardiaca IS muito_alta OR freq_cardiaca IS muito_baixa THEN
    risco IS alto
IF freq_cardiaca IS alta OR freq_cardiaca IS baixa THEN
    risco IS moderado
IF freq_cardiaca IS normal THEN
    risco IS baixo

```

**Figura 7. Algoritmo para inferência da situação de risco de saúde do paciente.**

#### 4. Trabalhos Relacionados

O estudo de trabalhos relacionados foi feito considerando as especificações do EXEHDA-CS: arquitetura utilizada em diferentes domínios de aplicação; arquitetura distribuída; abordagem colaborativa de processamento de contexto e suporte para identificação de situações.

Yuan e Herbert [Yuan and Herbert 2014] propõem uma arquitetura para suporte ao monitoramento e reconhecimento de atividades e hábitos para aplicações na área de vivência assistida. Neste trabalho é previsto o uso de um algoritmo de aprendizado baseado em casos e as regras da aplicação são criadas utilizando lógica fuzzy. No EXEHDA-CS a abordagem de processamento foi estruturada de forma a suportar a inclusão de novos algoritmos através de Componentes de Processamento. Desta forma, diversas técnicas de aprendizado e especificação podem ser combinadas.

Haghighi et al. [Haghighi et al. 2014] apresentam uma arquitetura para uma aplicação específica na área de monitoramento de pacientes hipertensos. Nesta aplicação não é provido suporte ao processamento distribuído. Já o EXEHDA-CS foi especificado para ser independente do domínio de aplicação e sua arquitetura foi construída para poder operar de forma distribuída.

Cimino et al. [Cimino et al. 2012] propõem uma arquitetura de software para ciência de situação. Os autores propõem um modelo de processamento que considera o uso de regras e de modelos semânticos para representação e processamento do contexto. No EXEHDA-CS considerou-se que o middleware deve prover suporte para diversos tipos de representação de contexto e modelos de processamento. Desta forma, através da API proposta para o Servidor de Contexto, a técnica de processamento e o padrão de representação do contexto são definidos através dos Componentes de Processamento.

Os trabalhos avaliados enfatizam a transparência dos mecanismos para processamento e modelagem de contexto. Considera-se, no entanto, que esta abordagem “caixa-preta” dos trabalhos relacionados não é apropriada para o desenvolvimento de aplicações com o grau de flexibilidade e generalidade previstas para o EXEHDA. Desta forma, a proposta para minimizar o conflito entre transparência e a capacidade de identificação de situações é a colaboração entre a aplicação e middleware no processamento contextual. Com isso, busca-se estabelecer uma abordagem flexível para o processamento e modelagem do contexto.

## 5. Considerações Finais

A principal contribuição do EXEHDA-CS é uma arquitetura para prover ciência de situação, através de uma abordagem que permite a colaboração entre o middleware e as aplicações. Na API proposta foram considerados os aspectos de flexibilidade na criação de mecanismos para o processamento das informações de contexto visando à identificação de situações.

Através do estudo de caso foi possível demonstrar que a abordagem proposta permitiu a colaboração entre o middleware e a aplicação para identificação de situações, otimizando a construção dos modelos de raciocínio fuzzy (vide Figura 7).

Como trabalho futuro pretende-se expandir a API proposta neste trabalho para dar suporte aos mecanismos de adaptação previstos no EXEHDA.

## Referências Bibliográficas

Bellavista, P., Corradi, A., Fanelli, M., and Foschini, L. (2012). A survey of context data distribution for mobile ubiquitous systems. *ACM Comput. Surv.*, 44(4):24:1–24:45.

- Cimino, M. G. C. A., Lazzerini, B., Marcelloni, F., and Ciaramella, A. (2012). An adaptive rule-based approach for managing situation-awareness. *Expert Syst. Appl.*, 39(12):10796–10811.
- Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. Doctoral dissertation, University of California.
- Haghighi, D. P., Averì, P., Indrawan-Santiago, M., and Minh, H. T. (2014). Situation-aware mobile health monitoring. *ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering)*, pages 248–256.
- Knappmeyer, M., Kiani, S., Reetz, E., Baker, N., and Tonjes, R. (2013). Survey of context provisioning middleware. *Communications Surveys Tutorials, IEEE*, 15(3):1492–1519.
- Kwapisz, J. R., Weiss, G. M., and Moore, S. a. (2011). Activity recognition using cell phone accelerometers. *ACM SIGKDD Explorations Newsletter*, 12:74.
- Lopes, J., Souza, R., Geyer, C., Costa, C., Barbosa, J., Pernas, A., and Yamin, A. (2014a). A middleware architecture for dynamic adaptation in ubiquitous computing. *Journal of Universal Computer Science*, 20(9):1327–1351.
- Lopes, J., Souza, R., Pernas, A., Yamin, A., and Geyer, C. (2014b). A distributed architecture for supporting context-aware applications in ubicomp. In *Advanced Information Networking and Applications (AINA), 2014 IEEE 28th International Conference on*, pages 584–590.
- Negrão, C. E. and Barreto, A. C. P. (2010). *Cardiologia do Exercício: do Atleta ao Cardiopata*. Manole, Barueri, SP - Brazil, 3 edition.
- Perera, C., Zaslavsky, A., Christen, P., and Georgakopoulos, D. (2013). Context Aware Computing for The Internet of Things : A Survey. *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*, X(X):1–41.
- Rabelo, D., Gil, C., and Araújo, S. D. (2006). Reabilitação cardíaca com ênfase no exercício: uma revisão sistemática. *Revista Brasileira de Medicina do Esporte*, 12(5):279–285.
- Razzaque, M. A., Milojevic-Jevric, M., Palade, A., and Clarke, S. (2015). Middleware for Internet of Things: a Survey. *Internet of Things Journal, IEEE*, PP(99):1.
- Sobrevilla, P. and Montseny, E. (2003). Fuzzy Sets in Computer Vision : an Overview. *Mathw. Soft Computing*, 10:71–83.
- Yuan, B. and Herbert, J. (2014). Context-aware hybrid reasoning framework for pervasive healthcare. *Personal and Ubiquitous Computing*, 18(4):865–881.