

Uma abordagem para a personalização automática de interfaces de usuário para dispositivos móveis em Ambientes Pervasivos

Ricardo Giuliani Martini¹, Giovanni Rubert Librelotto¹

¹Programa de Pós-Graduação em Informática - Universidade Federal de Santa Maria (UFSM)
Av. Roraima, 1000 - 97105-900 - Santa Maria - RS - Brazil

giulianimartini@gmail.com, librelotto@inf.ufsm.br

Abstract. *This paper presents the PIDIM architecture in order to assist in the customization of user interfaces for mobile devices in pervasive environments. This architecture uses concepts of Pervasive Computing and represents data on the user's profile, so that adaptation of the interfaces is entirely focused on the end user. The knowledge representation about the user profile is done through ontologies due to the possibility of reuse of stored information. In order to validate and demonstrate the flow of operation of the proposed approach is presented a case study, which has as scenario the adaptation of user interfaces when it is in motion.*

Resumo. *Este trabalho apresenta a arquitetura PIDIM, a fim de ajudar na personalização de interfaces de usuário para dispositivos móveis em ambientes pervasivos. A arquitetura utiliza conceitos de Computação Pervasiva, além de representar dados relativos ao perfil de usuários, para que a adaptação das interfaces seja totalmente focada no usuário final. A representação do conhecimento sobre o perfil do usuário é feita através de ontologias devido a possibilidade de reuso das informações armazenadas. A fim de validar e demonstrar o fluxo de funcionamento da abordagem proposta, é apresentado um caso de estudo, encontrado na literatura, o qual possui como cenário a adaptação de interfaces de usuários quando o mesmo se encontra em movimento.*

1. Introdução

Com o passar dos anos, após a invenção das interfaces gráficas de usuário (GUI) para *desktop*, as quais possuíam formas de interação baseadas em *mouse*, teclado e telas com tamanhos padronizados, o *design* não precisava se preocupar muito com adaptação de interface. Com o aparecimento dos dispositivos móveis, a área em questão passou a ser de fundamental importância, pois estes dispositivos possuem características particulares fundamentais para a composição de uma interface satisfatória ao usuário.

Com o avanço na computação pervasiva, surgem diversas necessidades, tais como independência de dispositivo, mapeamento de contexto, adaptação de conteúdo, entre outros. Além destas necessidades, é preciso que a computação pervasiva seja invisível aos olhos do usuário final. Para esta invisibilidade ser alcançada, devem existir interfaces de usuários (IU) intuitivas, além de poder ser aplicada em diferentes contextos.

Com o surgimento da computação pervasiva juntamente com a computação móvel, onde o uso de dispositivos computacionais não se prende mais somente nas estações de

trabalho, é necessário que a interface se adapte ao perfil do usuário, levando em conta as capacidades do dispositivo, bem como o contexto no qual estão inseridos.

Neste trabalho será apresentada uma abordagem para personalização de IU para dispositivos móveis em ambientes pervasivos. Para alcançar o objetivo da personalização da interface, são utilizados perfis de usuários, onde existem dados relevantes à adaptação da interface, bem como perfis de dispositivos, os quais descrevem as capacidades dos mesmos, além da necessidade de levar em conta o contexto do ambiente.

Para descrever o perfil do usuário utilizou-se ontologias, onde estão descritos dados como idade, sexo, dados culturais, preferências, deficiências, entre outros. Já para a descrição dos dispositivos utilizou-se um vocabulário chamado UAProf [Alliance 2001], o qual detalha as capacidades do dispositivo. O contexto do ambiente é descrito através do uso de sensores, como sensores de temperatura, localização, ruídos, etc.

O principal desafio deste trabalho é fazer com que o usuário final, tenha ele o perfil que tiver, consiga desenvolver suas tarefas em seu dispositivo móvel da melhor forma possível, com melhor desempenho, e, principalmente, com maior satisfação. Para isso, a abordagem proposta neste trabalho utiliza a computação sensível ao contexto de forma a capturar dados relevantes ao ambiente pervasivo em que o usuário se encontra através de sensores.

A proposta em questão de criar uma arquitetura para personalização automática de IU em um ambiente pervasivo veio através da necessidade de sistemas interativos com o usuário serem projetados para satisfazer as necessidades e desejos do usuário final.

De forma a apresentar a abordagem proposta, o artigo inicia com uma seção sobre Computação Pervasiva e Ontologias, a fim de relacionar as duas áreas no contexto do artigo. Após, uma breve descrição de perfis para adaptação de conteúdo é detalhado, com o propósito de apresentar dados relevantes para a adaptação das interfaces de forma individual, visando o usuário. Em seguida, a arquitetura proposta é apresentada, detalhando todo o fluxo de comunicação entre os módulos da mesma. Após a apresentação da arquitetura, um caso de estudo retirado da literatura e adaptado para a arquitetura proposta é detalhado, a fim de provar a coerência da abordagem proposta. Por fim, a conclusão e trabalhos futuros são apresentados.

2. Ontologias aplicadas à Computação Pervasiva

Aplicações na área da computação pervasiva devem ser sensíveis ao contexto e possibilitar a inferência de fatos com base no comportamento do mesmo, tomando assim uma decisão autônoma. Quais decisões podem ser tomadas autonomamente devem possuir maior atenção, pois elas não são desejáveis em determinados contextos.

De acordo com [Dey 2001], uma das noções fundamentais da computação pervasiva é a sensibilidade ao contexto, onde, por contexto entende-se qualquer informação relevante que possa ser utilizada para caracterizar a situação de alguma entidade, (por exemplo identidade e localização). [Yamin 2004] define contexto como qualquer informação gerada pela infraestrutura computacional do ambiente que pode ser relevante para uma aplicação na qual uma alteração pode disparar um processo para a adaptação desta aplicação.

Para alcançar sistemas sensíveis ao contexto na Computação Pervasiva é necessá-

rio descrever o ambiente pervasivo em questão. Para isso, são utilizadas ontologias, as quais são vistas como uma especificação formal e explícita de uma conceituação compartilhada. Conceituação se refere ao modelo abstrato de algum fenômeno do mundo o qual identifica conceitos relevantes do próprio fenômeno; formal se refere ao fato da ontologia ser entendida pela máquina; e compartilhada dá a noção de que uma ontologia obtém o conhecimento apresentado não somente por um único indivíduo, mas por um grupo [Studer et al. 1998].

A utilização de ontologias na Computação Pervasiva é vasta, podendo ser utilizada em áreas como gerenciamento e adaptação de conteúdo, comércio eletrônico, *Web* semântica, descoberta de serviços, transporte, educação, saúde, lazer, entre outras. Todas as aplicações nestas áreas em que ontologias podem ser utilizadas devem ser sensíveis ao contexto, visando a adaptação da aplicação assim que ocorrer qualquer mudança no ambiente pervasivo [Ye et al. 2012].

Para possibilitar que conteúdos desenvolvidos para a *Web* ou qualquer *software* hoje em dia na computação pervasiva sejam visualizados em qualquer dispositivo, é necessário que haja independência do mesmo. Criar uma interface para cada dispositivo de forma diferente e manual é inviável de forma administrativa e econômica, tendo em vista a demanda de dispositivos diferentes que existem. Essas diferenças vão desde o tamanho da tela até processamento, memória, funcionalidades, etc. Para resolver este problema, são descritos perfis, tanto de usuários como dispositivos. A seção 3 detalha estes dois tipos de perfis.

3. Perfis para Adaptação de Conteúdo

A necessidade de perfis para adaptação de conteúdo vem da inexistência de formas eficientes para obter as informações sobre dispositivos, usuários, e contexto em que ambos se encontram, além da complexidade de adaptar a interface do usuário de forma automática, devido à heterogeneidade dos dispositivos (diferentes processadores, memórias, tamanho de tela, etc).

Desta forma, este trabalho apresenta uma infraestrutura baseada no *framework Composite Capability/Preference Profiles* (CC/PP) [Kiss 2007] e na especificação *User Agent Profiling Specification* (UAProf) [Alliance 2001] capaz de oferecer as informações necessárias para que possa ser realizada a adaptação do conteúdo necessário. Tais especificações têm a finalidade de possibilitar o acesso a informações estáticas e dinâmicas dos dispositivos e dos usuários. O CC/PP e o UAProf são recomendações W3C, os quais são descritos na subseção 3.1. O perfil do usuário também é importante e está descrito logo a seguir na seção 3.2

3.1. CC/PP e UAProf

O CC/PP tem o objetivo de expressar características de dispositivos e preferências de usuários. Baseado no *Resource Description Framework*, o CC/PP é descrito em documentos RDF ou *eXtensible Markup Language* (XML), os quais, na criação de novos vocabulários, permitem uma maior flexibilidade.

Para descrever as características dos dispositivos e dos usuários, o CC/PP especifica documentos em XML, o qual pode ser dividido em componentes (*Hardware* e *Software*). Para descrever as características básicas dos dispositivos (dimensões de tela, sis-

tema operacional, etc.), o CC/PP fornece um vocabulário padrão. Entretanto, o CC/PP não pode evitar que cada fabricante crie um vocabulário próprio, dificultando assim a padronização de sistemas para a obtenção das características dos dispositivos. Desta forma, a *Open Mobile Alliance* [Alliance 2001] criou a especificação UAProf. Para descrever as características dos dispositivos, a especificação UAProf possui um conjunto de vocabulários. Um documento UAProf é criado e armazenado em um repositório *Web* do fabricante para cada dispositivo projetado.

3.2. Perfil do Usuário

Existem diversas formas de criar e personalizar interfaces de usuários. Uma das mais populares formas é o *design* centrado no usuário (*User-Centered Design* - UCD), o qual mantém uma participação ativa com o usuário, afim de dar atenção aos requisitos necessários em cada estágio do processo de *design* da interface, e no fim, testar com usuários reais.

Para que uma interface se adapte às necessidades do usuário, é necessário levantar determinados aspectos para definir o perfil do usuário final. Baseado no estudo de [Shneiderman 2000], um dos pioneiros na área de IHC, esses aspectos, separados em grupos são deficiências (motora, visual, auditiva, cognitiva), preferências (idioma, somente texto, somente imagens, cor, etc.), experiência com computador (iniciante, mediano, experiente), dados demográficos (idade, sexo, nacionalidade) e dados culturais.

4. Arquitetura PIDIM (Personalização de Interfaces para Dispositivos Móveis)

A partir das descrições das seções anteriores, esta seção trata de uma arquitetura que utiliza destes conceitos para personalizar uma IU para dispositivos móveis. O objetivo principal da arquitetura é adaptar a IU para dispositivos móveis baseando-se no perfil do usuário, através do uso de ontologias, capacidades do dispositivo, utilizando um vocabulário UAProf, e o contexto do ambiente, o qual é captado por sensores. Assim, esta seção tem por meta descrever uma arquitetura para personalização de IU. A figura 1 apresenta a arquitetura para alcançar este objetivo.

Na arquitetura proposta existem dois domínios, os quais comunicam-se entre eles. O primeiro domínio se refere ao aplicativo, lado cliente da arquitetura, o qual contém a parte lógica da aplicação móvel, e também a linguagem descritiva de IU. O segundo domínio, servidor, possui a ontologia que descreve o perfil de usuário, expressa em OWL.

Além da ontologia, o lado servidor da arquitetura também retém três módulos que buscam dados relativos ao perfil do usuário (SQWRLPerfilUsuario), sensores do ambiente (Sensores), e repositório de perfis UAProf dos dispositivos (UAPROFDispositivo). Ainda no lado servidor, existe um outro módulo que possui o papel de executar regras com todos os dados oriundos dos módulos SQWRLPerfilUsuario, Sensor e UAPROFDispositivo, a fim de personalizar a IU.

Também existe um módulo para atualizar a ontologia (AtualizaOntologia) no lado servidor da arquitetura, que pode ser visto na figura 2, o qual recebe o usuário captado pelos sensores e o resultado de testes de deficiências e preferências do usuário para atualizar na ontologia perfil do usuário. Uma cache para armazenar os documentos sobre os perfis de dispositivos também está no servidor.

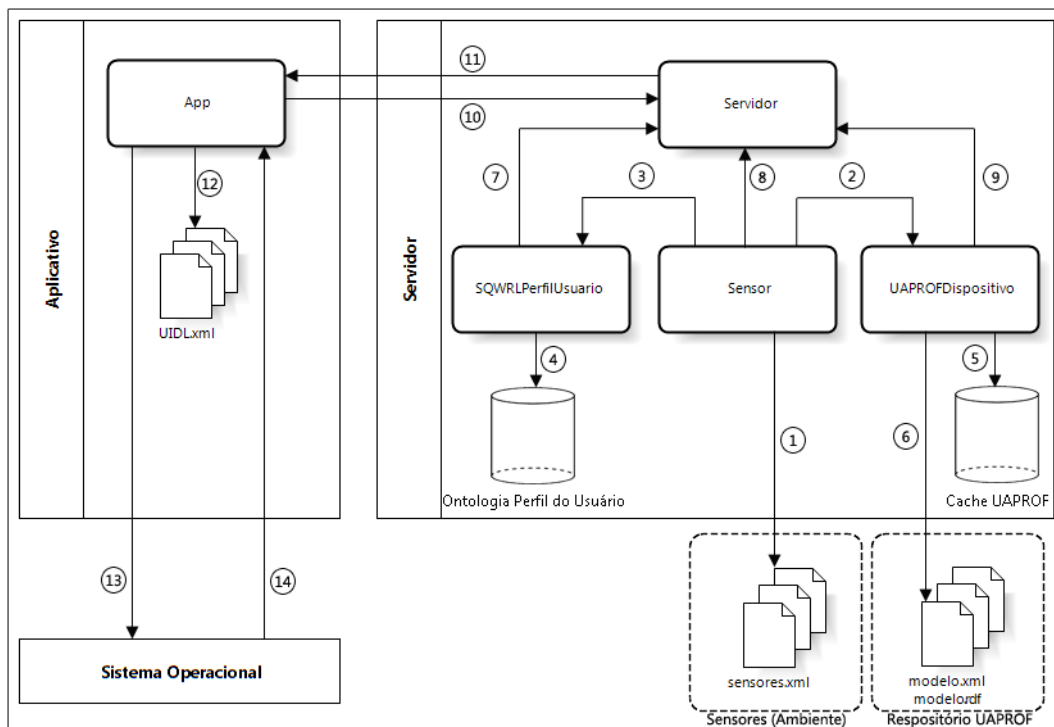


Figura 1. Arquitetura PIDIM

A partir da numeração da figura 1 será apresentado o fluxo de comunicação entre os domínios da arquitetura PIDIM. No ambiente pervasivo existem sensores, os quais captam mudanças de contexto que possam acarretar em fatores importantes para a personalização da IU. Após cada mudança captada pelos sensores, um serviço publica um documento XML descrevendo a mesma.

Logo no início do fluxo, em (a), o módulo AtualizaOntologia busca o documento XML referente ao usuário, no serviço que disponibilizou o documento XML com dados captados pelos sensores. Logo em seguida, em (b), o Aplicativo (App) repassa o resultado de alguns testes referentes às preferências e deficiências do usuário, e por fim, em (c), a ontologia perfil do usuário é atualizada através da OWL API.

O fluxo da arquitetura segue da seguinte maneira: (1) o módulo Sensor busca os documentos XML publicados pelo serviço dos sensores do ambiente pervasivo. Agora, o módulo Sensor retém os dados do contexto, como dispositivos (modelos), usuários presentes no ambiente, luminosidade, etc;

Após, em (2), o módulo Sensor passa os dados dos dispositivos (modelos) para o módulo UAPROFDispositivo; Em (3), o módulo Sensor passa o usuário que está no ambiente (a ser encontrado na ontologia) para o módulo SQWRLPerfilUsuario; Após processados os dados dos sensores, o passo (4) faz com que o módulo SQWRLPerfilUsuario busque dados do usuário na ontologia perfil do usuário com a utilização da OWL API; Em seguida, no passo (5), o módulo UAPROFDispositivo tenta buscar o dispositivo (modelo) na cache UAPROF. Caso o modelo não exista na cache, executa o passo (6), onde o módulo UAPROFDispositivo utiliza a classe HttpURLConnection do pacote java.net para buscar o documento XML ou RDF do modelo do dispositivo no repositório

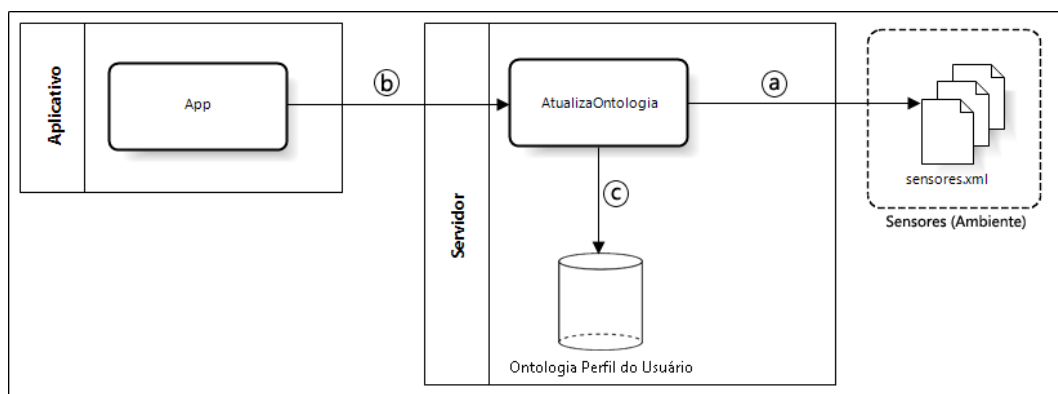


Figura 2. Atualização da Ontologia Perfil do Usuário

UAPROF.

O módulo SQWRLPerfilUsuario passa os dados referentes ao usuário para o módulo Servidor, em (7). O mesmo faz o módulo UAPROFDispositivo, em (8), enviando os dados do dispositivo, e o módulo Sensor em (9), repassando o restante dos dados (luminosidade, ruído, temperatura, etc). Neste ponto, o módulo Servidor executa as regras cruzando todos os dados captados pelos sensores, perfil de usuário e perfil de dispositivo.

O aplicativo móvel envia os arquivos XML, ou id e atributo referentes a IU para a classe Servidor modificar em (10). Esse envio é realizado via HTTP com a classe Java HttpURLConnection. Após receber os dados do lado cliente (aplicativo), para personalização da interface, em (11), o módulo Servidor modifica os arquivos referentes aos IUs recebidos (ou valores de atributos xml) e devolve para o aplicativo. Por fim, o aplicativo, em (12), atualiza os arquivos relativos ao IU (UIDL.xml). E em (13) requisita para o sistema operacional renderizar a GUI. Assim, em (14), o sistema operacional renderiza a IU.

5. Estudo de caso: interfaces para usuários em movimento (*Walking User Interface*)

Para validar a abordagem proposta, utilizou-se um estudo de caso onde o usuário utiliza um *player* de música em duas situações: quando está parado e quando está em movimento. O processo de validação é baseado na figura da arquitetura 1, onde existe uma numeração que orienta o controle do fluxo dos dados dentro da arquitetura.

O cenário deste caso de estudo é baseado no trabalho de [Kane et al. 2008]. Um usuário utilizando um dispositivo móvel encontra-se em um ambiente pervasivo povoado de sensores, como de luminosidade e ruído. No momento que o usuário entra em movimento, muitos fatores influenciam na perda de desempenho, perda de atenção e habilidade motora do mesmo. Para diminuir essa perda de desempenho e deficiências motoras situacionais, a arquitetura proposta neste trabalho visa adaptar a interface do aplicativo utilizado pelo usuário.

Testes com diferentes perfis de usuários foram realizados. Primeiramente, um aplicativo para tocar músicas (*music player*) foi desenvolvido para Android, com o intuito de realizar os testes com usuários reais em um dispositivo real. O aplicativo desenvolvido

possui uma interface padrão, a qual deve ser adaptada conforme o perfil do usuário que está utilizando o aplicativo no momento, o contexto do ambiente, e as capacidades do dispositivo.

Além do aplicativo desenvolvido, um percurso real foi estabelecido, com o intuito de receber um *feedback* dos usuários do teste logo após o término do percurso. O percurso possuía pessoas, bancos e árvores como obstáculos.

Cinco usuários foram testados, sendo três do sexo masculino e dois do sexo feminino. Dentre eles haviam usuários com daltonismo, membro amputado, idosos e jovens. A tabela 1 mostra o perfil de cada usuário que participou dos testes.

Tabela 1. Tabela de usuários que realizaram os testes

Tabela de usuários que realizaram os testes						
Usuário	Idade	Sexo	Cor do Texto	Cor Background	Fonte	Experiência
Usuario_1	21	F	Verde	Branco	Arial.ttf	Ocasional
Usuario_2	58	F	Preto	Branco	Calibri.ttf	Ocasional
Usuario_3	14	M	Preto	Vermelho	Impact.ttf	Ocasional
Usuario_4	63	M	Azul Fraco	Preto	Arial.ttf	Novato
Usuario_5	25	M	Branco	Vermelho	Verdana.ttf	Especialista

Além dos dados descritos na tabela 1, todos os usuários possuem nacionalidade brasileira e entendem o idioma português. O Usuario_2 possui daltonismo e o Usuario_5 possui a mão esquerda amputada. Outro dado importante é a orientação de cada usuário, visto que em um teste no qual o usuário está em movimento, certas habilidades motoras são perdidas. O Usuario_1, Usuario_2 e Usuario_4 são destros, enquanto os usuários Usuario_3, e Usuario_5 são canhotos.

Os dados destes perfis de usuários foram obtidos, inicialmente, a partir de um questionário, feito através do dispositivo móvel. A partir de então, periodicamente realizam-se alguns testes com os usuários através de interfaces especializadas, de forma a atualizar a ontologia com perfil destes usuários.

Para complementar o desenvolvimento da interface adaptada, simulou-se os dados oriundos dos sensores, como localização do usuário no ambiente, situação atual (parado ou em movimento), obstáculos no caminho e a presença de outras pessoas no ambiente.

O dispositivo utilizado para os testes foi um Samsung Galaxy 5 (modelo I5500), com o sistema operacional Android. A figura 3 mostra a interface padrão do aplicativo desenvolvido para este caso.

A adaptação da IU neste caso de estudo foi baseada na numeração da figura da arquitetura 1. Os sensores do ambiente pervasivo captam mudanças de contexto que possam ser importantes para a personalização da IU. A cada mudança captada pela simulação dos sensores, cria-se um documento XML descrevendo a mesma. No início do fluxo, em (a), o módulo AtualizaOntologia busca o documento XML referente ao usuário, nos sensores. Logo em seguida, em (b), o Aplicativo (App) repassa o resultado de alguns testes referentes às preferências e deficiências do usuário, e por fim, em (c), a ontologia perfil do usuário é atualizada. Após a atualização da ontologia, em (1), são buscados os documentos XML criados pelos sensores, pelo módulo Sensor. Neste ponto o módulo Sensor



Figura 3. Layout padrão do aplicativo *music player*

possui dados do contexto, como a localização de usuários ou dispositivos, luminosidade, modelo do dispositivo, etc.

```
<?xml version="1.0" encoding="UTF-8" ?>
<sensor xmlns:xsi="sensor.xsd"
  <id>SR0011</id>
  <!-- código identificador do sensor -->
  <info>Samsung I5500</info>
  <!-- atributo valor -->
  <tipo>Dispositivo</tipo>
  <!-- Tipo de informação que o sensor detecta -->
</sensor>
```

Figura 4. Dados capturados pelo sensor de dispositivos.

Após a captura dos dados relevantes ao sistema, em (2), o módulo Sensor passa o modelo do dispositivo Samsung I5500 para o módulo UAPROFDispositivo. Os dados do Usuario_5 também são passados para o módulo SQWRLPerfilUsuario, no passo (3). No passo (4), SQWRLPerfilUsuario busca na ontologia perfil do usuário através da OWL API os dados do Usuario_5.

A fim de capturar as capacidades do dispositivo móvel, em (5), o módulo UAPROFDispositivo tenta buscar o modelo do dispositivo na cache UAProf. Caso o modelo não exista na cache, deve buscar no repositório UAProf. Isso acontece no passo (6), onde o módulo UAPROFDispositivo utiliza a classe java HttpURLConnection do pacote java.net para buscar o documento XML ou RDF referente ao modelo Samsung I5500.

Seguindo a execução, em (7), o módulo SQWRLPerfilUsuario tem o papel de passar os dados do Usuario_5 para o módulo Servidor. Os dados referentes ao Usuario_5 são dados como corPreferidaTexto, corPreferidaBackground, fontePreferida, possuiExperiencia, idade, nacionalidade, sexo, possuiDeficiencia e MembroCorpoHumano.

No passo (8), o módulo UAPROFDispositivo deve passar os dados do dispositivo (Samsung I5500) para o módulo Servidor. Os dados do dispositivo Samsung I5500 são:

- <prf:ColorCapable>Yes</prf:ColorCapable>
- <prf:OSName>Android</prf:OSName>
- <prf:OSVersion>2.3</prf:OSVersion>

Logo após, em (9), o módulo Sensor passa o restante dos dados captados pelos sensores para o módulo Servidor. Neste momento o módulo Servidor deve executar regras com a junção dos dados dos sensores, dispositivo, e perfil do usuário. Depois do servidor executar as regras necessárias, no passo (10), o aplicativo móvel envia para o

módulo Servidor os atributos e identificadores (id), ou arquivos XML referentes a IU para a personalização da mesma. O envio é realizado através da classe `HttpURLConnection`.

A partir disso, em (11), os arquivos referentes a IU são modificados e devolvidos para o aplicativo móvel por http. Depois da modificação, em (12), o aplicativo móvel recebe os dados do módulo Servidor e atualiza os arquivos relativos a IU (`list_bg.xml`, `text_bg.xml`, `main.xml`, `list.xml`). Após atualizar os arquivos da IU, o aplicativo móvel requisita a renderização da interface através do método `invalidate()` da classe `android.view.View` do android. Isso ocorre no passo (13). Por fim, em (14), o sistema operacional renderiza a IU.

Após o último passo da abordagem proposta concluído, o usuário verá sua interface adaptada. A figura 5 mostra como a interface final ficou adaptada para o `Usuario_5`.

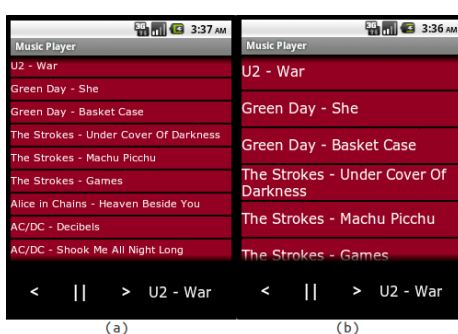


Figura 5. Interface final adaptada para as necessidades e preferências do `Usuario_5`, onde (a) corresponde a interface quando o usuário está parado, e (b) quando o usuário está em movimento.

Segundo [dos Santos 2008], existem diversos autores que procuraram ao longo do tempo, encontrar o conjunto ideal de regras e métricas para medir o grau de usabilidade de sistemas. Muitas regras são semelhantes e por isso, a unificação dos critérios avaliados pelos autores pôde ser concluída. Com a união das métricas, chegou-se a um conjunto de quatro critérios importantes: facilidade no aprendizado, facilidade de relembrar, eficiência, e eficácia e satisfação. Esses quatro critérios foram utilizados para avaliar o aplicativo *music player* para que os usuários passassem o *feedback* de avaliação da IU adaptada conforme seu perfil. O gráfico da figura 6 mostra o grau de satisfação dos usuários conforme os critérios avaliados.

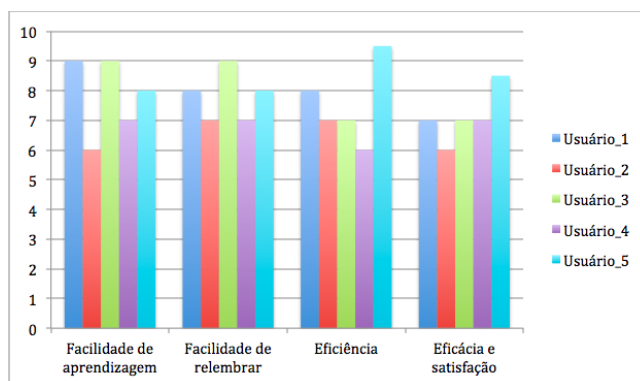


Figura 6. Grau de satisfação dos usuários baseado em critérios de usabilidade

6. Conclusão

Com os diferentes tipos de plataformas e dispositivos eletrônicos, é importante que se trabalhe para obter-se uma hegemonia nas interfaces, de modo que a interface se adapte ao usuário. Desta forma, diminui-se a curva de aprendizado entre diferentes dispositivos e por consequência reduz-se as possibilidades de erros e estresse.

O principal objetivo deste trabalho foi adaptar a IU para dispositivos móveis baseado no perfil de usuário, com auxílio de ontologias, perfil de dispositivo, com uso de um vocabulário UAProf, e contexto do ambiente pervasivo, captado por sensores.

A partir dos testes realizados pelo caso de estudo, foi alcançado um *feedback* dos usuários quanto a adaptação da IU segundo os critérios do gráfico da figura 6. Quanto ao critério de facilidade de aprendizado, onde o sistema deve apresentar facilidade de uso, os resultados foram bons, obtendo uma média de 7,8 pontos dentre os 5 usuários. Quando medido o critério de facilidade de lembrar tarefas e ações sobre o sistema, os resultados foram os mesmos, com média entre todos os usuários de 7,8 pontos.

No ponto de avaliação sobre eficiência, onde o sistema deve desempenhar seu papel de forma correta e habilitar o usuário a atingir um alto grau de produtividade e desempenho em suas tarefas, a média obtida foi de 7,5 pontos, o que mostra que o sistema desempenha bem suas tarefas.

No momento da avaliação de eficácia e satisfação, onde por eficácia entende-se o sistema realizar a tarefa da melhor forma possível e por satisfação entende-se a percepção do usuário com o sistema e a forma com que a IU é adaptada, a média dos testes dos usuários foi de 7,1 pontos, o que demonstra que a adaptação realizada atingiu de forma satisfatória o objetivo. A média geral das avaliações, que foi de 7,5 pontos, nos remete que o resultado da adaptação da IU baseada na arquitetura PIDIM atingiu seu objetivo satisfatoriamente, com média superior a 7 de 10 pontos no total.

A fim de embasar alguns trabalhos futuros sobre adaptação de IU para dispositivos móveis, algumas alterações na abordagem deste trabalho poderão ocorrer em trabalhos subsequentes. Estas alterações podem ser a expansão da abordagem, tratando IU dinâmicas, especificação de maiores detalhes sobre a ontologia Perfil do Usuário, tentando absorver mais dados do usuário quanto puder, em questões como objetos utilizados pelo usuário, entre outros.

Referências

- Alliance, O. M. (2001). Especificação uaprof. <http://www.openmobilealliance.org/tech/affiliates/wap/wap-248-uaprof-20011020-a.pdf>. 2001. Disponível em.
- Dey, A. K. (2001). Understanding and using context. *Personal Ubiquitous Comput.*, 5:4-7.
- dos Santos, R. C. (2008). Systems usability evaluation metrics review. *Global Business And Technology Association Conference (GBATA)*.
- Kane, S. K., Wobbrock, J. O., and Smith, I. E. (2008). Getting off the treadmill: evaluating walking user interfaces for mobile devices in public spaces. In *Proceedings of the*

10th international conference on Human computer interaction with mobile devices and services, pages 109–118, New York, NY, USA. ACM.

Kiss, C. (2007). Composite capability/preference profiles: Structure and vocabularies. <http://www.w3.org/TR/2007/WD-CCPP-struct-vocab2-20070430/>. 2007. Disponível em.

Shneiderman, B. (2000). Universal usability. *Commun. ACM*, 43:84–91.

Studer, R., Benjamins, V. R., and Fensel, D. (1998). Knowledge engineering: Principles and methods. *Data Knowl. Eng.*, pages 161–197.

Yamin, A. C. (2004). *Arquitetura para um Ambiente de Grade Computacional Direcionado às Aplicações Distribuídas, Móveis e Conscientes do Contexto da Computação Pervasiva*. Tese de doutorado, II/UFRGS, Porto Alegre, RS, Brasil.

Ye, J., Dobson, S., and McKeever, S. (2012). Review: Situation identification techniques in pervasive computing: A review. *Pervasive Mob. Comput.*, pages 36–66.